The EDA project need to include the following steps, presented in their sequence:

1. Generate descriptive statistics for the dataset.
2. Check any records with missing values, and handle the missing data as appropriate.
3. Build graphs visualizing the following and comment on the results: A) the distribution of one or more individual continuous variables B) the relationship of a pair of continuous variables. C) the relationship b/w a categorical variable and a continuous one.
4. Display unique values of a categorical variable and their frequencies.
5. Build a contingency table of two potentially related categorical variables. Conduct a statistical test of the independence between them and interpret the results.
6. Retrieve one or more subset of rows based on two or more criteria and present descriptive statistics on the subset(s).
7. Conduct a statistical test of the significance of the difference between the means of two subsets of the data and interpret the results.
8. Create one or more tables that group the data by a certain categorical variable and display summarized information for each group (e.g. the mean or sum within the group).
9. Implement a linear regression model and interpret its output.

Note: Each step of the analysis should have a clear header and be documented with comments that: describe what the step is meant to achieve, justify the implementation choices, and interpret the results of the step. Besides, the project report should start with an introduction section motivating the project (i.e., the issues to explore and their practical implications), and should finish with a conclusion section summarizing key findings and learning.

In [113]:

```
import pandas as pd
```

In [114]:

```
pip install openpyxl
```

```
Requirement already satisfied: openpyxl in c:\users\vijay\anaconda3\lib\site-packages (3.0.9)Note: you may need to restart
the kernel to use updated packages.

Requirement already satisfied: et-xmlfile in c:\users\vijay\anaconda3\lib\site-packages (from openpyxl) (1.1.0)
```

In [115]:

```
#Read file now
data_1 = pd.read_excel("C:/Users/Vijay/Downloads/college(1).xlsx")
```

In [116]:

```
#View data
data_1
```

Out[116]:

	Institution	Private	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate	PhD
0	Abilene Christian University	Yes	1660.0	1232.0	721.0	3422.0	18.1	7041.0	60.0	>50
1	Adelphi University	Yes	2186.0	1924.0	512.0	3910.0	12.2	10527.0	56.0	<50
2	Adrian College	Yes	1428.0	1097.0	336.0	1135.0	12.9	8735.0	54.0	>50
3	Agnes Scott College	Yes	417.0	349.0	137.0	573.0	7.7	19016.0	59.0	>50
4	Alaska Pacific University	Yes	193.0	146.0	55.0	1118.0	11.9	10922.0	15.0	>50
...	...	...	...	...	...	...	...	...	...	...
772	Worcester State College	No	2197.0	1515.0	543.0	5118.0	21.0	4469.0	40.0	>50
773	Xavier University	Yes	1959.0	1805.0	695.0	3956.0	13.3	9189.0	83.0	>50
774	Xavier University of Louisiana	Yes	2097.0	1915.0	695.0	2959.0	14.4	8323.0	49.0	>50
775	Yale University	Yes	10705.0	2453.0	1317.0	5300.0	5.8	40386.0	99.0	>50
776	York College of Pennsylvania	Yes	2989.0	1855.0	691.0	4714.0	18.1	4509.0	99.0	>50

777 rows × 10 columns

1. Generate descriptive statistics for the dataset.

In [118]:

```
data_1.shape
```

Out[118]:

```
(777, 10)
```

Total Number of rows = 777 and columns = 10

In [119]:

```
data_1.describe(include="all")
```

Out[119]:

	Institution	Private	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate	PhD
count	777	765	775.000000	774.000000	772.000000	776.000000	774.000000	775.000000	774.000000	772
unique	777	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2
top	Abilene Christian University	Yes	NaN	NaN	NaN	NaN	NaN	NaN	NaN	>50
freq	1	554	NaN	NaN	NaN	NaN	NaN	NaN	NaN	692
mean	NaN	NaN	3004.927742	2014.164083	781.781088	4550.208763	14.090698	9642.797419	65.449612	NaN
std	NaN	NaN	3874.120093	2447.981568	931.034168	5858.384381	3.965024	5210.996785	17.194855	NaN
min	NaN	NaN	81.000000	72.000000	35.000000	3.000000	2.500000	3186.000000	10.000000	NaN
25%	NaN	NaN	778.000000	601.750000	242.750000	1225.500000	11.500000	6747.500000	53.000000	NaN
50%	NaN	NaN	1558.000000	1109.500000	435.500000	2095.000000	13.600000	8367.000000	65.000000	NaN
75%	NaN	NaN	3635.000000	2418.500000	902.250000	5121.000000	16.500000	10816.000000	78.000000	NaN
max	NaN	NaN	48094.000000	26330.000000	6392.000000	38338.000000	39.800000	56233.000000	118.000000	NaN

Here, We can analyse from the table that there are Majority of Private Universities, and the PhD Faculty are greater than 50 in most universities. We check the averages(mean) and deduce that the average graduation rate is 65.44. The average number of students accepted in a university is 2014.

