Master’s of Science in Information Technology [Taught

and Research (Mainly Taught)]

2024-2025

CIS5212 : Geographic Information Systems

Matthew Misfud (279503L)
Parthi Vishnawat (2325322)

URECA Form

Draft/Submitted REDP Forms

Show 10 entries Search:

Application ID	Application Date	Project Title	Faculty	Status
ICT-2025-00097	26/05/2025	GIS Project	Faculty of Information & Communication Technology	Pending Endorsement (supervisor)

Division of work:

Parthi worked on the spatial analysis aspect of the assignment, while Matthew worked on the user interface. Both Parthi and Matthew worked together to create the documentation of this assignment.

Plagiarism Form

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own work, of work of another person, whether or not such work has been published" (Regulations Governing Conduct at Examinations, 1997, Regulation 1 (viii), University of Malta).

I / We*, the undersigned, declare that the [assignment / Assigned Practical Task report / Final Year Project report] submitted is my / our* work, except where acknowledged and referenced.

I / We* understand that the penalties for making a false declaration may include, but are not limited to, loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

* Delete as appropriate.

(N.B. If the assignment is meant to be submitted anonymously, please sign this form and submit it to the Departmental Officer separately from the assignment).

Matthew Mifsud
Student Name — Signature

Parthi Vishnawat
Student Name — Signature

Student Name — Signature

Student Name — Signature

CIS5212
Course Code

Mapping Shooting Incidents and Hospital Proximity in Philadelphia: A GIS-Based Approach
Title of work submitted

30/05/2025
Date

Title: "Mapping Shooting Incidents and Hospital Proximity in Philadelphia: A GIS-Based Approach"

Introduction

Philadelphia continues to struggle with high rates of gun violence, ranking among the top U.S. cities for shootings in recent years. These events not only endanger residents but also put a heavy load on local hospitals and emergency responders, since quick access to trauma care is often crucial for survival.

In this study, we map shooting incidents and nearby hospitals to see how well medical resources match the areas most affected by gun violence. We used two open datasets from Data.GOV:

- **Shooting Victims** (https://catalog.data.gov/dataset/shooting-victims), which gives dates and coordinates for each shooting.
- **Philadelphia Hospitals** (https://catalog.data.gov/dataset/philadelphia-hospitals), which lists every hospital in the city along with its location and details.
- **Licensing Approval** (https://metadata.phila.gov/#help/help-faqs/what-are-the-terms-of-use/).

Our goals are to:

1. **Find clusters of shootings** to spot neighborhoods with the most incidents.

2. **Check hospital access** for those high-incident areas by measuring distances to the nearest trauma centers.

3. **Reveal underserved areas** that may lack quick access to emergency care when shootings occur.

Using these public datasets lets us create a clear, repeatable process for city planners, health officials, and EMS teams to understand where help is needed most.

Problem Identification:

Philadelphia has long struggled with high levels of gun violence, frequently ranking among the nation's most affected cities. This spike in shootings not only jeopardizes public safety but also strains local hospitals and emergency responders, as rapid access to trauma care can greatly affect patient survival and recovery.

A major obstacle in tackling this problem is understanding where shootings happen compared to where hospitals are located. Without clear spatial insights, it is difficult to deploy resources effectively or develop targeted strategies to improve both safety and healthcare access.

This study examines where shooting incidents occur in Philadelphia and how easily victims in those areas can reach nearby hospitals. We rely on two openly available datasets from Data.gov—one detailing shooting victims and another listing all Philadelphia hospitals with their coordinates—to:

- Identify clusters and hotspots of gun violence across the city.
- Measure how close high-incident neighborhoods are to the nearest hospital.
- Highlight neighborhoods that lack convenient access to emergency care.

By mapping and analyzing these spatial relationships, our goal is to provide planners and policymakers with actionable insights that support data-driven decisions aimed at enhancing community safety and ensuring timely trauma care.

Libraries and Software used

Library/ Software	Purpose
Pandas [2]	Data manipulation
Geopandas [3]	GIS support and spatial operations
Shapely [4]	Geometry objects (Point, Polygon)
Matplotlib [5]	Plotting graphs and maps
Contextily [6]	Add basemaps to plots
Scipy [7]	Spatial analysis (nearest neighbours)
QGIS [8]	To create shapefiles and maps
Pgadmin (Postgres) [9]	Database
Docker [10]	Creation of database

Table 1: Software and Library used

Dataset Description

A. Hospital Dataset

This dataset contains location and attribute information for hospitals across the city of Philadelphia. It was prepared to support spatial analysis on healthcare accessibility in relation to gun violence incidents.The dataset includes the following fields:

Field Name	**Description**
OBJECTID	A system-generated unique identifier for each hospital entry.
HOSPITAL_NAME	The official name of the hospital.
STREET_ADDRESS	The physical street address of the hospital.
ZIP_CODE	The postal ZIP code for the hospital location.
PHONE_NUMBER	Contact number for general inquiries or emergency communication.
HOSPITAL_TYPE	Classification of the facility (e.g., General Hospital, Clinic Trauma Center, Long Term Care).
LAT, LNG	Latitude and Longitude coordinates derived through geocoding for spatial mapping.

Table 2: Hospital dataset description

The dataset was geocoded to enable accurate plotting on a map and to support spatial proximity and network analysis with respect to Shooting incident locations.

B. Shooting Dataset

This dataset documents reported shooting incidents within the City of Philadelphia. It captures both spatial and non-spatial attributes that are essential for analysing patterns of gun violence and understanding their potential impact on different population groups. The dataset is structured with the following fields:

Field Name	**Description**
year, date_,time	Temporal attributes indicating when the incident occured.

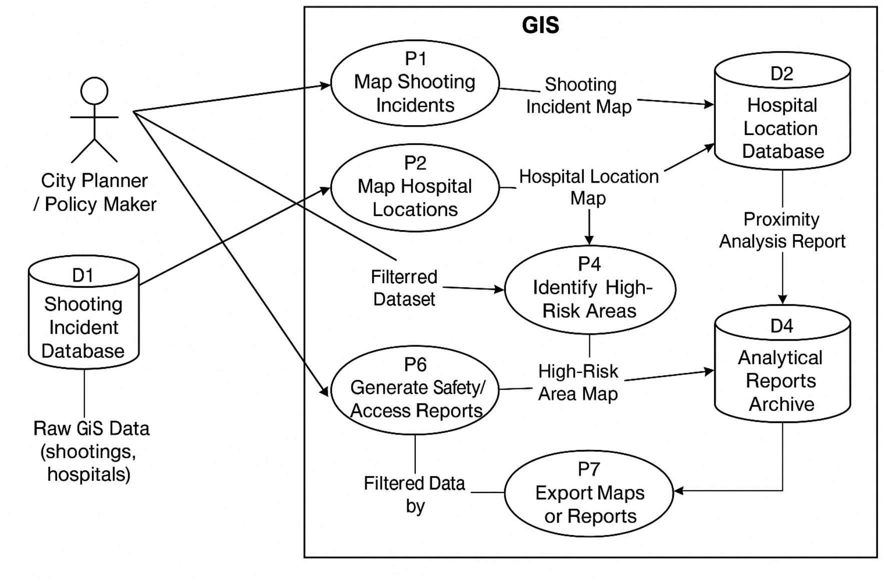
age	Age of the individual(s) involved in the shooting.
race	Race or ethnicity of the involved individual(s).
sex	Gender of the involved individual(s).
officer_involved	Indicates whether a law enforcement officer was involved in the shooting (Y/N).
fatal	Indicates if the incident resulted in a fatality (0/1).
wound	Indicates if the individual was wounded -classification (e.g., Arm , Leg, Back etc.)
Lat, Ing	Latitude and longitude coordinates representing the location of the incident.

Table 3: Shooting dataset description

This dataset enables spatial and demographic analysis of gun violence, including clustering, heat mapping and proximity analysis in relation to hospitals. This integration of temporal and outcome data supports trend analysis and emergency response planning.

