BENEATH THE BADGE

An Insightful Exploration of Police Shooting Incidents in the USA

Addressed To: Jennifer Jenkins, Steven Rich, Andrew Ba Tran, Paige Moody, Julie Tate, Ted Mellnik

Co-Authored By:

Shubham Singh (02131135)

Sai Gopal Jarabana (02073973)

ISSUES

The datasets are given by **The Washington Post** on their official website and they are updating on weekly basis. There are two datasets that contains the total number of shootings happened in the United States from the year January 2015 to October 2023. The first data contains the information the shootings happened in the different cities of the United States that contains the general background information about how the people shot and their name, age, whether they are armed or not, body camera, and some other information related to the police station.

The second data set includes the information about the police officer along with their id, name, type (Position of an Officer), ORI codes, and the number of encounters they have done.

This report aims to analyze the challenges encountered by the police shooting as per The Washington Post:

- Was the encounter has been done by the police of the United States as per the racial base and number of killings was done from the year from January 2015 to October 2023?
- Were the shootings fair enough to justify the encounter whether the person is armed during an incident?
- Is there any correlation between the two datasets whether the shootings happened by the police officers in different cities was based on the ranking order of the police department and the type of cops?
- How can we justify the encounters by the police as per the time to verify the shootings were unanimous as some of the persons are unarmed while shooting?

FINDINGS

- While hovering to the first dataset which contains the information about the shootings as per the information of people killed from 2015 to 2023. We found that the most encounters have been done in previous year 2022 and the moderate value is ranging in the years from 2015 to 2021. Anyway, the number of shootings happened in every year were kind of similar to the number which is around 1000 and more, except the encounters were being less in this year till now somewhere around 700.

- Later, we tried to analyze the data as per the racial and gender basis and our analysis provides the valuable insights into the relationship between killings happened on the basis of race and gender. We come to the result that the most people killed by the police are belong to White which is around 4000 and the least number of people encountered are Asian, Native Americans, and Others are below 500. The Black and Hispanic and along with the Unknown category where people were not able to defined at the time of killing were in same ratio and the number is around 1500 to 2000. There will be no comparison between the killings in male and female because after visualizing the data on bar graph, it is very evident that the encounters of male were much higher as compare to female in numbers.

- Considering the shot people during an incident whether they were armed or not. To analyze this scenario, I applied the Random Forest regression analysis method of machine learning technique to find out the actual data whether the persons were armed or not during the encounter happened. Initially, I plotted the bar graph to get the glimpse of the data as per the people armed with weapons and I came to conclusion that the most people were carried a gun during an incident. Knife, Unarmed, Undetermined Vehicle, Blunt Object, etc. were in the hierarchical order. I tried to apply random Forest technique in two different models, one considering Race as targeted in which I got the accuracy score 0.48 and for the second predictive model in which I considered Armed as targeted value, I got the accuracy 0.57.

- Our analysis is defined on the basis of clustering technique in which we clustered both the datasets as per the variables and get the outcomes as how it can be relatable to each other. We have applied K-Means clustering technique in both datasets and got the result as clusters defines the behavior of variables as race, armed, city, and state wise. For the second dataset, we tried to clusters the encounters done by the cops as per their officer's ID by which we can define the order of killing of shooting is based on the ranking or department.

- We analyze and visualize the encounters done by the police in different cities of the United States and the information of the killed people along with the information of cops who encountered them. To do so, we tried to use the altair library technique to plot the datasets on Geo Histogram so it would be easy to understand the correlations between the two datasets.

- We have applied Monte-Carlo approximation in our analysis for addressing the complexity of the data points and find out the approximation value for age mean and estimation of standard deviation.

- To understand and interpret patterns of the data points and the trends and behavior of shootings happened as per the time flows from 2015 to 2023, we used the Time series analysis technique and found the forecasting, prediction, and pattern recognition of the killings happened in different cities.

DISCUSSION

- To get the estimation for the data with respect to people who were armed and the number of encounters happened during an incident whether the person is equipped with gun or not. Found out the approximation values, checked for the estimation on the basis of police shootings for the people were killed in the different age category.

- To determine the shootings happened in the country was fair enough or not, we have analyzed that the data given as per the Washington Post till date is likely to justify the encounters happened for those people who were unarmed were in the scenario of being encountered. As per our analysis and the predictive model we found that the number of people who were unarmed while shooting with 0.57 accuracy rate as per the regression analysis.

- To analyze the scenario of the police shootings happened in the United States, we focused on the race, gender and people with armed. Spatial analysis is done by our side to justify the shootings happened as per the racial basis or not. We tried to determine the key factors among each and every variable. After doing so, we got the result that there are many variations are distinguished in each variable except as compared to the city.

- We extract the correlations between the police shooting happened in the United States where we can define the number of encounters has been done by the police officer as per the detailed information of the killed people. With this, we found the maximum numbers of shootings happened in the country and the outliers revealed the top 3 states of the country as California, Arizona, and Colorado. If roughly we will hover to the map, we found that the more shootings have been done in the eastern part of country. The officer's ranking IDs till 5000 are mainly responsible for the encounters as they more shooting records in their career.

- Moreover, we understood and interpret the patterns, trends, and the sequential data points where police shootings were evident as per the time. It can relatable to the encounters happened in the regions as per the day and date is varying from 2015 to 2023 and forecasted the most shootings happened in the country at the early times of 2015.

Appendix A: Method

Data Collection:

The datasets for this project provide a detailed record of police shooting incidents across the United States from January 2015 to October 2023. These datasets were rigorously cleaned and prepared for analysis. To ensure the integrity of our analysis, missing values in the dataset were addressed as follows:

Age: Replaced with the median value.

Categorical Fields (Name, Gender, City, Flee): Filled with 'Unknown'.

Armed Status: Imputed using the most frequent (mode) value.

These steps were taken to minimize data distortion and maintain the dataset's original distribution.

Variable Creation:

The analysis focused on various key variables, including:

- **Date**: Incident date, crucial for time series analysis.
- **Demographic Data**: Age, gender, and race of the individuals involved.
- **Armed Status and Mental Illness**: Data on whether the individual was armed and indications of mental illness.
- **Geographical Information**: City, state, and exact location coordinates (longitude and latitude) of each incident.

Analytic Methods:

Various statistical and analytical methods were employed to dissect and understand the data:

- **Geospatial Analysis**: Utilizing longitude and latitude data to create maps that visually represent the distribution of incidents across different regions to get the correlation in between both the datasets.
- **Time Series Analysis**: Identification of yearly and daily trends using Seasonal Decomposition and SARIMAX modeling to forecast future incidents and understand temporal patterns.
- **K-Means Clustering**: This machine learning technique categorized patterns within the data, such as clustering incidents based on key variable similarities to get the insights for the encounters done by the cops, used Elbow method to find optimal number of clusters.
- **Monte Carlo Simulation**: Used to estimate the average age distribution of individuals involved in these incidents, adding a probabilistic dimension to the analysis.
- **Correlation Analysis**: Creation of a heatmap to visually analyze the correlations between various variables.
- **Random Forest Regression**: Employed to predict the likelihood of an individual being armed during a police encounter. This machine learning model was trained on factors such as demographic data, race, and people with armed to create a predictive model that estimates the probability of an individual being armed during an incident.

These analyses were conducted using Python and its extensive library ecosystem, including pandas for data handling, matplotlib and seaborn for visualization, scikit-learn for machine learning techniques. This methodological approach was designed to provide a comprehensive and multi-dimensional understanding of the dynamics and patterns in police shooting incidents.

Appendix B: Result

Visualizing the data set on the Geo Histogram which is used to determine the density of the total shootings happened in the United States.

