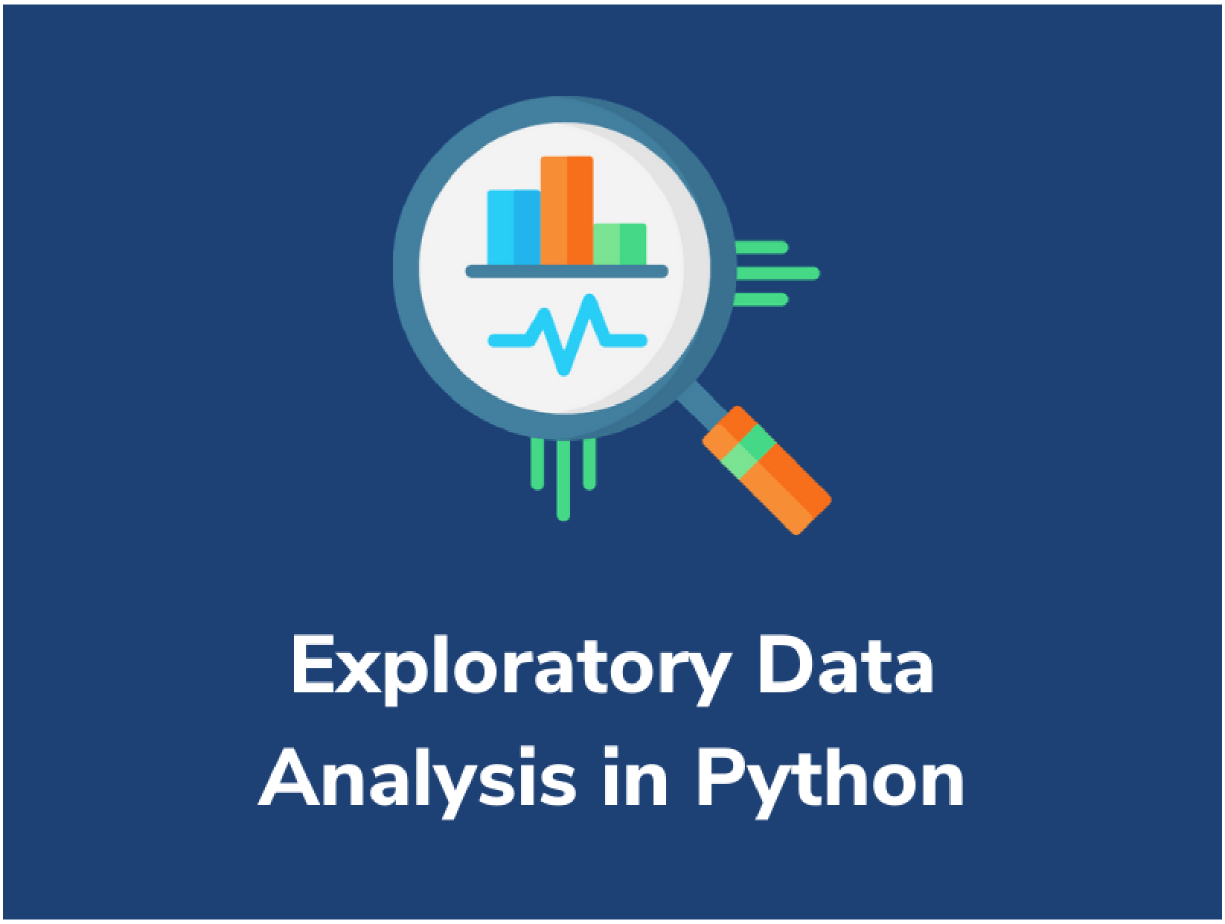


Exploratory Data Analysis in Python

Candidate No: 220298193

BNM864 – Software Analytics

EDA PROJECT

INTRODUCTION

Python is a strong tool when it comes to developing and maintaining data since it has a wide variety of statistical libraries which can be used to modify, visualise and analyse complex data structures. The analysis is focused on the College Dataset, which has ten different variables. The motivation to do the exploratory data analysis on this data set was that it has a good number of continuous variables. In case of a continuous variable, variety of analysis options are available which can be used to study the cause of variation. Also, the use of seaborn library helped in creating visually attractive and informative graphs.

