

Criminal geography is a branch of criminology that focuses on the spatial distribution and temporal patterns of criminal behavior. Research in this area reveals that crime occurrence is not random but is influenced by a variety of socioeconomic factors. By creating visual maps, people can better understand crime hotspots and help prevent crime. Merseyside is an important metropolitan area in the UK. Its complex social structure and dense population distribution make it an ideal case for understanding urban crime patterns. At the same time, this is also the area where I live and understand, so I would like to know the security situation around the area where I live and during this period.

Designed to explore Merseyside crime data, this dashboard provides an interactive view that allows users to understand crime distribution from multiple perspectives. By integrating the map with various visualization tools, the goal is to reveal relationships between crime types, locations, and their changes over time. The project not only demonstrates how to effectively utilize spatial data for in-depth analysis, but also seeks to enhance the user experience through a series of interactive elements. The project mainly includes the crime data released by the Merseyside Police in the past six months. Through map display, it provides a way for the public living in Merseyside to understand the public safety around them and promotes relevant Reflection and discussion on social and public safety.

At first load all necessary library functions

```
In [1]: import osmnx as ox
import folium
import pandas as pd
from pathlib import Path
import seaborn as sns
import matplotlib.pyplot as plt
import osmnx as ox
import pandas as pd
import folium
import panel as pn
from folium.plugins import MarkerCluster
from pathlib import Path
from IPython.display import HTML
from bokeh.resources import INLINE
```

Visualization of the geographical boundaries of Merseyside. This section is dedicated to visualizing the geographical boundaries of Merseyside, England. I utilize the "osmnx" library to obtain boundary data in GeoDataFrame format. The purpose of this visualization is to provide a clear and immediate understanding of the extent and layout of Merseyside. and also serves as a basic visual tool for further analysis of spatial data in the region.

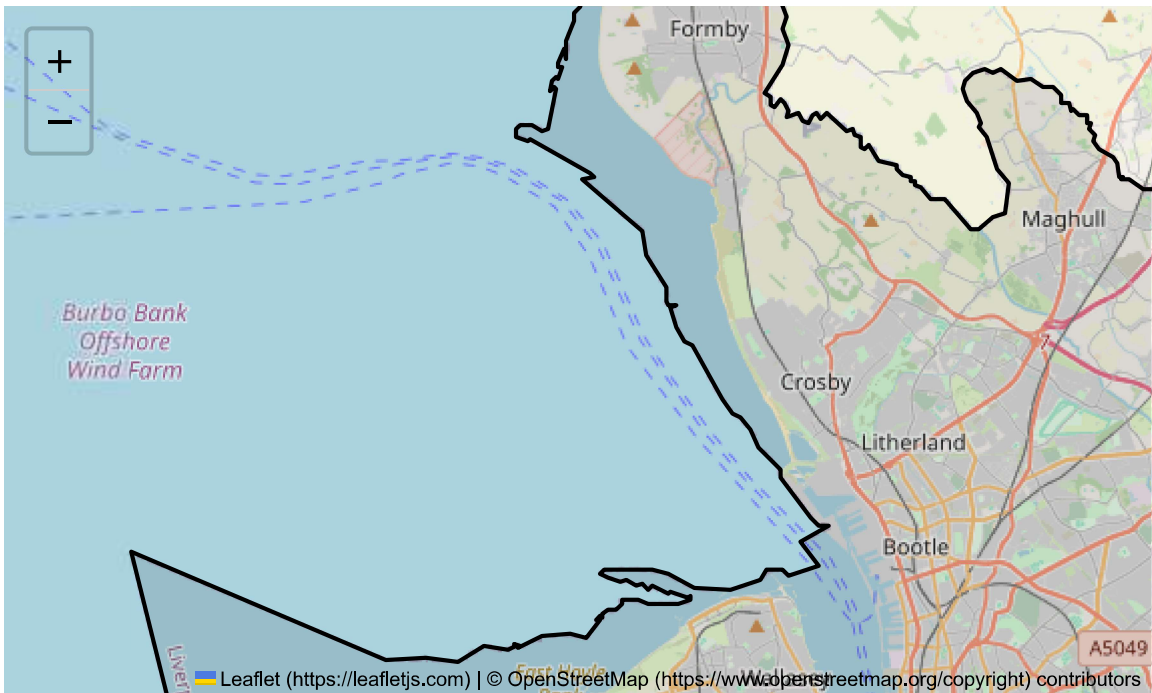
```
In [2]: # Get the geographical boundaries of Merseyside
place_name = "Merseyside, England"
merseyside_boundary = ox.geocode_to_gdf(place_name)

# Create map
```

```
def show_boundary_map():
    center_lat, center_lon = 53.4084, -2.9916 # Approximate central location of M
    map_boundary = folium.Map(location=[center_lat, center_lon], zoom_start=10)
    folium.GeoJson(
        merseyside_boundary,
        name='Merseyside Boundary',
        style_function=lambda feature: {'color': 'black', 'weight': 2, 'fillOpacity': 0},
    ).add_to(map_boundary)
    return map_boundary

# show map
boundary_map = show_boundary_map()
boundary_map
```

