Beyond the Headlines:
Deep Dive into Police Shootings Data and Patterns

Addressed to:

Jennifer Jenkins
Steven Rich
Andrew Ba Tran
Paige Moody
Julie Tate
Ted Mellnik

Co-Authors:

Aman Thakur
Ayush Tiwari
Sneh Pillai
Tabeesh Fatiya

Issues

A startling fact has been revealed by The Washington Post's database on police shootings in the United States since 2015: almost 1,000 fatalities annually, a glaring undercount based on ongoing investigation. The Post revealed to the FBI that there had been serious underreporting following Michael Brown's 2014 death. Just one-third of occurrences were included in the FBI database by 2021. The Post improved accountability in 2022. We address the questions:

- Do certain police departments show a higher rate of incidents of no body camera usage?
- Is there a difference in the mean age between different racial groups?
- What proportion of individuals involved in incidents were armed? Are there age-related patterns in the type of weapons used in fatal incidents?
- Are there any trends in the types of arms used by different races?
- Are there statistically significant differences in the prevalence of mental illness indicators across different racial?
- Are there discernible spatial characteristics associated with police shootings?

Findings

- According to our analysis, Texas (207), California (161), Georgia (129), North Carolina (103), and Tennessee (106), are the top 5 states with the greatest percentage of police departments that do not use body cameras during fatal incidents. This raises concerns about accountability and transparency as it appears that these departments have consistently not used body cameras as documented. Additional research into these states' body camera policies and practises may shed light on law enforcement protocols and point out areas where incident reporting and scrutiny might be strengthened.
- The mean age differences highlight significant disparities between Black and White populations, with a substantial 7.20 years
- We found clear evidence that the average ages vary significantly across different racial groups. Our analysis used statistical methods to compare these groups, and the results strongly suggest that the mean ages are not the same. In simpler terms, it's likely that there are real differences in the average ages between these racial categories
- A significant majority of individuals involved in incidents were definitively armed, comprising 87.75% of the total. Further exploration revealed that among those who were armed, the majority (57.96%) carried guns.
- The analysis reveals varying percentages of mental illness indicators across racial groups. Asians exhibit a relatively high percentage at 22.6%, suggesting a notable prevalence of mental health concerns in police incidents. Black individuals show a lower but still significant rate of 15%, while Whites demonstrate the highest at 26.6%.
- Discernible demographic patterns exist in the spatial distribution of police shootings. The data reveals notable clusters in certain regions, suggesting a correlation with specific demographics. Further exploration is needed to understand the nuances of these patterns and their implications

Discussion

- Normality was not detected in any racial group. Levene's test indicates significant differences in variances across groups, violating ANOVA assumptions. Traditional ANOVA and Welch's ANOVA both confirm significant age differences between races, but Welch's ANOVA, robust to unequal variances, was preferred due to homogeneity violations. The low p-values in both ANOVA tests provide strong evidence against the null hypothesis, suggesting substantial age variations among racial groups. The reliance on Welch's ANOVA underscores its suitability for handling heterogeneous variances. In conclusion, the results indicate significant age disparities across races, emphasizing the importance of using appropriate statistical methods for robust conclusions.

- Our determination is grounded in rigorous analysis and statistical evidence from the dataset. The evident 7+ year average age difference between Black and White individuals in police shooting incidents prompts thoughtful consideration. While speculation on potential reasons arises, it's crucial to approach causation with caution. The observed age difference is a product of the available data, emphasizing the need for additional information and verifiable explanations. Future datasets may present distinct average age disparities, urging a nuanced exploration of factors contributing to differential outcomes between Black and White populations in police interactions.

- A small proportion, 3.98%, fell into the ”undetermined” category, suggesting uncertainty about their armed status. A minor percentage of incidents, 2.39%, had missing data, indicating a lack of information regarding whether the individuals were armed.

- The finding regarding mental illness underscores the importance of tailored approaches in law enforcement, mental health support, and community outreach. Addressing the nuanced needs of each racial group is crucial for fostering equitable and effective responses, contributing to a more inclusive and empathetic approach to policing and mental health services in different communities.

- The analysis uncovers a highly significant spatial pattern in police shootings (Moran's I = 0.9725, p = 0.0010). Concentrated clusters, notably in the central-right region (Cluster 1), suggest non-random occurrences. Demographic exploration within these clusters unveils complex factors influencing the spatial distribution, warranting further investigation for targeted intervention.

Appendix A: Method

Data Collection: Every person shot and killed by an on-duty police officer in the US is tracked by The Washington Post since 2015. Reporters have documented thousands of deaths since then. In order to improve departmental accountability, The Post updated its database in 2022 to standardise and make public the names of the police agencies engaged in each shooting. **Data Loading and Preprocessing**: Multiple data preprocessing steps are performed, including splitting multiple police departments into separate rows, extracting the state from the police department column, and filtering incidents with no body camera usage. **Data Analysis**: The script then analyzes the data to answer specific questions:

- Investigating if certain police departments show a higher rate of incidents with no body camera usage.
- Examining differences in the mean age between different racial groups using ANOVA and Welch's ANOVA.
- Exploring trends in the types of arms used by different races.
- Investigating the prevalence of mental illness indicators across different racial groups using chi-squared test.
- Performing clustering analysis using DBSCAN to identify spatial patterns in police shootings.

Data Visualization: Folium is employed to create interactive maps, including a heatmap of police shooting incidents and clusters identified by DBSCAN. **Machine Learning**: Logistic Regression is applied to predict the type of arms used based on age. Standardization and encoding of categorical variables are performed using LabelEncoder and StandardScaler. **GeoSpatial Analysis**: GeoPandas is used to work with GeoJSON files and perform geospatial analysis. Folium is utilized to create interactive maps, providing a visual representation of police shooting incidents across different locations **Statistical Tests**:

- Shapiro-Wilk test is used for checking normality of age distribution within each racial group. Levene's test is employed to check the homogeneity of variances across different racial groups.
- ANOVA and Welch's ANOVA are performed to assess significant differences in mean ages across racial groups.
- Chi-squared test is utilized to examine the association between race and signs of mental illness.