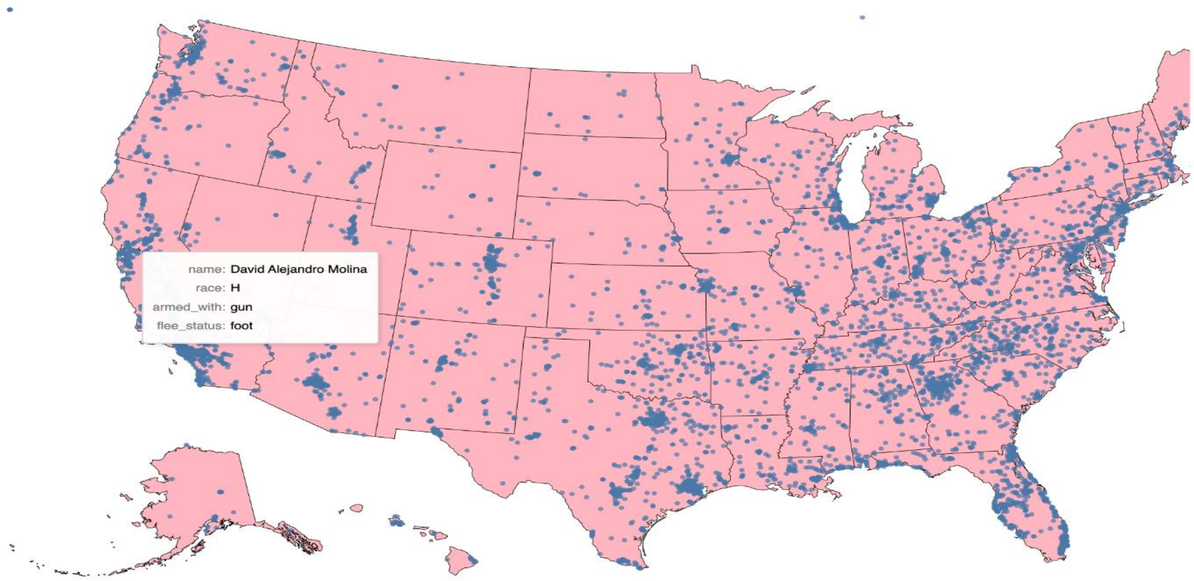
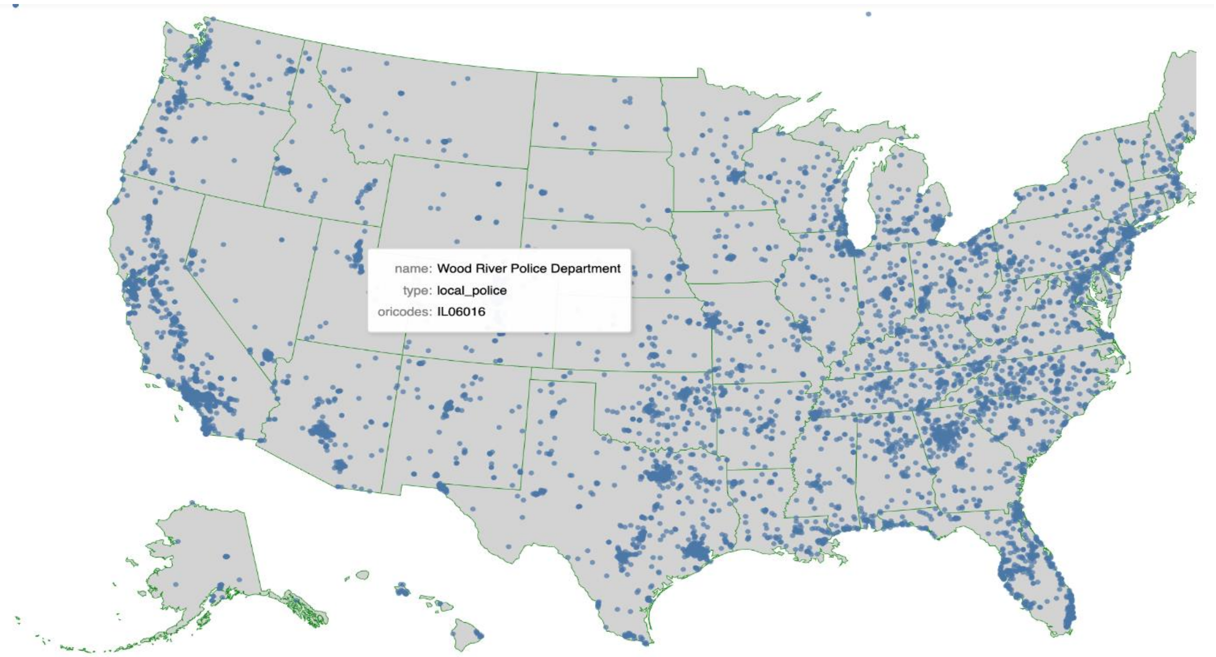


Fig: Total Shootings happened in the country as per killed people information

The data set shows that the eastern region is having more crime as compared to the western region. The total shootings happened in the country shows that the encounters by the police have been done the in eastern region more. But on the other hand, If we will hover the map properly, we can see that the highest encounters have bene done in California, Arizona and then in Colorado. States.One can refer in the map that there is a pop up showing the information about the killed people by the police.



name: Wood River Police Department
type: local_police
oricodes: IL06016

Fig: Encounters done by the cops with the information listed as dots

The above picture shows, the information about the cop who has been involved in an encounter in different regions. The pop up is showing in the map for each and every dot are the information about the police officer.

We separated the variables according to race and gender based on the police shootings. After conducting the analysis, we were able to determine that White people had the highest number of contacts, followed by Black people in terms of gunshot frequency. While the number of gunshots involving Asians, Native Americans, and other people is somewhat similar, the number of killings between the Unknown and Hispanics is almost the same. As we previously stated in my blog, there is a significant variation in the number of gunshots based on gender, with male interactions being significantly greater than female encounters. See the figure below to get a sense of the gunshots that occurred for the two variables Race and Gender.

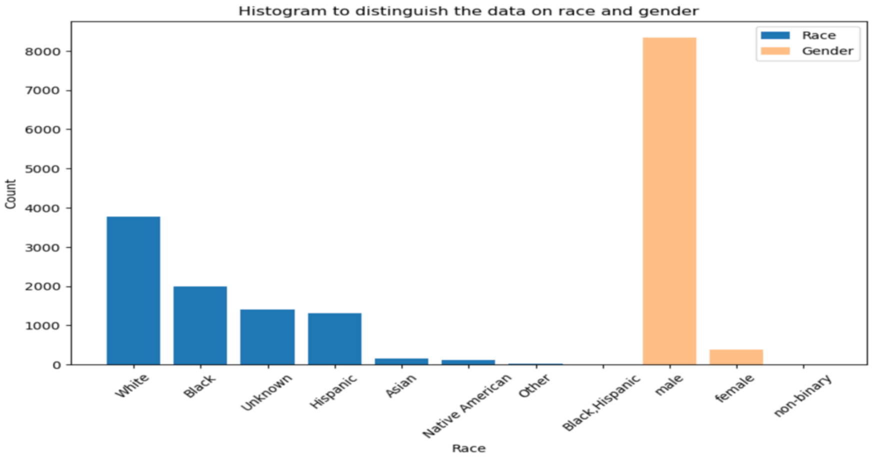


Fig: Bar Graph to show the Killings as per Race and Gender

The analysis of data spanning 2015 to 2023 shows a notable trend in the frequency of police shooting incidents. Starting in 2015, the number of incidents each year shows fluctuations, with an overall upward trend peaking in 2022. This trend highlights a significant increase over the years, underlining the growing magnitude of such incidents.

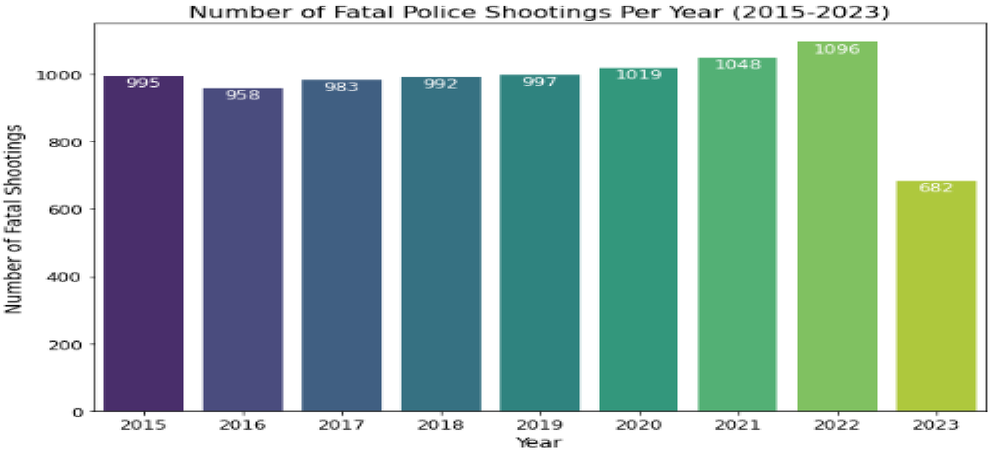


Fig: Yearly Distribution of Police Shooting Incidents (2015-2023)

The investigation into incidents by day of the week unveils a varied distribution. The data indicates a somewhat uniform distribution of incidents throughout the week, with slight variances. Wednesday emerges as the day with the highest number of incidents, followed closely by Tuesday and Thursday. In contrast, the weekend days (Saturday and Sunday) and the beginning of the week (Monday) show slightly lower frequencies.

