Introduction to Statistical Learning Chapter 2 Exercises

B-Loesch

1 For each of parts (a) through (d), indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answer.

- (a) The sample size n is extremely large, and the number of predictors p is small.

Better: With a large sample size, flexible methods can capture complex relationships without overfitting.

- (b) The number of predictors p is extremely large, and the number of observations n is small.

Worse, With a large number of predictors, flexible methods are prone to overfitting.

- (c) The relationship between the predictors and response is highly non-linear.

Better, Flexible methods are designed to capture complex relationships between predictors and the response variable.

- (d) The variance of the error terms, i.e. $\sigma^2 = \text{Var}(\epsilon)$, is extremely high.

Worse, High variance in the error implies that there is a lot of noise in the data. Flexible methods can be significantly affected by this noise.

2 Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide *n* and *p*.

- (a) We collect a set of data on the top 500 firms in the US. For each firm we record proft, number of employees, industry and the CEO salary. We are interested in understanding which factors afect CEO salary.

Regression - The response in this case is quantitative. ***Inference*** - We are interested in understanding how the predictors impact the response (salary). $n = 500$, $p =$ profit, number of employees, and industry

- (b) We are considering launching a new product and wish to know whether it will be a *success* or a *failure*. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

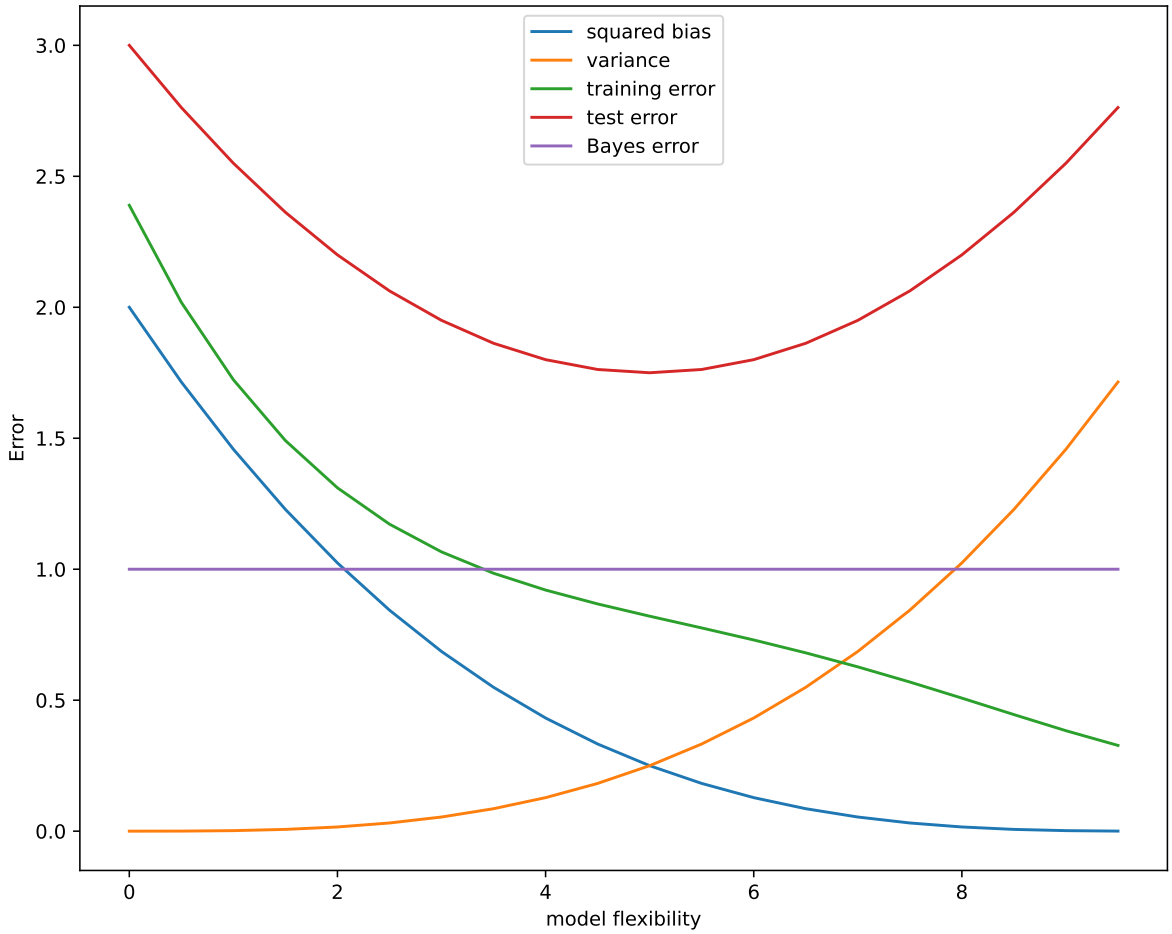
Classification - The response is a binary value. ***Prediction*** - We are only interested in the results and not how each predictor impacts the success. $n = 20$, and $p =$ price, budget, competition price, 10 others.

- (c) We are interested in predicting the % change in the USD/EURO exchange rate in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the USD/Euro, the % change in the US market, the % change in the British market, and the % change in the German market.

Regression - The response in this case is quantitative. ***Prediction*** - We only want to predict the response. $n = 52$, and $p =$ % change in US/British/German market.

3 We now revisit the bias-variance decomposition.

- (a) Provide a sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the values for each curve. There should be five curves. Make sure to label each one.



- (b) Explain why each of the five curves has the shape displayed in part (a)

Squared bias - Bias is reduced as flexbility increases.

Variance - More complex models are more sensitive to new/different data being introduced. The model may be fitted to random noise only introduced by the training data and thus not generalizing to test data.

Training error - As the model gets more complex, the tighter it can be fit to the training data.

Test error - The expected test error is the sum of Variance, Bias and Bayes (or random error). This shape represents the bias-variance trade-off.

Bayes error - This term is constant by definition because it doesn't depend on the flexibility of the model.

4 You will now think of some real-life application for statistical learning.

- (a) Describe three real-life applications in which classifcation might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

1. Determining whether a baseball team will win or lose a game. Response would be win or loss. Predictors could be starting pitchers' ERA, starting hitters' OPS+, home or away, and current standings. Depending on the analyst it could be inference or prediction, perhaps the front office would be interested in which predictors should be invested more heavily in while a better would only be interested in the results.
2. Determining whether a certain medicine will work on a patient. Response would be success/cured or not. Predictors could be symptoms, blood test results etc. This would be prediction.
3. Determining whether a written symbol represents a letter/number. Response would be the letter/number. Predictors would be an image. This would also be prediction.

- (b) Describe three real-life applications in which regression might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

1. Determining the traffic demand during any random time. Response would be the number of vehicles. Predictors would be past number of vehicles, day of the week, time of day, weather, etc. This would be prediction.
2. Determining the test scores of students on a standardized exam. Response would be the score. Predictors could be ZIP code, parents' education levels, gender, race/ethnicity etc. This would be inference.
3. Determining the amount of product sold. Response would be the amount of product sold. Predictors could be advertisment $ spent, comparison of similar products, price etc. This would be inference.

- (c) Describe three real-life applications in which cluster analysis might be useful.

1. Clustering national parks into groups based on number of guests throughout the year to assign park workers by season. Some parks may see higher traffic during Spring/Summer while other may see high traffic all year.
2. Clustering for recommendations to use on various streaming services. Given a list of movies/TV shows someone watches, a list of new content can be provided.
3. Clustering types of ships into different types based on standard characteristics like length and depth. This could be useful for quantifying fuel burn/emissions when the type is unknown.

5 What are the advantages and disadvantages of very flexible (versus a less flexible) approach for regression or classification? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred?

Very flexible models are capable of capturing complex patterns (i.e. nonlinear) in a dataset, which are generally present in real-life examples. But this makes them very prone to overfitting the random noise of a training data set which results in higher test MSE.

A more flexible approach might be preferred when we are interested more in prediction than inference. The opposite is true for a less flexible method, this is because simpler models are easier to interpret.

6 Describe the differences between a parametric and a non-parametric statistical learning approach. What are the advantages of a parametric approach to regression or classication (as opposed to a non-parametric approach)? What are its disadvantages?

Parametric: First, makes an assumption about the functional form of f (linear vs nonlinear). Second, finds a procedure that uses training data to fit or train the model. This reduces the problem of estimating f down to one of estimating a set of parameters rather than the overall form of the data. The inital assumption generally doesn't match the unknown form and thus will be more innacurate.

Non-Parametric: Does not make any assumptions about the form of f, and estimates f as close as possible to the actual data. This could potentially be much more accurate because it is more flexible. But this does require a large amount of observation to make an accurate estimate.

7 The table below provides a training data set containing six observations, three predictors, and one qualitative response variable. Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using *K*-nearest neighbors.

Obs.	X1	X2	X3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	−1	0	1	Green
6	1	1	1	Red

- (a) Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.

One can use $\sqrt{X_1^2 + X_2^2 + X_3^2} = distance$ to get:

Obs.	X1	X2	X3	Distance(0,0,0)	Y
1	0	3	0	3	Red
2	2	0	0	2	Red
3	0	1	3	3.2	Red
4	0	1	2	2.2	Green
5	−1	0	1	1.4	Green
6	1	1	1	1, 7	Red

- (b) What is our prediction with K = 1? Why?

Green, with K = 1 we are only using the closest neighbor to classify the point. The minimum distance is observation 5 so we choose green.

- (c) What is our prediction with K = 3? Why?

Red, with K = 3 we find the 3 closest neighbors in the data to classify the point. These are 2, 5, and 6 so a 2/3 probability of Red and 1/3 probability of Green. We choose the group with the highest probability - red.

- (d) If the Bayes decision boundary in this problem is highly non-linear, then would we expect the *best* value for K to be large or small? Why?

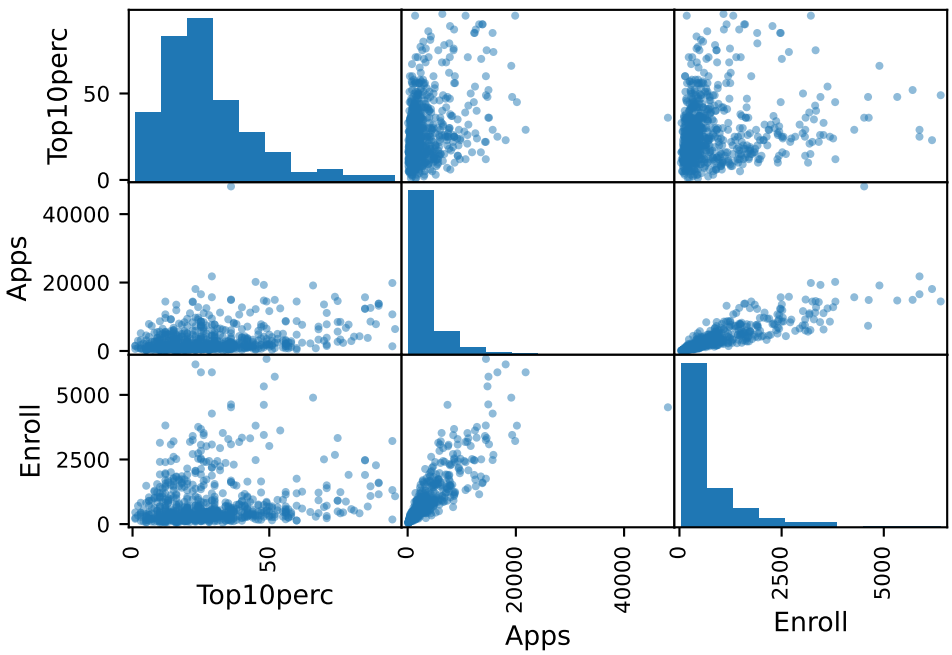
The best value for K in a highly non-linear scenario would be 1, which provides the highest flexibility. For classification problems (KNN), flexiblity is inversely proportional to K.

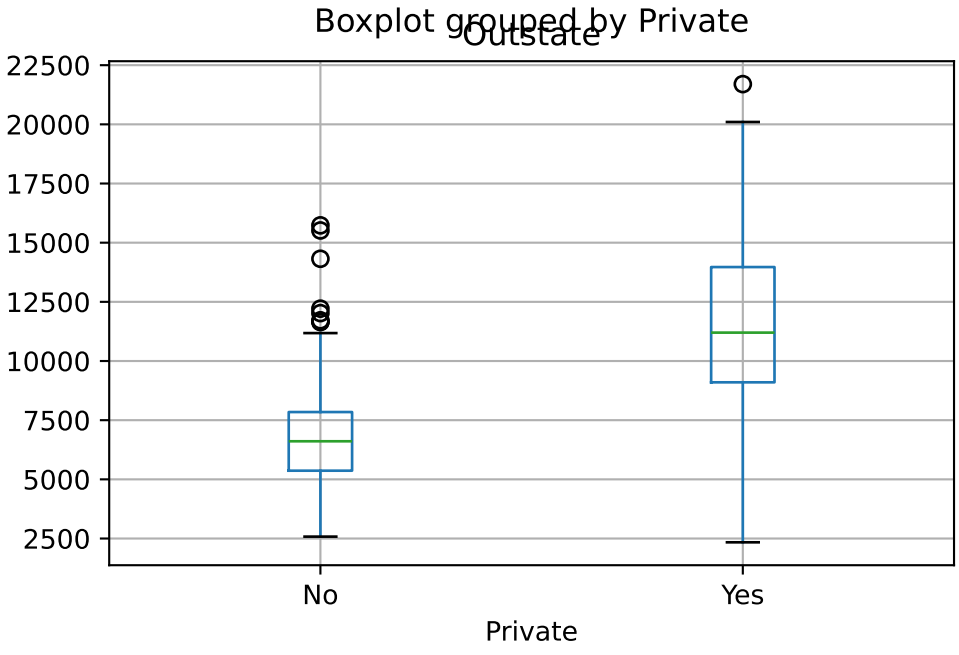
Applied problems will not have their associated text within this document, please check the ISLP book to see what each problem is asking.

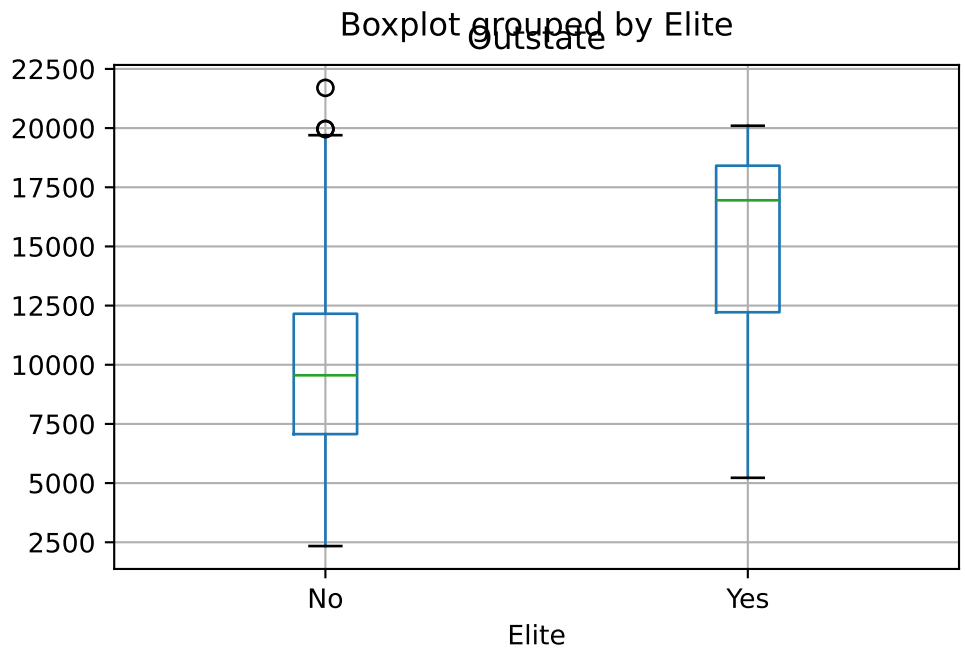
8

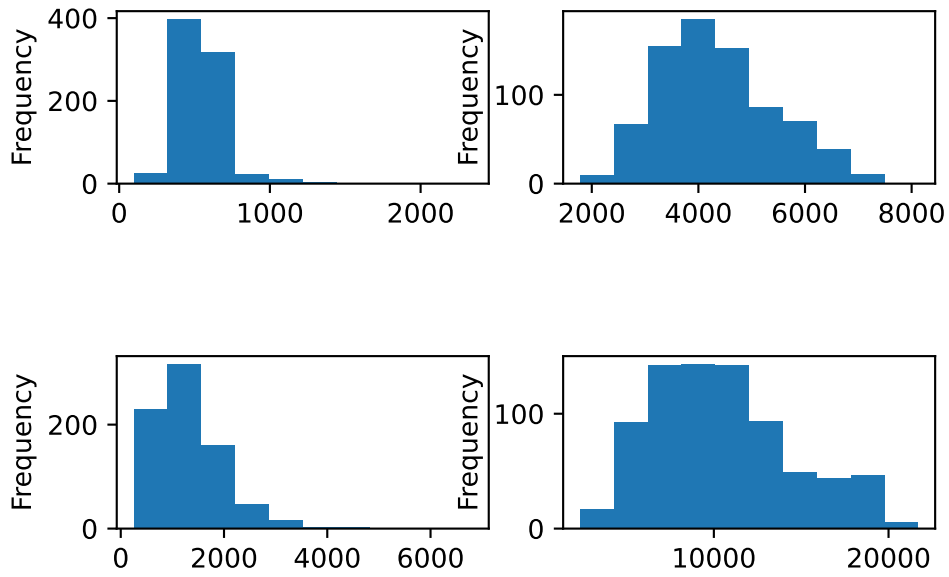
```
import pandas as pd
import matplotlib.pyplot as plt
college = pd.read_csv("D:/Brandon Loesch/ISLP-solutions/Data/College.csv")
college.head()

college2 = pd.read_csv("D:/Brandon Loesch/ISLP-solutions/Data/College.csv", index_col = 0)
college2.head()

college3 = college.rename({"Unnamed: 0": "College"}, axis = 1)
college3 = college3.set_index("College")

college = college3
college.describe()

pd.plotting.scatter_matrix(college[["Top10perc", "Apps", "Enroll"]])

college.boxplot("Outstate", by = "Private")

college["Elite"] = pd.cut(college["Top10perc"], bins = [0, 50, 100], labels = ["No", "Yes"])
college["Elite"].value_counts()
college.boxplot("Outstate", by = "Elite")

fig = plt.figure()

plt.subplot(221)
college["Books"].plot.hist()
plt.subplot(222)
college["Room.Board"].plot.hist()
plt.subplot(223)
college["Personal"].plot.hist()
plt.subplot(224)
college["Outstate"].plot.hist()
```

```
fig.subplots_adjust(hspace=1)
```






Boxplot grouped by Elite
Outstate
22500
20000
17500
15000
12500
10000
7500
5000
2500
No
Yes
Elite

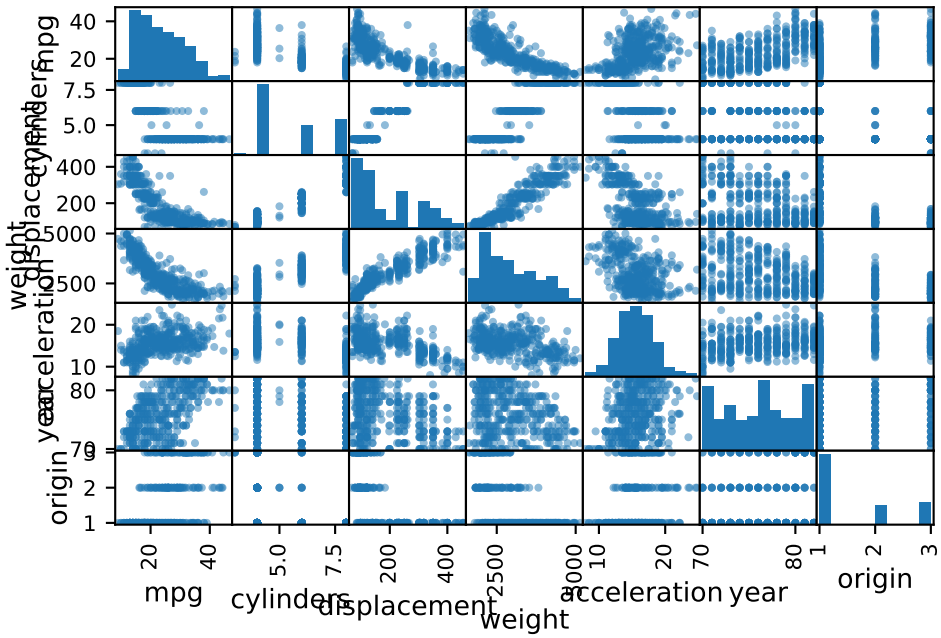

Frequency
400
200
0
0
1000
2000
Frequency
100
0
2000
4000
6000
8000
Frequency
200
0
0
2000
4000
6000
Frequency
100
0
10000
20000

9

```
import pandas as pd
import matplotlib.pyplot as plt

cars = pd.read_csv("D:/Brandon Loesch/ISLP-solutions/Data/Auto.csv")
cars.isnull().sum(axis = 0)

cars_quantitative = cars.select_dtypes(include = "number")
cars_quantitative.max() - cars_quantitative.min()
cars_quantitative.describe()

cars_trim = cars.drop(index = range(10,85))
cars_quantitative = cars_trim.select_dtypes(include = "number")
cars_quantitative.describe()

pd.plotting.scatter_matrix(cars)
```

```
array([[<Axes: xlabel='mpg', ylabel='mpg'>,
        <Axes: xlabel='cylinders', ylabel='mpg'>,
        <Axes: xlabel='displacement', ylabel='mpg'>,
        <Axes: xlabel='weight', ylabel='mpg'>,
        <Axes: xlabel='acceleration', ylabel='mpg'>,
        <Axes: xlabel='year', ylabel='mpg'>,
        <Axes: xlabel='origin', ylabel='mpg'>],
       [<Axes: xlabel='mpg', ylabel='cylinders'>,
        <Axes: xlabel='cylinders', ylabel='cylinders'>,
        <Axes: xlabel='displacement', ylabel='cylinders'>,
        <Axes: xlabel='weight', ylabel='cylinders'>,
        <Axes: xlabel='acceleration', ylabel='cylinders'>,
        <Axes: xlabel='year', ylabel='cylinders'>,
        <Axes: xlabel='origin', ylabel='cylinders'>],
       [<Axes: xlabel='mpg', ylabel='displacement'>,
        <Axes: xlabel='cylinders', ylabel='displacement'>,
        <Axes: xlabel='displacement', ylabel='displacement'>,
        <Axes: xlabel='weight', ylabel='displacement'>,
        <Axes: xlabel='acceleration', ylabel='displacement'>,
        <Axes: xlabel='year', ylabel='displacement'>,
        <Axes: xlabel='origin', ylabel='displacement'>],
       [<Axes: xlabel='mpg', ylabel='weight'>,
        <Axes: xlabel='cylinders', ylabel='weight'>,
        <Axes: xlabel='displacement', ylabel='weight'>,
        <Axes: xlabel='weight', ylabel='weight'>,
```

```
 <Axes: xlabel='acceleration', ylabel='weight'>,
 <Axes: xlabel='year', ylabel='weight'>,
 <Axes: xlabel='origin', ylabel='weight'>],
[<Axes: xlabel='mpg', ylabel='acceleration'>,
 <Axes: xlabel='cylinders', ylabel='acceleration'>,
 <Axes: xlabel='displacement', ylabel='acceleration'>,
 <Axes: xlabel='weight', ylabel='acceleration'>,
 <Axes: xlabel='acceleration', ylabel='acceleration'>,
 <Axes: xlabel='year', ylabel='acceleration'>,
 <Axes: xlabel='origin', ylabel='acceleration'>],
[<Axes: xlabel='mpg', ylabel='year'>,
 <Axes: xlabel='cylinders', ylabel='year'>,
 <Axes: xlabel='displacement', ylabel='year'>,
 <Axes: xlabel='weight', ylabel='year'>,
 <Axes: xlabel='acceleration', ylabel='year'>,
 <Axes: xlabel='year', ylabel='year'>,
 <Axes: xlabel='origin', ylabel='year'>],
[<Axes: xlabel='mpg', ylabel='origin'>,
 <Axes: xlabel='cylinders', ylabel='origin'>,
 <Axes: xlabel='displacement', ylabel='origin'>,
 <Axes: xlabel='weight', ylabel='origin'>,
 <Axes: xlabel='acceleration', ylabel='origin'>,
 <Axes: xlabel='year', ylabel='origin'>,
 <Axes: xlabel='origin', ylabel='origin'>]], dtype=object)
```



Displacement and weight visually seem reasonable to use as predictors for MPG.

10

```
import pandas as pd
import matplotlib.pyplot as plt

Boston = pd.read_csv("D:/Brandon Loesch/ISLP-solutions/Data/Boston.csv", index_col = 0)

print(f"The shape of the Boston data set is: {Boston.shape}\n")

pd.plotting.scatter_matrix(Boston, figsize=(10,10))
print("Correlation between all columns and crim, sorted:\n")
print(Boston.corrwith(Boston["crim"]).sort_values())

Boston.nlargest(10, "crim")
print("\n")
Boston.nlargest(10, "tax")
print("\n")
Boston.nlargest(10, "ptratio")
```

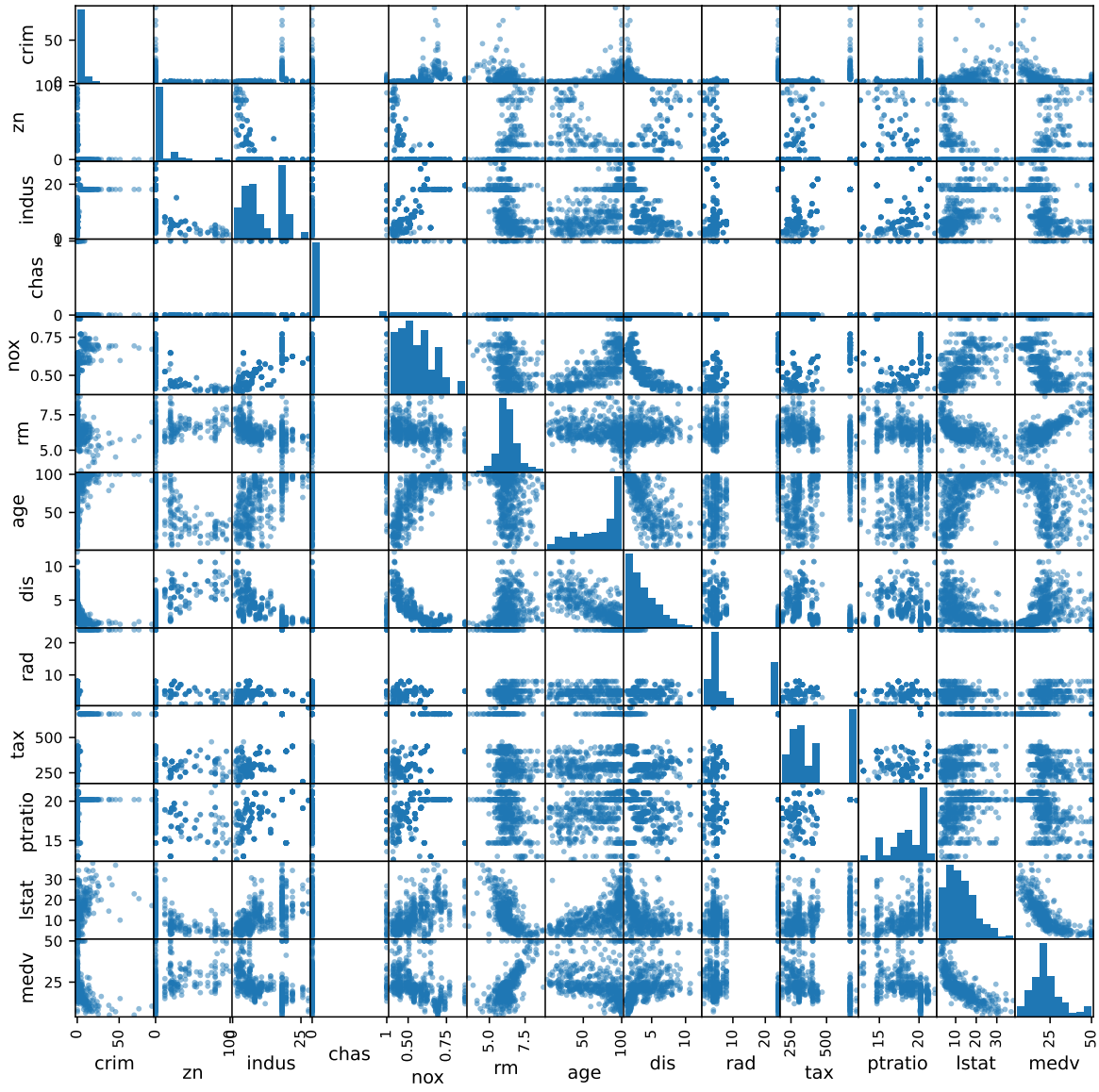
```
print("Range across all columns:\n")
print(Boston.max() - Boston.min())

print("Number of suburbs bounding Charles River:\n")
print(Boston["chas"].sum())

CharlesRiver = Boston[Boston["chas"] == 1]
print("Median ptration for suburbs bounding Charles River:\n")
print(CharlesRiver["ptratio"].median())

print("Lowest median value of owner-occupied homes and how that compares to all other suburbs
print(Boston.nsmallest(1, "age"))

print(Boston.max() - Boston.nsmallest(1, "age"))

print("Number of houses with 7 and 8 rooms (8 is described below:)")
print(len(Boston[Boston["rm"] > 7]))
print(len(Boston[Boston["rm"] > 8]))

Boston[Boston["rm"] > 8].describe()
```

```
The shape of the Boston data set is: (506, 13)

Correlation between all columns and crim, sorted:

medv      -0.388305
dis       -0.379670
rm        -0.219247
zn        -0.200469
chas      -0.055892
ptratio    0.289946
age        0.352734
indus      0.406583
nox        0.420972
lstat      0.455621
tax        0.582764
rad        0.625505
crim       1.000000
dtype: float64
```

```
Range across all columns:

crim        88.96988
zn         100.00000
indus       27.28000
chas         1.00000
nox          0.48600
