MINI PROJECT

Contact Tracing Data Visualization and Analysis Using CNL Model

By:

Megha.S :124CS22027

Pooja.B :124CS22034

Premalatha.K :124CS22041

Thrupthi.P.M :124CS22062

To:

Madam Vandana Gowri

Abstract:

Contact tracing data visualization and analysis are crucial in public health for tracking and understanding the spread of infectious diseases. The primary goal is to identify and map interactions between individuals to determine how diseases spread through communities. In real-world applications, such as during the COVID-19 pandemic, contact tracing helps public health officials monitor outbreaks, implement containment measures, and prevent further transmission.

This project focuses on visualizing and analyzing contact tracing data using Python. By leveraging spatial and temporal data, we can create visual representations to better understand movement patterns and interactions. The provided code demonstrates how to load contact tracing data, visualize it on a map, and generate time-based heatmaps to illustrate density and movement over time. This approach can inform decision-making and enhance the effectiveness of public health interventions.

By identifying high-contact areas and peak times, health officials can allocate resources more effectively, such as deploying testing stations or increasing healthcare support in hotspots. Detailed analysis of contact patterns supports strategic decision-making, such as designing targeted lockdowns or implementing localized health interventions to prevent outbreaks.

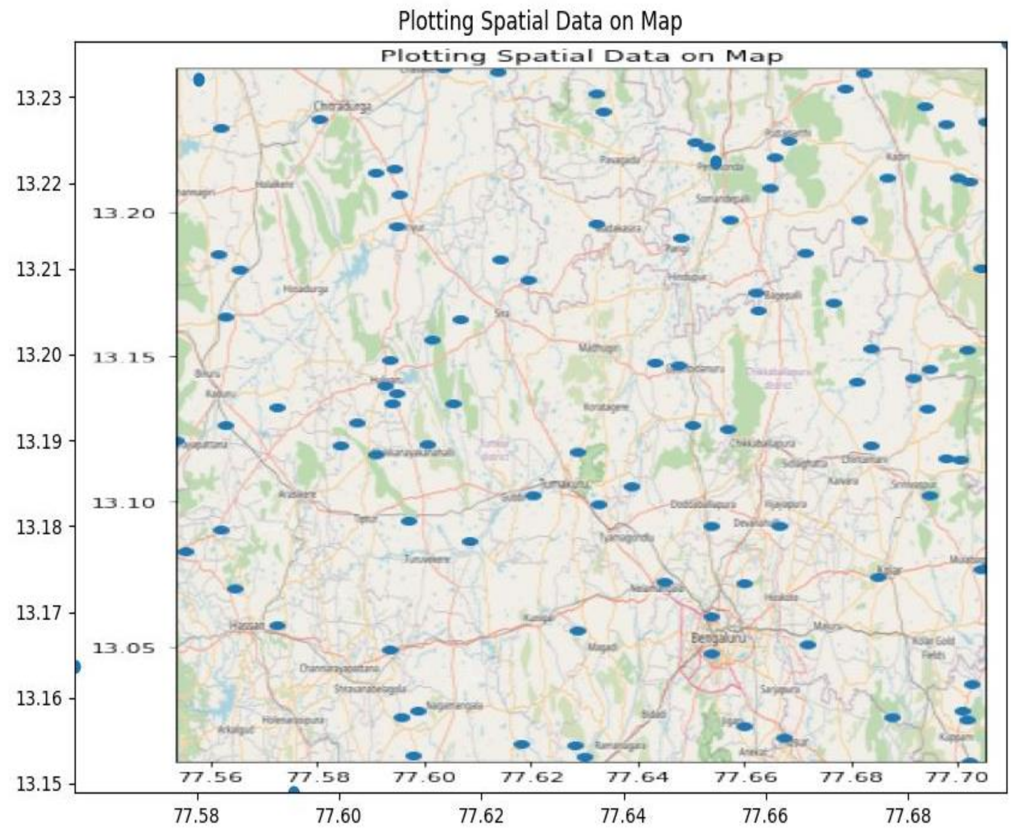
Code Implementation

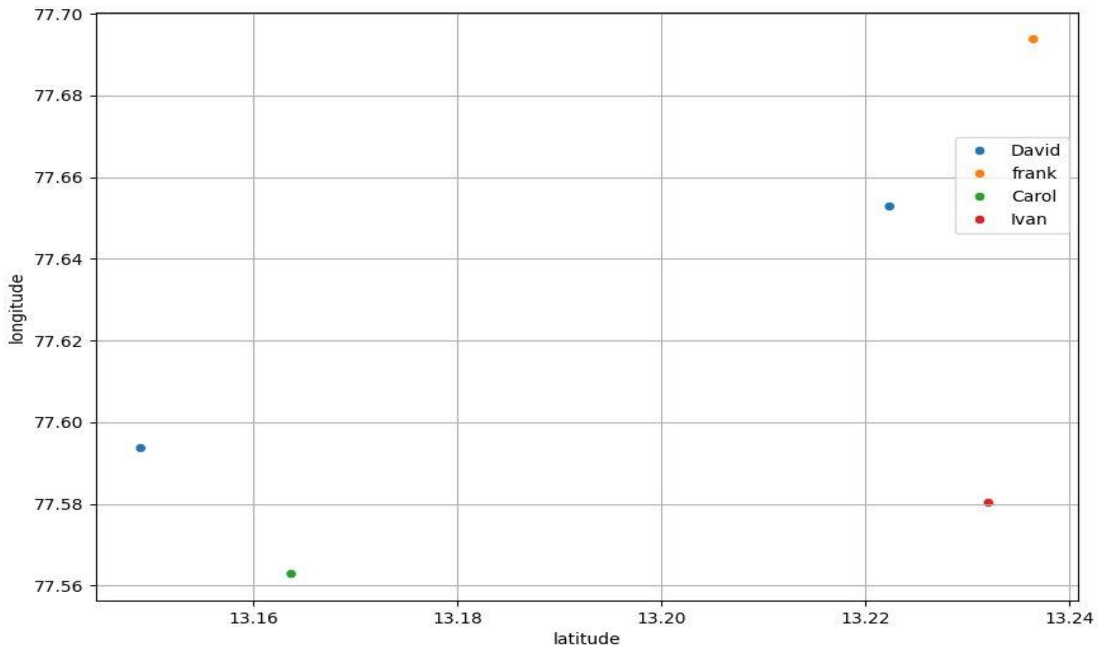
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
from sklearn.cluster import DBSCAN
df = pd.read_csv('D:\\contact_tracing.CSV')
print(df.head())
BBox = (df.longitude.min(),df.longitude.max(),df.latitude.min(),
df.latitude.max())
print(BBox)
banglore_m = plt.imread('D:\\map.png')
fig, ax = plt.subplots(figsize = (10,10))
ax.scatter(df.longitude, df.latitude)
ax.set_title('Plotting Spatial Data on Map')
ax.set_xlim(BBox[0],BBox[1])
ax.set_ylim(BBox[2],BBox[3])
ax.imshow(banglore_m, extent = BBox, aspect= 'equal')
plt.show()
plt.figure(figsize=(10,10))
sns.scatterplot(x="latitude",y="longitude",data=df,hue="id")
plt.legend(bbox_to_anchor= [1, 0.8])
plt.grid()
plt.show()
a=df[df.id=='Ivan']
print(a.head())
def generateBaseMap(default_location, default_zoom_start=12):
    base_map = folium.Map(location=default_location,
control_scale=True, zoom_start=default_zoom_start)
    return base_map
print(df.info())
print(df.duplicated().value_counts())
```

