Insights from Analysis: Fatal Police Shootings in the U.S.

Kruthika Reddy Murthy – 02109003

Neeraj Reddy Neelapu – 02077978

1. The Issues

This report addresses the patterns and insights derived from the analysis of fatal police shootings in the United States. The goal is to understand the demographics of individuals involved, the circumstances surrounding the incidents, and their geographical distribution.

The key issues addressed in this report include:

Incidents by race: To understand the distribution of fatal police shootings across different racial groups, a bar chart was utilized.

Age distribution of subjects: The age distribution of individuals involved in fatal police shootings was examined, and the findings were presented in the form of a histogram.

Threat Levels and Armed Status: Investigating the occurrence of each threat level perceived during these incidents and exploring the relationship between age and armed status through a scatter plot.

Proximity to Police Stations: We have Examined the geographical distribution of fatal police shootings involved calculating the distance between the incidents and nearby police stations using KNN (K Nearest Neighbors).

Mapping Incident Locations: Utilizing geospatial mapping, this report visualizes incident locations for understanding where these incidents cluster across regions.

Age and Manner of death: Identification of patterns within the age and manner of death variables was addressed using DBSCAN clustering. This technique allows for the identification of distinct groups or clusters within the dataset, also provides insights into the characteristics and commonalities among incidents.

2. Findings

The analysis of fatal police shootings in the United States has uncovered several key insights. First, we examined how many people from different racial backgrounds were involved, aiming to highlight any notable disparities in these incidents. Additionally, we explored the age distribution of individuals involved, providing insights into the demographics of those affected. Another aspect focused on understanding the relationship between age and whether individuals were armed during these incidents, helping us understand if there's a connection between age and the likelihood of being armed during such situations. Finally, we used a KNN (K-Nearest Neighbors) algorithm to calculate the average distance from the incident to the nearest police station.We created maps to visually show where these incidents happened. This helps us see if there are specific areas or regions where fatal police shootings are more concentrated. The application of DBSCAN clustering unveiled distinct patterns in age distribution and manner of death, providing deeper insights into the characteristics of different incident clusters.

3. Discussion

The findings have important implications for law enforcement and public safety. Firstly, we observed disparities in fatal shootings among different racial groups, signalling a need for targeted interventions to address these discrepancies. Understanding the age distribution of individuals involved in these incidents provides valuable insights for shaping policies and training programs. Geospatial insights, such as where incidents cluster, offer critical information for resource allocation and strategic planning, allowing law enforcement to focus efforts where they are most needed. Additionally, assessing the proximity of incidents to police stations highlights areas where enhanced law enforcement presence or quicker response times may be crucial for improving overall public safety. The utilization of DBSCAN clustering has revealed distinct patterns in the age distribution and manner of death. Understanding these patterns is crucial for developing policies. For example, clusters with specific age ranges and consistent manners of death may need specialized training programs to address the unique characteristics of these groups. The findings from DBSCAN clustering contribute to a more comprehensive understanding of the underlying structures within the data, uncovering potential trends or outliers that may influence the interpretation of fatal police shootings.

4. Appendix A: Method

Data Collection:

The data for this analysis was obtained from the Washington Post data repository on fatal police shootings in the United States. The repository provides a comprehensive dataset with information on incidents, including demographic details, circumstances, and geographical coordinates. The latitude and longitude of US police stations were sourced separately to enable geospatial analysis. The primary motivation for selecting this dataset is to contribute to a better understanding of the dynamics surrounding fatal police shootings.

Variable Creation:

1. Demographic Variables:

   Race, age, gender, city, and state, threat level, armed status was directly extracted from the dataset.

   - Race: Indicates the racial background of the individual involved in the fatal police shooting. The 'race' column includes the following unique values:
     - 'A': Asian
     - 'W': White
     - 'H': Hispanic
     - 'B': Black
     - 'O': Other
     - 'N': Native American
   - Age: Represents the age of the person at the time of the incident.
   - Gender: Specifies the gender of the individual.
   - City and State: Identify the location (city and state) where the incident occurred.

- Threat Level: Describes the perceived threat level during the incident, categorized into different levels.
- Manner of Death: Describes the way the incident resulted in death, categorized into 2 types i.e. "shot," "shot and Tasered".
- Armed Status: Indicates whether the individual involved in the fatal police shooting was armed during the incident. Types of armed attacks may include:
  - Armed Attack: The individual was armed with a weapon.
  - Other: The nature of the threat was different from a conventional armed attack.
  - Undetermined: The type of threat was unclear or not determined.

2. Geographical Variables:

Latitude and longitude coordinates were crucial for geospatial mapping.

- Latitude and Longitude: Geographic coordinates pinpointing the exact location of the incident.
- Distance to Nearest Police Station: Calculated using the k-NN algorithm, representing the distance from the incident location to the nearest police station.

Analytic Methods:

1. Descriptive Statistics:

Utilized to provide an overview of the dataset, including counts, means, standard deviation, minimum and maximum of variables and distributions.

2. Data Cleaning:

Addressed missing values and duplicates to ensure data integrity.

3. Data Visualization:

Employed bar charts to visualize the count of incidents by race, histograms to count the distribution of age, scatter plots to understand the relationship between age and armed status, and pie charts to visualize the occurrence of each threat level using Matplotlib and Seaborn to visualize patterns and relationships.

4. Geospatial Mapping:

Implemented Folium to create interactive maps illustrating incident locations and police station proximity.

Also illustrated the geospatial visualization for mapping of police shootings.

5. k-Nearest Neighbors (k-NN):

The k-Nearest Neighbors (k-NN) algorithm was employed to determine the nearest police stations for each incident. The algorithm operates on the principle that similar incidents are close to each other in a multidimensional space, and it classifies or predicts the group of an incident based on the majority group of its k-nearest neighbors.

The k parameter, representing the number of neighbors considered, was chosen to be 1, providing the single nearest police station for each incident. The algorithm used the geographical coordinates of both incidents and police stations to calculate distances and determine proximity. Visualization was achieved by mapping the incidents and their respective nearest police stations using Folium, a Python library for creating interactive maps.

6. DBSCAN Clustering

The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm was employed for clustering analysis. This algorithm identifies clusters based on the density of data points in the feature space, allowing for the detection of patterns and outliers. In this context, age and the count of the manner of death were used as features for clustering.

The algorithm categorized data points into clusters, with each cluster assigned a unique identifier (0, 1, etc.), and outliers labeled as -1. Visualization was performed using a histogram, where the x-axis represents age, the y-axis represents the count of the manner of death, and bars are color-coded by cluster. The findings from the clustering analysis were interpreted based on the age distribution and the manner of death within each cluster

5. Appendix B: Results

The results section aims to provide a comprehensive overview of the descriptive statistics, insights and key findings derived from the analysis of fatal police shootings in the United States.

