Shuowen Wei
weisw9@gmail.com

Question 1

- **Programmatically download and load into your favorite analytical tool the trip data for September 2015.**
- **Report how many rows and columns of data you have loaded.**

A:
Please refer to the program *Question1_SW.py*

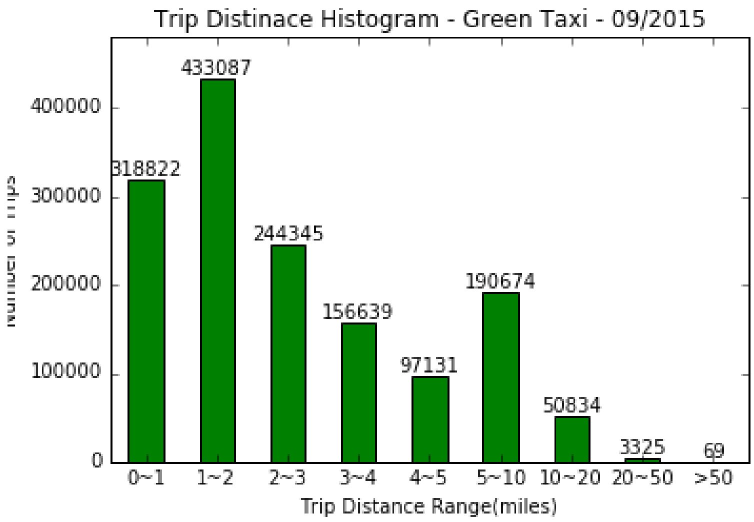
After successfully downloading the Green taxi trip data of Sep. 2015, there're 1494926 rows and 21 columns of data were loaded.

Question 2

- **Plot a histogram of the number of the trip distance ("Trip Distance").**
- **Report any structure you find and any hypotheses you have about that structure.**

A:
Please refer to the program *Question2_SW.py*

Here is the histogram of the number of the trip distance:



Since the bins on x-axis are rescaled, the structure of the histogram can be a little different but it's still clear that all the higher bars are on the left side of the graph, that indicates the majority of trips can be fairly defined as "short trip".

Out of total 1494926 records in the data set (without data cleaning), some analysis is as below (also printed in the python program):
Number of trips with distances < 5 miles: 1250024, or 83.62%.
Number of trips with distances < 10 miles: 1440698, or 96.37%.
Number of trips with distances >= 100 miles: 10, or 0.0007%.

Question 3

- **Report mean and median trip distance grouped by hour of day.**
- **We'd like to get a rough sense of identifying trips that originate or terminate at one of the NYC area airports. Can you provide a count of how many transactions fit this criteria, the average fair, and any other interesting characteristics of these trips.**

A:
Please refer to the program *Question3_SW.py.* This program needs proper python environment setup before running, as it utilizes libraries like *geopandas* and *shapely*, please read the notes first in the comments.

Below is the mean and median trip distance grouped by hour of day:

Hour	[mean, median]	Hour	[mean, median]	Hour	[mean, median]
0	[3.12, 2.2]	8	[3.05, 1.98]	16	[2.78, 1.8]
1	[3.02, 2.12]	9	[3.0, 1.96]	17	[2.68, 1.78]
2	[3.05, 2.14]	10	[2.94, 1.92]	18	[2.65, 1.8]
3	[3.21, 2.2]	11	[2.91, 1.88]	19	[2.72, 1.85]
4	[3.53, 2.36]	12	[2.9, 1.89]	20	[2.78, 1.9]
5	[4.13, 2.9]	13	[2.88, 1.84]	21	[3.0, 2.03]
6	[4.06, 2.84]	14	[2.86, 1.83]	22	[3.19, 2.2]
7	[3.28, 2.17]	15	[2.86, 1.81]	23	[3.19, 2.22]

NYC area airports mainly include the John F. Kennedy International Airport and LaGuardia Airport, the third party reference data set "Pediacities NYC Neighborhoods" is utilized here.

Regarding the trips originate or terminate at one of the two airports, some interesting characteristics of those trips are below:

	Originate at airport	Terminate at airport
Number of transactions	701	34262
Average fair	$29.46	$28.95
Average tip	$4.04	$3.98
Average tolls	$0.27	$1.10
Average distance	3.68 miles	9.17 miles

From the table above, we can that people trend to take a taxi when they go to airport to catch a flight (as more trips terminate at airports), and they trend to choose roll roads (this may because their time before flight is limited), and people who usually take taxies to airport may live farther than those take taxies from airport going back home.

Question 4

- **Build a derived variable for tip as a percentage of the total fare.**
- **Does the tip percentage follow the same distribution for trips originating in upper Manhattan as those originating in the outer boroughs?**
- **Build a predictive model for tip as a percentage of the total fare. Use as much of the data as you like (or all of it). We will validate a sample.**

A:

Please refer to the program *Question4_SW.py.* This program requires the same python environment as Question 3.

