**GROUP PROJECT-1**

**Project Report:** Cleaning and Analyzing Crime Data

**Course:** IE6400 Foundations Data Analytics Engineering

Fall 2024

**Group Number – 15**

Sainath Boreda (002305951)

Nishchay Linge Gowda (002309438)

Yaswanth Kumar Reddy (002415723)

# Introduction

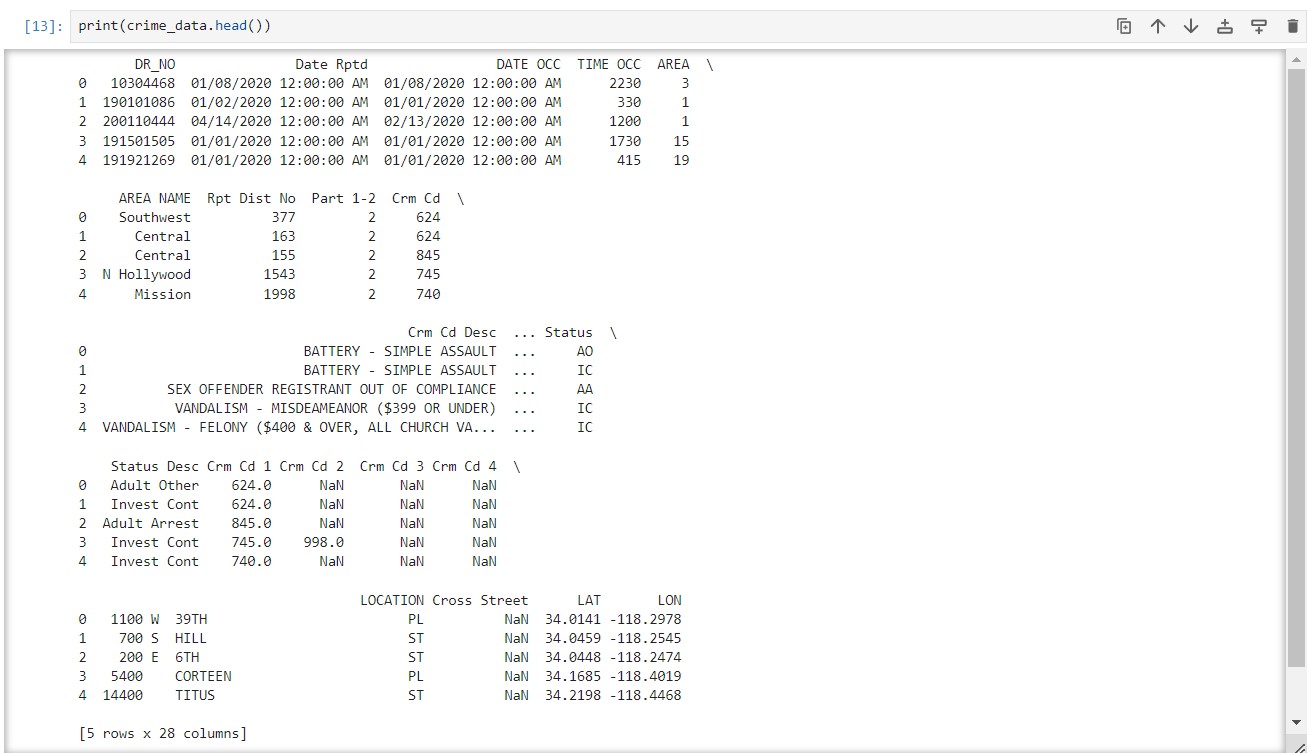
The objective of this project is to analyze the given crime dataset to discern patterns, correlations, and trends in criminal activity from 2020 to the present day. The analysis involved extensive data cleaning, exploratory analysis, and application of statistical techniques. This report documents the process and findings from the project aimed to cleaning, analyzing, and understanding crime trends from 2020 to the present day. The insights into overall crime trends, seasonal patterns, common crime types, regional differences, correlations with economic factors, and day-of-the-week effects on crime frequency was provided with the analysis . Additionally, predictive modeling was attempted for future crime trends and also the impact of major events on crime rates was assessed.

