

Insights Engineer Exercise

@author: Santiago Canas

1. Private Methods

1.1 `def trace_analysis(df_user, subscriber):`

In this method, we go through the different subscribers. Then, the distance between the previous position and the current position are computed. If the distance is higher than a trivial value, we consider it a change in trace.

Then we collect the current position and the previous position which correspond to this change in trace. The difference of the time lapse between them, will be used to calculate the speed of the movement.

Then it will return a data frame called trace which will give us the current position, previous position, time, distance and speed.

```
trace[element] = { 'current [lat/lon]' : current, 'previuos[lat/lon]' : previous,  
                  'time[h]': time, 'distance[Km]': dist, 'Speed[Km/h]': speed }
```

1.2 `def rute_analysis(df_user, subscriber):`

In this method, we go through the different subscribers. Then, the starting position and the ending position are compared. In the method we identify if there are subscriber with the same starting point, the same ending point or even both.

1.3 `def distance(pos1, pos2):`

In this method, we calculate the distance measure in km given a two position with latitude and longitude each one.

1.4 `def distance_analysis(path, non_zero_distance):`

In this method, we calculate if the distance measure in km id the same for different subscribers.

1.5 `def speed_analysis(df_user, subscriber):`

In this method, we go through the different steps of the traces of each subscriber. Then, the speed of the different change of trace will be categorized; as walking user (Speed [Km/h]<= 5.0) , or as a driving user (Speed [Km/h]<= 50.0) any other case will be categorized as other user.

```
return users, { 'walking' : walking_users, 'driving' : driving_users, 'other' : other_users }
```

1.6 `def get_unique(lista)`

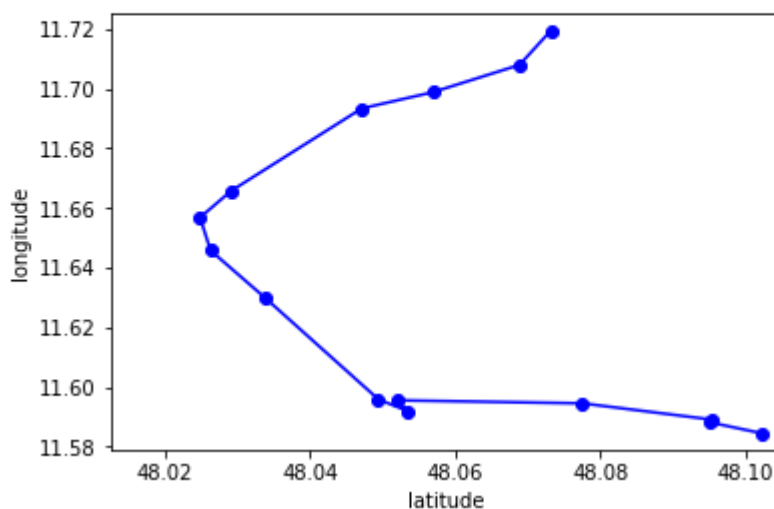
It returns a list with the unique elements of a given list

2. Main Module

In the main module *challenge.py* of the python program, there is a visual representation of the traces of the subscribers. There are two options, the program has an input parameter called TMSI so we can introduce a specific subscriber or we can observe all the subscribers

Let's consider TMSI = 'EB0C64B0'

