Data Communication and Networking Assignment-2
On
Finding Location Trail of a Device

Task :

Find the location trail of your phone. You can use any tool/ programming language etc. to find the location trail.

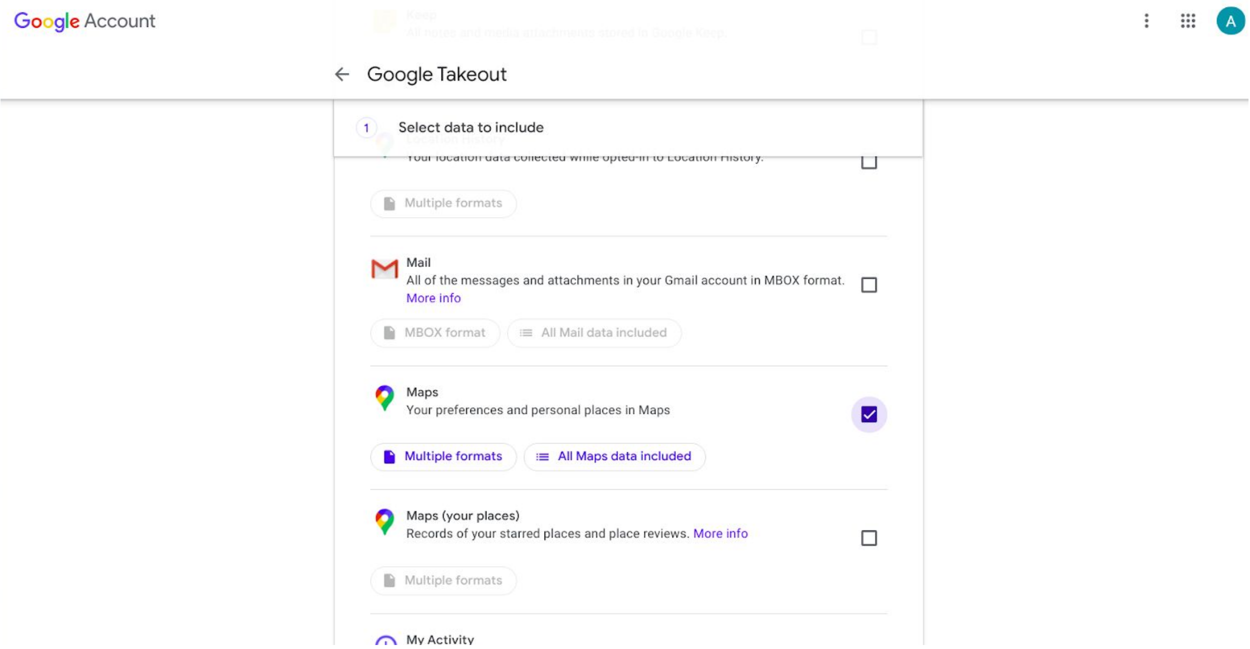
Solution:

Phase 1 : Collection of location trail data

GPS in our devices do not keep log of the locations we have been to. To store this information we have to use third party applications. One such application that stores this information for us is Google Maps.

Google Maps stores all of our location data to its servers continuously, given Google Maps has our location rights (which it gets when we create a new google account). And by using **Google Takeout** we can get access to this data (**Google Takeout** is a project by the *Google Data Liberation Front* that allows users of Google products, such as Maps , YouTube and Gmail to export their data to a downloadable archive file).

I downloaded the location history data of my device from Google Takeout which was in JSON format. It had the location history data saved since December, 2015.



1 Select data to include

Your location data collected while opted-in to Location History. □

Multiple formats

Mail
All of the messages and attachments in your Gmail account in MBOX format. □
More info

MBOX format | All Mail data included

Maps
Your preferences and personal places in Maps ☑

Multiple formats | All Maps data included

Maps (your places)
Records of your starred places and place reviews. More info □

Multiple formats

My Activity

Phase 2 : Loading and Preprocessing of the location history data

Python programming language was used to load and preprocess the location history data.