Clustering Analysis: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is applied for spatial clustering of police shootings, identifying patterns in the geographic distribution of incidents. **Output and Visualization**: The script outputs various statistics, classification reports, and visualizations, including bar plots, heatmaps, and maps.

Appendix B: Results

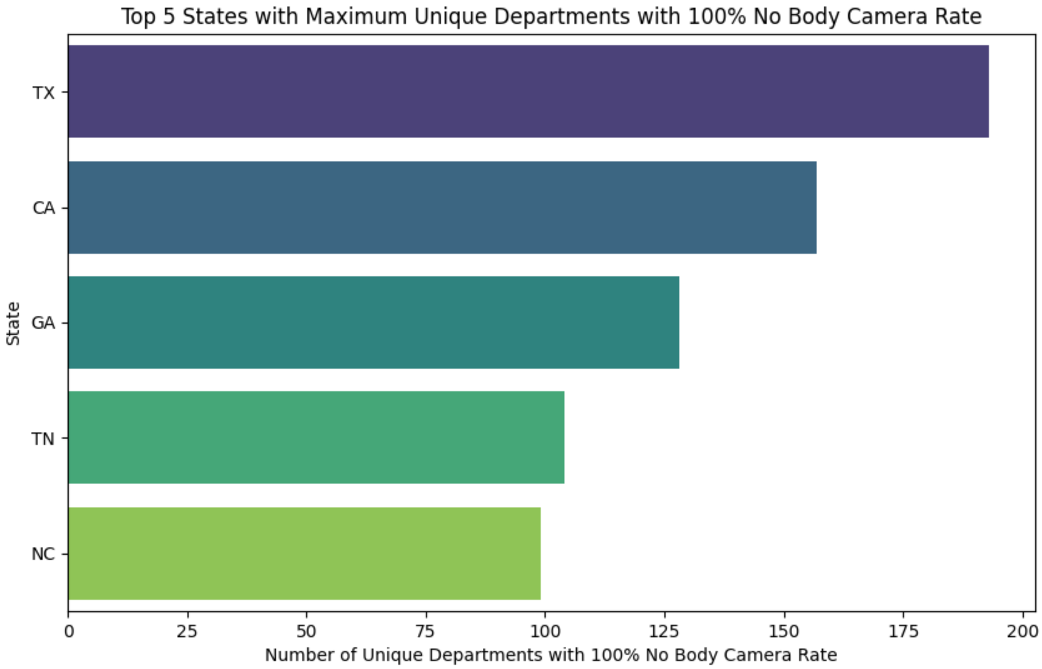


Figure 1: States with 100% no body camera rate

The Shapiro-Wilk tests for normality reveal that the age distribution is not uniform across racial groups, as evidenced by low p-values (e.g., p = 1.892e-26 for White). Additionally, Levene's Test for Homogeneity of Variances supports the assertion that there are significant differences in age variances across these groups (p = 8.917e-54). The Welch's ANOVA F-Statistic of 130.99 further reinforces the conclusion that there are substantial differences in ages across races (p = 2.40e-160).

The ANOVA results also extend to weapon types, indicating statistically significant differences in age across different categories (p = 3.98e-18). The classification model, with an accuracy of 58%, provides a nuanced understanding of its performance through precision, recall, and F1-score metrics for various weapon types.

