Assignment II

09/09/2019

Instructor: Assefaw Gebremedhin

Student: Yu-Chieh Wang 11641327

1. College

(A) Read 'College.csv' in Python

```
import csv
count = 0

with open('College.csv', newline='') as csvfile:
    college = csv.reader(csvfile, quotechar='|')
    print("**start read college.csv**")
    for row in college:
        if count > 0:
            print(', '. join(row))
        count += 1
```

- Use a variable 'count' with an initial value of 0 to avoid storing the attributes of the first line.

- The output is as follow:

```
/Users/angel/PycharmProjects/untitled3/venv/bin/python "/Users/angel/PycharmProjects/untitled3/v
**start read college.csv**
Abilene Christian University, Yes, 1660, 1232, 721, 23, 52, 2885, 537, 7440, 3300, 450, 2200, 70
Adelphi University, Yes, 2186, 1924, 512, 16, 29, 2683, 1227, 12280, 6450, 750, 1500, 29, 30, 12
Adrian College, Yes, 1428, 1097, 336, 22, 50, 1036, 99, 11250, 3750, 400, 1165, 53, 66, 12.9, 30
Agnes Scott College, Yes, 417, 349, 137, 60, 89, 510, 63, 12960, 5450, 450, 875, 92, 97, 7.7, 37
Alaska Pacific University, Yes, 193, 146, 55, 16, 44, 249, 869, 7560, 4120, 800, 1500, 76, 72, 1
Albertson College, Yes, 587, 479, 158, 38, 62, 678, 41, 13500, 3335, 500, 675, 67, 73, 9.4, 11,
Albertus Magnus College, Yes, 353, 340, 103, 17, 45, 416, 230, 13290, 5720, 500, 1500, 90, 93, 1
Albion College, Yes, 1899, 1720, 489, 37, 68, 1594, 32, 13868, 4826, 450, 850, 89, 100, 13.7, 37
Albright College, Yes, 1038, 839, 227, 30, 63, 973, 306, 15595, 4400, 300, 500, 79, 84, 11.3, 23
Alderson-Broaddus College, Yes, 582, 498, 172, 21, 44, 799, 78, 10468, 3380, 660, 1800, 40, 41,
Alfred University, Yes, 1732, 1425, 472, 37, 75, 1830, 110, 16548, 5406, 500, 600, 82, 88, 11.3,
Allegheny College, Yes, 2652, 1900, 484, 44, 77, 1707, 44, 17080, 4440, 400, 600, 73, 91, 9.9, 4
Allentown Coll. of St. Francis de Sales, Yes, 1179, 780, 290, 38, 64, 1130, 638, 9690, 4785, 600
Alma College, Yes, 1267, 1080, 385, 44, 73, 1306, 28, 12572, 4552, 400, 400, 79, 87, 15.3, 32, 9
Alverno College, Yes, 494, 313, 157, 23, 46, 1317, 1235, 8352, 3640, 650, 2449, 36, 69, 11.1, 26
American International College, Yes, 1420, 1093, 220, 9, 22, 1018, 287, 8700, 4780, 450, 1400, 7
Amherst College, Yes, 4302, 992, 418, 83, 96, 1593, 5, 19760, 5300, 660, 1598, 93, 98, 8.4, 63,
Anderson University, Yes, 1216, 908, 423, 19, 40, 1819, 281, 10100, 3520, 550, 1100, 48, 61, 12.
Andrews University, Yes, 1130, 704, 322, 14, 23, 1586, 326, 9996, 3090, 900, 1320, 62, 66, 11.5,
Angelo State University, No, 3540, 2001, 1016, 24, 54, 4190, 1512, 5130, 3592, 500, 2000, 60, 62
Antioch University, Yes, 713, 661, 252, 25, 44, 712, 23, 15476, 3336, 400, 1100, 69, 82, 11.3, 3
```

(B) Print the median cost of books for all schools

```
import csv
import matplotlib.pyplot as plt

listMedian = []
count = 0

with open('College.csv', newline='') as csvfile:
    college = csv.reader(csvfile, quotechar='|')
    for row in college:
        if count > 0:
            listMedian.append(int(row[11]))
        count += 1

listMedian.sort()
print("The median cost of books for all schools in "
      "this dataset is: ", listMedian[int(count/2)])
```

```
question B
/Users/angel/PycharmProjects/untitled3/venv/bin/python "/Users/angel
The median cost of books for all schools in this dataset is:  500
```

- Use a list which is called listMedian to store every value of column books, and then sort the list from small to large.

- Next, getting the median of this list, and then print it out. The answer is 500.

