QUESTION 2

The following questions were addressed using the dataset including user location information gathered over time from GPS-enabled mobile devices.

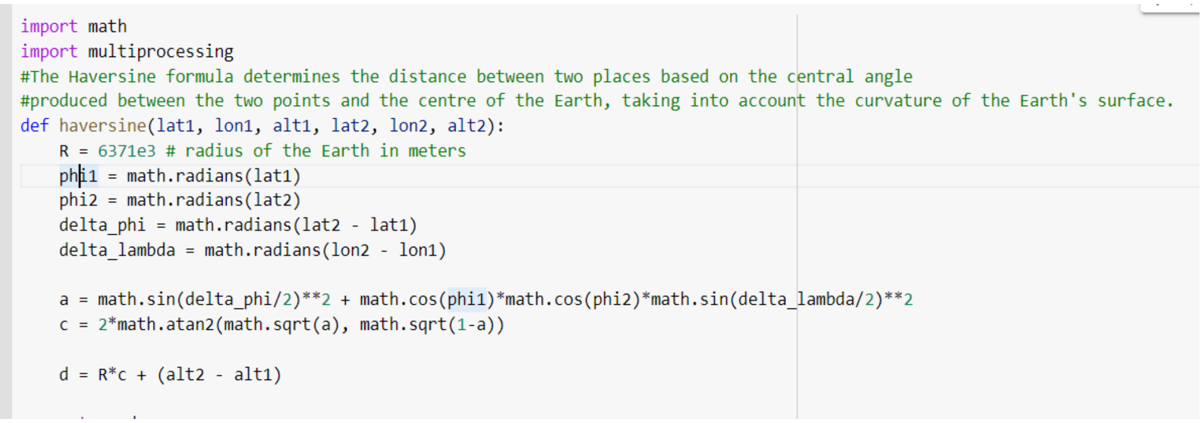
PART 1 WRITE A FUNCTION TO CALCULATE THE TOTAL DISTANCE TRAVELED BY EACH USER IN THE DATASET.

The Haversine formula is used to determine the distance between two places based on the information of latitude, longitude and Altitude.

The length of the dataset is 24876978 thus for the ease of calculation new_df has been created which gets the top 500 rows with each unique 'individual_id' and use multiprocessing library to calculates the total distance traveled by each individual in the DataFrame.

New_df contains 87248 rows.

The result is as follows: https://docs.google.com/document/d/1A7SajI0_VCj3CCCaWRN9sqFksS8TPF-6DLTHYaF7bSg/edit



```
import math
import multiprocessing
#The Haversine formula determines the distance between two places based on the central angle
#produced between the two points and the centre of the Earth, taking into account the curvature of the Earth's surface.
def haversine(lat1, lon1, alt1, lat2, lon2, alt2):
    R = 6371e3 # radius of the Earth in meters
    phi1 = math.radians(lat1)
    phi2 = math.radians(lat2)
    delta_phi = math.radians(lat2 - lat1)
    delta_lambda = math.radians(lon2 - lon1)

    a = math.sin(delta_phi/2)**2 + math.cos(phi1)*math.cos(phi2)*math.sin(delta_lambda/2)**2
    c = 2*math.atan2(math.sqrt(a), math.sqrt(1-a))

    d = R*c + (alt2 - alt1)
```

PART 2 WRITE A FUNCTION TO EXTRACT AND VISUALIZE THE SPATIAL AND TEMPORAL HOTSPOTS OF BEJING CITY.