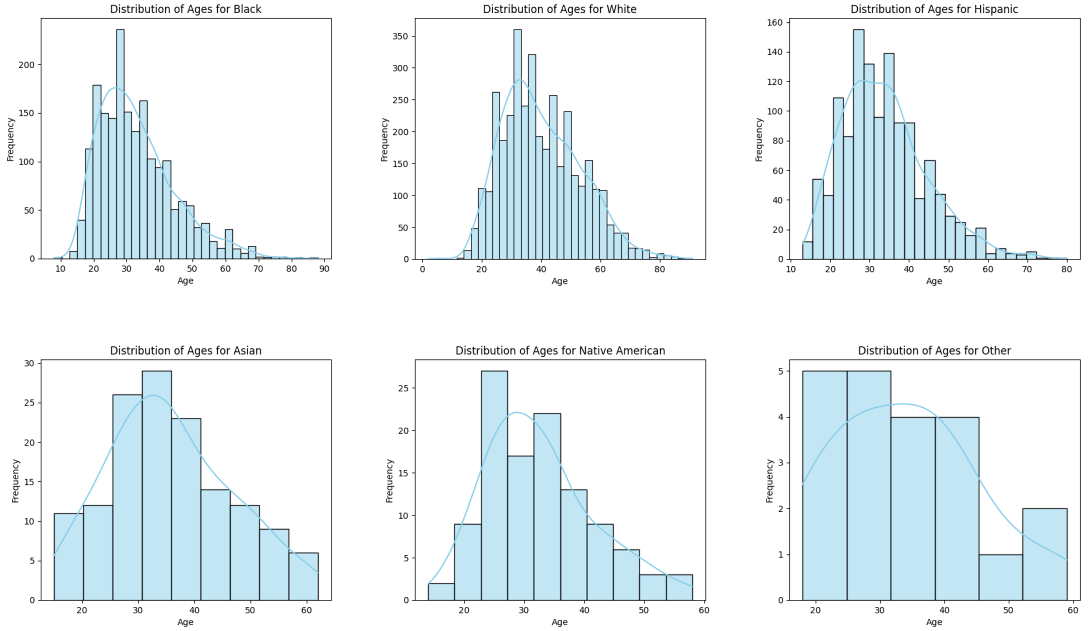


Figure 2: Distribution of Ages for Different Races

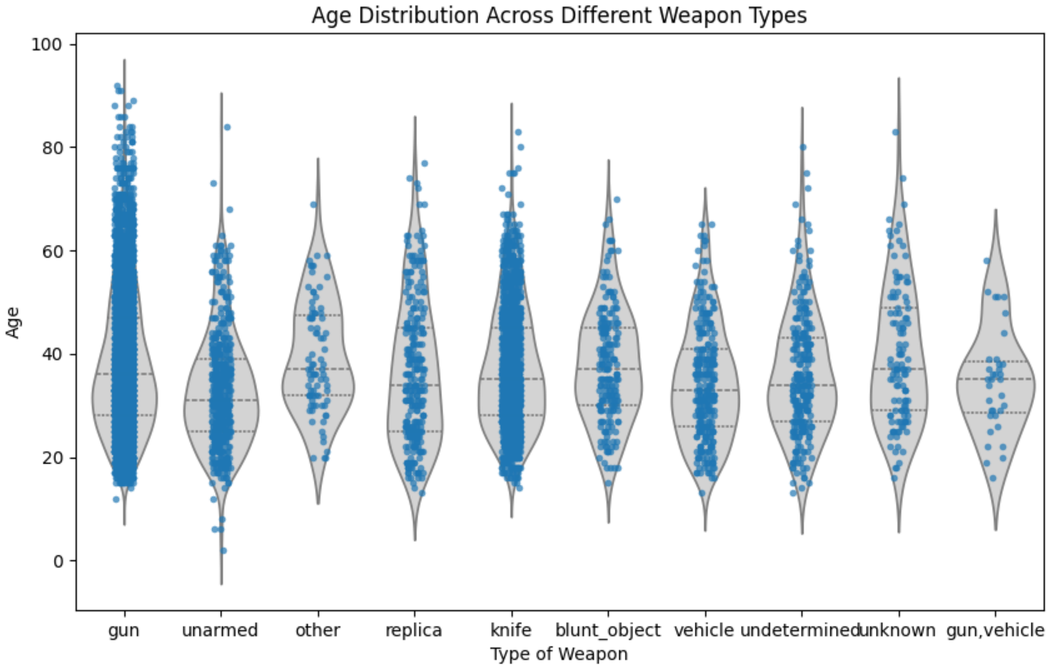


Figure 3: Violin Plot of Age Distribution Across Different Weapon Types

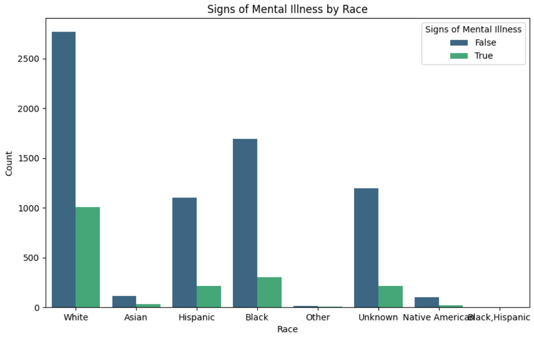
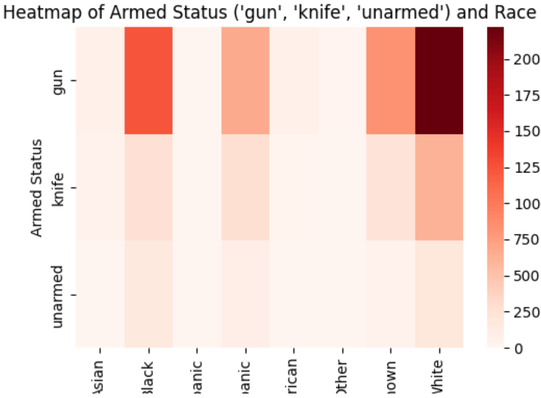


Figure 4: Signs of Mental Illness by Race

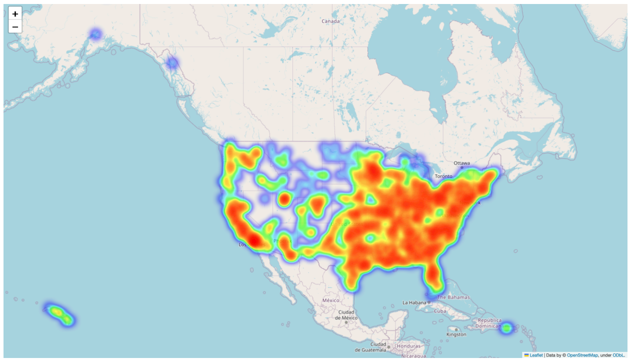


The prevalence of mental illness indicators exhibits noteworthy disparities among racial groups, as indicated by a p-value of 1.26e-32. The percentage distribution of signs of mental illness ranges from 0% for Black, Hispanic individuals to 28.57% for those categorized as 'Other.'

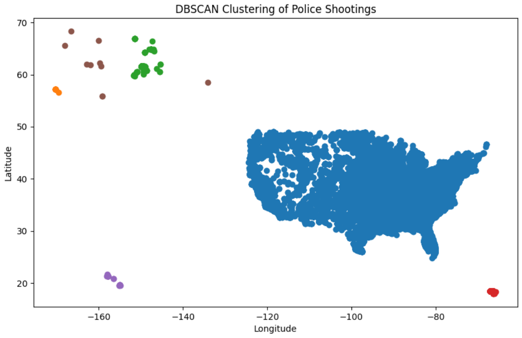
```
Asian              22.602740
Black              15.045135
Black,Hispanic      0.000000
Hispanic           16.045627
Native American    15.384615
Other              28.571429
Unknown            15.286624
White              26.617179
```

Geographical data, featuring city names, latitude, longitude, and cluster assignments, reveals the existence of five distinct clusters. This spatial categorization provides insights into potential regional patterns or disparities within the dataset.

These numerical values serve as critical benchmarks, substantiating the statistical significance of the observed patterns. The low p-values in normality tests and ANOVA underscore the robustness of the findings, while the accuracy of the classification model offers insights into its reliability. The geographic clustering further adds a spatial dimension to the analysis, potentially revealing patterns that may have implications for law enforcement practices or resource allocation.