Out[2]:



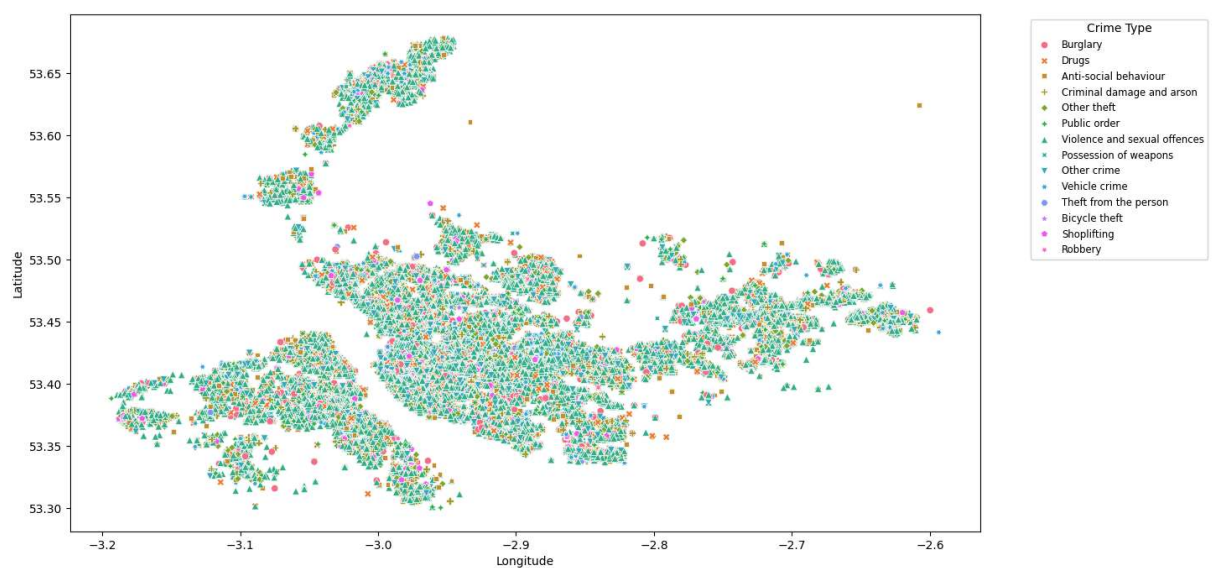
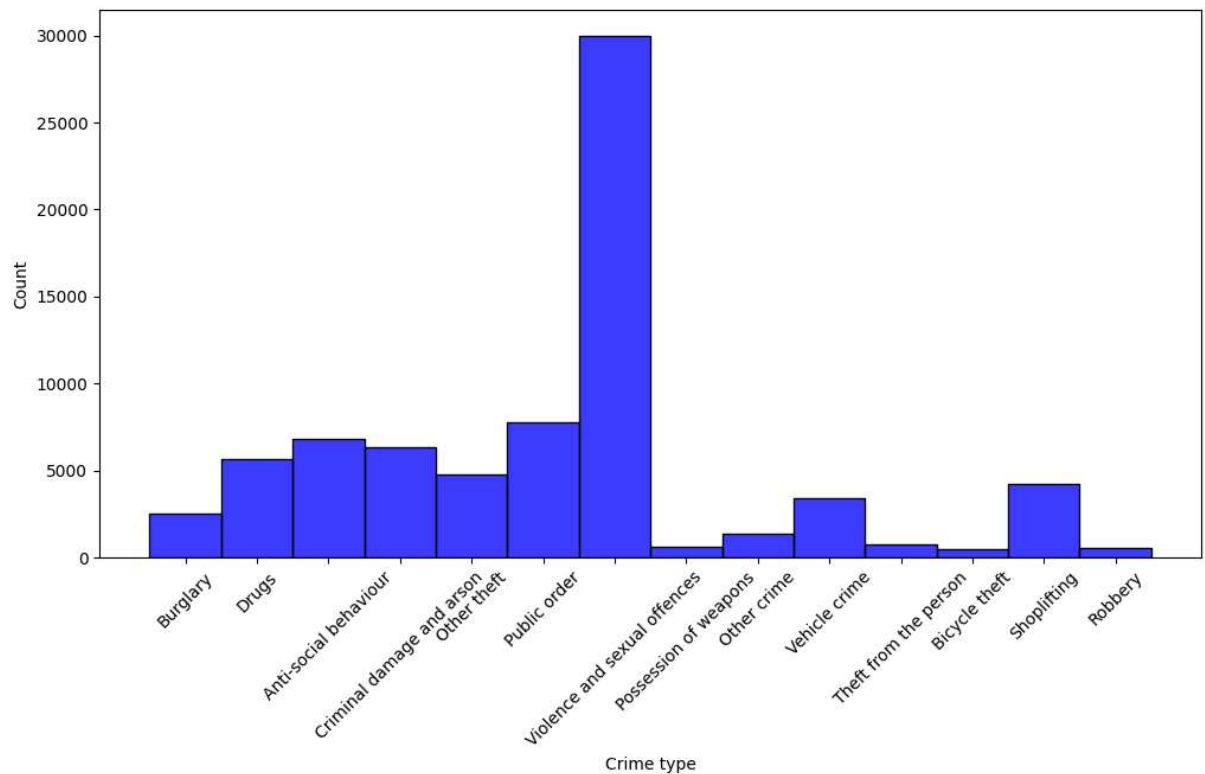
For this dashboard we used the Merseyside publicly available crime dataset, which includes detailed records of criminal activity from September 2023 to February 2024. This is the latest six-month figure released by police. The dataset contains various attributes such as crime type, end result, and approximate location, enabling comprehensive time series and geospatial analysis. During the initial visualization design process, I designed visual representations such as histograms and scatter plots to make data interpretation intuitive. From the histogram, it can be seen that Violence and sexual offenses are the most frequently recorded crimes, while Possession of weapons is the least. This level of detail allows us to represent criminal activity across multiple dimensions, helping to understand its dynamics and spatial correlations.

```
In [3]: # Data Loading
data_folder = Path("D:/envs456/data/crime data")
files = [data_folder / f"2023-{i:02d}-merseyside-street.csv" for i in range(9, 13)]
files += [data_folder / f"2024-{i:02d}-merseyside-street.csv" for i in range(1, 3)]
data_frames = [pd.read_csv(file) for file in files]
all_data = pd.concat(data_frames, ignore_index=True)

#Data preprocessing
all_data['Month'] = pd.to_datetime(all_data['Month']).dt.to_period('M')
```

```
#Set the size of the graphic
plt.figure(figsize=(12, 6))
# Histogram
sns.histplot(data=all_data, x='Crime type', color='blue')
plt.xticks(rotation=45)
plt.show()

# Set the size of the scatter plot
plt.figure(figsize=(14, 8))
# Scatter plot
scatter_plot = sns.scatterplot(data=all_data, x='Longitude', y='Latitude', hue='Cr
scatter_plot.legend(title='Crime Type', bbox_to_anchor=(1.05, 1), loc='upper left'
plt.show()
```