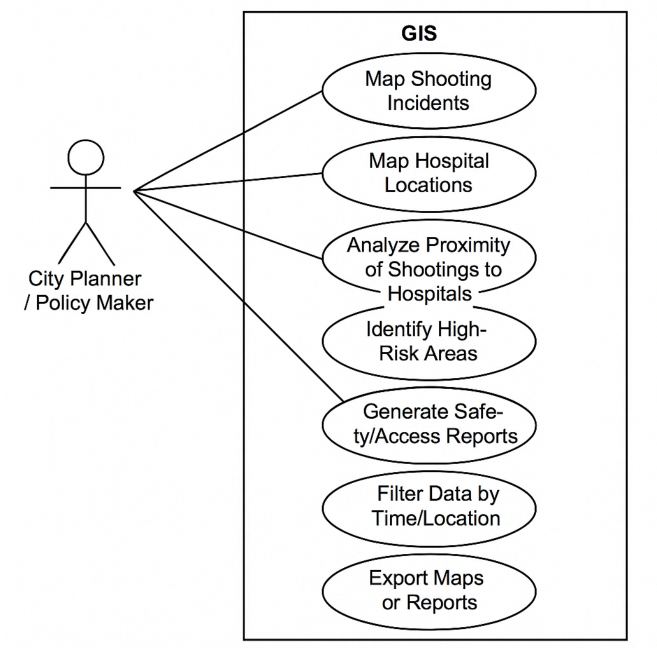
Stage 1 - Initial design requirements

Data Flow Diagram



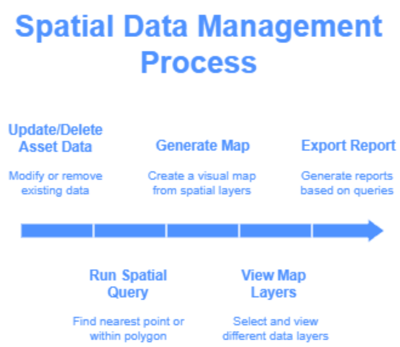
The data flow diagram presents a GIS-based support system that enables municipal decision-makers to analyze urban safety and healthcare access by integrating shooting incident and hospital location data. Raw shooting incident records (D1) and hospital geographic information (D2) are first imported, after which users can generate visual layers for incident locations (P1) and hospital positions (P2). These mapped layers feed into a risk evaluation module (P4) that overlays shooting density with proximity to the nearest trauma center, producing a high-risk area map and a proximity analysis report that are stored in the analytical archive (D4). Building on these outputs, the system can compile comprehensive safety and access reports (P6) and allows exporting of maps and documents (P7) for stakeholder distribution. Throughout, city planners and policymakers interact at each stage—guiding analyses, reviewing results, and using the insights to inform resource allocation and policy decisions.

Use Case Diagram



The Use Case diagram outlines how city planners or policymakers leverage a GIS to improve urban safety and healthcare planning by providing a suite of core functions. Users can visualize shooting incident locations and hospital sites, then perform spatial analysis to measure proximity between incidents and medical facilities—crucial for coordinating emergency response. The system also integrates incident and hospital data to identify high-risk neighborhoods, generates automated safety and access reports, and allows filtering by date or area for more precise analysis. Finally, planners can export maps and documentation to share insights with stakeholders and guide policy or resource

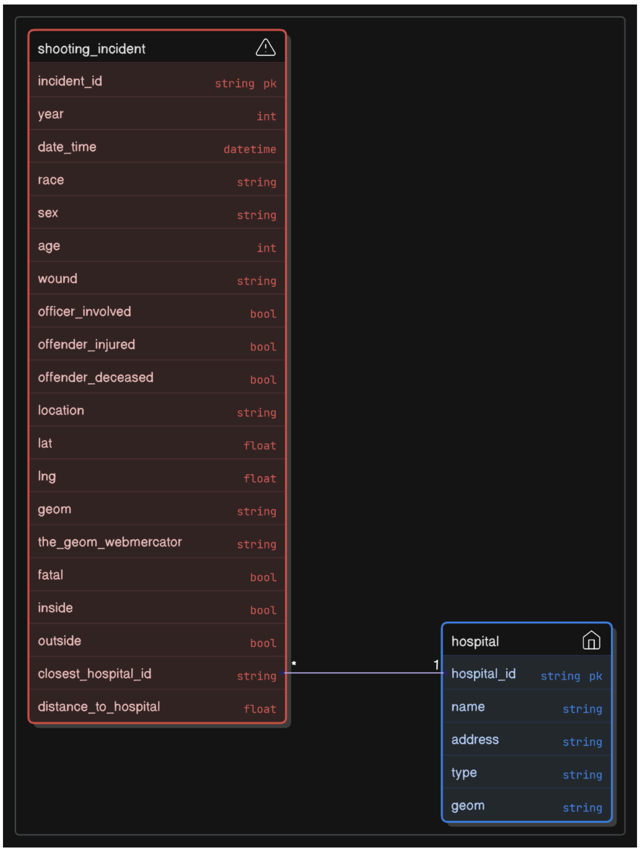
A Concept Map



The concept map represents the main components and workflow involved in developing a GIS-based artefact. It begins with data sources, including both spatial data (e.g., OpenStreetMap) and non-spatial data (e.g., text or numeric attributes). The processed data is stored in a spatially enabled DBMS (PostgreSQL with PostGIS), structured via tables derived from an ERD as shown below. SQL queries, including spatial functions, are used to extract insights. The data is then visualized using a GIS platform (like QGIS or ArcGIS), which generates layered maps based on the database. A user interface, built in python, provides interactivity for data input, query execution, and map visualizations, integrating GIS and DBMS functionalities.

ERM Diagram

The ERM diagram was created to display the relations observed within the data. It illustrates the connections between key entities such as incidents, locations, and hospitals. Spatial functions were used to determine the closest hospital to each incident based on geographic coordinates, and the resulting hospital's ID is stored in the incident table to establish a clear relationship. This relational structure supports efficient querying and spatial analysis, enabling the GIS system to link each event with nearby medical facilities and visualize proximity-based patterns across the dataset.



Stage 2 - Implementation

Spatial Data Analysis

Methodology

This GIS initiative explored the spatial connections between shooting events and the accessibility of hospitals across Philadelphia. To capitalize on our team's diverse skill sets, we divided the work into two interrelated streams that each addressed a critical aspect of the analysis. The first stream concentrated on the back-end tasks: importing and cleaning raw data, transforming coordinate information, conducting spatial joins, calculating distances, and performing advanced statistical and machine-learning analyses entirely in Python. Meanwhile, the second stream took responsibility for front-end and database components, using QGIS to design and generate clear, informative maps; configuring a PostgreSQL database (with PostGIS extensions) to store and query geographic features; and implementing a user interface that allows stakeholders to interactively explore both shooting locations and nearest-hospital relationships. Although each stream operated with a distinct technical focus, all efforts were synchronized to serve the overarching objective: to reveal spatial patterns of gun violence and assess how well emergency medical services are distributed throughout the city.

1. Data Collection and setup

Two primary datasets were obtained from the official U.S. government open data portal (Data.gov):

- **Shooting Records (cleaned_shootings.csv)** with events dates, latitudes and longitudes.
- **Hospital Inventory (hospitals_with_coordinates.csv)** listing facility IDs, names, and geographic coordinates.

These datasets were downloaded in standard formats (e.g., CSV, GeoJSON) suitable for spatial analysis.

```
[5]  shootings_df.columns

     Index(['the_geom', 'the_geom_webmercator', 'objectid', 'year', 'dc_key',
            'code', 'date_', 'time', 'race', 'sex', 'age', 'wound',
            'officer_involved', 'offender_injured', 'offender_deceased', 'location',
            'latino', 'point_x', 'point_y', 'dist', 'inside', 'outside', 'fatal',
            'lat', 'lng', 'distance_km'],
           dtype='object')

[6]  hospitals_df.columns

     Index(['OBJECTID', 'HOSPITAL_NAME', 'STREET_ADDRESS', 'CITY', 'STATE',
            'ZIP_CODE', 'PHONE_NUMBER', 'HOSPITAL_TYPE', 'lat', 'lng'],
           dtype='object')
```

Figure 1. Columns names of the shootings dataset(top) and the hospitals dataset (bottom).

Loading: Imported both datasets into pandas DataFrames and converted to GeoDataFrames (EPSG:4326).

Projections: Reprojected to Web Mercator (EPSG:3857) for accurate planar distance measurements.

2. Data Preprocessing

Using Python libraries such as *pandas, geopandas and shapely* the datasets were cleaned and prepared:

- **Cleaning:** Removed incomplete or incorrectly formatted records using pandas filters.
- **Geometry Conversion:** Transformed latitude/longitude pairs into *Shapely Point* geometries.
- **CRS Standardization:** Ensured both *GeoDataFrames* shared the same spatial reference before analysis.