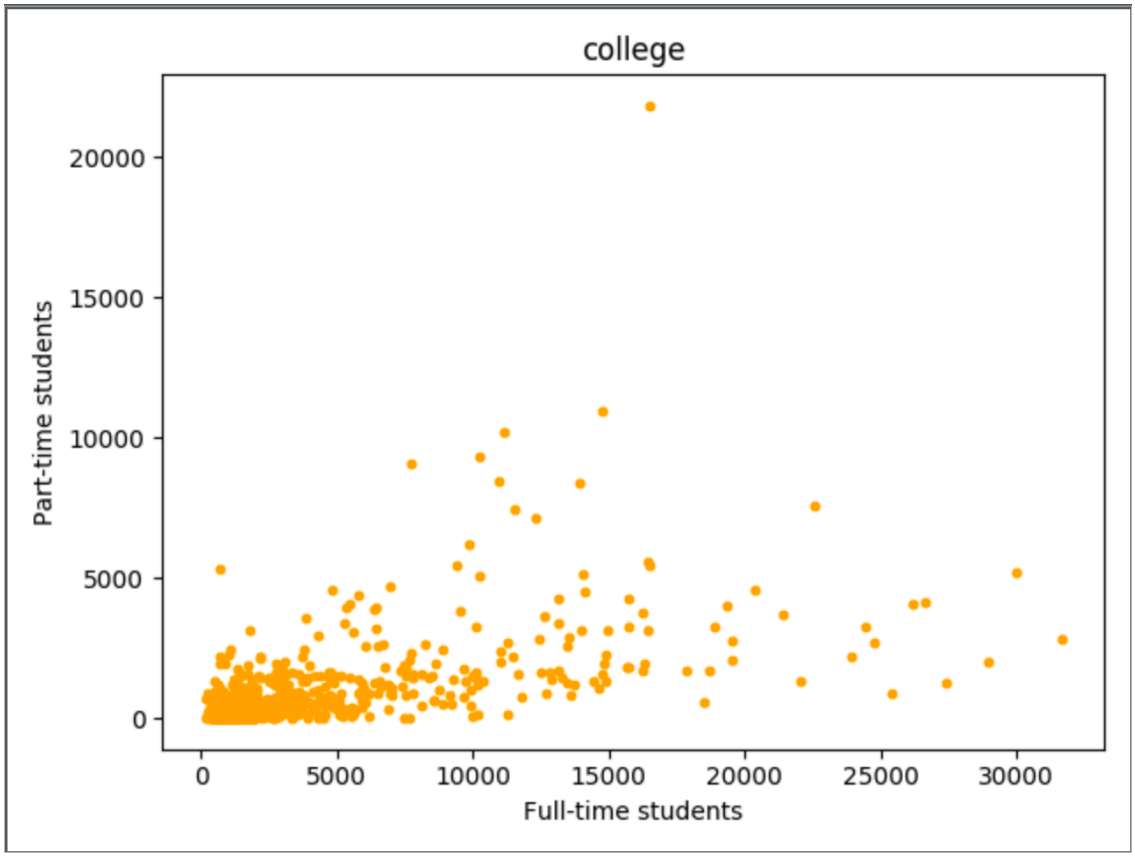
(C) Use a Scatterplot to show the quantitative relationship between part-time and full-time undergraduates.

```
FullTimeStudent = []
PartTimeStudent = []

with open('College.csv', newline='') as csvfile:
    college = csv.reader(csvfile, quotechar='|')
    for row in college:
        if count > 0:
            FullTimeStudent.append(int(row[7]))
            PartTimeStudent.append(int(row[8]))
        count += 1

plt.title("college")
plt.scatter(FullTimeStudent, PartTimeStudent, label="Group 1",
            color="orange", marker="o", s=10)
plt.xlabel('Full-time students')
plt.ylabel('Part-time students')
plt.show()
```

- Use two lists ‘FullTimeStudent’ and ‘PartTimeStudent’ to store the value of column ‘F.Undergrad’ and ‘P.Undergrad’.

- Next, draw a scatterplot to show the relationship between ‘F.Undergrad’ and ‘P.Undergrad’.

- The result is as follow:



- From this plot, we can find that most schools with less than 5000 full-time undergrads have less than 2500 part-time undergrads. Also, in this dataset, most schools have more full-time students than part-time students.