In [4]:

```
# Importing the pandas and numpy libraries for analysing the "College" dataset.

import pandas as pd
import numpy as np
```

In [5]:

```
# Loading the data in a data frame using panda library function "read_excel"
df_clg=pd.read_excel("College.xlsx")

# Inspecting the data by reading the first three rows of the data
df_clg.head(3)
```

Out[5]:

	Institution	Private	Apps	Accept	Enroll	Students	SFRatio	Expend	GradRate	PhD
0	Johns Hopkins University	Yes	8474.0	3446.0	911.0	5135.0	3.3	56233.0	90.0	>50
1	Washington University	Yes	7654.0	5259.0	1254.0	6153.0	3.9	45702.0	90.0	>50
2	Antioch University	Yes	713.0	661.0	252.0	735.0	11.3	42926.0	48.0	>50

In [6]:

```
# Gaining the information on the dataset before analysing it.
df_clg.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 777 entries, 0 to 776
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Institution  777 non-null    object 
 1   Private      765 non-null    object 
 2   Apps         775 non-null    float64
 3   Accept       774 non-null    float64
 4   Enroll       772 non-null    float64
 5   Students     776 non-null    float64
 6   SFRatio      774 non-null    float64
 7   Expend       775 non-null    float64
 8   GradRate     774 non-null    float64
 9   PhD          772 non-null    object 
dtypes: float64(7), object(3)
memory usage: 60.8+ KB
```

The data set consists of 777 rows and 10 columns , where each column refers to one variable.

Also, it is observed that there are 7 numerical and 3 categorical variables in the data set. It can also be observed through the non-null count that there are missing entries in some cells.

Checking the dataset for duplicate records

It is important to check the data set for the duplicate records as the duplication in the data can disrupt the analysis.

In [7]:

```
# Checking the size of the data frame
df_clg.shape
```

Out[7]:

```
(777, 10)
```

In [8]:

```
# Removing the duplicates of the data set
df_clg.drop_duplicates(inplace=True)

# Checking the size of the data frame again to check the number of duplicates
df_clg.shape
```

Out[8]:

```
(777, 10)
```

As the size of the data frame remains same, it is implied that there are no duplicate values in the data set.

Q1 : GENERATE DESCRIPTIVE STATISTICS FOR THE DATA SET

The characteristics and key features of a data set can easily be described using descriptive statistics.

In [9]:

```

# Generating descriptive statistics for the college data set
df_clg.describe()
```

Out[9]:

	Apps	Accept	Enroll	Students	SFRatio	Expend	GradRate
count	775.000000	774.000000	772.000000	776.000000	774.000000	775.000000	774.000000
mean	3004.927742	2014.164083	781.781088	4550.208763	14.090698	9642.797419	65.449612
std	3874.120093	2447.981568	931.034168	5858.384381	3.965024	5210.996785	17.194855
min	81.000000	72.000000	35.000000	3.000000	2.500000	3186.000000	10.000000
25%	778.000000	601.750000	242.750000	1225.500000	11.500000	6747.500000	53.000000
50%	1558.000000	1109.500000	435.500000	2095.000000	13.600000	8367.000000	65.000000
75%	3635.000000	2418.500000	902.250000	5121.000000	16.500000	10816.000000	78.000000
max	48094.000000	26330.000000	6392.000000	38338.000000	39.800000	56233.000000	118.000000

The descriptive statistics for 7 numerical variables have been generated. However, the descriptive statistics for other variables have not been generated as they are categorical variables.The key findings are:

a) The count suggests that there are missing values in the data set.

b)The standard deviation for number of students and expenditure rate is quite high which explains that there is a major spread in the data for students and instructional expenditure per student.

c) Around 75% of the colleges and universities have less than 5121 students.

Q2.Check any records with missing values and, handle the missing data as appropriate

In [10]:

```
# Inspecting the complete data set for the missing values .
df_clg.isnull().sum()
```

Out[10]:

```
Institution     0
Private        12
Apps            2
Accept          3
Enroll          5
Students        1
SFRatio         3
Expend          2
GradRate        3
PhD             5
dtype: int64
```

As per the output, it is clear that there are missing values in all the columns except Institution.

In [11]:

```
# Dropping the rows having missing values
df_clg.dropna(inplace=True, axis="rows")

# Checking the data frame again for missing values
df_clg.isnull().sum()
```

Out[11]:

```
Institution    0
Private        0
Apps           0
Accept         0
Enroll         0
Students       0
SFRatio        0
Expend         0
GradRate       0
PhD            0
dtype: int64
```

Now, it can be observed that all the rows having missing values have been deleted.