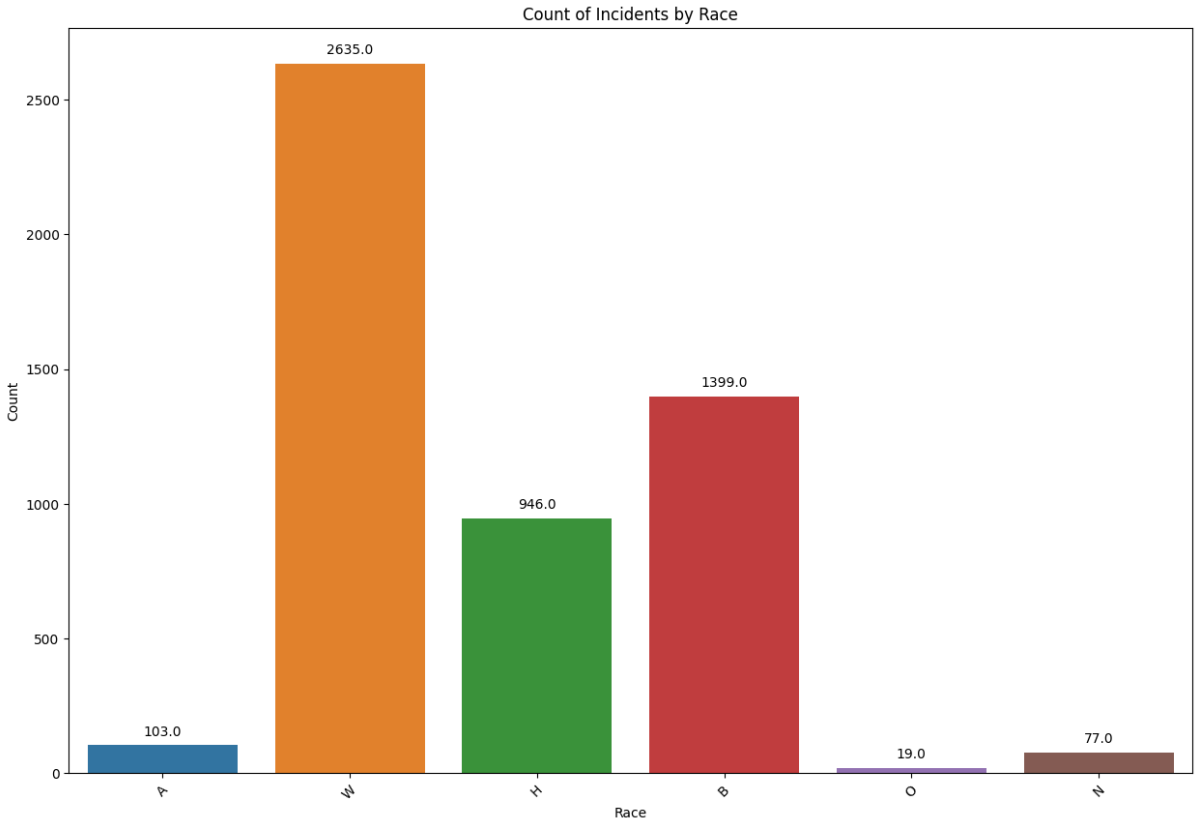
Summary statistics for numerical columns are:

Summary statistics:

	id	age	longitude	latitude
count	5179.000000	5179.000000	5179.000000	5179.000000
mean	3542.199073	36.738366	-96.675573	36.676024
std	2197.010309	12.691245	16.442735	5.338691
min	3.000000	6.000000	-158.137000	19.498000
25%	1618.000000	27.000000	-111.947000	33.488000
50%	3429.000000	35.000000	-93.466000	36.155000
75%	5302.500000	45.000000	-82.946500	39.999000
max	8677.000000	91.000000	-68.014000	71.301000

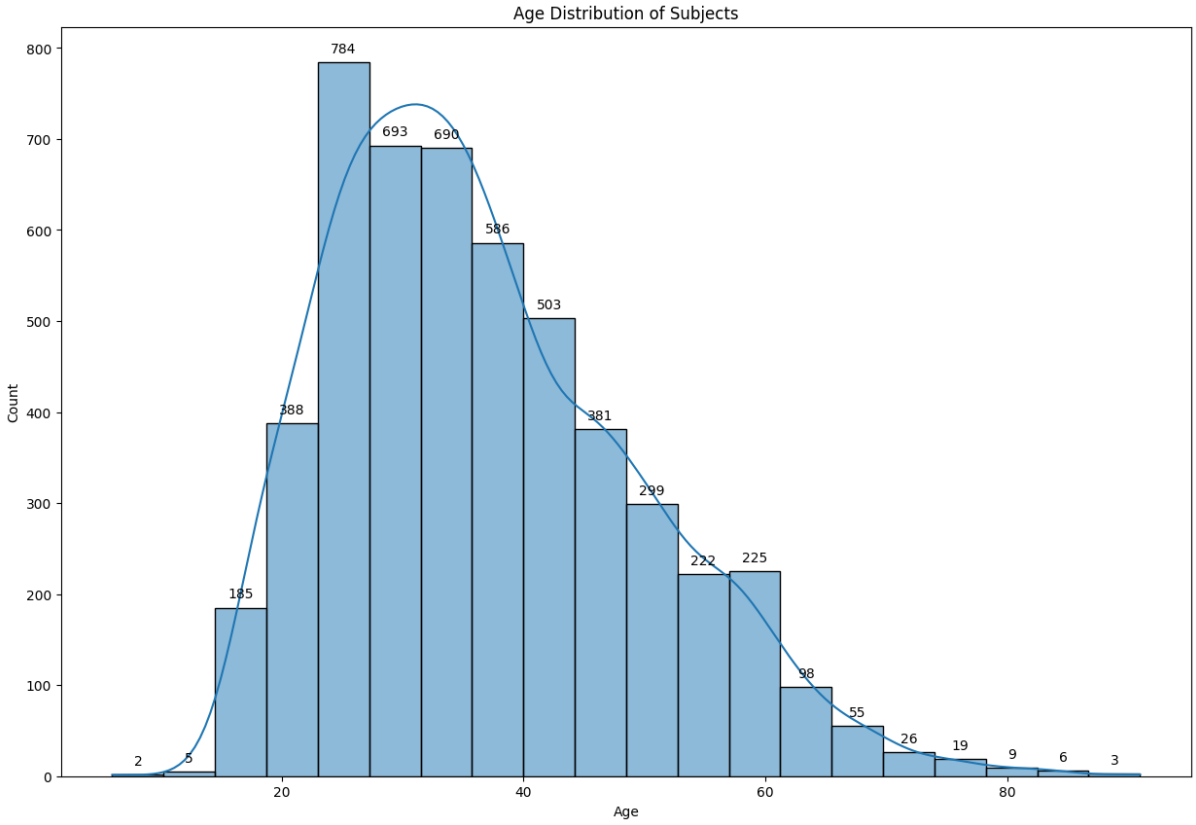
This is the summary statistics which shows all the statistical calculation values like mean, median, mode, standard deviation. We can know that the average age for the shooting dataset is around 37 years. The age has a standard deviation of approximately 12.69, suggesting a moderate amount of variability in the ages of individuals involved in incidents. 25% of individuals involved in fatal police shootings are aged 27 years or younger. 75% of individuals involved in fatal police shootings are aged 45 years or younger. The median age is 35 years, representing the middle point of the age distribution.

