greatlearning
Power Ahead

ADVANCE STATISTICS PROJECT

Pooja Kabadi
PGP-DSBA Online
Batch- A4
14-11-2021

Table of Contents:

1.1 State the null and the alternate hypothesis for conducting one-way ANOVA for both Education and Occupation individually. 4

1.2 Perform one-way ANOVA for Education with respect to the variable 'Salary'. State whether the null hypothesis is accepted or rejected based on the ANOVA results. 4

1.3 Perform one-way ANOVA for variable Occupation with respect to the variable 'Salary'. State whether the null hypothesis is accepted or rejected based on the ANOVA results. 4

1.4 If the null hypothesis is rejected in either (1.2) or in (1.3), find out which class means are significantly different. Interpret the result. 5

1.5 What is the interaction between the two treatments? Analyse the effects of one variable on the other (Education and Occupation) with the help of an interaction plot. 5

1.6 Perform a two-way ANOVA based on the Education and Occupation (along with their interaction Education*Occupation) with the variable 'Salary'. State the null and alternative hypotheses and state your results. How will you interpret this result? 6

1.7 Explain the business implications of performing ANOVA for this particular case study. 7

2.1 Perform Exploratory Data Analysis [both univariate and multivariate analysis to be performed]. What insight do you draw from the EDA? 7

2.2 Is scaling necessary for PCA in this case? Give justification and perform scaling. 16

PCA 18

2.3 Comment on the comparison between the covariance and the correlation matrices from this data. [on scaled data] 18

2.4 Check the dataset for outliers before and after scaling. What insight do you derive here? 21

2.5 Extract the eigenvalues and eigenvectors. [Using Sklearn PCA Print Both] 22

2.6 Perform PCA and export the data of the Principal Component (eigenvectors) into a data frame with the original features. 24

2.7 Write down the explicit form of the first PC (in terms of the eigenvectors. Use values with two places of decimals only). [hint: write the linear equation of PC in terms of eigenvectors and corresponding features] 24

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? 24

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] 28

List of Figures:

Figure 1: Interaction plot between 'Education' and 'Occupation'…….5

Figure 2: Boxplot & Displot of Apps…….10

Figure 3: Boxplot & Displot of Accept……..10

Figure 4: Boxplot & Displot of Enroll……....10

Figure 5: Boxplot & Displot of Top10perc……11

Figure 6: Boxplot & Displot of Top25perc……11

Figure 7: Boxplot & Displot of F. Undergrad……11

Figure 8: Boxplot & Displot of P. Undergrad……11

Figure 9: Boxplot & Displot of Outstate……12

Figure 10: Boxplot & Displot of Room. Board…….12

Figure 11: Boxplot & Displot of Books……12
Figure 12: Boxplot & Displot of Personal……..12
Figure 13: Boxplot & Displot of PhD……...…..13
Figure 14: Boxplot & Displot of Terminal……13
Figure 15: Boxplot & Displot of S.F. Ratio……...13
Figure 16: Boxplot & Displot of perc. Alumni……...13
Figure 17: Boxplot & Displot of Expend……...14
Figure 18: Boxplot & Displot of Grad.Rate……...14
Figure 19: Pair plot of all the numeric variables) ……...15
Figure 20: Correlation Heatmap……16
Figure 21: Boxplot Numeric values before scaling dataset……....21
Figure 22: Boxplot Numeric values after scaling dataset…….....21
Figure 23: Scree Plot……..25
Figure 24: plot the component loading on a heatmap……26
Figure 25: Heatmap of final principal components……27

List of Tables:

Table 1: Inferences of Univariate Data visualization……14

Problem 1A [Salary Dataset ANOVA Analysis]:

Salary is hypothesized to depend on educational qualification and occupation. To understand the dependency, the salaries of 40 individuals [SalaryData.csv] are collected and each person's educational qualification and occupation are noted. Educational qualification is at three levels, High school graduate, Bachelor, and Doctorate. Occupation is at four levels, Administrative and clerical, Sales, Professional or specialty, and Executive or managerial. A different number of observations are in each level of education – occupation combination.

[Assume that the data follows a normal distribution. In reality, the normality assumption may not always hold if the sample size is small]

Exploratory Data Analysis:

	Education	Occupation	Salary
0	Doctorate	Adm-clerical	153197
1	Doctorate	Adm-clerical	115945
2	Doctorate	Adm-clerical	175935
3	Doctorate	Adm-clerical	220754
4	Doctorate	Sales	170769
5	Doctorate	Sales	219420
6	Doctorate	Sales	237920
7	Doctorate	Sales	160540
8	Doctorate	Sales	180934
9	Doctorate	Prof-specialty	248156

Checking for Null-values:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Education   40 non-null     object
 1   Occupation  40 non-null     object
 2   Salary      40 non-null     int64 
dtypes: int64(1), object(2)
memory usage: 1.1+ KB
```

Descriptive Data Analysis

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Education	40	3	Doctorate	16	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Occupation	40	4	Prof-specialty	13	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Salary	40.0	NaN	NaN	NaN	162186.875	64860.407506	50103.0	99897.5	169100.0	214440.75	260151.0

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

Education:

Null Hypothesis (H0): The mean Salary is the same across all the 3 categories of Education (High school graduate, Bachelor, and Doctorate)

Alternate Hypothesis (Ha): The mean Salary is different in at least one category of Education.

α = 0.05

Occupation:

Null Hypothesis (H0): The mean Salary is the same across all the 4 categories of Occupation (Administrative & clerical, Sales, Professional or specialty, and Executive or managerial)

Alternate Hypothesis (Ha): The mean salary is different in at least one category of Occupation.

α = 0.05

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

Education:

```
formula = 'Salary ~ C(Education)'
model = ols(formula, df_salary).fit()
aov_table = anova_lm(model)
print(aov_table)
```

```
                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 = 1.257709e-08 is less than the significance level (α = 0.05), **we can reject the null hypothesis and conclude that there is a significant difference in the mean salaries for at least one category of education.**

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

Occupation:

```
formula = 'Salary ~ C(Occupation)'
model = ols(formula, df_salary).fit()
aov_table = anova_lm(model)
print(aov_table)
```

```
                 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 = 0.458508 is greater than the significance level (α = 0.05), **we fail to reject the null hypothesis (i.e., we accept Ha) and conclude that there is no significant difference in the mean salaries across the 4 categories of occupation.**

1.4 If the null hypothesis is rejected in either (1.2) or in (1.3), find out which class means are significantly different. Interpret the result.

The null hypothesis is rejected in (1.2). Hence, for the further analysis to check which class means are significantly different for Education. The analysis for this is carried out by Tukey's HSD test. This test is conducted only when the null hypothesis is rejected. It basically compares all possible pairs of means

```
          Multiple Comparison of Means - Tukey HSD, FWER=0.05
=============================================================================
  group1     group2     meandiff   p-adj     lower        upper     reject
-----------------------------------------------------------------------------
 Bachelors  Doctorate   43274.0667 0.0146    7541.1439  79006.9894   True
 Bachelors    HS-grad  -90114.1556  0.001 -132035.1958 -48193.1153   True
 Doctorate    HS-grad -133388.2222  0.001 -174815.0876 -91961.3569   True
```

- The column of interest for us is the "Reject" column. Since all three rows say "True", we conclude that Salary varies for individuals with different levels of Education. This further confirms our conclusion from ANOVA.
- Using the data above, we can see that Doctorate level of education and HS-grad level of education have the highest difference in class means.

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

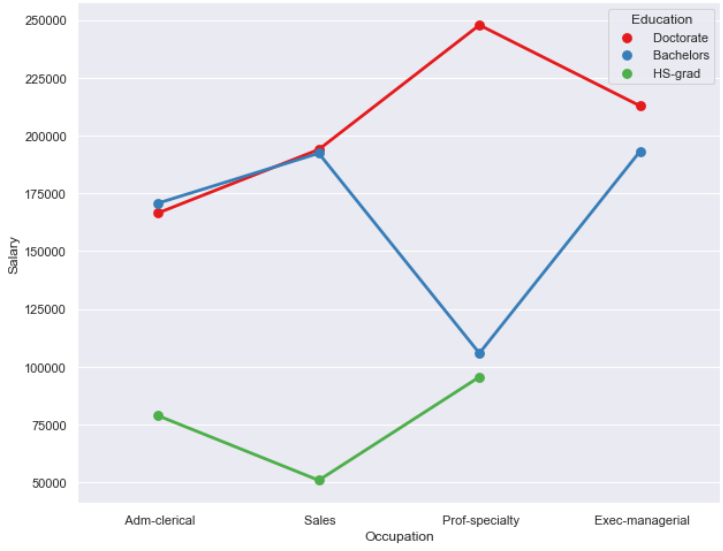


Figure 2: Interaction plot between 'Education' and 'Occupation'

The interaction plot shows that there is significant amount of interaction between the categorical variables, Education and Occupation.

The following are some of the observations from the interaction plot:

- People with education qualification as High school graduate earn the minimum salaries.
- People with education qualification as Doctorate earn the maximum salaries.

- Professional or specialty people with education qualification as Doctorate earn maximum salaries and people with education as HS-Grad earn the minimum.
- According to plot, there are no people with education qualification as High school graduate who hold Exec-managerial occupation.
- People with education qualification as Bachelor's degree with occupation as Professional or specialty earn lesser than people with education qualification as Bachelor's degree with occupations as Administrative and clerical and Sales.
- Salespeople with Bachelors or Doctorate education earn the same salaries and people with high school graduate for same occupation earns the least.
- We can see the huge difference in Salaries in Professional or specialty occupation between those of doctorate and bachelor's graduates.

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

Null Hypothesis:

*H*0: There is no interaction effect between the two independent variables, Education and Occupation w.r.t mean Salary (The effect of the independent variable 'Education' on the mean 'Salary' does not depend on the effect of the other independent variable 'Occupation')

Alternate Hypothesis:

*H*a: There is an interaction between the independent variable 'Education' and the independent variable 'Occupation' on the mean 'Salary'.

```
formula = 'Salary ~ C(Education) + C(Occupation) + C(Education):C(Occupation)'
model = ols(formula, df_salary).fit()
aov_table = anova_lm(model)
print(aov_table)
```

```
                             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
```

From the table, we see that there is a significant amount of interaction between the variables, Education and Occupation.

As p value = 2.232500e-05 is lesser than the significance level (alpha = 0.05), we reject the null hypothesis.

Thus, we see that there is an interaction between Education and Occupation on the mean salary.

From the ANOVA method and the interaction plot, we clearly see that people with education as Doctorate draw the maximum salaries and people with education HS-grad earn the least. Thus, conclude that Salary is dependent on educational qualifications and occupation.

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

- The objective of using ANOVA for this case study is to determine whether Educational qualifications and Occupation are dependent on Salary variable individually and together.
- We get to understand the relationship between Education and Occupation categories as well as its interaction and dependency on the Salary variable.
- From the results of ANOVA performed on Salary with respect to Education, we conclude that Salary is dependent on Education. It is different for different levels of education.
- From the results on ANOVA performed on Salary with respect to Occupation, we conclude that mean Salary is same for all categories of Occupation.
- From ANOVA performed on Salary with respect to Education and Occupation including the interaction effect of the two treatments, we are able to conclude that the salary is different for different levels of Education and Occupation.
- The main variable affecting Salary is Education
- This means that the organization or firm performing these tests have to inspect the results and inferences and take required decisions whether really Education and Occupation backgrounds affect Salary or not.

Problem 2 [PCA on Education Dataset]:

The dataset Education - Post 12th Standard.csv contains information on various colleges. You are expected to do a Principal Component Analysis for this case study according to the instructions given. The data dictionary of the 'Education - Post 12th Standard.csv' can be found in the following file: Data Dictionary.xlsx.

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

Exploratory Data Analysis:

	Names	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
0	Abilene Christian University	1660	1232	721	23	52	2885	537	7440	3300	450	2200	70	78	18.1	12	7041	60
1	Adelphi University	2186	1924	512	16	29	2683	1227	12280	6450	750	1500	29	30	12.2	16	10527	56
2	Adrian College	1428	1097	336	22	50	1036	99	11250	3750	400	1165	53	66	12.9	30	8735	54
3	Agnes Scott College	417	349	137	60	89	510	63	12960	5450	450	875	92	97	7.7	37	19016	59
4	Alaska Pacific University	193	146	55	16	44	249	869	7560	4120	800	1500	76	72	11.9	2	10922	15
5	Albertson College	587	479	158	38	62	678	41	13500	3335	500	675	67	73	9.4	11	9727	55
6	Albertus Magnus College	353	340	103	17	45	416	230	13290	5720	500	1500	90	93	11.5	26	8861	63
7	Albion College	1899	1720	489	37	68	1594	32	13868	4826	450	850	89	100	13.7	37	11487	73
8	Albright College	1038	839	227	30	63	973	306	15595	4400	300	500	79	84	11.3	23	11644	80
9	Alderson-Broaddus College	582	498	172	21	44	799	78	10468	3380	660	1800	40	41	11.5	15	8991	52

Checking for Null-values:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 777 entries, 0 to 776
Data columns (total 18 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Names        777 non-null    object 
 1   Apps         777 non-null    int64  
 2   Accept       777 non-null    int64  
 3   Enroll       777 non-null    int64  
 4   Top10perc    777 non-null    int64  
 5   Top25perc    777 non-null    int64  
 6   F.Undergrad  777 non-null    int64  
 7   P.Undergrad  777 non-null    int64  
 8   Outstate     777 non-null    int64  
 9   Room.Board   777 non-null    int64  
 10  Books        777 non-null    int64  
 11  Personal     777 non-null    int64  
 12  PhD          777 non-null    int64  
 13  Terminal     777 non-null    int64  
 14  S.F.Ratio    777 non-null    float64
 15  perc.alumni  777 non-null    int64  
 16  Expend       777 non-null    int64  
 17  Grad.Rate    777 non-null    int64  
dtypes: float64(1), int64(16), object(1)
memory usage: 109.4+ KB
```