Checking the size of data frame again

In [12]:

```
df_clg.shape
```

Out[12]:

```
(745, 10)
```

Q3. Building Graphs

A. The distribution of one or more continuous variables

Histogram is used to study the distribution of a continuous variable.

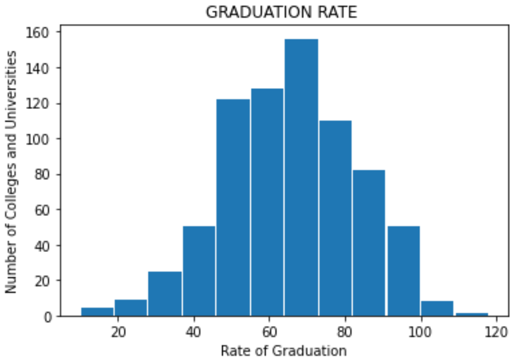
In [13]:

```
# Creating a histogram to study the distribution of the continuous variable"Graduation Rate"
# Importing matplotlib for creating graphs

import matplotlib.pyplot as plot
df_clg["GradRate"].plot(kind="hist", bins=12 , title="GRADUATION RATE ",rot=0,rwidth=0.95)
plot.xlabel("Rate of Graduation")
plot.ylabel("Number of Colleges and Universities")
```

Out[13]:

```
Text(0, 0.5, 'Number of Colleges and Universities')
```



The number of bins were decided on the number of class intervals. Each class range has 2 bins.The histogram depicts that the distribution for the Graduation Rate is negatively skewed that is there are more values on the right side of the distribution.

B. Relationship between 2 continuous variables

Scatter plot is used to study the relationship between 2 continuous variables.

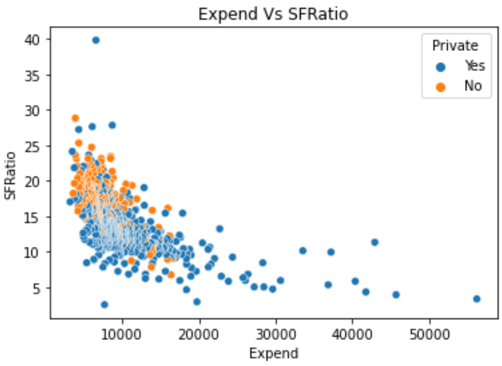
In [14]:

```
# Creating a scatter plot to study the relationship between Student Faculty Ratio and Expend
# Importing seaborn to create visually attractive and informative graphs

import seaborn as sb
sb.scatterplot(data=df_clg,x="Expend", y="SFRatio",hue="Private").set(title="Expend Vs SFRatio")
```

Out[14]:

```
[Text(0.5, 1.0, 'Expend Vs SFRatio')]
```



The scatter plot depicts that the variables are negatively proportionate to each other implying that as the expenditure is increasing, the student faculty ratio is decreasing.

C. Relationship between a categorical and continuous variable

A box plot is used to study the relationship between a categorical and a continuous variable.

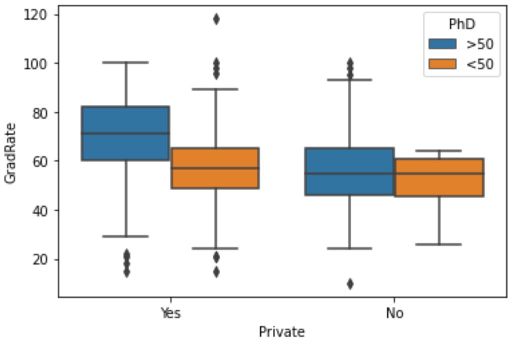
In [15]:

```
# Creating a box pot to study the relationship between Grad Rate & Private where Private is further categorized by PhD

sb.boxplot(x='Private', y='GradRate', hue="PhD",data=df_clg)
```

Out[15]:

```
<AxesSubplot:xlabel='Private', ylabel='GradRate'>
```



The median graduation rate is greater in private universities with PhD >50 as compared to non-private universities with PhD>50.Also, for non private universities, the median graduation rate is same for both the categories of PhD

Q4. Display unique values of a categorical variable and their frequencies

In [32]:

```
# Displaying the unique values of the categorical variable "PhD"
df_clg["PhD"].unique()
```