The interactive design of our dashboard is centered on enhancing user engagement. I included a time filter slider and a crime type dropdown to allow users to tailor the data views to specific months or crime types. This customization makes the analysis

more targeted and increases user interaction with the dataset. The map's interactive markers reveal detailed information about each crime event, such as type and outcome, with Liverpool city center showing a high concentration of incidents. This approach not only enriches the dashboard's informational depth but also highlights spatial trends in crime distribution, making data exploration more dynamic and informative.

```
In [4]: # Initialize Panel and set OSMnx to use cache
pn.extension()
ox.settings.use_cache = True

# Month selector
months = all_data['Month'].dt.strftime('%Y-%m').unique().tolist()
month_slider = pn.widgets.DiscreteSlider(name='Select Month', options=months, value=months[0])

# Crime type selector
crime_types = ['All Crimes'] + sorted(all_data['Crime type'].unique().tolist())
crime_selector = pn.widgets.Select(name='Select Crime Type', options=crime_types, value='All Crimes')

# Crime type and color mapping
crime_color_map = {
    'Anti-social behaviour': 'blue',
    'Bicycle theft': 'green',
    'Burglary': 'purple',
    'Criminal damage and arson': 'red',
    'Drugs': 'gray',
    'Other crime': 'darkblue',
    'Other theft': 'darkgreen',
    'Possession of weapons': 'orange',
    'Public order': 'lightblue',
    'Robbery': 'pink',
    'Shoplifting': 'cadetblue',
    'Theft from the person': 'lightgreen',
    'Vehicle crime': 'black',
    'Violence and sexual offences': 'darkred'
}

# Function to create the map
def create_folium_map(data):
    center_lat, center_lon = 53.4084, -2.9916 # Approximate center of Merseyside
    folium_map = folium.Map(location=[center_lat, center_lon], zoom_start=10, min_zoom=10, max_zoom=15)
    marker_cluster = MarkerCluster().add_to(folium_map)

    # Add Merseyside boundary to the map
    folium.GeoJson(
        merseyside_boundary,
        name='Merseyside Boundary',
        style_function=lambda feature: {'color': 'black', 'weight': 2, 'fillOpacity': 0}
    ).add_to(folium_map)

    # Create and add Legend
    legend_html = '''
    <div style="position: fixed;
        top: 10px; right: 10px; width: 120px; height: auto;
        background-color: rgba(255,255,255,0.7); border:1px solid grey; z-index: 1000; padding: 5px 10px;">
        &nbsp;   Crime Type <br>
