Practicum 4: Blue Bike Commuting

Due Date: October 1st at 11:59pm EST

The Bluebikes system is a bicycle sharing system that operates in the greater Boston area. Anyone who subscribes to the service can rent a blue bicycle at a "station" and ride it to another station, where they drop off the bike. For example, you can rent a bike in Central Square in Cambridge and drop it off at a station at Northeastern University in Boston.

Bluebikes collects data on the usage of their bikes, including which stations people are using to pick up and drop off bikes, and how long their trips are. This helps them make sure they're meeting the supply and demand of different areas and charging appropriate rates. We will use Bluebike travel data from a single bike used during August 2021 to study urban biking patterns.

Download the bike's travel data and place it *in the same folder* as the `practicum04.py` file.

We'll use the commuting data to practice the following:

1. Reading many lines of data from a file
2. Calculate summary statistics
3. Visualize distributions of data

1 Reading in Trip Durations

The data file provides data for the ride history of a single bike, which was ridden around the greater Boston area by different people over the course of a month. It is a comma-separated value (.csv) file, which is one of the most common ways of storing and sharing data. You can think of CSV files as giant spreadsheets. Each line (row) represents one trip with the bike. In each row there are several values (columns), each of which is separated by a comma. In order in each row, those values are:

- The duration of the trip, in seconds
- The name of the station where the trip started
- The latitude of the station where the trip started
- The longitude of the station where the trip started
- The name of the station where the trip ended
- The latitude of the station where the trip ended
- The longitude of the station where the trip ended
- The type of Blue Bike user, either a "Subscriber" who regularly pays to use Bluebike or a "Customer" who paid just once to use Bluebike

If we read a line of a CSV, we can get the different values of each line by using the `split()` function. What `split()` does is separate a string based on a character, like a space or comma. When it does this, it returns the different parts of the string as a *list*. To see this, write a sentence in Python, put it into a variable, and print out what happens when you use `split()`.

```
sentence = "Blue Bikes help promote bike commuting"
split_sentence = sentence.split()
print(split_sentence)
```

You should see that Python has split the sentence into a list of words. By default, `split()` separates strings based on white space. We can tell it to separate based on commas instead. Try the following:

```
line = "1,2,3,4,5"
split_line = line.split(",")
print(split_line)
```

Write Python code that does the following:

1. Defines an empty list that will hold trip duration values. To make an empty list, do something like: `durations = []`
2. Opens the bike trip data file.
3. Starts a while loop.
4. Reads a line and uses the `split()` function described above to separate the values of the line. Remember to use `strip()` to remove the newline character `\n` from the end of each line *before* you do the split.
5. Assigns each value in the list to an appropriate variable. Remember that the order of the values in each line is described above. For example, the first value in the list is always the trip duration. We can store it in its own variable by accessing the first element in the list: `duration = split_line[0]`. Remember to transform the duration into a number, so that it's not just a string.
6. Places the `duration` in the `durations` list that you defined earlier. You can place a value into a list using `append()`, which places the value at the end of the list: `durations.append(duration)`
7. Exits the while loop when there are no more lines to read

Try `print(durations)` after you're done to make sure that the list has been populated like you expected.

2 Subscribers and Customers

People who subscribe to Bluebike regularly (Subscribers) and people who only use Bluebike once at a time (Customers) may have different commuting patterns. Write Python code that does the following:

1. Defines two lists: one that holds trip durations of subscribers, and one that holds durations of customers.
2. In the while loop, checks if a line represents a trip made by a Subscriber or a Customer, and stores the duration in the respective list.

Once you have the duration data for both Subscribers and Customers, you'll calculate some summary statistics. Write Python code that does the following:

1. Calculates the shortest trip for Subscribers and Customers using the built-in `min()` function. Note: the durations are in seconds. You may want to convert to minutes by dividing by 60 to make it easier to interpret the statistic.
2. Calculates the longest trip for Subscribers and Customers using the built-in `max()` function.
3. Calculates the average trip duration for Subscribers and Customers. The average can be calculated by taking the sum of all trip durations (use the built-in `sum()` function) and dividing by the total number of trips (use the built-in `len()` function).
4. Prints the statistics for the user to see.

The program output should look like this:

```
The average trip duration for Subscribers was 10 minutes
The shortest trip was 0.2 minutes, and the longest trip was 60 minutes

The average trip duration for Customers was 5 minutes
The shortest trip was 0.15 minutes, and the longest trip was 40 minutes
```

Finally, we'll use box plots to visualize the data. Box plots show the distribution of data by marking several summary statistics: the minimum and maximum values, the median, the 25th and 75th percentiles, and the range between those percentiles. You can make a boxplot like:

```
# Remember to import matplotlib

# Note that the position is wrapped in a list, [1]
plt.boxplot(durations, positions=[1])
```

You'll want to change `positions` so that it's `positions=[2]` for the second box plot.

3 Trip Distances and Speeds

Let's calculate summary statistics for the speed of the Subscriber and Customer trips as well. We can calculate speed as:

$$speed = \frac{distance}{total\ time}$$

Using the latitude (lat) and longitude (lon), we can calculate the distance as:

$$distance = 2R * \arcsin\sqrt{\sin(\frac{lat_{finish} - lat_{start}}{2})^2 + \cos(lat_{start}) * \cos(lat_{finish}) * \sin(\frac{lon_{finish} - lon_{start}}{2})^2}$$

Use the `sin()`, `cos()`, and `asin()` functions from the `math` library. In the above equation, R is the radius of the earth. Put the constant `RADIUS = 3959.87433` at the top of your Python code and use that for R. The final distance will be in terms of miles.

Note, this is the straight line distance (approximately) between the start station and end station. Since we can rarely bike in a single straight line from one place to another in Boston, it would be more accurate to calculate the distance in terms of the road route that was taken. Unfortunately, the exact routes are not available in our data, so this will have to do.

Write Python code that does the following:

1. Calculates the speed of each trip. Convert the speed to miles per hour by converting the trip duration from seconds to hours.
2. Calculates the minimum, maximum, and average speeds of the trips for Subscribers and Customers.
3. Prints out the summary statistics.
4. Visualizes the speed distributions using box plots.

Finish Early?

1. Calculate the standard deviation for the trip durations and speeds. Write your own code to do this, and do not use a function from an external package. You can calculate the standard deviation as follows. N is the total number of trips, and $\bar{x}$ is the average.

   $$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

2. Try visualizing your data with the hist() function instead.

3. Did you notice that the slowest trip speed for both subscribers and customers is 0 miles per hour? Why is that? How should you account for that in your analyses?

4 Submit Your Work

When you are done, submit the following to Canvas in a zip folder:

1. The Python file (.py) containing your code answers for this practicum.
2. The Blue Bike data file (.csv)
3. A text file (.txt, .doc, .pdf, etc.) with any comments about anything you were not able to get working and what you tried to solve it.

Please name your zip folder `LastName_Practicum04.zip`.

This assignment was originally created by Carolina Mattsson and Stefan McCabe.
It has been updated here by Ryan Gallagher.