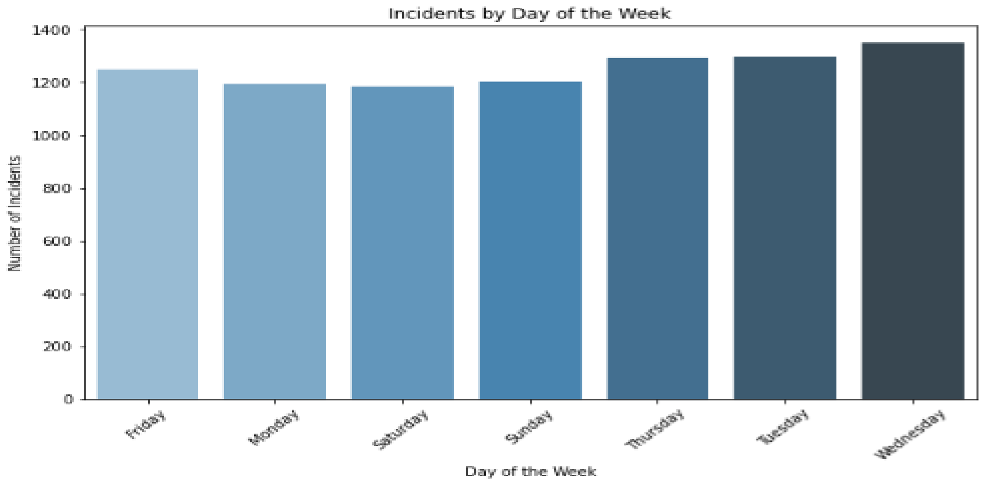
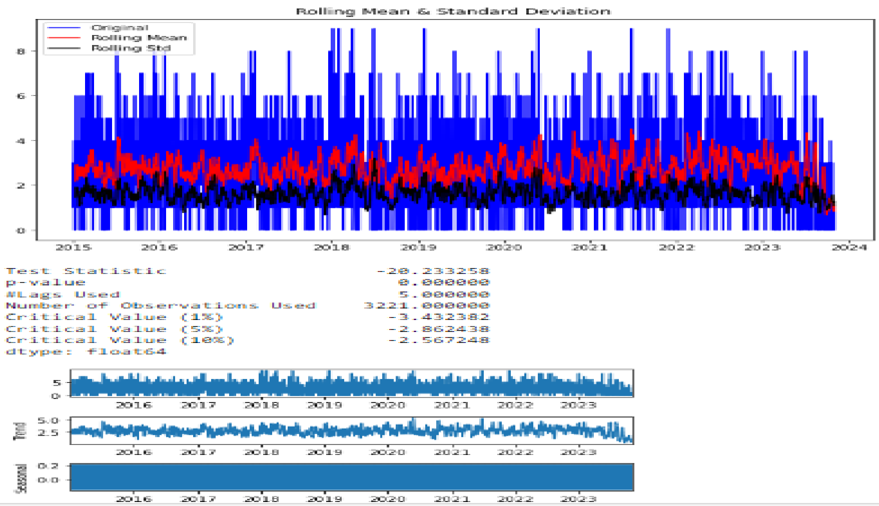


Fig: Weekly Distribution of Police Shooting Incidents by Day

The time series analysis conducted on police shooting incidents has yielded significant insights into the patterns and predictability of these events. Utilizing the SARIMAX model, the study confirmed the stationarity of the time series, indicating that the properties such as mean and variance do not vary over time. This stationarity makes the series appropriate for forecasting and further analysis. A notable aspect of the model is the significance of the Moving Average (MA) term, highlighting that past errors significantly influence current values. This finding reveals a strong temporal dependency within the data, suggesting that recent events have a considerable impact on future outcomes.



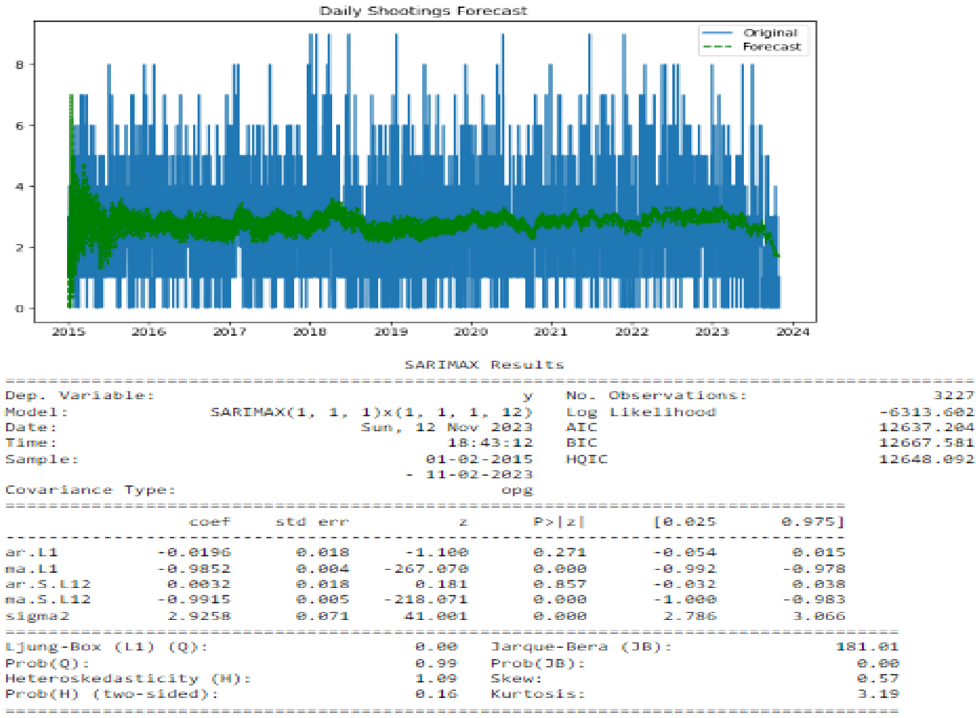
```
Test Statistic                  -20.233258
p-value                           0.000000
#Lags Used                        5.000000
Number of Observations Used    3221.000000
Critical Value (1%)              -3.432382
Critical Value (5%)              -2.862438
Critical Value (10%)             -2.567248
dtype: float64
```

Fig: Time Series Analysis of Police Shootings - Rolling Mean and Standard Deviation

The adequacy of the model's fit was evident from its AIC and BIC values, along with the results of the Ljung-Box test, which collectively indicated that the model captures essential patterns in the time series without overfitting. However, the Jarque-Bera test pointed to some deviations from normality in the residuals, implying that certain aspects of the data might not be fully captured by the model. This could be due to outliers or non-linear relationships not accounted for by a SARIMAX model.

Crucially, the analysis suggests a degree of predictability in police shooting incidents, with the model providing a quantitative foundation for understanding their dynamics. This aspect is particularly significant for law enforcement agencies for strategic planning and resource allocation. The forecasting potential of the model, which can predict future values with reasonable accuracy, offers a valuable tool for anticipating such incidents.

Overall, the study not only confirms the predictability of police shooting incidents but also highlights the importance of historical context in understanding and anticipating these events. It presents a framework that can be instrumental for future studies or policy development in this domain, underscoring the role of temporal patterns in shaping the nature and frequency of police shooting incidents



```
                                     SARIMAX Results
========================================================================================
Dep. Variable:                                  y   No. Observations:                 3227
Model:             SARIMAX(1, 1, 1)x(1, 1, 1, 12)   Log Likelihood               -6313.602
Date:                            Sun, 12 Nov 2023   AIC                          12637.204
Time:                                    18:43:12   BIC                          12667.581
Sample:                                01-02-2015   HQIC                         12648.092
                                     - 11-02-2023
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.0196      0.018     -1.100      0.271      -0.054       0.015
ma.L1         -0.9852      0.004   -267.070      0.000      -0.992      -0.978
ar.S.L12       0.0032      0.018      0.181      0.857      -0.032       0.038
ma.S.L12      -0.9915      0.005   -218.071      0.000      -1.000      -0.983
sigma2         2.9258      0.071     41.001      0.000       2.786       3.066
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):               181.01
Prob(Q):                              0.99   Prob(JB):                         0.00
Heteroskedasticity (H):               1.09   Skew:                             0.57
Prob(H) (two-sided):                  0.16   Kurtosis:                         3.19
===================================================================================
```

Fig: Forecasting Police Shootings with SARIMAX Model - Results and Predictions

We utilized the Elbow Method to identify the optimal number of clusters for k-means clustering. This approach helped us determine the most appropriate cluster count to effectively segment the data, ensuring more precise and meaningful insights from our clustering analysis.

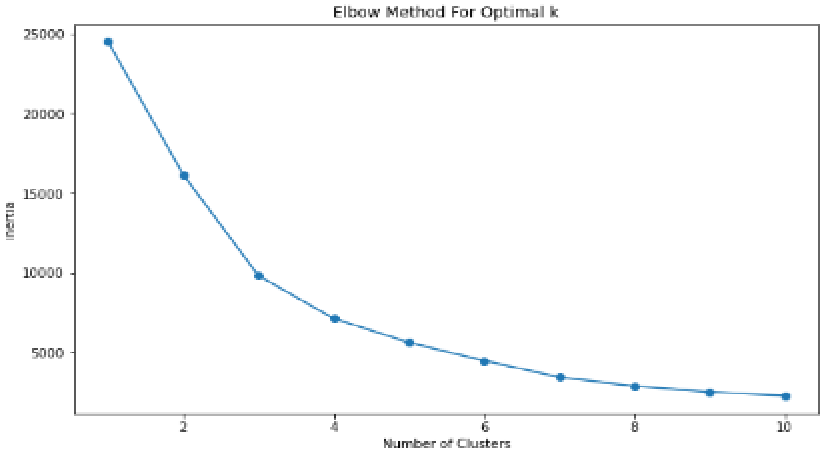


Fig: The plot shows the clusters for the dataset in which K=10 clusters

K-means Clustering applied for the second dataset where we got the output as the number of encounters has been done by the police as per their ID.