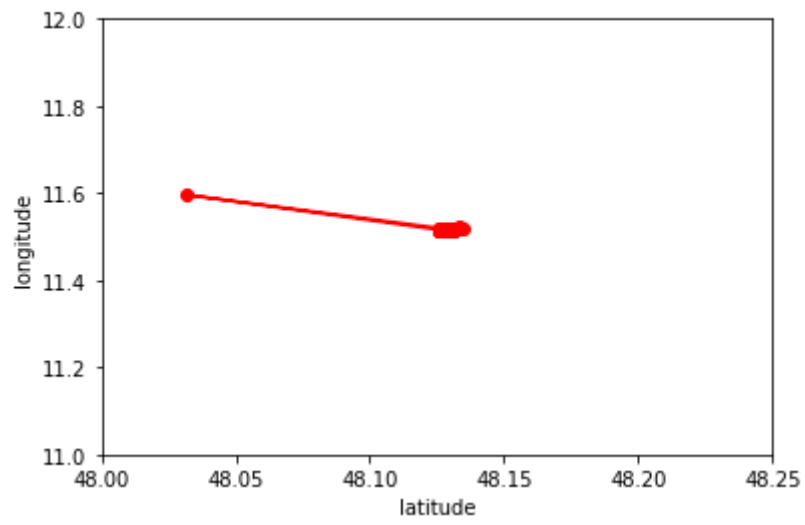
```
Enter your TMSI: EB0C64B0
  Destination  Distance[Km]   Origin  Speed[Km/h]  Time[h]
0   48.102268    10.534023  48.073162  62.578353   0.168333
1   11.584433    10.534023  11.719347  62.578353   0.168333
```



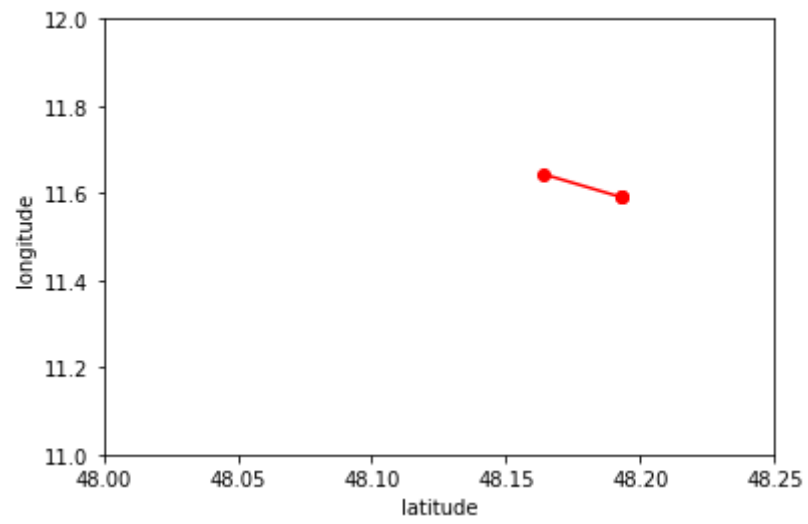
The program will display the path follow by the subscriber corresponding to the TMSI = 'EB0C64B0'. In this case we can appreciate 15 steps in the trace.

In the case we do not want to study the concrete trace of a particular user, we can enter TMSI = '0'. In this case, the program is going to show us all the subscribers that make significant changes in their traces which it is calculated at the method *speed_analysis*.