3. Spatial Mapping

The next step involved mapping both datasets:

- **Base Map:** Created a Philadelphia basemap using Folium for interactive visualization.
- **Layer Plotting:** Plotted shooting incidents and hospital locations with geopandas and matplotlib for static maps, and Folium layers for interactive maps.

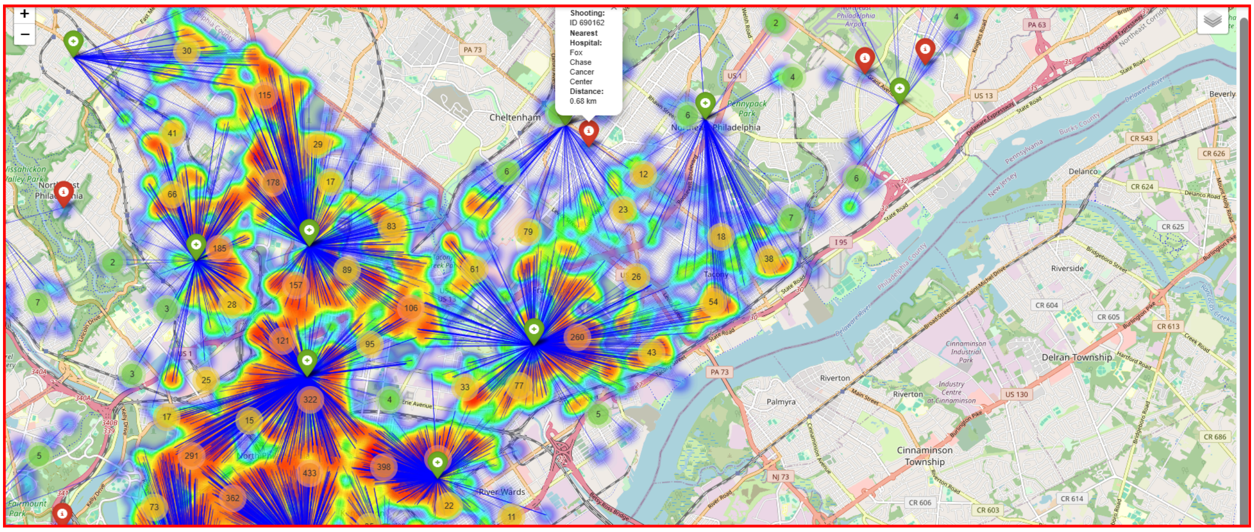


Figure 2. Combined visualization showing shooting incident density, hospital locations and incident-to-hospital connections.

A heatmap in the background reveals areas of high shooting concentration. Green markers indicate hospital locations, while colored circles represent clustered shooting counts. Blue lines connect individual shooting points to their nearest hospital, illustrating spatial accessibility. Pop Ups (visible at top center) display details for a selected shooting, including its ID, nearest hospital, and distance.

4. Nearest-Hospital Analysis

- **Spatial Indexing:** Built an R-tree index on projected hospital points for efficient neighbor searches.

```
    nearest_geom = nearest_points(row.geometry, merged)[1]
    nearest_row = candidates[candidates.geometry == nearest_geom].iloc[0]
    return nearest_row

nearest_list = [find_nearest(r, hospitals_proj, hospital_sindex) for _, r in shootings_proj.iterrows()]
nearest_hospitals = gpd.GeoDataFrame(nearest_list)
```

```
[15] nearest_hospitals
```

	OBJECTID	HOSPITAL_NAME	STREET_ADDRESS	CITY	STATE	ZIP_CODE	PHONE_NUMBER	HOSPITAL_TYPE	lat	lng	geometry
16	17	Pennsylvania Hospital	800 Spruce Street	Philadelphia	PA	19107	215-829-3000	General medical	39.9487	-75.1593	POINT (-8366695.004 4858490.301)
14	15	Thomas Jefferson University Hospitals Methodist	2301 South Broad St.	Philadelphia	PA	19148	215-952-9000	General medical	39.9207	-75.1595	POINT (-8366717.268 4854425.302)
14	15	Thomas Jefferson University Hospitals Methodist	2301 South Broad St.	Philadelphia	PA	19148	215-952-9000	General medical	39.9207	-75.1595	POINT (-8366717.268 4854425.302)
14	15	Thomas Jefferson University Hospitals Methodist	2301 South Broad St.	Philadelphia	PA	19148	215-952-9000	General medical	39.9207	-75.1595	POINT (-8366717.268 4854425.302)
13	14	Mercy Philadelphia Hospital	501 South 54th Street	Philadelphia	PA	19143	215-748-9000	General medical	39.9448	-75.2288	POINT (-8374431.709 4857924.005)
...	...	...	...	...	...	...	...	...	...	...	...
19	20	St. Christopher's Hospital for Children	3601 A Street	Philadelphia	PA	19134	215-427-5000	General medical	39.9925	-75.1133	POINT (-8361574.308 4864852.46)
19	20	St. Christopher's Hospital for Children	3601 A Street	Philadelphia	PA	19134	215-427-5000	General medical	39.9925	-75.1133	POINT (-8361574.308 4864852.46)
21	22	Temple University Hospital - Main Campus	3401 N. Broad Street	Philadelphia	PA	19140	215-707-2000	General medical	40.0118	-75.1456	POINT (-8365169.927 4867657.172)
21	22	Temple University Hospital - Main Campus	3401 N. Broad Street	Philadelphia	PA	19140	215-707-2000	General medical	40.0118	-75.1456	POINT (-8365169.927 4867657.172)
21	22	Temple University Hospital - Main Campus	3401 N. Broad Street	Philadelphia	PA	19140	215-707-2000	General medical	40.0118	-75.1456	POINT (-8365169.927 4867657.172)

16555 rows × 11 columns

Figure 3. Table of nearest hospitals for each shooting incident, showing facility attributes and geometry using R-tree indexing

Each row lists a shooting's closest hospital (object ID, name, address, coordinates) along with the hospital's spatial point in the projected CRS. This table summarizes the result of the nearest-neighbor join for all 16,555 shooting records.

- **Candidate Selection:** Queried bounding-box candidates per shooting; used Shapely's nearest_points() on merged geometries to find the closest hospital.
- **Distance Computation:** Calculated straight-line distances in meters and returned results to WGS84 for mapping.

5. Exploratory Spatial Patterns

- **Incident Heatmap:** Generated a density layer via Folium's HeatMap plugin to reveal concentration hotspot.

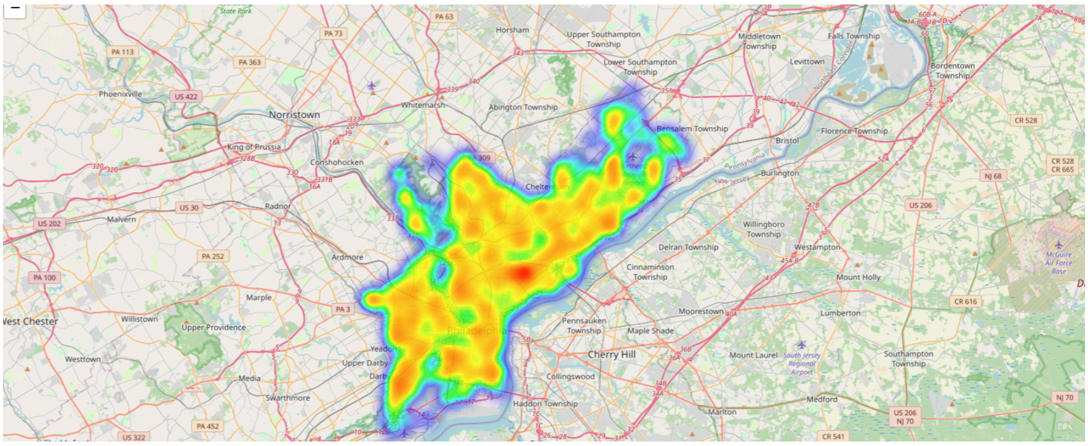


Figure 4. Heatmap showing density of shooting incident Philadelphia

- **DBSCAN Clustering:** Applied DBSCAN on projected coordinates with user-defined eps and min_samples, visualized via MarkerCluster.