Out[32]:

```
array(['>50', '<50'], dtype=object)
```

The output clearly shows that there are 2 unique values of the categorical variable "PhD which are ">50" and "<50"

In [17]:

```
# Displaying the frequencies of the unique values of the categorical variable
df_clg["PhD"].value_counts()
```

Out[17]:

```
>50    670
<50     75
Name: PhD, dtype: int64
```

The above result implies that there are 670 colleges where the percentage of faculty with "PhD" is greater than 50% and there are around 75 colleges where the percentage of faculty with " PhD" is less than 50%

Q5(a). Build a contingency table of two potentially related categorical variables

In [18]:

```
# Creating a contingency table using crosstab function for two potentially related variables "PhD" and "Private
con_table = pd.crosstab(df_clg['PhD'], df_clg['Private'])

# Printing the contingency table
con_table
```

Out[18]:

Private	No	Yes
PhD		
<50	12	63
>50	196	474

The above contingency table displays the co-occurence of different values of "Private" and "PhD".

For example it shows that there are 196 colleges which are not private and where the percentage of faculty with "PhD" is greater than 50%

Q5(b). Conduct a statistical test of the independence between them and interpret the results.

The independence of the 2 categorical variables can be tested using a Chi-square test.

It will be carried out in 7 steps.

1. Formulating the null and alternative hypothesis

Null Hypothesis(H0) :The variables are independent.

Alternative Hypothesis(Ha):The variables are dependent on each other.

2. Statistical Test & Level of Significance.

Here, the Chi Square test will be used.

Level of Significance,α=0.05.

The rejection area is 5% at either tails of the probability distribution.

In [ ]:

3. Contingency Table

In order to calculate the p-value, a contingency table needs to be created, which has already been done using crosstab function.

In [19]:

```
# Printing the contingency table again for the reference.
con_table
```

Out[19]:

Private	No	Yes
PhD		
<50	12	63
>50	196	474

4. Calculating Chi-Square Test Statistic

In [33]:

```
# Importing scipy libraries
from scipy import stats

# Using function "chi2_contingency" to find the value of chi-square statistic.
# It returns 4 values;chi2,pvalue,degrees of freedom(dof) and the tables of expected values (expected)
chi2, p_val, dof, expected = stats.chi2_contingency(con_table)

# The calculated p-values are
print(f"The p-value is {p_val}")
```

```
The p-value is 0.02198044239708907
```

5. Conclusion

As p-value is less than 0.05, so we reject the null hypothesis and conclude that the two variables, PhD and Private are dependent on each other.

Q6. Retrieve one or more subset of rows based on 2 or more criteria and present descriptive statistics on the subsets.

SUBSET 1

In [21]:

```
# Selecting colleges with Accept greater than 10000 and Enroll greater than 2000 and storing it in a new data frame.
df_clg_subset1=df_clg[(df_clg.Accept>10000) & (df_clg.Enroll>2000)]

# Printing the subset 1
df_clg_subset1
```

Out[21]:

	Institution	Private	Apps	Accept	Enroll	Students	SFRatio	Expend	GradRate	PhD
61	University of California at Irvine	No	15698.0	10775.0	2478.0	13541.0	16.1	15934.0	66.0	>50
76	University of Michigan at Ann Arbor	No	19152.0	12940.0	4893.0	23384.0	11.5	14847.0	87.0	>50
175	University of Wisconsin at Madison	No	14901.0	10932.0	4631.0	26145.0	11.5	11006.0	72.0	>50
197	University of Delaware	Yes	14446.0	10516.0	3252.0	18652.0	18.3	10650.0	75.0	>50
204	Michigan State University	No	18114.0	15096.0	6180.0	30760.0	14.0	10520.0	71.0	>50
207	Rutgers at New Brunswick	No	48094.0	26330.0	4520.0	25113.0	19.5	10474.0	77.0	>50
216	University of Massachusetts at Amherst	No	14438.0	12414.0	3816.0	18222.0	16.7	10276.0	68.0	>50
316	University of Maryland at College Park	No	14292.0	10315.0	3409.0	23331.0	18.1	9021.0	63.0	>50
321	Pennsylvania State Univ. Main Campus	No	19315.0	10344.0	3450.0	30963.0	18.1	8992.0	63.0	>50
332	Virginia Tech	No	15712.0	11719.0	4277.0	19115.0	13.8	8944.0	73.0	>50
355	Indiana University at Bloomington	No	16587.0	13243.0	5873.0	27480.0	21.3	8686.0	68.0	>50
361	Purdue University at West Lafayette	No	21804.0	18744.0	5874.0	30278.0	18.2	8604.0	67.0	>50
368	University of Illinois - Urbana	No	14939.0	11652.0	5705.0	26333.0	17.4	8559.0	81.0	>50
377	Texas A&M Univ. at College Station	No	14474.0	10519.0	6392.0	34441.0	23.1	8471.0	69.0	>50
745	Arizona State University Main campus	No	12809.0	10308.0	3761.0	30178.0	18.9	4602.0	48.0	>50