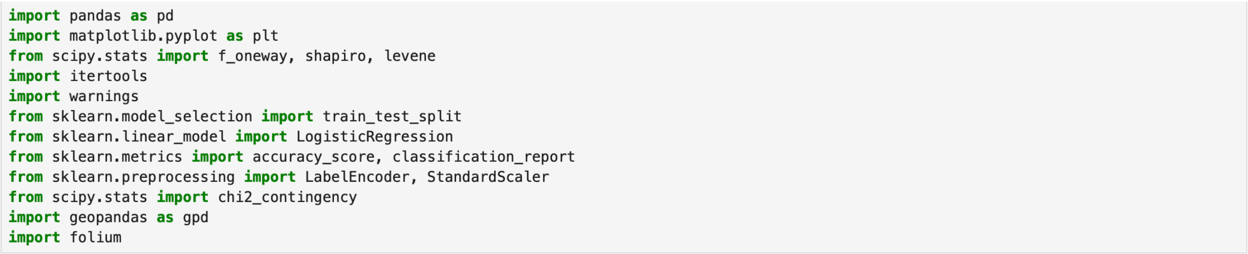


(a) HeatMap layer for the police shooting incidents

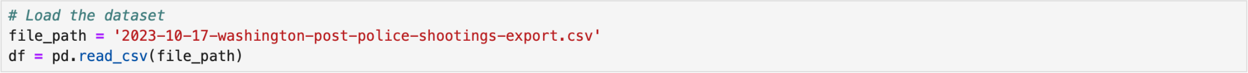


(b) Clusters

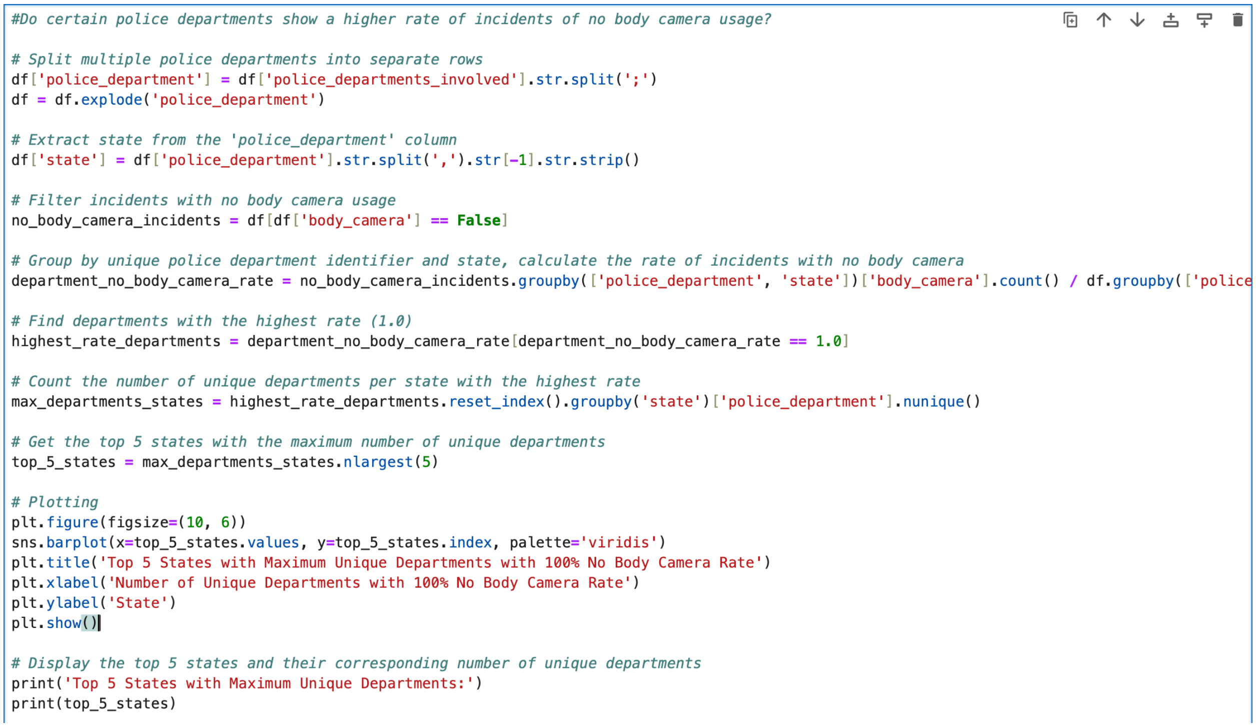
Appendix C: Code



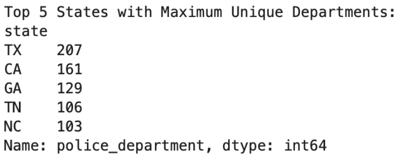
```python
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import f_oneway, shapiro, levene
import itertools
import warnings
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder, StandardScaler
from scipy.stats import chi2_contingency
import geopandas as gpd
import folium
```



```python
# Load the dataset
file_path = '2023-10-17-washington-post-police-shootings-export.csv'
df = pd.read_csv(file_path)
```



```python
#Do certain police departments show a higher rate of incidents of no body camera usage?

# Split multiple police departments into separate rows
df['police_department'] = df['police_departments_involved'].str.split(';')
df = df.explode('police_department')

# Extract state from the 'police_department' column
df['state'] = df['police_department'].str.split(',').str[-1].str.strip()

# Filter incidents with no body camera usage
no_body_camera_incidents = df[df['body_camera'] == False]

# Group by unique police department identifier and state, calculate the rate of incidents with no body camera
department_no_body_camera_rate = no_body_camera_incidents.groupby(['police_department', 'state'])['body_camera'].count() / df.groupby(['police

# Find departments with the highest rate (1.0)
highest_rate_departments = department_no_body_camera_rate[department_no_body_camera_rate == 1.0]

# Count the number of unique departments per state with the highest rate
max_departments_states = highest_rate_departments.reset_index().groupby('state')['police_department'].nunique()

# Get the top 5 states with the maximum number of unique departments
top_5_states = max_departments_states.nlargest(5)

# Plotting
plt.figure(figsize=(10, 6))
sns.barplot(x=top_5_states.values, y=top_5_states.index, palette='viridis')
plt.title('Top 5 States with Maximum Unique Departments with 100% No Body Camera Rate')
plt.xlabel('Number of Unique Departments with 100% No Body Camera Rate')
plt.ylabel('State')
plt.show()

# Display the top 5 states and their corresponding number of unique departments
print('Top 5 States with Maximum Unique Departments:')
print(top_5_states)
```



```
Top 5 States with Maximum Unique Departments:
state
TX    207
CA    161
GA    129
TN    106
NC    103
Name: police_department, dtype: int64
```

```python
#Is there a difference in the mean age between different racial groups?

# Drop rows where 'age' is null
df = df.dropna(subset=['age'])

# Function to check normality and plot distribution
def check_normality_and_plot(data, race):
    stat, p_value = shapiro(data)
    print(f"Shapiro-Wilk Test for {race}: p-value = {p_value}")
    if p_value < 0.05:
        print(f"Normality not detected for {race}")
    else:
        print(f"Normality detected for {race}")
    plt.figure(figsize=(4, 3))
    sns.histplot(data, kde=True, color='skyblue')
    plt.title(f'Distribution of Ages for {race}')
    plt.xlabel('Age')
    plt.ylabel('Frequency')
    # plt.show()

# Check normality for each race
for race in df['race'].unique():
    race_data = df[df['race'] == race]['age']
    if len(race_data) >= 3:
        check_normality_and_plot(race_data, race)

# Check homogeneity of variances
valid_race_data = [df[df['race'] == race]['age'] for race in df['race'].unique() if len(df[df['race'] == race]) >= 3]
stat, p_value_levene = levene(*valid_race_data)
print(f"Levene's Test for Homogeneity of Variances: p-value = {p_value_levene}")
alpha_homogeneity = 0.05
if p_value_levene < alpha_homogeneity:
    print("Reject the null hypothesis. There is evidence of a significant difference in variances across groups.\n")
else:
    print("Fail to reject the null hypothesis. There is no significant difference in variances across groups.\n")
```

```
Shapiro-Wilk Test for White: p-value = 1.8929681926494327e-26
Normality not detected for White
Shapiro-Wilk Test for Asian: p-value = 0.009995344409493208
Normality not detected for Asian
