ECON 312- Data Science

Python Assignment – ISLP Ch. 2

Submitted by

Musa Ahmed(BE-135-075)

Iffat Anan(BE-135-127)

Md. Abu Sayem Pappu(BE-135-002)

Shapnil Das(JN-135-081)

Submitted to

Dr. Rushad Faridi

(Assistant Professor)

Department of Economics

University of Dhaka

This exercise relates to the College data set, which can be found in the file College.csv on the book website. It contains a number of variables for 777 different universities and colleges in the US.

The variables are:

- **Private : Public/private indicator**
- **Apps : Number of applications received**
- **Accept : Number of applicants accepted**
- **Enroll : Number of new students enrolled**
- **Top10perc : New students from top 10% of high school class**
- **Top25perc : New students from top 25% of high school class**
- **F.Undergrad : Number of full-time undergraduates**
- **P.Undergrad : Number of part-time undergraduates**
- **Outstate : Out-of-state tuition**
- **Room.Board : Room and board costs**
- **Books : Estimated book costs**
- **Personal : Estimated personal spending**
- **PhD : Percent of faculty with Ph.D.s**
- **Terminal : Percent of faculty with terminal degree**
- **S.F.Ratio : Student/faculty ratio**
- **perc.alumni : Percent of alumni who donate**
- **Expend : Instructional expenditure per student**
- **Grad.Rate : Graduation rate**

Before reading the data into Python, it can be viewed in Excel or a text editor.

(a) Use the pd.read_csv() function to read the data into Python. Call the loaded data college. Make sure that you have the directory set to the correct location for the data.

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
college = pd.read_csv('College.csv')
college.head()
```

Out[2]:

	Unnamed: 0	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Underg
0	Abilene Christian University	Yes	1660	1232	721	23	52	2885	
1	Adelphi University	Yes	2186	1924	512	16	29	2683	
2	Adrian College	Yes	1428	1097	336	22	50	1036	
3	Agnes Scott College	Yes	417	349	137	60	89	510	
4	Alaska Pacific University	Yes	193	146	55	16	44	249	

(b) Look at the data used in the notebook by creating and running a new cell with just the code college in it. You should notice that the first column is just the name of each university in a column named something like Unnamed: 0. We don't really want pandas to treat this as data. However, it may be handy to have these names for later. Try the following commands and similarly
look at the resulting data frames:
This has used the first column in the file as an index for the data frame. This means that pandas has given each row a name
corresponding to the appropriate university. Now you should see that the first data column is Private. Note that the names of
the colleges appear on the left of the table. We also introduced a new python object above: a dictionary, which is specified by dictionary
(key, value) pairs. Keep your modified version of the data with the following:

In [3]:
```
college = college.rename({"Unnamed: 0": "College"}, axis=1)
college = college.set_index("College")
college.head()
```

Out[3]:

College	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad
Abilene Christian University	Yes	1660	1232	721	23	52	2885	537
Adelphi University	Yes	2186	1924	512	16	29	2683	1227
Adrian College	Yes	1428	1097	336	22	50	1036	99
Agnes Scott College	Yes	417	349	137	60	89	510	63
Alaska Pacific University	Yes	193	146	55	16	44	249	869

(c) Use the describe() method of to produce a numerical summary of the variables in the data set.

In [4]:
```
college.describe()
```

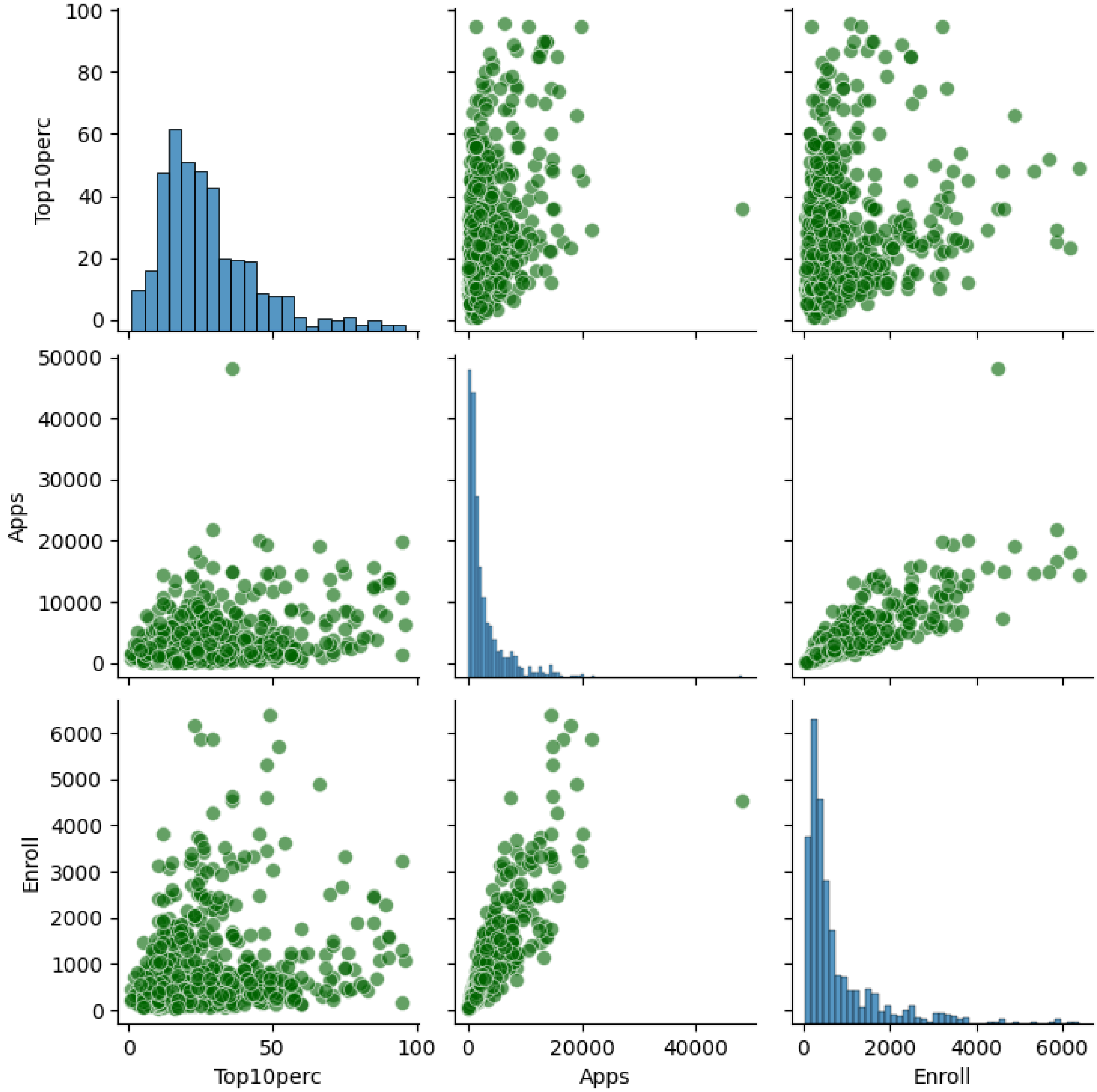
Out[4]:

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.U
count	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	7
mean	3001.638353	2018.804376	779.972973	27.558559	55.796654	3699.907336	8
std	3870.201484	2451.113971	929.176190	17.640364	19.804778	4850.420531	15
min	81.000000	72.000000	35.000000	1.000000	9.000000	139.000000	
25%	776.000000	604.000000	242.000000	15.000000	41.000000	992.000000	
50%	1558.000000	1110.000000	434.000000	23.000000	54.000000	1707.000000	3
75%	3624.000000	2424.000000	902.000000	35.000000	69.000000	4005.000000	9
max	48094.000000	26330.000000	6392.000000	96.000000	100.000000	31643.000000	218

(d) Use the pd.plotting.scatter_matrix() function to produce a scatterplot matrix of the first columns [Top10perc, Apps, Enroll].
Recall that you can reference a list C of columns of a data frame A using A[C].

In [5]:
```
sns.pairplot(data=college,vars=["Top10perc","Apps","Enroll"],plot_kws={'alpha': 0.6
```

Out[5]:
```
<seaborn.axisgrid.PairGrid at 0x1fa2267cec0>
```

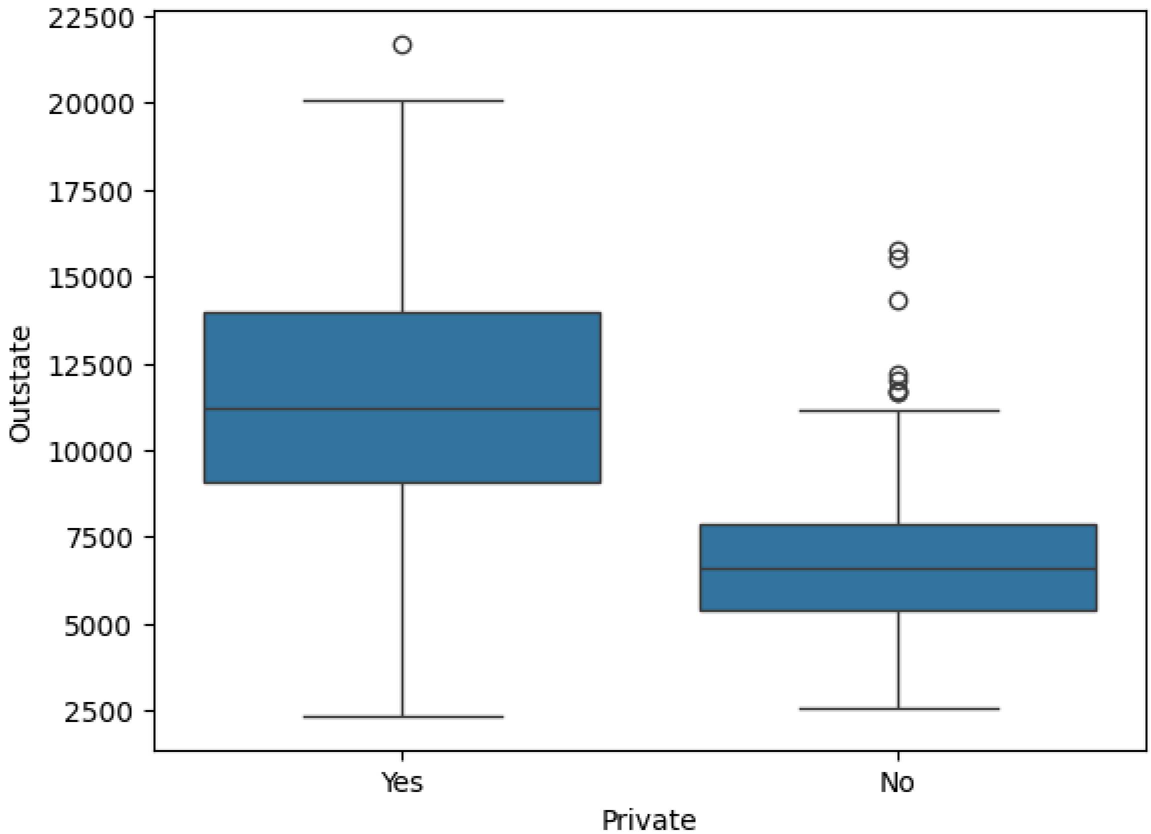


(e) Use the boxplot() method of college to produce side-by-side boxplots of Outstate versus Private.

In [6]:

```
sns.boxplot(data=college,y="Outstate",x="Private")
```

Out[6]: <Axes: xlabel='Private', ylabel='Outstate'>



(f) Create a new qualitative variable, called Elite, by binning the Top10perc variable into two groups based on whether or not the
proportion of students coming from the top 10% of their high school classes exceeds 50%.

In [7]:
```
college["Elite"] = pd.cut(college["Top10perc"],bins=[0,50,100],labels=["No","Yes"])
college["Elite"]
```

Out[7]:
```
College
Abilene Christian University       No
Adelphi University                 No
Adrian College                     No
Agnes Scott College               Yes
Alaska Pacific University          No
                                 ...
Worcester State College            No
Xavier University                  No
Xavier University of Louisiana     No
Yale University                   Yes
York College of Pennsylvania       No
Name: Elite, Length: 777, dtype: category
Categories (2, object): ['No' < 'Yes']
```

Use the value_counts() method of college['Elite'] to see how many elite universities there are. Finally, use the boxplot() method again to produce side-by-side boxplots of Outstate versus Elite.

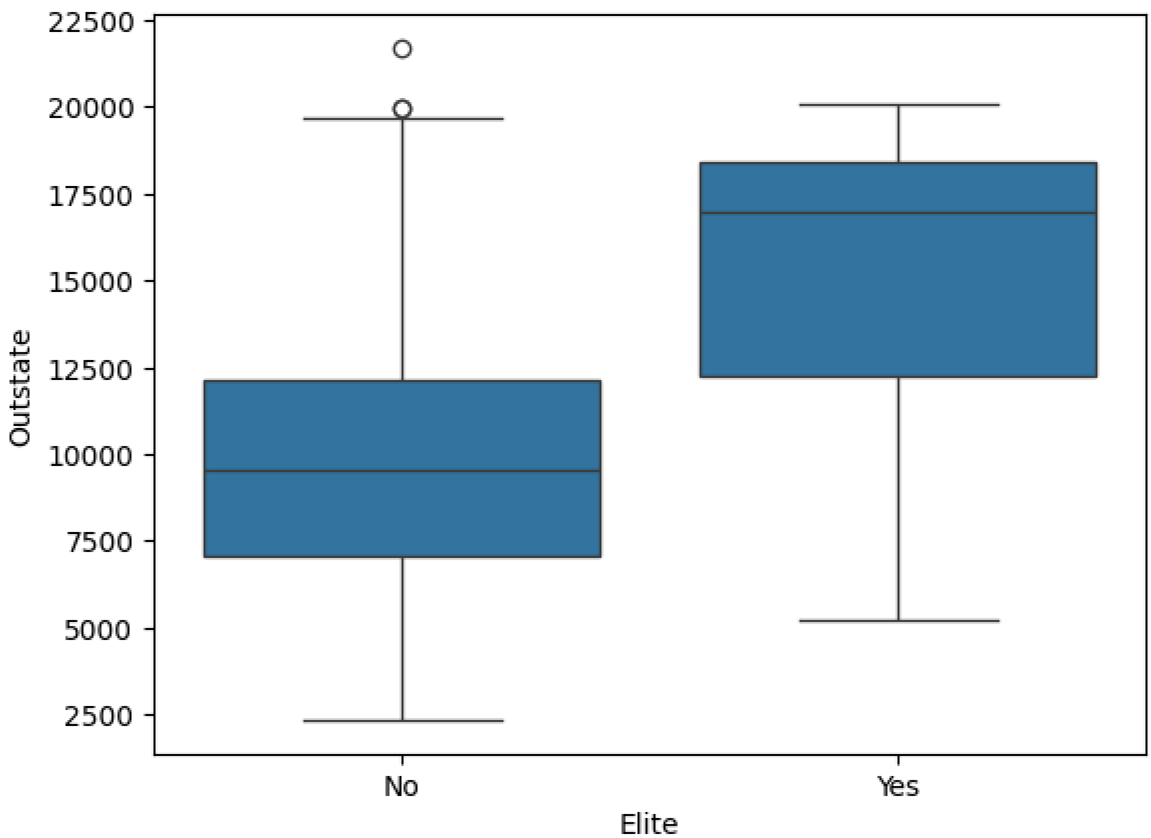
In [8]:
```python
college["Elite"].value_counts()
```

Out[8]:
```
Elite
No     699
Yes     78
Name: count, dtype: int64
```

There are **78** Elite universities.

In [9]:
```python
sns.boxplot(data=college,x="Elite",y="Outstate")
```

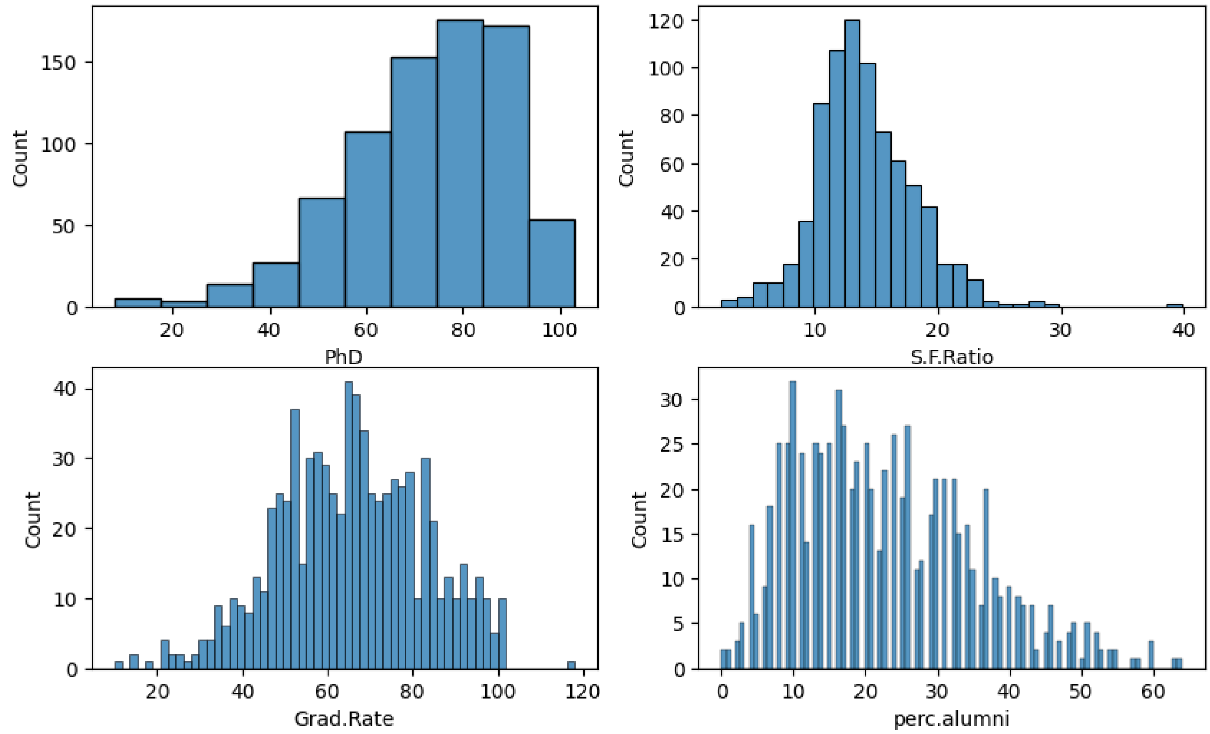
Out[9]:
```
<Axes: xlabel='Elite', ylabel='Outstate'>
```



(g) Use the plot.hist() method of college to produce some histograms with differing numbers of bins for a few of the quantitative variables. The command plt.subplots(2, 2) may be useful:
it will divide the plot window into four regions so that four plots can be made simultaneously. By changing the arguments you can divide the screen up in other combinations.

In [10]:
```python
fig, axes = plt.subplots(2, 2, figsize=(10, 6))
sns.histplot(x=college["PhD"],bins=10 ,ax=axes[0,0])
sns.histplot(x=college["S.F.Ratio"],bins=30,ax=axes[0,1])
sns.histplot(x=college["Grad.Rate"],bins=60,ax=axes[1,0])
sns.histplot(x=college["perc.alumni"],bins=100,ax=axes[1,1])
```

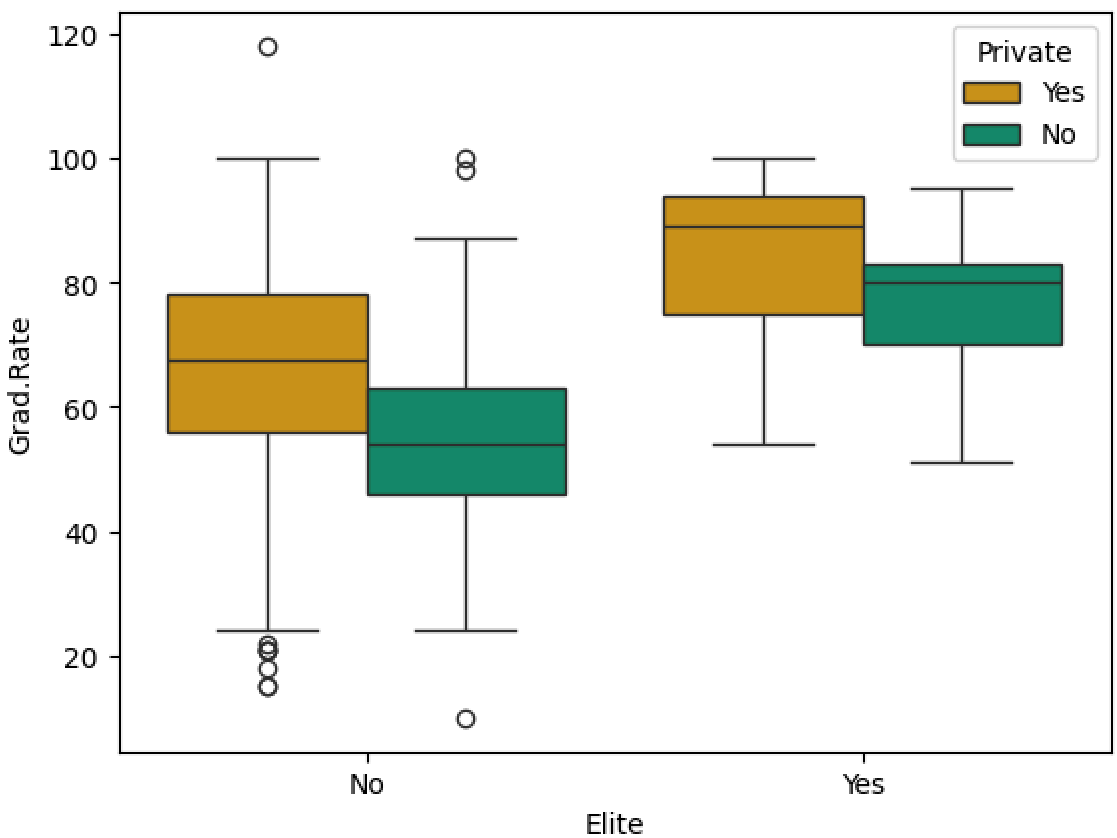
Out[10]:
```
<Axes: xlabel='perc.alumni', ylabel='Count'>
```



(h) Continue exploring the data, and provide a brief summary of what you discover.

In [11]:
```
sns.boxplot(data=college,x="Elite",y="Grad.Rate",hue="Private",palette=["#E69F00","
```

Out[11]: <Axes: xlabel='Elite', ylabel='Grad.Rate'>



Colleges classified as elite, meaning they have more students from the top 10% of their high school class, tend to have higher graduation rates.
In both elite and non-elite groups, private colleges generally show better graduation outcomes than public colleges.

10. This exercise involves the Boston housing data set.
(a) To begin, load in the Boston data set, which is part of the ISLP library.

In [12]:
```
from sklearn.datasets import fetch_openml
boston = fetch_openml(name="boston", version=1, as_frame=True)
df = boston.frame
```

(b) How many rows are in this data set? How many columns? What do the rows and columns represent?

In [13]:
```
df.head()
```

Out[13]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90

In [14]:
```
print(f"Number of Rows = {df.shape[0]}")
print(f"Number of Column = {df.shape[1]}")
print(boston.DESCR)
```

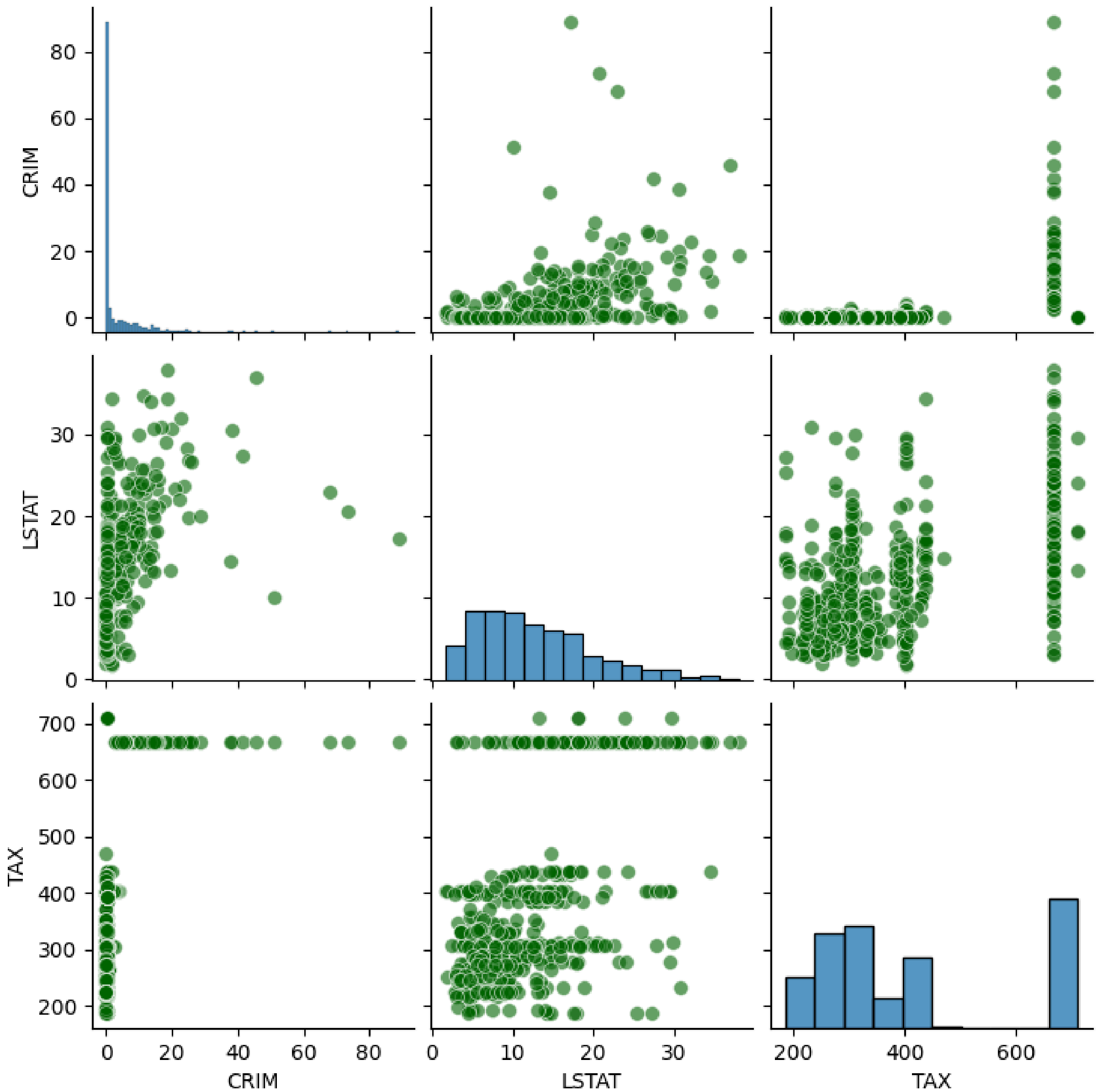
```
Number of Rows = 506
Number of Column = 14
**Author**:
**Source**: Unknown - Date unknown
**Please cite**:

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980.   N.B. Various transformations are used in the table on
pages 244-261 of the latter.
Variables in order:
CRIM     per capita crime rate by town
ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS    proportion of non-retail business acres per town
CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX      nitric oxides concentration (parts per 10 million)
RM       average number of rooms per dwelling
AGE      proportion of owner-occupied units built prior to 1940
DIS      weighted distances to five Boston employment centres
RAD      index of accessibility to radial highways
TAX      full-value property-tax rate per $10,000
PTRATIO  pupil-teacher ratio by town
B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT    % lower status of the population
MEDV     Median value of owner-occupied homes in $1000's


Information about the dataset
CLASSTYPE: numeric
CLASSINDEX: last

Downloaded from openml.org.
```

Also each row represents a suburb of Boston ,so there are **506** suburbs in the dataset with **13** feature variable mentioned above and a target variavle (Median price of House per 1000s)

(c) Make some pairwise scatterplots of the predictors (columns) in this data set. Describe your findings.

In [15]:
```
sns.pairplot(data=df,vars=["CRIM","LSTAT","TAX"],plot_kws={'alpha': 0.6, 's': 50,"c
```

Out[15]:
```
<seaborn.axisgrid.PairGrid at 0x1fa26c4e490>
```



(d) Are any of the predictors associated with per capita crime rate? If so, explain the relationship.

In [16]:
```
df.corr(numeric_only=True)["CRIM"].sort_values()
```

Out[16]:
```
MEDV      -0.388305
B         -0.385064
DIS       -0.379670
RM        -0.219247
ZN        -0.200469
PTRATIO    0.289946
AGE        0.352734
INDUS      0.406583
NOX        0.420972
LSTAT      0.455621
TAX        0.582764
CRIM       1.000000
Name: CRIM, dtype: float64
```

The correlations show that **crime rate (CRIM)** is positively associated with features like **TAX**, **LSTAT**, **NOX**, **INDUS**, and **AGE**, meaning areas with higher taxes, pollution, industry, and lower socioeconomic status tend to have more crime. Conversely, features like **DIS**, **RM**, **ZN**, and **B** are negatively correlated, suggesting that areas farther from the city center, with larger homes, more zoning for residential use.

(e) Do any of the suburbs of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the range of each predictor.

In [17]:
```
df["CRIM"].describe()
```

Out[17]:
```
count    506.000000
mean       3.613524
std        8.601545
min        0.006320
25%        0.082045
50%        0.256510
75%        3.677083
max       88.976200
Name: CRIM, dtype: float64
```

In [18]:
```
df["TAX"].describe()
```

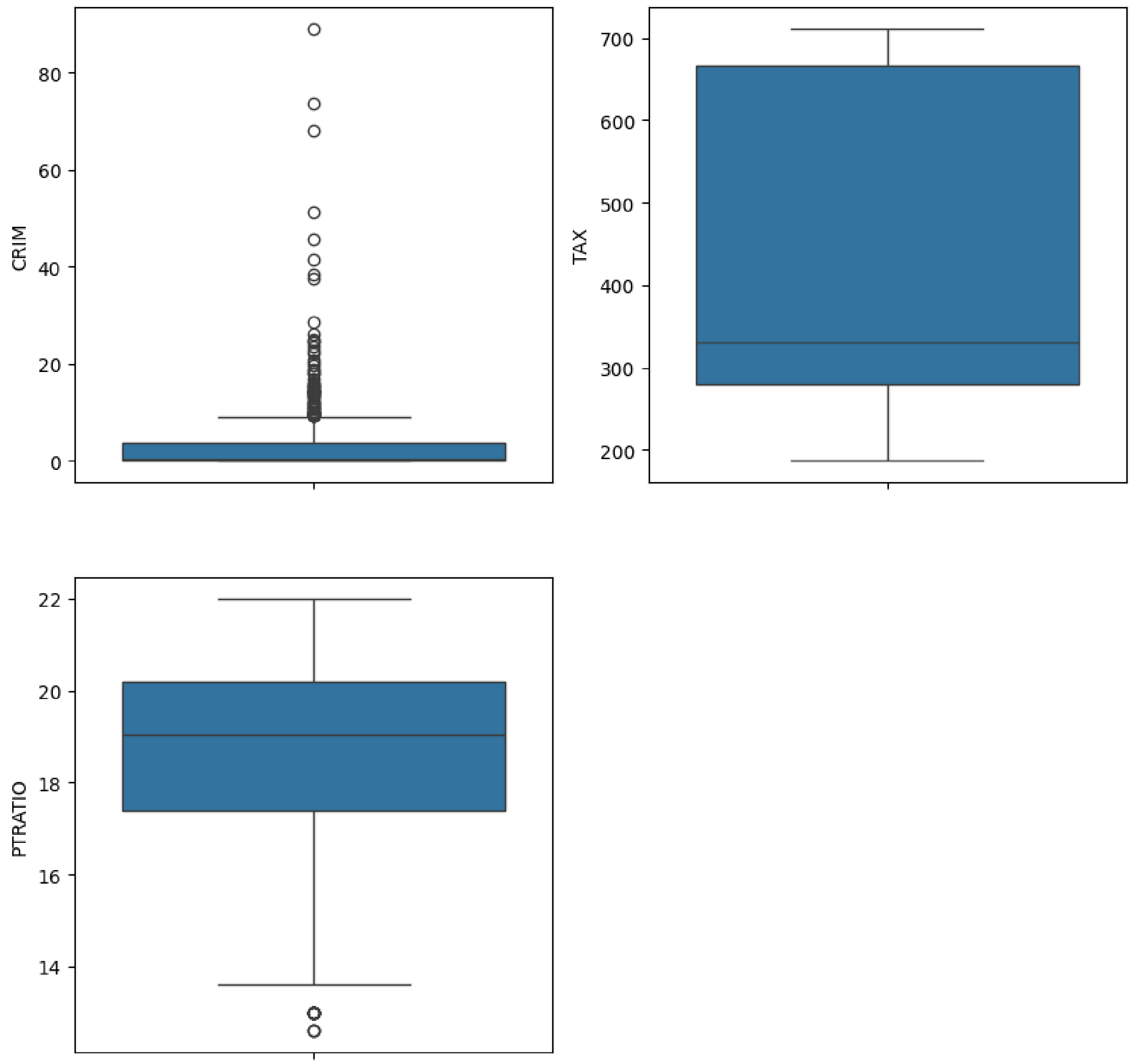
Out[18]:
```
count    506.000000
mean     408.237154
std      168.537116
min      187.000000
25%      279.000000
50%      330.000000
75%      666.000000
max      711.000000
Name: TAX, dtype: float64
```

In [19]:
```
df["PTRATIO"].describe()
```

Out[19]:
```
count    506.000000
mean      18.455534
std        2.164946
min       12.600000
25%       17.400000
50%       19.050000
75%       20.200000
max       22.000000
Name: PTRATIO, dtype: float64
```

In [20]:
```
fig,axes = plt.subplots(2,2,figsize=(10,10))