# Task 1: Data Acquisition and Data Inspection

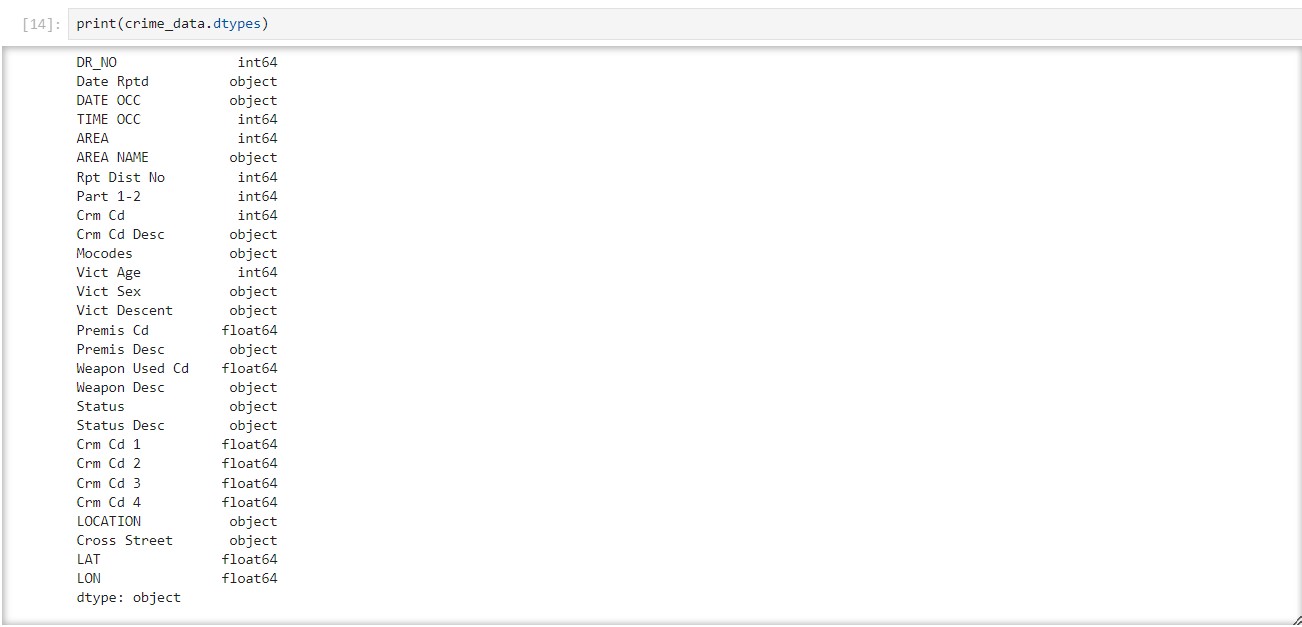
The dataset was retrieved from the “Crime Data from 2020 to Present” repository into a python environment using the python library.

Initial Observations:

First few rows displayed: It provides an overview of the various columns and the nature of the data contained in the dataset.



