Homework 1

Due April 12, 2019 by 11:59pm

Instructions: **All code exercises must be completed using Python.** Upload your answers to the questions below to Canvas. Submit the answers to the questions (including the relevant output of the code) in a PDF file and your code in a (single) separate file. Be sure to comment your code to indicate which lines of your code correspond to which question part.

1. Read Chapter 2 in *Elements of Statistical Learning*.

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) You want to predict whether a particular customer is going to click on an online advertisement or not. You have information on whether or not they clicked on 200 other ads, in addition to whether the ad was in the same category, whether the ad was shown during regular working hours, whether the ad was shown on a weekend, and the percent of all customers who had previously clicked on the ad.
   Prediction, classification. $n = 200, p = 4$ (p is the number of predictors).

   (b) Suppose it is the end of the quarter and you wish to predict your score on the final exam. You have data from 20 classes you have previously taken, consisting of your final exam scores, your average scores on the midterms (i.e., one average midterm score per class), your average homework scores (i.e., one average homework score per class), and whether the final exam was take-home or not.
   Prediction, regression. $n = 20, p = 3$.

   (c) You work for an ice cream shop and are in charge of determining what factors affect how much ice cream is sold each day. For 300 days you have information on how much ice cream the shop sold, in addition to whether the day was sunny or not, what the temperature was, whether school is in session or not, whether your most popular flavor was available that day, and whether you had recently run any advertisements.
   Inference, regression. $n = 300, p = 5$.

3. In this problem you will brainstorm real-life applications for statistical learning. Your answers aren't allowed to be the same as any of the examples in the other homework problems.

(a) Describe three real-life applications in which classification might be useful, **one from political science, one from sports, and one from an area of your choice**. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

- Political science: Predict whether a bill in the senate will pass or not (response). Predictors: party/parties of people who developed the bill, fraction of Republicans in the Senate, whether the bill is highly controversial, whether similar bills have passed before.
- Sports: Learn what factors affect whether the Sounders win/lose/draw (response). Predictors: Weather, location, record of opponent, what players played. Goal: Inference.
- Misc: Predict whether a YouTube video has offensive content (response). Predictors: objects detected in video, words in comments, words in description, characteristics of user who uploaded the video. https://www.nytimes.com/2017/03/23/business/media/youtube-advertisers-offensi html?_r=0

(b) Describe three real-life applications in which regression might be useful, **one from agriculture, one from business, and one from an area of your choice**. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

- Engineering: Estimate how long a stretch of I-5 will last before it needs to be repaved (response) given previous interval lengths between pavings, average number of vehicles per day that traveled on it during those intervals (predictors). Goal: Prediction.
- Business: Determine what factors affect customers' monthly mobile data usage (response). Predictors: Age, socioeconomic status, location, time of year, carrier, plan type. Goal: Inference.
- Misc: Predict how much someone will spend at an Amazon Go grocery store (response) based on the following data collected the previous times they shopped: how much they spent, the time of day, day of week, month of year, the temperature, and whether they shopped with anyone else. Goal: Prediction.

(c) Describe three real-life applications in which cluster analysis might be useful, **one from education, one from meteorology, and one from an area of your choice**. Be sure to describe why it would be useful.

- Education: Clustering students based on results on a math placement exam. Purpose: divide students into groups so you can teach different material to the different groups based on students' abilities.
- Meteorology: Clustering points where you have observed precipitation above some threshold in order to determine where the different storms are

- Misc: Clustering people's OkCupid responses in order to find your true love: https://www.wired.com/2014/01/how-to-hack-okcupid/

4. 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

Obs.	X_1	X_2	X_3	Y
1	0	4	0	Green
2	2	0	1	Red
3	0	1	3	Red
4	-1	1	2	Green
5	-3	0	1	Green
6	2	0	1	Red

to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

(a) Compute the Euclidean distance between each observation and the test point $X_1 = X_2 = X_3 = 0$.
$4, \sqrt{5}, \sqrt{10}, \sqrt{6}, \sqrt{10}, \sqrt{5}$

(b) What is our prediction with $K = 1$? Why?
Red, since the second and last points are closest and are both red (We would randomly choose one if they were different colors).

(c) What is our prediction with $K = 3$? Why?
The closest 3 are obs 2, 4, 6. They are red, green, and red, so prediction is red as the majority are red.