1.Bar chart to visualize the count of incidents by race



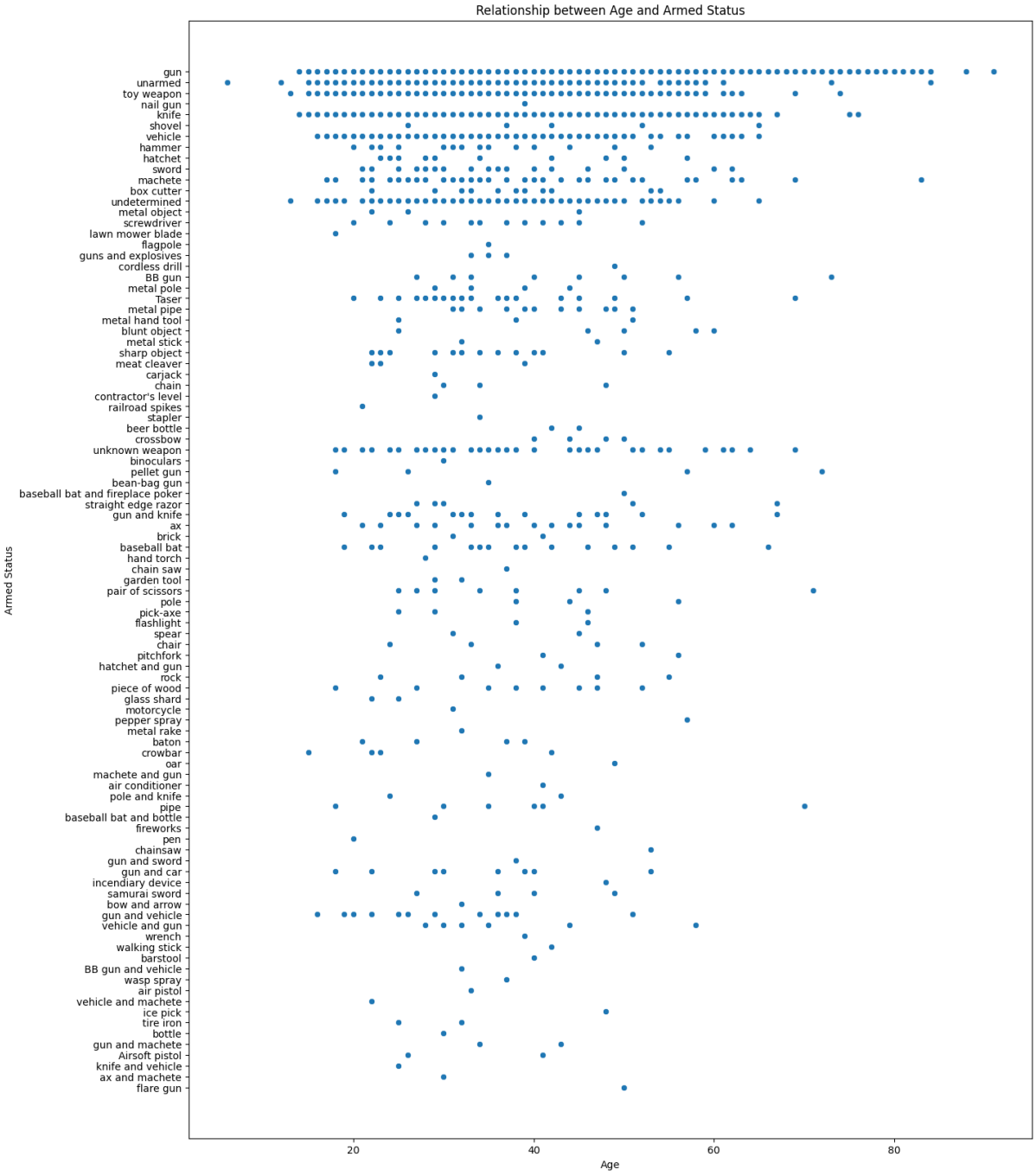
The bar chart provides a visual representation of the count of incidents by race. Notable disparities are observed among different racial groups, with 'W' (White) having the highest count, followed by 'B' (Black) and 'O' (Other) being the lowest.

2. Histogram to visualize the distribution of age



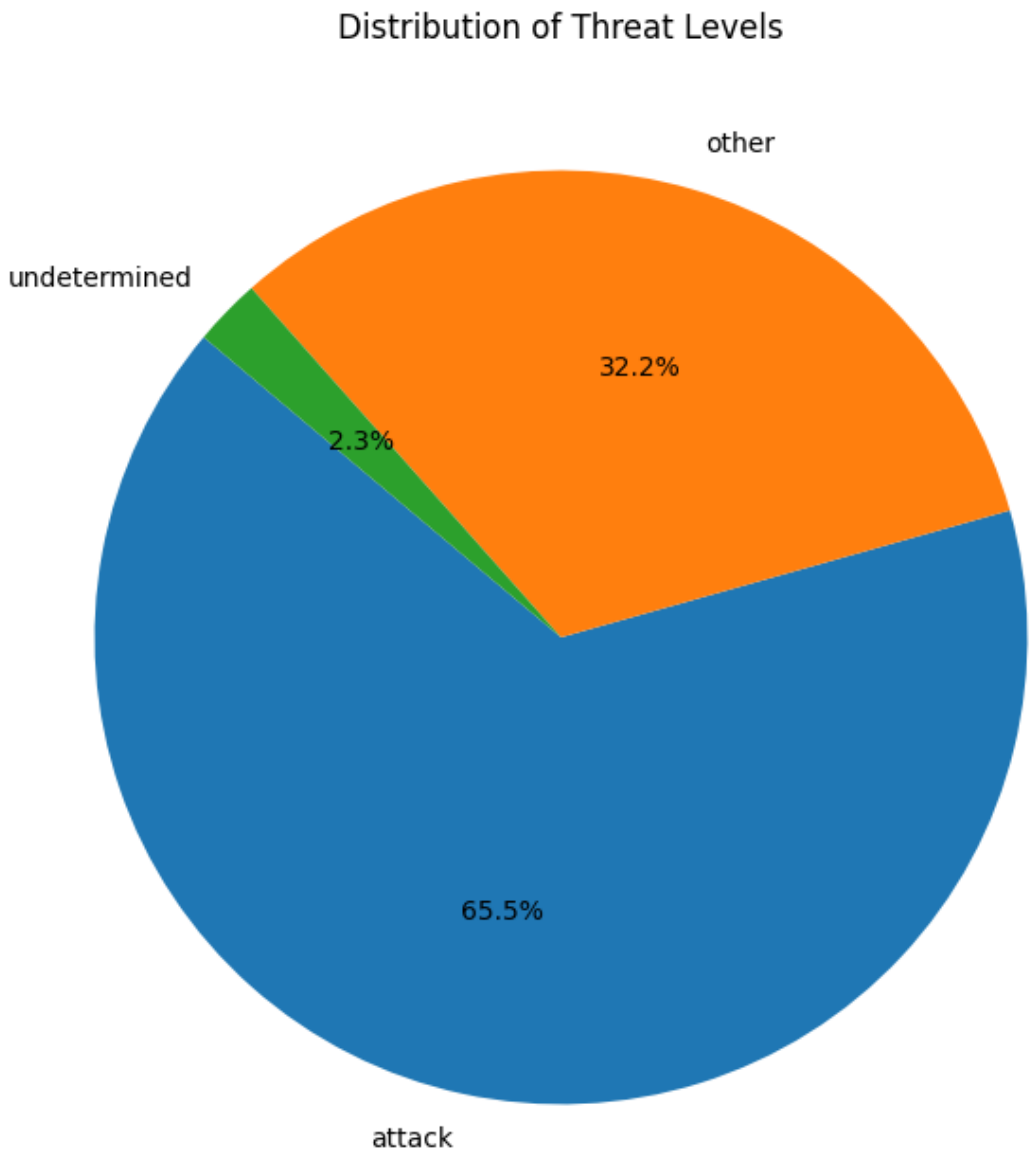
The histogram illustrates the distribution of ages among individuals involved in fatal police shootings. Many incidents involve individuals in their 20s and 40s, with a peak around 25 years old.