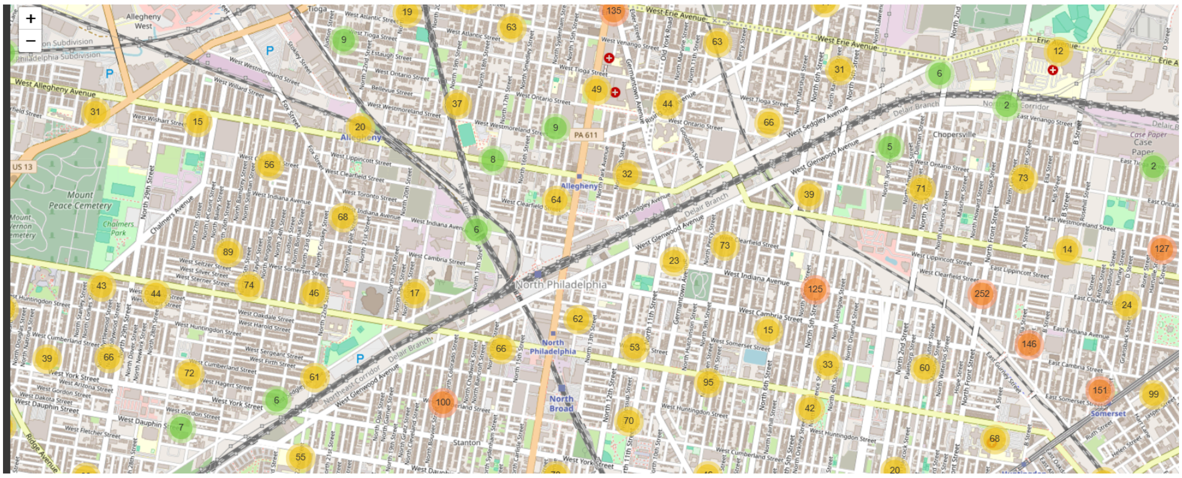


Figure 5. DBSCAN clusters of shooting incidents displayed as colored, numbered circle markers on a Philadelphia basemap.

- **Buffer Analysis:** Created 500m,1km, and 2 km buffers around hospitals; counted shootings within each buffer using spatial joins.

	nearest_hosp_id	shooting_count	radius_m
0	1	92	500
1	3	32	500
2	5	172	500
3	6	174	500
4	7	45	500

Figure 6: Buffer analysis

6. Density Surfaces and Service Areas

- **Kernel Density Estimation (KDE):** Computed a continuous density surface on a grid; visualised as Matplotlib contours and Folium ImageOverlay.

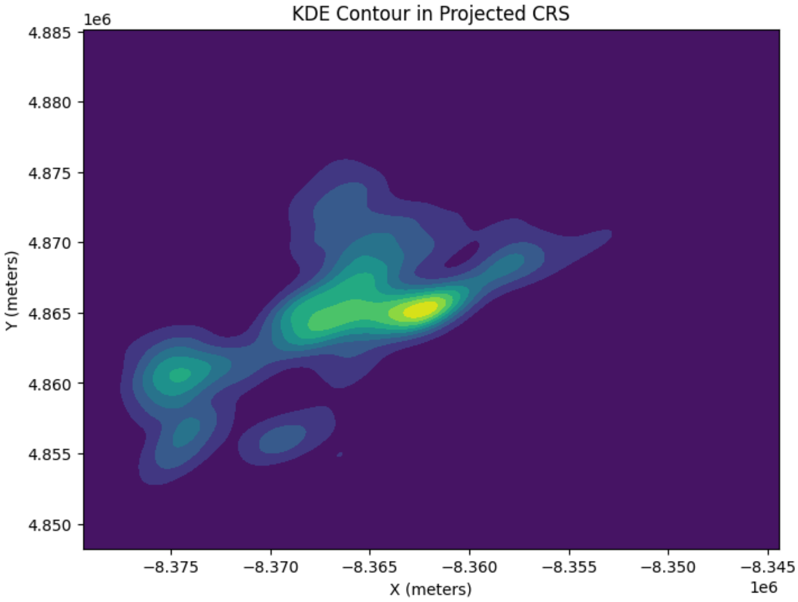


Figure 7. Contour plot of kernel density estimates for shooting locations in projected coordinates.

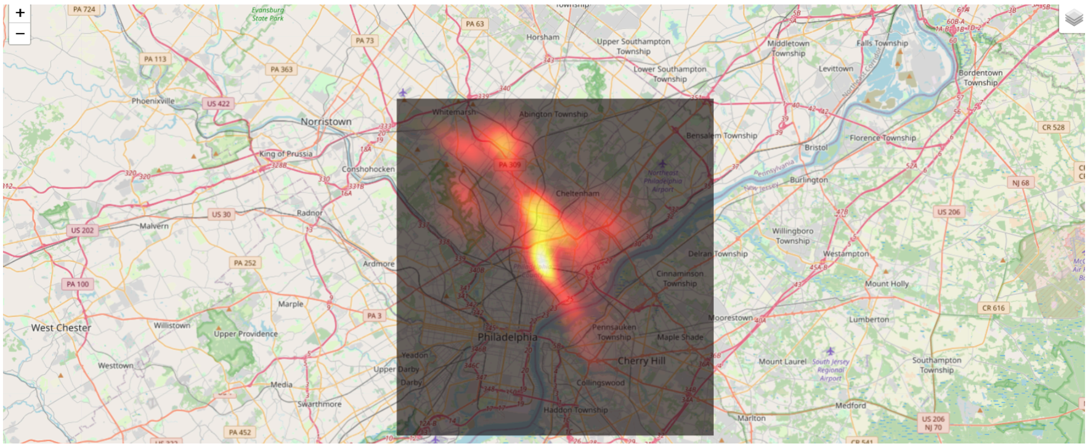


Figure 8. KDE heatmap overlay on a philadelphia basemap showing areas of high shooting density.

7. Spatial Statistics

- **Global Autocorrelation (Moran's I):** Created a KNN weight matrix (k = 8) to assess spatial clustering of nearest-hospital distances.
- ***Local Hotspots (Getis-Ord Gi):**** Divided the area into a 1 km fishnet grid, counted incidents per cell, and identified statistically significant hotspots.

```
81] # Identify hotspots (high Z-score and significant p)
    hotspots = grid[(grid['GiZ'] > 2) & (grid['GiP'] < 0.05)]
```

```
hotspots
```

	geometry	count	GiZ	GiP
124	POLYGON ((-8376420.856 4861228.051, -8375420.8...	74.0	2.210761	0.002
158	POLYGON ((-8375420.856 4858228.051, -8374420.8...	116.0	2.209184	0.002
159	POLYGON ((-8375420.856 4859228.051, -8374420.8...	100.0	3.240913	0.001
160	POLYGON ((-8375420.856 4860228.051, -8374420.8...	145.0	3.446758	0.001
161	POLYGON ((-8375420.856 4861228.051, -8374420.8...	151.0	2.965288	0.001
196	POLYGON ((-8374420.856 4859228.051, -8373420.8...	149.0	2.025612	0.002
197	POLYGON ((-8374420.856 4860228.051, -8373420.8...	177.0	2.576236	0.001
198	POLYGON ((-8374420.856 4861228.051, -8373420.8...	151.0	2.194910	0.002
234	POLYGON ((-8373420.856 4860228.051, -8372420.8...	49.0	2.457521	0.001

Figure 9. Table of hotspot grid cells with geometry, incident count, GiZ score and p-value.