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

Checking the sum of missing values in each column

In [120]:

```
data_1.isnull().sum()
```

Out[120]:

```
Institution     0
Private        12
Apps            2
Accept          3
Enroll          5
Students        1
S.F.Ratio       3
Expend          2
Grad.Rate       3
PhD             5
dtype: int64
```

We can observe the total number of missing records in each column.

In [121]:

```
data_1['Private'].fillna("Yes", inplace = True)
data_1['PhD'].fillna(">50", inplace = True)
```

In [122]:

```
#checking the college data
data_1
```

Out[122]:

	Institution	Private	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate	PhD
0	Abilene Christian University	Yes	1660.0	1232.0	721.0	3422.0	18.1	7041.0	60.0	>50
1	Adelphi University	Yes	2186.0	1924.0	512.0	3910.0	12.2	10527.0	56.0	<50
2	Adrian College	Yes	1428.0	1097.0	336.0	1135.0	12.9	8735.0	54.0	>50
3	Agnes Scott College	Yes	417.0	349.0	137.0	573.0	7.7	19016.0	59.0	>50
4	Alaska Pacific University	Yes	193.0	146.0	55.0	1118.0	11.9	10922.0	15.0	>50
...	...	...	...	...	...	...	...	...	...	...
772	Worcester State College	No	2197.0	1515.0	543.0	5118.0	21.0	4469.0	40.0	>50
773	Xavier University	Yes	1959.0	1805.0	695.0	3956.0	13.3	9189.0	83.0	>50
774	Xavier University of Louisiana	Yes	2097.0	1915.0	695.0	2959.0	14.4	8323.0	49.0	>50
775	Yale University	Yes	10705.0	2453.0	1317.0	5300.0	5.8	40386.0	99.0	>50
776	York College of Pennsylvania	Yes	2989.0	1855.0	691.0	4714.0	18.1	4509.0	99.0	>50

777 rows × 10 columns

In [123]:

```
#Replace the missing records with the mean of column when its a numerical-non continuous variable
#Round off the mean to have a better understanding as decimals put no values in these fields
data_1['Apps'].fillna(round(data_1['Apps'].mean()),inplace = True)
data_1['Accept'].fillna(round(data_1['Accept'].mean()),inplace = True)
data_1['Enroll'].fillna(round(data_1['Enroll'].mean()),inplace = True)
data_1['Students'].fillna(round(data_1['Students'].mean()),inplace = True)
```

In [124]:

```
#Replace missing records of conrinous variables with the means of the columns
data_1['Grad.Rate'].fillna(data_1['Grad.Rate'].mean(), inplace = True)
data_1['Expend'].fillna(data_1['Expend'].mean(), inplace = True)
data_1['S.F.Ratio'].fillna(data_1['S.F.Ratio'].mean(), inplace = True)
```

In [125]:

```
#The data is now clean with no missing records left and is kept in a new variable
vijay = data_1
vijay
```

Out[125]:

	Institution	Private	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate	PhD
0	Abilene Christian University	Yes	1660.0	1232.0	721.0	3422.0	18.1	7041.0	60.0	>50
1	Adelphi University	Yes	2186.0	1924.0	512.0	3910.0	12.2	10527.0	56.0	<50
2	Adrian College	Yes	1428.0	1097.0	336.0	1135.0	12.9	8735.0	54.0	>50
3	Agnes Scott College	Yes	417.0	349.0	137.0	573.0	7.7	19016.0	59.0	>50
4	Alaska Pacific University	Yes	193.0	146.0	55.0	1118.0	11.9	10922.0	15.0	>50
...	...	...	...	...	...	...	...	...	...	...
772	Worcester State College	No	2197.0	1515.0	543.0	5118.0	21.0	4469.0	40.0	>50
773	Xavier University	Yes	1959.0	1805.0	695.0	3956.0	13.3	9189.0	83.0	>50
774	Xavier University of Louisiana	Yes	2097.0	1915.0	695.0	2959.0	14.4	8323.0	49.0	>50
775	Yale University	Yes	10705.0	2453.0	1317.0	5300.0	5.8	40386.0	99.0	>50
776	York College of Pennsylvania	Yes	2989.0	1855.0	691.0	4714.0	18.1	4509.0	99.0	>50