In [22]:

```
# Finding the descriptive statistics for the subset 1
df_clg_subset1.describe()
```

Out[22]:

	Apps	Accept	Enroll	Students	SFRatio	Expend	GradRate
count	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000
mean	18318.333333	13056.466667	4567.400000	25195.733333	17.100000	9972.400000	69.866667
std	8584.121368	4325.631463	1216.946929	5848.016385	3.292416	2686.180315	8.919214
min	12809.000000	10308.000000	2478.000000	13541.000000	11.500000	4602.000000	48.000000
25%	14460.000000	10517.500000	3605.500000	21223.000000	15.050000	8645.000000	66.500000
50%	15698.000000	11652.000000	4520.000000	26145.000000	18.100000	9021.000000	69.000000
75%	18633.000000	13091.500000	5789.000000	30228.000000	18.600000	10585.000000	74.000000
max	48094.000000	26330.000000	6392.000000	34441.000000	23.100000	15934.000000	87.000000

It can be inferred that there are only 15 colleges with more than 10,000 applications and more than 2000 enrolments.

SUBSET 2

In [23]:

```
# Selecting colleges with
# Grad Rate greater than 60 and less than 76
# PhD equal to >50
# Expend less than 20,000
# Storing the subset of rows in a new data frame.
df_clg_subset2=df_clg[(df_clg.GradRate>60) & (df_clg.GradRate<76)&(df_clg.PhD == ">50") &(df_clg.Expend<20000)]

# Printing the subset 2
df_clg_subset2
```

Out[23]:

	Institution	Private	Apps	Accept	Enroll	Students	SFRatio	Expend	GradRate	PhD
27	Case Western Reserve University	Yes	3877.0	3156.0	713.0	3564.0	2.9	19733.0	67.0	>50
31	Sweet Briar College	Yes	462.0	402.0	146.0	568.0	6.5	18953.0	61.0	>50
34	Scripps College	Yes	855.0	632.0	139.0	576.0	8.2	18372.0	73.0	>50
35	Saint Louis University	Yes	3294.0	2855.0	956.0	5716.0	4.6	18367.0	67.0	>50
46	University of Southern California	Yes	12229.0	8498.0	2477.0	14688.0	11.4	17007.0	68.0	>50
...	...	...	...	...	...	...	...	...	...	...
739	Southwest Baptist University	Yes	1093.0	1093.0	642.0	2737.0	15.9	4718.0	71.0	>50
750	Radford University	No	5702.0	4894.0	1742.0	8549.0	19.6	4519.0	62.0	>50
760	Westfield State College	No	3100.0	2150.0	825.0	4175.0	15.7	4222.0	65.0	>50
765	Flagler College	Yes	1415.0	714.0	338.0	1389.0	18.1	3930.0	69.0	>50
769	Campbell University	Yes	2087.0	1339.0	657.0	4395.0	21.8	3739.0	63.0	>50

217 rows × 10 columns

In [24]:

```
# Finding the descriptive statistics for the subset 2
df_clg_subset2.describe()
```

Out[24]:

	Apps	Accept	Enroll	Students	SFRatio	Expend	GradRate
count	217.000000	217.000000	217.000000	217.000000	217.000000	217.000000	217.000000
mean	3349.345622	2437.566820	898.350230	5111.714286	13.963594	9349.069124	67.926267
std	4217.779181	3080.199858	1184.422111	7071.688534	3.423761	2947.480509	4.087481
min	167.000000	130.000000	46.000000	397.000000	2.900000	3739.000000	61.000000
25%	817.000000	632.000000	266.000000	1236.000000	11.600000	7473.000000	65.000000
50%	1457.000000	1080.000000	401.000000	1854.000000	13.400000	8847.000000	67.000000
75%	4019.000000	2855.000000	936.000000	5352.000000	16.200000	10520.000000	72.000000
max	21804.000000	18744.000000	6392.000000	35206.000000	23.100000	19733.000000	75.000000

It can inferred that there are only 217 colleges, where the expenditure incurred is less than 20000 and graduation rate is between 60% and 75 %.