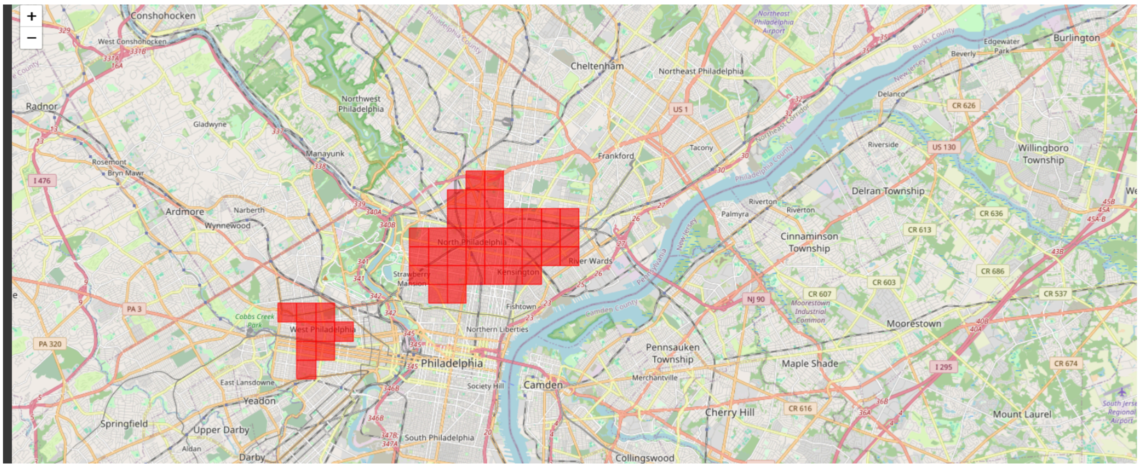


Figure 10. Red grid cells indicating significant shooting hotspots

The map displays 1km fishnet cells where the Getis-Ord Gi* statistics identified high-value clusters of shooting incidents. Each red polygon represents a grid cell with both a high Z-score and a p < 0.05, signifying areas of concentrated gun violence.

8. Directional and Temporal Analyses

- **Standard Deviational Ellipse:** Derived the two-standard-deviation ellipse to characterize the directional trend and dispersion of shootings.

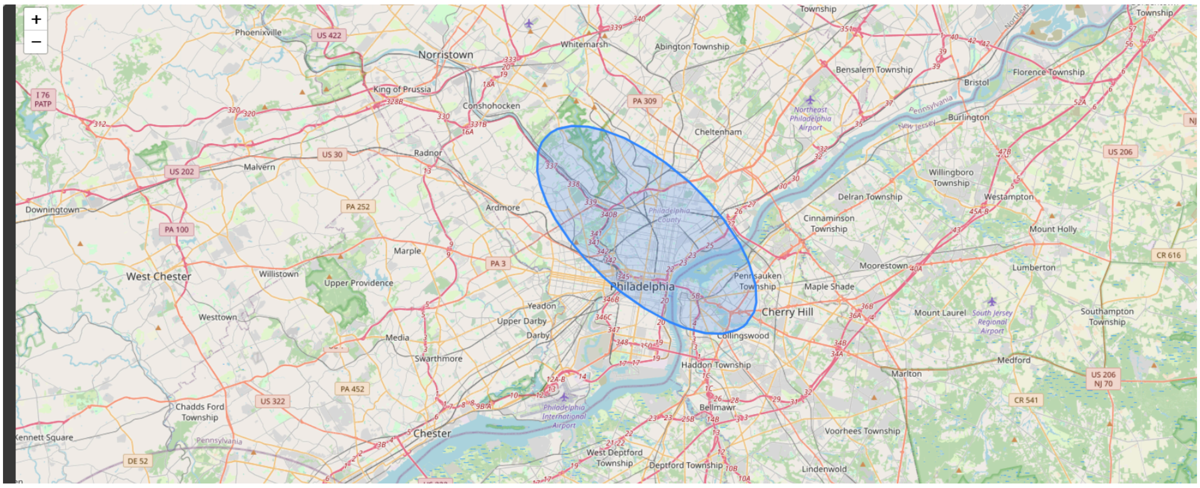


Figure 11. Standard deviational ellipse overlaid, illustrating the primary orientation and spread of shooting locations.

The blue ellipse represents a two-standard- deviation summary of incident coordinates, highlighting the directional trend and geographic dispersion of gun violence across the city.

- **Time Slider Animation:** Parsed event dates, filtered valid records, and animated incidents over time using Folium's TimestampedGeoJson.

Each blue marker represents a shooting event, and the timeline control at the bottom allows stepwise visualization by date. As the slider moves, incidents appear in chronological order, revealing peaks and lulls in gun violence over time.

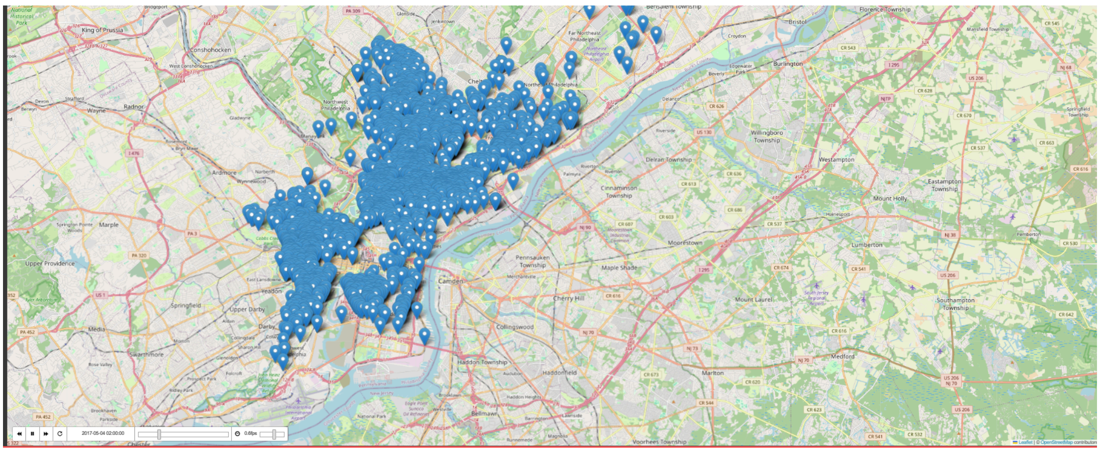


Figure 12. Time -slider animation of shooting incidents across Philadelphia, showing temporal progression on the map.

9. Machine Learning Insights

K-Means Clustering (Hospitals): Standardized hospital metrics (incident count, mean distance), segmented into three clusters, and visualised in scatter plots and on the map.

Each point represents a hospital, with its x-coordinate showing how many nearby shootings it handled and the y-coordinate indicating the mean distance from incidents. The colors correspond to clusters identified by K-means, highlighting facilities that share similar accessibility and incident load characteristics.

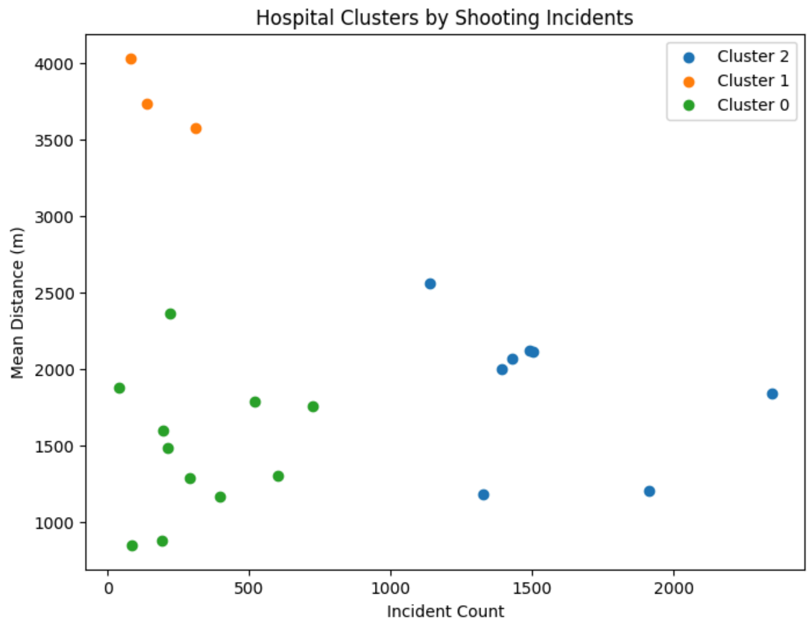


Figure 13. Scatter plot of hospitals grouped by shooting incident count and average distance, colored by K -means cluster.