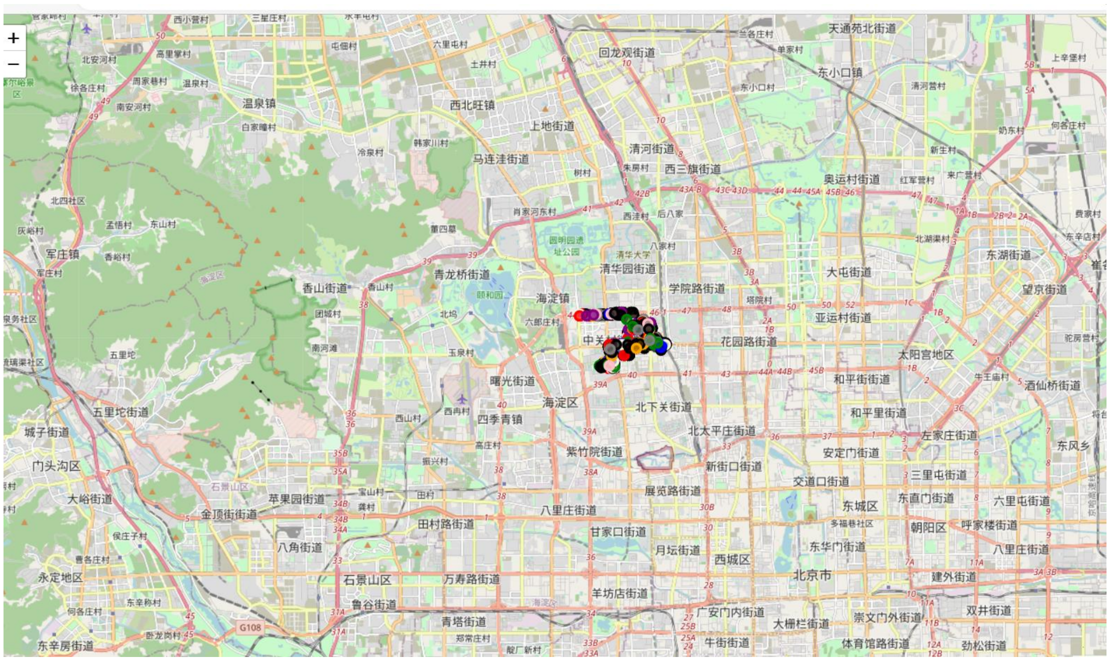
In order to extract and visualize the spatial and temporal hotspots i have used 2 methods:

- I have used DBSCAN because it can discover clusters that are both geographically and Since the dataset is quite large to visualize it had been grouped according to individual_id and randomly 7000 dataset has been selected by the provided groups. So that it could shows the trajectory of individual_id 1 over time, with latitude, longitude, altitude, date, and time information for each location. It also includes trajectory_id, which may be used to distinguish different trajectories for the same individual. The individual_id's Silhouette score here quantifies how similar an object is to its own cluster in comparison to other clusters. A score of 1 indicates that the object is highly similar to its own cluster and quite different from other clusters, whereas a score of -1 suggests the reverse. The scale goes from -1 to 1.

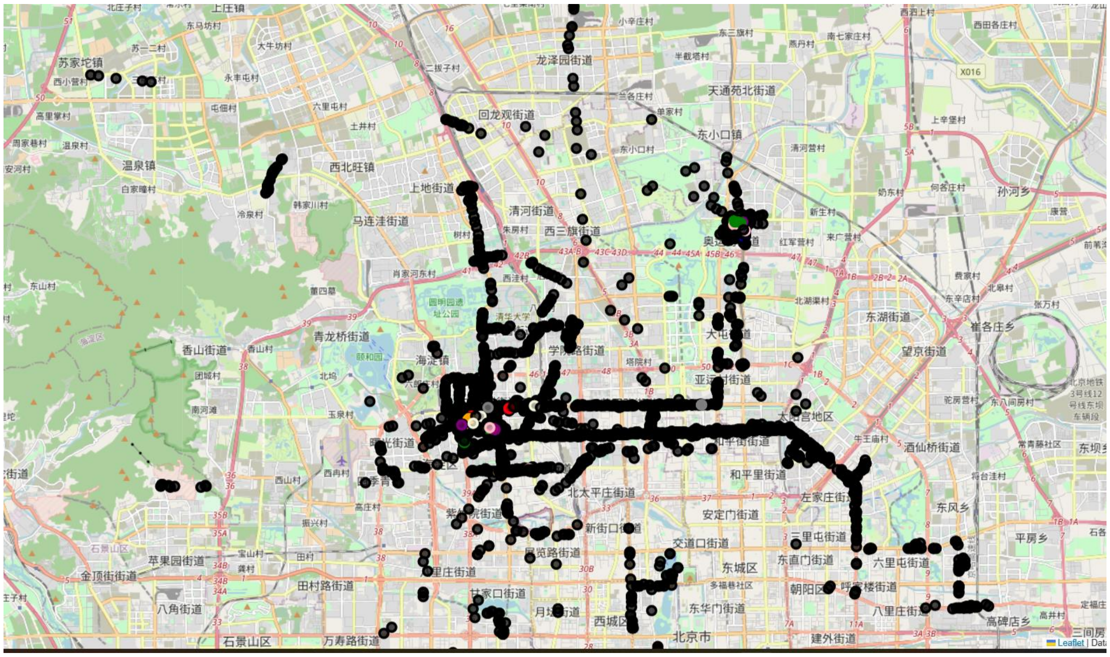
  Silhouette score: https://docs.google.com/document/d/1BNmDfNzFQO9sX4o60XSj7ti8TIIvCzQoe_RZD1M6-LA/edit