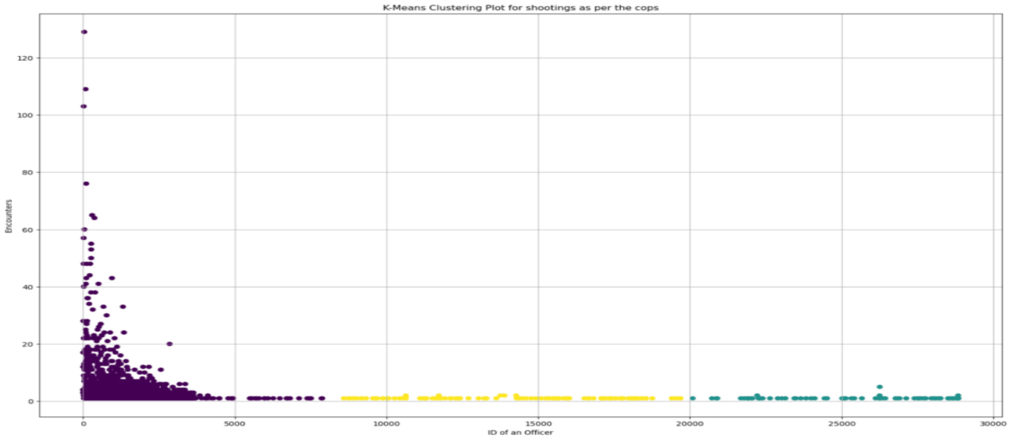
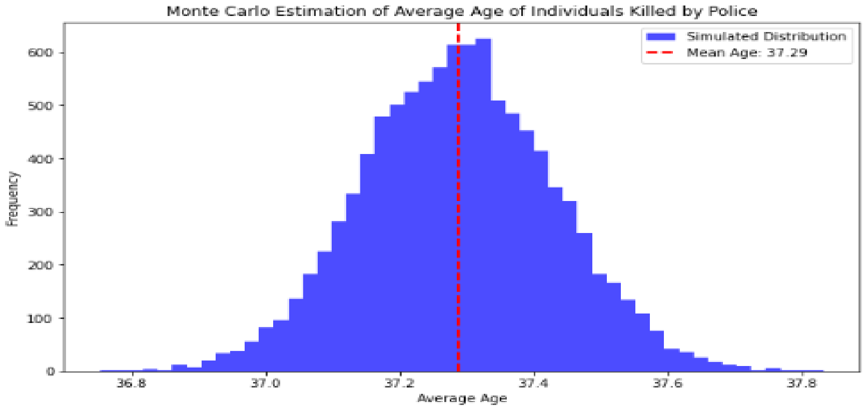


Fig: The Scatter plot shows the number of encounters done a cop

The scatter plot gives the information that the ID of a police from 0 to 5000 carrying some ranking which are highly recommended for the encounter because they have more encounter records in their career. As per the analysis of the graph the maximum encounter has been done by the single police is 125.

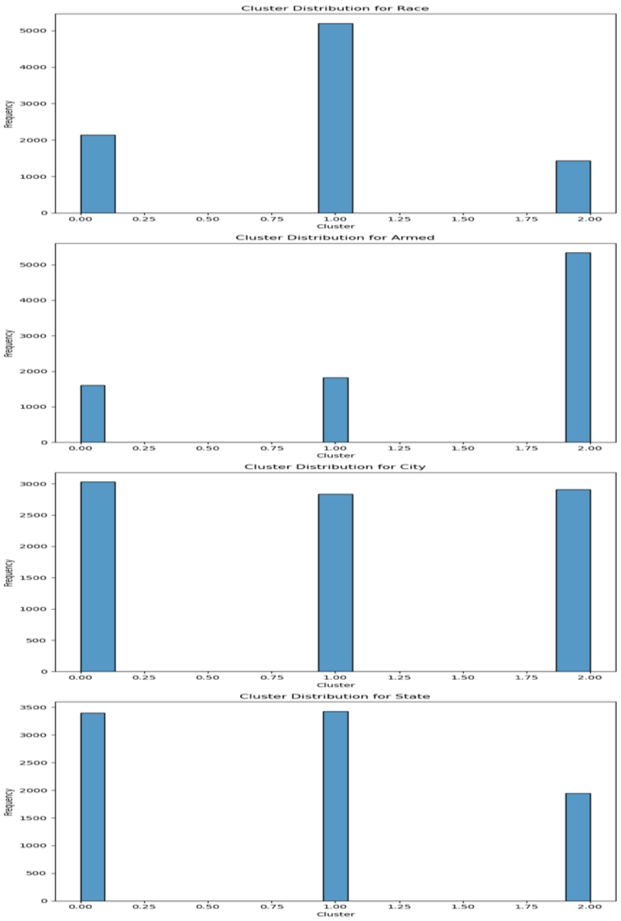
In the Monte Carlo simulation, we estimated the mean age of individuals involved in police shooting incidents to be approximately 37.29 years, with a relatively low standard deviation of 0.143. These findings suggest that the majority of these incidents involve individuals in their late thirties, a demographic that has significant implications for community and family structures. This analysis is crucial for policymakers and law enforcement agencies to develop targeted interventions and policies to address and potentially reduce the incidence of police shootings.



```
Estimated Mean Age: 37.28781464788732
Standard Deviation: 0.14291728649828825
```

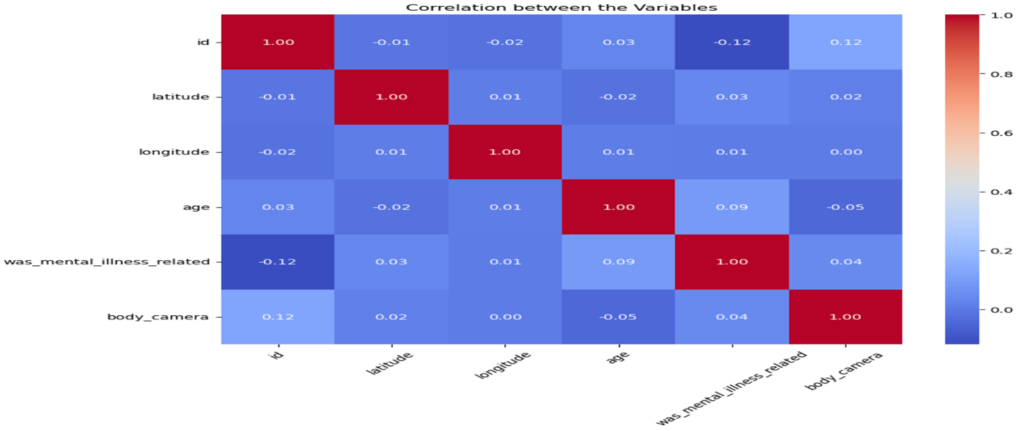
Fig: Distribution of Estimated Mean Age from Monte Carlo Simulation

The three clusters for the four variables as Race, Armed, City, and State are plotted below:

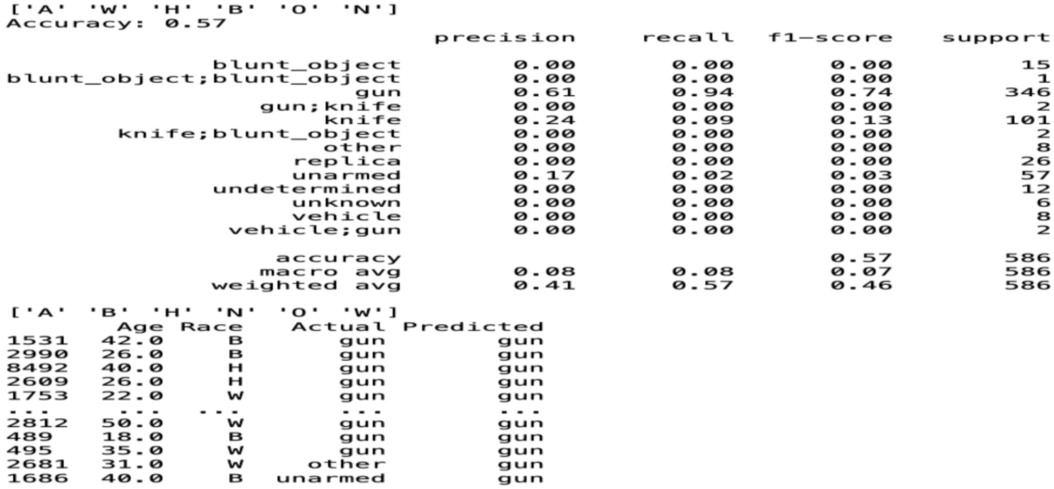


By seeing the above figure, we can define that the clusters for all variables are distinguished in variation except the city variable. The clusters for the city are being similar to one another. This shows the that the killings are quite occurred in everywhere in the country.