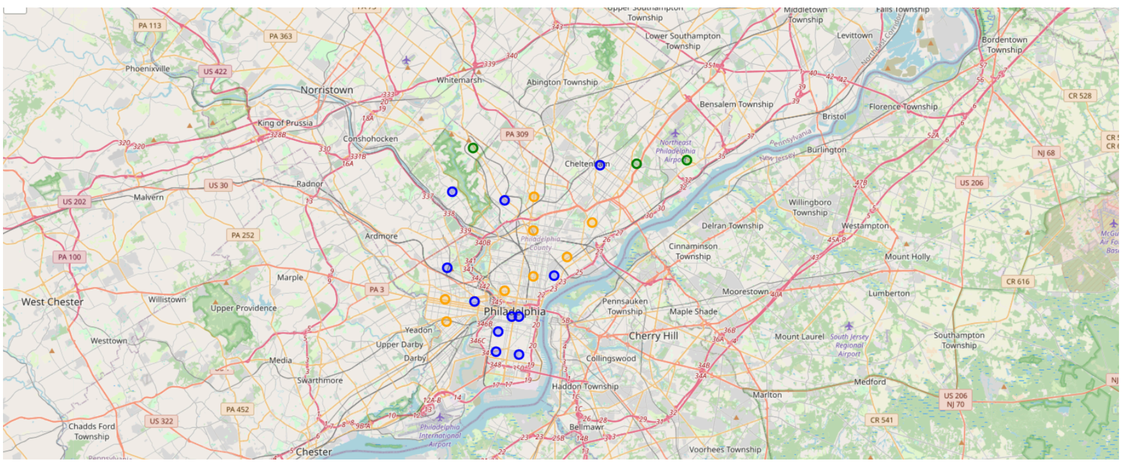


Figure 14. Map of hospitals colored by K-means cluster.

Each circle represents a hospital, with color indicating its cluster membership based on nearby shooting incident counts and average distances. This visualization highlights spatial groupings of facilities that share similar demand and accessibility profiles.

Random Forest Hotspot Classification: Labeled fishnet cells as hotspots/non-hotspots; trained a classifier on centroid coordinates, evaluated performance, and mapped probability predictions.

```
111] # Evaluate
     from sklearn.metrics import classification_report
     y_pred = rf.predict(X_test)
     print(classification_report(y_test, y_pred))

                   precision    recall  f1-score   support

               0       0.98      1.00      0.99       312
               1       0.88      0.58      0.70        12

        accuracy                           0.98       324
       macro avg       0.93      0.79      0.85       324
    weighted avg       0.98      0.98      0.98       324
```

Figure 15. Random Forest classifier evaluation output displaying precision, recall, f1-score and accuracy.

The figure 15. reports performance metrics for non-hotspot (0) and hotspot (1) classes on the test set. Class 0 achieves near-perfect precision and recall, while Class 1 shows lower recall, indicating some under-detection of hotspots despite high overall accuracy (98%).

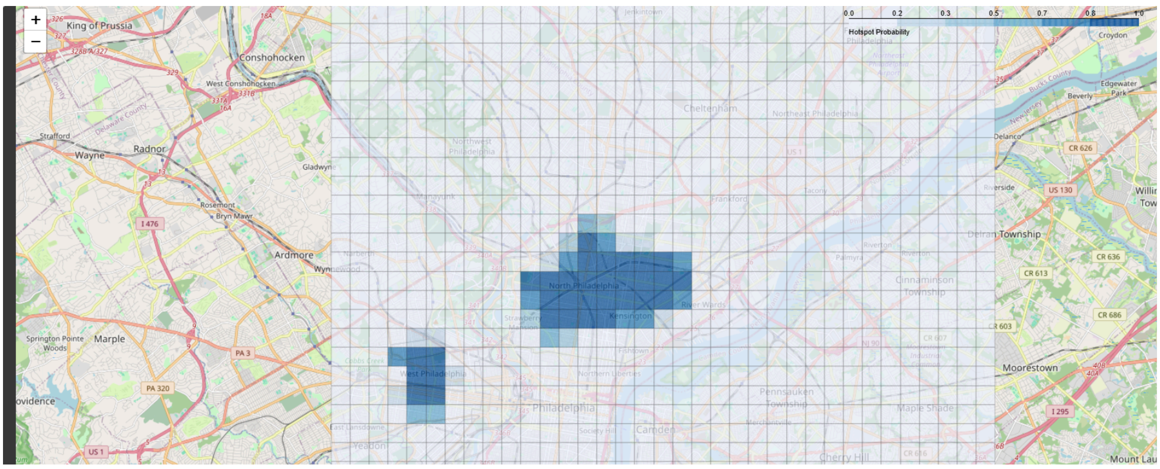


Figure 16. Choropleth map depicting predicted hotspot probabilities across grid cells.

The grid cells are shaded from light to dark based on the Random Forest’s estimated probability of each cell being a shooting hotspots. Darker hues indicate higher likelihoods, highlighting areas forecasted to experience significant incidents.

Hospital Incident Summary and Visualisation

	nearest_hosp_id	incident_count	mean_distance_m	HOSPITAL_NAME	STREET_ADDRESS	CITY	STATE	ZIP_CODE	PHONE_NUMBER	HOSPITAL_TYPE	lat	lng
0	1	1429	2066.723746	Aria Health- Frankford Campus	4900 Frankford Avenue	Philadelphia	PA	19124	215-831-2000	General medical	40.0178	-75.0895
1	2	81	4032.649953	Aria Health- Torresdale Campus	10800 Knights Road	Philadelphia	PA	19114	215-612-4000	General medical	40.0634	-74.9990
2	3	521	1790.752642	Belmont Center for Comprehensive Treatment	4200 Monument Road	Philadelphia	PA	19131	215-877-2000	Behavioral health	39.9845	-75.2282
3	4	138	3732.040666	Chestnut Hill Hospital	8835 Germantown Avenue	Philadelphia	PA	19118	215-248-8200	General medical	40.0723	-75.2034
4	5	600	1302.033471	The Children's Hospital of Philadelphia	34th Street and Civic Center Boulevard	Philadelphia	PA	19104	215-590-1000	General medical	39.9597	-75.2024
5	6	1502	2109.791803	Einstein Medical Center - Philadelphia	5501 Old York Rd	Philadelphia	PA	19141	215-456-7890	General medical	40.0365	-75.1451
6	7	222	2363.678998	Fox Chase Cancer Center	333 Cottman Avenue	Philadelphia	PA	19111	215-728-6900	General medical	40.0596	-75.0818
7	8	1392	1996.879898	Girard Medical Center	8th St. & Girard Avenue	Philadelphia	PA	19122	215-787-2000	Long term care	39.9780	-75.1459
8	12	290	1289.536101	Kindred Hospital South Philadelphia	1930 S. Broad Street	Philadelphia	PA	19145	267-570-5200	Long term care	39.9227	-75.1812
9	13	87	850.190172	Magee Rehabilitation Hospital	1513 Race Street	Philadelphia	PA	19102	215-587-3333	Rehabilitation	39.9489	-75.1661
10	14	1490	2124.052627	Mercy Philadelphia Hospital	501 South 54th Street	Philadelphia	PA	19143	215-748-9000	General medical	39.9448	-75.2288

Figure 17. Shooting and Hospital datasets are merged

Top 10 hospitals by incident counts

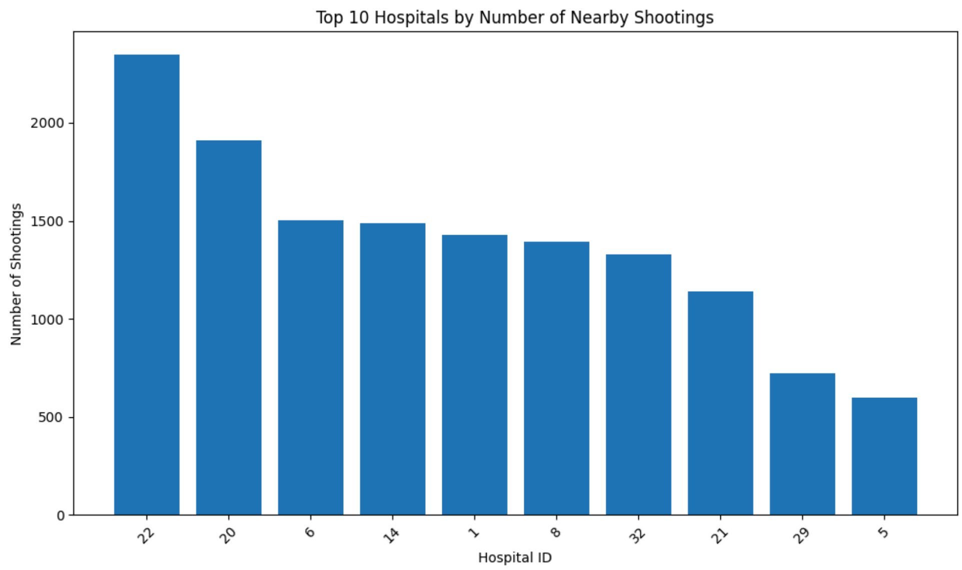


Figure 18. Top 10 Hospitals by number of nearby shootings.

Each circle's radius in figure 19 is proportional to how many shootings occurred closest to that hospital, with an example popup showing Hospital 20 handled 1,911 incidents at an average distance of 1,204 m. This map highlights facilities bearing the highest incident loads.

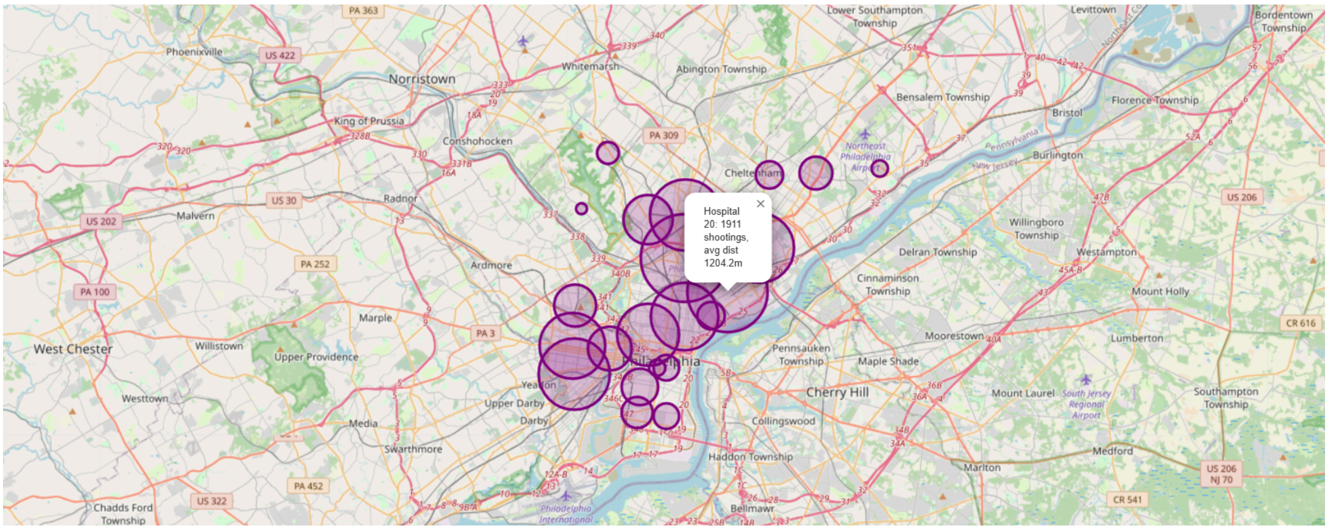


Figure 19. Circle markers representing hospitals, sized by the number of nearby shooting incidents and color-coded in purple.

Link:
https://colab.research.google.com/drive/1pPGoEK-eEQ40B5Ey6oWd43G_GOXqYesZ?usp=sharing

Set-up

To properly set up the selected dataset, initially, Docker [10] was used to create a spatially enabled database. After importing the dataset into the database, spatial functions were applied to calculate the nearest hospital for each shooting incident and determine the distance between them. The code used is as follows:

```sql
-- Add geometry column to hospitals
ALTER TABLE hospitals ADD COLUMN geom geometry(Point, 4326);

-- Populate it
UPDATE hospitals
SET geom = ST_SetSRID(ST_MakePoint(lng, lat), 4326);


ALTER TABLE shooting_incidents ADD COLUMN geom geometry(Point, 4326);

UPDATE shooting_incidents
SET geom = ST_SetSRID(ST_MakePoint(lng, lat), 4326);

CREATE INDEX hospitals_geom_idx ON hospitals USING GIST (geom);
CREATE INDEX shootings_geom_idx ON shooting_incidents USING GIST (geom);

ALTER TABLE shooting_incidents
ADD COLUMN closest_hospital text,
ADD COLUMN distance_to_hospital double precision;

UPDATE shooting_incidents s
SET closest_hospital = h.hospital_name,
    distance_to_hospital = ST_Distance(s.geom::geography,
h.geom::geography) / 1000  -- in km
FROM (
    SELECT s.objectid AS sid, h.hospital_name, h.geom
    FROM shooting_incidents s
    JOIN LATERAL (
        SELECT hospital_name, geom
        FROM hospitals
        ORDER BY s.geom <-> geom
        LIMIT 1
    ) h ON true
) AS h
WHERE s.objectid = h.sid;
```

After setting the data, QGIS [8] was connected to the database so that layers can be created from data placed in the database.

User Interface

When the created python script is run using OSGeo4W (which is the terminal used in QGIS [8]), the user interface shows up as seen below in figure 3.1

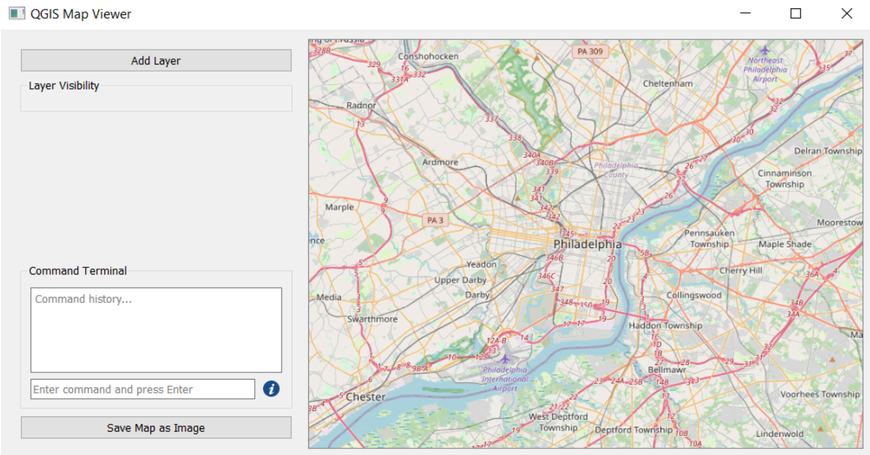


Figure 3.1 - Empty User Interface

The map is interactive, since the developed interface is being opened using OpenStreet Maps [11]. Additionally, since the chosen data is about shootings happening in Philadelphia, USA, the interface automatically zooms into that location when opened. The interface has some functionality, such as:

- Adding predefined layers stored within a local directory.
- Adding layers dynamically by filtering the dataset.
- Select/ deselect layers.
- Save map as an image.
- Info button which tells the user how to use the command terminal (when hovering over it).

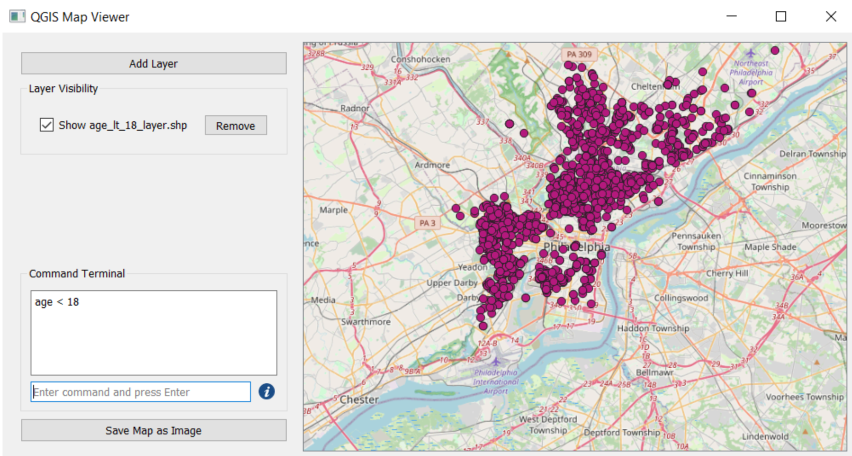


Figure 3.2 - User Interface with filtered data

Figure 3.2 shows how when the user placed “age < 18” in the command terminal, the dataset was filtered depending on this command. Afterwards, the layer is being created by QGIS, and the user is prompted to add the layer to the interface, resulting in the layers shown in figure 3.2. This command terminal may be useful in spatial analysis, where the user can quickly filter out data points, allowing the user to understand the contents of the dataset. For example, the command in figure 3.2 can be used to help analyse shooting crimes in Philadelphia which happened by people who are under 18 years old. This can be useful in studies which investigate whether teenagers and children have become more or less violent over a period of time.