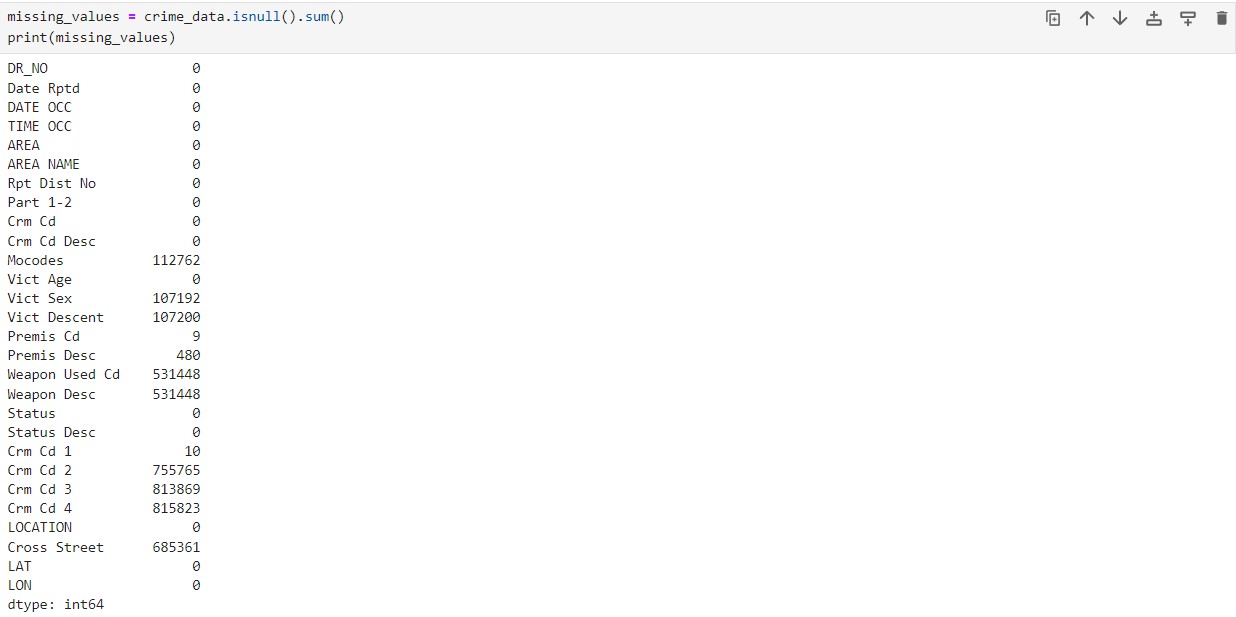
Data types checked: For the compatibility with analysis methods, each column's data type was reviewed.



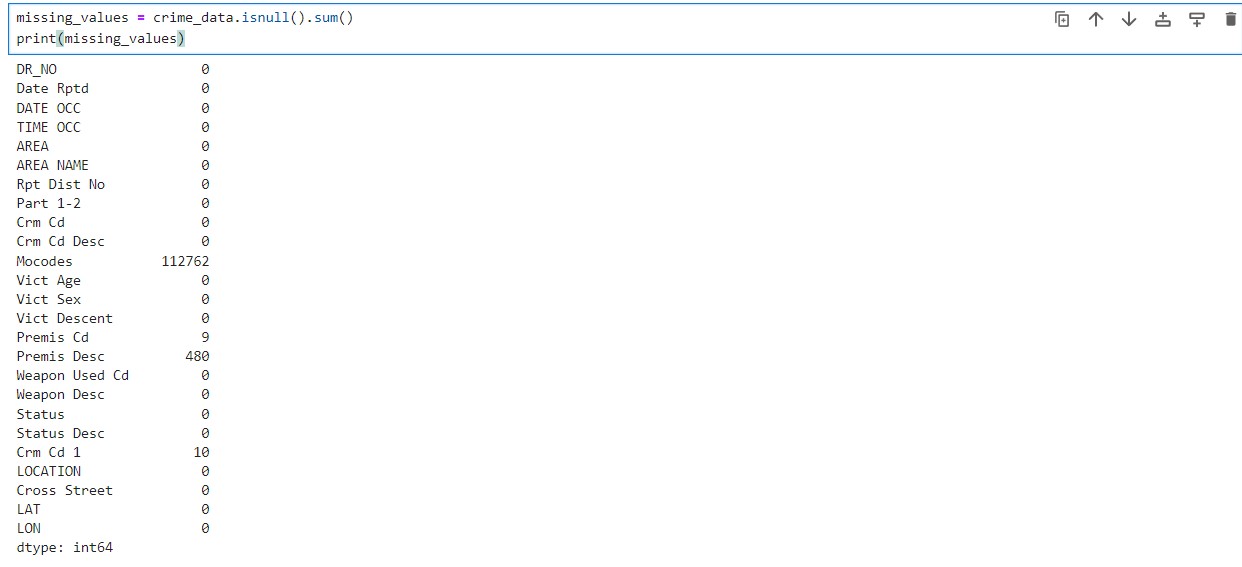
Column Review: The dataset included typical crime data columns related to weapon details, location, victim details, report details.