The derived variable for tip as a percentage of the total fare can be calculated as below:

```
save_file_name = r'green_tripdata_201509.csv'
df = pd.read_csv(save_file_name, sep=',')
df['tip_pct'] = round( df['Tip_amount'] / df['Fare_amount'],4) * 100
```

Identify upper Manhattan area and outer boroughs of NYC area first:

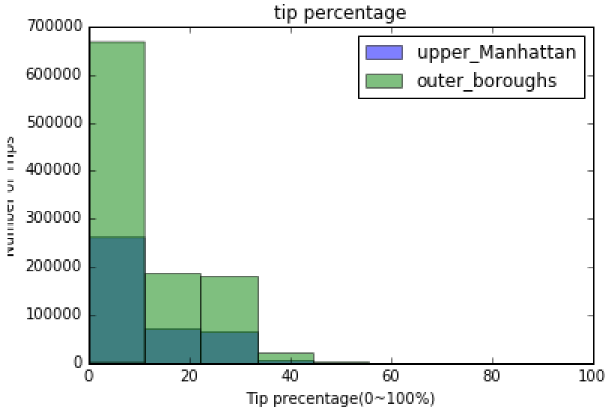
- Upper Manhattan: Midtown, Harlem, Marble Hill, Morningside Heights, Inwood, East Harlem, Upper East Side, Washington Heights, Upper West Side, Randall's Island.
- Outer boroughs: Bronx, Brooklyn, Queens, Staten Island.

```
# trips from the upper Manhattan area and outer boroughs
df = GeoDataFrame(df)
df['geometry'] = df.apply(lambda row: Point(row['Pickup_longitude'], row['Pickup_latitude']), axis=1)

upper_ManhattanTrips = df[ df['geometry'].intersects(upper_Manhattan['geometry'].unary_union) ]
outer_boroughsTrips = df[ df['geometry'].intersects(outer_boroughs['geometry'].unary_union) ]
print('There’re {0:d} trips originating in upper Manhattan. '.format(len(upper_ManhattanTrips)))
print('There’re {0:d} trips originating in outer boroughs. '.format(len(outer_boroughsTrips)))
```

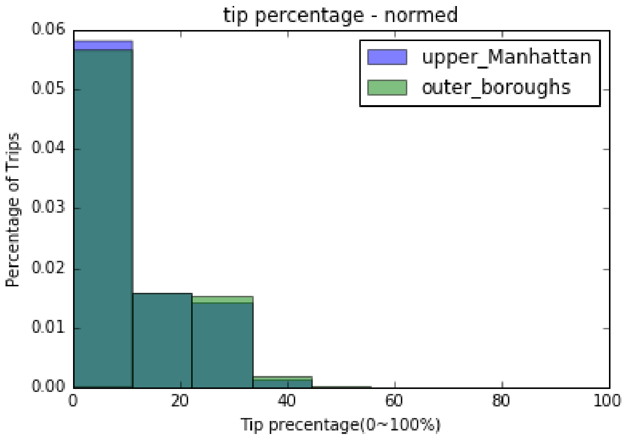
There’re 406,317 trips originating in upper Manhattan and there’re 1,068,821 trips originating in outer boroughs.

Thus, their distributions of the tip percentage of trips are as follows:



Since the numbers of trips originating in outer boroughs are far more than those originating in upper Manhattan, after normalizing the distribution, we have:

```
print('There’re {0:d} trips originating in upper Manhattan. '.format(len(upper_ManhattanTrips)))
print('There’re {0:d} trips originating in outer boroughs. '.format(len(outer_boroughsTrips)))
```



Hence they follow the same distribution.

For building a predictive model for tip as a percentage of the total fare, we cleaned the data first, which results in keeping 99.87% of the data set, then selected original 12 features and derived a few extra features, like 'hourofDay' and 'dayofweel'.

Then the dataset is split randomly into training dataset (67%) and testing dataset (33%):

```
X_train, X_test, y_train, y_test = train_test_split(X_features, Y_labels, test_size=0.33, random_state=42)
```

Then the random forest regression model is applied and its performance is measured by mean squared error (mse), which is 0.012, this is a pretty result.

```
estimator = RandomForestRegressor(random_state=0, n_estimators=100)
modelSettings = estimator.fit (X_train, y_train)
pred_train = estimator.predict (X_test)
mse = mean_squared_error(y_test, pred_train)
print(mse)
```

Question 5

Choose only one of these options to answer for Question 5. There is no preference as to which one you choose. Please select the question that you feel your particular skills and/or expertise are best suited to. If you answer more than one, only the first will be scored.