777 rows × 10 columns

In [126]:

```
#Check the statistics again and see if all the missing records are sorted
vijay.describe(include='all')
```

Out[126]:

	Institution	Private	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate	PhD
count	777	777	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777
unique	777	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2
top	Abilene Christian University	Yes	NaN	NaN	NaN	NaN	NaN	NaN	NaN	>50
freq	1	566	NaN	NaN	NaN	NaN	NaN	NaN	NaN	697
mean	NaN	NaN	3004.927928	2014.163449	781.782497	4550.208494	14.090698	9642.797419	65.449612	NaN
std	NaN	NaN	3869.124449	2443.245063	928.029855	5854.608432	3.957352	5204.277250	17.161586	NaN
min	NaN	NaN	81.000000	72.000000	35.000000	3.000000	2.500000	3186.000000	10.000000	NaN
25%	NaN	NaN	780.000000	604.000000	243.000000	1226.000000	11.500000	6751.000000	53.000000	NaN
50%	NaN	NaN	1561.000000	1110.000000	438.000000	2096.000000	13.600000	8377.000000	65.000000	NaN
75%	NaN	NaN	3624.000000	2402.000000	891.000000	5118.000000	16.500000	10813.000000	78.000000	NaN
max	NaN	NaN	48094.000000	26330.000000	6392.000000	38338.000000	39.800000	56233.000000	118.000000	NaN

3. Build graphs visualizing the following and comment on the results:

A) the distribution of one or more individual continuous variables

In [127]:

```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\vijay\anaconda3\lib\site-packages (3.5.1)
Requirement already satisfied: numpy>=1.17 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib) (1.21.5)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: packaging>=20.0 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: six>=1.5 in c:\users\vijay\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotli
b) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [128]:

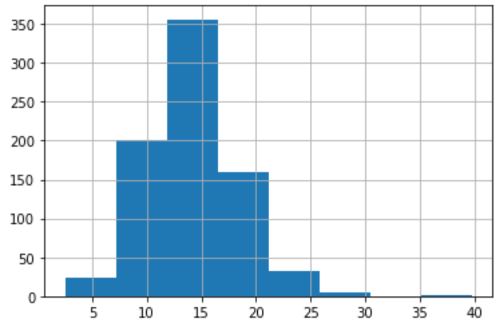
```
import matplotlib.pyplot as plt
```

In [129]:

```
#Use of histogram for continous variable
sf_ratio = vijay['S.F.Ratio'].hist(bins = 8)
sf_ratio
```

Out[129]:

```
<AxesSubplot:>
```



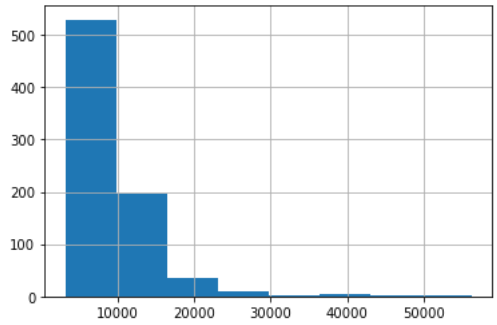
Here, The Histogram is normally distributed.

In [130]:

```
expend = vijay['Expend'].hist(bins = 8)
expend
```

Out[130]:

```
<AxesSubplot:>
```



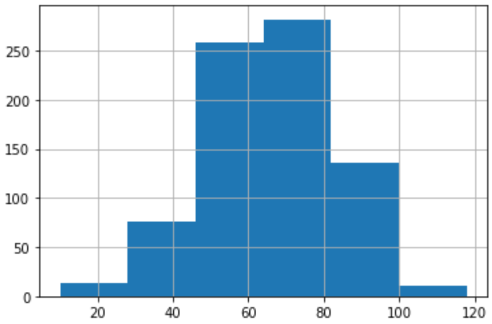
Here, The graph is right skewed.

In [131]:

```
G_rate= vijay['Grad.Rate'].hist(bins =6)
G_rate
```

Out[131]:

```
<AxesSubplot:>
```



Here, The Histogram is normally distributed

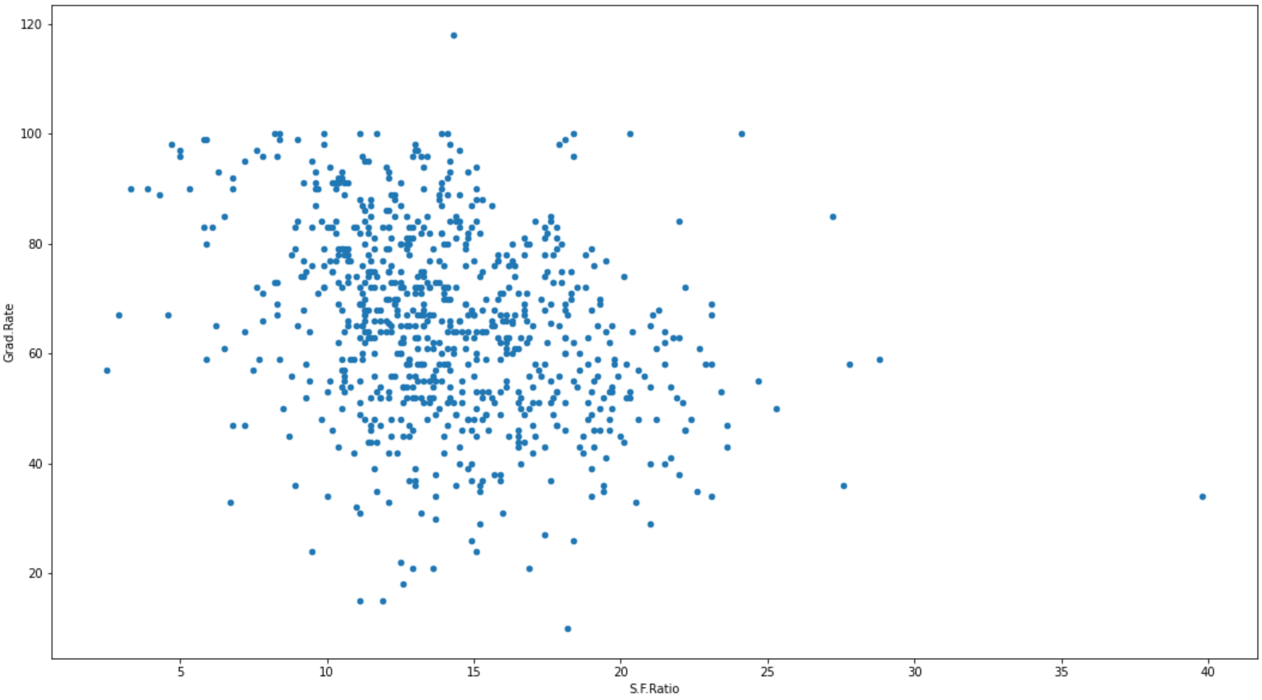
B) Relationship of a pair of continuous variables.

In [132]:

```
#We use scatterplot for pair of continuous variables
rela_1 = vijay.plot.scatter(x="S.F.Ratio",y="Grad.Rate", figsize=(18, 10))
rela_1
```

Out[132]:

```
<AxesSubplot:xlabel='S.F.Ratio', ylabel='Grad.Rate'>
```



In [133]:

```
pip install seaborn
```

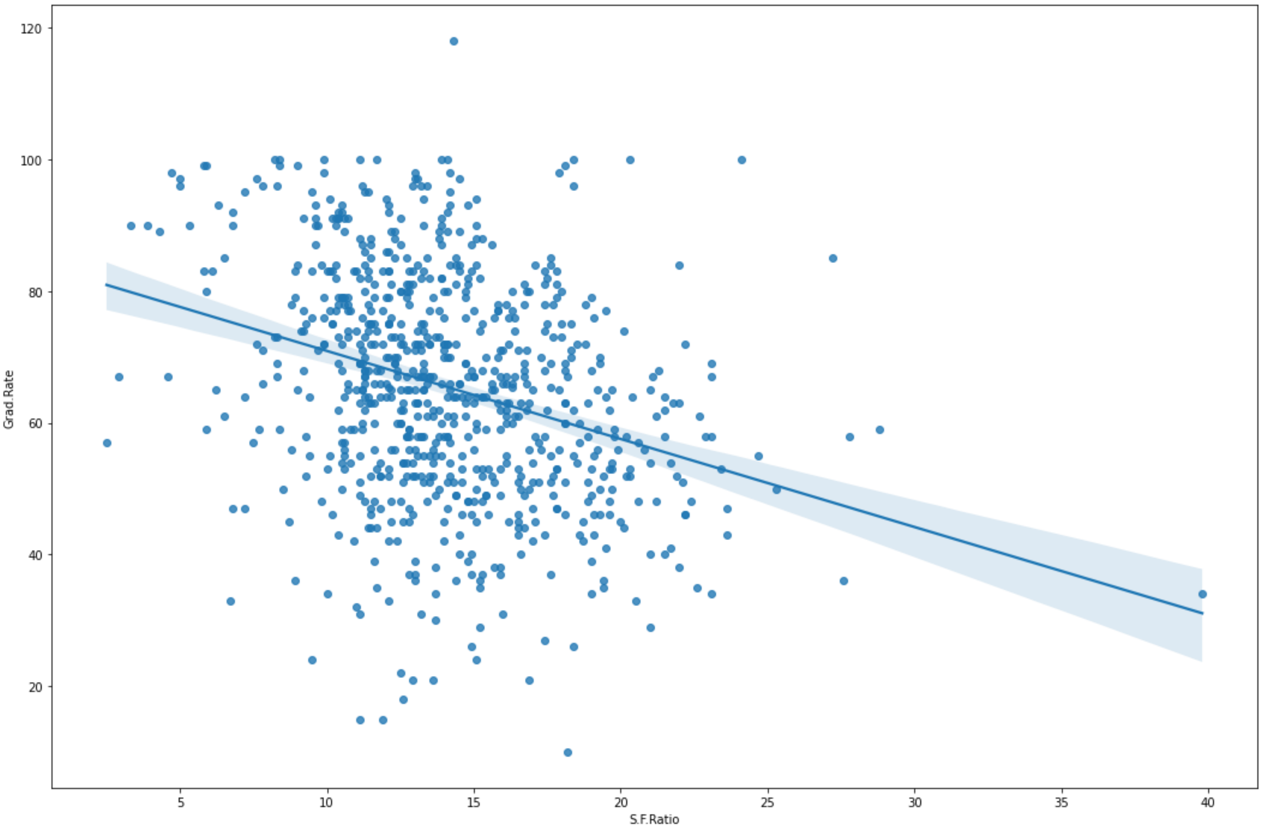
```
Requirement already satisfied: seaborn in c:\users\vijay\anaconda3\lib\site-packages (0.11.2)
Requirement already satisfied: pandas>=0.23 in c:\users\vijay\anaconda3\lib\site-packages (from seaborn) (1.4.2)
Requirement already satisfied: matplotlib>=2.2 in c:\users\vijay\anaconda3\lib\site-packages (from seaborn) (3.5.1)
Requirement already satisfied: scipy>=1.0 in c:\users\vijay\anaconda3\lib\site-packages (from seaborn) (1.7.3)
Requirement already satisfied: numpy>=1.15 in c:\users\vijay\anaconda3\lib\site-packages (from seaborn) (1.21.5)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib>=2.2->seabo
rn) (1.3.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib>=2.2->se
aborn) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn)
(9.0.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib>=2.2->seabo
rn) (4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib>=2.2->seabor
n) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn)
(0.11.0)
Requirement already satisfied: packaging>=20.0 in c:\users\vijay\anaconda3\lib\site-packages (from matplotlib>=2.2->seabor
n) (21.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\vijay\anaconda3\lib\site-packages (from pandas>=0.23->seaborn) (202
1.3)
Requirement already satisfied: six>=1.5 in c:\users\vijay\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotli
b>=2.2->seaborn) (1.16.0)
```

In [134]:

```
import seaborn
```

In [135]:

```
rela_2 = seaborn.regplot(x=vijay['S.F.Ratio'],
                y=vijay['Grad.Rate'], ci=95).figure.set_size_inches(18, 12)
rela_2
```



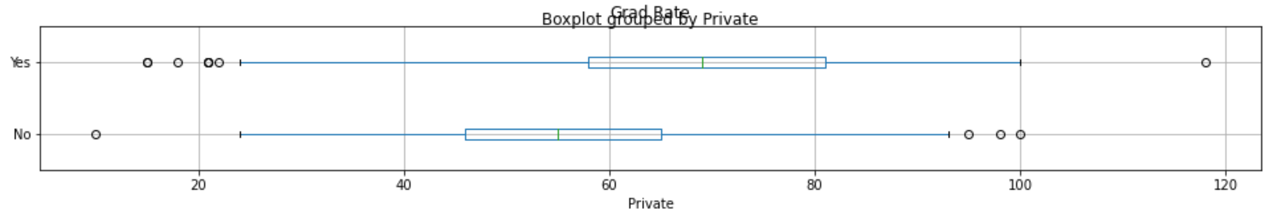
C) The relationship b/w a categorical variable and a continuous one.

In [136]:

```
rela_3 =vijay.boxplot(column = 'Grad.Rate',by = 'Private', figsize=(16,2),vert=False )
rela_3
```

Out[136]:

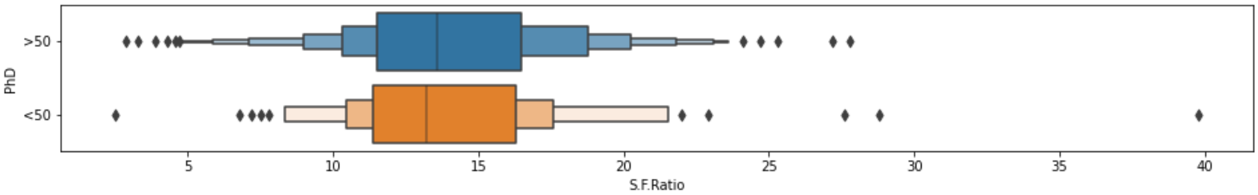
```
<AxesSubplot:title={'center':'Grad.Rate'}, xlabel='Private'>
```



Here, In the above boxplot there are few outliers in the data that can be ignored . We can see the both Grad Rate and Private Universities is normally distributed.

In [137]:

```
rela_4 = seaborn.boxenplot(data = vijay, x = 'S.F.Ratio', y='PhD', orient='h').figure.set_size_inches(16, 2)
rela_4
```



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

In [138]:

```
PhD_a = vijay['PhD'].value_counts()
PhD_a
```

Out[138]:

```
>50    697
<50     80
Name: PhD, dtype: int64
```

In [139]:

```
vijay.groupby(['PhD']).size()
```

Out[139]:

```
PhD
<50     80
>50    697
dtype: int64
```

Here, There are 80 values in th PhD column which are leass than 50 and 697 values in the column that are greater than 50. So faculty with PhD qualifications are always more than 50 in most Universities.

5. Build a contingency table of two potentially related categorical variables. Conduct a statistical test of the independence between them and interpret the results.

In [140]:

```
pip install scipy
```

```
Requirement already satisfied: scipy in c:\users\vijay\anaconda3\lib\site-packages (1.7.3)
Requirement already satisfied: numpy<1.23.0,>=1.16.5 in c:\users\vijay\anaconda3\lib\site-packages (from scipy) (1.21.5)
Note: you may need to restart the kernel to use updated packages.
```

In [141]:

```
from scipy import stats
```

In [142]:

```
table1 = pd.crosstab(vijay['Private'], vijay['PhD'])
table1
```

Out[142]:

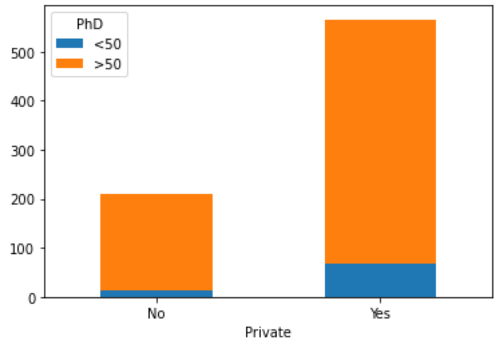
PhD	<50	>50
Private		
No	12	199
Yes	68	498

In [143]:

```
table1.plot(kind="bar", stacked=True, rot=0)
```

Out[143]:

```
<AxesSubplot:xlabel='Private'>
```



Here, From the Bar chart we can conclude that private Universities generally have faculty more than 50 with PhD qualifications.

In [144]:

```
chi2, p_val, dof, expected = stats.chi2_contingency(table1)

#print the p_value
print(f"p-value: {p_val}")
```

```
p-value: 0.014352590870496078
```

Here, p-value is less that 0.05. Therefor, we reject Null Hypothesis

6. Retrieve one or more subset of rows based on two or more criteria and present descriptive statistics on the subset(s).

In [145]:

```
#subset with Applications > 1400 and acceptance > 80% of all Applicarions
df_S1 = vijay[(vijay['Apps']>=1400) & (vijay['Accept']>0.8 * vijay['Apps'])]
df_S1
```

Out[145]:

	Institution	Private	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate	PhD
1	Adelphi University	Yes	2186.0	1924.0	512.0	3910.0	12.2	10527.0	56.0	<50
7	Albion College	Yes	1899.0	1720.0	782.0	1626.0	13.7	11487.0	73.0	>50
10	Alfred University	Yes	1732.0	1425.0	472.0	1940.0	11.3	10932.0	73.0	>50
23	Arizona State University Main campus	No	12809.0	10308.0	3761.0	30178.0	18.9	4602.0	48.0	>50
25	Arkansas Tech University	No	1734.0	1729.0	951.0	4541.0	19.6	4739.0	48.0	>50
...	...	...	...	...	...	...	...	...	...	...
760	Willamette University	Yes	1658.0	1327.0	395.0	1754.0	13.3	10779.0	68.0	>50
769	Wittenberg University	Yes	1979.0	1739.0	575.0	2124.0	12.8	10414.0	78.0	>50
771	Worcester Polytechnic Institute	Yes	2768.0	2314.0	682.0	2888.0	15.2	10774.0	82.0	>50
773	Xavier University	Yes	1959.0	1805.0	695.0	3956.0	13.3	9189.0	83.0	>50
774	Xavier University of Louisiana	Yes	2097.0	1915.0	695.0	2959.0	14.4	8323.0	49.0	>50

123 rows × 10 columns

In [146]:

```
#Subset with students greater than 2100 but enrollment < 500
df_S2 = vijay[(vijay['Students']>=2100) & (vijay['Enroll']<500)]
df_S2
```

Out[146]:

	Institution	Private	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate	PhD
14	Alverno College	Yes	494.0	313.0	157.0	2552.0	11.1	8127.0	55.000000	<50
17	Anderson University	Yes	1216.0	908.0	423.0	2100.0	12.1	7994.0	65.449612	<50
26	Assumption College	Yes	2135.0	1700.0	491.0	2397.0	13.8	7100.0	88.000000	>50
28	Augsburg College	Yes	662.0	513.0	257.0	2800.0	12.8	7836.0	58.000000	>50
38	Barry University	Yes	990.0	784.0	279.0	4955.0	12.6	9084.0	72.000000	>50
...	...	...	...	...	...	...	...	...	...	...
732	Webster University	Yes	665.0	462.0	226.0	3289.0	20.6	6951.0	48.000000	>50
735	Wentworth Institute of Technology	Yes	1480.0	1257.0	452.0	3533.0	15.4	17858.0	64.000000	<50
739	West Liberty State College	No	1164.0	1062.0	478.0	2365.0	16.3	4249.0	60.000000	<50
744	Western New England College	Yes	1650.0	1471.0	409.0	2919.0	15.4	8409.0	59.000000	>50
759	Wilkes University	Yes	1631.0	1431.0	434.0	2406.0	13.3	8543.0	67.000000	>50

77 rows × 10 columns

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

In [147]:

```
df_S1['Grad.Rate'].mean()
```

Out[147]:

```
64.0280456292935
```

In [148]:

```
df_S2['Grad.Rate'].mean()
```

Out[148]:

```
63.36947548575456
```

Independent two samples t-test

H0 : μ1 - μ2 = 0

H1 : μ1 - μ2 != 0

α = 0.05

In [149]:

```
t_val,p_val = stats.ttest_ind(df_S1['Grad.Rate'],df_S2['Grad.Rate'])

print(f"t-value : {t_val}, p-value : {p_val}")
```

```
t-value : 0.27196266362715515, p-value : 0.7859341024654471
```

Here the p-value is more that 0.05. Therefor, We fail to reject the Null Hypothesis

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

In [150]:

```
private_me = vijay.groupby(['Private']).median().round(2)
private_me
```

Out[150]:

	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate
Private							
No	4345.0	2900.0	1372.0	9068.0	17.30	6717.0	55.0
Yes	1152.5	861.5	333.0	1608.5	12.75	8954.0	69.0

Here, We find that the median of all the categories with respect to Private and non Private Universities. We can see how students and faculty ratio in non Private univesities is more than the private universities. We can conclude that the expenditure is also low in non private universities. The graduations rate of the private universities is however better.

In [151]:

```
private_me = vijay.groupby(['Private']).mean().round(2)
private_me
```

Out[151]:

	Apps	Accept	Enroll	Students	S.F.Ratio	Expend	Grad.Rate
Private							
No	5742.28	3901.06	1644.75	10567.13	17.14	7464.30	56.02
Yes	1984.46	1310.74	460.07	2307.15	12.95	10454.92	68.97

9. Implement a linear regression model and interpret its output.

In [152]:

```
pip install statsmodels
```

```
Requirement already satisfied: statsmodels in c:\users\vijay\anaconda3\lib\site-packages (0.13.2)
Requirement already satisfied: numpy>=1.17 in c:\users\vijay\anaconda3\lib\site-packages (from statsmodels) (1.21.5)
Requirement already satisfied: scipy>=1.3 in c:\users\vijay\anaconda3\lib\site-packages (from statsmodels) (1.7.3)
Requirement already satisfied: pandas>=0.25 in c:\users\vijay\anaconda3\lib\site-packages (from statsmodels) (1.4.2)
Requirement already satisfied: patsy>=0.5.2 in c:\users\vijay\anaconda3\lib\site-packages (from statsmodels) (0.5.2)
Requirement already satisfied: packaging>=21.3 in c:\users\vijay\anaconda3\lib\site-packages (from statsmodels) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\vijay\anaconda3\lib\site-packages (from packaging>=21.3
->statsmodels) (3.0.4)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\vijay\anaconda3\lib\site-packages (from pandas>=0.25->sta
tsmodels) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\vijay\anaconda3\lib\site-packages (from pandas>=0.25->statsmodels)
(2021.3)
Requirement already satisfied: six in c:\users\vijay\anaconda3\lib\site-packages (from patsy>=0.5.2->statsmodels) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [153]:

```
import statsmodels.api as sm
```

In [103]:

```
x = vijay[['Students','Apps','Accept','Enroll','S.F.Ratio','Expend']]
y = vijay['Grad.Rate']
model = sm.OLS(y,x).fit()
model.summary()
```

Out[103]:

```
OLS Regression Results

    Dep. Variable:        Grad.Rate       R-squared (uncentered):   0.929
           Model:               OLS  Adj. R-squared (uncentered):   0.929
          Method:     Least Squares                  F-statistic:   1686.
            Date:  Fri, 16 Dec 2022            Prob (F-statistic):   0.00
            Time:          06:29:43               Log-Likelihood:  -3348.6
No. Observations:               777                          AIC:   6709.
    Df Residuals:               771                          BIC:   6737.
        Df Model:                 6
 Covariance Type:         nonrobust

                coef  std err       t   P>|t|  [0.025  0.975]
 Students    -0.0022    0.000  -6.899   0.000  -0.003  -0.002
     Apps  3.931e-05    0.001   0.073   0.942  -0.001   0.001
   Accept     0.0021    0.001   2.085   0.037   0.000   0.004
   Enroll     0.0047    0.003   1.856   0.064  -0.000   0.010
S.F.Ratio     2.7117    0.082  33.190   0.000   2.551   2.872
   Expend     0.0028    0.000  26.883   0.000   0.003   0.003

      Omnibus:  92.105    Durbin-Watson:     1.854
Prob(Omnibus):   0.000  Jarque-Bera (JB):  277.588
         Skew:  -0.577         Prob(JB):  5.28e-61
     Kurtosis:   5.691         Cond. No.  1.62e+03


Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 1.62e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]: