ENTS 749C

Vehicular Adhoc Networks

Miniproject-1 Report

Assumptions

The following assumptions are made about the units of the different signal measurements:

Distance – miles

Speed – miles per hour (mph)

Time – seconds

Engine speed – rotations per minute (rpm

Torque – Newton Meter (N.m)

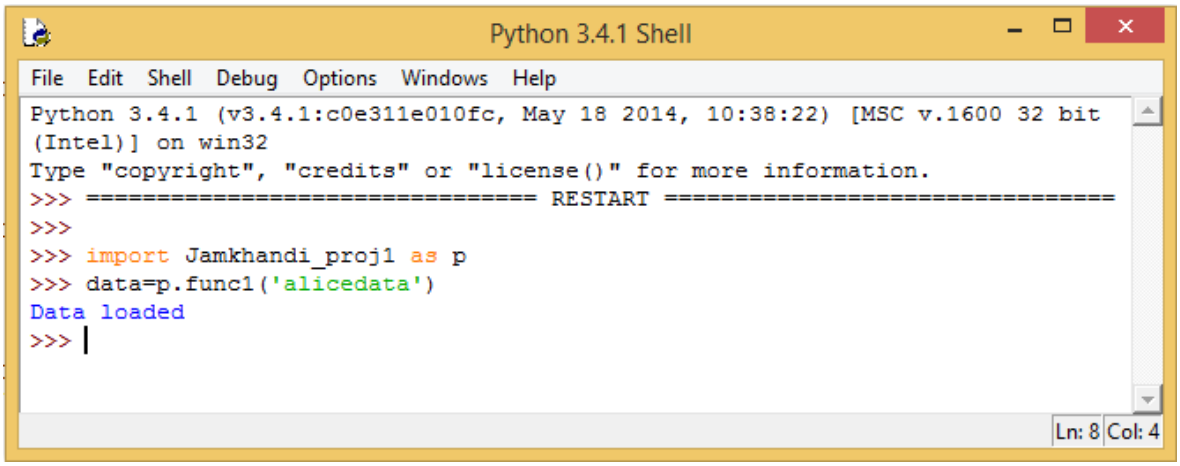
Fuel – Gallons

Python Data Analysis

1. **Function 1**

Python code:

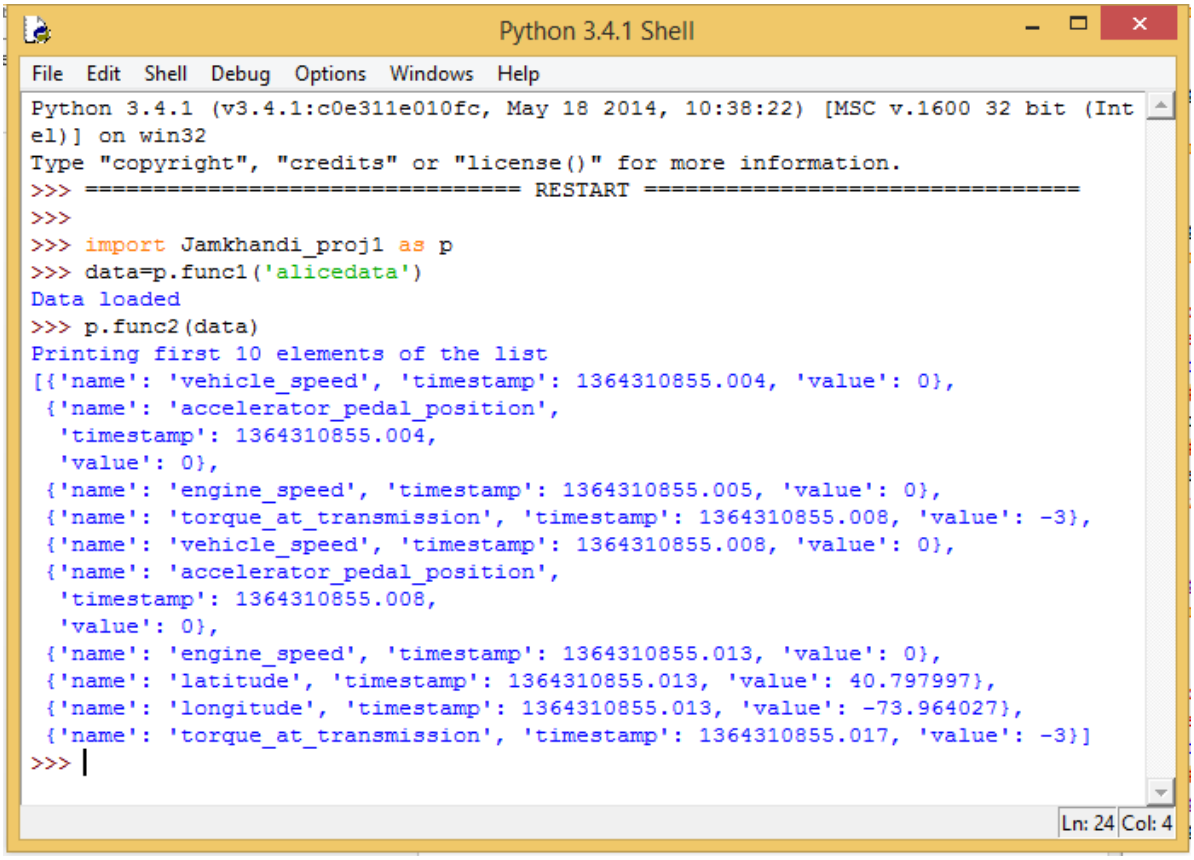
```
#function 1 definition
#This function reads the JSON file and returns a list of python dictionaries.
def func1(data_file):
    #Storing file name in a string
    data_file = data_file + ".json"
    #Declaring an empty list to read data file as a Python list of Python dictionary items
    signal_data = []
    with open(data_file, "r") as f:
        for line in f:
            signal_data.append(json.loads(line))
    return signal_data
```



2. **Function 2**

Python code:

```
#function 2 definition
#This function accepts the processed data as input and prints the first 10 list items
def func2(signal_data):
    #Using pprint to print the first 10 list items
    pprint(signal_data[:10])
```

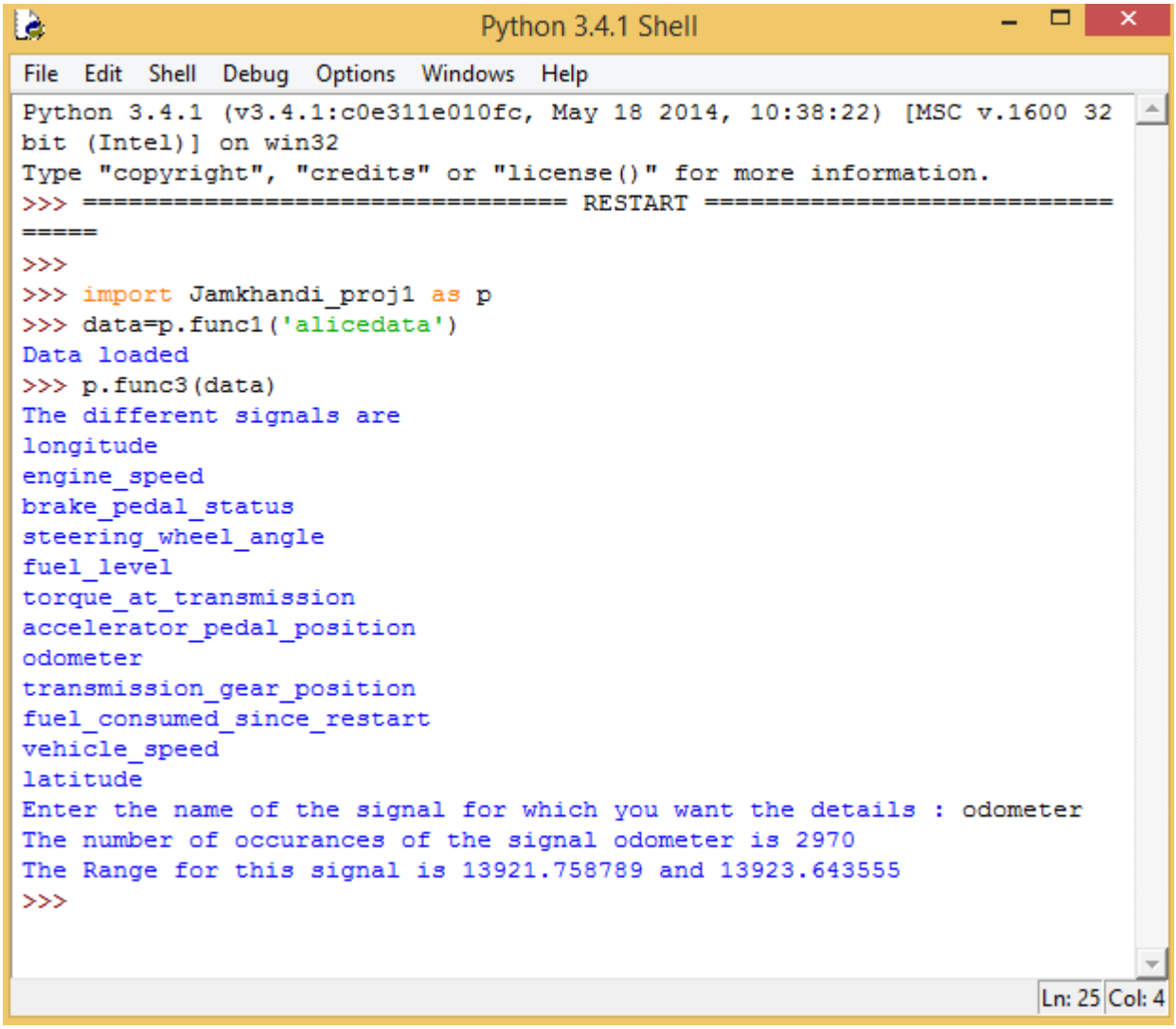


3. Function 3

Python code:

```
#function 3 definition
#This function prints all the different signals in the list.
#It then prints the number of occurrences and the range for the signal that the user
#inputs.
def func3(signal_data):
    #Declaring an empty dictionary
    signals = {}
    #Initializing all the values to empty list
    for line in signal_data:
        signals[line['name']] = []
    #Appending the values to the corresponding key in the dictionary
    for line in signal_data:
        signals[line['name']].append(line['value'])
    print("The different signals are")
    #Printing all the keys-->signals
```

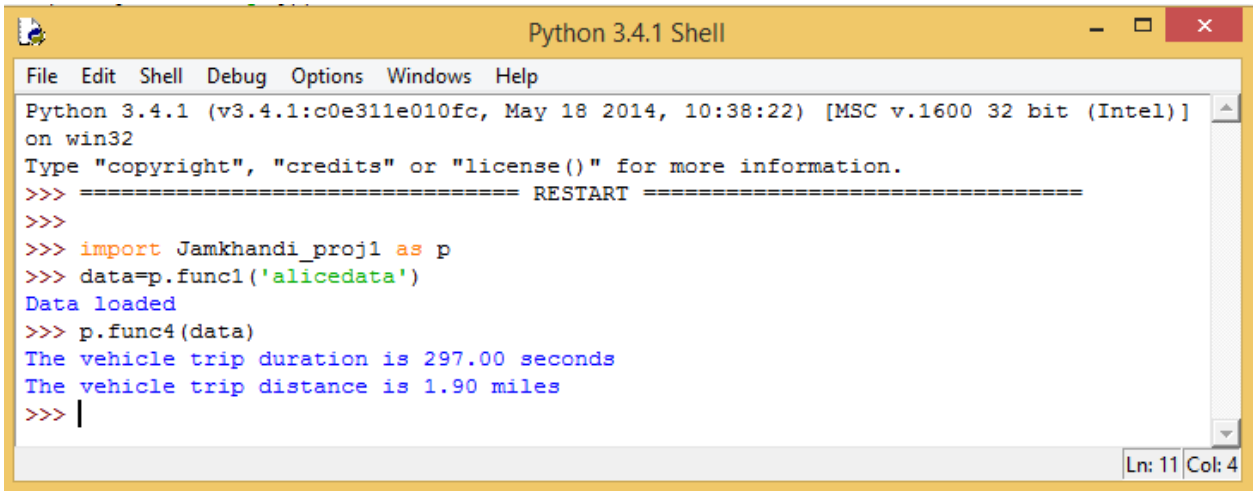
```
for signal in signals.keys():
   print(signal)
input_signal = input("Enter the name of the signal for which you want the details : ")
#Storing the list of values corresponding to the entered signal
values = signals[input_signal]
print("The number of occurances of the signal {0} is {1}".format(input_signal,
        len(values)))
print("The Range for this signal is {0} and {1}".format(min(values), max(values)))
```



4. **Function 4**

Python code:

```
#function 4 definition
#This function finds the vehicle trip duration and trip distance.
#It calculates the average speed and time to find the distance.
def func4(signal_data):
    #Declaring an empty list for timestamp values
    time = []
    for line in signal_data:
        time.append(float(line['timestamp']))
    #Finding the time duration
    duration = (max(time)- min(time))/60
    print("The vehicle trip duration is {0:.2f} seconds".format(duration))
    #Declaring an empty list for vehicle_speed values
    speed = []
    for line in signal_data:
        if 'vehicle_speed' in line.values():
            speed.append(float(line['value']))
    #Finding the average speed of the vehicle for the trip duration
    average_speed = float(sum(speed) / duration)
    #Finding the distance travelled during the trip
    distance = (average_speed*duration)/3600
    print("The vehicle trip distance is {0:.2f} miles".format(distance))
```

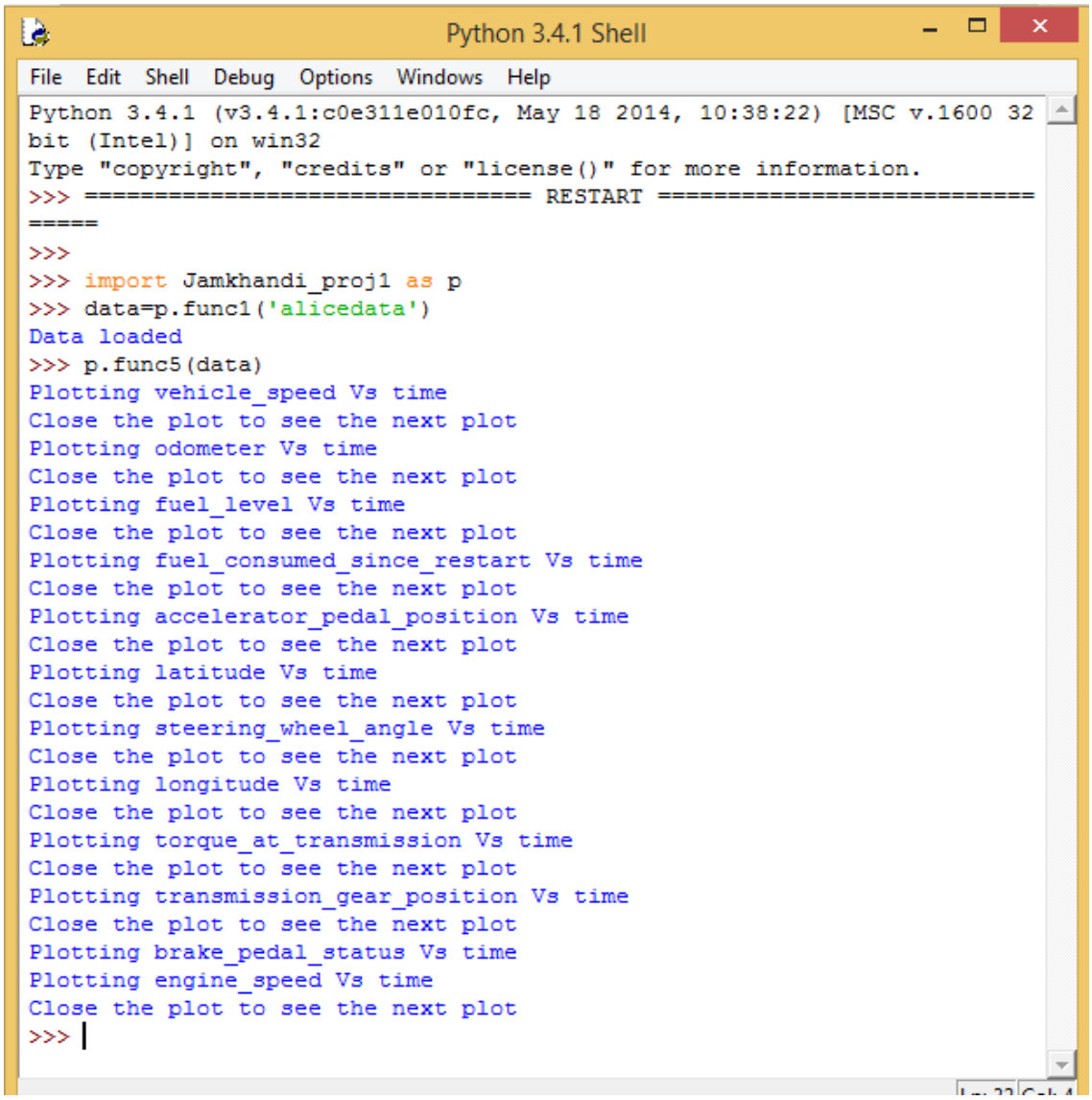


5. **Function 5**

Python code:

```
#function 5 definition
#This function plots the different signals with time.
def func5(signal_data):
    signals = {}
    #Initializing all the values to empty list
    for line in signal_data:
        signals[line['name']] = []
    #Appending the values to the corresponding key in the dictionary
    for line in signal_data:
        signals[line['name']].append([line['value'],line['timestamp']])
    for signal in signals.keys():
        print('Plotting {} Vs time'.format(signal))
        x = []
        y = []
        if (signal != 'brake_pedal_status') and (signal != 'transmission_gear_position'):
            for line in signals[signal]:
                y.append(line[0])
                x.append(line[1])
            pylab.plot(x,y)
            pylab.xlabel('Time')
            pylab.ylabel(signal)
            title = signal + ' Vs ' + 'time'
            pylab.title(title)
            pylab.show()
            print('Close the plot to see the next plot')
        if  (signal == 'transmission_gear_position'):
            for line in signals[signal]:
                x.append(line[1])
                if line[0] == 'first':
                    y.append(1)
                if line[0] == 'second':
                    y.append(2)
                if line[0] == 'third':
                    y.append(3)
                if line[0] == 'fourth':
                    y.append(4)
                if line[0] == 'neutral':
```

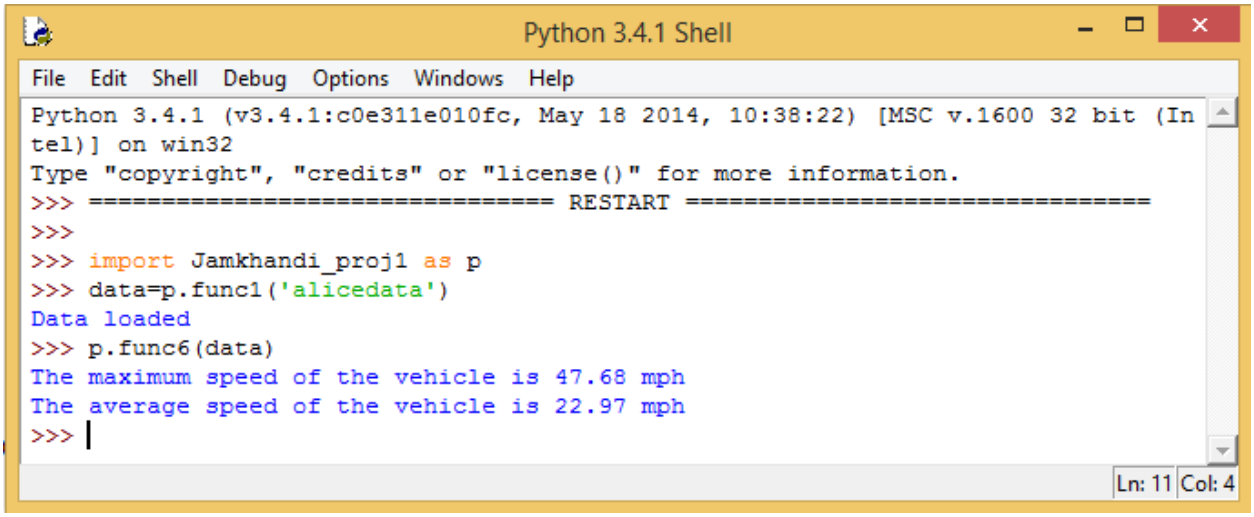
```
        y.append(5)
    pylab.plot(x,y)
    pylab.xlabel('Time')
    pylab.ylabel(signal)
    title = signal + ' Vs ' + 'time'
    pylab.title(title)
    pylab.show()
    print('Close the plot to see the next plot')
```



6. Function 6

Python code:

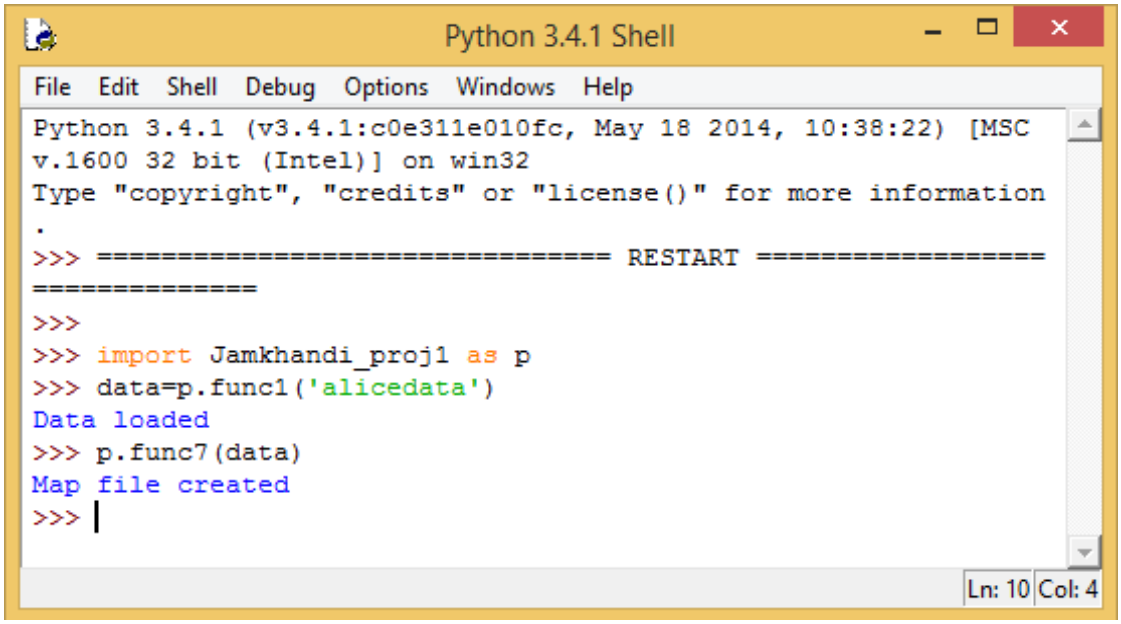
```
#function 6 definition
#This function finds the maximum and average speed of the vehicle.
def func6(signal_data):
    #Declaring an empty list for vehicle_speed values
    speed = []
    for line in signal_data:
        if 'vehicle_speed' in line.values():
            speed.append(float(line['value']))
    #Finding the maximum speed of the vehicle for the trip duration
    max_speed = max(speed)
    #Finding the average speed of the vehicle for the trip duration
    average_speed = float(sum(speed) / duration)
    print("The maximum speed of the vehicle is {} mph".format(max_speed))
    print("The average speed of the vehicle is {} mph".format(average_speed))
```

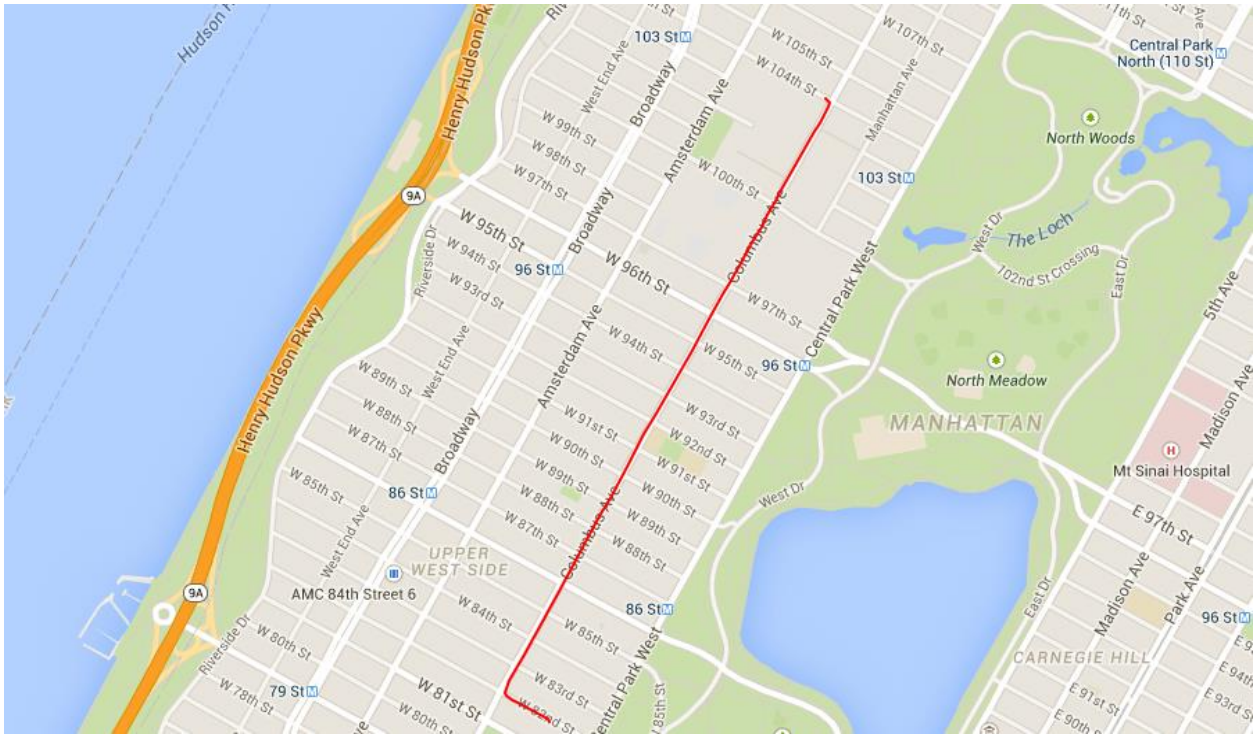


7. **Function 7**

Python code:

```
#function 7 definition
#This function draws the vehicle's path on a google map.
def func7(signal_data):
    #Declaring an empty list to hold the list of coordinates
    coordinates = []
    #Declaring a smaller list to hold the current coordinates
    position = []
    count = 0
    for line in signal_data:
        if 'latitude' in line.values() and count is 0:
            position.append(float(line['value']))
            count = count + 1
            continue
        if 'latitude' in line.values() and count is 1:
            position.insert(0,float(line['value']))
            coordinates.append(tuple(position))
            #Reinitialize position to empty list when both coordinates are found
            position = []
            count = 0
            continue
        if 'longitude' in line.values() and count is 0:
            position.append(float(line['value']))
            count = count + 1
            continue
        if 'longitude' in line.values() and count is 1:
            position.append(float(line['value']))
            coordinates.append(tuple(position))
            #Reinitialize position to empty list when both coordinates are found
            position = []
            count = 0
            continue
    mymap = pygmaps.maps(coordinates[0][0], coordinates[0][1],10)
    mymap.addpath(coordinates, "#FF0000")
    mymap.draw('./mymap.html')
```


Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ========== RESTART ==========
>>>
>>> import Jamkhandi_proj1 as p
>>> data=p.func1('alicedata')
Data loaded
>>> p.func7(data)
Map file created
>>>
Ln: 10 Col: 4


Henry Hudson Pkwy
Broadway
Amsterdam Ave
Columbus Ave
Central Park West
West End Ave
Riverside Dr
103 St
96 St
86 St
79 St
W 81st St
North Woods
The Loch
North Meadow
MANHATTAN
Mt Sinai Hospital
CARNEGIE HILL
UPPER WEST SIDE
AMC 84th Street 6
Central Park North (110 St)
5th Ave
Madison Ave
Park Ave
E 97th St
West Dr
East Dr

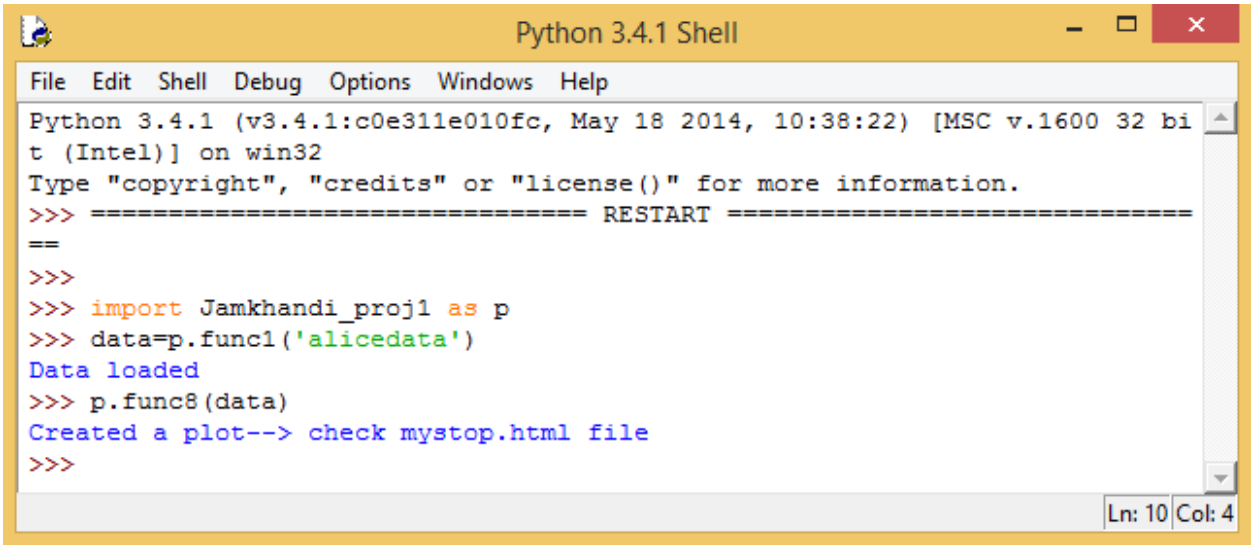
8. Function 8

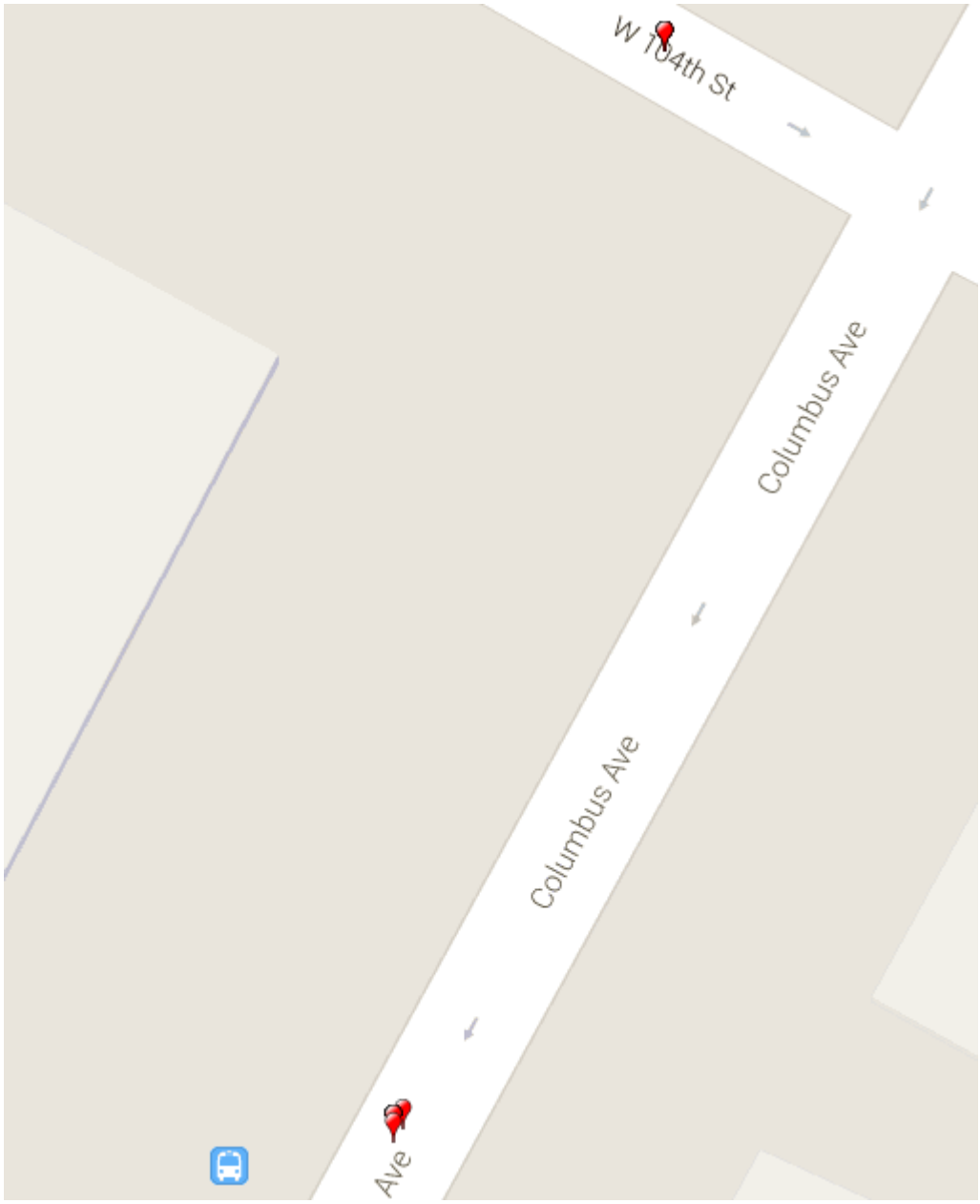
This is an open-ended question. I used the signal data to see if the driver stopped at the right places, like a stop signal or a signal at an intersection. The timestamps where the vehicle speed is zero are collected. The latitude and longitude corresponding to those timestamps are plotted on a google map. A street view of the map gives information about the stop signal, pedestrian crossing, etc. As seen in the picture, Alice's data shows that she stopped at 2 places. One was at a pedestrian crossing and the other probably at a signal. This information can be used to check if Alice stopped at places where she was supposed to and if she if following traffic rules.

Python code:

```
#function 8 definition
#This function plots the points where the vehicle stopped on a google map.
def func8(signal_data):
    #Creating a dictionary with keys and values as times when vehicle speed was zero
    #We want for 1 second resolution and we want distinct time values
    timestamps = {}
    for line in signal_data:
        if ('vehicle_speed' in line.values()) and (float(line['value']) == 0):
            time = float(line['timestamp'])
            timestamps[round(time, 0)] = 1
    #Declaring an empty list to hold the list of coordinates
    coordinates = {}
    #Declaring a smaller list to hold the current coordinates
    position = []
    count = 0
    for line in signal_data:
        if 'latitude' in line.values() and count is 0:
            position.append(float(line['value']))
            count = count + 1
            continue
        if 'latitude' in line.values() and count is 1:
            position.insert(0,float(line['value']))
            time = float(line['timestamp'])
            coordinates[round(time, 0)] = (tuple(position))
            #Reinitialize position to empty list when both coordinates are found
            position = []
            count = 0
```

```
            continue
        if 'longitude' in line.values() and count is 0:
            position.append(float(line['value']))
            count = count + 1
            continue
        if 'longitude' in line.values() and count is 1:
            position.append(float(line['value']))
            time = float(line['timestamp'])
            coordinates[round(time, 0)] = (tuple(position))
            #Reinitialize position to empty list when both coordinates are found
            position = []
            count = 0
            continue
    #Creating a list of points where we have latitude and longitude values for times when
    #vehicle speed was zero
    points = []
    for time in timestamps.keys():
        for line in coordinates.keys():
            if (time == line):
                points.append(coordinates[line])
    #Plotting those points on a google map
    mymap = pygmaps.maps(points[0][0], points[0][1],17)
    for point in points:
        mymap.addpoint(point[0], point[1], "#FF0000")
    print('Created a plot--> check mystop.html file')
    mymap.draw('./mystop.html')
```



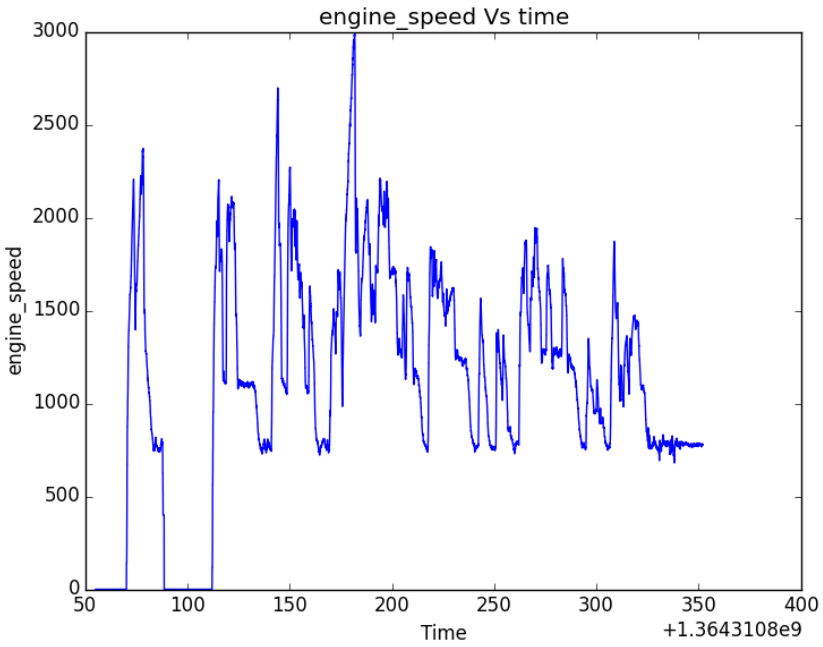

W 104th St
Columbus Ave
Columbus Ave
Ave

Coordinates where Alice stopped

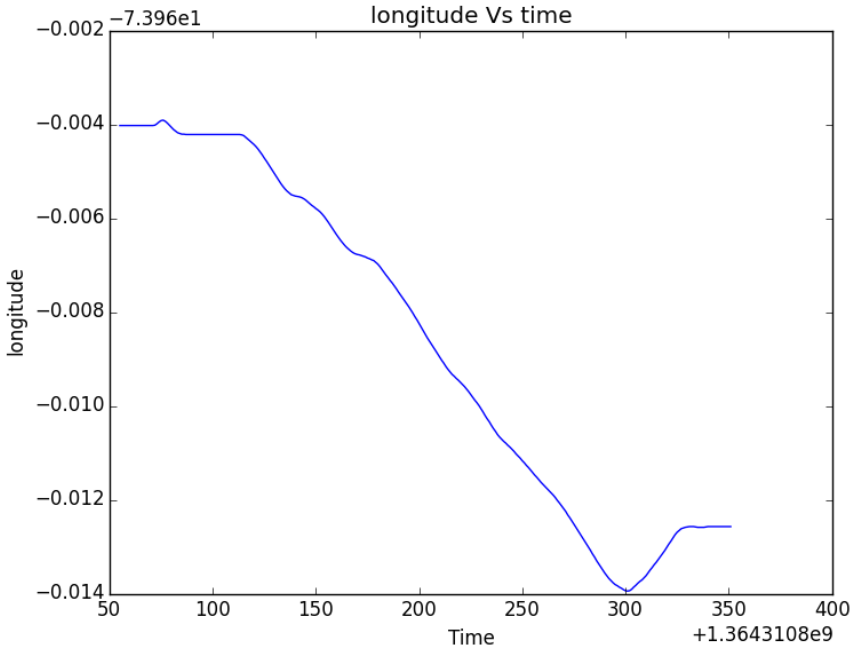
Data Plots

The different signal plots are :

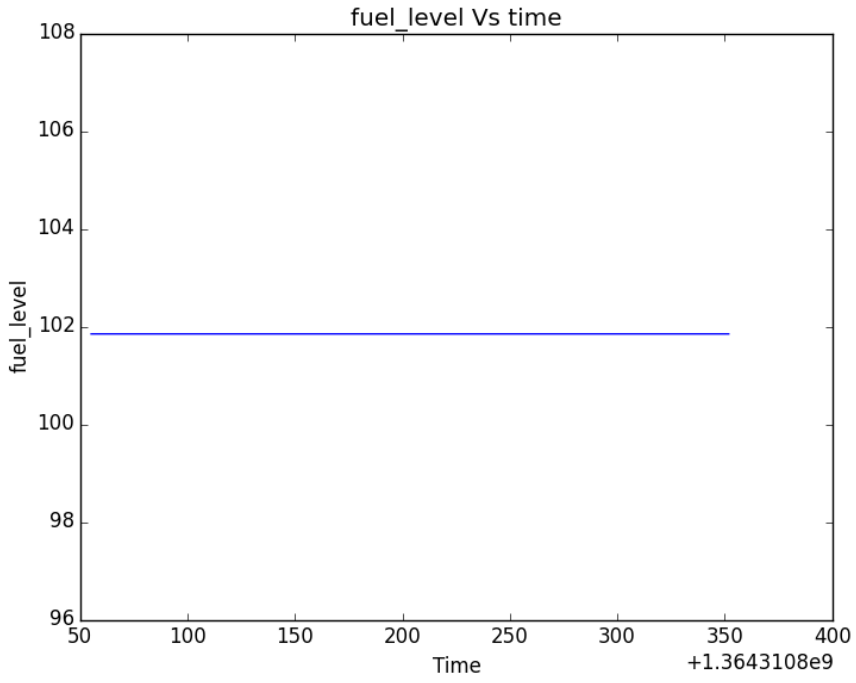
1. Engine speed with time



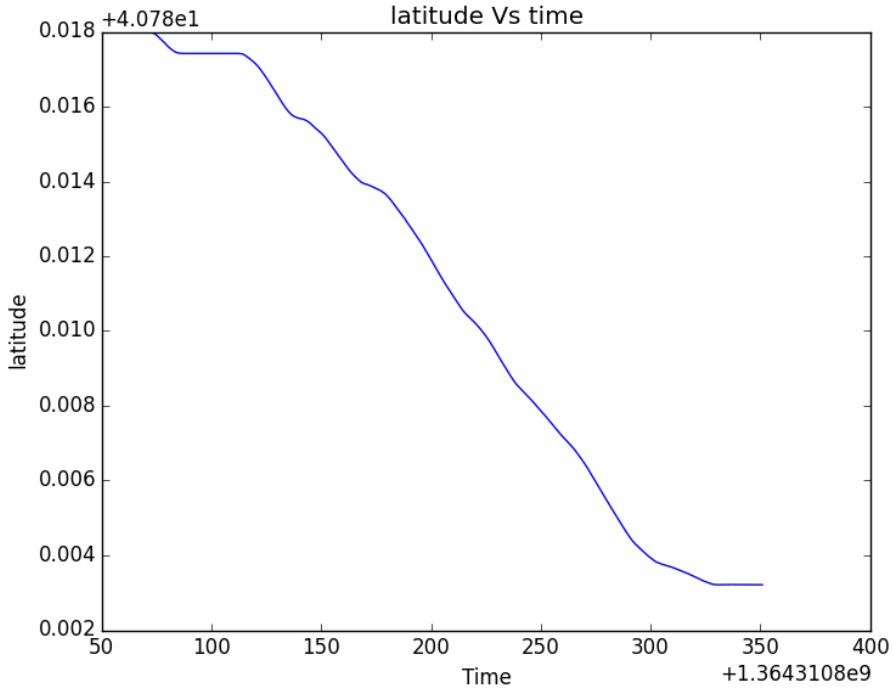
2. Longitude with time



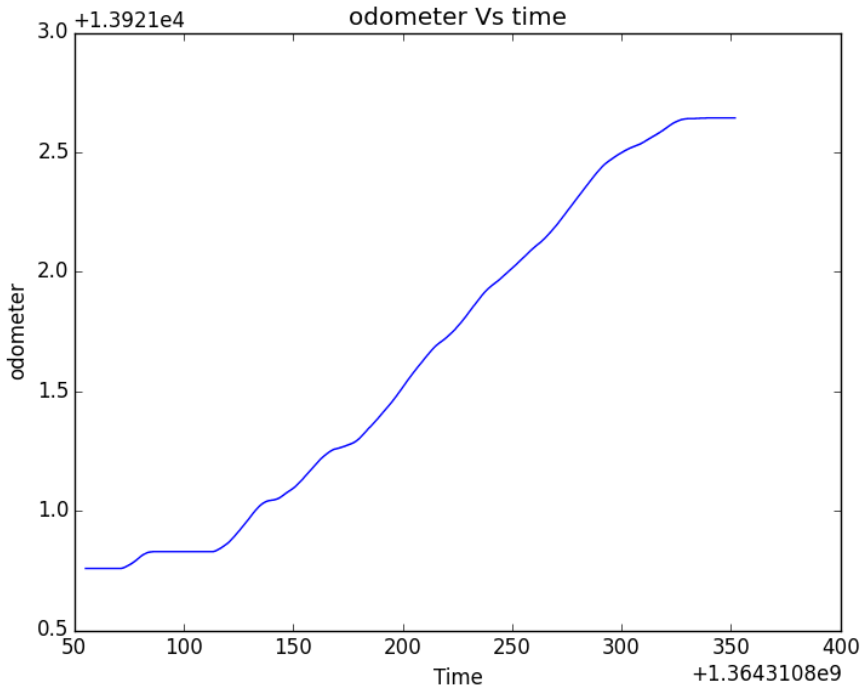
3. Fuel level with time



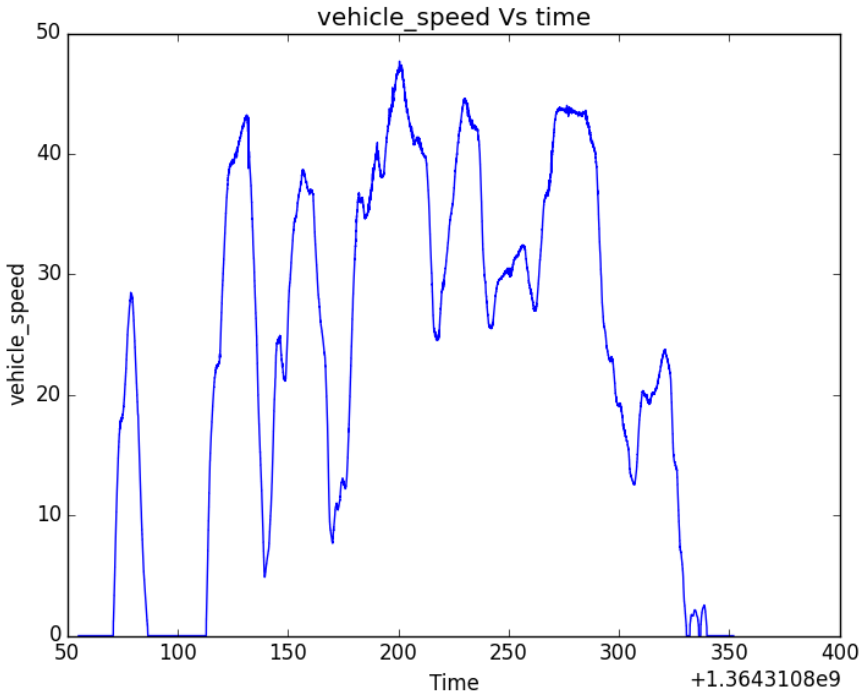
4. Latitude with time



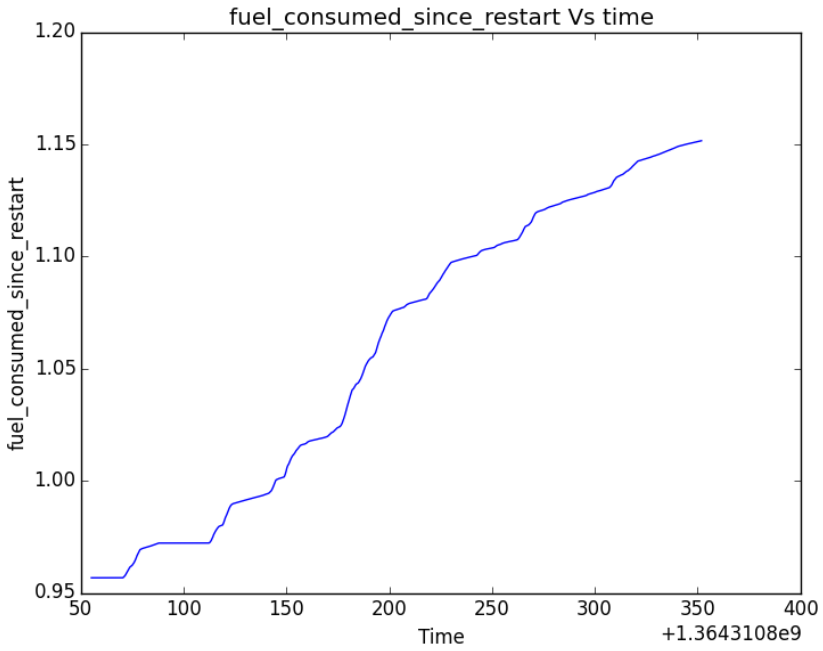
5. Odometer with time



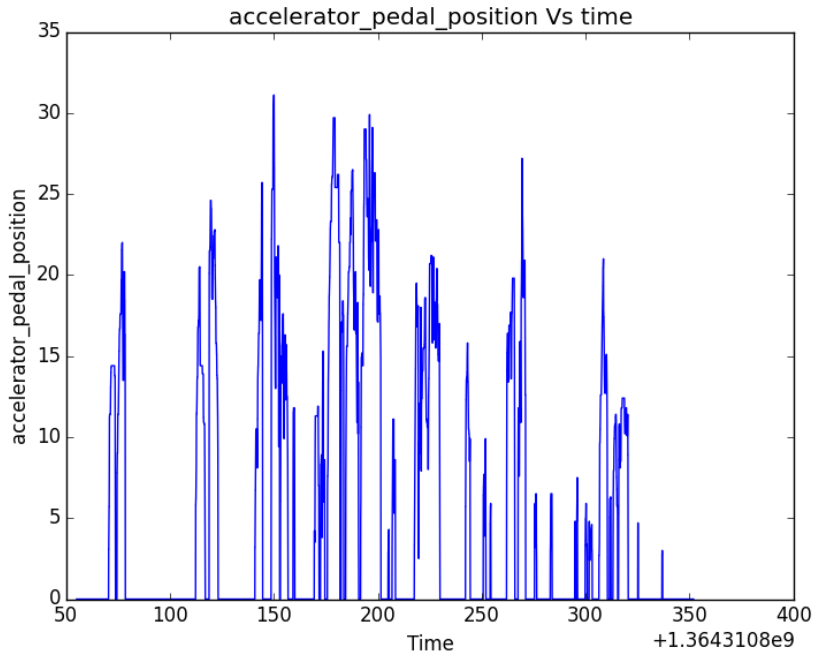
6. Vehicle speed with time



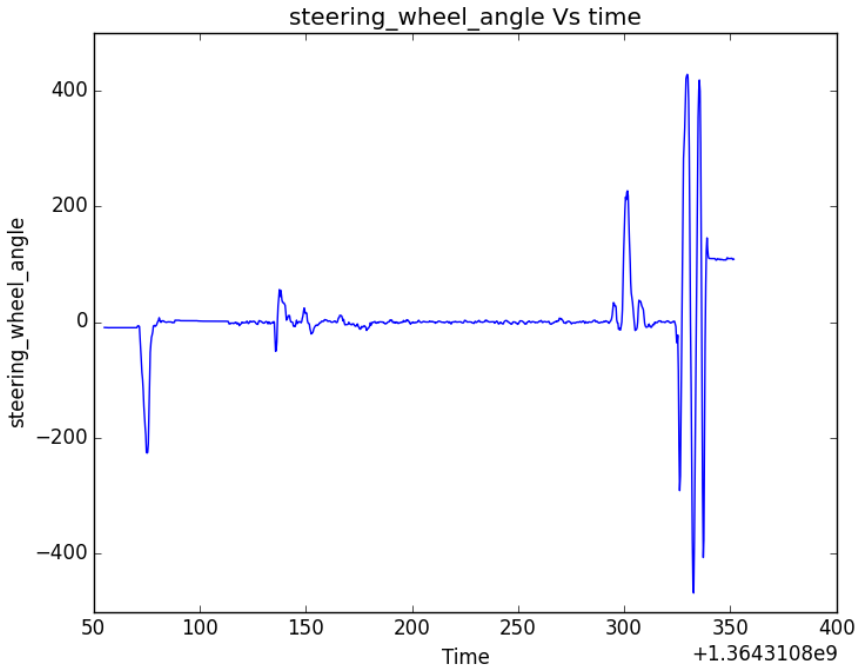
7. Fuel consumed since restart with time



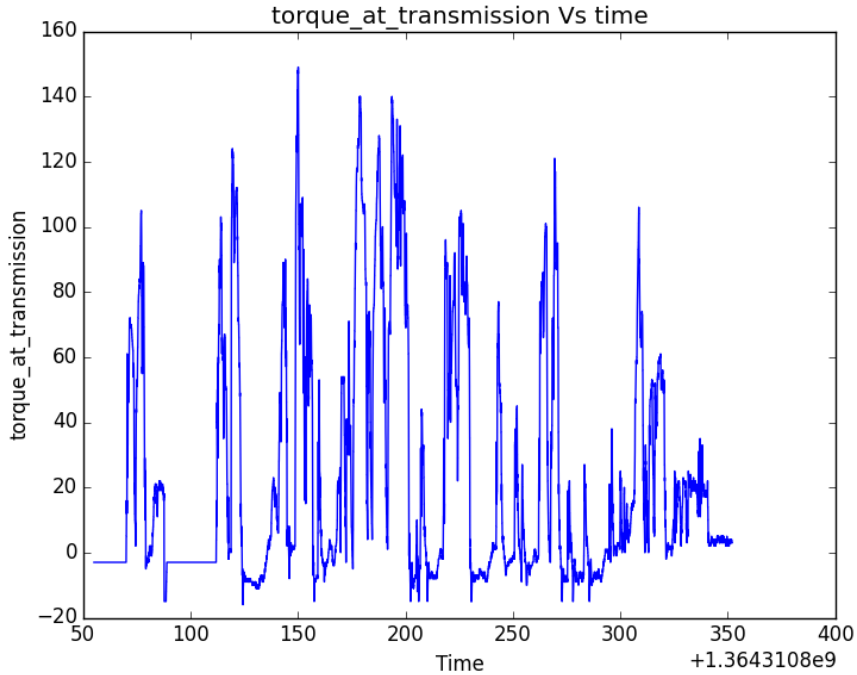
8. Accelerator pedal position with time



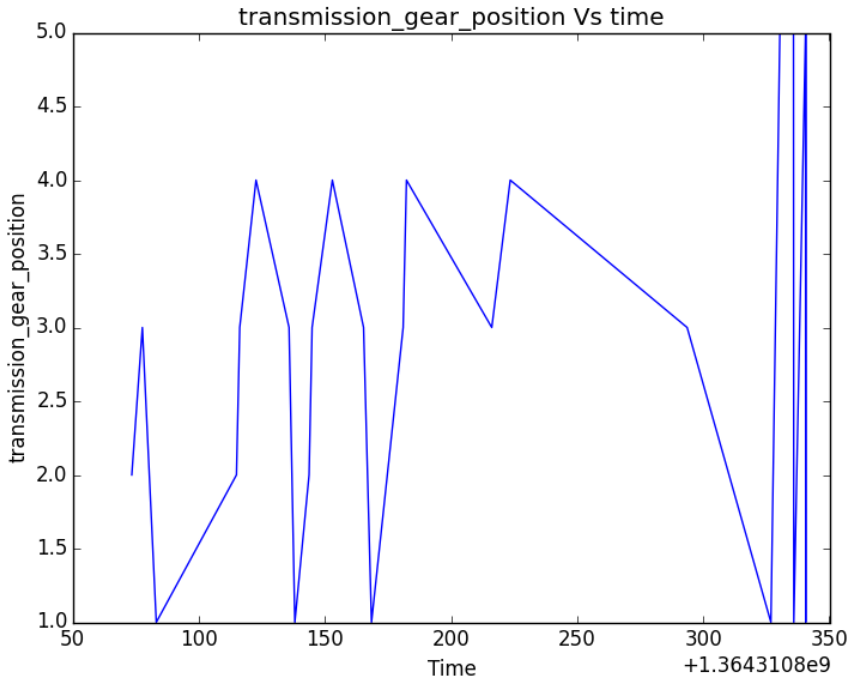
9. Steering wheel angle with time



10. Torque at transmission with time



11. Transmission gear position with time



Observations

a. From the data plots, we can see that the vehicle speed was variable and the car stopped at two places. The car completely stopped, since the speed was zero and engine rpm was zero as well. So, the vehicle was not idle during the trip. The maximum speed was 47.68 mph and the average speed was 22.97 mph. From the google map, we can say that Alice drove from near the Theodore Roosevelt Park to Amsterdam Avenue in Manhattan.

b. The vehicle speed plot shows that the speed of the vehicle was more or less between 20 mph and 50 mph which shows that Alice was driving at good speed. Also the speed changes indicate normal behavior.

   However, the steering angle plot shows sudden changes during the end of the trip.

   Transmission gear position shows sudden changes as well.

   Overall, this indicates that her general driving behavior is good (gradual acceleration and deceleration). But during the end of the trip, she seems to have lost control over her driving.