sns.boxplot(df,y="CRIM",ax=axes[0,0])

sns.boxplot(df,y="TAX",ax=axes[0,1])

sns.boxplot(df,y="PTRATIO",ax=axes[1,0])
```

```
axes[1, 1].set_visible(False)
```



The CRIM (crime rate) feature shows a significant number of high-value outliers, indicating that some suburbs experience much higher crime rates than others.
In contrast, the PTRATIO (pupil-to-teacher ratio) feature has only a few mild outliers, suggesting relatively consistent values across most areas.

(f) How many of the suburbs in this data set bound the Charles river?

In [21]:
```
df["CHAS"].value_counts()
```

Out[21]:
```
CHAS
0    471
1     35
Name: count, dtype: int64
```

35 suburbs are set bound the Charles river.

(g) What is the median pupil-teacher ratio among the towns in this data set?

In [22]:
```
df["PTRATIO"].median()
```

Out[22]:
```
np.float64(19.05)
```

The median pupil-teacher ratio is **19.05**

(h) Which suburb of Boston has lowest median value of owner-occupied homes? What are the values of the other predictors
for that suburb, and how do those values compare to the overall ranges for those predictors? Comment on your findings.

In [23]:
```
df[df["MEDV"]== df["MEDV"].min()]
```

Out[23]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	I
398	38.3518	0.0	18.1	0	0.693	5.453	100.0	1.4896	24	666.0	20.2	396.9
405	67.9208	0.0	18.1	0	0.693	5.683	100.0	1.4254	24	666.0	20.2	384.9

In [24]:
```
df.describe()
```

Out[24]:

	CRIM	ZN	INDUS	NOX	RM	AGE	DIS
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.554695	6.284634	68.574901	3.795043
std	8.601545	23.322453	6.860353	0.115878	0.702617	28.148861	2.105710
min	0.006320	0.000000	0.460000	0.385000	3.561000	2.900000	1.129600
25%	0.082045	0.000000	5.190000	0.449000	5.885500	45.025000	2.100175
50%	0.256510	0.000000	9.690000	0.538000	6.208500	77.500000	3.207450
75%	3.677083	12.500000	18.100000	0.624000	6.623500	94.075000	5.188425
max	88.976200	100.000000	27.740000	0.871000	8.780000	100.000000	12.126500

we can see the suburb **405** has very high value of **CRIM**, which is more than **20 times** of the mean CRIM.

(i) In this data set, how many of the suburbs average more than seven rooms per dwelling? More than eight rooms per dwelling?
Comment on the suburbs that average more than eight rooms per dwelling.

In [25]:
```
more_than_7 = df[df["RM"] > 7]
len(more_than_7)
```

Out[25]: 64

64 suburbs has more than 7 rooms per dwelling.

In [26]:
```
more_than_8 = df[df["RM"]>8]
len(more_than_8)
```

Out[26]: 13

13 suburbs has more than 8 rooms per dwelling.

In [27]:
```
more_than_8.describe()
```

Out[27]:

	CRIM	ZN	INDUS	NOX	RM	AGE	DIS	
count	13.000000	13.000000	13.000000	13.000000	13.000000	13.000000	13.000000	13.00
mean	0.718795	13.615385	7.078462	0.539238	8.348538	71.538462	3.430192	325.07
std	0.901640	26.298094	5.392767	0.092352	0.251261	24.608723	1.883955	110.97
min	0.020090	0.000000	2.680000	0.416100	8.034000	8.400000	1.801000	224.00
25%	0.331470	0.000000	3.970000	0.504000	8.247000	70.400000	2.288500	264.00
50%	0.520140	0.000000	6.200000	0.507000	8.297000	78.300000	2.894400	307.00
75%	0.578340	20.000000	6.200000	0.605000	8.398000	86.500000	3.651900	307.00
max	3.474280	95.000000	19.580000	0.718000	8.780000	93.900000	8.906700	666.00

In [28]:
```
df.describe()
```

Out[28]:

	CRIM	ZN	INDUS	NOX	RM	AGE	DIS
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.554695	6.284634	68.574901	3.795043
std	8.601545	23.322453	6.860353	0.115878	0.702617	28.148861	2.105710
min	0.006320	0.000000	0.460000	0.385000	3.561000	2.900000	1.129600
25%	0.082045	0.000000	5.190000	0.449000	5.885500	45.025000	2.100175
50%	0.256510	0.000000	9.690000	0.538000	6.208500	77.500000	3.207450
75%	3.677083	12.500000	18.100000	0.624000	6.623500	94.075000	5.188425
max	88.976200	100.000000	27.740000	0.871000	8.780000	100.000000	12.126500

we can conclude that suburbs having more than 8 rooms per dwelling have very less crime rater (CRIM) values.

Also, these have less value of LSTAT, and theit target value, price is signifanctly larger than the overall mean price of the whole dataset