We performed the ANOVA analysis for the determination of the variables and unraveling the dots in between them. To do this, we plot the data variables on Heat Map to check how each variable are in correlation. Considering the major variables as id, age, sign of mental illness, and body camera of the person.



For the high predictive accuracy of the model and reduction of variance considering as the data cleaning approach, we tried the Random Forest machine learning technique and ensemble learning. It involves combining the predictions of multiple models to improve overall performance and robustness.



```
['A' 'W' 'H' 'B' 'O' 'N']
Accuracy: 0.57
                               precision    recall  f1-score   support

                 blunt_object       0.00      0.00      0.00        15
    blunt_object;blunt_object       0.00      0.00      0.00         1
                          gun       0.61      0.94      0.74       346
                    gun;knife       0.00      0.00      0.00         2
                        knife       0.24      0.09      0.13       101
           knife;blunt_object       0.00      0.00      0.00         2
                        other       0.00      0.00      0.00         8
                      replica       0.00      0.00      0.00        26
                      unarmed       0.17      0.02      0.03        57
                 undetermined       0.00      0.00      0.00        12
                      unknown       0.00      0.00      0.00         6
                      vehicle       0.00      0.00      0.00         8
                  vehicle;gun       0.00      0.00      0.00         2

                     accuracy                           0.57       586
                    macro avg       0.08      0.08      0.07       586
                 weighted avg       0.41      0.57      0.46       586
['A' 'B' 'H' 'N' 'O' 'W']
       Age Race   Actual Predicted
1531  42.0    B      gun       gun
2990  26.0    B      gun       gun
8492  40.0    H      gun       gun
2609  26.0    H      gun       gun
1753  22.0    W      gun       gun
...    ...  ...      ...       ...
2812  50.0    W      gun       gun
489   18.0    B      gun       gun
495   35.0    W      gun       gun
2681  31.0    W    other       gun
1686  40.0    B  unarmed       gun
```

The above picture shows the outcomes for the Random Forest Regression Analysis in which we considered Armed as a targeted factor and Race and Age as featured variables. The Accuracy for the model we achieved as 0.57 and the precision, f-1 score, and support are mentioned as the data types in variable. At last, the it shows the actual and predicted output for the armed people during the incident happened.

Appendix C: Data and Code

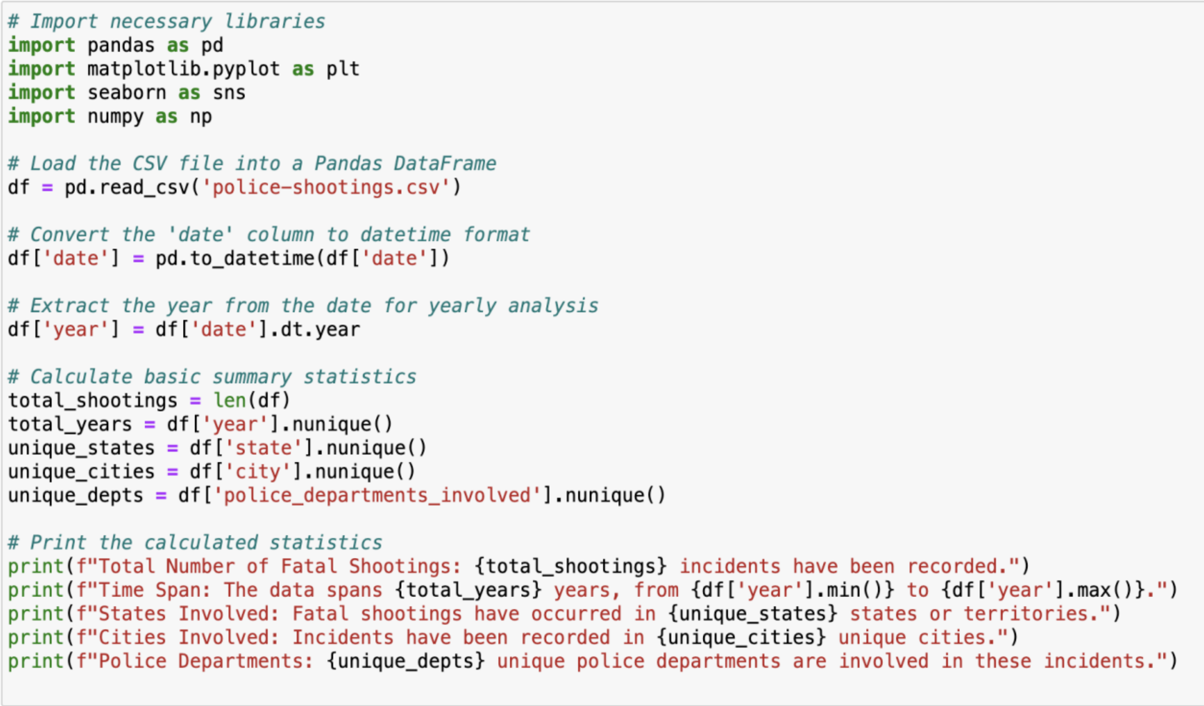
In this comprehensive analysis, we leveraged the powerful Python programming environment, primarily using Jupyter Notebooks within the Anaconda distribution. Our work extensively utilized a suite of Python libraries, which were pivotal in handling various aspects of the project. These libraries included pandas for data manipulation, matplotlib and seaborn for data visualization, statsmodels for time series analysis, and sklearn for machine learning tasks.

Particularly noteworthy in our approach was the tailored use of specific Python libraries and code structures to address distinct analytical needs. For time series analysis, we relied on the SARIMAX model from the statsmodels library, which was crucial in forecasting and understanding the trends in police shooting incidents. The Monte Carlo simulation, implemented to estimate the mean age of individuals involved in these incidents, was another highlight, showcasing the versatility of Python in statistical simulation.

Furthermore, we employed geospatial analysis techniques using longitude and latitude data, providing a geographical dimension to our findings. The analysis also extended to investigating correlations and conducting random forest regression to predict the likelihood of an armed encounter during police incidents. This multi-faceted approach, combining various statistical and machine learning methodologies, underlines the depth and breadth of our analytical capabilities within the Python ecosystem.

The codes in sequential workflows are as follows:

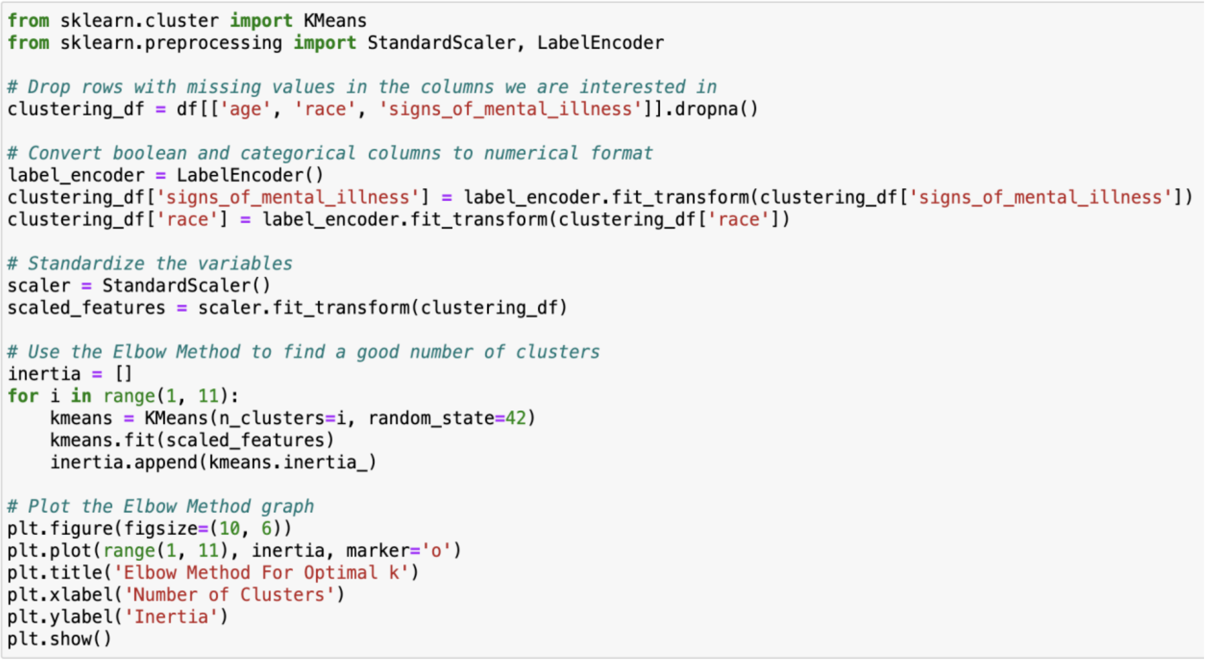
- Importing data and Summary Statistics.



```python
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Load the CSV file into a Pandas DataFrame
df = pd.read_csv('police-shootings.csv')

# Convert the 'date' column to datetime format
df['date'] = pd.to_datetime(df['date'])

# Extract the year from the date for yearly analysis
df['year'] = df['date'].dt.year

# Calculate basic summary statistics
total_shootings = len(df)
total_years = df['year'].nunique()
unique_states = df['state'].nunique()
unique_cities = df['city'].nunique()
unique_depts = df['police_departments_involved'].nunique()

# Print the calculated statistics
print(f"Total Number of Fatal Shootings: {total_shootings} incidents have been recorded.")
print(f"Time Span: The data spans {total_years} years, from {df['year'].min()} to {df['year'].max()}.")
print(f"States Involved: Fatal shootings have occurred in {unique_states} states or territories.")
print(f"Cities Involved: Incidents have been recorded in {unique_cities} unique cities.")
print(f"Police Departments: {unique_depts} unique police departments are involved in these incidents.")
```