Descriptive Data Analysis:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Names	777	777	University of North Carolina at Charlotte	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Apps	777.0	NaN	NaN	NaN	3001.638353	3870.201484	81.0	776.0	1558.0	3624.0	48094.0
Accept	777.0	NaN	NaN	NaN	2018.804376	2451.113971	72.0	604.0	1110.0	2424.0	26330.0
Enroll	777.0	NaN	NaN	NaN	779.972973	929.17619	35.0	242.0	434.0	902.0	6392.0
Top10perc	777.0	NaN	NaN	NaN	27.558559	17.640364	1.0	15.0	23.0	35.0	96.0
Top25perc	777.0	NaN	NaN	NaN	55.796654	19.804778	9.0	41.0	54.0	69.0	100.0
F.Undergrad	777.0	NaN	NaN	NaN	3699.907336	4850.420531	139.0	992.0	1707.0	4005.0	31643.0
P.Undergrad	777.0	NaN	NaN	NaN	855.298584	1522.431887	1.0	95.0	353.0	967.0	21836.0
Outstate	777.0	NaN	NaN	NaN	10440.669241	4023.016484	2340.0	7320.0	9990.0	12925.0	21700.0
Room.Board	777.0	NaN	NaN	NaN	4357.526384	1096.696416	1780.0	3597.0	4200.0	5050.0	8124.0
Books	777.0	NaN	NaN	NaN	549.380952	165.10536	96.0	470.0	500.0	600.0	2340.0
Personal	777.0	NaN	NaN	NaN	1340.642214	677.071454	250.0	850.0	1200.0	1700.0	6800.0
PhD	777.0	NaN	NaN	NaN	72.660232	16.328155	8.0	62.0	75.0	85.0	103.0
Terminal	777.0	NaN	NaN	NaN	79.702703	14.722359	24.0	71.0	82.0	92.0	100.0
S.F.Ratio	777.0	NaN	NaN	NaN	14.089704	3.958349	2.5	11.5	13.6	16.5	39.8
perc.alumni	777.0	NaN	NaN	NaN	22.743887	12.391801	0.0	13.0	21.0	31.0	64.0
Expend	777.0	NaN	NaN	NaN	9660.171171	5221.76844	3186.0	6751.0	8377.0	10830.0	56233.0
Grad.Rate	777.0	NaN	NaN	NaN	65.46332	17.17771	10.0	53.0	65.0	78.0	118.0

Data Pre-processing:

Checking for duplicates in this dataset:

```
# Are there any duplicates?
dups = df.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
df[dups]
```

```
Number of duplicate rows = 0
```

Checking for Null & missing values, the Skewness and Kurtosis in the dataset:

```
df.isnull().sum()
```

```
Names          0
Apps           0
Accept         0
Enroll         0
Top10perc      0
Top25perc      0
F.Undergrad    0
P.Undergrad    0
Outstate       0
Room.Board     0
Books          0
Personal       0
PhD            0
Terminal       0
S.F.Ratio      0
perc.alumni    0
Expend         0
Grad.Rate      0
dtype: int64
```

```
df.skew()
```

```
Apps           3.723750
Accept         3.417727
Enroll         2.690465
Top10perc      1.413217
Top25perc      0.259340
F.Undergrad    2.610458
P.Undergrad    5.692353
Outstate       0.509278
Room.Board     0.477356
Books          3.485025
Personal       1.742497
PhD           -0.768170
Terminal      -0.816542
S.F.Ratio      0.667435
perc.alumni    0.606891
Expend         3.459322
Grad.Rate     -0.113777
dtype: float64
```

```
df.kurt()
```

```
Apps           26.774253
Accept         18.938099
Enroll          8.831544
Top10perc       2.208065
Top25perc      -0.564121
F.Undergrad     7.696586
P.Undergrad    55.034518
Outstate       -0.413832
Room.Board     -0.187553
Books          28.333097
Personal        7.124017
PhD             0.564773
Terminal        0.242019
S.F.Ratio       2.561209
perc.alumni    -0.096807
Expend         18.771500
Grad.Rate      -0.205226
dtype: float64
```

Observations:

- Dataset has 18 columns and 777 rows
- The entire dataset is of integer data type. However, we understand that 'Name' column is categorical in nature
- There are no null values and duplicate rows in the dataset.
- From the statistical summary, we can interpret that the data is not normally distributed considering the fact that the mean and the median (50%) are significantly different from each other.
- A new data frame without the “Names” column is created for further analysis.
- Skewness and Kurtosis is also calculated for each column, Data with high skewness indicates lack of symmetry and high value of kurtosis indicates heavily tailed data.
- This indicates the presence of large number of outliers in the data.
- A normal distribution is ideal hence detection of skewness and outliers is an important.
- Here we can see that, the PhD and Grad. Rate columns are in the percentage values and the maximum values in both the columns are exceeding 100. It is necessary to treat the anomalies.

Anomalies Check:

Here we can see that, the **PhD** and **Grad. Rate** columns are in the percentage values and the maximum values in both the columns are exceeding 100. It is necessary to treat the anomalies.

```
df[df.PhD > 100]
```

	Names	Apps	Accept	Enroll	Top10perc	Top25perc	F_Undergrad	P_Undergrad	Outstate	Room_Board	Books	Personal	PhD	Terminal	S_F.Ratio
582	Texas A&M University at	529	481	243	22	47	1206	134	4860	3122	600	650	103	88	17.4

```
df.PhD .replace(to_replace= 103, value= 100 , inplace= True)
```

```
df[df.Grad_Rate > 100]
```

Enroll	Top10perc	Top25perc	F_Undergrad	P_Undergrad	Outstate	Room_Board	Books	Personal	PhD	Terminal	S_F.Ratio	perc_alumni	Expend	Grad_Rate
527	9	35	1010	12	9384	4840	600	500	22	47	14.3	20	7697	118

```
df.Grad_Rate.replace(to_replace= 118, value= 100 , inplace= True)
```

Data Visualization

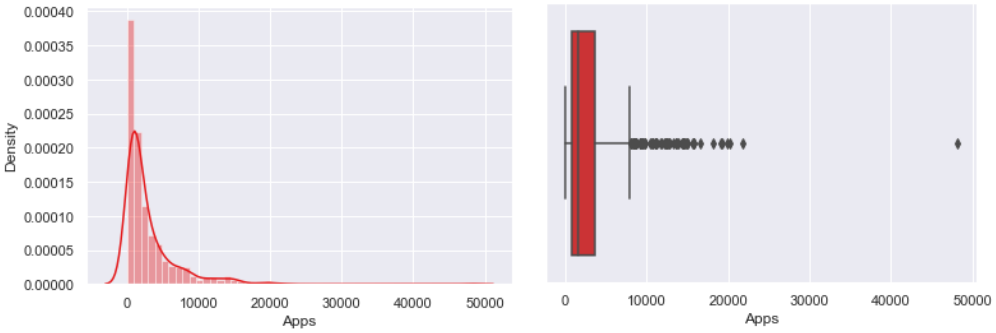
Univariate Analysis:

Let us define a function 'univariate Analysis numeric' to display information as part of univariate analysis of numeric variables. The function will accept column name and number of bins as arguments.

The function will display the statistical description of the numeric variable, histogram or distplot to view the distribution and the box plot to view 5-point summary and outliers if any.

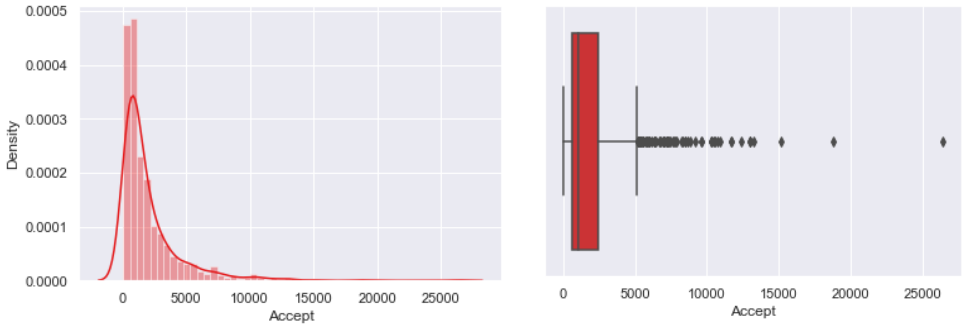
Apps: Number of applications received *(Figure 2: Boxplot & Displot of Apps)*

```
Description of Apps
-----------------------
count      777.000000
mean      3001.638353
std       3870.201484
min         81.000000
25%        776.000000
50%       1558.000000
75%       3624.000000
max      48094.000000
```



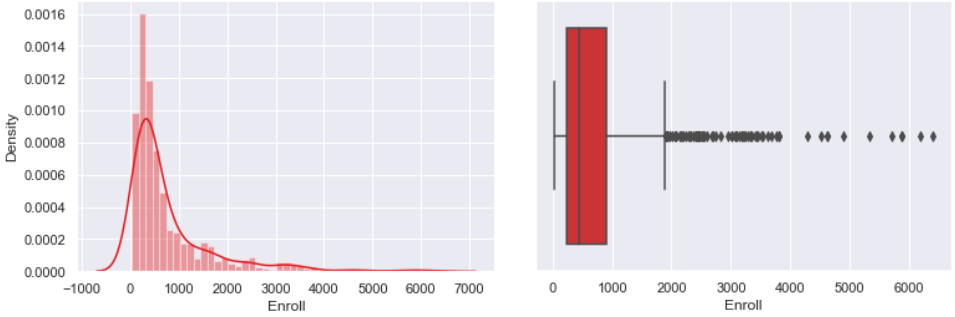
Accept: Number of applications accepted *(Figure 3: Boxplot & Displot of Accept)*

```
Description of Accept
----------------------
count      777.000000
mean      2018.804376
std       2451.113971
min         72.000000
25%        604.000000
50%       1110.000000
75%       2424.000000
max      26330.000000
```



Enroll: Number of new students enrolled *(Figure 4: Boxplot & Displot of Enroll)*

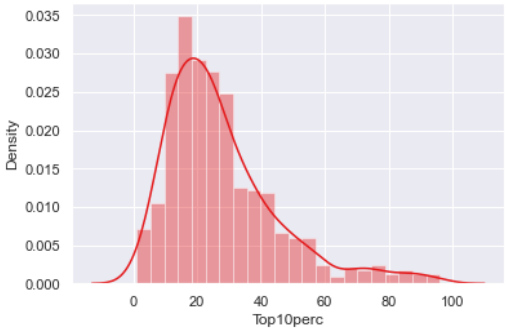
```
Description of Enroll
----------------------
count      777.000000
mean       779.972973
std        929.176190
min         35.000000
25%        242.000000
50%        434.000000
75%        902.000000
max       6392.000000
```

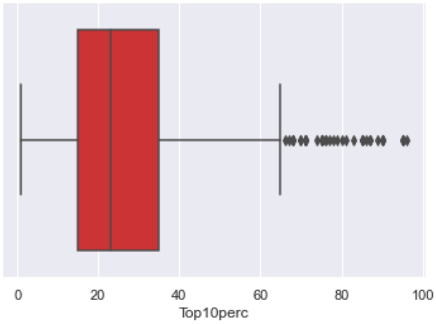


Top10perc: Percentage of new students from top 10% of Higher Secondary class

(Figure 5: Boxplot & Displot of Top10perc)

```
Description of Top10perc
-------------------------
count    777.000000
mean      27.558559
std       17.640364
min        1.000000
25%       15.000000
50%       23.000000
75%       35.000000
max       96.000000
```

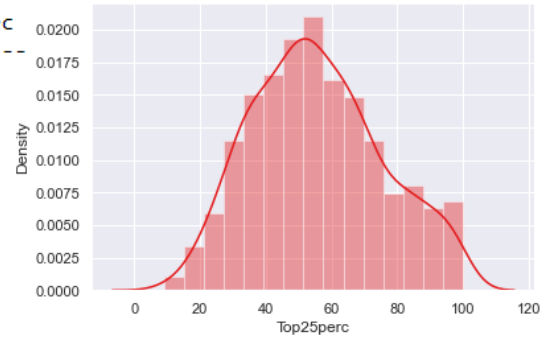


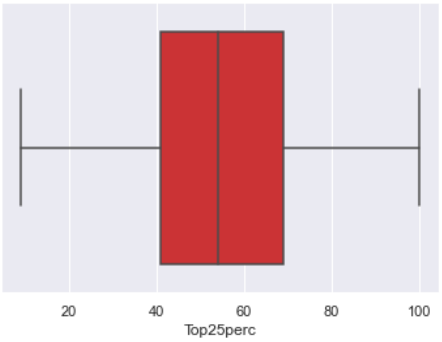


Top25perc: Percentage of new students from top 25% of Higher Secondary class

(Figure 6: Boxplot & Displot of Top25perc)