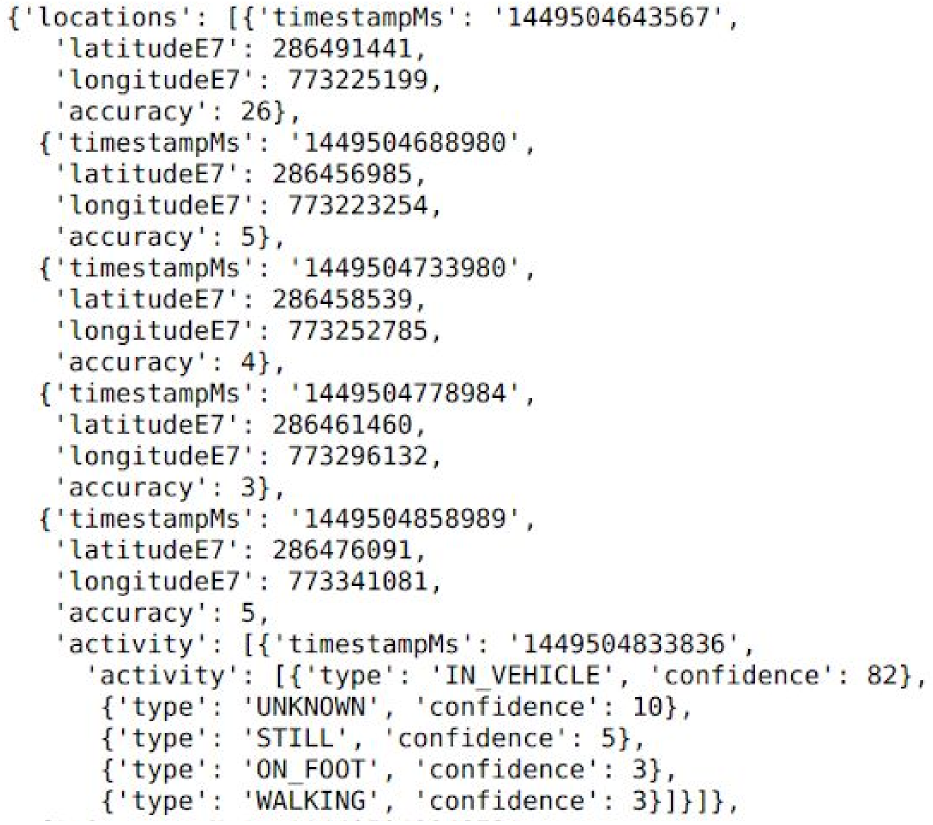
Loading data :

Downloaded location history data was in JSON format which was then converted to a python dictionary object.

Each location data item has the following fields:-

- timestampMs :- Timestamp information
- latitudeE7 :- Latitude of a location
- longitudeE7 :- Longitude of a location
- accuracy :- Accuracy information for a given location coordinates
- activity :- Activity log for activities in a location