Fig: Code for imports and stats of the dataset

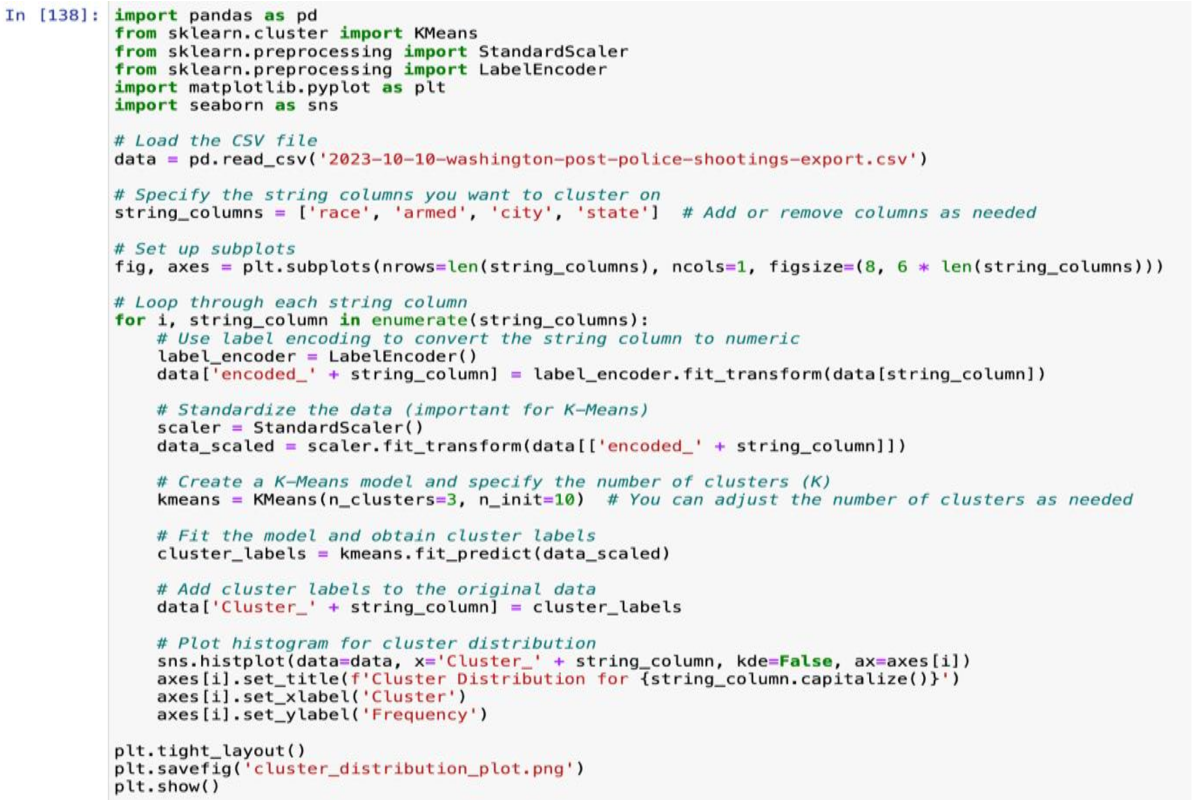
- Code to find optimal number of Clusters (Elbow Method)



```python
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Drop rows with missing values in the columns we are interested in
clustering_df = df[['age', 'race', 'signs_of_mental_illness']].dropna()

# Convert boolean and categorical columns to numerical format
label_encoder = LabelEncoder()
clustering_df['signs_of_mental_illness'] = label_encoder.fit_transform(clustering_df['signs_of_mental_illness'])
clustering_df['race'] = label_encoder.fit_transform(clustering_df['race'])

# Standardize the variables
scaler = StandardScaler()
scaled_features = scaler.fit_transform(clustering_df)

# Use the Elbow Method to find a good number of clusters
inertia = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(scaled_features)
    inertia.append(kmeans.inertia_)

# Plot the Elbow Method graph
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method For Optimal k')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.show()
```

Fig: Code for Elbow Method

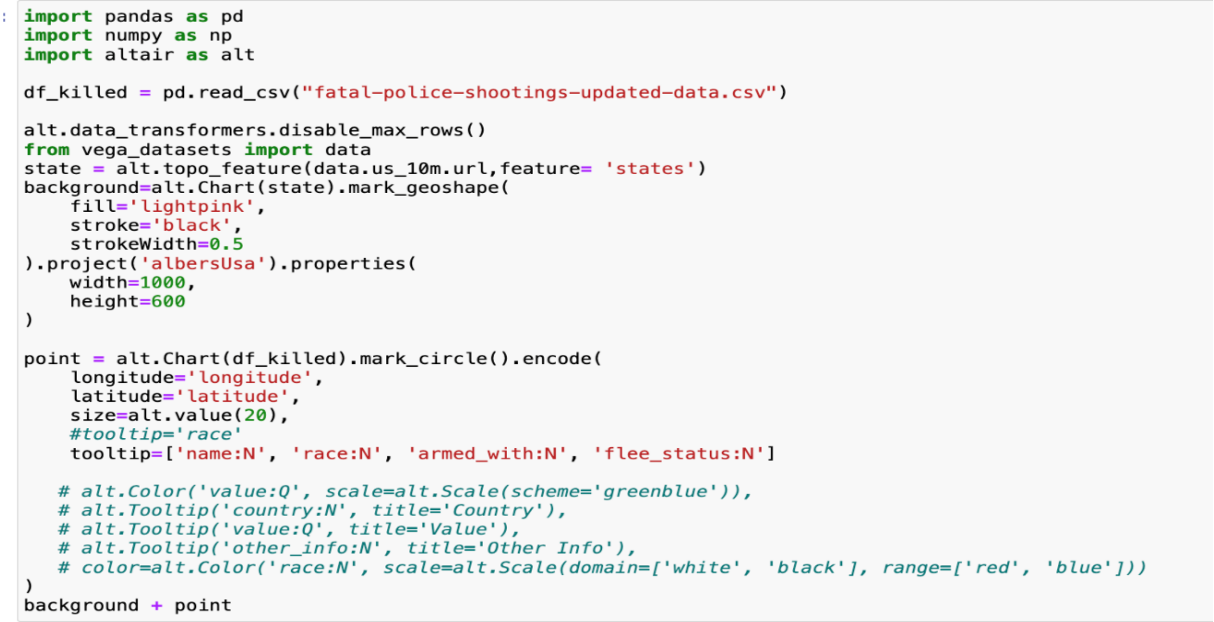
- Code to perform the K-means Clustering.



```python
In [138]: import pandas as pd
          from sklearn.cluster import KMeans
          from sklearn.preprocessing import StandardScaler
          from sklearn.preprocessing import LabelEncoder
          import matplotlib.pyplot as plt
          import seaborn as sns

          # Load the CSV file
          data = pd.read_csv('2023-10-10-washington-post-police-shootings-export.csv')

          # Specify the string columns you want to cluster on
          string_columns = ['race', 'armed', 'city', 'state']  # Add or remove columns as needed

          # Set up subplots
          fig, axes = plt.subplots(nrows=len(string_columns), ncols=1, figsize=(8, 6 * len(string_columns)))

          # Loop through each string column
          for i, string_column in enumerate(string_columns):
              # Use label encoding to convert the string column to numeric
              label_encoder = LabelEncoder()
              data['encoded_' + string_column] = label_encoder.fit_transform(data[string_column])

              # Standardize the data (important for K-Means)
              scaler = StandardScaler()
              data_scaled = scaler.fit_transform(data[['encoded_' + string_column]])

              # Create a K-Means model and specify the number of clusters (K)
              kmeans = KMeans(n_clusters=3, n_init=10)  # You can adjust the number of clusters as needed

              # Fit the model and obtain cluster labels
              cluster_labels = kmeans.fit_predict(data_scaled)

              # Add cluster labels to the original data
              data['Cluster_' + string_column] = cluster_labels

              # Plot histogram for cluster distribution
              sns.histplot(data=data, x='Cluster_' + string_column, kde=False, ax=axes[i])
              axes[i].set_title(f'Cluster Distribution for {string_column.capitalize()}')
              axes[i].set_xlabel('Cluster')
              axes[i].set_ylabel('Frequency')

          plt.tight_layout()
          plt.savefig('cluster_distribution_plot.png')
          plt.show()
```