# Task 2: Data Cleaning

Checked the number of missing values of each column to address the missing data before analysis.

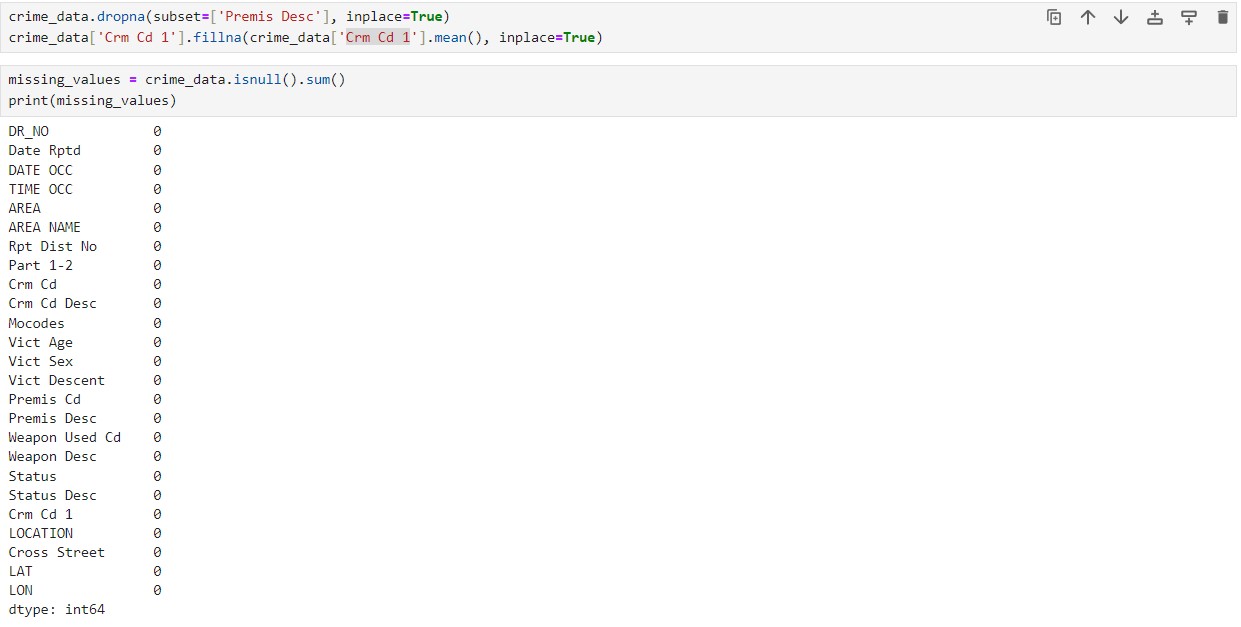


The columns 'crm cd 2', 'crm cd 3', and 'crm cd 4' were removed because they had a significant number of missing values. In columns 'Vict Sex', 'Vict Descent', 'Cross Street', 'Weapon Desc', and 'Weapon Used Cd', missing values were replaced with "Unknown", given their object data type. After these modifications, the dataset was re-evaluated for any outstanding missing values.



In the dataset, columns such as "Mocodes" and "Premis Desc" have a relatively small number of rows with null values compared to their total row count. For the "Crm Cd 1" column, null values were addressed by replacing them with the column's mean value.