Q7. Conduct a statistical test of the significance of the difference between the means of two subsets of the data and interpret the results

The data related to Graduation Rate for PhD > 50 and PhD < 50 will be extracted. The test will be conducted to test the significance of the difference between the means of the 2 subsets with PhD > 50 and PhD < 50.

In [25]:

```
# Subset 1

# Selecting the GradRate of colleges with "Phd = <50" and storing it in a new data frame.
df_clg_PhS= df_clg[df_clg['PhD'] == "<50" ]['GradRate']

# Printing the new dataframe
df_clg_PhS
```

Out[25]:

```
39     64.0
114    47.0
120    44.0
141    15.0
156    57.0
       ...
754    46.0
756    27.0
759    60.0
763    57.0
766    59.0
Name: GradRate, Length: 75, dtype: float64
```

In [26]:

```
# Subset 2

# Selecting the GradRate of colleges with "Phd = >50" and storing it in a new data frame.
df_clg_PhG=df_clg[df_clg['PhD']==">50"]["GradRate"]

# Printing the new dataframe
df_clg_PhG
```

Out[26]:

```
0       90.0
1       90.0
2       48.0
3       89.0
4       99.0
       ...
769     63.0
770     78.0
771     10.0
772    100.0
774     54.0
Name: GradRate, Length: 670, dtype: float64
```

Conducting the test for checking the equality of the means of 2 subsets.

The test of significance will be carried out in 4 steps.

1. Formulating the null and alternative hypothesis

Here, the null hypothesis will be that the difference between the two means is equal to 0, i.e. there is no difference between them against the alternative hypothesis that they are different.

H_0:μ **= 0 (Here,** μ **=** μ_S **-** μ_G**)**
where G represents PhD>50 and S represents PhD<50

H_A:μ **≠ 0**

2. Statistical Test & Level of Significance.

The subsets in consideration are unrelated to each other, so the INDEPENDENT TWO-SAMPLE t TEST will be carried out

to check the significance of the difference between 2 means.

Significance Level; α=0.05

The alternative hypothesis states that the mean Graduation Rate for PhD < 50 and mean Graduation Rate for PhD > 50 are different. Thus,a Two-tailed test will be used. The rejection area will appear both at the right and left tails of the probability distribution.

3.Calculating the test statistic

Calculating the value of the t-statistic, and the associated probability that there is no difference between the two means ("p-value") using ttest_ind function.

The ttest_ind function will take two arguments i.e. two series corresponding to the two samples, thus returning the t-statistic and the p-value.

In [27]:

```
# Calculating the value of test statistics
t_value, p_value = stats.ttest_ind(df_clg_PhS,df_clg_PhG )

# Printing the values
print(f"The t statistic for the test is {t_value} and the p value is {p_value}")
```

```
The t statistic for the test is -4.60941671836498 and the p value is 4.752455557237492e-06
```

4. Conclusion

Here, p_value<0.05, so we REJECT the null hypothesis that there is no difference between the means of the 2 subsets.

It implies that there is a difference between the means of the 2 subsets.

Also, here the t statistic is less than 0 i.e.

μ < 0 which means, μ_S - μ_G < 0 implying μ_S < μ_G

It clearly infers that the mean Graduation rate for the Colleges with PhD<50 is less than the mean Graduation rate for the colleges with PhD>50

Q8. Create one or more tables that group the data by a certain categorical variable and display summarized information for each group

In [28]:

```
# Grouping the data with PhD and displaying mean for all the numerical cariables in it
df_clg_T1=df_clg.groupby('PhD').mean(numeric_only=True)
df_clg_T1
```

Out[28]:

	Apps	Accept	Enroll	Students	SFRatio	Expend	GradRate
PhD							
<50	922.666667	748.600000	295.880000	1792.573333	14.309333	7479.093333	56.64000
>50	3239.211940	2170.465672	839.932836	4925.320896	14.086567	9888.829851	66.18209