- **Option A: Distributions**
- **Build a derived variable representing the average speed over the course of a trip.**
- **Can you perform a test to determine if the average trip speeds are materially the same in all weeks of September? If you decide they are not the same, can you form a hypothesis regarding why they differ?**

- **Can you build up a hypothesis of average trip speed as a function of time of day?**

A: Please refer to the program *Question5_optionA_SW.py.*

Derive "timeDiff" variables first, and then compute the "aveSpeed":

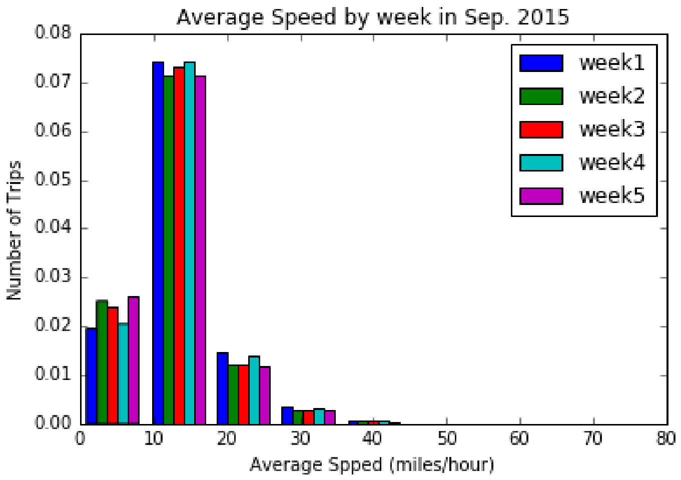
```
df['timeDiff'] = (df['Lpep_dropoff_datetime'] - df['lpep_pickup_datetime']).map(lambda diff: diff.seconds/3600)
df['aveSpeed'] = df['Trip_distance']/df['timeDiff']
```

The by spot checking the data, some criterias are set for cleaning data purpose:

```
df_cleandata = df[  (df.Trip_distance > 0) &
                    (df.timeDiff > 0) &
                    (df.aveSpeed < 80)]
# spot check anormly
#anormly =  df_cleandata[(df_cleandata.aveSpeed > 80 )  ]
```

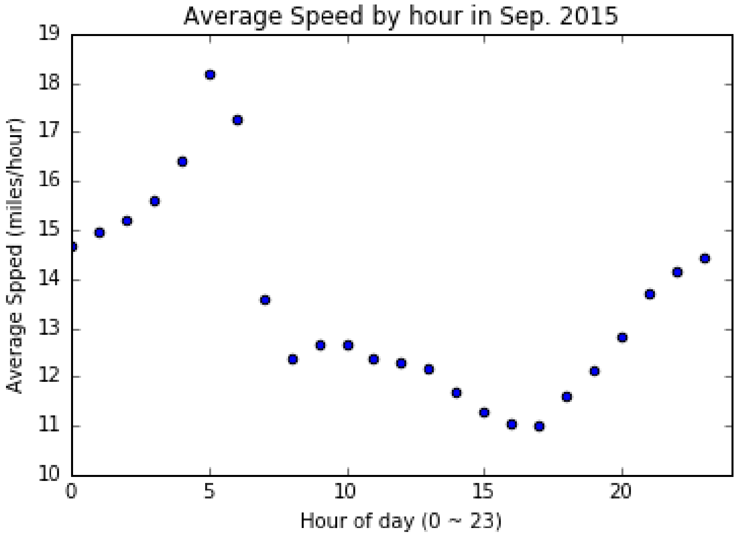
After cleaning the data, the average speed over all the remaining trips is 12.99 miles/hour, this is pretty much expected as New York City is a pretty crowd city and there're a log of traffic lights.

Now group the cleaned dataset by the week of Sep. 2015, compute their average speeds and normalize the conuts in each speed interval (this is because the number of days in the first and last week of September is less than the others'), below is the distribution



It's fairly to propose the hypothesis that average trip speeds are materially the same in all weeks of September.

And below is the scatter plot of average speed v.s. the hour of the day:



The average speed is much higher from 12:00 A.M. to 6:00 A.M., and starts to decline during day time from 7:00AM to 5:00 PM. The lowest average speed is around 5:00PM the traffic rush hour, and the average speed starts to increase after that, all the way to 12:00PM.

References

1. scikit-learn user guide, release 0.12-git by scikit-learn developers, http://www.math.unipd.it/~aiolli/corsi/1213/aa/user_guide-0.12-git.pdf

2. https://www.ocf.berkeley.edu/~dlevitt/2015/12/13/final-project-nyc-taxi-and-uber-data/

3. http://nyctaxi.herokuapp.com/

4. http://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/

5. https://en.wikipedia.org/wiki/Borough_(New_York_City)

6. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

7. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html