```
df=df.drop_duplicates()
print(df.isnull().sum())
# Converting column to datetime
df['timestamp']=pd.to_datetime(df['timestamp'])
# Creating hour column
df['hour']=df['timestamp'].apply(lambda x: x.hour+1)
print(df.head())
df2=pd.DataFrame(df.groupby(['hour','id'])['timestamp'].max())
print(df2.reset_index(inplace=True))
print(df2.head())
df3=pd.merge(df2,df,left_on=['hour','id','timestamp'],right_on=['hour',
'id','timestamp'])
print(df3.head())
lat_long_list = []
for i in range(1,25):
    temp=[]
    for index, instance in df3[df3['hour'] == i].iterrows():
        temp.append([instance['latitude'],instance['longitude']])
    lat_long_list.append(temp)
from branca.element import Figure
fig=Figure(width=550,height=350)
import folium
from folium.plugins import HeatMapWithTime
import webbrowser
fig7=Figure(width=850,height=550)
m7=folium.Map(location=[13.12, 77.8],zoom_start=10)
HeatMapWithTime(lat_long_list,radius=15,auto_play=True,position=
'bottomright').add_to(m7)
m7.save('bangalore_heatmap.html')
webbrowser.open('bangalore_heatmap.html')
```

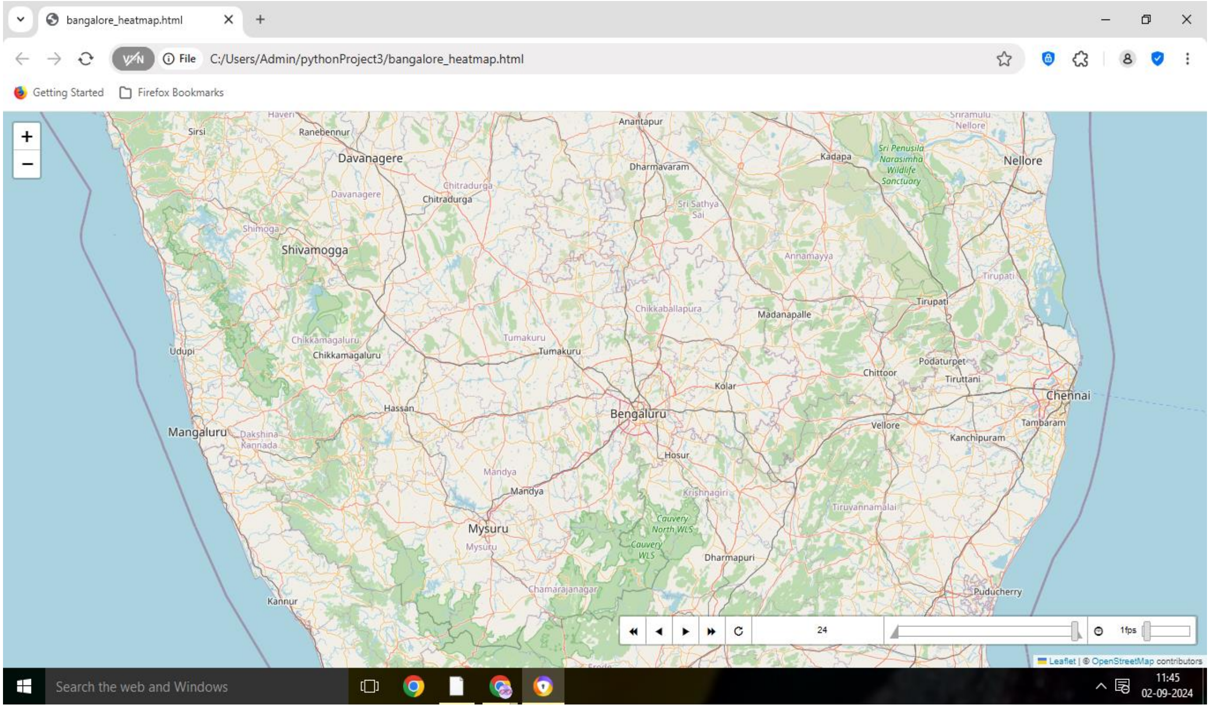
OUTPUT:

```
   Unnamed: 0     id          timestamp   latitude  longitude
0           0  David  04-07-2020 15:35  13.148953  77.593651
1           1  David  04-07-2020 16:35  13.222397  77.652828
2           2  frank  04-07-2020 14:35  13.236507  77.693792
3           3  Carol  04-07-2020 21:35  13.163716  77.562842
4           4   Ivan  04-07-2020 22:35  13.232095  77.580273
(77.562842, 77.693792, 13.148953, 13.236507)
```

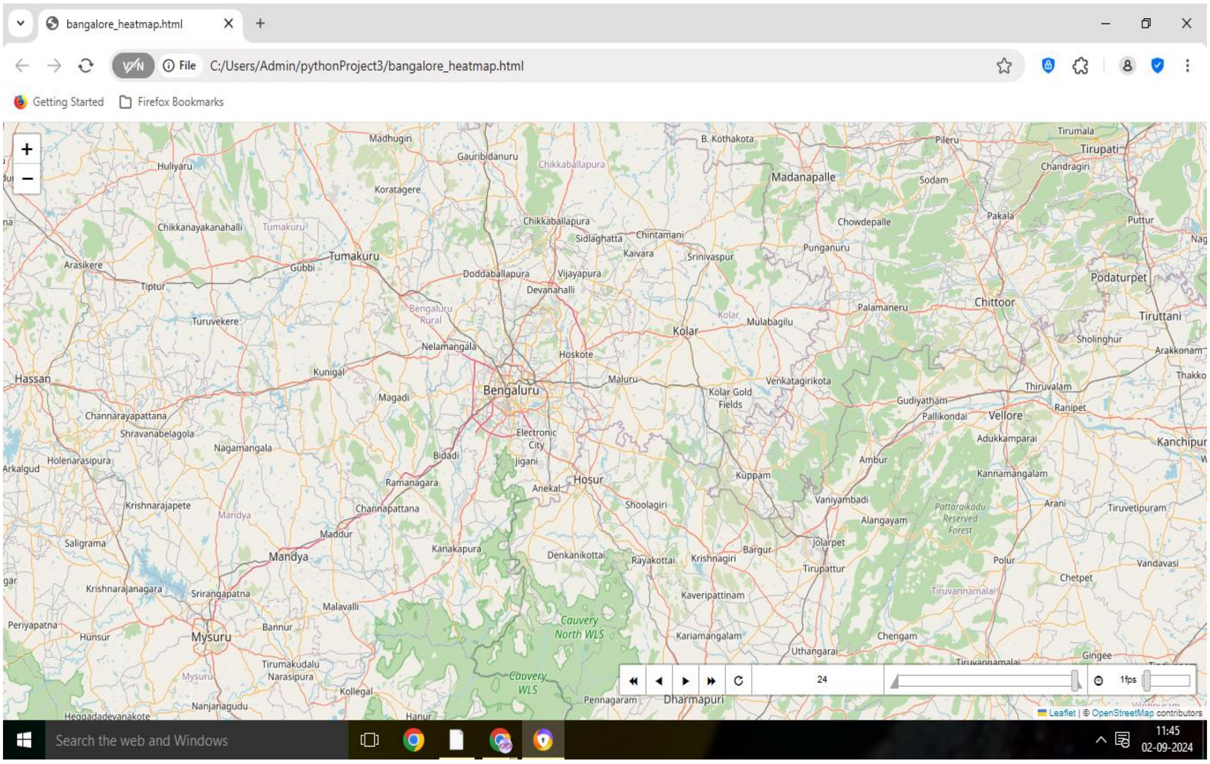





```
   Unnamed: 0    id         timestamp   latitude  longitude
4           4  Ivan  04-07-2020 22:35  13.232095  77.580273
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  5 non-null      int64
 1   id          5 non-null      object
 2   timestamp   5 non-null      object
 3   latitude    5 non-null      float64
 4   longitude   5 non-null      float64
dtypes: float64(2), int64(1), object(2)
memory usage: 328.0+ bytes
None
False    5
Name: count, dtype: int64
Unnamed: 0    0
id            0
timestamp     0
latitude      0
longitude     0
dtype: int64
```



bangalore_heatmap.html
C:/Users/Admin/pythonProject3/bangalore_heatmap.html
Getting Started
Firefox Bookmarks
Davanagere
Shivamogga
Nellore
Bengaluru
Chennai
Mangaluru
Mysuru
24
1fps
Leaflet | © OpenStreetMap contributors
Search the web and Windows
11:45
02-09-2024



bangalore_heatmap.html
C:/Users/Admin/pythonProject3/bangalore_heatmap.html
Getting Started
Firefox Bookmarks
Madanapalle
Tumakuru
Bengaluru
Kolar
Hosur
Chittoor
Vellore
Mandya
Mysuru
Dharmapuri
24
1fps
Leaflet | © OpenStreetMap contributors
Search the web and Windows
11:45
02-09-2024

Conclusion:

This project demonstrated the effective use of data visualization techniques to analyze contact tracing data, providing valuable insights into spatial and temporal patterns of contact occurrences. By plotting contact points on a map of Bangalore and overlaying them with spatial data, we were able to identify key clusters and areas with high contact density. The initial scatter plots over the map highlighted the distribution of contact points and allowed for a visual assessment of geographical spread, setting the stage for deeper analysis.

Temporal analysis, facilitated by creating an ‘hour’ column and grouping data accordingly, revealed variations in contact patterns throughout the day. The dynamic heatmap generated using Folium's ‘HeatMapWithTime’ feature vividly illustrated how contact points evolved over time, making it easier to identify peak hours and shifts in contact intensity. This visualization provided a clear, interactive representation of contact patterns, enhancing our understanding of time-based trends and their implications.

The insights gained from this analysis are crucial for public health monitoring and strategic response planning. Identifying high-contact areas and peak contact times can inform targeted interventions and resource allocation. Additionally, the project highlights the effectiveness of combining spatial and temporal data visualizations to gain a comprehensive view of contact patterns. This approach can be applied to similar datasets in other regions, offering a robust method for analyzing and responding to contact tracing data.