3. Scatter plot to explore the relationship between age and armed status



The scatter plot depicts the relationship between age and armed status. Notably, each data point corresponds to an armed status, indicating that individuals of every age are associated with an armed status.

4. Pie chart to Count the occurrences of each threat level



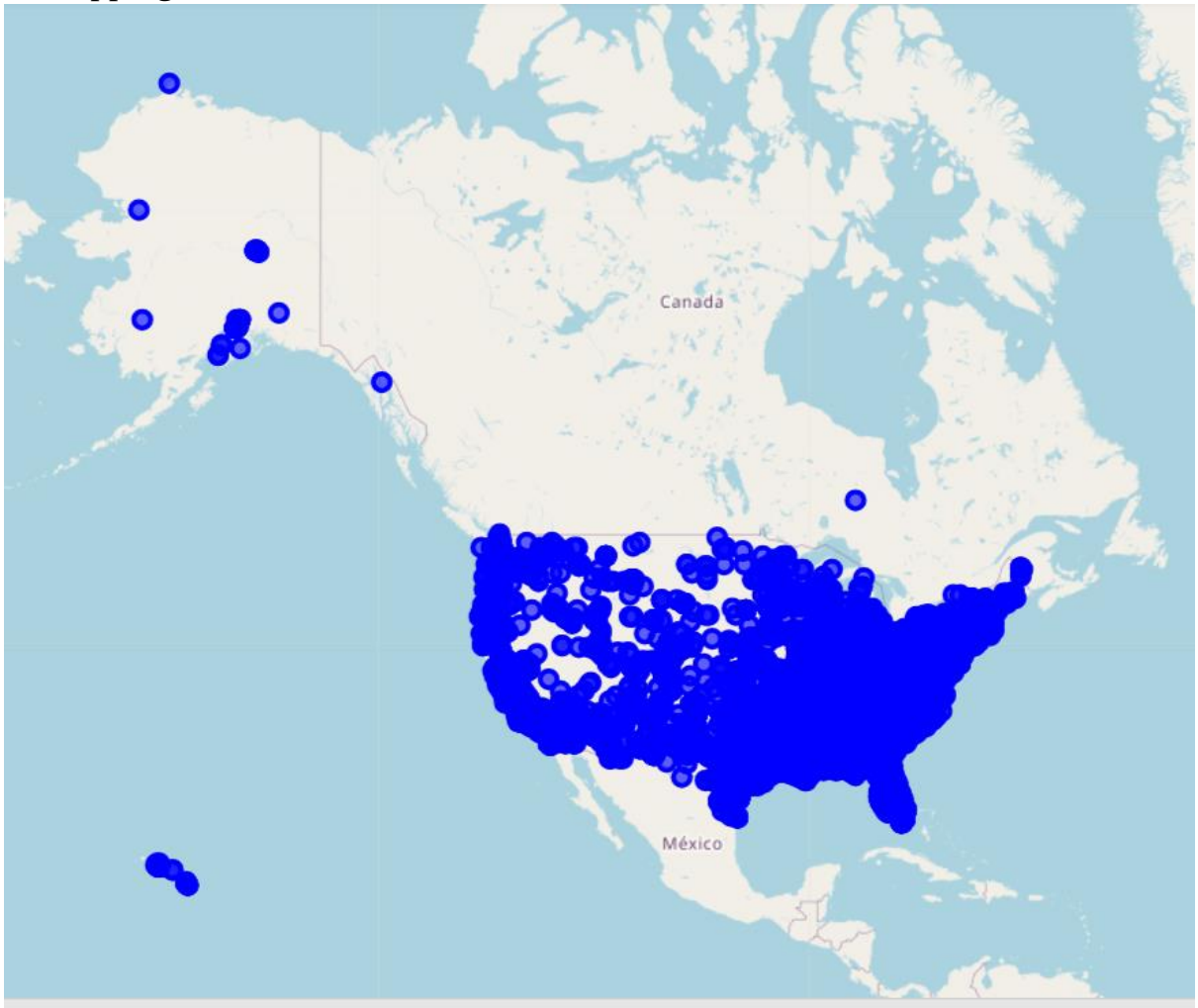
The pie chart visually represents the distribution of threat levels, highlighting the proportion of each category.