(d) If the Bayes decision boundary in this problem is linear but the data is noisy, then would we expect the best value for K to be larger or smaller? Why?
Larger- if we have a small K there may be a good chance that we'll pick one of the noisy points from the other class and hence that we'll be capturing the noise. In addition, the larger the K, the more linear the decision boundary will be.

5. This exercise relates to the `College` data set, which can be found here http://www-bcf.usc.edu/~gareth/ISL/College.csv. 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 `pandas.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 if the file is saved on your computer.

(b) Look at the data using the `head` attribute. You should notice that the first column is just the name of each university. We don't really want Python to treat this as data. However, it may be handy to have these names for later. Try the following commands:

```
college.rename(columns={'Unnamed: 0': 'School'}, inplace=True)
college.set_index('School')
```

(The line before 0 denotes a space.) You should see that there is now a `School` column with the name of each university recorded. This means that Python has given each row a name corresponding to the appropriate university. Python will not try to perform calculations on the row names.

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

ii. Use the `scatter_matrix()` function from the package `pandas.plotting` to produce a scatterplot matrix of the second through fourth columns of the data.

iii. Use the `boxplot` attribute to produce side-by-side boxplots of `Room.Board` versus `Private`. Hint: Use the `column` option to select the continuous variable and the `by` option to select the categorical variable.

iv. Create a new qualitative variable, called `Elite`, by binning the `Top10perc` variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50%.

```
Elite = np.array([False]*len(college))
Elite[college['Top10perc'] > 50] = True
college['Elite'] = pd.Series(Elite, index=college.index)
```

Use the `describe` attribute to see how many elite universities there are. For this you might find the option `include=['bool']` useful. Now use the `boxplot` attribute to produce side-by-side boxplots of `Room.Board` versus `Elite`.

v. Use the `hist` attribute to produce some histograms with differing numbers of bins for a few of the quantitative variables. You may find the `layout` option useful: it will divide the figure into regions so that plots can be made simultaneously.

vi. Continue exploring the data, and provide a brief summary of what you discover.

```python
# Problem 4
import numpy as np
import pandas as pd

# Part (a)
college = pd.read_csv('http://www-bcf.usc.edu/~gareth/ISL/College.csv')
```

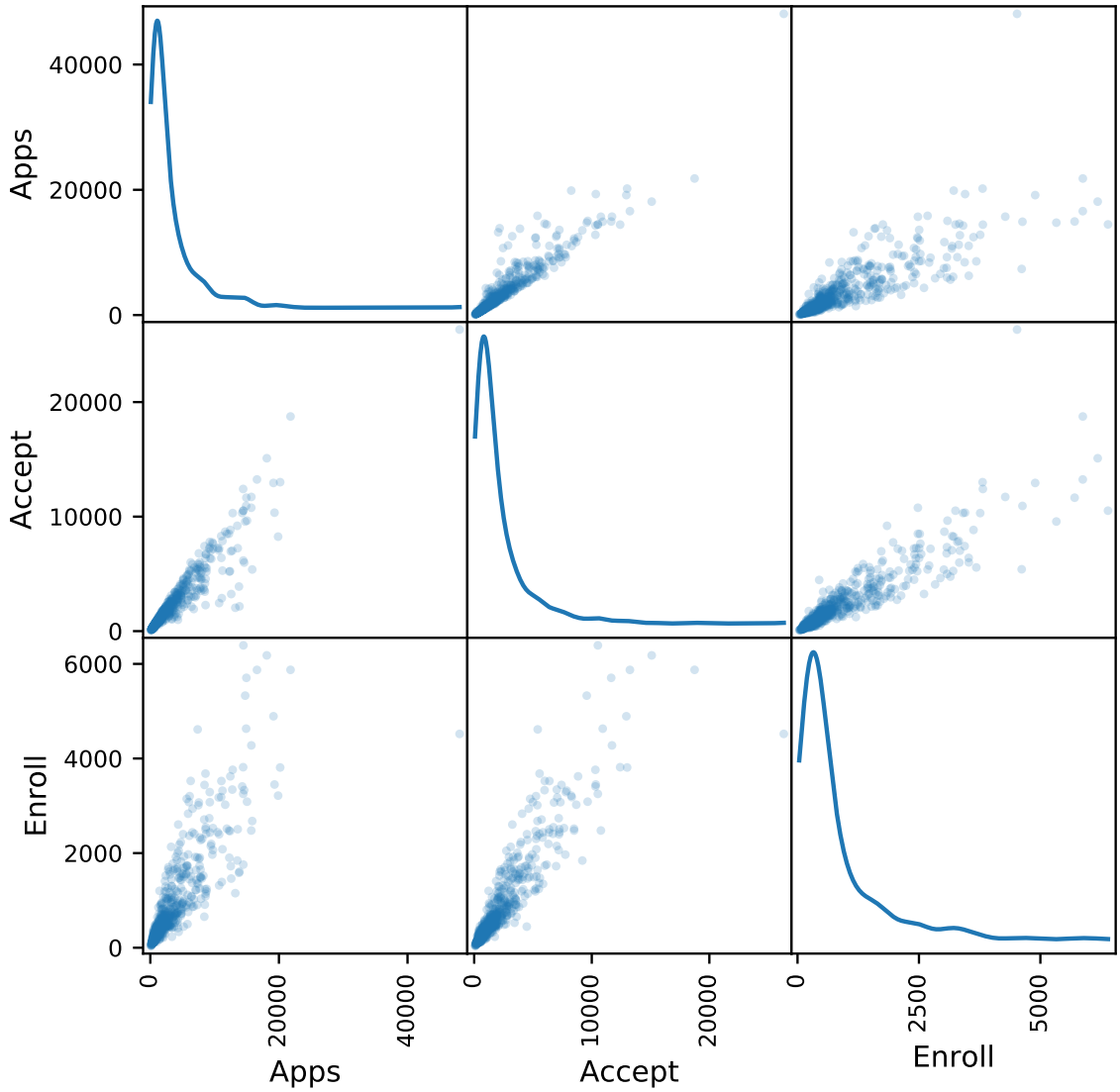
```python
# Part (b)
print(college.head())
college.rename(columns={'Unnamed: 0': 'School'}, inplace=True)
college.set_index('School');
```

```python
# Part (c)
# Part i
print(college.describe())
```