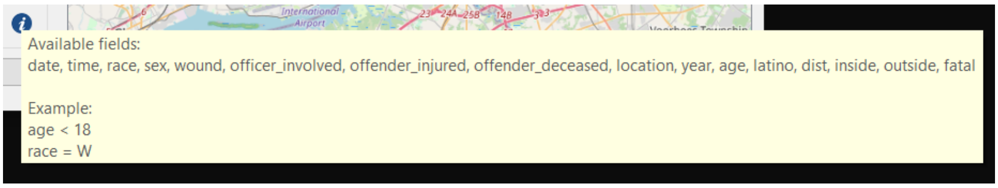


Figure 3.3 - Hovering over information button

Assuming the user does not know the contents of the dataset, the information button was implemented in the interface so the user can know which commands can be used to filter out data. By hovering over the button, the prompt shown in figure 3.3 shows up, which identifies what variables can be used. Two examples are also given to further clarify how to use the terminal.

To input layers into the interface, there are two options which can be used. The first one was to filter out data points dynamically as shown in figure 3.2, and the other is to use the “Add Layer” button shown in figure 3.1 and 3.2. Once pressed, a prompt with already defined layers comes up. This was implemented to show complex layers using a .qml file, as shapefiles can only store data points.

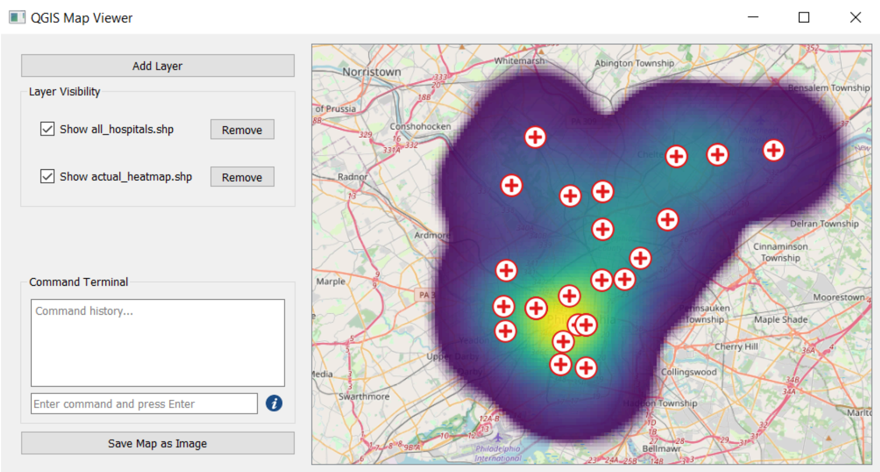


Figure 3.4 - Hospitals and heatmap layers

Figure 3.4 displays a layer which was previously created using QGIS. After the .qml files were exported, they were placed in the interface to show the heatmap seen in figure 3.4. This heatmap visualises each shooting's distance to its nearest hospital in Philadelphia. This type of information can be useful for city planners to investigate where a hospital should be placed for example. Regardless of where a victim is shot, this type of visualisation can be used to reduce injuries and fatalities by helping identify the nearest medical facilities. In the U.S., 38% of gun-related deaths are classified as homicides [1], highlighting the urgency of rapid medical response.

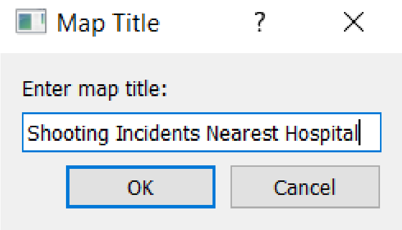


Figure 3.5 Inserting map title

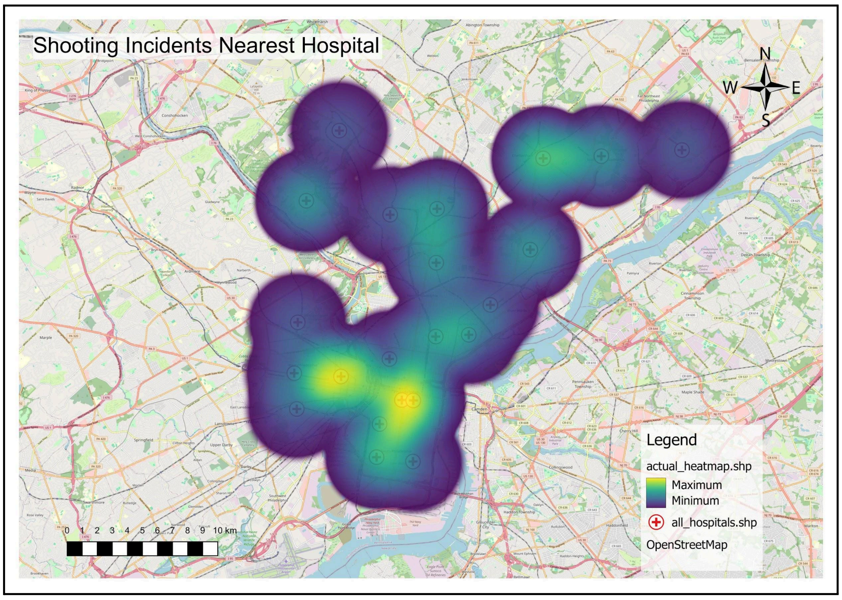


Figure 3.6 Visualisation of heatmap on a map

Once the user is satisfied with the selected layers, the "save map as image" may be used to convert the layers into a map. If the button is pressed, the user can input the map's title as seen in figure 3.5, and afterwards the map in figure 3.6 is output. The map also includes additional details, such as a compass, map scale and legend, resulting in the map shown in figure 3.6.

Despite this, the user interface has some limitations, which include:

- Users cannot use add/ or functions.
- Even if the layer is unselected, it will show in the exported map.
- Users have no control over the colour of the data points, so if multiple filters are applied, the data points may be difficult to distinguish.

Future improvements to the interface could focus on removing these limitations to create a more flexible and user-friendly experience for data visualisation.

References

[1] E. Gramlich and J. Schaeffer, “What the data says about gun deaths in the U.S.,” *Pew Research Center*, Mar. 5, 2025. [Online]. Available: https://www.pewresearch.org/short-reads/2025/03/05/what-the-data-says-about-gun-deaths-in-the-us/

[2] W. McKinney, “Data structures for statistical computing in Python,” in *Proc. 9th Python in Science Conf.*, 2010, pp. 51–56. [Online]. Available: https://pandas.pydata.org/

[3] K. Jordahl, J. Van den Bossche, et al., “GeoPandas: Python tools for geographic data,” 2020. [Online]. Available: https://geopandas.org/

[4] S. Gillies, “Shapely: Manipulation and analysis of geometric objects,” 2007. [Online]. Available: https://shapely.readthedocs.io/

[5] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007. [Online]. Available: https://matplotlib.org/

[6] J. Makin and S. Hinder, “Contextily: Context maps for Python,” 2020. [Online]. Available: https://contextily.readthedocs.io/

[7] P. Virtanen *et al.*, “SciPy 1.0: Fundamental algorithms for scientific computing in Python,” *Nat. Methods*, vol. 17, pp. 261–272, 2020. [Online]. Available: https://scipy.org/

[8] QGIS Development Team, *QGIS Geographic Information System*, Open Source Geospatial Foundation Project, 2025. [Online]. Available: https://qgis.org/

[9] pgAdmin Development Team, *pgAdmin: Open Source Administration and Development Platform for PostgreSQL*, 2025. [Online]. Available: https://www.pgadmin.org/

[10] Docker, Inc., “Docker Desktop: The #1 Containerization Tool for Developers,” *Docker*, 2025. [Online]. Available: https://www.docker.com/products/docker-desktop/ [Accessed: 13-Jun-2025].

[11] OpenStreetMap contributors, “Map of area centered at latitude 35.9448°, longitude 14.3810°,” *OpenStreetMap*, 2004–2025. [Online]. Available: https://www.openstreetmap.org/#map=11/35.9448/14.3810 [Accessed: 13-Jun-2025].