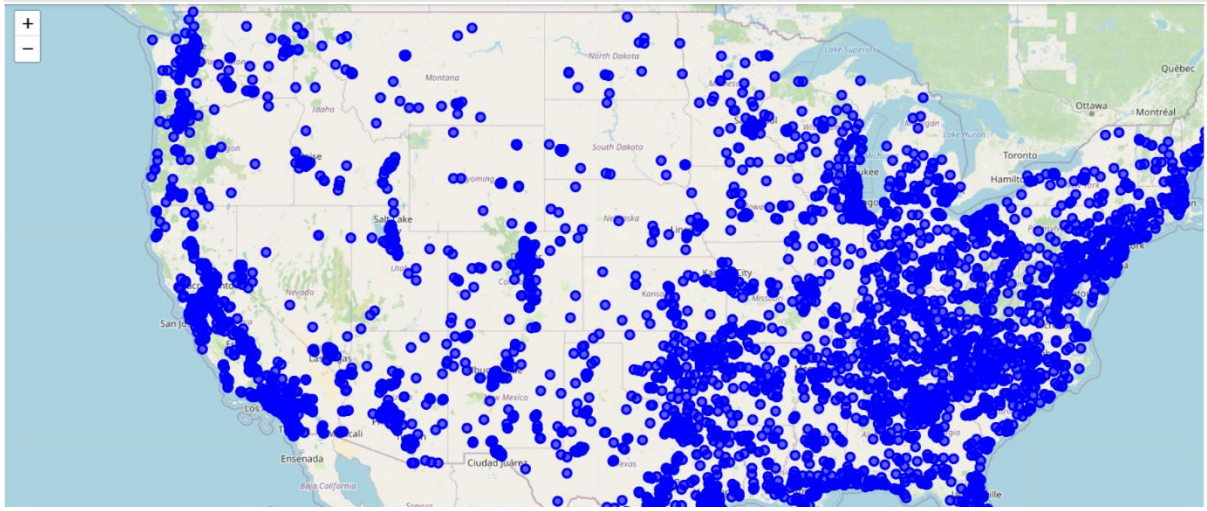
Armed Attack: The individual was armed and actively engaged in an attack, totaling 3392 incidents.

Other: The individual was armed with a different context or intention, constituting 1666 incidents.

Undetermined: The armed status could not be conclusively determined in 121 incidents.