Fig: Code for K-Means Clustering Technique

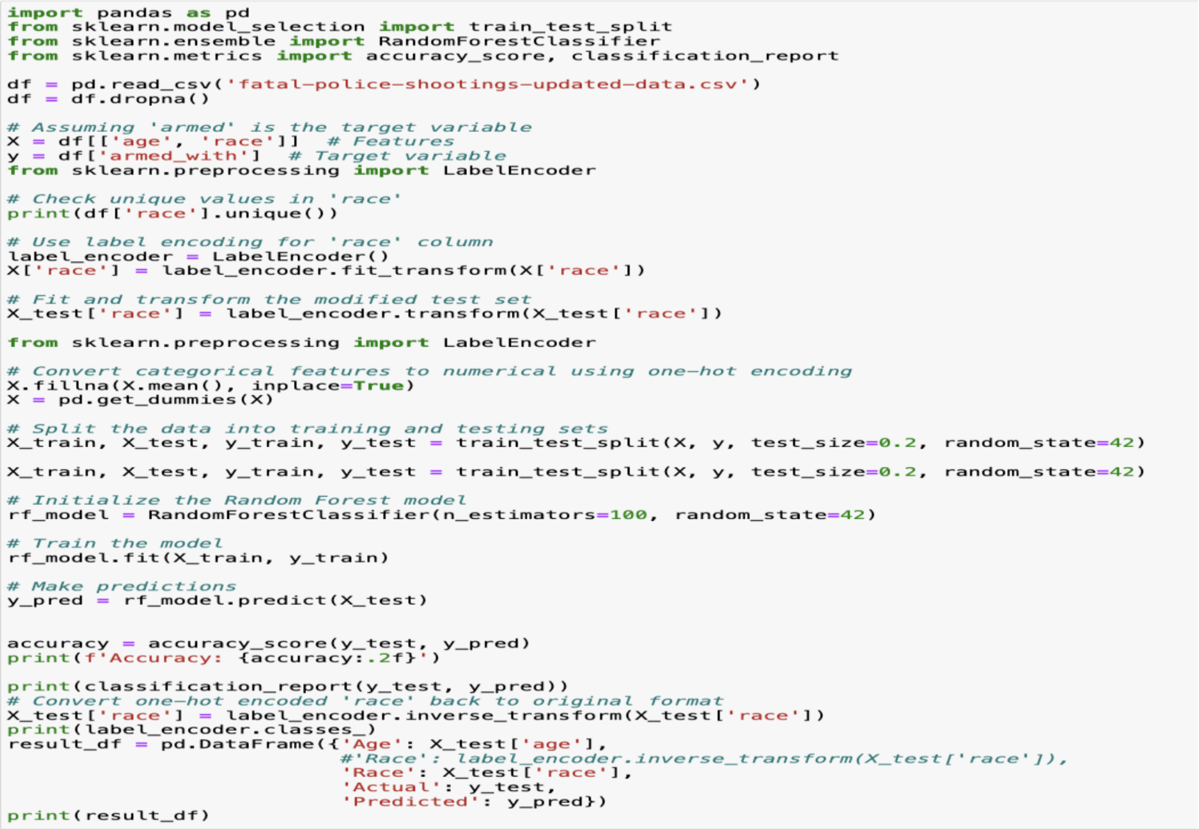
- Code to plot the Geo Histogram and altair library for building on top of the Vega Visualization.



```
import pandas as pd
import numpy as np
import altair as alt

df_killed = pd.read_csv("fatal-police-shootings-updated-data.csv")

alt.data_transformers.disable_max_rows()
from vega_datasets import data
state = alt.topo_feature(data.us_10m.url,feature= 'states')
background=alt.Chart(state).mark_geoshape(
    fill='lightpink',
    stroke='black',
    strokeWidth=0.5
).project('albersUsa').properties(
    width=1000,
    height=600
)

point = alt.Chart(df_killed).mark_circle().encode(
    longitude='longitude',
    latitude='latitude',
    size=alt.value(20),
    #tooltip='race'
    tooltip=['name:N', 'race:N', 'armed_with:N', 'flee_status:N']

   # alt.Color('value:Q', scale=alt.Scale(scheme='greenblue')),
   # alt.Tooltip('country:N', title='Country'),
   # alt.Tooltip('value:Q', title='Value'),
   # alt.Tooltip('other_info:N', title='Other Info'),
   # color=alt.Color('race:N', scale=alt.Scale(domain=['white', 'black'], range=['red', 'blue']))
)
background + point
```

Fig: Code for the Geo Histogram to visualize the data on the map.

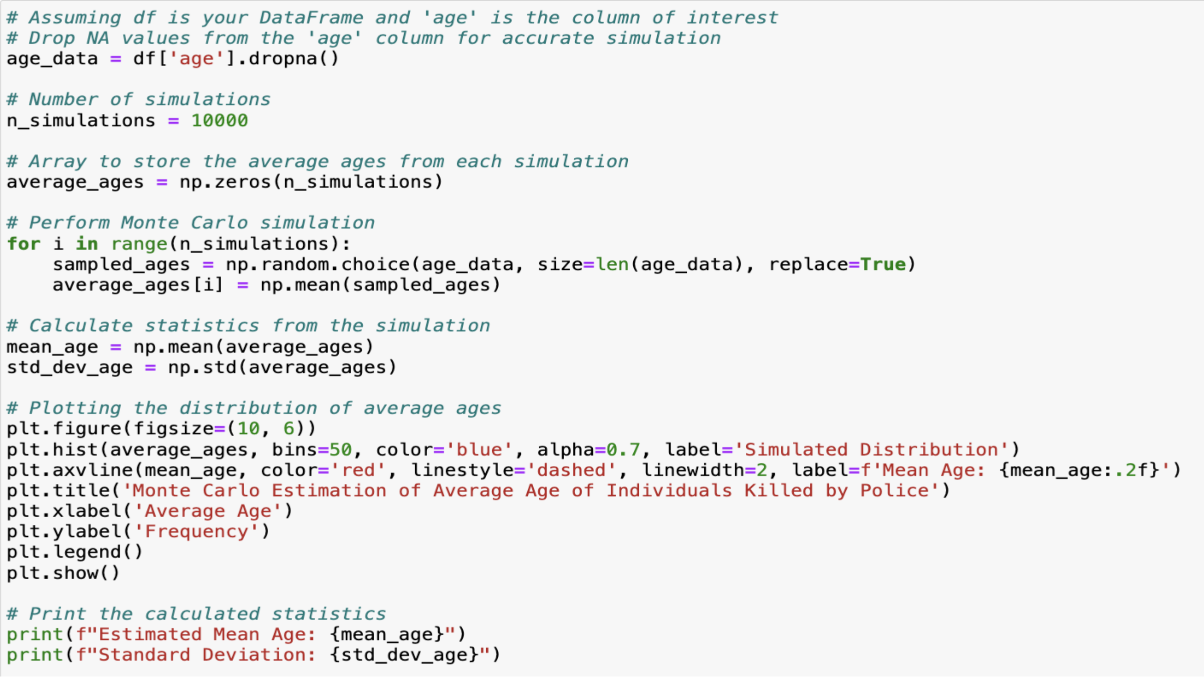
- Implementation Random Forest Regression Analysis and imported randomforestclassifier library.



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

df = pd.read_csv('fatal-police-shootings-updated-data.csv')
df = df.dropna()

# Assuming 'armed' is the target variable
X = df[['age', 'race']]  # Features
y = df['armed_with']  # Target variable
from sklearn.preprocessing import LabelEncoder

# Check unique values in 'race'
print(df['race'].unique())

# Use label encoding for 'race' column
label_encoder = LabelEncoder()
X['race'] = label_encoder.fit_transform(X['race'])

# Fit and transform the modified test set
X_test['race'] = label_encoder.transform(X_test['race'])

from sklearn.preprocessing import LabelEncoder

# Convert categorical features to numerical using one-hot encoding
X.fillna(X.mean(), inplace=True)
X = pd.get_dummies(X)

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

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize the Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model
rf_model.fit(X_train, y_train)