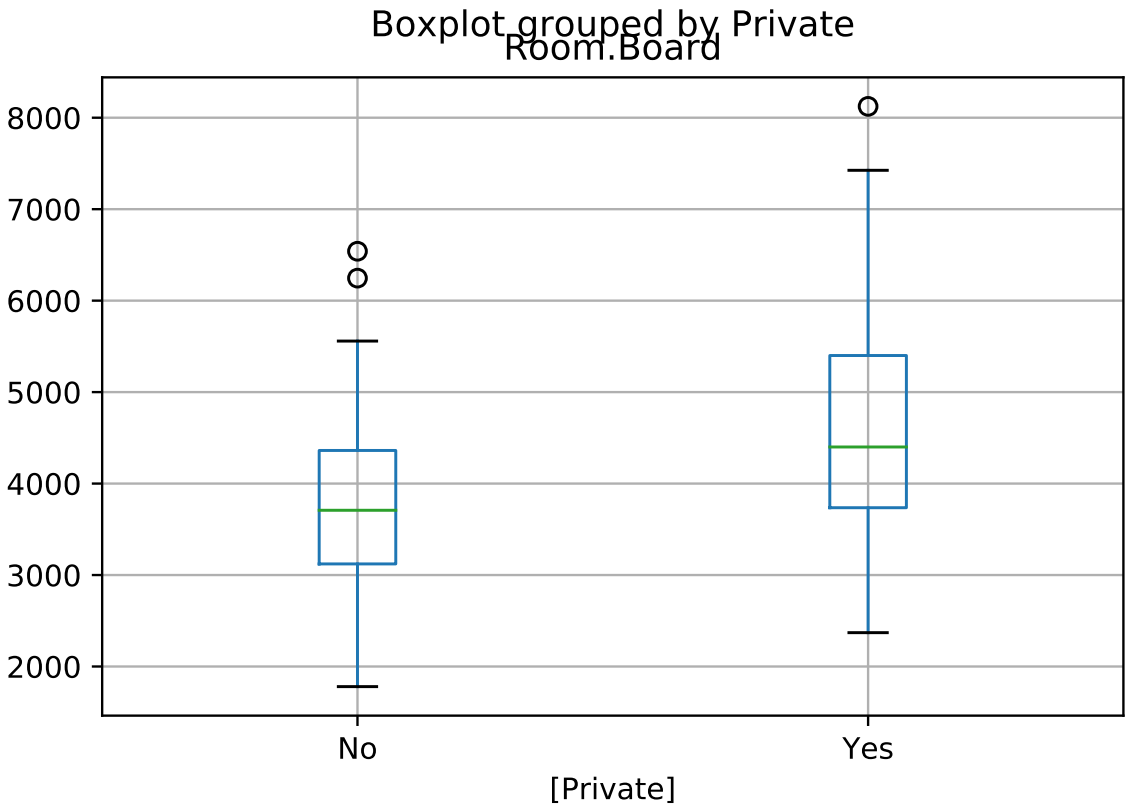
```
                Apps        Accept       Enroll   Top10perc   Top25perc
 count     777.000000    777.000000   777.000000  777.000000  777.000000
mean     3001.638353   2018.804376   779.972973   27.558559   55.796654
std      3870.201484   2451.113971   929.176190   17.640364   19.804778
min        81.000000     72.000000    35.000000    1.000000    9.000000
25%       776.000000    604.000000   242.000000   15.000000   41.000000
50%      1558.000000   1110.000000   434.000000   23.000000   54.000000
75%      3624.000000   2424.000000   902.000000   35.000000   69.000000
max     48094.000000  26330.000000  6392.000000   96.000000  100.000000

         F.Undergrad   P.Undergrad      Outstate   Room.Board
Books    count     777.000000    777.000000    777.000000   777.000000
777.000000
mean     3699.907336    855.298584  10440.669241  4357.526384
549.380952
```

```
std     4850.420531   1522.431887   4023.016484  1096.696416
165.105360
min      139.000000      1.000000   2340.000000  1780.000000
96.000000
25%      992.000000     95.000000   7320.000000  3597.000000
470.000000
50%     1707.000000    353.000000   9990.000000  4200.000000
500.000000
75%     4005.000000    967.000000  12925.000000  5050.000000
600.000000
max    31643.000000  21836.000000  21700.000000  8124.000000
2340.000000

          Personal         PhD    Terminal   S.F.Ratio  perc.alumni   count
mean   1340.642214   72.660232   79.702703   14.089704    22.743887
std     677.071454   16.328155   14.722359    3.958349    12.391801
min     250.000000    8.000000   24.000000    2.500000     0.000000
25%     850.000000   62.000000   71.000000   11.500000    13.000000
50%    1200.000000   75.000000   82.000000   13.600000    21.000000
75%    1700.000000   85.000000   92.000000   16.500000    31.000000
max    6800.000000  103.000000  100.000000   39.800000    64.000000

             Expend  Grad.Rate
count     777.000000  777.00000
mean     9660.171171   65.46332
std      5221.768440   17.17771
min      3186.000000   10.00000
25%      6751.000000   53.00000
50%      8377.000000   65.00000
75%     10830.000000   78.00000
max     56233.000000  118.00000
```

```
# Part ii
import matplotlib.pyplot as plt
%matplotlib inline
from pandas.plotting import scatter_matrix
scatter_matrix(college[college.columns[2:5]], alpha=0.2,
 figsize=(6, 6), diagonal='kde');
plt.show()
```