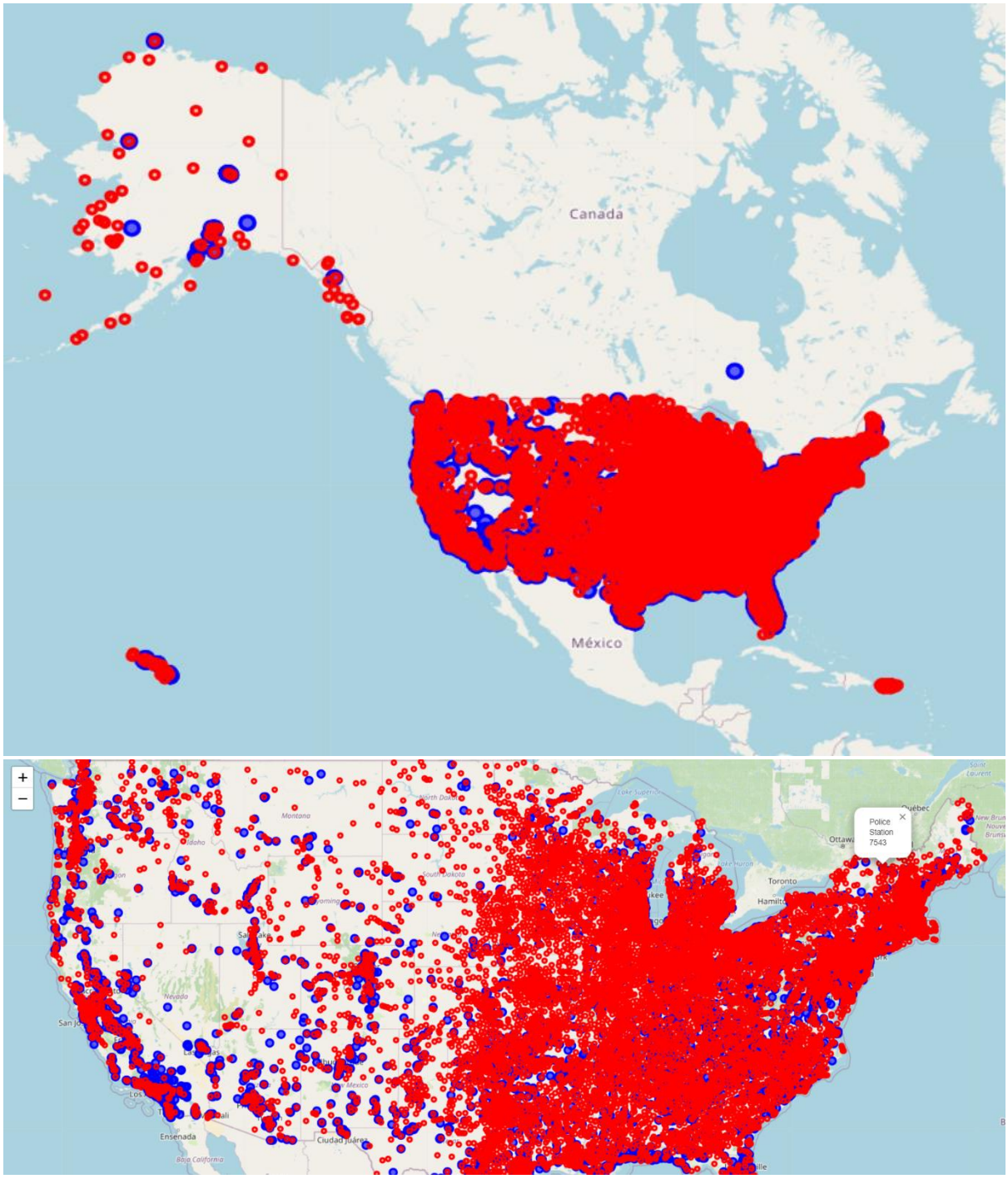
5. Mapping of incident locations





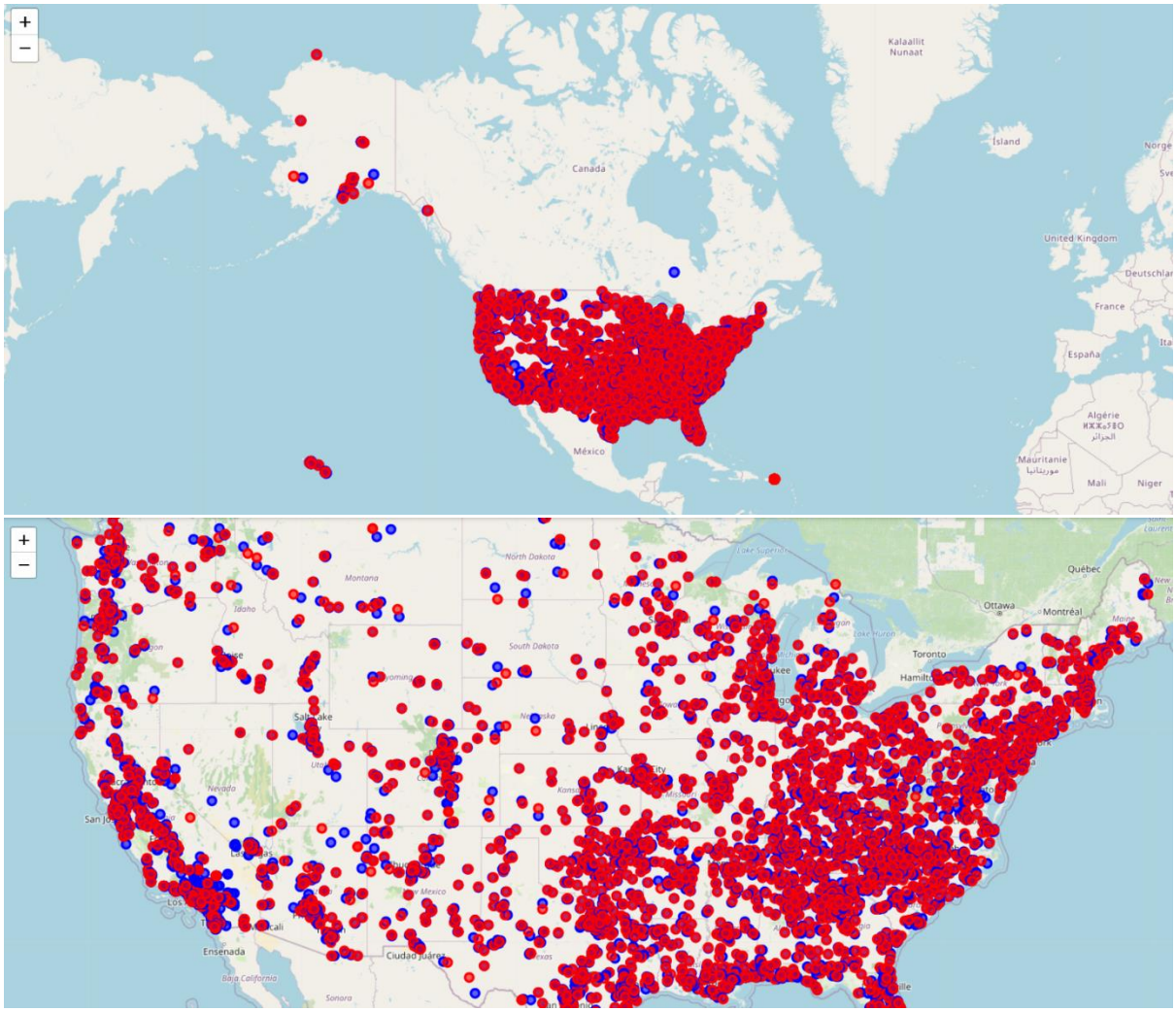
Here we have plotted the places on USA map where the police shootings happen using the longitude and latitude values from fatal police shooting data set. Here the Blue colored circular marks represent the place the shooting took place. We can observe that the large number of shootings take place in east coast and south-east coast compared to west coast. Central states have less shootings compared to coasts.

6. Locations of incidents and police stations



Here we have used both fatal police shooting location values and police station location values from two different datasets. Using those values and plotting it on a map is very helpful to understand the distance between the shooting spot and the nearest police station. The police stations are indicated by red colour and shooting spot is indicated by blue colour. Using zoom option, we can also observe more closely. And when you click on any red point it will show the police station count number and when you click on blue it shows the county or area name in tool tip. By this visualization it's clear that most of the police shootings took place near by police stations.

7. Proximity to a police station from an incident place



Here we're using the K-Nearest Neighbors (KNN) algorithm to find the nearest police stations to a set of incidents based on their latitude and longitude coordinates. We begin by loading two important datasets: one that contains the locations of police stations and another that holds information about various incidents. To find the nearest police stations, we specify the number of nearest neighbors(k) to find, which in this case is set to 1.

We also make sure that both datasets contain 'latitude' and 'longitude' columns. In case there are any missing values, we fill them with zeros to ensure consistency in the data. The Nearest Neighbors model is then initialized using the coordinates of police stations, and for each incident, it calculates and identifies the nearest police station.

The code leverages the Folium library to create an interactive map that displays the results. On this map, you'll see blue circle markers representing the incident locations and red circle markers representing their nearest police station(s). Each marker comes with a pop-up that provides additional information about the specific incident or police station. This visual representation offers a clear overview of the incidents and their proximity to nearby police stations, making it easier to understand and analyze the dataset.

To find the nearest police stations for each incident we performed,

distances, indices = nn.kneighbors(incident_coords)

where,

nn.kneighbors(incident_coords): This function is part of the k-NN algorithm, which is used for finding the k-nearest neighbors of a point in a dataset. In this case, incident_coords represents the coordinates (latitude and longitude) of the incidents. The function returns two arrays:

distances: An array containing the distances between each incident and its k-nearest neighbors i.e the nearest police stations.

indices: An array containing the indices of the k-nearest neighbors for each incident.

We also calculated the average distance to a police station from an incident place,

avg_distance_after_knn = distances.mean()

Average distance to nearest police station after KNN: 7.1 miles

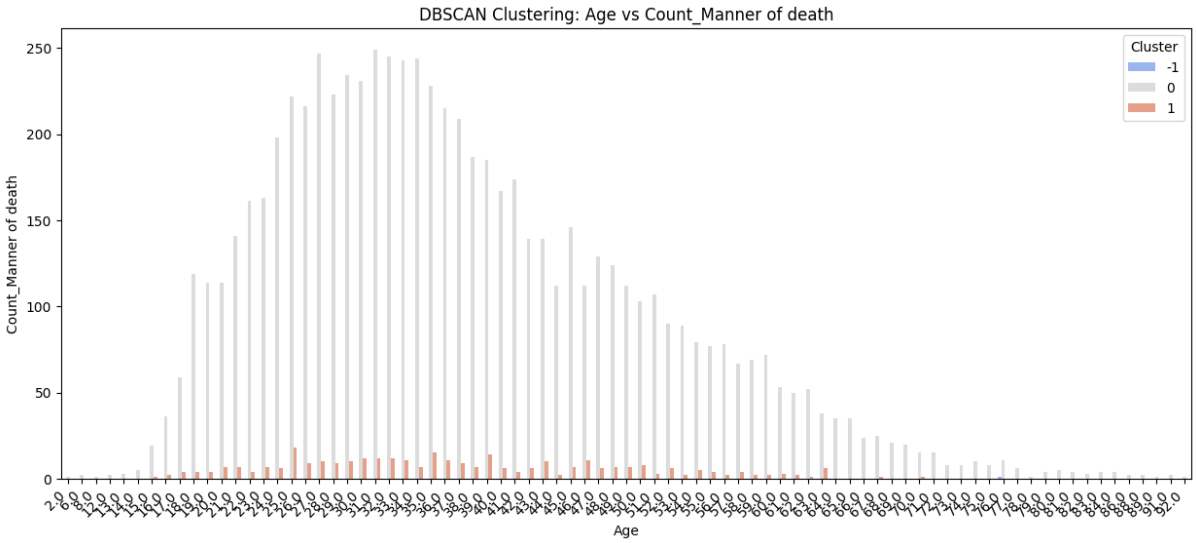
We also calculated the distance in meters for all incidents to the nearest police station and stored it as list and then created a dataframe for it,