In [29]:

```
# Grouping the data with PhD and displaying mean for all the numerical cariables in it
df_clg_T2=df_clg.groupby('Private').mean(numeric_only=True)
df_clg_T2
```

Out[29]:

	Apps	Accept	Enroll	Students	SFRatio	Expend	GradRate
Private							
No	5762.846154	3932.721154	1646.225962	10617.125000	17.132212	7485.274038	55.836538
Yes	1938.175047	1289.294227	451.640596	2283.139665	12.937989	10483.260708	68.856611

Here, the mean for both the groups have been calculated which will assist in generating the key findings for the data set i.e. it can be clearly observed from the above table that the mean no of application received by a private college university is quite less as compared to its counterpart.

Q9. Implement Linear regression model and interpret its output.¶

In [30]:

```
# Importing statsmodels library for implementing a multiple linear regression model
import statsmodels.api as sm
from bokeh.io import output_notebook
output_notebook()

from bokeh.plotting import figure
from bokeh.io import show
```

(https://bokeh.org) Loading BokehJS ...

In [31]:

```
# Taking Grad Rate as a dependent variable and the rest as the predictors.
model = sm.OLS.from_formula('GradRate~ Apps + Students + Expend + Enroll + PhD + Private', data=df_clg).fit()
model.summary()
```

Out[31]:

```
OLS Regression Results

     Dep. Variable:         GradRate          R-squared:      0.311
            Model:               OLS     Adj. R-squared:      0.306
           Method:     Least Squares         F-statistic:     55.57
             Date:  Thu, 15 Dec 2022  Prob (F-statistic):  1.21e-56
             Time:          22:17:08      Log-Likelihood:   -3038.5
 No. Observations:               745                 AIC:     6091.
     Df Residuals:               738                 BIC:     6123.
         Df Model:                 6
  Covariance Type:         nonrobust

                    coef  std err       t   P>|t|  [0.025   0.975]
      Intercept  41.9020    2.221  18.867   0.000  37.542   46.262
      PhD[T.>50]  7.7150    1.792   4.305   0.000   4.196   11.233
  Private[T.Yes] 11.8988    1.601   7.433   0.000   8.756   15.041
           Apps   0.0017    0.000   6.057   0.000   0.001    0.002
       Students  -0.0015    0.000  -5.618   0.000  -0.002   -0.001
         Expend   0.0006    0.000   4.987   0.000   0.000    0.001
         Enroll   0.0050    0.002   2.735   0.006   0.001    0.009

       Omnibus:  24.425     Durbin-Watson:     1.964
 Prob(Omnibus):   0.000  Jarque-Bera (JB):    41.011
          Skew:  -0.247          Prob(JB):  1.24e-09
      Kurtosis:   4.038         Cond. No.   6.51e+04


Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.51e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

The model was implemented twice to find the most parsimonious model. The earlier models (not shown here) were not parsimonious. Reimplemented the model by removing the "Accept" and "SF Ratio" variables as they were insignificant.

Interpretation of the Output:

1. Significance of the variables : The p values for all the predictors are less than 0.05 indicating that it is a PARSIMONIOUS model and all the variables are exerting a significant impact on Graduation Rate

2. Regression Equation : The model can be expressed as

Graduation Rate = 41.9020 + 7.7150 Phd>50 + 11.898 Private(Yes) + 0.017 Apps + 0.0006 Expend + 0.0050 Enroll - 0.0015

Students

Here, it is observed that the Graduation Rate is positively affected by Apps, Expend and Enroll whereas it is negatively affected by Students.

4. Goodness of Fit : The adjusted R square is 0.306 claiming that the model is not a good fit to the data as it is explaining only 30% of the causes of variation in the graduation rate.

3. Adequacy of the model : The model's adequacy can be determined by studying the scatter plot and histogram of the residuals. However, here the p value of Jarque-Bera test is less than 0.05 , making us reject the null Hypothesis of the normal distribution of residuals. It means that the model is unadequate and cannot be used to predict the graduation rate.

Key Fidings and Conclusion

From the above analysis, it can be concluded that there is more inclination towards the non-private college and universities. Also, the number of students are more in colleges having PhD>50.The graduation rate is also comparitively higher in colleges having PhD>50 implying that Private and PhD are 2 important factors affecting the Graduation Rate.