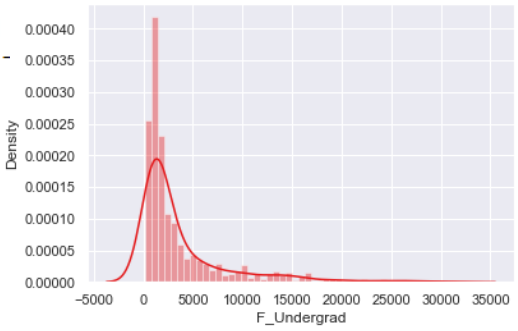
```
Description of Top25perc
-------------------------
count    777.000000
mean      55.796654
std       19.804778
min        9.000000
25%       41.000000
50%       54.000000
75%       69.000000
max      100.000000
```

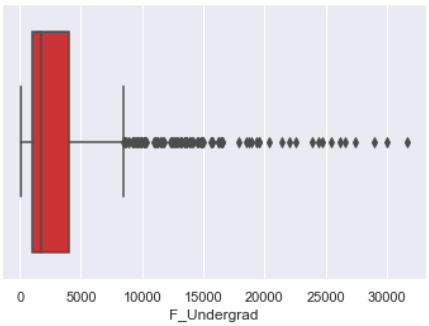




F. Undergrad: Number of full-time undergraduate students *(Figure 7: Boxplot & Displot of F.Undergrad)*

```
Description of F_Undergrad
--------------------------
count      777.000000
mean      3699.907336
std       4850.420531
min        139.000000
25%        992.000000
50%       1707.000000
75%       4005.000000
max      31643.000000
```

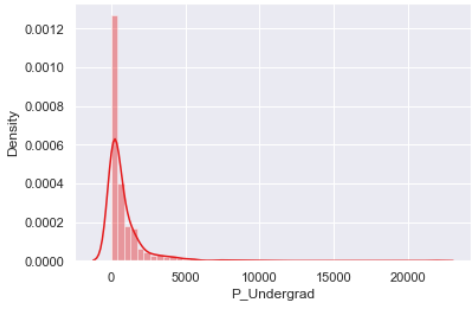


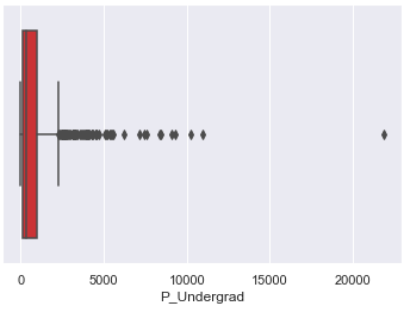


P. Undergrad: Number of part-time undergraduate students

(Figure 8: Boxplot & Displot of P.Undergrad)

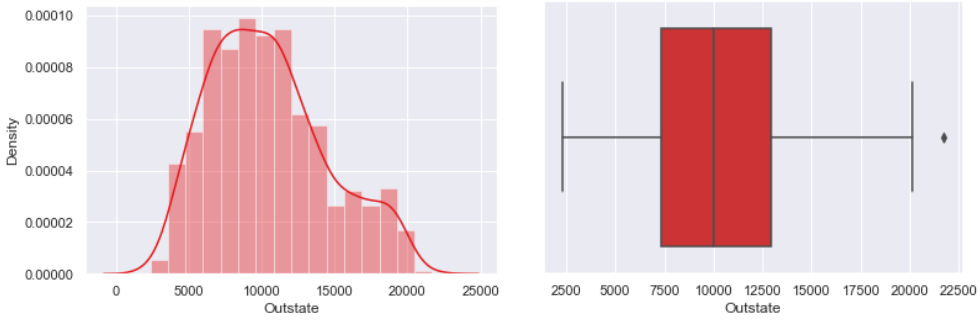
```
Description of P_Undergrad
--------------------------
count      777.000000
mean       855.298584
std       1522.431887
min          1.000000
25%         95.000000
50%        353.000000
75%        967.000000
max      21836.000000
```





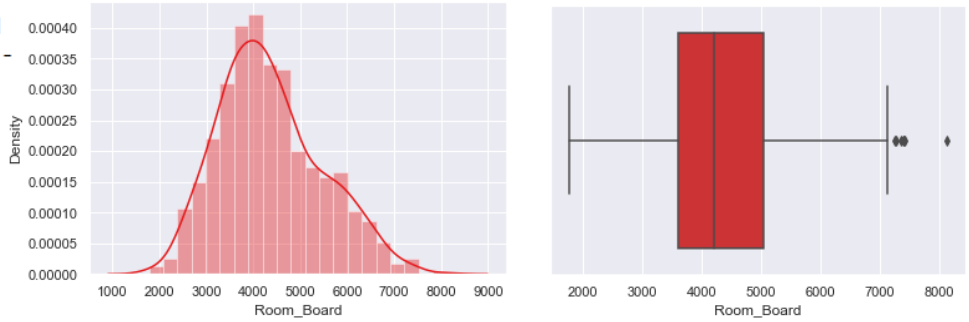
Outstate: Number of students for whom the particular college or university is Out-of-state tuition *(Figure 9: Boxplot & Displot of Outstate)*

```
Description of Outstate
-----------------------
count      777.000000
mean     10440.669241
std       4023.016484
min       2340.000000
25%       7320.000000
50%       9990.000000
75%      12925.000000
max      21700.000000
```



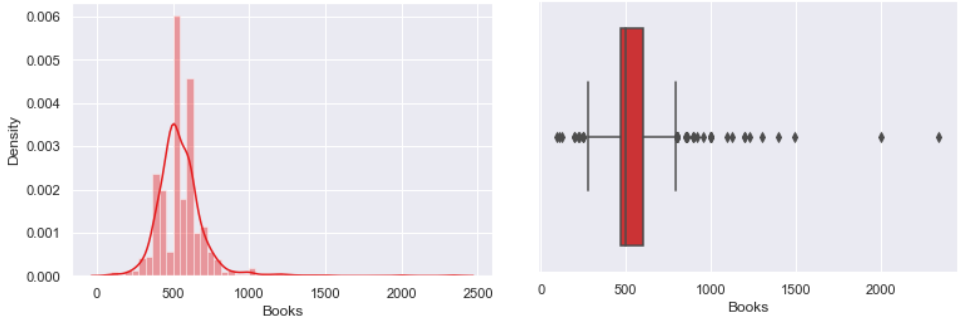
Room. Board: Cost of Room and board *(Figure 10: Boxplot & Displot of Room. Board)*

```
Description of Room_Board
-------------------------
count     777.000000
mean     4357.526384
std      1096.696416
min      1780.000000
25%      3597.000000
50%      4200.000000
75%      5050.000000
max      8124.000000
```



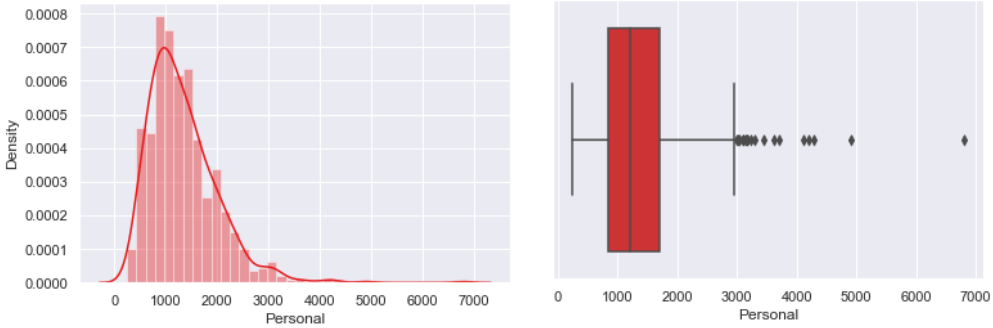
Books: Estimated book costs for a student *(Figure 11: Boxplot & Displot of Books)*

```
Description of Books
--------------------
count     777.000000
mean      549.380952
std       165.105360
min        96.000000
25%       470.000000
50%       500.000000
75%       600.000000
max      2340.000000
```



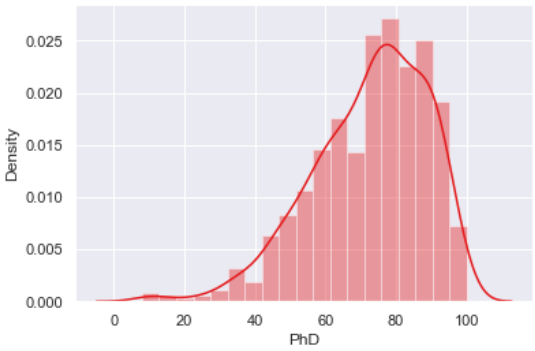
Personal: Estimated personal spending for a student *(Figure 12: Boxplot & Displot of Personal)*

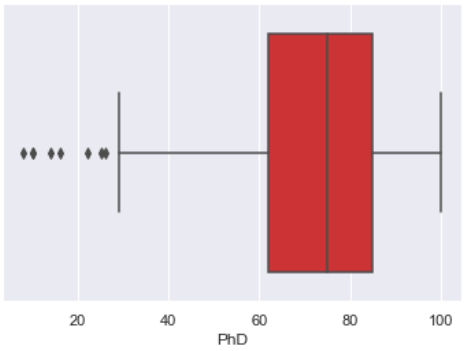
```
Description of Personal
-----------------------
count     777.000000
mean     1340.642214
std       677.071454
min       250.000000
25%       850.000000
50%      1200.000000
75%      1700.000000
max      6800.000000
```



PhD: Percentage of faculties with Ph.D.'s *(Figure 13: Boxplot & Displot of PhD)*

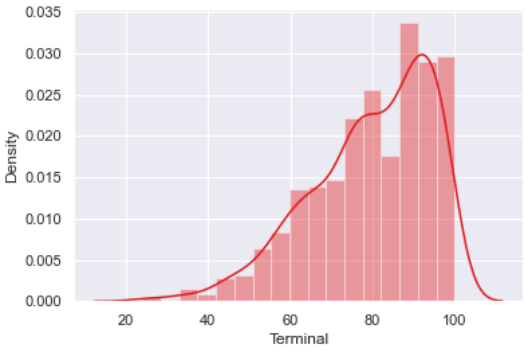
```
Description of PhD
----------------------
count     777.000000
mean       72.656371
std        16.321324
min         8.000000
25%        62.000000
50%        75.000000
75%        85.000000
max       100.000000
```

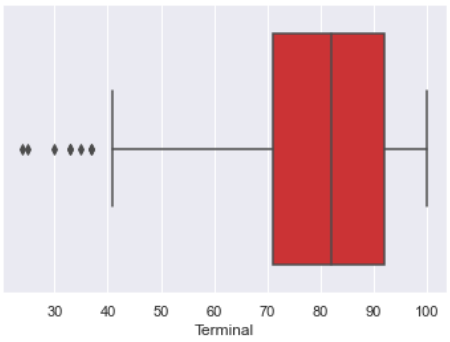




Terminal: Percentage of faculties with terminal degree *(Figure 14: Boxplot & Displot of Terminal)*

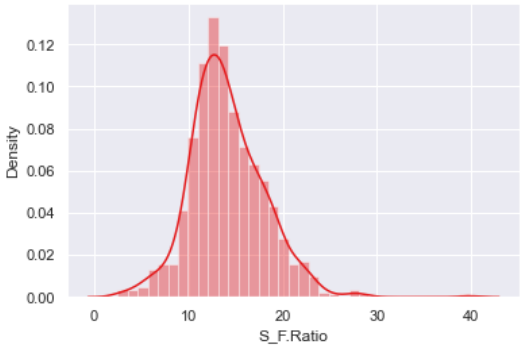
```
Description of Terminal
-----------------------
count     777.000000
mean       79.702703
std        14.722359
min        24.000000
25%        71.000000
50%        82.000000
75%        92.000000
max       100.000000
```

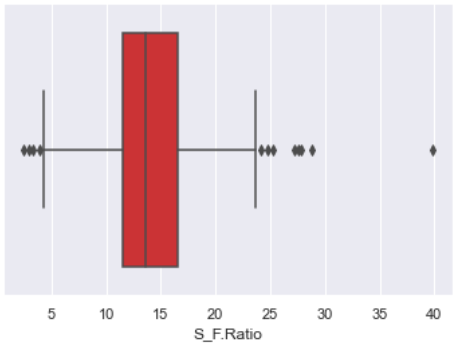




S.F. Ratio: Student/faculty ratio *(Figure 15: Boxplot & Displot of S.F. Ratio)*

```
Description of S_F.Ratio
------------------------
count     777.000000
mean       14.089704
std         3.958349
min         2.500000
25%        11.500000
50%        13.600000
75%        16.500000
max        39.800000
```

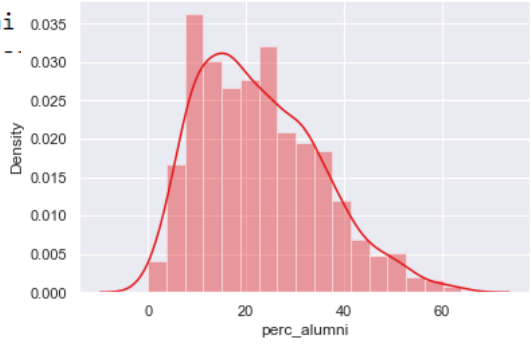


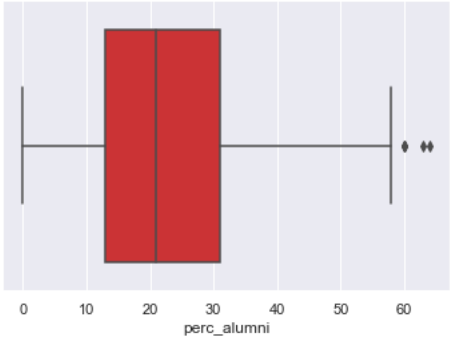


---

perc. Alumni: Percentage of alumni who donate *(Figure 16: Boxplot & Displot of perc. Alumni)*