	Incident_City	Station_Number	Average_Distance
0	Shelton	12729	240.967711
1	Aloha	12248	238.160502
2	Wichita	11292	190.868132
3	San Francisco	10531	226.535729
4	Evans	2238	205.167669
...	...	...	...
7157	Mesa	2672	205.390764
7158	Mariposa County	14863	222.645977
7159	Tulare	963	219.991674
7160	Topeka	11681	190.534660
7161	Daytona Beach	15043	155.914797

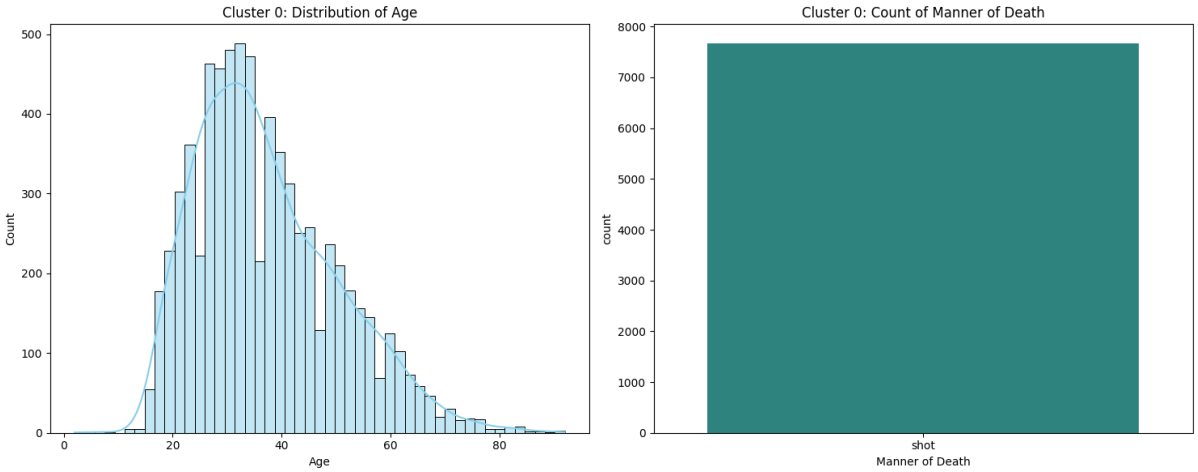
[7162 rows x 3 columns]

By calculating the average distance between the incident place and the nearest police station while displaying the corresponding station number serves a crucial role in public safety and law enforcement. This information offers a comprehensive view of the geographical accessibility of police resources. For residents, it helps assess the safety of their neighbourhoods and emergency response times, contributing to overall peace of mind. Law enforcement agencies can use this data for community policing, optimizing resource allocation, and crime analysis, which ultimately leads to more effective public safety strategies. Furthermore, it promotes transparency and trust between the police and the community, while policymakers can leverage these insights to develop and refine policies that enhance the safety and well-being of citizens. In essence, this data-driven approach empowers communities and authorities alike to create safer environments and improve emergency response capabilities.

8. DBSCAN Clustering for age and manner of death.

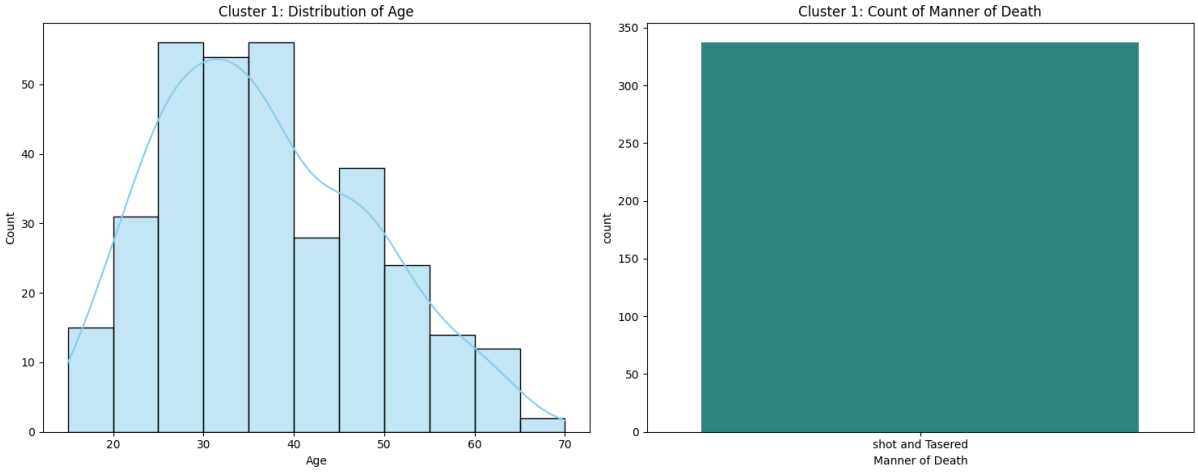


In the chart, the x-axis represents age, while the y-axis represents the count of the manner of death. Each bar is color-coded to represent different clusters identified by the DBSCAN algorithm, with -1 usually indicating noise or outliers that do not fit well into any cluster, and other numbers (0, 1, etc.) indicating different clusters.



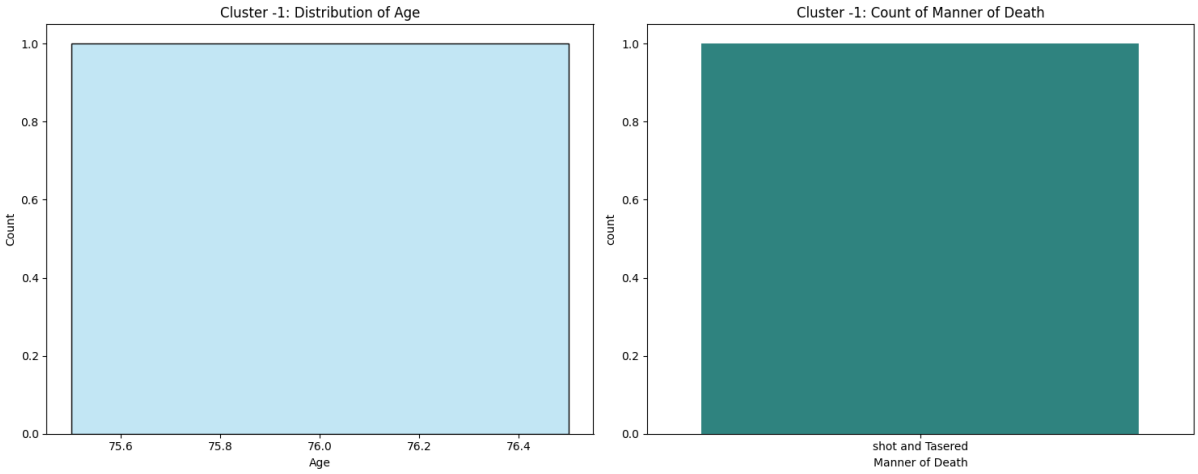
This histogram shows the age distribution for cluster "0". The distribution appears to be roughly normally distributed with a peak around the 20-30 age range.