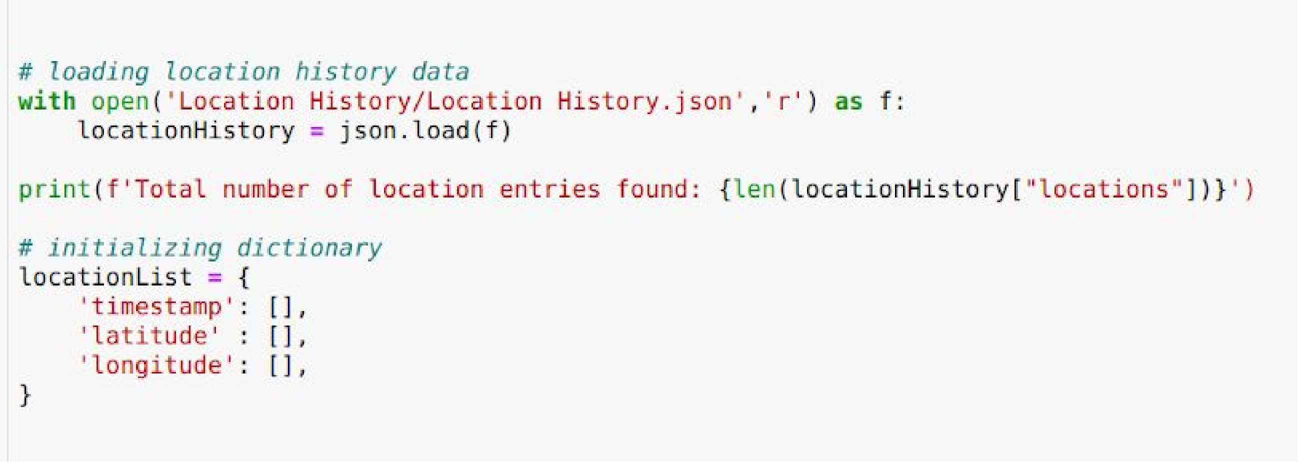
Downloaded data when converted to python's dictionary object:-



```
{'locations': [{'timestampMs': '1449504643567',
   'latitudeE7': 286491441,
   'longitudeE7': 773225199,
   'accuracy': 26},
  {'timestampMs': '1449504688980',
   'latitudeE7': 286456985,
   'longitudeE7': 773223254,
   'accuracy': 5},
  {'timestampMs': '1449504733980',
   'latitudeE7': 286458539,
   'longitudeE7': 773252785,
   'accuracy': 4},
  {'timestampMs': '1449504778984',
   'latitudeE7': 286461460,
   'longitudeE7': 773296132,
   'accuracy': 3},
  {'timestampMs': '1449504858989',
   'latitudeE7': 286476091,
   'longitudeE7': 773341081,
   'accuracy': 5,
   'activity': [{'timestampMs': '1449504833836',
     'activity': [{'type': 'IN_VEHICLE', 'confidence': 82},
      {'type': 'UNKNOWN', 'confidence': 10},
      {'type': 'STILL', 'confidence': 5},
      {'type': 'ON_FOOT', 'confidence': 3},
      {'type': 'WALKING', 'confidence': 3}]}]},
```

For this assignment only the timestampMs, latitude and longitude were needed. So I created another dictionary object to save only the needed information.

Code snippet for loading data:-



```
# loading location history data
with open('Location History/Location History.json','r') as f:
    locationHistory = json.load(f)

print(f'Total number of location entries found: {len(locationHistory["locations"])}')

# initializing dictionary
locationList = {
    'timestamp': [],
    'latitude' : [],
    'longitude': [],
}
```

Preprocessing of the data:

1. All the locations which had their accuracy above 1000 were discarded.

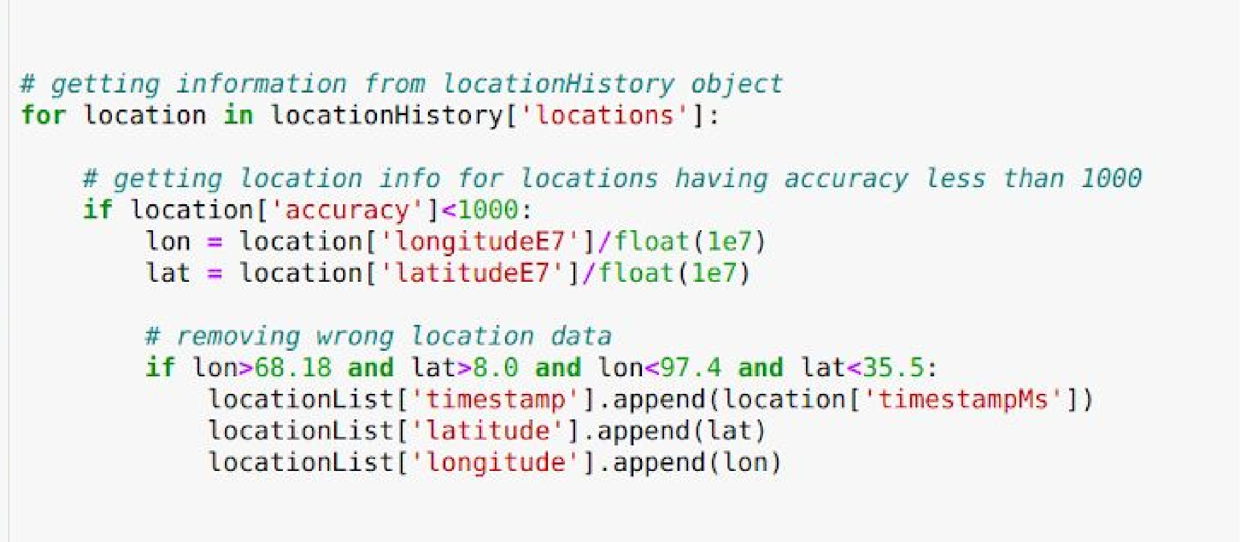
   Note: The accuracy is the radius Google Timeline application gives when registering a position. It is expressed in meters, and it is represented as the radius of the circle around the provided location.

2. Google location data is not 100% accurate and while eyeballing the data I encountered some inconsistencies in data such as latitude and longitude suggesting a location where the device has never been (foreign countries). To tackle this problem I extracted the bounding box coordinates of India from the internet (because the device has never been outside of India) and only saved those locations, which were inside the bounding box, to the new dictionary object.

3. Latitude and Longitude were divided by 10000000.

4. New field 'datetime' in the new dictionary object was created to show date and time in more understandable format (using timestamp information)

Code snippet for preprocessing:-



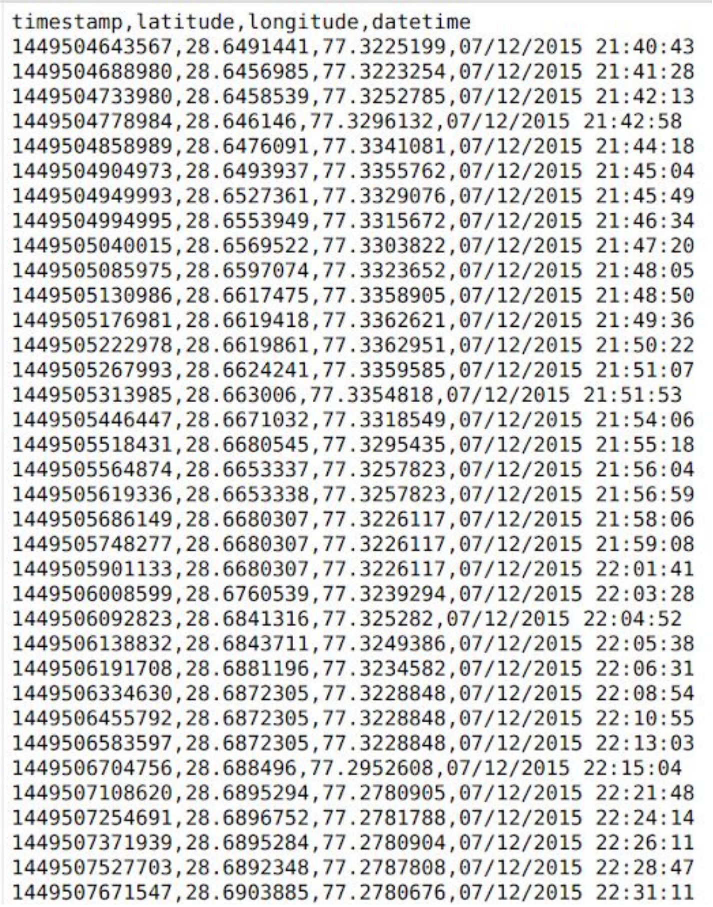
```python
# getting information from locationHistory object
for location in locationHistory['locations']:

    # getting location info for locations having accuracy less than 1000
    if location['accuracy']<1000:
        lon = location['longitudeE7']/float(1e7)
        lat = location['latitudeE7']/float(1e7)

        # removing wrong location data
        if lon>68.18 and lat>8.0 and lon<97.4 and lat<35.5:
            locationList['timestamp'].append(location['timestampMs'])
            locationList['latitude'].append(lat)
            locationList['longitude'].append(lon)
```