(D) Use a Histogram to show the relationship between the number of students in public and private schools.

```
import csv
import matplotlib.pyplot as plt

PublicSchool = []
PrivateSchool = []

with open('College.csv', newline='') as csvfile:
    college = csv.reader(csvfile, quotechar='|')
    for row in college:
        if row[1] == "Yes":
            PrivateSchool.append(int(row[7]) + int(row[8]))
        elif row[1] == "No":
            PublicSchool.append(int(row[7]) + int(row[8]))

plt.title("college")
plt.hist(PrivateSchool, density=1, bins=60, label='PrivateSchool',
         color='red', stacked=True, alpha = 0.5)
plt.hist(PublicSchool, density=1, bins=60, label='PublicSchool',
         color='blue', stacked=True, alpha = 0.5)
plt.xlabel("The number os students")
plt.ylabel("Probability")
plt.legend()
plt.show()
```

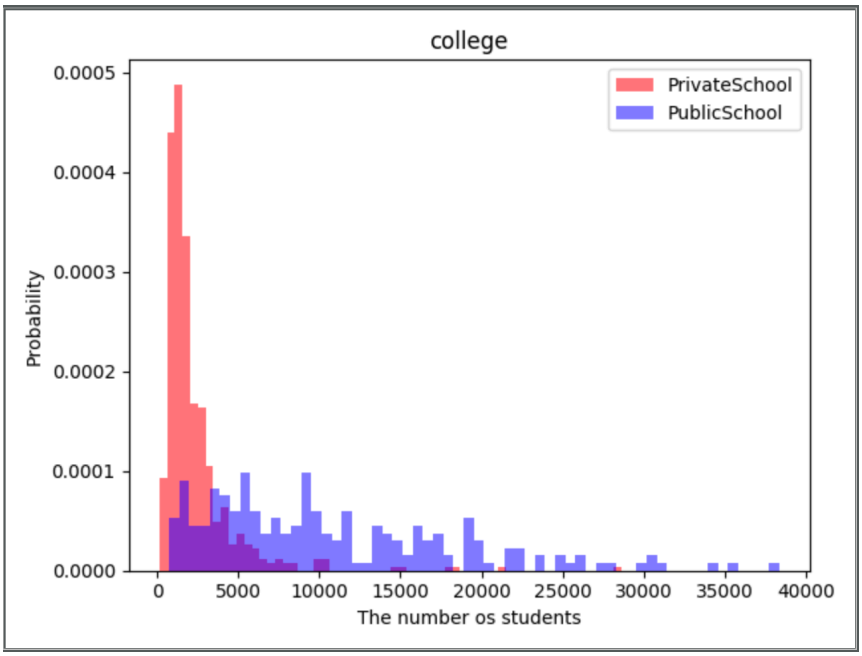
- First of all, set up two lists which are called PublicSchool and PrivateSchool.

- Second, use the value of column “Private” to identify if the following data belong to a private group or public group.

- Next, count the total number of students in each school, and then assign the result to the list which it belongs to.

- Finally, use the histogram function to show the plot. Moreover, make two plots overlap to facilitate observation of the relationship.

-The result is as follow:



- In this plot, we can observe that the number of students in most private schools is below 5000. However, the number of students in public schools is relatively average compared to private schools.

(E) Use box plots to show the acceptance rate of top and not top Universities.

```
import csv
import matplotlib.pyplot as plt

Top = []
Untop = []
count = 0

with open('College.csv', newline='') as csvfile:
    college = csv.reader(csvfile, quotechar='|')
    print("**start read college.csv**")
    for row in college:
        if count > 0:
            if int(row[6]) <= 50:
                Untop.append(int(row[3]) / int(row[2]))
            elif int(row[6]) > 50:
                Top.append(int(row[3]) / int(row[2]))
        count += 1

Top.sort()
Untop.sort()
```

- First of all, set up two lists which are called Top and Untop.

- Second, use the value of column “Top25perc” to identify if the following data belong to a Top group or Untop group. If the value is bigger than 50, then the follow data belongs to the Top list; otherwise, it belongs to the Untop list.