Shapiro-Wilk Test for Hispanic: p-value = 2.257409919961122e-22
Normality not detected for Hispanic
Shapiro-Wilk Test for Black: p-value = 8.179339732895038e-28
Normality not detected for Black
Shapiro-Wilk Test for Other: p-value = 0.5033304691314697
Normality detected for Other
Shapiro-Wilk Test for Unknown: p-value = 1.2972030331024812e-16
Normality not detected for Unknown
Shapiro-Wilk Test for Native American: p-value = 0.0012239518109709024
Normality not detected for Native American
Levene's Test for Homogeneity of Variances: p-value = 8.917476503322734e-54
Reject the null hypothesis. There is evidence of a significant difference in variances across groups.
```

```python
# Perform ANOVA
f_statistic, p_value = f_oneway(*valid_race_data)
print("F-statistic:", f_statistic)
print("P-value:", p_value)

if p_value < 0.05:
    print("Reject the null hypothesis. There are significant differences in ages across different races.\n")
else:
    print("Fail to reject the null hypothesis. No significant differences in ages across different races.\n")

# Perform Welch's ANOVA
f_statistic, p_value_welch = f_oneway(*valid_race_data)
print(f"Welch's ANOVA F-Statistic = {f_statistic}, p-value = {p_value_welch}")

# Interpretation
alpha_welch = 0.05
if p_value_welch < alpha_welch:
    print("Reject the null hypothesis. There is evidence of a significant difference in ages across races (Welch's ANOVA).\n")
else:
    print("Fail to reject the null hypothesis. There is no significant difference in ages across races (Welch's ANOVA).\n")

# Get unique races
unique_races = df['race'].unique()

# Calculate mean age for each race
mean_ages = {race: df[df['race'] == race]['age'].mean() for race in unique_races}

# Find and print mean age differences between each pair of races
for race1, race2 in itertools.combinations(unique_races, 2):
    mean_diff = mean_ages[race1] - mean_ages[race2]
    print(f"Mean age difference between {race1} and {race2}: {mean_diff:.2f} years")
```

```
Mean age difference between White and Asian: 4.43 years
Mean age difference between White and Hispanic: 6.45 years
Mean age difference between White and Black: 7.20 years
Mean age difference between White and Native American: 7.61 years
Mean age difference between White and Black,Hispanic: 13.21 years
Mean age difference between Asian and Hispanic: 2.01 years
Mean age difference between Asian and Black: 2.77 years
Mean age difference between Asian and Native American: 3.18 years
Mean age difference between Asian and Black,Hispanic: 8.77 years
Mean age difference between Hispanic and Black: 0.76 years
Mean age difference between Hispanic and Native American: 1.17 years
Mean age difference between Hispanic and Black,Hispanic: 6.76 years
Mean age difference between Black and Native American: 0.41 years
Mean age difference between Black and Black,Hispanic: 6.00 years
Mean age difference between Native American and Black,Hispanic: 5.59 years
```

```
# Filter relevant columns
df_age_weapon = df[['age', 'armed']]
# Drop rows with missing values
df_age_weapon = df_age_weapon.dropna()
# Get the top N weapons by count
top_weapons = df_age_weapon['armed'].value_counts().nlargest(10).index

# Filter the dataframe for only the top weapons
df_top_weapons = df_age_weapon[df['armed'].isin(top_weapons)]

# Perform ANOVA
anova_result = f_oneway(*[df_top_weapons[df_top_weapons['armed'] == weapon]['age'] for weapon in top_weapons])

# Check p-value
p_value = anova_result.pvalue

# Print the result
print(f'ANOVA p-value: {p_value}')

# Interpret the result
if p_value < 0.05:
    print('There is a statistically significant difference in age across different weapon types.')
else:
    print('There is no statistically significant difference in age across different weapon types.')

#Log regression to predict

# Filter relevant columns
df_age_weapon = df[['age', 'armed']].dropna()

# Encode the 'armed' column to numerical values
le = LabelEncoder()
df_age_weapon['armed_encoded'] = le.fit_transform(df_age_weapon['armed'])

# Features (X) and target variable (y)
X = df_age_weapon[['age']]
y = df_age_weapon['armed_encoded']
```