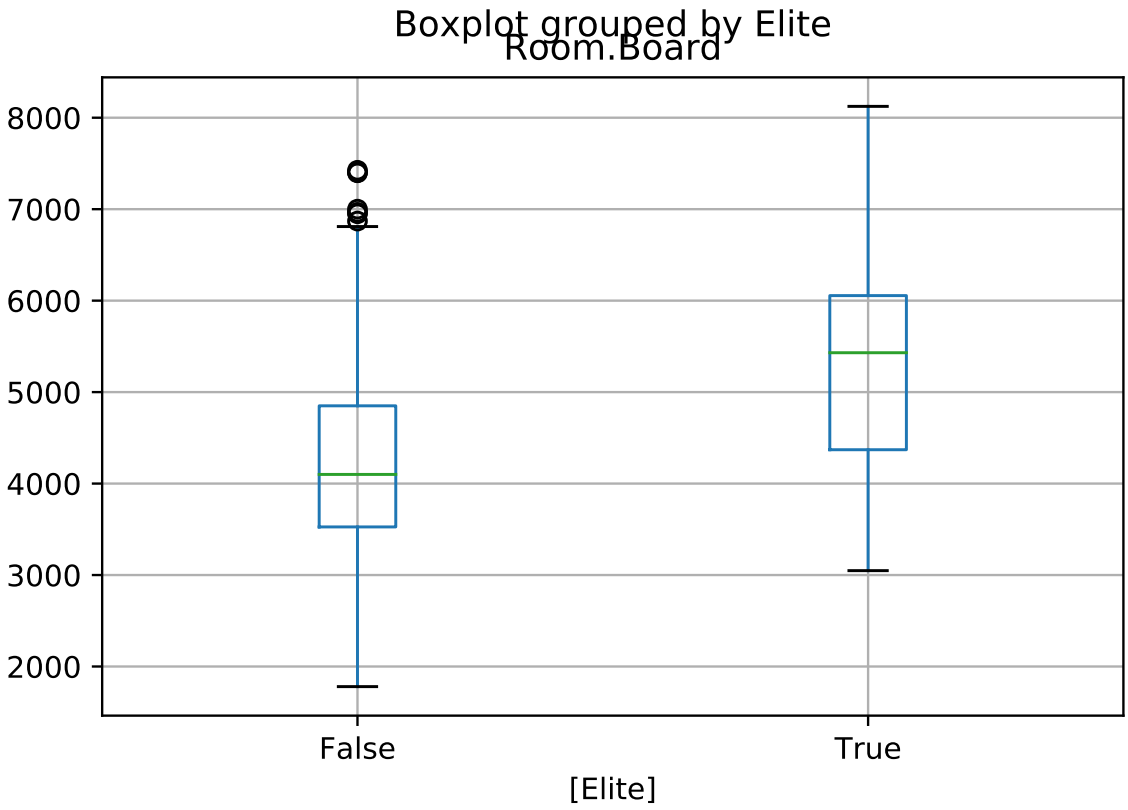



```
# Part iii
bp = college.boxplot(column='Room.Board', by=['Private'])
```