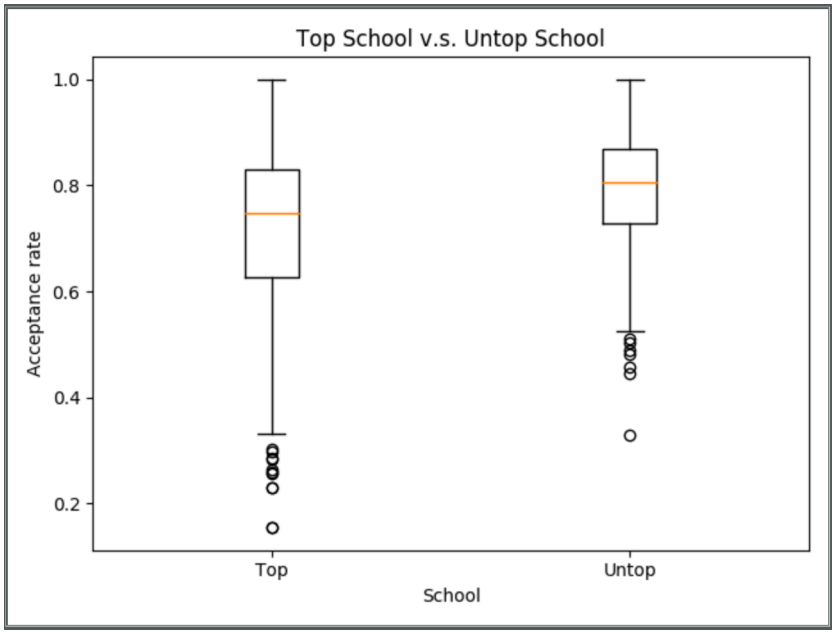
- Next, count the number of acceptance rate (the number of acceptance / the number of application) of each school, and then assign the result to the list which it belongs to.

```
plt.title("Top School v.s. Untop School")
plt.boxplot([Top, Untop])
plt.xticks((1, 2), ('Top', 'Untop'))
plt.xlabel("School")
plt.ylabel("Acceptance rate")
plt.show()
print("There are", len(Top), "top universities")
```

```
question E
/Users/angel/PycharmProjects/untitled3/venv/bin/python
**start read college.csv**
There are 449 top universities
```

- Finally, use the box plot function to show the plot, and then show the answer of how many top universities are in the dataset. The answer is 449 universities.

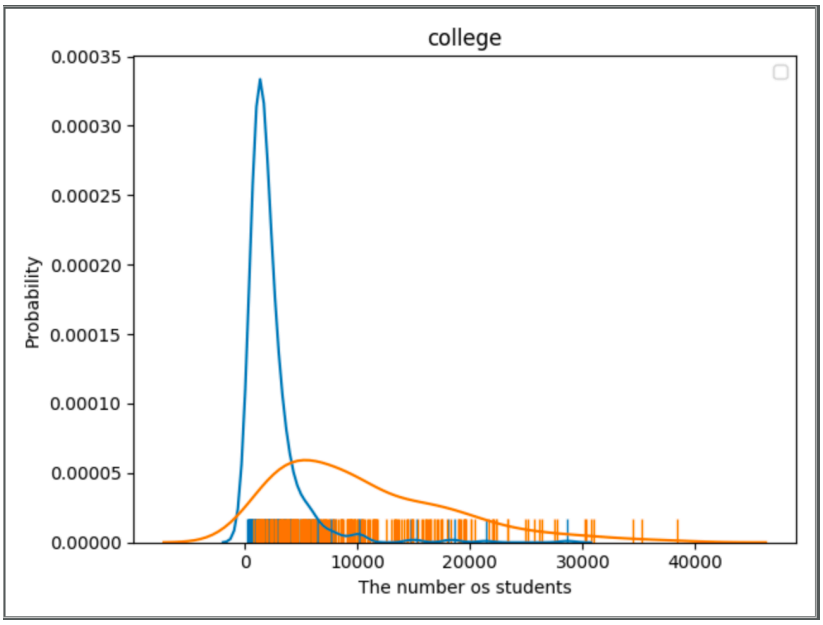
-The result is as follow:

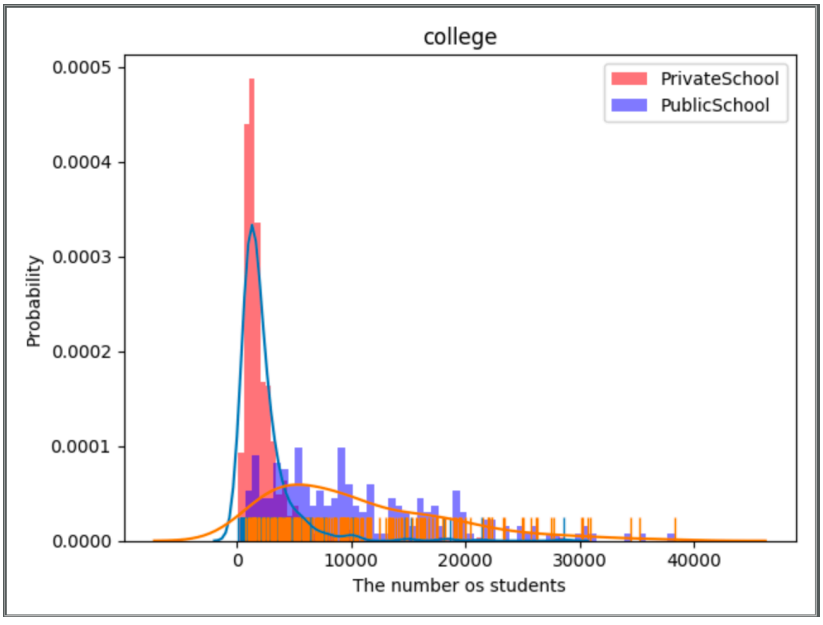


(F) Use different plots to show the relationship between full-time and part-time students.

- When I was doing question D, I found that there are another ways can show the relationship.

- First, I not only can use rectangles but also can use curves in the plots.

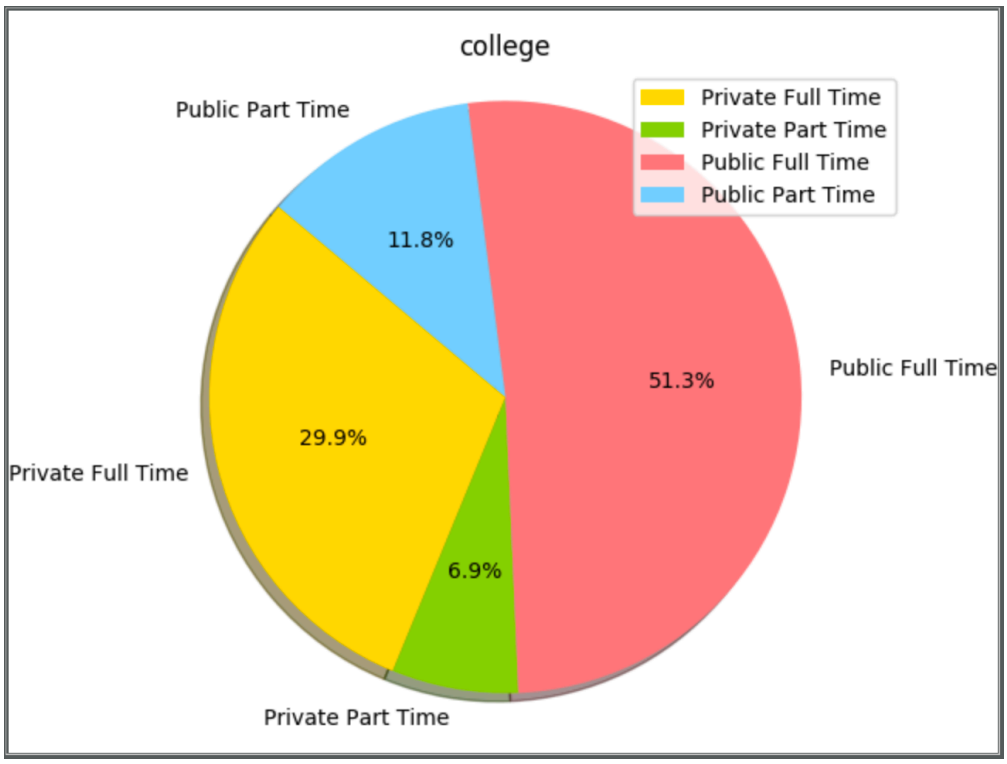




- Then, I was wondering that if I can calculate the curves in the plots, maybe I can make predictions. Therefore, I tried to use Newton's divided difference interpolation formula to get the curves. However, the result look doesn't match the curves in the plots.

- Next, I used a pie chat to show the number of full-time and part-time students of Private

and Public Schools which make people easy understand the differences between them.



- As the result, I like the colors of these plots, and I believe these plots will be helpful in my research in the future.

2. Auto

(A) Quantitative v.s. qualitative

- In my opinion, I think that mpg, cylinders, displacement, horsepower, weight, acceleration, and year are quantitative, but origin and name are qualitative. Although some classmates think that year should be counted in qualitative, I tend to believe that the automotive industry is improving every year. For example, the raw materials, accessories and drivers that produce the car each year can affect the weight and horsepower of the car. Although there is no absolute relationship at present, when the year is combined with other conditions, there may be wonderful results.

- After I tried to store all value of the dataset, I found that there is a problem. In column 'horsepower', there are some question marks which should be counted, so I skipped those

marks.