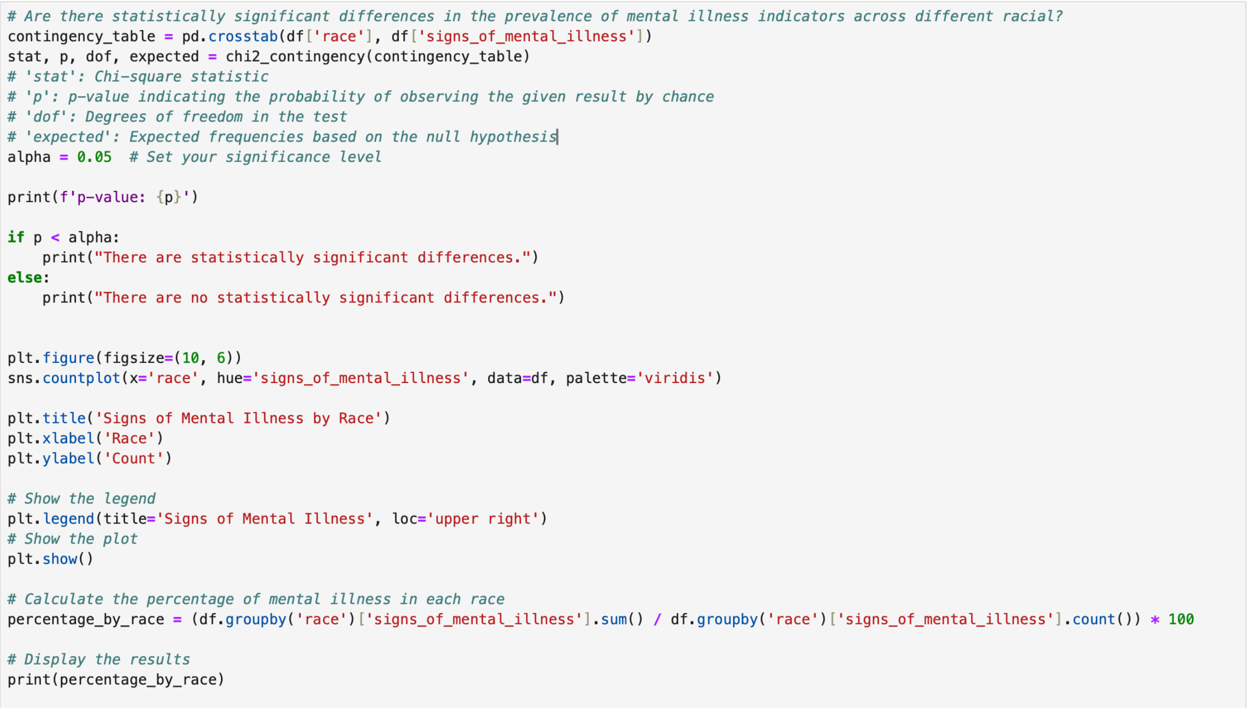
```
# Standardize the features (scaling)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Create and train a logistic regression model
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Ensure the number of classes in the classification report matches the number of unique classes in the original 'armed' column
unique_classes = df_age_weapon['armed'].nunique()
report = classification_report(y_test, y_pred, labels=range(unique_classes), target_names=le.classes_[:unique_classes], zero_division=1)

print(f'Accuracy: {accuracy_score(y_test, y_pred):.2f}')
print('Classification Report:')
print(report)
```

```
ANOVA p-value: 3.97628510118805e-18
There is a statistically significant difference in age across different weapon types.
Accuracy: 0.58
Classification Report:
                           precision    recall  f1-score   support

             blunt_object       1.00      0.00      0.00        42
blunt_object,blunt_object       1.00      1.00      1.00         0
       blunt_object,knife       1.00      1.00      1.00         0
                      gun       0.58      1.00      0.73       917
                gun,knife       1.00      0.00      0.00         5
              gun,vehicle       1.00      0.00      0.00        10
                    knife       1.00      0.00      0.00       290
       knife,blunt_object       1.00      0.00      0.00         1
            knife,vehicle       1.00      1.00      1.00         0
                    other       1.00      0.00      0.00        20
 other,blunt_object,knife       1.00      1.00      1.00         0
                other,gun       1.00      0.00      0.00         1
                  replica       1.00      0.00      0.00        63
            replica,knife       1.00      1.00      1.00         0
          replica,vehicle       1.00      1.00      1.00         0
                  unarmed       1.00      0.00      0.00        85
             undetermined       1.00      0.00      0.00        65
                  unknown       1.00      0.00      0.00        27
                  vehicle       1.00      0.00      0.00        67
              vehicle,gun       1.00      1.00      1.00         0
      vehicle,knife,other       1.00      0.00      0.00         1

                micro avg       0.58      0.58      0.58      1594
                macro avg       0.98      0.38      0.37      1594
             weighted avg       0.76      0.58      0.42      1594
```

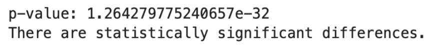
```
#Are there any trends in the types of arms used by different races?

# Calculate the count of different categories and missing values
total_count = len(df)
unarmed_count = df['armed'].eq('unarmed').sum()
undetermined_count = df['armed'].eq('undetermined').sum()
missing_count = df['armed'].isna().sum()

# Calculate the count of definitely armed individuals
definitely_armed_count = total_count - unarmed_count - undetermined_count - missing_count

# Calculate the percentages
definitely_armed_percentage = (definitely_armed_count / total_count) * 100
undetermined_percentage = (undetermined_count / total_count) * 100
missing_percentage = (missing_count / total_count) * 100
gun_count = df['armed'].eq('gun').sum()
gun_percentage = (gun_count / total_count) * 100

# Print the results
print(f'Definitely Armed: {definitely_armed_percentage:.2f}%')
print(f'Undetermined: {undetermined_percentage:.2f}%')
print(f'Missing data: {missing_percentage:.2f}%')
print(f'Gun: {gun_percentage:.2f}%')
```

```
Definitely Armed: 87.75%
Undetermined: 3.98%
Missing data: 2.39%
Gun: 57.96%
```