rm           5.21900
age         97.10000
dis         10.99690
rad         23.00000
tax        524.00000
ptratio      9.40000
lstat       36.24000
medv        45.00000
dtype: float64
Number of suburbs bounding Charles River:

35
Median ptration for suburbs bounding Charles River:

17.6
Lowest median value of owner-occupied homes and how that compares to all other suburbs:

       crim   zn  indus  chas    nox    rm  age     dis  rad  tax  ptratio  \
42  0.12744  0.0   6.91     0  0.448  6.77  2.9  5.7209    3  233     17.9

    lstat  medv
42   4.84  26.6
        crim     zn  indus  chas    nox    rm   age     dis   rad    tax  \
42  88.84876  100.0  20.83   1.0  0.423  2.01  97.1  6.4056  21.0  478.0

    ptratio  lstat  medv
42      4.1  33.13  23.4
Number of houses with 7 and 8 rooms (8 is described below:)
64
13
```

	crim	zn	indus	chas	nox	rm	age	dis	rad
count	13.000000	13.000000	13.000000	13.000000	13.000000	13.000000	13.000000	13.000000	13.00

	crim	zn	indus	chas	nox	rm	age	dis	rad
mean	0.718795	13.615385	7.078462	0.153846	0.539238	8.348538	71.538462	3.430192	7.461
std	0.901640	26.298094	5.392767	0.375534	0.092352	0.251261	24.608723	1.883955	5.332
min	0.020090	0.000000	2.680000	0.000000	0.416100	8.034000	8.400000	1.801000	2.000
25%	0.331470	0.000000	3.970000	0.000000	0.504000	8.247000	70.400000	2.288500	5.000
50%	0.520140	0.000000	6.200000	0.000000	0.507000	8.297000	78.300000	2.894400	7.000
75%	0.578340	20.000000	6.200000	0.000000	0.605000	8.398000	86.500000	3.651900	8.000
max	3.474280	95.000000	19.580000	1.000000	0.718000	8.780000	93.900000	8.906700	24.00



Houses with more than 8 rooms generally have a lower crime rate, lower indus, lower LSTAT, and almost double the median value.