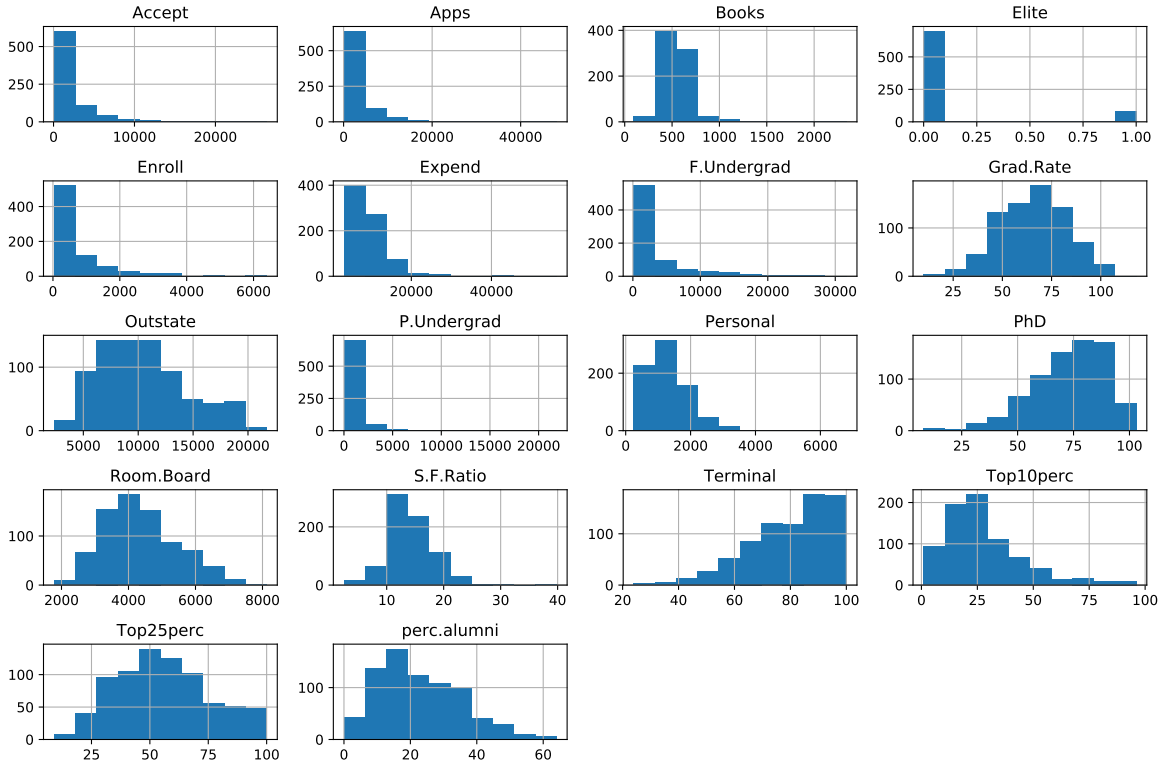



```
# Part iv
Elite = np.array([False]*len(college))
Elite[college['Top10perc'] > 50] = True
college['Elite'] = pd.Series(Elite, index=college.index)
print(college.describe(include=['bool']))
# There are 78 elite universities.
bp = college.boxplot(column='Room.Board', by=['Elite'])
```

```
        Elite
count     777
unique      2
top     False
freq      699
```




```
# Part v
plt.rcParams["figure.figsize"] = [12, 8]
college.hist(layout=[5, 4])
plt.tight_layout()
```