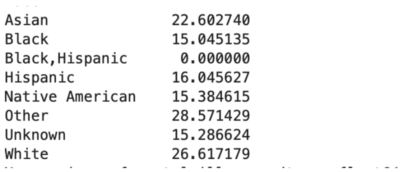
```
#Are there any trends in the types of arms used by different races?
filtered_df = df[df['armed'].isin(['gun', 'knife', 'unarmed'])]

# A cross-tabulation (contingency table) of armed and race
cross_tab = pd.crosstab(filtered_df['armed'], filtered_df['race'])

plt.figure(figsize=(6, 4))
sns.heatmap(cross_tab, cmap="Reds", annot=False, fmt='d')

plt.xlabel("Race")
plt.ylabel("Armed Status")
plt.title("Heatmap of Armed Status ('gun', 'knife', 'unarmed') and Race")
filename = f'9.png'
plt.savefig(filename)
plt.show()
plt.show()
```



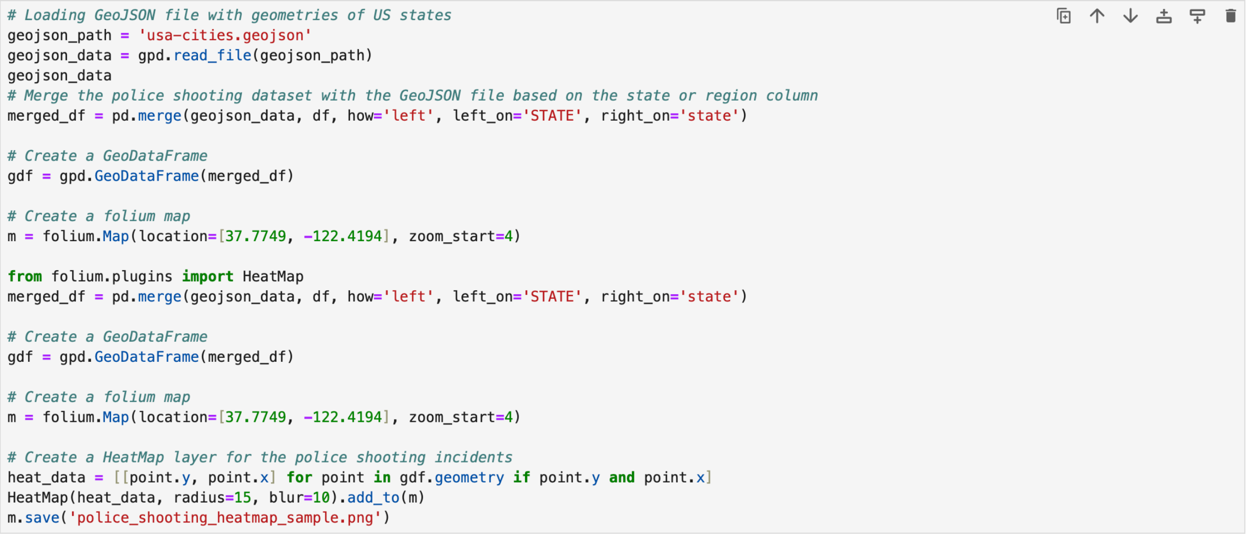
```python
# Are there statistically significant differences in the prevalence of mental illness indicators across different racial?
contingency_table = pd.crosstab(df['race'], df['signs_of_mental_illness'])
stat, p, dof, expected = chi2_contingency(contingency_table)
# 'stat': Chi-square statistic
# 'p': p-value indicating the probability of observing the given result by chance
# 'dof': Degrees of freedom in the test
# 'expected': Expected frequencies based on the null hypothesis
alpha = 0.05  # Set your significance level

print(f'p-value: {p}')

if p < alpha:
    print("There are statistically significant differences.")
else:
    print("There are no statistically significant differences.")


plt.figure(figsize=(10, 6))
sns.countplot(x='race', hue='signs_of_mental_illness', data=df, palette='viridis')

plt.title('Signs of Mental Illness by Race')
plt.xlabel('Race')
plt.ylabel('Count')

# Show the legend
plt.legend(title='Signs of Mental Illness', loc='upper right')
# Show the plot
plt.show()

# Calculate the percentage of mental illness in each race
percentage_by_race = (df.groupby('race')['signs_of_mental_illness'].sum() / df.groupby('race')['signs_of_mental_illness'].count()) * 100

# Display the results
print(percentage_by_race)
```