**Task 3: Exploratory Data Analysis (EDA)**

**Visualization of overall crime trends from 2020 to the present year:**



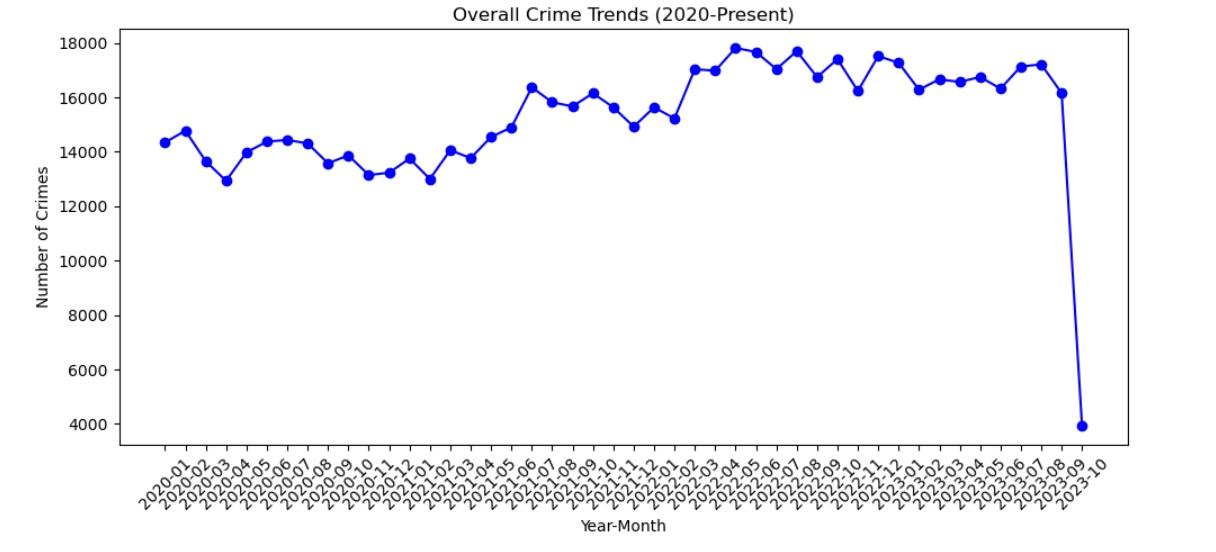
Tools like pandas and matplotlib.pyplot, are imported which are necessary and

It changes the "Date Rptd" field in the crime\_data DataFrame to a datetime format to get the information of year and month later. Also the two new columns are added to the DataFrame called "Year" and "Month." These columns are extracted from the "Date Rptd" column.

The condition crime\_data['Year'] >= 2020 is used to limit the data so that only records from 2020 and later are included. The result is saved in a new DataFrame called crime\_data\_filtered.

The saved data is then put into groups by "Year" and "Month," and the number of records in each group is counted. This information is then saved in a new DataFrame called crime\_counts. After this, it sets up a Matplotlib figure with a certain figure size.

The 'Year-Month' is used as the x-axis, and the 'Number of Crimes' is used as the y-axis. The "Year" and "Month" fields are joined together to make a single string in the "YYYY-MM" format. This is what the x-axis data looks like. For the line, the marker, linestyle, and color properties are used. Also, It sets the title, label, and x-axis label for the plot. Then, It rotates the x-axis tick by 45 degrees for the better readability. It makes sure that the plan is tightly laid out so that nothing gets cut off. Lastly, plt.show() is used to show the plot.



The chart shows how crime has changed over time from 2020 to now. It shows how many crimes happen each month and how crime rates change over time. However, there is a clear and obvious drop in the most recent month shown. According to this, there was a big event or change that caused a big drop in crime.