- The code of read ‘Auto.csv’ is as follow:

```
import csv
import statistics
mpg = []
cylinders = []
displacement = []
horsepower = []
weight = []
acceleration = []
year = []
count = 0

with open('Auto.csv', newline='') as csvfile:
    Auto = csv.reader(csvfile, quotechar='|')
    for row in Auto:
        if count > 0:
            mpg.append(float(row[0]))
            cylinders.append(int(row[1]))
            displacement.append(float(row[2]))
            if row[3]!='?':
                horsepower.append(int(row[3])) #some ?
            weight.append(int(row[4]))
            acceleration.append((float(row[5])))
            year.append(int(row[6]))
        count += 1
```

- First of all, set up seven lists which are called mpg and cylinders, displacement, horsepower, weight, acceleration, and year.

- Second, use a variable ‘count’ with an initial value of 0 to avoid storing the attributes of the first line.

- Third, store all values which are read from the dataset, and then skip all question marks. (Cause we skip those marks, the length of list horsepower will shorter than other lists.)

(B) The range, mean and standard deviation of each quantitative predictor

- After store all data, we can count the ranges, means and standard deviations.

-The result is as follow:

```
print("Range of mpg: ", min(mpg), '~', max(mpg), '/ mean: ',
      sum(mpg)/count, '/ standard deviation: ', statistics.stdev(mpg))
print("Range of cylinders: ", min(cylinders), '~', max(cylinders), '/ mean: ',
      sum(cylinders)/count, '/ standard deviation: ', statistics.stdev(cylinders))
print("Range of displacement: ", min(displacement), '~', max(displacement), '/ mean: ',
      sum(displacement)/count, '/ standard deviation: ', statistics.stdev(displacement))
print("Range of horsepower: ", min(horsepower), '~', max(horsepower), '/ mean: ',
      sum(horsepower)/len(horsepower), '/ standard deviation: ', statistics.stdev(horsepower))
print("Range of weight: ", min(weight), '~', max(weight), '/ mean: ',
      sum(weight)/count, '/ standard deviation: ', statistics.stdev(weight))
print("Range of acceleration: ", min(acceleration), '~', max(acceleration), '/ mean: ',
      sum(acceleration)/count, '/ standard deviation: ', statistics.stdev(acceleration))
print("Range of year: ", min(year), '~', max(year), '/ mean: ',
      sum(year)/count, '/ standard deviation: ', statistics.stdev(year))
```

```
question A ×
/Users/angel/PycharmProjects/DataScienceHW2.2/venv/bin/python "/Users/angel/PycharmProjects/DataScienceHW
Range of mpg:  9.0 ~ 46.6 / mean:  23.456783919597996 / standard deviation:  7.825803928946562
Range of cylinders:  3 ~ 8 / mean:  5.444723618090452 / standard deviation:  1.7015769807918462
Range of displacement:  68.0 ~ 455.0 / mean:  193.0464824120603 / standard deviation:  104.37958329992955
Range of horsepower:  46 ~ 230 / mean:  104.46938775510205 / standard deviation:  38.49115993282849
Range of weight:  1613 ~ 5140 / mean:  2962.7989949748744 / standard deviation:  847.9041194897246
Range of acceleration:  8.0 ~ 24.8 / mean:  15.516582914572849 / standard deviation:  2.749995292976151
Range of year:  70 ~ 82 / mean:  75.80402010050251 / standard deviation:  3.690004901461682
```

- Because I import statistics package, I can use a function which is called ‘stdev’ to count the standard deviations.

(C) The range, mean and standard deviation of each quantitative predictor(remove 45th~85th)

- In this part, I use some conditions to filter the number of rows accessed.

```
for row in Auto:
    if (count > 0 and count < 45) or (count > 85):
        mpg.append(float(row[0]))
        cylinders.append(int(row[1]))
        displacement.append(float(row[2]))
        if row[3]!='?':
            horsepower.append(int(row[3])) #some '?'
        weight.append(int(row[4]))
        acceleration.append((float(row[5])))
        year.append(int(row[6]))
    count += 1
```