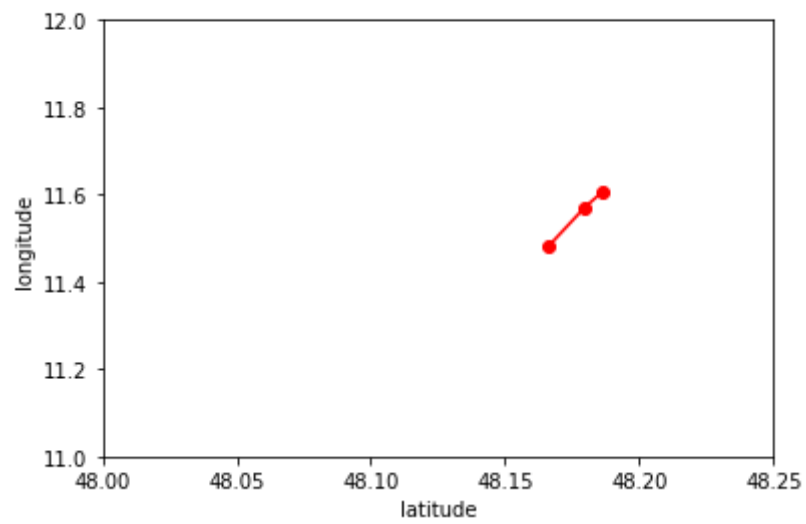
Subscriber: E5049A7F



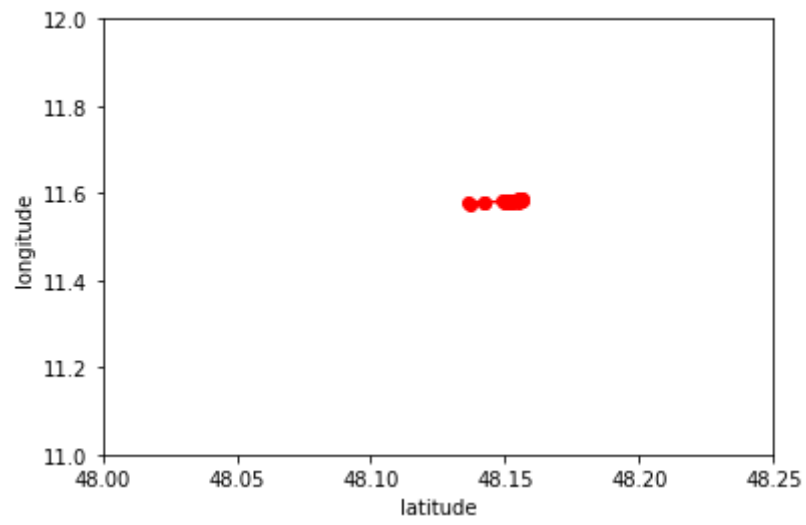
Subscriber: F09805AB



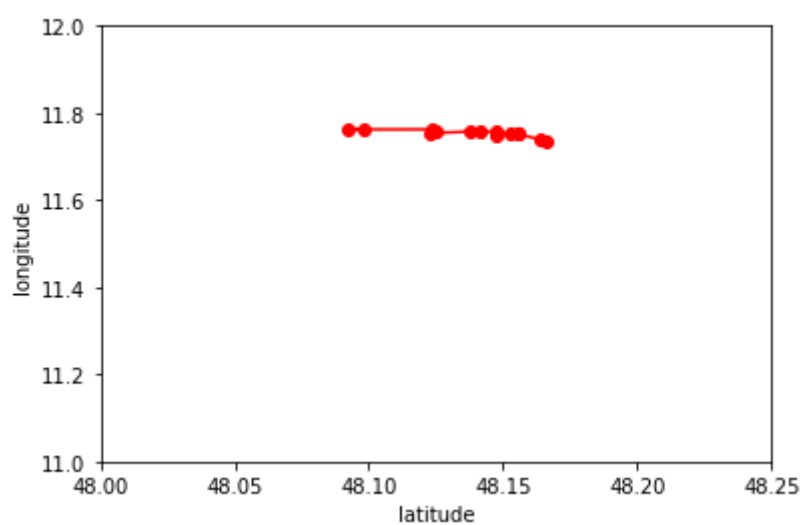
Subscriber: E201302B



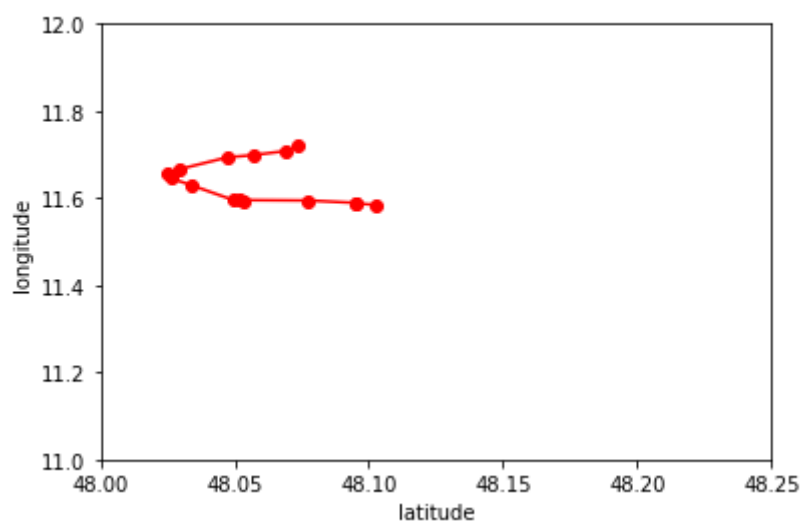
Subscriber: EF0E37A5



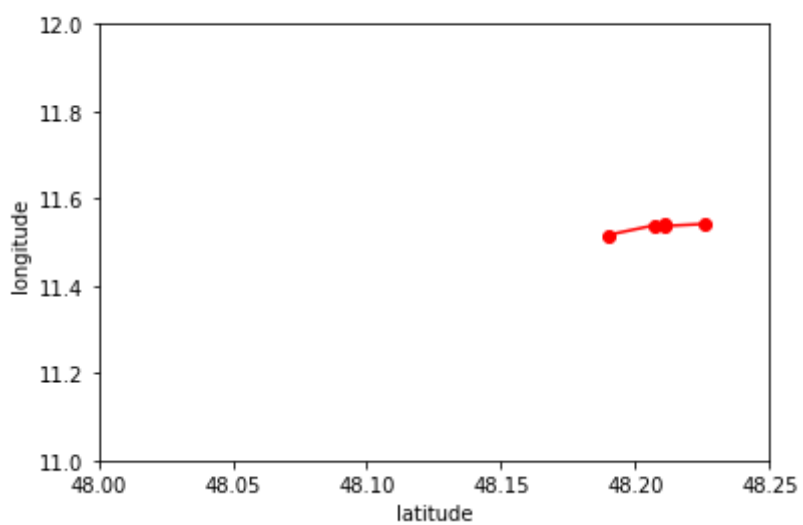
Subscriber: E70B38A4



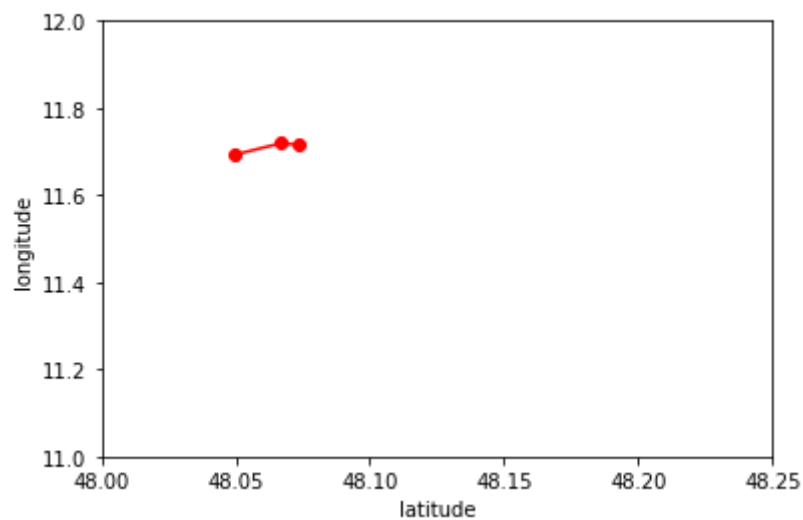
Subscriber: EB0C64B0



Subscriber: F38E5B39



Subscriber: C8954099



3. Conclusion/ Performance

- The program will provided us with the data frame called *user*, which contains info about the different ways of transport and their particular speed in different parts of the trace.
- The program will provide us with an estimation of the users that are driving, walking or other kind of transport. As well as finding the common threads among traces (origin, destiny and both)
- The program will print a specific route of a chosen user. It will give us the origin, the destination, the distance [km], the time [h] to travel the route and finally the speed [km/h]
- The program will print different routes of users. It will give us a visual representation of the subscriber that actually have some relevant change in their trace.
- The program will enable us to see which users have use car, walk or other transport

In other to make easier the illustration the file 'Tracebox_subset_81chains_small.csv' has been applied.

4. How the program works

First, we run the module *challenge.py* it will ask us which kind of data set we want to use; [0] large data set and [1] for small data set.

After that, we will have to enter the TMSI, this parameter can be a specific value like TMSI = 'EB0C64B0' in this case it will only plot the trace of that specific user, or TMSI = 0 which will plot all the traces of the users that have travelled more than 1 km of distance (arbitrary chosen)