6. This exercise involves the `Auto` data set found here: http://www-bcf.usc.edu/~gareth/ISL/Auto.data. Make sure that the missing values have been removed from the data.

   (a) Which of the predictors are quantitative, and which are qualitative?

   (b) What is the range of each quantitative predictor?

   (c) What is the mean and standard deviation of each quantitative predictor?

   (d) Now remove the last 35 observations. What is the range, mean, and standard deviation of each quantitative predictor in the subset of the data that remains?

   (e) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

   (f) Suppose that we wish to predict gas mileage (`mpg`) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting `mpg`? Justify your answer.

```
# Problem 5
auto = pd.read_csv('http://www-bcf.usc.edu/~gareth/ISL/Auto.data',
 delim_whitespace=True)
auto = auto.dropna()
```

```
# Part (a)
print(auto.head())   # Origin and name are qualitative
```

```
    mpg  cylinders  displacement horsepower  weight  acceleration
year   0  18.0          8         307.0      130.0  3504.0          12.0
70
1  15.0          8         350.0      165.0  3693.0          11.5
70
2  18.0          8         318.0      150.0  3436.0          11.0
70
3  16.0          8         304.0      150.0  3433.0          12.0
70
4  17.0          8         302.0      140.0  3449.0          10.5
70

   origin                       name
0       1  chevrolet chevelle malibu
1       1          buick skylark 320
2       1         plymouth satellite
3       1              amc rebel sst
4       1                ford torino
```

```
# Parts (b) and (c)
print(auto.describe())
```