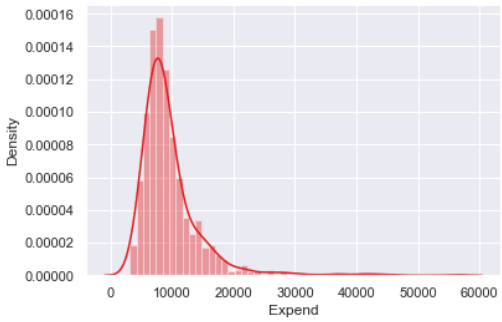
```
Description of perc_alumni
--------------------------
count     777.000000
mean       22.743887
std        12.391801
min         0.000000
25%        13.000000
50%        21.000000
75%        31.000000
max        64.000000
```

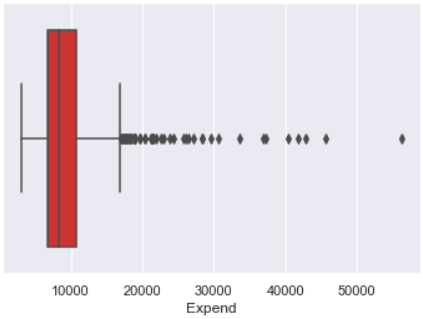




Expend: The Instructional expenditure per student *(Figure 17: Boxplot & Displot of Expend)*

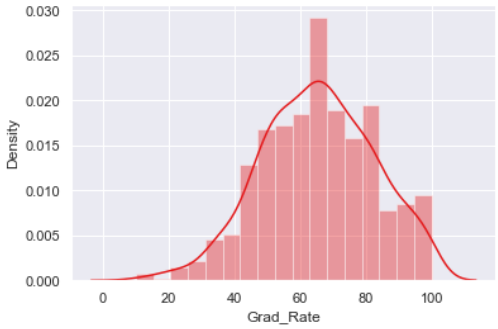
```
Description of Expend
-----------------------
count      777.000000
mean      9660.171171
std       5221.768440
min       3186.000000
25%       6751.000000
50%       8377.000000
75%      10830.000000
max      56233.000000
```

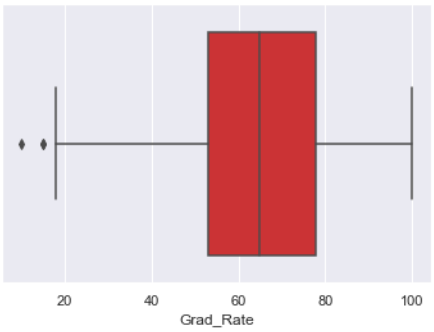




Grad.Rate: Graduation rate *(Figure 18: Boxplot & Displot of Grad.Rate)*

```
Description of Grad_Rate
--------------------------
count    777.000000
mean      65.440154
std       17.118804
min       10.000000
25%       53.000000
50%       65.000000
75%       78.000000
max      100.000000
```





Observations: *(Table 1: Inferences of Univariate Data visualization)*

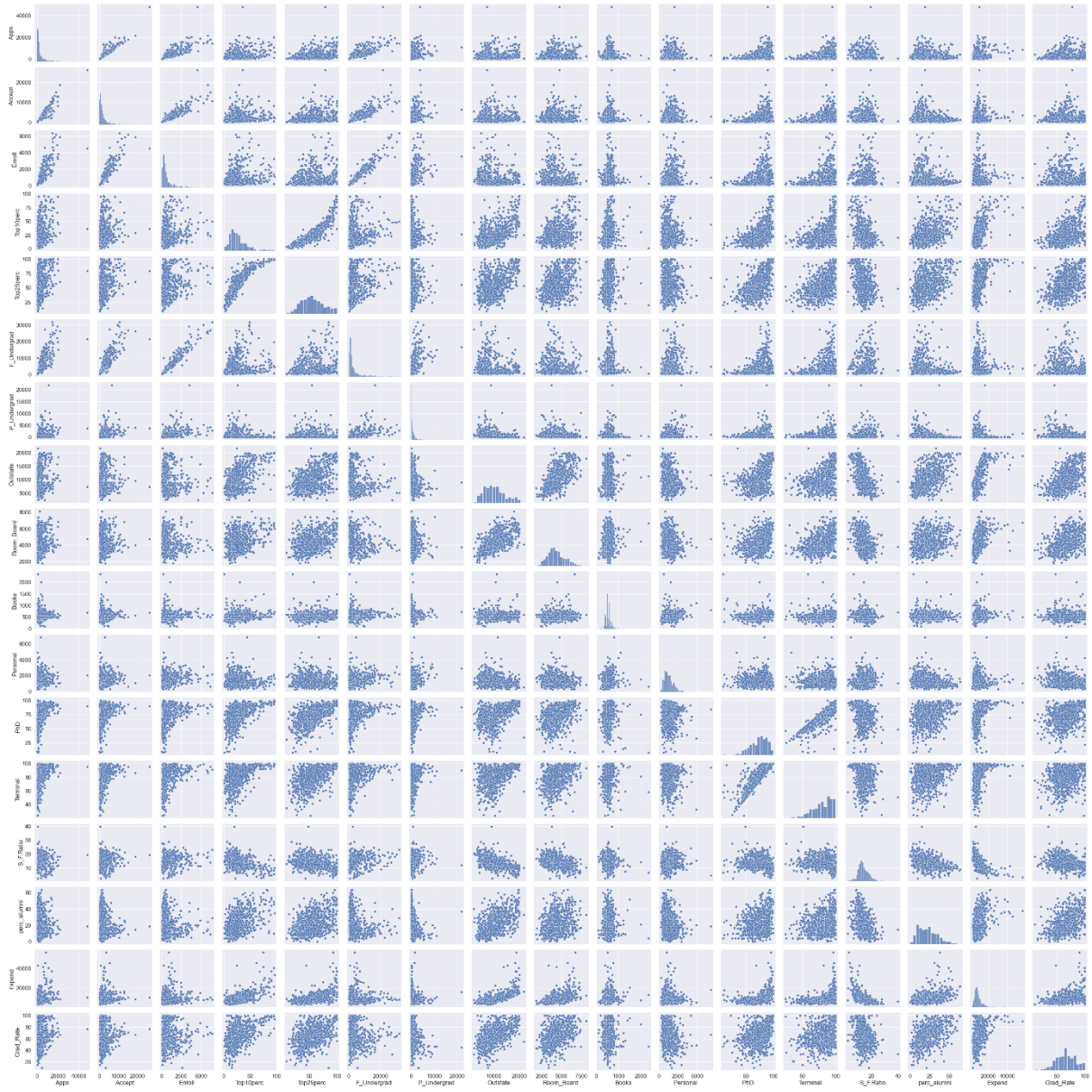
Sl. No	Column Name	Normal Distribution (Yes / No)	Skewness Value	Skew Type (Yes / No)	Outliers (Yes / No)
1	Apps	No	3.723	Yes (Right)	Yes
2	Accept	No	3.417	Yes (Right)	Yes
3	Enroll	No	2.690	Yes (Right)	Yes
4	Top10perc	No	1.413	Yes (Right)	Yes
5	Top25perc	Tending to normal	0.259	Yes	No
6	F. Undergrad	No	2.610	Yes (Right)	Yes
7	P. Undergrad	No	5.692	Yes (Right)	Yes
8	Outstate	Tending to normal	0.509	Yes	Yes
9	Room. Board	Tending to normal	0.477	Yes	Yes
10	Books	No	3.485	Yes (Right)	Yes
11	Personal	No	1.742	Yes (Right)	Yes
12	PhD	No	-0.768	Yes (Left)	Yes
13	Terminal	No	-0.816	Yes (Left)	Yes
14	S.F. Ratio	Tending to normal	0.667	Yes	Yes
15	Perc. Alumni	Tending to normal	0.606	Yes	Yes
16	Expend	No	3.459	Yes (Right)	Yes
17	Grad. Rate	Tending to normal	-0.113	Yes (Right)	Yes

Observations

- On performing Univariate Analysis, we use Box plots and distplot to check for outliers and skewness.
- All the columns have outliers except the Top25perc feature.
- Variables like Apps, Accept, Enroll, Top10perc, F. Undergrad, P. Undergrad, Books, Personal and expend are right skewed whereas variables like PhD, Terminal, Grad.Rate are left skewed.
- Variables like Top25perc, Outstate, Room. Board, and S.F. Ratio, perc. Alumni seem to follow normal distribution.

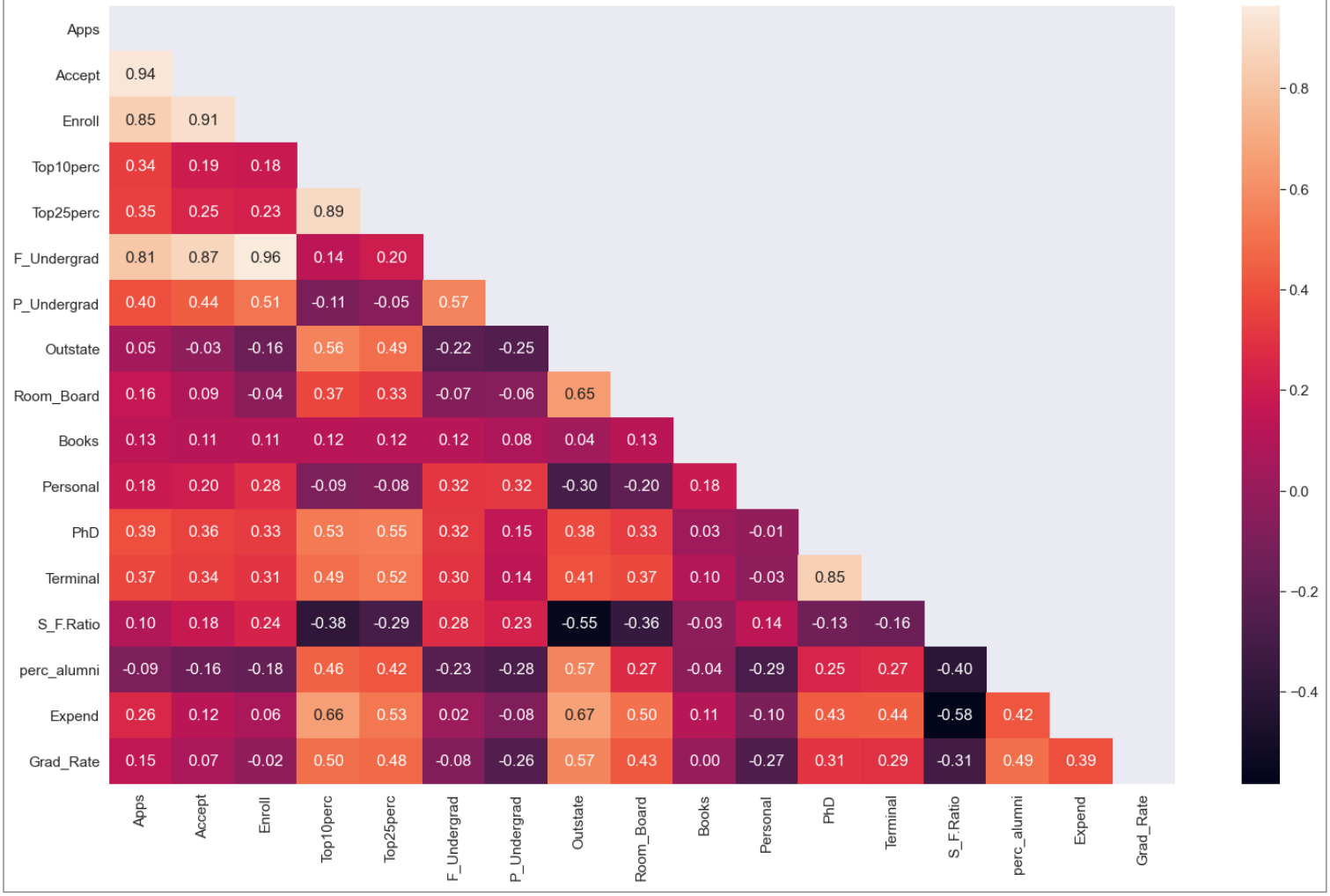
Multivariate Analysis

Pair plot:



(Figure 19: Pair plot of all the numeric variables)

Heat map:



(Figure 20: Correlation Heatmap)

Each square shows the correlation between the variables on each axis. Correlation ranges from -1 to +1. Values closer to zero indicate that there is no linear trend between the two variables. Closer to 1 the correlation is, more positively correlated are the variables that is as one increases so does the other. A correlation closer to -1 is similar, but instead of both increasing one variable will decrease as the other increases

Inferences

- The application with top 10, 25 of higher secondary class, outstate, room board, books, personal, PhD, terminal, S.F ratio, expenditure and Graduation ratio are positively correlated.
- We can see that, the application variable is highly positively correlated with application accepted, students enrolled and full-time graduates. So, this relationship gives the insights on when student submits the application it is accepted and the student is enrolled as fulltime graduate.
- We can see the negative correlation between application and percentage of alumni. This indicates that not all students are part of alumni of their college or university.

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

Yes, scaling is necessary for PCA in this case.
We have in total of 17 numerical variables in our dataset. Different variables have different meanings and have different units of measurements.

- Application, Accepted Application, Enrolled, F. Undergrad, P. Undergrad, Outstate - These are number of students

- Top 10 percent, Top 25 percent - These are the percentage of students
- Room board, Books, Personal and Expend - These are values which are associated with money
- PhD, SF Ratio, Terminal, Percentage of Alumni, Grad Rate - These are percentage values

PCA is a variance maximising exercise. It calculates a new projection of the dataset and the axis for this is dependent on the standard deviation of the data. Scaling helps to standardize the data. Basically, the scaling techniques scales the data in such a way that the mean value of the features tends to 0 and the standard deviation tends to 1. By performing scaling, we can easily compare these variables. The process of scaling is performed by using Standard Scaler from sklearn. At the end of this process, the data is in the form of an array. Hence, it is converted into a data frame and used for further analysis. Dropping the Names feature before we scale numeric values as the same will not add any value in model building.

Scaled Data frame (Z-score)

```
from scipy.stats import zscore
df_num_scaled=df_num.apply(zscore)
df_num_scaled.head(10)
```

	Apps	Accept	Enroll	Top10perc	Top25perc	F_Undergrad	P_Undergrad	Outstate	Room_Board	Books	Personal	PhD	Terminal	S_F.Ratio	perc_alumni	Expend	Grad_Rate
0	-0.346882	-0.321205	-0.063509	-0.258583	-0.191827	-0.168116	-0.209207	-0.746356	-0.964905	-0.602312	1.270045	-0.162859	-0.115729	1.013776	-0.867574	-0.501910	-0.317993
1	-0.210884	-0.038703	-0.288584	-0.655656	-1.353911	-0.209788	0.244307	0.457496	1.909208	1.215880	0.235515	-2.676529	-3.378176	-0.477704	-0.544572	0.166110	-0.551805
2	-0.406866	-0.376318	-0.478121	-0.315307	-0.292878	-0.549565	-0.497090	0.201305	-0.554317	-0.905344	-0.259582	-1.205112	-0.931341	-0.300749	0.585935	-0.177290	-0.668710
3	-0.668261	-0.681682	-0.692427	1.840231	1.677612	-0.658079	-0.520752	0.626633	0.996791	-0.602312	-0.688173	1.185939	1.175657	-1.615274	1.151188	1.792851	-0.376446
4	-0.726176	-0.764555	-0.780735	-0.655656	-0.596031	-0.711924	0.009005	-0.716508	-0.216723	1.518912	0.235515	0.204995	-0.523535	-0.553542	-1.675079	0.241803	-2.948375
5	-0.624307	-0.628611	-0.669812	0.592287	0.313426	-0.623421	-0.535212	0.760947	-0.932970	-0.299280	-0.983753	-0.346786	-0.455567	-1.185526	-0.948325	0.012806	-0.610258
6	-0.684808	-0.685356	-0.729043	-0.598931	-0.545505	-0.677472	-0.410988	0.708713	1.243144	-0.299280	0.235515	1.063321	0.903786	-0.654660	0.262933	-0.153145	-0.142634
7	-0.285088	-0.121984	-0.313353	0.535563	0.616579	-0.434450	-0.541127	0.852479	0.427443	-0.602312	-0.725120	1.002012	1.379560	-0.098515	1.151188	0.350074	0.441895
8	-0.507700	-0.481644	-0.595505	0.138490	0.363952	-0.562562	-0.361036	1.282036	0.038754	-1.511408	-1.242385	0.388922	0.292077	-0.705218	0.020681	0.380160	0.851066
9	-0.625600	-0.620854	-0.654735	-0.372032	-0.596031	-0.598459	-0.510893	0.006798	-0.891911	0.670422	0.678885	-2.002130	-2.630532	-0.654660	-0.625323	-0.128233	-0.785616

Descriptive Analytics of scaled Data:

```
df_num_scaled.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Apps	777.0	6.355797e-17	1.000644	-0.755134	-0.575441	-0.373254	0.160912	11.658671
Accept	777.0	6.774575e-17	1.000644	-0.794764	-0.577581	-0.371011	0.165417	9.924816
Enroll	777.0	-5.249269e-17	1.000644	-0.802273	-0.579351	-0.372584	0.131413	6.043678
Top10perc	777.0	-2.753232e-17	1.000644	-1.506526	-0.712380	-0.258583	0.422113	3.882319
Top25perc	777.0	-1.546739e-16	1.000644	-2.364419	-0.747607	-0.090777	0.667104	2.233391
F_Undergrad	777.0	-1.661405e-16	1.000644	-0.734617	-0.558643	-0.411138	0.062941	5.764674
P_Undergrad	777.0	-3.029180e-17	1.000644	-0.561502	-0.499719	-0.330144	0.073418	13.789921
Outstate	777.0	6.515595e-17	1.000644	-2.014878	-0.776203	-0.112095	0.617927	2.800531
Room_Board	777.0	3.570717e-16	1.000644	-2.351778	-0.693917	-0.143730	0.631824	3.436593
Books	777.0	-2.192583e-16	1.000644	-2.747779	-0.481099	-0.299280	0.306784	10.852297
Personal	777.0	4.765243e-17	1.000644	-1.611860	-0.725120	-0.207855	0.531095	8.068387
PhD	777.0	-1.035208e-16	1.000644	-3.964018	-0.653331	0.143686	0.756776	1.676411
Terminal	777.0	-4.481615e-16	1.000644	-3.785982	-0.591502	0.156142	0.835818	1.379560
S_F.Ratio	777.0	-2.057556e-17	1.000644	-2.929799	-0.654660	-0.123794	0.609307	6.499390
perc_alumni	777.0	-6.022638e-17	1.000644	-1.836580	-0.786824	-0.140820	0.666685	3.331452
Expend	777.0	1.213101e-16	1.000644	-1.240641	-0.557483	-0.245893	0.224174	8.924721
Grad_Rate	777.0	-2.091849e-16	1.000644	-3.240639	-0.727163	-0.025728	0.734160	2.020124

Z-score indicates how much a given value differs from the standard deviation. The Z-score, or standard score, is the number of standard deviations a given data point lies above or below mean. Standard deviation is essentially a reflection of the amount of variability within a given data set. Now we can easily compare all variables.

PCA

Statistical tests to be done before PCA

Bartletts Test of Sphericity

Bartlett's test of sphericity tests the hypothesis that the variables are uncorrelated in the population.

H0: All variables in the data are uncorrelated

Ha: At least one pair of variables in the data are correlated

If the null hypothesis cannot be rejected, then PCA is not advisable.

If the p-value is small, then we can reject the null hypothesis and agree that there is at least one pair of variables in the data which are correlated hence PCA is recommended.

```
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity
chi_square_value,p_value=calculate_bartlett_sphericity(df_num_scaled)
p_value
```

```
0.0
```

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.

```
from factor_analyzer.factor_analyzer import calculate_kmo
kmo_all,kmo_model=calculate_kmo(df_num_scaled)
kmo_model
```

```
0.8131319946422179
```

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

Covariance Matrix

The comparison between the covariance and correlation matrix is that both of these terms measure the relationship and the dependency between two variables respectively.

Scaling of data is just representation of the dataset in one unit so that it is easy to compare all the variables. The value and relationship of numbers will not change. We can see that even after scaling the correlation between the variables is same as the original data.

Covariance signifies the direction of the linear relationship between the two variables. By direction means if the variables are directly proportional or inversely proportional to each other. (Increasing the value of one variable might have a positive or a negative impact on the value of the other variable).

$$Cov(x,y) = \frac{\Sigma (x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

This below snip of the covariance matrix on scaled dataset. We can clearly understand covariance matrix indicates direction of the linear relationship between the variables.

```
cov_matrix = np.cov(df_num_scaled.T)
print(cov_matrix)
```

```
[[ 1.00128866e+00  9.44666359e-01  8.47913316e-01  3.39270321e-01
   3.52093041e-01  8.15540181e-01  3.98777500e-01  5.02236717e-02
   1.65151509e-01  1.32729421e-01  1.78961168e-01  3.91516047e-01
   3.69967622e-01  9.57562670e-02 -9.03421565e-02  2.59926503e-01
   1.47153003e-01]
 [ 9.44666359e-01  1.00128866e+00  9.12811453e-01  1.92694926e-01
   2.47794654e-01  8.75349854e-01  4.41839380e-01 -2.57877355e-02
   9.10157685e-02  1.13671647e-01  2.01247673e-01  3.56514197e-01
   3.38018401e-01  1.76456113e-01 -1.60196038e-01  1.24877730e-01
   6.68484304e-02]
 [ 8.47913316e-01  9.12811453e-01  1.00128866e+00  1.81527154e-01
   2.27037304e-01  9.65882744e-01  5.13729774e-01 -1.55677702e-01
  -4.02835287e-02  1.12856137e-01  2.81291483e-01  3.32172248e-01
   3.08671332e-01  2.37577072e-01 -1.81027112e-01  6.42519204e-02
  -2.20774244e-02]
 [ 3.39270321e-01  1.92694926e-01  1.81527154e-01  1.00128866e+00
   8.93144451e-01  1.41470801e-01 -1.05492050e-01  5.63055197e-01
   3.71959090e-01  1.19011599e-01 -9.34366503e-02  5.32810949e-01
   4.91767929e-01 -3.85370484e-01  4.56072227e-01  6.61765100e-01
   4.98759934e-01]
 [ 3.52093041e-01  2.47794654e-01  2.27037304e-01  8.93144451e-01
   1.00128866e+00  1.99701672e-01 -5.36456855e-02  4.90024494e-01
   3.31917067e-01  1.15676005e-01 -8.09144060e-02  5.46899712e-01
   5.25425061e-01 -2.95008517e-01  4.18402770e-01  5.28127127e-01
   4.80965361e-01]
 [ 8.15540181e-01  8.75349854e-01  9.65882744e-01  1.41470801e-01
   1.99701672e-01  1.00128866e+00  5.71247383e-01 -2.16020017e-01
  -6.89791682e-02  1.15698665e-01  3.17608307e-01  3.19002536e-01
   3.00405566e-01  2.80063790e-01 -2.29757920e-01  1.86756541e-02
  -7.83936405e-02]
 [ 3.98777500e-01  4.41839380e-01  5.13729774e-01 -1.05492050e-01
  -5.36456855e-02  5.71247383e-01  1.00128866e+00 -2.53839009e-01
  -6.14045345e-02  8.13041591e-02  3.20293836e-01  1.49481222e-01
   1.42086439e-01  2.32830164e-01 -2.81154208e-01 -8.36761155e-02
  -2.57466146e-01]
 [ 5.02236717e-02 -2.57877355e-02 -1.55677702e-01  5.63055197e-01
   4.90024494e-01 -2.16020017e-01 -2.53839009e-01  1.00128866e+00
   6.55099513e-01  3.89049389e-02 -2.99472320e-01  3.83965420e-01
   4.08508951e-01 -5.55536252e-01  5.66992142e-01  6.73645602e-01
   5.74350841e-01]
 [ 1.65151509e-01  9.10157685e-02 -4.02835287e-02  3.71959090e-01
   3.31917067e-01 -6.89791682e-02 -6.14045345e-02  6.55099513e-01
   1.00128866e+00  1.28127871e-01 -1.99685176e-01  3.30031652e-01
   3.75022201e-01 -3.63095039e-01  2.72714436e-01  5.02385989e-01
   4.26356391e-01]
```

```
[ 1.32729421e-01  1.13671647e-01  1.12856137e-01  1.19011599e-01
  1.15676005e-01  1.15698665e-01  8.13041591e-02  3.89049389e-02
  1.28127871e-01  1.00128866e+00  1.79525813e-01  2.68789637e-02
  1.00083507e-01 -3.19704202e-02 -4.02595501e-02  1.12553932e-01
  6.49958813e-04]
[ 1.78961168e-01  2.01247673e-01  2.81291483e-01 -9.34366503e-02
 -8.09144060e-02  3.17608307e-01  3.20293836e-01 -2.99472320e-01
 -1.99685176e-01  1.79525813e-01  1.00128866e+00 -1.07125431e-02
 -3.06525590e-02  1.36520535e-01 -2.86336597e-01 -9.80180427e-02
 -2.68934562e-01]
[ 3.91516047e-01  3.56514197e-01  3.32172248e-01  5.32810949e-01
  5.46899712e-01  3.19002536e-01  1.49481222e-01  3.83965420e-01
  3.30031652e-01  2.68789637e-02 -1.07125431e-02  1.00128866e+00
  8.50904189e-01 -1.30951361e-01  2.49562960e-01  4.33648094e-01
  3.11132316e-01]
[ 3.69967622e-01  3.38018401e-01  3.08671332e-01  4.91767929e-01
  5.25425061e-01  3.00405566e-01  1.42086439e-01  4.08508951e-01
  3.75022201e-01  1.00083507e-01 -3.06525590e-02  8.50904189e-01
  1.00128866e+00 -1.60310273e-01  2.67474531e-01  4.39364686e-01
  2.93911609e-01]
[ 9.57562670e-02  1.76456113e-01  2.37577072e-01 -3.85370484e-01
 -2.95008517e-01  2.80063790e-01  2.32830164e-01 -5.55536252e-01
 -3.63095039e-01 -3.19704202e-02  1.36520535e-01 -1.30951361e-01
 -1.60310273e-01  1.00128866e+00 -4.03448404e-01 -5.84584402e-01
 -3.08234490e-01]
[-9.03421565e-02 -1.60196038e-01 -1.81027112e-01  4.56072227e-01
  4.18402770e-01 -2.29757920e-01 -2.81154208e-01  5.66992142e-01
  2.72714436e-01 -4.02595501e-02 -2.86336597e-01  2.49562960e-01
  2.67474531e-01 -4.03448404e-01  1.00128866e+00  4.18250007e-01
  4.93521951e-01]
[ 2.59926503e-01  1.24877730e-01  6.42519204e-02  6.61765100e-01
  5.28127127e-01  1.86756541e-02 -8.36761155e-02  6.73645602e-01
  5.02385989e-01  1.12553932e-01 -9.80180427e-02  4.33648094e-01
  4.39364686e-01 -5.84584402e-01  4.18250007e-01  1.00128866e+00
  3.92700705e-01]
 [ 1.47153003e-01  6.68484304e-02 -2.20774244e-02  4.98759934e-01
  4.80965361e-01 -7.83936405e-02 -2.57466146e-01  5.74350841e-01
  4.26356391e-01  6.49958813e-04 -2.68934562e-01  3.11132316e-01
  2.93911609e-01 -3.08234490e-01  4.93521951e-01  3.92700705e-01
  1.00128866e+00]]
```

Correlation

Correlation measures the strength and the direction of the linear relationship between two variables. Strength is that is that positively correlated or negatively correlated. we can say the correlation values have standardized notions, whereas the covariance values are not standardized and cannot be used to compare how strong or weak the relationship is because the magnitude has no direct significance. It can assume values from -1 to +1.

$$Correlation = \frac{Cov\ (x, y)}{\sigma x * \sigma y}$$

where:

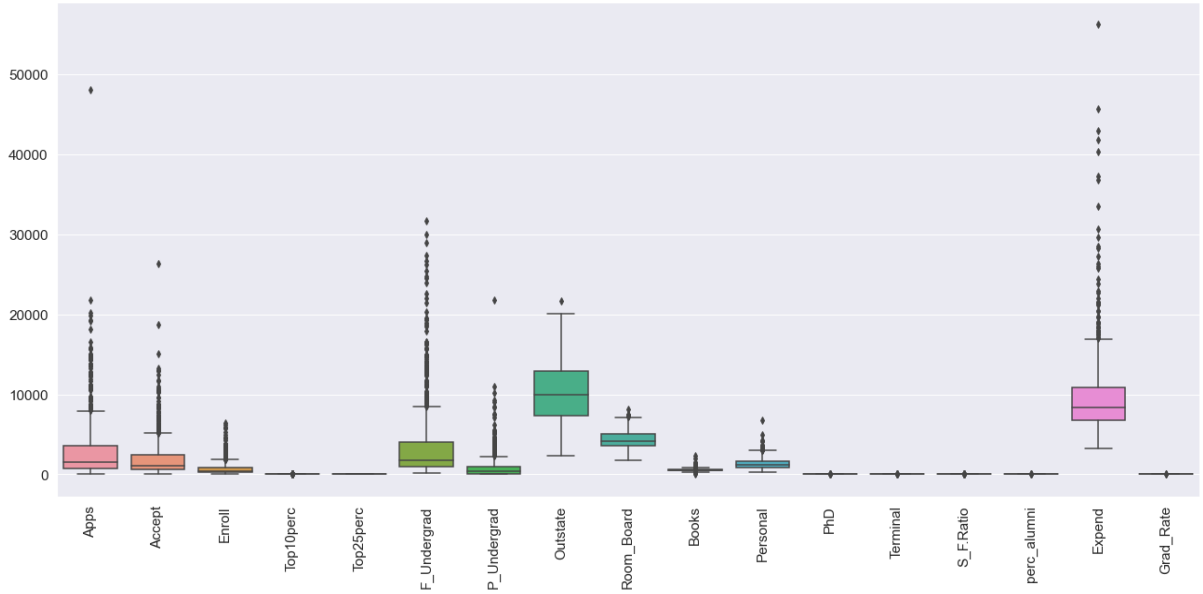
- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

This below snip is the correlation matrix. We can clearly understand the correlation which represents the strength and the relationship between the variables.

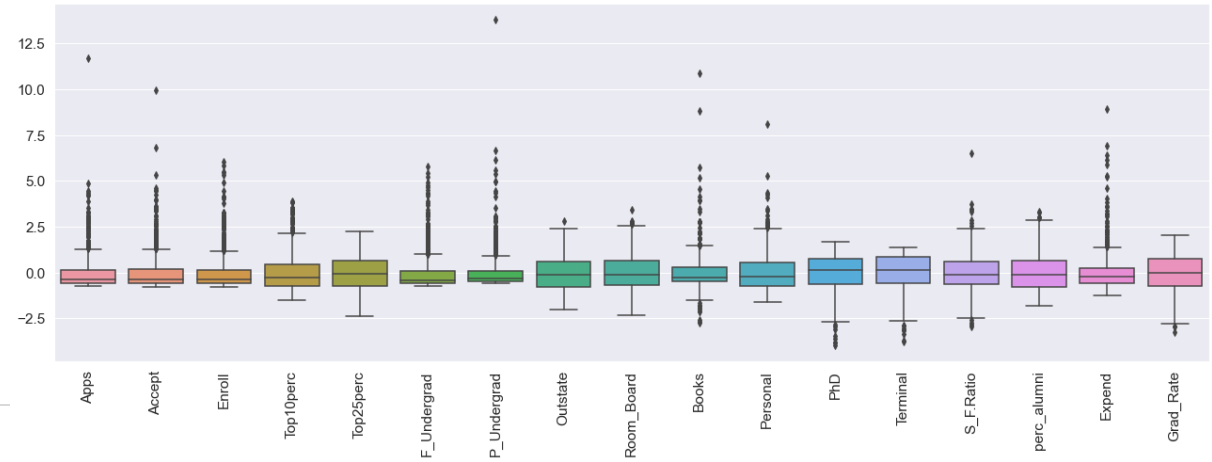
	Apps	Accept	Enroll	Top10perc	Top25perc	F_Undergrad	P_Undergrad	Outstate	Room_Board	Books	Personal	PhD	Terminal	S_F.Ratio	perc_alumni	Expend	Grad_Rate
Apps	1.000000	0.943451	0.846822	0.338834	0.351640	0.814491	0.398264	0.050159	0.164939	0.132559	0.178731	0.391012	0.369491	0.095633	-0.090226	0.259592	0.146964
Accept	0.943451	1.000000	0.911637	0.192447	0.247476	0.874223	0.441271	-0.025755	0.090899	0.113525	0.200989	0.356055	0.337583	0.176229	-0.159990	0.124717	0.066762
Enroll	0.846822	0.911637	1.000000	0.181294	0.226745	0.964640	0.513069	-0.155477	-0.040232	0.112711	0.280929	0.331745	0.308274	0.237271	-0.180794	0.064169	-0.022049
Top10perc	0.338834	0.192447	0.181294	1.000000	0.891995	0.141289	-0.105356	0.562331	0.371480	0.118858	-0.093316	0.532125	0.491135	-0.384875	0.455485	0.660913	0.498118
Top25perc	0.351640	0.247476	0.226745	0.891995	1.000000	0.199445	-0.053577	0.489394	0.331490	0.115527	-0.080810	0.546196	0.524749	-0.294629	0.417864	0.527447	0.480346
F_Undergrad	0.814491	0.874223	0.964640	0.141289	0.199445	1.000000	0.570512	-0.215742	-0.068890	0.115550	0.317200	0.318592	0.300019	0.279703	-0.229462	0.018652	-0.078293
P_Undergrad	0.398264	0.441271	0.513069	-0.105356	-0.053577	0.570512	1.000000	-0.253512	-0.061326	0.081200	0.319882	0.149289	0.141904	0.232531	-0.280792	-0.083568	-0.257135
Outstate	0.050159	-0.025755	-0.155477	0.562331	0.489394	-0.215742	-0.253512	1.000000	0.654256	0.038855	-0.299087	0.383471	0.407983	-0.554821	0.566262	0.672779	0.573612
Room_Board	0.164939	0.090899	-0.040232	0.371480	0.331490	-0.068890	-0.061326	0.654256	1.000000	0.127963	-0.199428	0.329607	0.374540	-0.362628	0.272363	0.501739	0.425808
Books	0.132559	0.113525	0.112711	0.118858	0.115527	0.115550	0.081200	0.038855	0.127963	1.000000	0.179295	0.026844	0.099955	-0.031929	-0.040208	0.112409	0.000649
Personal	0.178731	0.200989	0.280929	-0.093316	-0.080810	0.317200	0.319882	-0.299087	-0.199428	0.179295	1.000000	-0.010699	-0.030613	0.136345	-0.285968	-0.097892	-0.268588
PhD	0.391012	0.356055	0.331745	0.532125	0.546196	0.318592	0.149289	0.383471	0.329607	0.026844	-0.010699	1.000000	0.849809	-0.130783	0.249242	0.433090	0.310732
Terminal	0.369491	0.337583	0.308274	0.491135	0.524749	0.300019	0.141904	0.407983	0.374540	0.099955	-0.030613	0.849809	1.000000	-0.160104	0.267130	0.438799	0.293533
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	-0.130783	-0.160104	1.000000	-0.402929	-0.583832	-0.307838
perc_alumni	-0.090226	-0.159990	-0.180794	0.455485	0.417864	-0.229462	-0.280792	0.566262	0.272363	-0.040208	-0.285968	0.249242	0.267130	-0.402929	1.000000	0.417712	0.492887
Expend	0.259592	0.124717	0.064169	0.660913	0.527447	0.018652	-0.083568	0.672779	0.501739	0.112409	-0.097892	0.433090	0.438799	-0.583832	0.417712	1.000000	0.392195
Grad_Rate	0.146964	0.066762	-0.022049	0.498118	0.480346	-0.078293	-0.257135	0.573612	0.425808	0.000649	-0.268588	0.310732	0.293533	-0.307838	0.492887	0.392195	1.000000

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

Box plot before scaling dataset: (*Figure 21: Boxplot Numeric values before scaling dataset)*



Box plot after scaling dataset: (*Figure 22: Boxplot Numeric values after scaling dataset)*



Insights

From the above boxplots of BEFORE and AFTER scaling dataset we can see that the outliers are present in both the plots.

This is because, the scaling shrinks the range of the feature values as shown in the plots. However, the outliers have an influence when computing the empirical mean and standard deviation. This scaling of dataset does not have any effect on the outlier, The data gets converted to one unit. Hence, in both plots the only difference is the scaling of data which has changed from first plot to second plot. However, we can see in the above plot that after scaling, the median of all the features is quite close to each other.

2.5 Extract the eigenvalues and eigenvectors. [Using Sklearn PCA Print Both]

Eigen vectors:

```
#Extract eigen vectors
eigen_vectors = pca.components_
eigen_vectors
```

```
array([[ 2.48183494e-01,  2.06969666e-01,  1.75694894e-01,
         3.54244165e-01,  3.43942665e-01,  1.54038269e-01,
         2.60218245e-02,  2.94935587e-01,  2.49048365e-01,
         6.45834383e-02, -4.27212841e-02,  3.18442536e-01,
         3.16946970e-01, -1.77143862e-01,  2.05340243e-01,
         3.18874326e-01,  2.53800631e-01],
       [ 3.32025554e-01,  3.72491384e-01,  4.04001896e-01,
        -8.18674207e-02, -4.42586593e-02,  4.17901438e-01,
         3.15112160e-01, -2.49149211e-01, -1.37349991e-01,
         5.64823263e-02,  2.19795159e-01,  5.86942062e-02,
         4.68443038e-02,  2.46336958e-01, -2.46268610e-01,
        -1.31140311e-01, -1.69072352e-01],
       [-6.16148087e-02, -9.97638535e-02, -8.22289219e-02,
         3.47636016e-02, -2.46996496e-02, -6.09712128e-02,
         1.39028188e-01,  4.73522670e-02,  1.50278691e-01,
         6.78096050e-01,  4.98147475e-01, -1.29562447e-01,
        -6.83497434e-02, -2.90605689e-01, -1.46940854e-01,
         2.27294587e-01, -2.06564710e-01],
       [ 2.82568646e-01,  2.69149266e-01,  1.62610883e-01,
        -5.24694979e-02, -1.11389363e-01,  1.00948100e-01,
        -1.58472919e-01,  1.33121744e-01,  1.86480406e-01,
         7.94881916e-02, -2.36125536e-01, -5.33695158e-01,
        -5.19384634e-01, -1.63696255e-01,  1.76377241e-02,
         8.13756862e-02,  2.60607085e-01],
       [ 4.15795223e-03,  5.43184748e-02, -5.66031578e-02,
        -3.94960241e-01, -4.25699527e-01, -4.40351682e-02,
         3.03348454e-01,  2.21796925e-01,  5.59743828e-01,
        -1.28342769e-01, -2.21266623e-01,  1.43265072e-01,
         2.07795949e-01, -7.85572635e-02, -2.16205609e-01,
         7.56771855e-02, -1.11221583e-01],
       [-1.42383611e-02,  9.38001713e-03, -4.13474070e-02,
        -5.30637057e-02,  3.22879196e-02, -4.26578545e-02,
        -1.92743643e-01, -2.88600187e-02,  1.64167802e-01,
         6.41363539e-01, -3.33975699e-01,  8.76858530e-02,
         1.51537319e-01,  4.86123679e-01, -4.72277091e-02,
        -2.97658349e-01,  2.16104449e-01],
       [-3.91002763e-02, -9.99986298e-03, -2.63536670e-02,
        -1.62338428e-01, -1.20418652e-01, -2.45828649e-02,
         5.12488320e-02,  1.10254123e-01,  2.13121602e-01,
        -1.49628734e-01,  6.34529284e-01, -2.70577842e-03,
        -3.22255224e-02,  2.19169800e-01,  2.38710855e-01,
        -2.26666001e-01,  5.59961573e-01],
       [-1.03310494e-01, -5.61813492e-02,  5.85153839e-02,
        -1.24270163e-01, -1.03696355e-01,  7.86975228e-02,
         5.70821843e-01,  1.10621336e-02, -2.19257590e-01,
         2.11922720e-01, -2.26601574e-01, -7.71224789e-02,
        -1.25555420e-02, -8.13012041e-02,  6.81300021e-01,
        -5.63788867e-02, -4.33562643e-04],
```

```
[-9.00143496e-02, -1.77556828e-01, -1.28424051e-01,
  3.40477400e-01,  4.03222858e-01, -5.93938189e-02,
  5.60390667e-01, -3.51751983e-03,  2.76787474e-01,
 -1.34222140e-01, -9.17789776e-02, -1.85673273e-01,
 -2.55999525e-01,  2.76263080e-01, -2.52990818e-01,
 -5.03211730e-02,  4.32269312e-02],
[ 5.18404688e-02,  4.06506561e-02,  3.35003679e-02,
  6.43064702e-02,  1.39716747e-02,  1.97927106e-02,
 -2.23571664e-01,  1.84695967e-01,  2.95395690e-01,
 -8.26543140e-02,  1.36410997e-01, -1.21426900e-01,
 -8.43038146e-02,  4.73004827e-01,  4.22490920e-01,
  1.37174802e-01, -5.91709084e-01],
[ 4.31125638e-02, -5.84517965e-02, -6.92402226e-02,
 -8.56143312e-03, -2.73497922e-01, -8.09723157e-02,
  1.01830969e-01,  1.44196428e-01, -3.60563153e-01,
  3.26108661e-02, -1.92134201e-02,  4.06024281e-02,
 -6.03502966e-02,  4.43251575e-01, -1.32746868e-01,
  6.90442463e-01,  2.23167631e-01],
[ 2.42372672e-02, -1.45492263e-01,  1.11811311e-02,
  3.83798448e-02, -9.02505112e-02,  5.62647451e-02,
 -6.29635597e-02, -8.23344183e-01,  3.53706323e-01,
 -2.78181147e-02, -3.95028152e-02,  2.35331191e-02,
  1.56267370e-02, -1.09829849e-02,  1.81690531e-01,
  3.26528934e-01,  1.24426246e-01],
[ 5.96030775e-01,  2.92865648e-01, -4.45157955e-01,
  4.57887548e-04,  2.26581000e-02, -5.24069656e-01,
  1.26135734e-01, -1.42325248e-01, -7.02905817e-02,
  1.10452007e-02,  3.97075268e-02,  1.24170339e-01,
 -5.42444109e-02, -1.74893537e-02,  1.03821809e-01,
 -9.32330111e-02, -6.85908141e-02],
[ 7.75518256e-02,  3.15355868e-02, -8.32602175e-02,
 -1.08207844e-01,  1.51278477e-01, -5.34433270e-02,
  1.92900162e-02, -3.37053759e-02, -5.82036547e-02,
 -6.69447718e-02,  2.77232366e-02, -6.91987039e-01,
  6.70859121e-01,  4.13684713e-02, -2.82083093e-02,
  7.40538471e-02,  3.96685843e-02],
[ 1.33371016e-01, -1.45388897e-01,  2.95945213e-02,
  6.97751336e-01, -6.17186656e-01,  9.91243220e-03,
  2.09316376e-02,  3.85609596e-02,  3.44615841e-03,
 -9.48644074e-03, -3.04152168e-03, -1.12189433e-01,
  1.58993880e-01, -2.08929025e-02, -8.35207685e-03,
 -2.27820854e-01, -3.67678627e-03],
[ 4.59076616e-01, -5.18177361e-01, -4.04677937e-01,
 -1.48517751e-01,  5.19152165e-02,  5.60492277e-01,
 -5.28627570e-02,  1.01803555e-01, -2.58212787e-02,
  2.87114472e-03, -1.29478364e-02,  2.99363682e-02,
 -2.71848801e-02, -2.12392559e-02,  3.54422732e-03,
 -4.40175332e-02, -5.90489318e-03],
[ 3.59246931e-01, -5.43645791e-01,  6.09430563e-01,
 -1.44951595e-01,  8.04207346e-02, -4.14450876e-01,
  8.91036957e-03,  5.11101618e-02,  1.19217194e-03,
  7.61205432e-04, -1.14046486e-03,  1.39675866e-02,
  6.12953614e-03, -2.23053194e-03, -1.90639020e-02,
 -3.54348850e-02, -1.36253224e-02]])
```

Eigen values:

```
#Check the eigen values
#Note: This is always returned in descending order
pca.explained_variance_
```

```
array([5.45485033, 4.48406663, 1.17470127, 1.00536006, 0.9343371 ,
       0.84817556, 0.60551358, 0.58787041, 0.53053165, 0.40349818,
       0.31326156, 0.22048561, 0.16780564, 0.14368317, 0.08802439,
       0.03672101, 0.02302105])
```

2.6 Perform PCA and export the data of the Principal Component (eigenvectors) into a data frame with the original features.

Principal Component into Data frame with original Features:

	Apps	Accept	Enroll	Top10perc	Top25perc	F_Undergrad	P_Undergrad	Outstate	Room_Board	Books	Personal	PhD	Terminal	S_F.Ratio	perc_alumni	Expend	Grad_Rate
PC1	0.248183	0.206970	0.175695	0.354244	0.343943	0.154038	0.026022	0.294936	0.249048	0.064583	-0.042721	0.318443	0.316947	-0.177144	0.205340	0.318874	0.253801
PC2	0.332026	0.372491	0.404002	-0.081867	-0.044259	0.417901	0.315112	-0.249149	-0.137350	0.056482	0.219795	0.058694	0.046844	0.246337	-0.246269	-0.131140	-0.169072
PC3	-0.061615	-0.099764	-0.082229	0.034764	-0.024700	-0.060971	0.139028	0.047352	0.150279	0.678096	0.498147	-0.129562	-0.068350	-0.290606	-0.146941	0.227295	-0.206565
PC4	0.282569	0.269149	0.162611	-0.052469	-0.111389	0.100948	-0.158473	0.133122	0.186480	0.079488	-0.236126	-0.533695	-0.519385	-0.163696	0.017638	0.081376	0.260607
PC5	0.004158	0.054318	-0.056603	-0.394960	-0.425700	-0.044035	0.303348	0.221797	0.559744	-0.128343	-0.221267	0.143265	0.207796	-0.078557	-0.216206	0.075677	-0.111222
PC6	-0.014238	0.009380	-0.041347	-0.053064	0.032288	-0.042658	-0.192744	-0.028860	0.164168	0.641364	-0.333976	0.087686	0.151537	0.486124	-0.047228	-0.297658	0.216104
PC7	-0.039100	-0.010000	-0.026354	-0.162338	-0.120419	-0.024583	0.051249	0.110254	0.213122	-0.149629	0.634529	-0.002706	-0.032226	0.219170	0.238711	-0.226666	0.559962
PC8	-0.103310	-0.056181	0.058515	-0.124270	-0.103696	0.078698	0.570822	0.011062	-0.219258	0.211923	-0.226602	-0.077122	-0.012556	-0.081301	0.681300	-0.056379	-0.000434
PC9	-0.090014	-0.177557	-0.128424	0.340477	0.403223	-0.059394	0.560391	-0.003518	0.276787	-0.134222	-0.091779	-0.185673	-0.256000	0.276263	-0.252991	-0.050321	0.043227
PC10	0.051840	0.040651	0.033500	0.064306	0.013972	0.019793	-0.223572	0.184696	0.295396	-0.082654	0.136411	-0.121427	-0.084304	0.473005	0.422491	0.137175	-0.591709
PC11	0.043113	-0.058452	-0.069240	-0.008561	-0.273498	-0.080972	0.101831	0.144196	-0.360563	0.032611	-0.019213	0.040602	-0.060350	0.443252	-0.132747	0.690442	0.223168
PC12	0.024237	-0.145492	0.011181	0.038380	-0.090251	0.056265	-0.062964	-0.823344	0.353706	-0.027818	-0.039503	0.023533	0.015627	-0.010983	0.181691	0.326529	0.124426
PC13	0.596031	0.292866	-0.445158	0.000458	0.022658	-0.524070	0.126136	-0.142325	-0.070291	0.011045	0.039708	0.124170	-0.054244	-0.017489	0.103822	-0.093233	-0.068591
PC14	0.077552	0.031536	-0.083260	-0.108208	0.151278	-0.053443	0.019290	-0.033705	-0.058204	-0.066945	0.027723	-0.691987	0.670859	0.041368	-0.028208	0.074054	0.039669
PC15	0.133371	-0.145389	0.029595	0.697751	-0.617187	0.009912	0.020932	0.038561	0.003446	-0.009486	-0.003042	-0.112189	0.158994	-0.020893	-0.008352	-0.227821	-0.003677
PC16	0.459077	-0.518177	-0.404678	-0.148518	0.051915	0.560492	-0.052863	0.101804	-0.025821	0.002871	-0.012948	0.029936	-0.027185	-0.021239	0.003544	-0.044018	-0.005905
PC17	0.359247	-0.543646	0.609431	-0.144952	0.080421	-0.414451	0.008910	0.051110	0.001192	0.000761	-0.001140	0.013968	0.006130	-0.002231	-0.019064	-0.035435	-0.013625

2.7 Write down the explicit form of the first PC (in terms of the eigenvectors. Use values with two places of decimals only). [hint: write the linear equation of PC in terms of eigenvectors and corresponding features]

```
First_PC = eigen_vectors[0]
First_PC
```

```
array([ 0.24818349,  0.20696967,  0.17569489,  0.35424416,  0.34394266,
        0.15403827,  0.02602182,  0.29493559,  0.24904836,  0.06458344,
       -0.04272128,  0.31844254,  0.31694697, -0.17714386,  0.20534024,
        0.31887433,  0.25380063])
```

```
np.round( First_PC, decimals = 2)
```

```
array([ 0.25,  0.21,  0.18,  0.35,  0.34,  0.15,  0.03,  0.29,  0.25,
        0.06, -0.04,  0.32,  0.32, -0.18,  0.21,  0.32,  0.25])
```

The Linear equation of 1st component:

(0.25 x Apps) + (0.21 x Accept) + (0.18 x Enroll) + (0.35 x Top10perc) + (0.34 x Top25perc) + (0.15 x F. Undergrad) +(0.03 x P. Undergrad) + (0.29 x Outstate) + (0.25 x Room. Board) + (0.06 x Books) + (-0.04 x Personal) + (0.32 x PhD) + (0.32 x Terminal) + (-0.18 x S.F. Ratio) + (0.21 x perc. Alumni) + (0.32 x Expend) + (0.25 x Grad.Rate)

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?