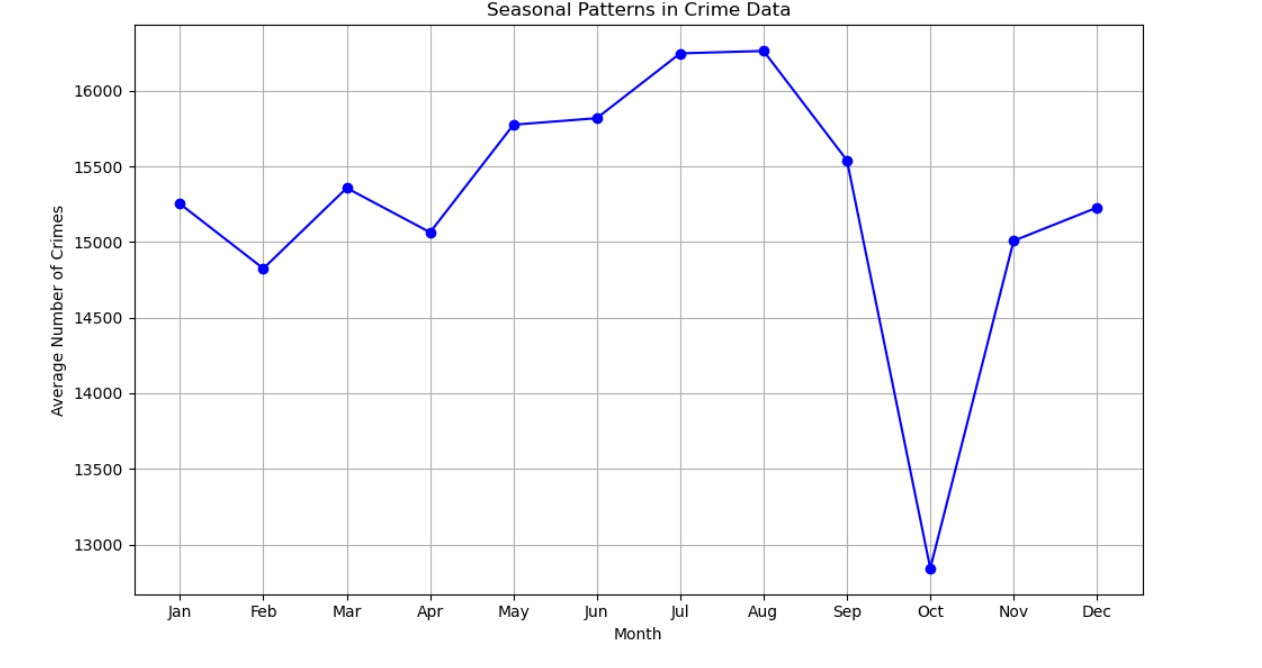
The code changes a "crime\_data" DataFrame by getting the month and year from the "Date Rptd" column. It then groups the data to find the average monthly crime count for each month across all years. The last thing that is written to the console is a list of the average number of crimes each month.

The numbers represent the average monthly crime counts for each respective month. We can interpret this result to understand how crime counts vary throughout the year.

**Analysis and visualization of seasonal patterns in crime data:**

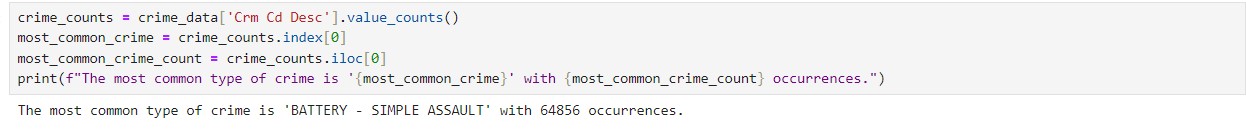


A board of a certain size is set up for the plot.The code uses months (1–12) on the x-axis and the average crime rates for each month on the y-axis to make a graph. There are circles that show where the data points are, and a steady blue line that ties them.The month (x-axis) and average number of crimes (y-axis) are given labels, and the plot is given a name ("Seasonal Patterns in Crime Data").The names on the x-axis can be changed to show month abbreviations, such as Jan, Feb, Mar, etc.The layout is changed to make sure that everything fits right.The finished plan is shown.