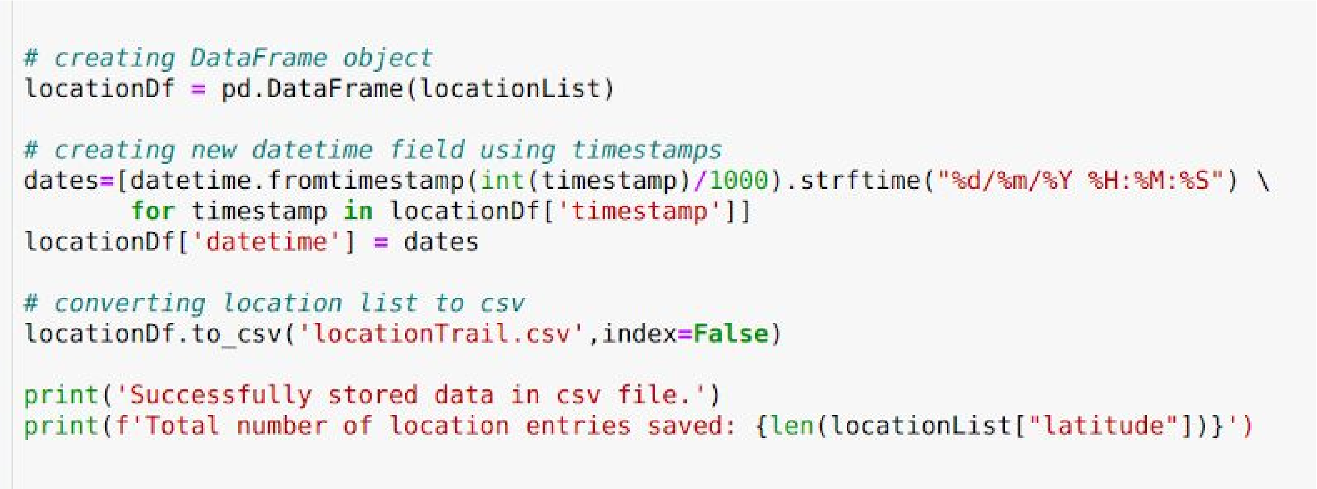
Saving all location trail data as a CSV file:

After the preprocessing step i got a new python dictionary object containing information about around **750000** location data items. This dictionary object was then converted and saved as CSV format.



```
timestamp,latitude,longitude,datetime
1449504643567,28.6491441,77.3225199,07/12/2015 21:40:43
1449504688980,28.6456985,77.3223254,07/12/2015 21:41:28
1449504733980,28.6458539,77.3252785,07/12/2015 21:42:13
1449504778984,28.646146,77.3296132,07/12/2015 21:42:58
1449504858989,28.6476091,77.3341081,07/12/2015 21:44:18
1449504904973,28.6493937,77.3355762,07/12/2015 21:45:04
1449504949993,28.6527361,77.3329076,07/12/2015 21:45:49
1449504994995,28.6553949,77.3315672,07/12/2015 21:46:34
1449505040015,28.6569522,77.3303822,07/12/2015 21:47:20
1449505085975,28.6597074,77.3323652,07/12/2015 21:48:05
1449505130986,28.6617475,77.3358905,07/12/2015 21:48:50
1449505176981,28.6619418,77.3362621,07/12/2015 21:49:36
1449505222978,28.6619861,77.3362951,07/12/2015 21:50:22
1449505267993,28.6624241,77.3359585,07/12/2015 21:51:07
1449505313985,28.663006,77.3354818,07/12/2015 21:51:53
1449505446447,28.6671032,77.3318549,07/12/2015 21:54:06
1449505518431,28.6680545,77.3295435,07/12/2015 21:55:18
1449505564874,28.6653337,77.3257823,07/12/2015 21:56:04
1449505619336,28.6653338,77.3257823,07/12/2015 21:56:59
1449505686149,28.6680307,77.3226117,07/12/2015 21:58:06
1449505748277,28.6680307,77.3226117,07/12/2015 21:59:08
1449505901133,28.6680307,77.3226117,07/12/2015 22:01:41
1449506008599,28.6760539,77.3239294,07/12/2015 22:03:28
1449506092823,28.6841316,77.325282,07/12/2015 22:04:52
1449506138832,28.6843711,77.3249386,07/12/2015 22:05:38
1449506191708,28.6881196,77.3234582,07/12/2015 22:06:31
1449506334630,28.6872305,77.3228848,07/12/2015 22:08:54
1449506455792,28.6872305,77.3228848,07/12/2015 22:10:55
1449506583597,28.6872305,77.3228848,07/12/2015 22:13:03
1449506704756,28.688496,77.2952608,07/12/2015 22:15:04
1449507108620,28.6895294,77.2780905,07/12/2015 22:21:48
1449507254691,28.6896752,77.2781788,07/12/2015 22:24:14
1449507371939,28.6895284,77.2780904,07/12/2015 22:26:11
1449507527703,28.6892348,77.2787808,07/12/2015 22:28:47
1449507671547,28.6903885,77.2780676,07/12/2015 22:31:11
```

Code snippet for saving data as CSV file:

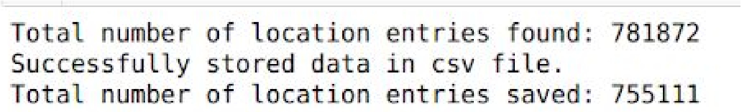


```
# creating DataFrame object
locationDf = pd.DataFrame(locationList)

# creating new datetime field using timestamps
dates=[datetime.fromtimestamp(int(timestamp)/1000).strftime("%d/%m/%Y %H:%M:%S") \
       for timestamp in locationDf['timestamp']]
locationDf['datetime'] = dates

# converting location list to csv
locationDf.to_csv('locationTrail.csv',index=False)

print('Successfully stored data in csv file.')
print(f'Total number of location entries saved: {len(locationList["latitude"])}')
```