```
np.cumsum(pca.explained_variance_ratio_)
```

```
array([0.32046058, 0.58388974, 0.65290088, 0.7119636 , 0.76685387,
       0.81668234, 0.85225494, 0.88679105, 0.91795863, 0.94166327,
       0.9600667 , 0.97301975, 0.98287797, 0.99131904, 0.99649028,
       0.99864756, 1.        ])
```

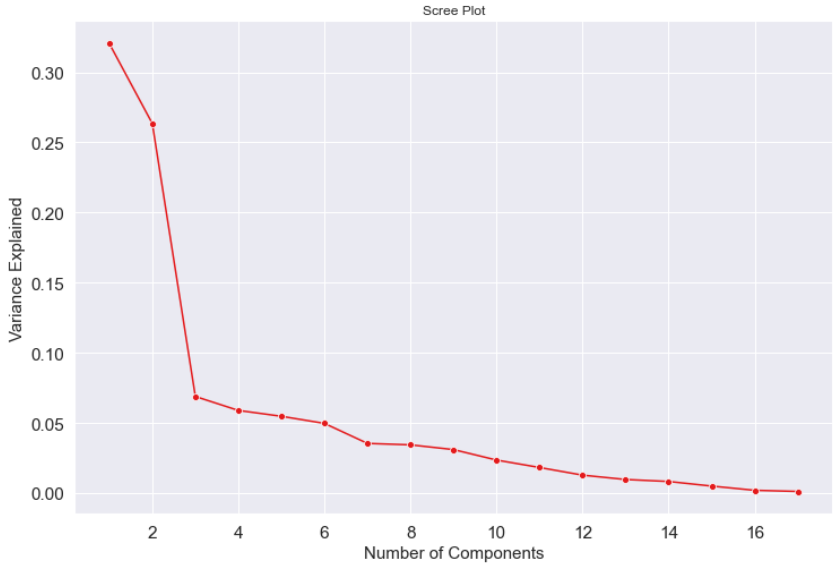
How does it help you to decide on the optimum number of principal components?

- Cumulative variance is amount of variance of the original data explained by each type of model plotted against the number of components.
- Always the first principal component captures the maximum variance of the dataset.
- The cumulative percentage for the second component is the sum of the percentage of variance for the first and second components, the cumulative of third component is sum of percentage of variance of first, second and third and so on.
- From the cumulative variance experienced, we can understand how much variance is captured by particular number of principal components. For example, if we wanted to work with approx. 90% variance captured, the number of components will be 8 in this particular case. The eigenvectors determine the directions of the new feature space. They indicate the principal components. The number of principal components depends on the Business requirement of percentage of variance to capture.

What do the eigenvectors indicate?

- The Eigenvectors indicate the direction of the principal components.
- Eigenvectors are the set of basic functions that are the most efficient set to describe data variability. They are also the coordinate system that the covariance matrix becomes diagonal allowing the new variables referenced to this coordinate system to be uncorrelated.
- An eigenvector is a vector whose direction remains unchanged when a linear transformation is applied to it.
- The eigenvectors (principal components) determine the directions of the new feature space, and the eigenvalues determine their magnitude.
- Furthermore, eigen decomposition forms the base of the geometric interpretation of covariance matrices.

Plotting Scree-plot to determine the number of factors to retain in an exploratory factor analysis



(Figure 23: Scree Plot)

In this case, we will check how many PC explain for approximately 90% of the variance in data. That will be the optimum number of principal components.

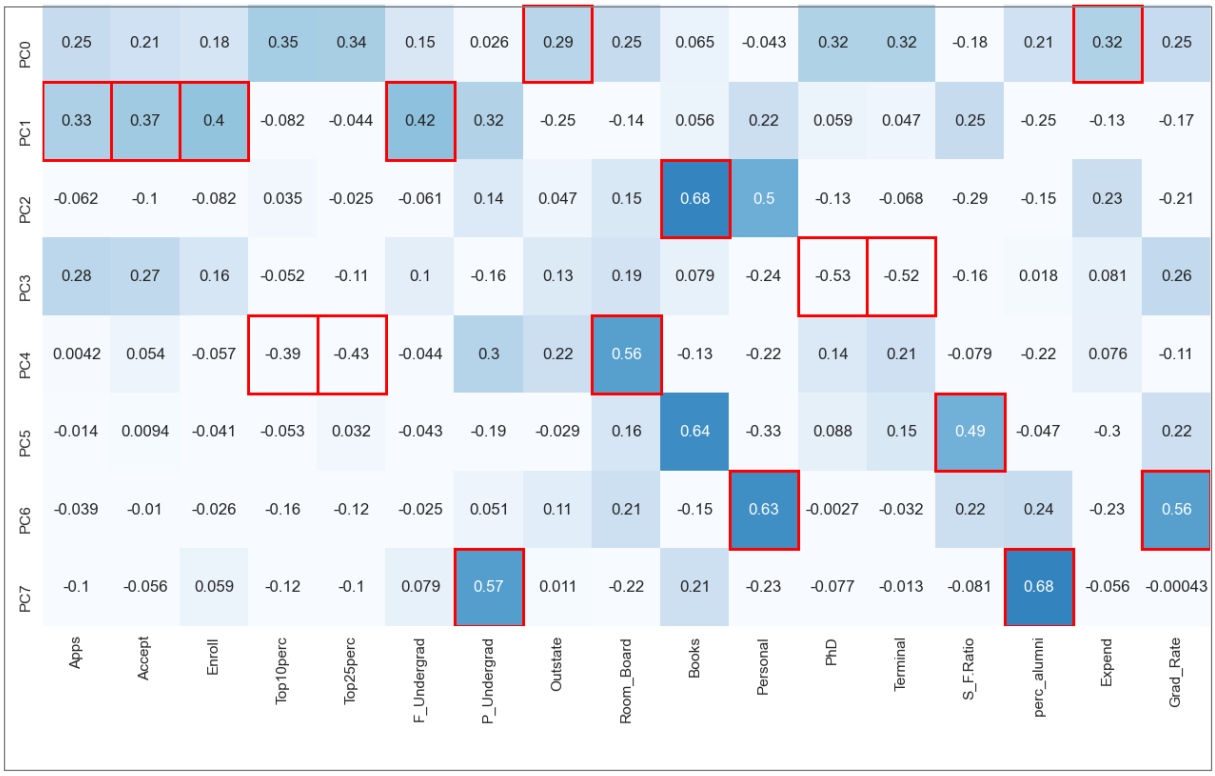
So, for cumulative sum we see that first 8 principal components explain 90% of variance in data. Therefore 8 PCs will be optimum for this dataset.

If we check the scree plot for the same, we will see that first 8 PCs give 90% of variance and therefore keeping 8 PCs for this dataset will be optimum.

Let's identify which features have maximum loading across the components.

- We will first plot the component loading on a heatmap.

- For each feature, we find the maximum loading value across the components and mark the same with help of rectangular box.
- Features marked with rectangular red box are the one having maximum loading on the respective component. We consider these marked features to decide the context that the component represents



(Figure 24: plot the component loading on a heatmap)

- PCO - Outstate & Expand
- PC1 - Apps, Accept, Enroll, F. Undergrad,
- PC2 - Books
- PC3 - PhD, Terminal
- PC4 - Top10Perc, Top25Perc, Room Board
- PC5 - S_F. Ratio
- PC6 - Personal, Grad Ratio
- PC7 – P. Undergrad, Perc Alumni

Giving names accordingly, we can create the final data frame with new reduced components as follows:

	pc_Outstate_Expand	pc_Applications	pc_Bookd	pc_PhD	pc_Top_percent	pc_S_F.Ratio	pc_Personal	pc_Alumni
0	-1.593266	0.764607	-0.107060	-0.928681	-0.738979	-0.305103	0.639973	-0.873641
1	-2.193362	-0.581284	2.296629	3.587686	1.037785	-0.152909	0.263006	0.050182
2	-1.430551	-1.094598	-0.435461	0.688978	-0.372722	-0.954627	-0.245467	0.307045
3	2.857891	-2.626056	0.136121	-1.279351	-0.174330	-1.066654	-1.262575	-0.159711
4	-2.217343	0.020351	2.382812	-1.106215	0.691579	-0.000224	-2.160878	-0.645014
...	...	...	...	...	...	...	...	...
772	-3.331877	1.216020	-0.382920	0.116750	0.776486	0.311989	-0.165859	0.346227
773	0.202842	-0.687321	0.053747	0.555232	0.370817	0.376331	0.849616	0.634481
774	-0.734117	-0.077759	0.000171	0.058964	-0.515758	0.470364	-1.315968	-0.140485
775	7.923697	-2.057366	2.078119	0.850451	-0.952035	-2.066705	0.088002	-0.551528
776	-0.464731	0.363929	-1.331397	-0.128448	-1.133059	0.836910	1.299212	0.638711

777 rows × 8 columns

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]

We finally combine the categorical variables and the new reduced components after performing PCA into a final data frame:

```
df_final = pd.concat([df_cat, df_pca], axis=1)
df_final
```

	Names	pc_Outstate_Expand	pc_Applications	pc_Bookd	pc_PhD	pc_Top_percent	pc_S_F.Ratio	pc_Personal	pc_Alumni
0	Abilene Christian University	-1.593266	0.764607	-0.107060	-0.928681	-0.738979	-0.305103	0.639973	-0.873641
1	Adelphi University	-2.193362	-0.581284	2.296629	3.587686	1.037785	-0.152909	0.263006	0.050182
2	Adrian College	-1.430551	-1.094598	-0.435461	0.688978	-0.372722	-0.954627	-0.245467	0.307045
3	Agnes Scott College	2.857891	-2.626056	0.136121	-1.279351	-0.174330	-1.066654	-1.262575	-0.159711
4	Alaska Pacific University	-2.217343	0.020351	2.382812	-1.106215	0.691579	-0.000224	-2.160878	-0.645014
...	...	...	...	...	...	...	...	...	...
772	Worcester State College	-3.331877	1.216020	-0.382920	0.116750	0.776486	0.311989	-0.165859	0.346227
773	Xavier University	0.202842	-0.687321	0.053747	0.555232	0.370817	0.376331	0.849616	0.634481
774	Xavier University of Louisiana	-0.734117	-0.077759	0.000171	0.058964	-0.515758	0.470364	-1.315968	-0.140485
775	Yale University	7.923697	-2.057366	2.078119	0.850451	-0.952035	-2.066705	0.088002	-0.551528
776	York College of Pennsylvania	-0.464731	0.363929	-1.331397	-0.128448	-1.133059	0.836910	1.299212	0.638711

777 rows × 9 columns

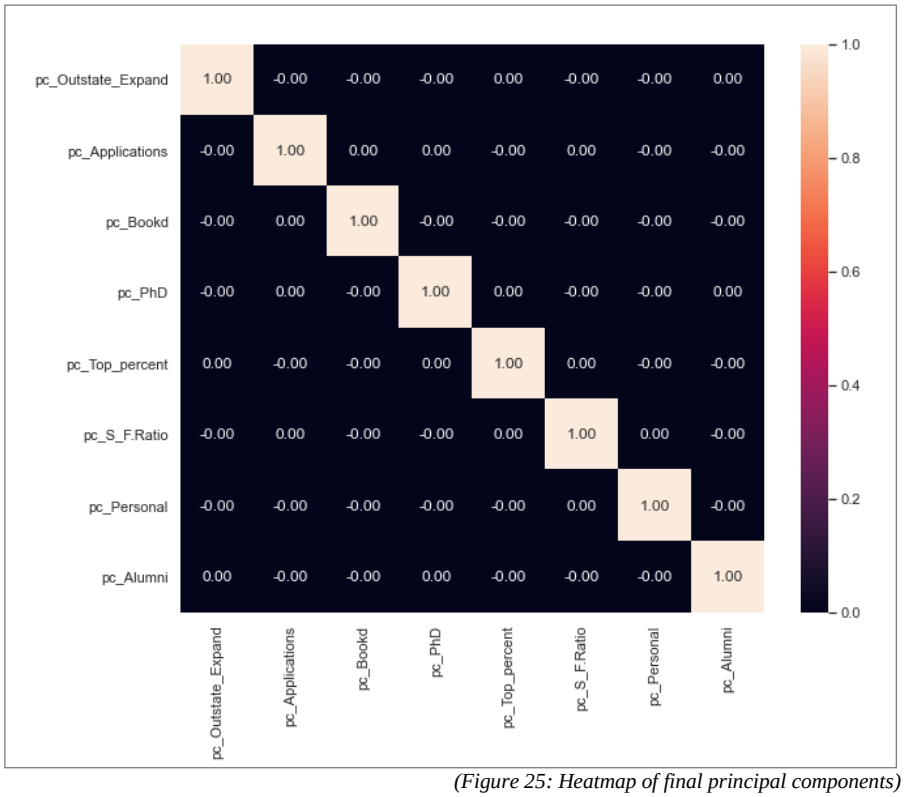
PCA helps in reducing multicollinearity between the variables.

PCA components are reduced based on the variance in the dataset.

In this case, we took 8 principal components which gives us the maximum variance in the dataset.

These 8 components give us a new data frame which gives us reduced multicollinearity.

Checking the correlation between the variables in the new data frame, we get the following heatmap on python:



(Figure 25: Heatmap of final principal components)

Thus, from the above heatmap, we can infer that all components have almost no collinearity between them. Hence, we have achieved, reducing the number of components of the dataset and reducing the multicollinearity between the variables of dataset.

Business implication of using Principal Component Analysis for this case study:

In this particular case study, we deal with 17 numerical columns. This is a high dimensional data. When it comes to high dimensional data, it is usually difficult to recognise and interpret patterns. This renders it difficult to work with data and gain any kind of insights.

The way PCA works is that, based on the original data, it calculates a set of variables that describe as much variance as possible in the data. Principal Component Analysis helps in transforming a large data set with a bunch of 17 correlated variables into a specific set of smaller variables called principal components.

We used this technique for extracting only the most important factors based on the eigenvalues and eigenvectors calculated using the same dataset. The eigenvectors provide the direction of the new dimension in which the spread will be observed. They give the direction of the new axis while eigenvalues explained the spread/variance in the data set. The right combination of eigenvectors and eigenvalues gave us the privilege of dropping the least important variables.

For example, in this case study, the first principal component captures 32.02% of the variance in the data, followed by the second principal component, which captures 26.34% of the variance in the data. Depending on how much variance in the data we want to work with, we can choose the number of principal components.

This made it easy for us in determining which factors to target the most and which are to be dropped. Reducing the dimensions of the data, making it more comprehensible and effective in targeting and segmentation of our business was achieved by PCA. In the case study at hand, we reduce the dimensions from 17 to 8 considering the required variance has approximately 90% of dataset. Understanding how much variance is captured by each principal component helps with dimensionality reduction of the data, which in turn makes it easier to recognize underlying patterns to gain better insights.

The outputs of the PCA can be used to highlight both, the similarities and the differences in the dataset. It also resolves the problem of multicollinearity which undermines the significance of an independent variable and can also affect the interpretation of a model where the data is used. Hence, PCs have varied use in further analysis of data.