  I am attaching a few of the map clusters:

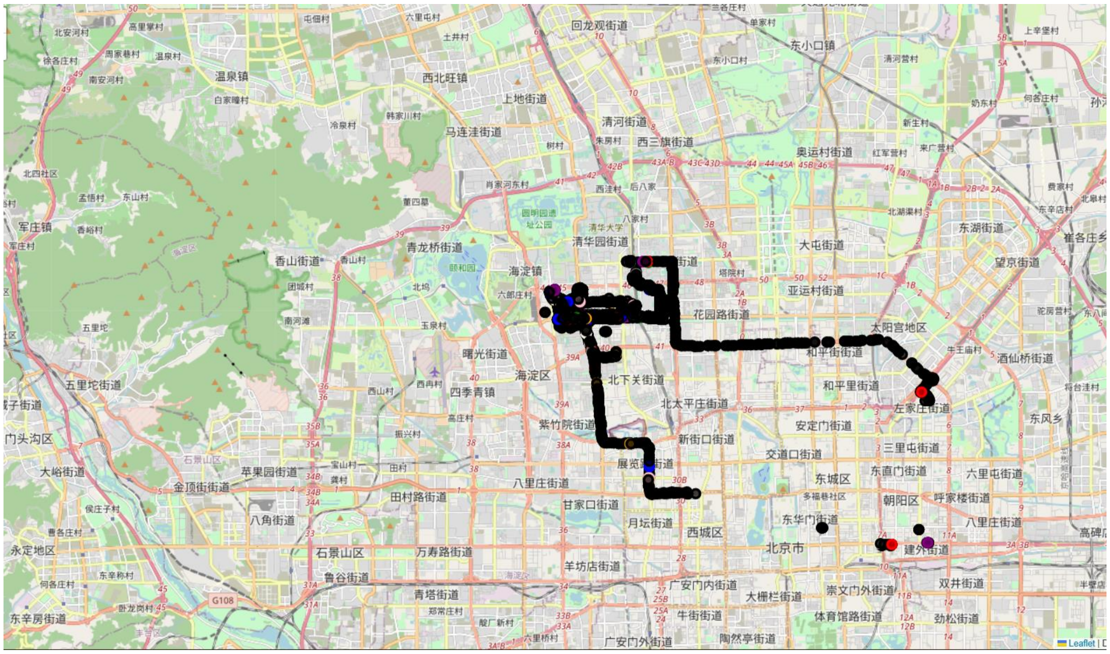
Individual_id= 98



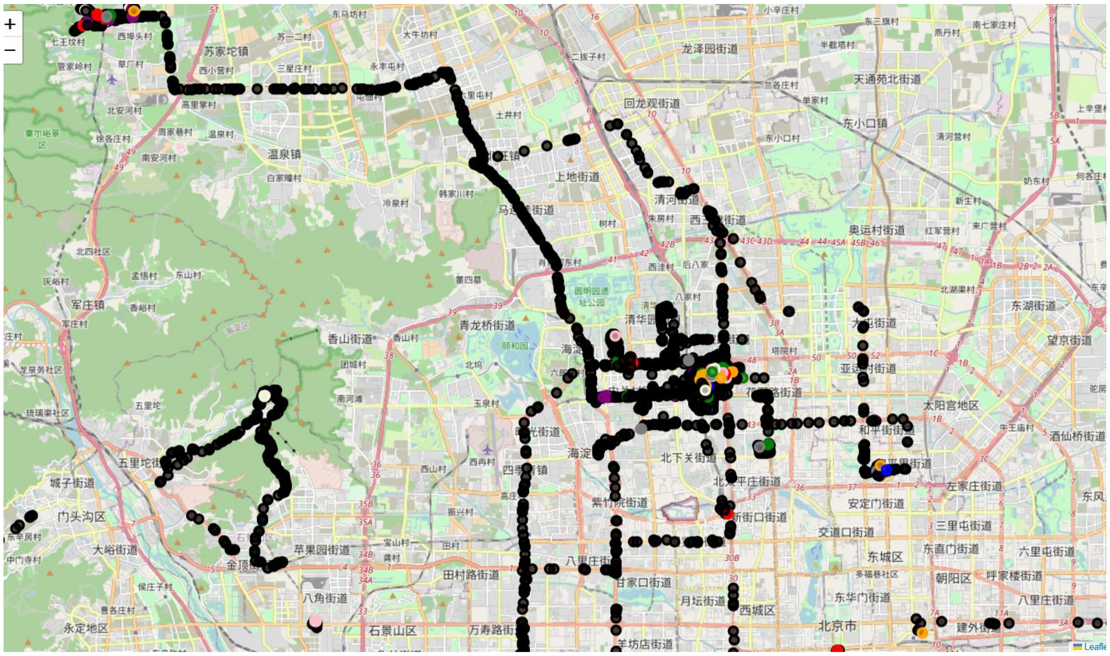
Individual_id= 93



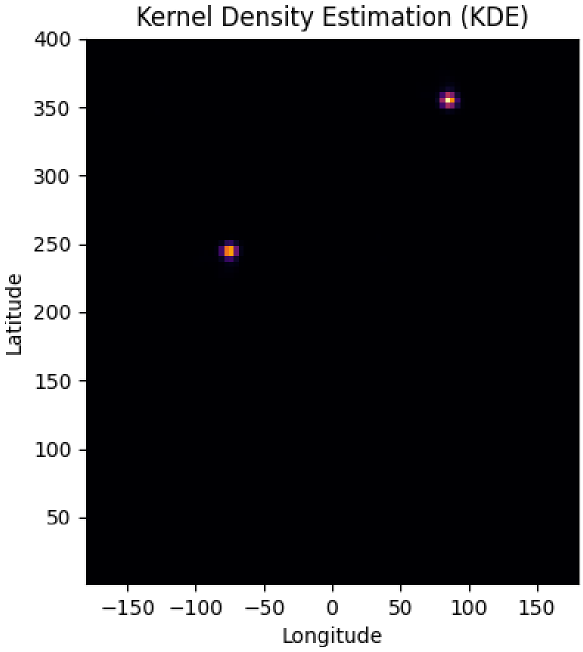
Individual_id= 103



Individual_id=8



- I have used Kernel Density Estimation (KDE) In order to identify locations with high or low densities of events depending on their geographic coordinates. The generated density map might highlight regions with high event densities, which can point to a spatial hotspot. This can be helpful in determining which parts of a city or region are more likely to experience particular occurrences (such as crimes, accidents, etc.). Random sampling is applied to randomly select 300000 dataset and then the plot is being created.



PART 3 IMAGINE THAT YOU HAVE ACCESS TO A GPS-TRACKING DATASET CONTAINING THE TRAJECTORIES OF THOUSANDS OF INDIVIDUALS OVER AN EXTENDED PERIOD OF TIME.

I would create a Travel Time Distribution Determination system that could be applied in emergency circumstances, resulting in quicker reaction times and better disaster management, if I had access to a GPS-tracking dataset comprising the trajectories of thousands of people.

Important steps to be taken in preprocessing of dataset for such a project would be Segment the trajectories of emergency vehicles into individual trips based on the start and end points of each trip to avoid heterogeneity in the traffic streams . The travel time distribution determination system can be given more context by supplementing the GPS-tracking data with additional data sources like traffic cameras, road sensors, and detailed interfering weather conditions from websites that provide historical meteorological data. This will increase the system's accuracy. Data cleaning would involve removing outliers caused by air interference, multipath signals, and signal loss.

Some test which i would suggest for checking the goodness-of-fit would be:

- Chi-square test: This test is frequently used to examine categorical data, including collision and traffic volume statistics.
- Pearson correlation test: This test can be used to determine the magnitude and direction of the linear relationship between two continuous variables, such as traffic flow and speed.
- ANOVA (Analysis of Variance): This test may be used to compare the means of continuous data sets with more than two groups, such as data on traffic flow at various times of day or in various places.
- Regression analysis: The link between traffic flow and several predictor factors, such as the time of day, the weather, and road features, may be modelled using regression analysis.

It will be possible to get insight into the normal journey times for emergency vehicles at various times of the day and in various traffic circumstances by modelling the distribution of travel times using statistical techniques like the probability density function (PDF) or cumulative distribution function (CDF). Emergency personnel may estimate their trip time in real-time to help them negotiate traffic and get to the impacted regions more swiftly.