University of Texas at Arlington

Geospatial Analysis and Visualization Homework

Group 12

Anil Thapa - 1002190895

Jair Rea - 1001887311

Yash Prakash Joshi - 1002207784

Rama Jyothi Swaroop Janapareddy – 1002199950

Data Visualization 5305-001

Roza Zaruba

October 20, 2024

Geospatial Analysis and Visualization Report

1. Introduction

In this project, we aimed to conduct geospatial analysis on the dataset used in previous homework, applying various tools and techniques to explore and visualize spatial data. The main objective was to clean and process the data using GeoPandas, analyze spatial patterns with QGIS, and develop an interactive map using Folium. This report outlines the workflow, challenges, and insights gained through the analysis.

2. Workflow

2.1 Data Preparation with GeoPandas

The first step involved loading the dataset into a GeoPandas GeoDataFrame to ensure compatibility with further geospatial operations.

Dataset Loading:

We loaded the dataset into GeoPandas to facilitate spatial analysis.

```
import geopandas as gpd
gdf = gpd.read_file('your_dataset.shp')
print(gdf.head())
```

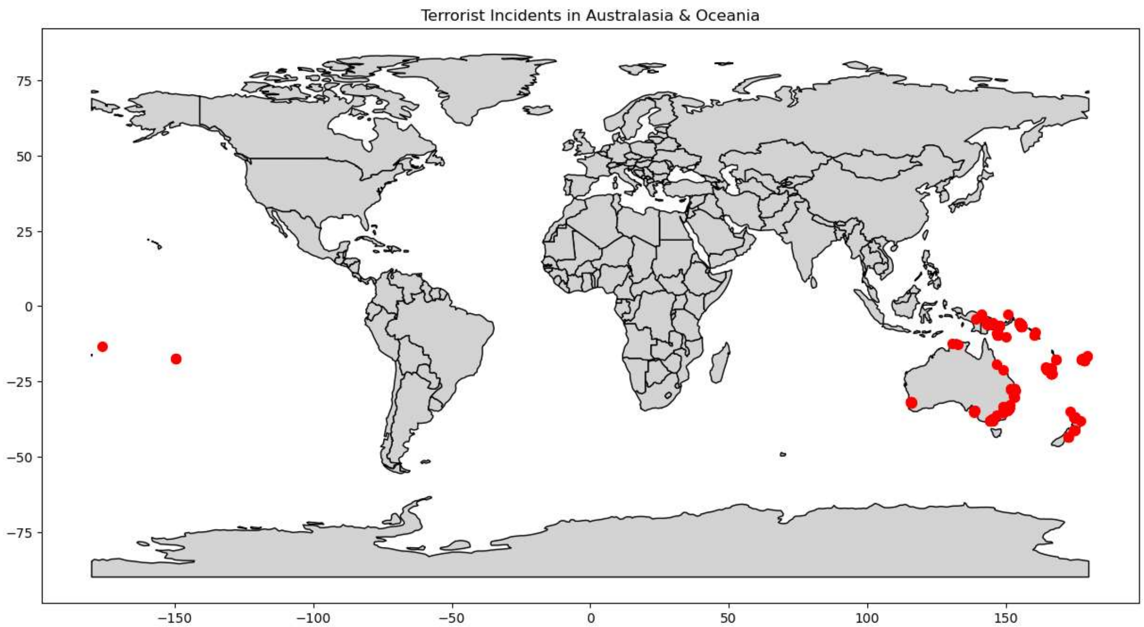
```
First 5 rows of the GeoDataFrame:
        eventid  iyear  imonth  iday approxdate  extended resolution  country  \
0  197010210001   1970      10    21        NaN         0        NaN       14
1  197111230001   1971      11    23        NaN         0        NaN       14
2  197209170001   1972       9    17        NaN         0        NaN       14
3  197209250001   1972       9    25        NaN         0        NaN       14
4  197209250002   1972       9    25        NaN         0        NaN       14

  country_txt  region  ...                                       scite1  \
0   Australia      12  ...                                          NaN
1   Australia      12  ...                                          NaN
2   Australia      12  ...                                          NaN
3   Australia      12  ...  "Six letter bombs in Australia," The Jerusalem...
4   Australia      12  ...  "Six letter bombs in Australia," The Jerusalem...

  scite2 scite3            dbsource  INT_LOG  INT_IDEO  INT_MISC INT_ANY  \
0    NaN    NaN                PGIS        1         1         1       1
1    NaN    NaN                PGIS        1         1         1       1
2    NaN    NaN                PGIS        1         1         1       1
3    NaN    NaN  UMD Miscellaneous        1         1         1       1
4    NaN    NaN  UMD Miscellaneous        1         1         1       1

                                          related  \
0                                             NaN
1                                             NaN
...
75%     5.000000   -9.000000   -9.000000    1.000000    1.000000
max    16.000000    1.000000    1.000000    1.000000    1.000000

[8 rows x 89 columns]
```