- The result of the ranges, means and standard deviations is as follow:

```
print("Range of mpg: ", min(mpg), '~', max(mpg), '/ mean: ',
      sum(mpg)/len(mpg), '/ standard deviation: ', statistics.stdev(mpg))
print("Range of cylinders: ", min(cylinders), '~', max(cylinders), '/ mean: ',
      sum(cylinders)/len(cylinders), '/ standard deviation: ', statistics.stdev(cylinders))
print("Range of displacement: ", min(displacement), '~', max(displacement), '/ mean: ',
      sum(displacement)/len(displacement), '/ standard deviation: ', statistics.stdev(displacement))
print("Range of horsepower: ", min(horsepower), '~', max(horsepower), '/ mean: ',
      sum(horsepower)/len(horsepower), '/ standard deviation: ', statistics.stdev(horsepower))
print("Range of weight: ", min(weight), '~', max(weight), '/ mean: ',
      sum(weight)/len(weight), '/ standard deviation: ', statistics.stdev(weight))
print("Range of acceleration: ", min(acceleration), '~', max(acceleration), '/ mean: ',
      sum(acceleration)/len(acceleration), '/ standard deviation: ', statistics.stdev(acceleration))
print("Range of year: ", min(year), '~', max(year), '/ mean: ',
      sum(year)/len(year), '/ standard deviation: ', statistics.stdev(year))
```

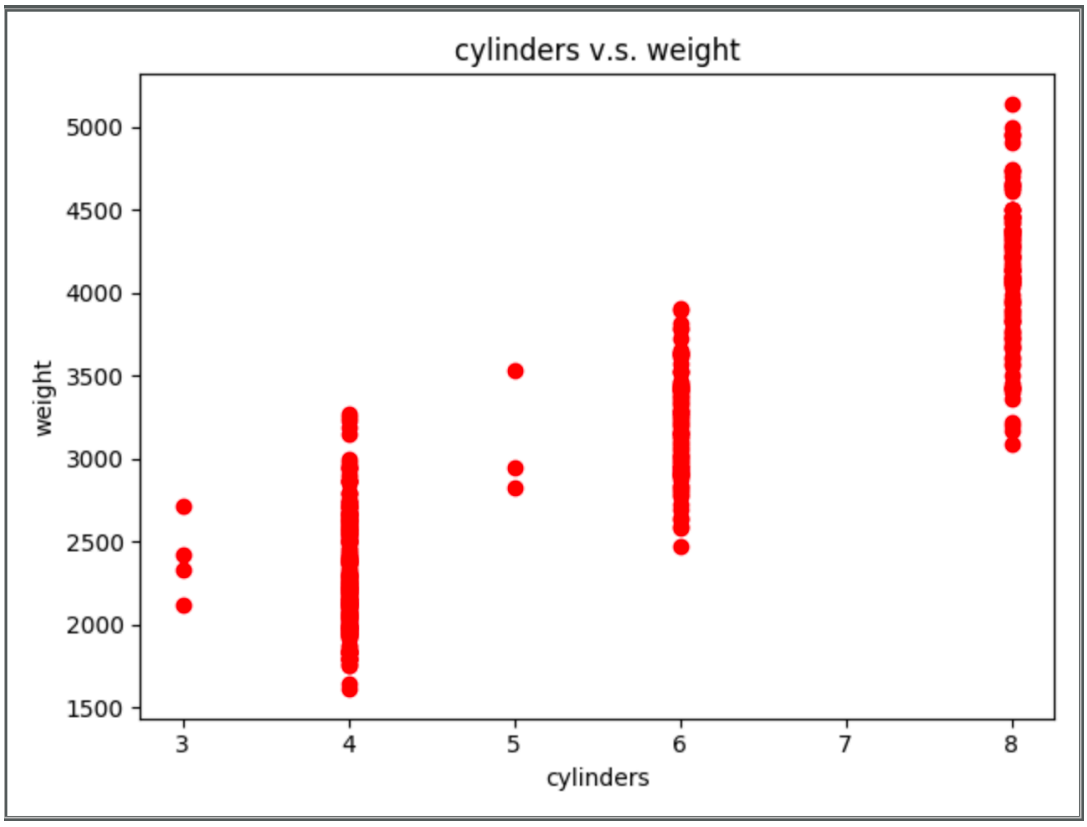
with open('Auto.csv', newline='... › for row in Auto › if (count > 0 and count < 45) o...

question C ×

```
/Users/angel/PycharmProjects/DataScienceHW2.2/venv/bin/python "/Users/angel/PycharmProjects/DataScienceHW2
Range of mpg:  9.0 ~ 46.6 / mean:  23.85337078651685 / standard deviation:  7.919259098055202
Range of cylinders:  3 ~ 8 / mean:  5.455056179775281 / standard deviation:  1.678598425142619
Range of displacement:  68.0 ~ 455.0 / mean:  192.93258426966293 / standard deviation:  102.67314996632257
Range of horsepower:  46 ~ 230 / mean:  103.86324786324786 / standard deviation:  38.23676002029744
Range of weight:  1649 ~ 4997 / mean:  2966.1516853932585 / standard deviation:  828.0294949652233
Range of acceleration:  8.0 ~ 24.8 / mean:  15.560674157303362 / standard deviation:  2.7395946231443182
Range of year:  70 ~ 82 / mean:  76.49157303370787 / standard deviation:  3.57336576223417
```