The count plot for manner of death within this cluster indicates that all individuals were categorized under one manner of death: "shot". The uniformity of the manner of death in this cluster could reflect that the clustering process captured a group where this was the common characteristic.



The left histogram represents the distribution of age within a cluster labeled "1". It appears to be a normally distributed age range with a possible skew towards younger ages. Many individuals in this cluster fall in the 20-40 age range, with fewer occurrences as age increases.

The right histogram displays the count of manner of death within the same cluster. It indicates that all individuals in this cluster have the same manner of death, labeled as "shot and Tasered". This could suggest that the cluster has a specific pattern or characteristic in common, which in this case is the manner of death.



The cluster -1 contains noisy data points which are usually outliers. These points don’t have proper scale for age and count, which are not easy for understanding. According to the output we have range for age between 75.6 to 76.4 and for which manner of death is shot and tasered.

6. Appendix C: Code

```
import pandas as pd
from sklearn.neighbors import NearestNeighbors
import folium
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.impute import SimpleImputer
import seaborn as sns

# Load the police station dataset
police_stations_data = pd.read_excel("/content/Latitude-longitude of US police stations.xlsx")

# Create a map centered on the USA
usa_map = folium.Map(location=[37.0902, -95.7129], zoom_start=4)

# Filter out rows with missing latitude or longitude values in the fatal shootings dataset
data_path = data_path.dropna(subset=['latitude', 'longitude'])

# Add circle markers for each data point in the fatal shootings dataset
for index, row in data_path.iterrows():
    folium.CircleMarker(
        location=[row['latitude'], row['longitude']],
        radius=5,  # Adjust the radius
        color='blue',  # Adjust the color
        fill=True,
        fill_color='blue',  # Adjust the fill color
        fill_opacity=0.6,
        popup=row['city']  # Customize the popup information
```

```
    ).add_to(usa_map)

# Filter out rows with missing latitude or longitude values
data = data.dropna(subset=['latitude', 'longitude'])

# Add markers for each police station
for index, row in police_stations_data.iterrows():
    folium.CircleMarker(
        location=[row['Latitude'], row['longitude']],
        radius=3,  # Adjust the radius
        color='red',  # Adjust the color
        fill=True,
        fill_color='red',  # Adjust the fill color
        fill_opacity=0.2,
        popup=f"Police Station {index + 1}"
    ).add_to(usa_map)

# Display the map with circle markers for fatal shootings and markers for police stations
usa_map

# Choose the number of nearest police stations to find
k = 1

# Ensure both dataframes have columns 'latitude' and 'longitude'
police_stations[['latitude', 'longitude']] = police_stations[['Latitude', 'longitude']].fillna(0)
incidents[['latitude', 'longitude']] = incidents[['latitude', 'longitude']].fillna(0)

# Extract latitude and longitude coordinates
police_coords = police_stations[['latitude', 'longitude']].values
incident_coords = incidents[['latitude', 'longitude']].values
```

```
# Initialize the Nearest Neighbors model
nn = NearestNeighbors(n_neighbors=k, algorithm='ball_tree')
nn.fit(police_coords)

# Find the nearest police stations for each incident
distances, indices = nn.kneighbors(incident_coords)

# Create a map centered on the USA
usa_map = folium.Map(location=[37.0902, -95.7129], zoom_start=4)

# Add circle markers for each incident and its nearest police stations
for i, row in incidents.iterrows():
    folium.CircleMarker(
        location=[row['latitude'], row['longitude']],
        radius=5,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.6,
        popup=row['city']
    ).add_to(usa_map)

    for idx in indices[i]:
        station = police_stations.iloc[idx]
        folium.CircleMarker(
            location=[station['latitude'], station['longitude']],
            radius=5,
            color='red',
            fill=True,
```

```
            fill_color='red',
            fill_opacity=0.6,
            popup=f"Police Station {idx + 1}"
        ).add_to(usa_map)

# Display the interactive map
#usa_map.save('usa_map.html')
usa_map
# Calculate the average distance to the nearest police station after KNN
avg_distance_after_knn = distances.mean()

# Print the results
print("Average distance to nearest police station after KNN:", avg_distance_after_knn)
import pandas as pd
import folium
from sklearn.neighbors import NearestNeighbors
import numpy as np

# Create lists to store the results
results = []

# Iterate through incidents and their nearest police stations
for i, row in fatal_shootings_data.iterrows():
    incident_location = (row['latitude'], row['longitude'])
    station_location = (
        police_stations_data.iloc[indices[i]]['Latitude'],
        police_stations_data.iloc[indices[i]]['longitude']
    )

    # Calculate the distance between the incident and the nearest police station
```

```
    distance = np.linalg.norm(np.array(incident_location) - np.array(station_location))

    # Add the results to the list
    results.append({
        'Incident_City': row['city'],
        'Station_Number': indices[i][0] + 1,
        'Average_Distance': distance
    })

# Create a DataFrame from the results
results_df = pd.DataFrame(results)

# Print the DataFrame
print(results_df)

# Load your dataset (replace 'data.csv' with your dataset file)
data = pd.read_excel("/content/fatal-police-shootings-data.xls")

# Choose two variables for clustering (age and shooting)
features = ['age', 'manner_of_death']

# Extract the selected features
X = data[features]

# Convert 'shooting' column to numeric using Label Encoding
label_encoder = LabelEncoder()
X['manner_of_death'] = label_encoder.fit_transform(X['manner_of_death'])

# Handle NaN values
imputer = SimpleImputer(strategy='mean')  # You can choose a different strategy if needed
```

```
X = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply DBSCAN clustering
dbscan = DBSCAN(eps=0.5, min_samples=5)
clusters = dbscan.fit_predict(X_scaled)

# Visualize the clusters
plt.figure(figsize=(15, 6))
sns.countplot(data=data, x='age', hue=clusters, palette='coolwarm')
plt.title('DBSCAN Clustering: Age vs Count_Manner of death')
plt.xlabel('Age')
plt.ylabel('Count_Manner of death')
plt.xticks(rotation=45, ha='right')  # Adjust x-axis labels for better readability
plt.legend(title='Cluster', loc='upper right')
plt.show()
```

Colab link of code:
https://colab.research.google.com/drive/1TWRb2mo7nJrFFj3jh70pzIzc7ZpiM7y?usp=sharing

7. References

Textbook Reference: “An Introduction to Statistical Learning with Applications in Python”.

8. Author Contributions

In this collaborative project, both authors made equal contributions.