# Make predictions
y_pred = rf_model.predict(X_test)


accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

print(classification_report(y_test, y_pred))
# Convert one-hot encoded 'race' back to original format
X_test['race'] = label_encoder.inverse_transform(X_test['race'])
print(label_encoder.classes_)
result_df = pd.DataFrame({'Age': X_test['age'],
                          #'Race': label_encoder.inverse_transform(X_test['race']),
                          'Race': X_test['race'],
                          'Actual': y_test,
                          'Predicted': y_pred})
print(result_df)
```

Fig: Code for the implementation of Random Forest Analysis in the dataset.

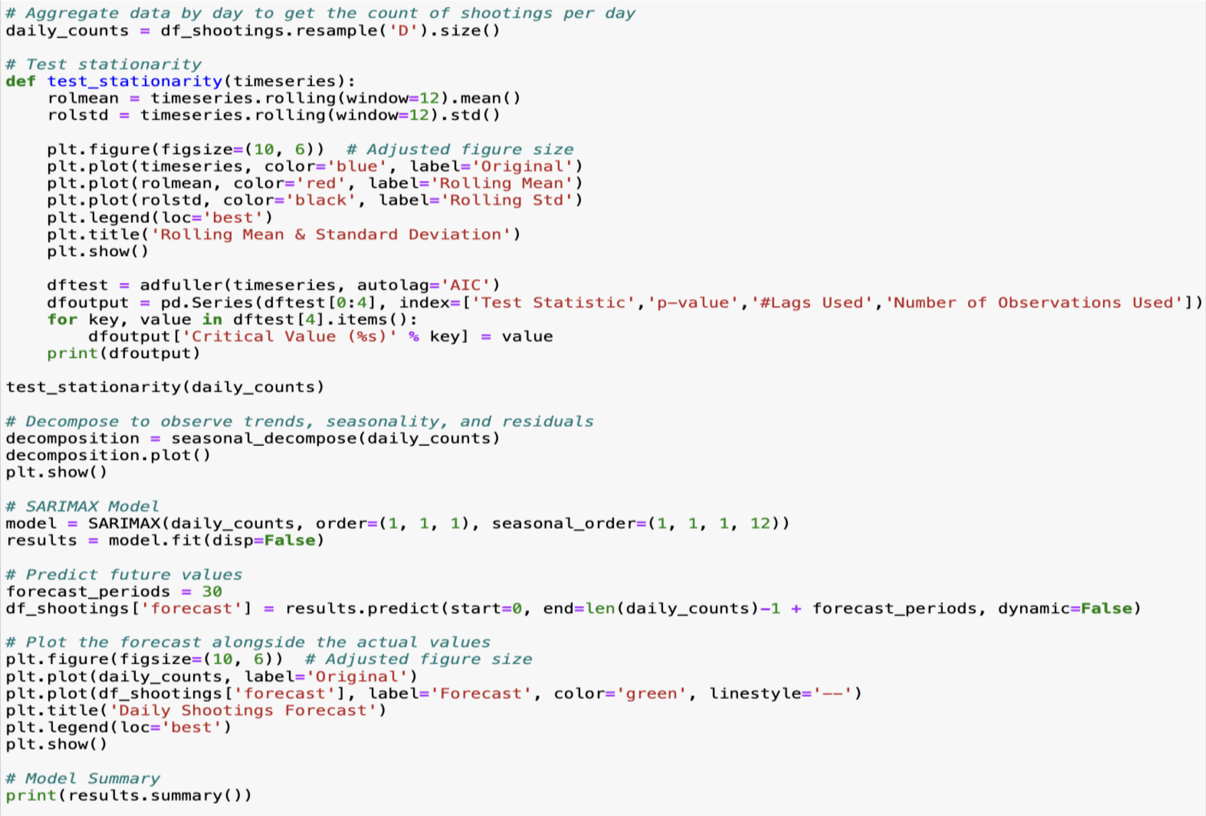
- Code to perform the Monte Carlo Simulation.



```python
# Assuming df is your DataFrame and 'age' is the column of interest
# Drop NA values from the 'age' column for accurate simulation
age_data = df['age'].dropna()

# Number of simulations
n_simulations = 10000

# Array to store the average ages from each simulation
average_ages = np.zeros(n_simulations)

# Perform Monte Carlo simulation
for i in range(n_simulations):
    sampled_ages = np.random.choice(age_data, size=len(age_data), replace=True)
    average_ages[i] = np.mean(sampled_ages)

# Calculate statistics from the simulation
mean_age = np.mean(average_ages)
std_dev_age = np.std(average_ages)

# Plotting the distribution of average ages
plt.figure(figsize=(10, 6))
plt.hist(average_ages, bins=50, color='blue', alpha=0.7, label='Simulated Distribution')
plt.axvline(mean_age, color='red', linestyle='dashed', linewidth=2, label=f'Mean Age: {mean_age:.2f}')
plt.title('Monte Carlo Estimation of Average Age of Individuals Killed by Police')
plt.xlabel('Average Age')
plt.ylabel('Frequency')
plt.legend()
plt.show()

# Print the calculated statistics
print(f"Estimated Mean Age: {mean_age}")
print(f"Standard Deviation: {std_dev_age}")
```

Fig: Monte Carlo Approximation for the analysis of the mean age

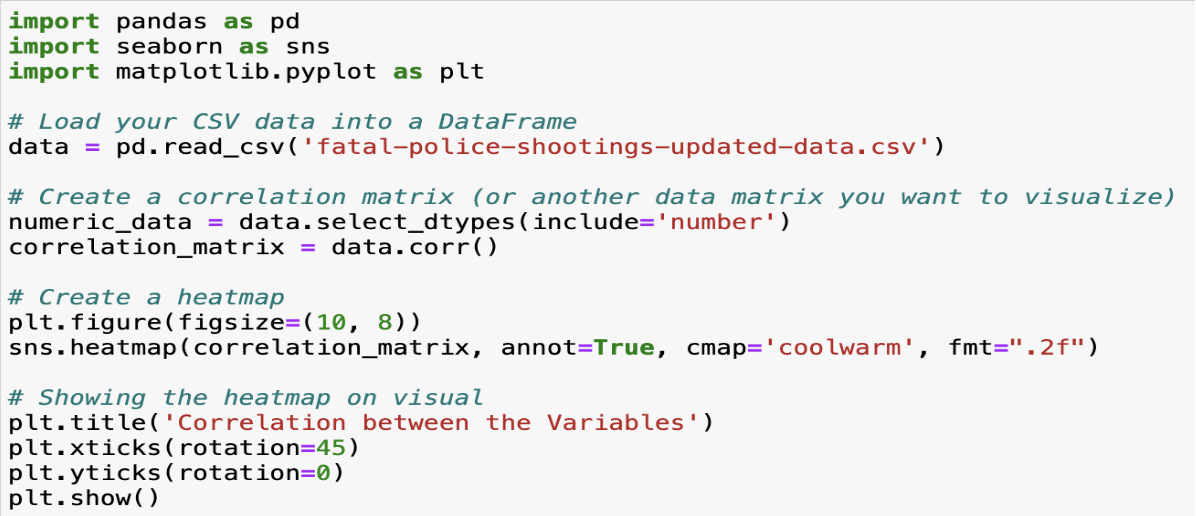
- For the purpose of Time Series analysis.



```python
# Aggregate data by day to get the count of shootings per day
daily_counts = df_shootings.resample('D').size()

# Test stationarity
def test_stationarity(timeseries):
    rolmean = timeseries.rolling(window=12).mean()
    rolstd = timeseries.rolling(window=12).std()

    plt.figure(figsize=(10, 6))  # Adjusted figure size
    plt.plot(timeseries, color='blue', label='Original')
    plt.plot(rolmean, color='red', label='Rolling Mean')
    plt.plot(rolstd, color='black', label='Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show()

    dftest = adfuller(timeseries, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])
    for key, value in dftest[4].items():
        dfoutput['Critical Value (%s)' % key] = value
    print(dfoutput)

test_stationarity(daily_counts)

# Decompose to observe trends, seasonality, and residuals
decomposition = seasonal_decompose(daily_counts)
decomposition.plot()
plt.show()

# SARIMAX Model
model = SARIMAX(daily_counts, order=(1, 1, 1), seasonal_order=(1, 1, 1, 12))
results = model.fit(disp=False)

# Predict future values
forecast_periods = 30
df_shootings['forecast'] = results.predict(start=0, end=len(daily_counts)-1 + forecast_periods, dynamic=False)

# Plot the forecast alongside the actual values
plt.figure(figsize=(10, 6))  # Adjusted figure size
plt.plot(daily_counts, label='Original')
plt.plot(df_shootings['forecast'], label='Forecast', color='green', linestyle='--')
plt.title('Daily Shootings Forecast')
plt.legend(loc='best')
plt.show()

# Model Summary
print(results.summary())
```

Fig: Code Snippet for Time Series Analysis with Rolling Statistics and SARIMAX Forecasting

- Code to ANOVA and showing the Heat Map visuals along with seaborn library.



```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load your CSV data into a DataFrame
data = pd.read_csv('fatal-police-shootings-updated-data.csv')

# Create a correlation matrix (or another data matrix you want to visualize)
numeric_data = data.select_dtypes(include='number')
correlation_matrix = data.corr()

# Create a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")

# Showing the heatmap on visual
plt.title('Correlation between the Variables')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.show()
```

Fig: Code to plot the Heat Map for ANOVA

REFERENCES

https://www.washingtonpost.com/graphics/investigations/police-shootings-database/

CONTRIBUTIONS

We both contributed equally in the project.