    ...'''
```

```

for crime, color in crime_color_map.items():
    legend_html += f'<i style="background: {color}; width: 12px; height: 12px;'
legend_html += '</div>'
folium_map.get_root().html.add_child(folium.Element(legend_html))

# Add crime data markers
for idx, row in data.iterrows():
    marker_color = crime_color_map.get(row['Crime type'], 'gray')
    folium.Marker(
        location=[row['Latitude'], row['Longitude']],
        popup=f"Crime Type: {row['Crime type']}<br>Last Outcome: {row['Last ou'
        icon=folium.Icon(color=marker_color)
    ).add_to(marker_cluster)

return folium_map

# Function to update the map
@pn.depends(month_slider.param.value, crime_selector.param.value)
def update_map(selected_month, selected_crime):
    filtered_data = all_data[(all_data['Month'] == selected_month) & (all_data['Cr
    folium_map_html = create_folium_map(filtered_data)._repr_html_()
    return pn.pane.HTML(folium_map_html, width=1000, height=600, sizing_mode='fixe

# Build and display the dashboard
dashboard = pn.Column(
    pn.Row(month_slider, crime_selector, width=1000, height=50, sizing_mode='fixed'
    update_map,
    width=1000, height=660, sizing_mode='fixed' # Added sizing_mode for explicit
)
dashboard.save('D:/envs456/labs/dashboard.html', embed=True, resources=INLINE)
with open('D:/envs456/labs/dashboard.html', 'r') as file:
    html_content = file.read()
HTML(html_content)

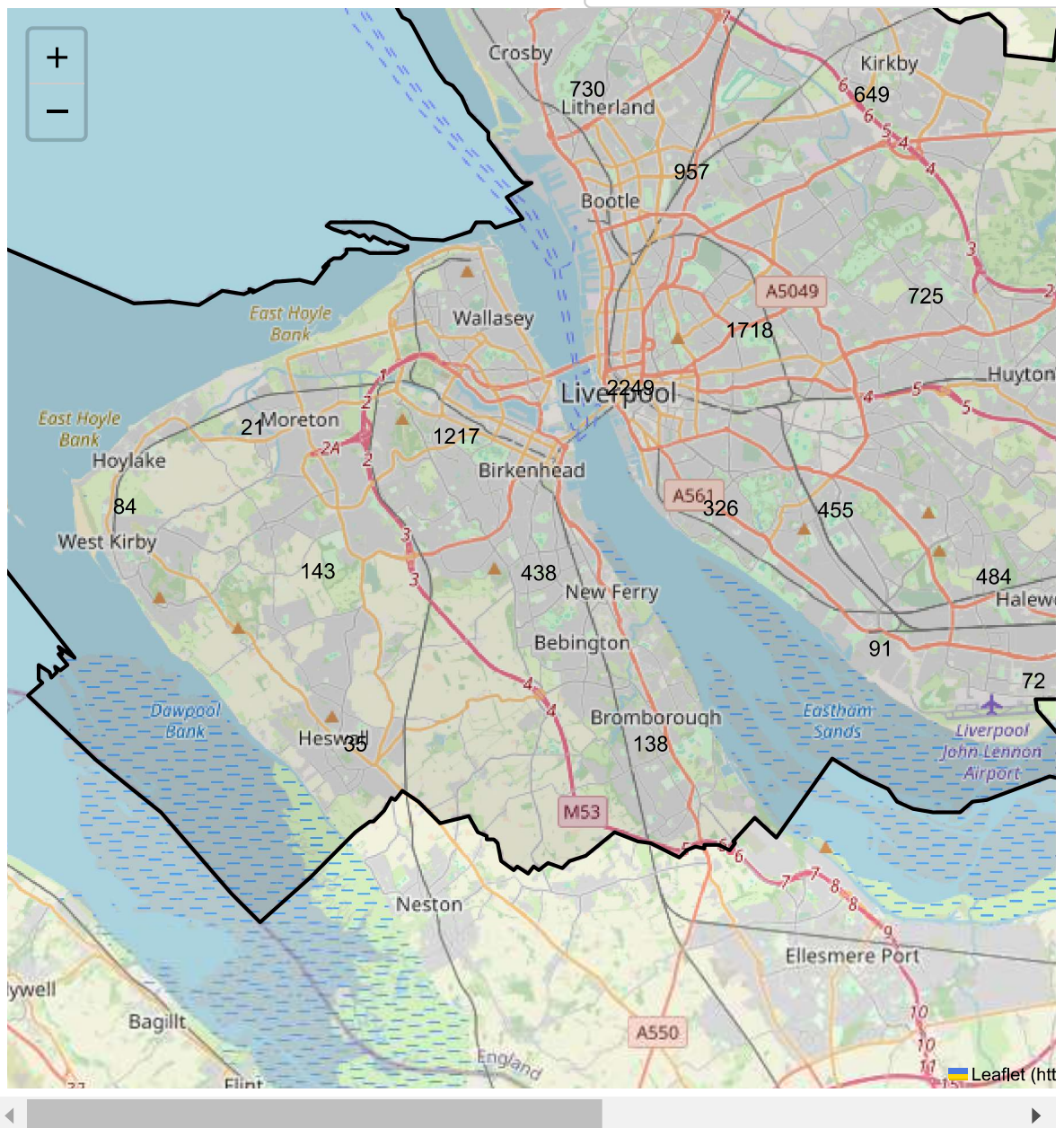
```

WARNING:bokeh.core.validation.check:W-1005 (FIXED_SIZING_MODE): 'fixed' sizing mode requires width and height to be set: Column(id='5a082c88-d4d3-488c-b8a9-15422df084d1', ...)

Out[4]: Select Month: **2023-09**

Select Crime Type

All Crimes



The resulting map allows for viewing the location and details of different areas and crime cases at appropriate zoom levels. In addition, I also made different color legends for different crime types, so that when viewing all crimes, you can clearly see where and what kind of crimes occurred. From this final map, I could clearly see where crime was occurring, down to the street and block level. At the same time, I can clearly understand what type of cases occurred where on the map through the color legend. If I need to check the geographical information of a specific time or a specific type of crime. I can make selections via the map's widget. These designs provide an intuitive and concise crime information visualization map for ordinary people and crime researchers.