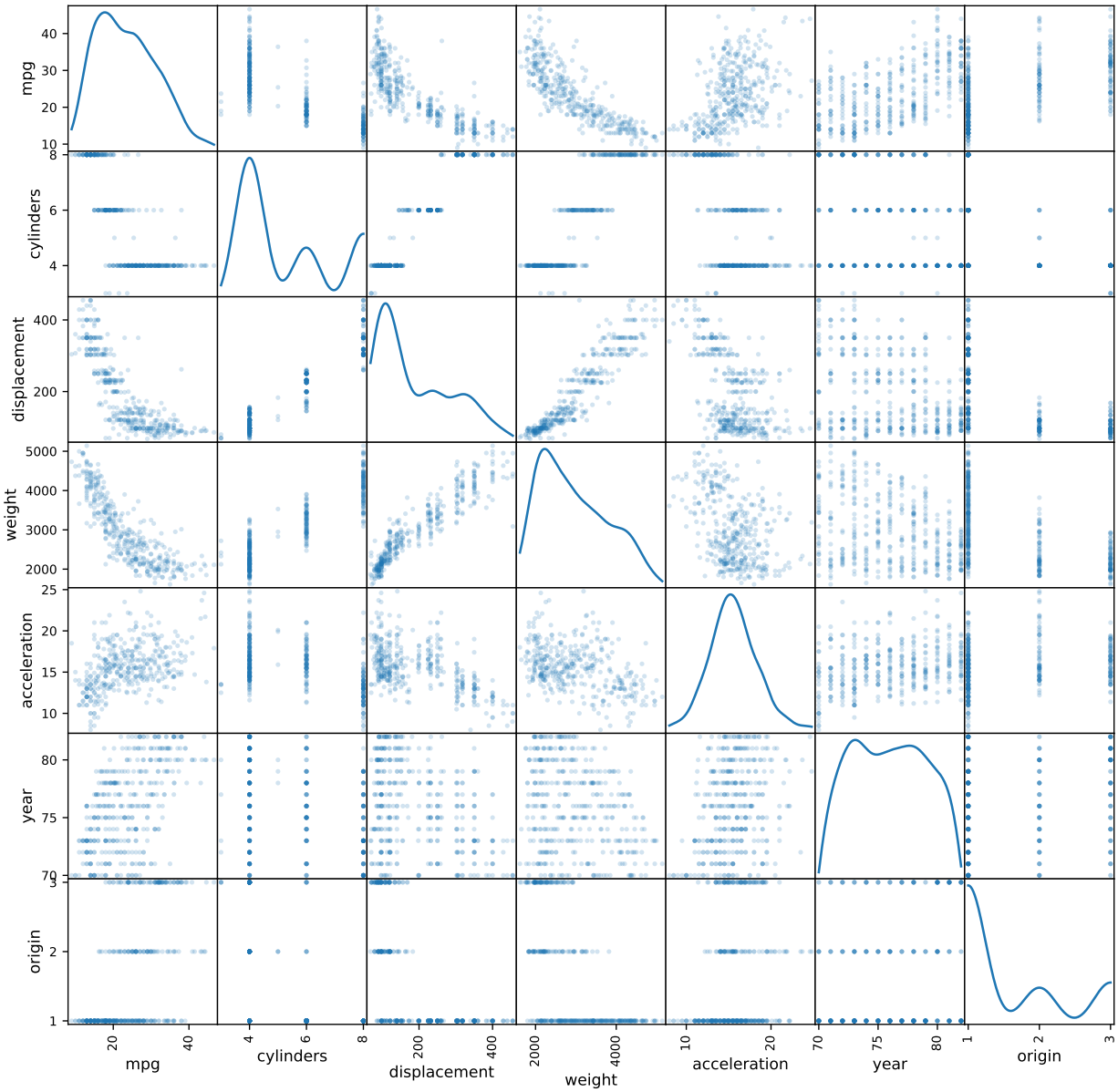
```
              mpg   cylinders  displacement       weight  acceleration
 count  397.000000  397.000000    397.000000   397.000000    397.000000
mean    23.515869    5.458438    193.532746  2970.261965     15.555668
std      7.825804    1.701577    104.379583   847.904119      2.749995
min      9.000000    3.000000     68.000000  1613.000000      8.000000
25%     17.500000    4.000000    104.000000  2223.000000     13.800000
50%     23.000000    4.000000    146.000000  2800.000000     15.500000
75%     29.000000    8.000000    262.000000  3609.000000     17.100000
max     46.600000    8.000000    455.000000  5140.000000     24.800000

             year      origin
count  397.000000  397.000000
mean    75.994962    1.574307
std      3.690005    0.802549
min     70.000000    1.000000
25%     73.000000    1.000000
50%     76.000000    1.000000
75%     79.000000    2.000000
max     82.000000    3.000000
```

```
# Part (d)
auto2 = auto[:-35]
auto2.describe()
```

```
              mpg   cylinders  displacement       weight  acceleration
 count  362.000000  362.000000    362.000000   362.000000    362.000000
```

```
mean    22.830939    5.549724    198.443370  3009.872928     15.464088
std      7.640078    1.727731    106.600558   866.541817      2.773077
min      9.000000    3.000000     68.000000  1613.000000      8.000000
25%     16.600000    4.000000    100.250000  2228.500000     13.500000
50%     21.550000    5.000000    153.000000  2872.500000     15.400000
75%     28.000000    8.000000    302.000000  3670.000000     17.000000
max     46.600000    8.000000    455.000000  5140.000000     24.800000

             year      origin
count  362.000000  362.000000
mean    75.428177    1.569061
std      3.357149    0.792579
min     70.000000    1.000000
25%     73.000000    1.000000
50%     75.000000    1.000000
75%     78.000000    2.000000
max     81.000000    3.000000
```

```
# Part (e)
scatter_matrix(auto, alpha=0.2, figsize=(12, 12), diagonal='kde');
plt.show()
```




```
# Part (f)
# cylinders, displacement, horsepower, weight, year
```