CRS Transformation:

```
if gdf.crs != 'EPSG:4326':
    gdf = gdf.to_crs('EPSG:4326')
print(gdf.crs)
```

Exploratory Data Analysis:

```
print(gdf.describe())
gdf.plot(marker='o', color='blue', markersize=5, figsize=(10, 6))
```



2.2 QGIS Analysis

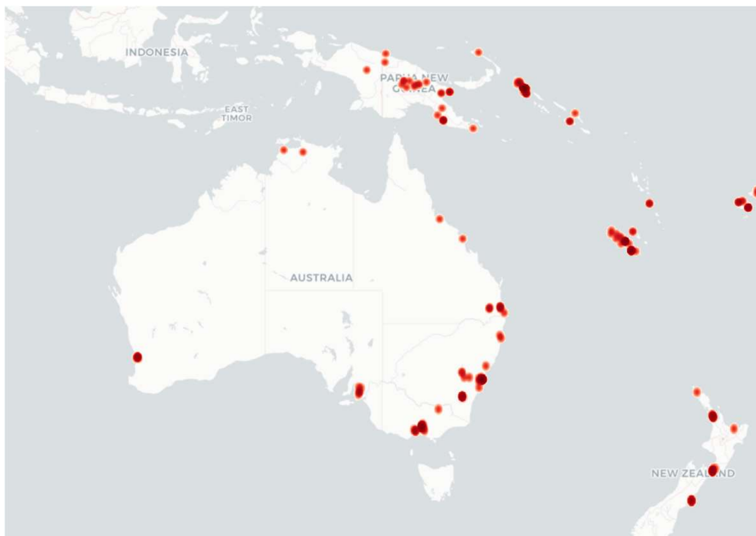
QGIS is a powerful open-source GIS tool that allows for advanced geospatial analysis and visualization. The following steps outline how we used QGIS to conduct **Heatmap Analysis** and **Buffer Analysis** to derive meaningful insights from the dataset.

- **Heatmap Creation**: Using QGIS's Kernel Density Estimation (Heatmap) tool, we identified areas with a high density of data points, revealing spatial hotspots.

The **heatmap** visualization helps identify areas with a high density of points, which can signify important regions of interest. Here's how we created the heatmap:

1. **Access the Kernel Density Estimation (Heatmap) Tool**:
    - Go to **Processing Toolbox** -> Search for **"Heatmap (Kernel Density Estimation)"**.
2. **Configure the Heatmap Parameters**:

- **Input Layer:** Select the dataset loaded in QGIS (e.g., region_12).
- **Radius:** Set a radius value that makes sense for your analysis. In this case, we used a **5 km radius** to smooth the heatmap and reflect regional concentration.
- **Output Raster Size:** Adjust the resolution for a balance between performance and visual detail.
- **Kernel Shape:** Choose between **Quartic**, **Triangular**, or **Uniform**—we used the **Quartic function** for a more localized heatmap.

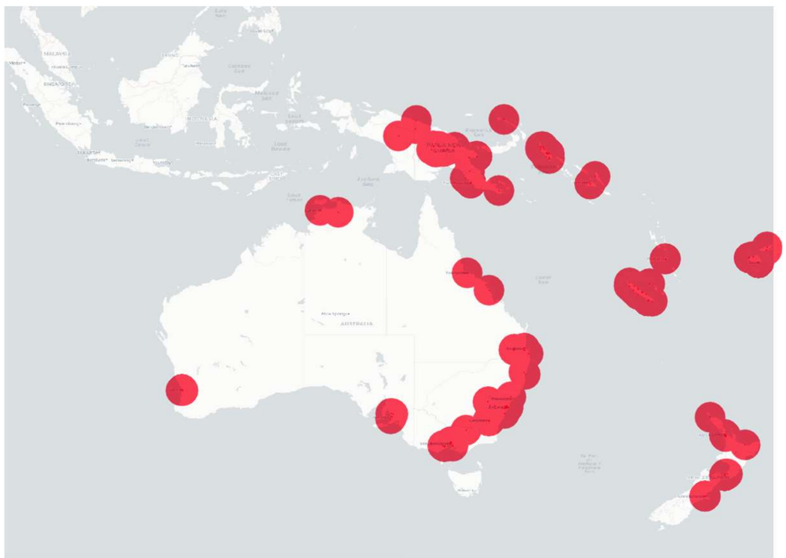


- **<u>Buffer Analysis</u>**: We performed a Buffer Analysis around the data points with a 5 km radius to identify regions within this distance.