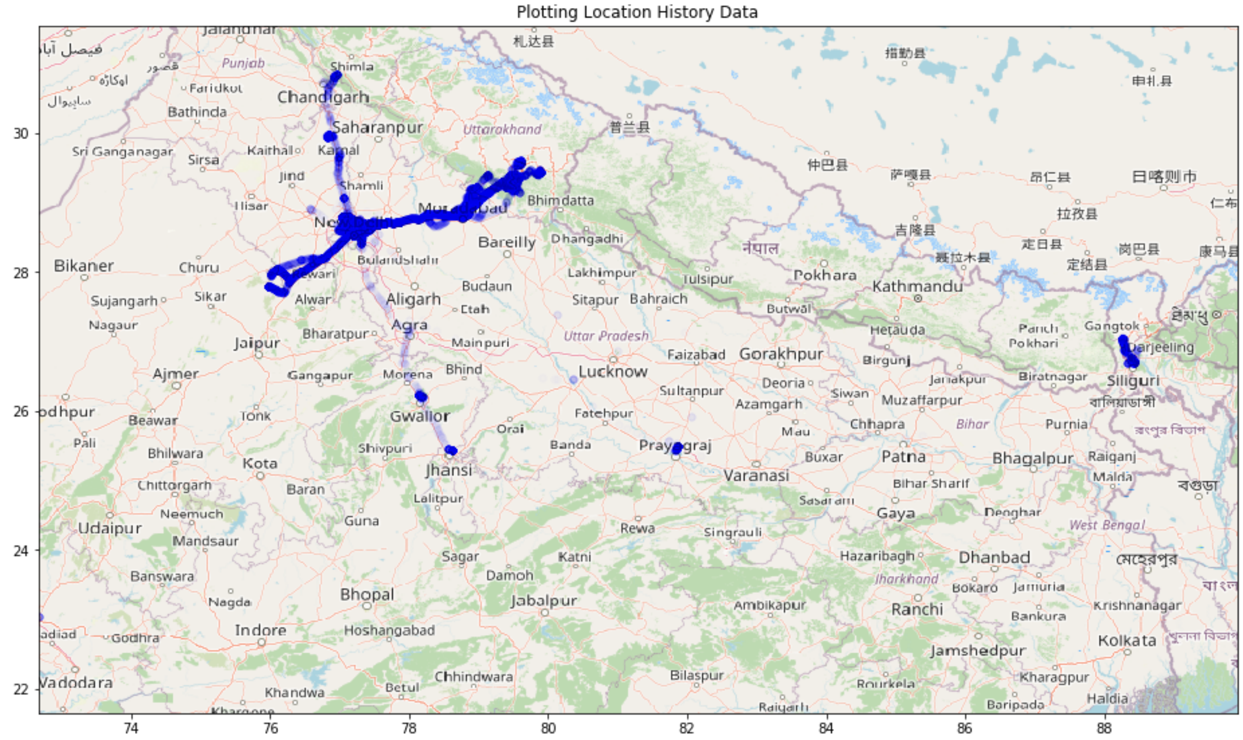


```
Total number of location entries found: 781872
Successfully stored data in csv file.
Total number of location entries saved: 755111
```