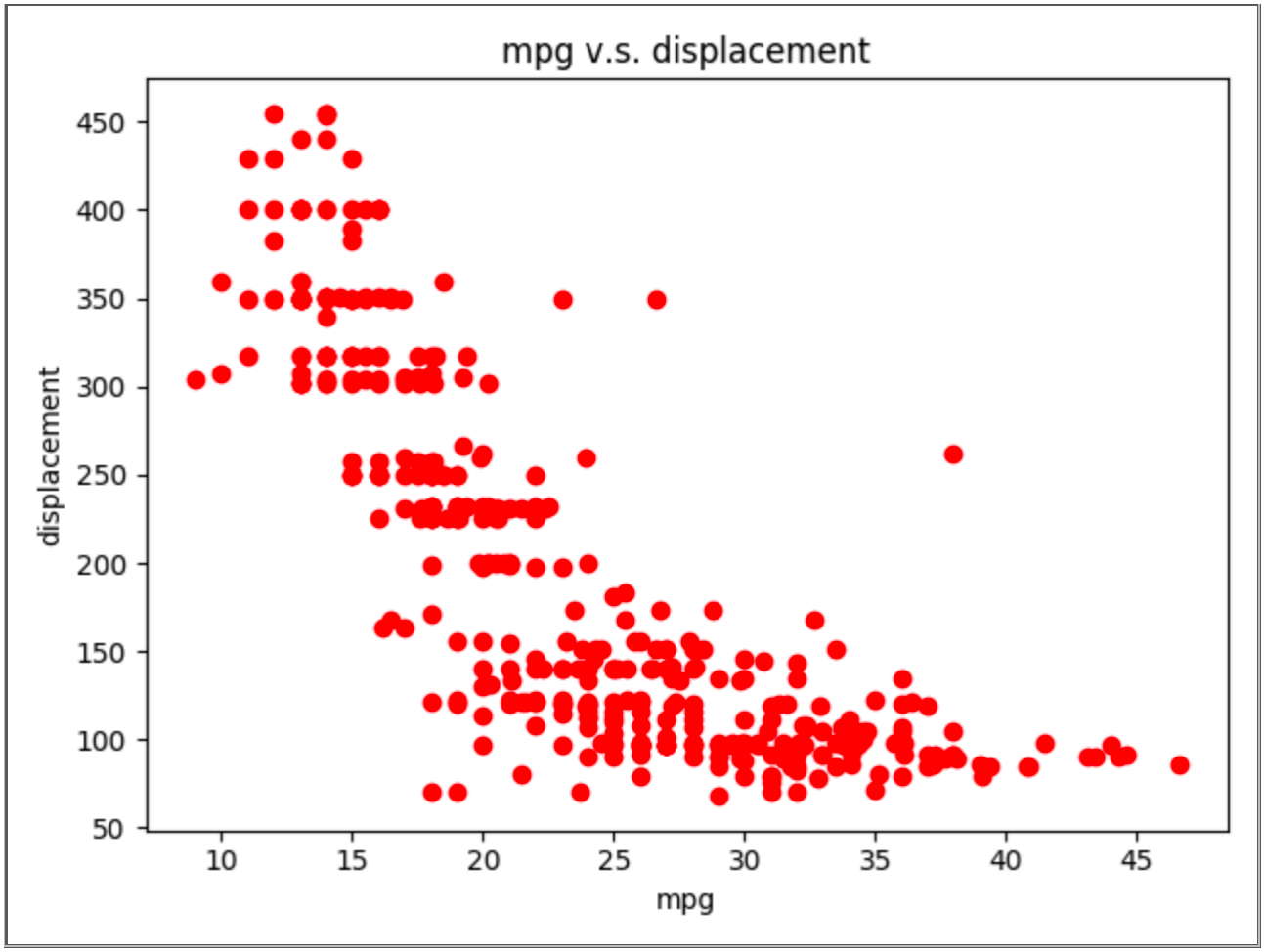
(D) Use a scatterplot to show the relationship between cylinders and weight.

- The result is as follow:



- According to the scatterplot, we can find that the weight will grow as the cylinders become larger.

(E) The variable displacement is helpful for predicting mpg



- According to this scatterplot, there is a curve visible to the naked eye. If this curve can be calculated, we can use it to predict mpg in the future.