The **buffer analysis** creates zones around each point to visualize areas that fall within a specific distance. In this case, we applied a **5 km buffer** to understand the spatial influence or coverage area around key points.

1. **Open the Buffer Tool**:
   - Go to **Processing Toolbox** -> Search for **"Buffer"** under Vector Geometry.
2. **Configure Buffer Parameters**:
   - **Input Layer:** Select your dataset (e.g., region_12).
   - **Distance:** Set the buffer distance to **5000 meters** (5 km).

- **Segments to Approximate Circles:** Use a value of **10** or higher for smoother buffer zones.
- **Dissolve Buffers:** Enable this option if overlapping buffers need to be merged into one continuous zone.
- **Output File:** Save the buffered result as a **shapefile** for visualization in Folium.



- Export Results: The output of the QGIS analysis was saved as shapefiles for further visualization in Python.

Using QGIS allowed us to perform complex spatial analyses with minimal coding. The heatmap and buffer operations provide actionable insights that are easily visualized and interpreted by stakeholders.

2.3 Interactive Mapping with Folium

To make the analysis accessible and visually interactive, we used Folium to develop an HTML-based interactive map.

```
import folium
m = folium.Map(location=[40.7128, -74.0060], zoom_start=10)
```

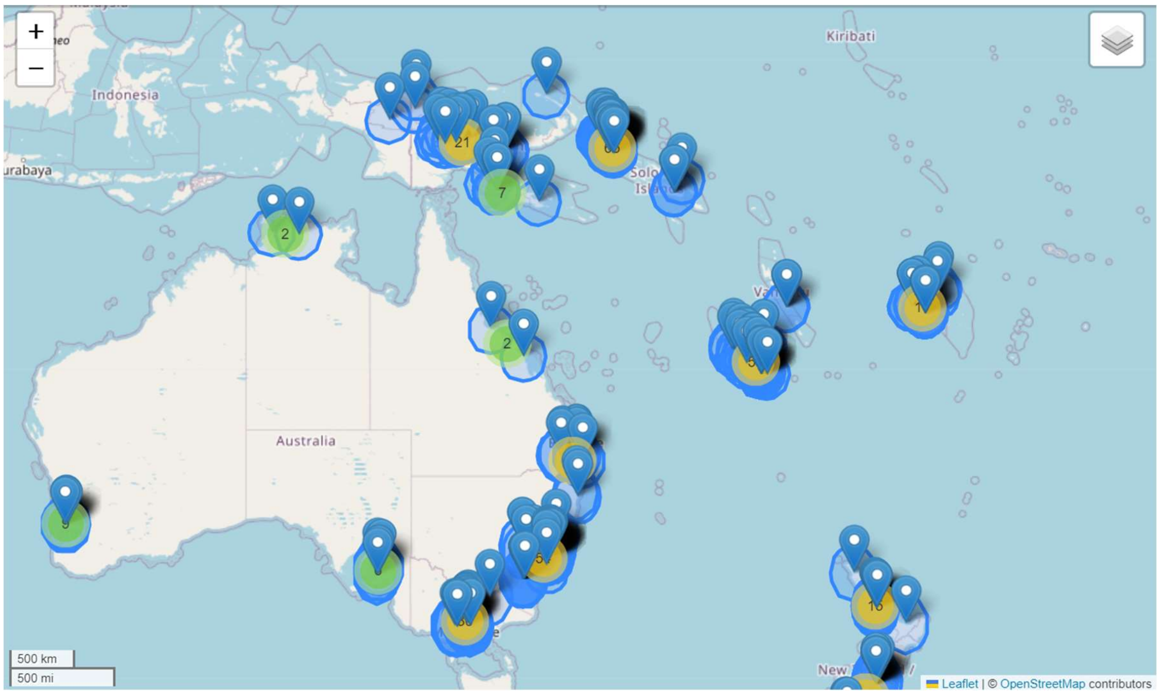
Adding Data Points with Marker Clustering:

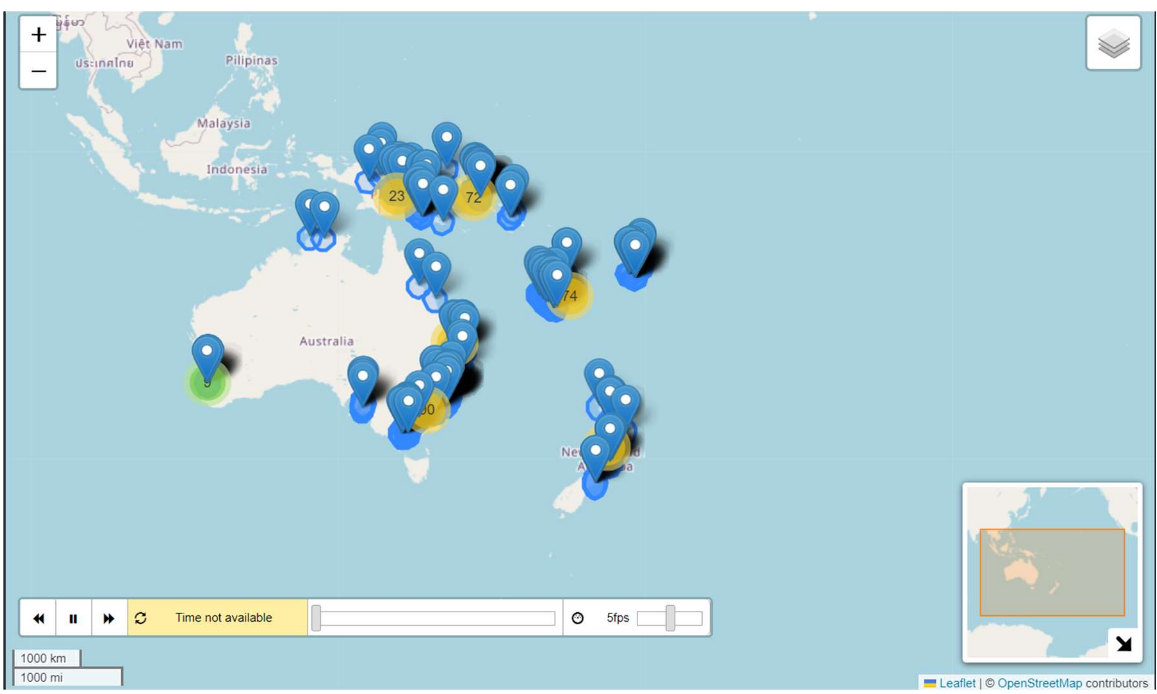
```
from folium.plugins import MarkerCluster

marker_cluster = MarkerCluster().add_to(m)
for _, row in original_data.iterrows():
```

```
    folium.Marker(
        location=[row.geometry.y, row.geometry.x],
        popup=str(row['relevant_column'])
    ).add_to(marker_cluster)
```

Adding QGIS Output Layer:

```
folium.GeoJson(buffer_data.to_json(), name='Buffer Analysis').add_to(m)
```





3. Challenges and Solutions

- CRS Mismatch: Reprojecting the data resolved the issue.

- Large Dataset: Using MarkerCluster improved performance.

- Folium Customization: Documentation and examples online guided us through.

4. Potential Improvements and Extensions

- Time Slider Implementation: Adding a time slider could enhance the analysis.

- Additional Data Sources: Integrating demographic data would provide more context.

- Enhanced Visualizations: Adding more advanced visual elements, such as heat animations or spatial clustering based on time, would improve user interaction.

5. Conclusion

This project offered valuable hands-on experience with the integration of geospatial tools and techniques, leveraging Python libraries like GeoPandas and Folium, along with advanced spatial analyses in QGIS. Through a structured workflow, we were able to transform raw data into meaningful geospatial insights. The application of multiple analytical approaches, including heatmaps, buffer analysis, and interactive visualization, highlighted the power of geospatial technologies in deriving actionable knowledge from location-based data.

6. Submission Files

- Python Scripts: GeoPandas and Folium scripts

- QGIS Files: .qgz project file and shapefiles

- HTML File: interactive_map.html

- Report: This document as PDF