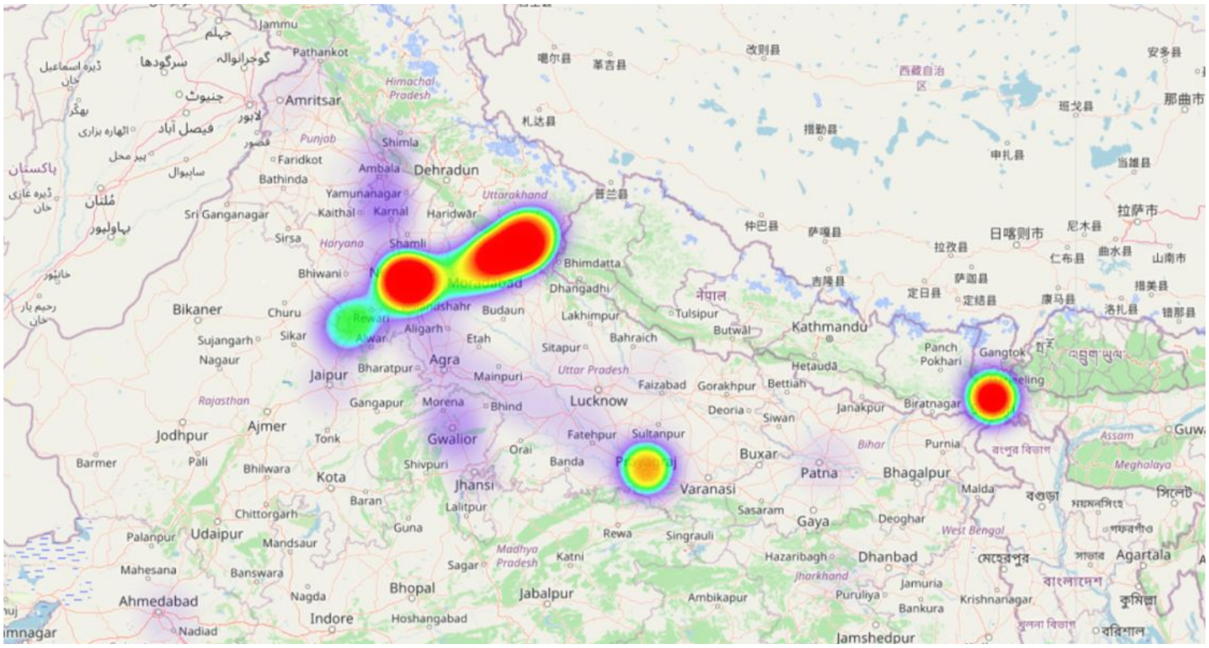
Phase 3 : Plotting data

For better visualization of the location data I used Matplotlib library to plot the location trail on a map.

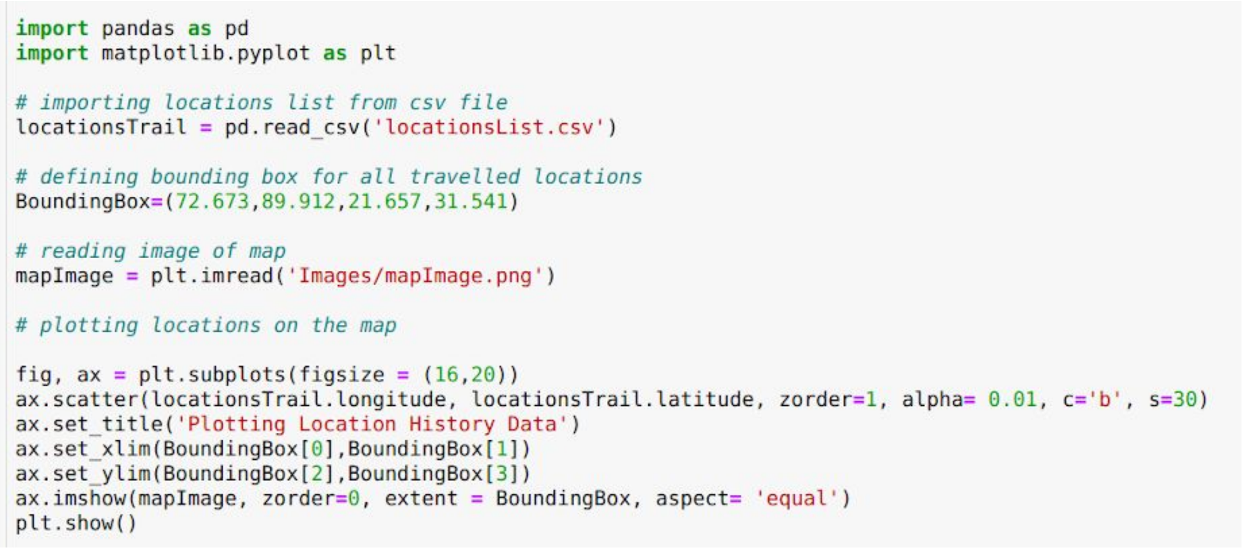
Here is the map plot:



Location Heat Map:



Code snippet for plotting location trail on map:



```
import pandas as pd
import matplotlib.pyplot as plt

# importing locations list from csv file
locationsTrail = pd.read_csv('locationsList.csv')

# defining bounding box for all travelled locations
BoundingBox=(72.673,89.912,21.657,31.541)

# reading image of map
mapImage = plt.imread('Images/mapImage.png')

# plotting locations on the map

fig, ax = plt.subplots(figsize = (16,20))
ax.scatter(locationsTrail.longitude, locationsTrail.latitude, zorder=1, alpha= 0.01, c='b', s=30)
ax.set_title('Plotting Location History Data')
ax.set_xlim(BoundingBox[0],BoundingBox[1])
ax.set_ylim(BoundingBox[2],BoundingBox[3])
ax.imshow(mapImage, zorder=0, extent = BoundingBox, aspect= 'equal')
plt.show()
```

Submitted by: Anurag Joshi
Roll number: 07