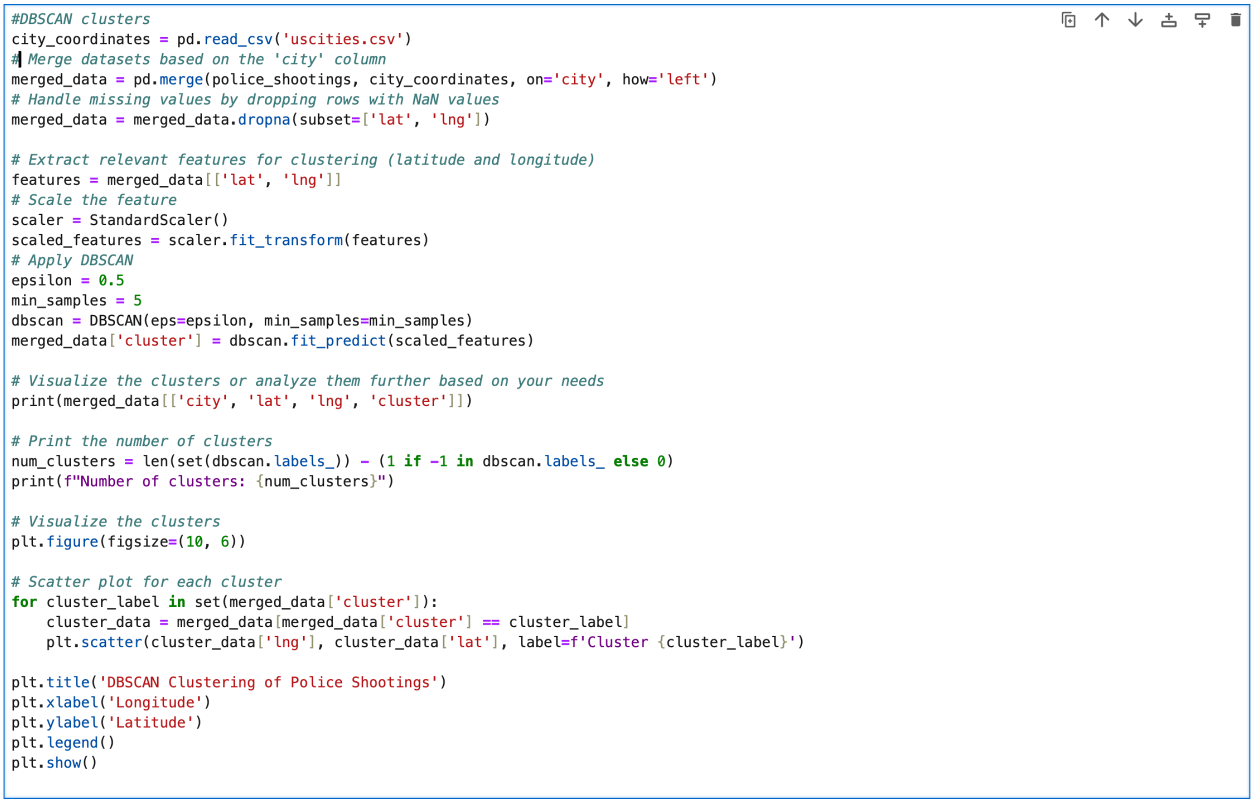
```
p-value: 1.264279775240657e-32
There are statistically significant differences.
```



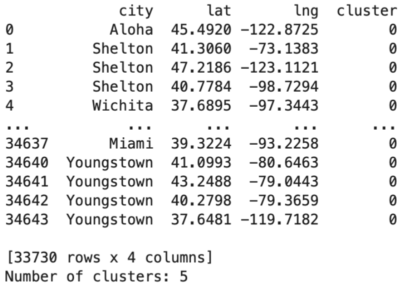
```
Asian              22.602740
Black              15.045135
Black,Hispanic      0.000000
Hispanic           16.045627
Native American    15.384615
Other              28.571429
Unknown            15.286624
White              26.617179
```



```python
# Loading GeoJSON file with geometries of US states
geojson_path = 'usa-cities.geojson'
geojson_data = gpd.read_file(geojson_path)
geojson_data
# Merge the police shooting dataset with the GeoJSON file based on the state or region column
merged_df = pd.merge(geojson_data, df, how='left', left_on='STATE', right_on='state')

# Create a GeoDataFrame
gdf = gpd.GeoDataFrame(merged_df)

# Create a folium map
m = folium.Map(location=[37.7749, -122.4194], zoom_start=4)

from folium.plugins import HeatMap
merged_df = pd.merge(geojson_data, df, how='left', left_on='STATE', right_on='state')

# Create a GeoDataFrame
gdf = gpd.GeoDataFrame(merged_df)

# Create a folium map
m = folium.Map(location=[37.7749, -122.4194], zoom_start=4)

# Create a HeatMap layer for the police shooting incidents
heat_data = [[point.y, point.x] for point in gdf.geometry if point.y and point.x]
HeatMap(heat_data, radius=15, blur=10).add_to(m)
m.save('police_shooting_heatmap_sample.png')
```



```python
#DBSCAN clusters
city_coordinates = pd.read_csv('uscities.csv')
# Merge datasets based on the 'city' column
merged_data = pd.merge(police_shootings, city_coordinates, on='city', how='left')
# Handle missing values by dropping rows with NaN values
merged_data = merged_data.dropna(subset=['lat', 'lng'])

# Extract relevant features for clustering (latitude and longitude)
features = merged_data[['lat', 'lng']]
# Scale the feature
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)
# Apply DBSCAN
epsilon = 0.5
min_samples = 5
dbscan = DBSCAN(eps=epsilon, min_samples=min_samples)
merged_data['cluster'] = dbscan.fit_predict(scaled_features)

# Visualize the clusters or analyze them further based on your needs
print(merged_data[['city', 'lat', 'lng', 'cluster']])

# Print the number of clusters
num_clusters = len(set(dbscan.labels_)) - (1 if -1 in dbscan.labels_ else 0)
print(f"Number of clusters: {num_clusters}")

# Visualize the clusters
plt.figure(figsize=(10, 6))

# Scatter plot for each cluster
for cluster_label in set(merged_data['cluster']):
    cluster_data = merged_data[merged_data['cluster'] == cluster_label]
    plt.scatter(cluster_data['lng'], cluster_data['lat'], label=f'Cluster {cluster_label}')

plt.title('DBSCAN Clustering of Police Shootings')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend()
plt.show()
```



```
             city      lat       lng  cluster
0           Aloha  45.4920 -122.8725        0
1         Shelton  41.3060  -73.1383        0
2         Shelton  47.2186 -123.1121        0
3         Shelton  40.7784  -98.7294        0
4         Wichita  37.6895  -97.3443        0
...           ...      ...       ...      ...
34637       Miami  39.3224  -93.2258        0
34640  Youngstown  41.0993  -80.6463        0
34641  Youngstown  43.2488  -79.0443        0
34642  Youngstown  40.2798  -79.3659        0
34643  Youngstown  37.6481 -119.7182        0

[33730 rows x 4 columns]
Number of clusters: 5
```

Contributions:

All co-authors played an equal part towards the creation of the project.