The graph shows the seasonal trends in crime data. The average amount of events is shown on the y-axis, and each month of the year is shown on the x-axis. From January to June, the number of crimes changes in a pretty stable way. There is, however, a clear peak in August, followed by a sharp drop in September. Then, in November and December, near the end of the year, the numbers go up again. This information points to a possible pattern or cause that might be causing crime rates in these particular months.

**Identifying the Most Common Type of Crime:**

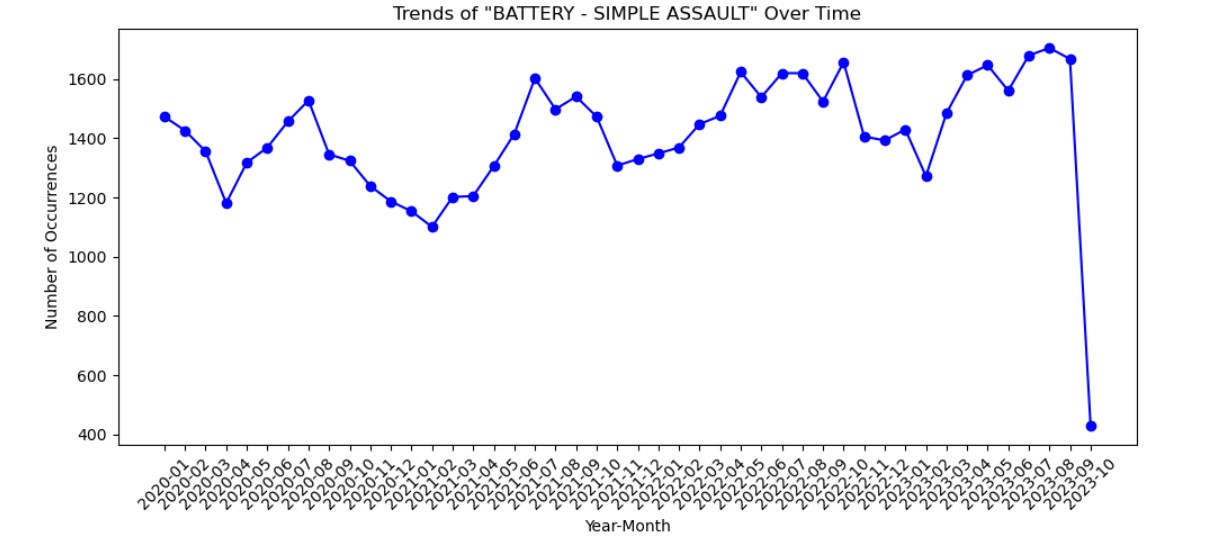


In the 'crime\_data' DataFrame's 'Crm Cd Desc' field, it keeps track of how many times each type of crime happens.It lists the most popular type of crime and how many of them there are.It writes a message that lists the most common crime and how many of them there are.To put it more simply, the code finds and records the type of crime that happens most often in the information.

**Analyzing Trends of the Most Common Crime Over Time:**



Only the rows with the most common type of crime ('most\_common\_crime') are chosen from the 'crime\_data' DataFrame.It sorts this cleaned data into groups by "Year" and "Month" to see how many times the most common crime happened each month.It makes a line plot so you can see how the most common crimes have changed over time. The year and month are shown on the x-axis, and the amount of the most common crime is shown on the y-axis. The code separates and shows the most common type of crime that happens each month over time. This lets to see how this type of crime has changed.



The graph shows the patterns of events marked as "BATTERY - SIMPLE ASSAULT" over time. The y-axis shows the number of times something happened, and the x-axis shows the passage of time, broken down by year and month. In that time period, there are clear changes in the number of events, with several peaks and valleys. Notably, there is a big drop toward the most recent data point, which shows that the number of recorded cases of "BATTERY - SIMPLE ASSAULT" has gone down a lot. There needs to be more research into this sudden drop to find out what happened or what causes may have caused it.

**Grouping Data by regions:**



The code summarizes the number of crimes in each geographical area, providing a count of crimes for each area in a new DataFrame.

**Comparing Crime Rates:**



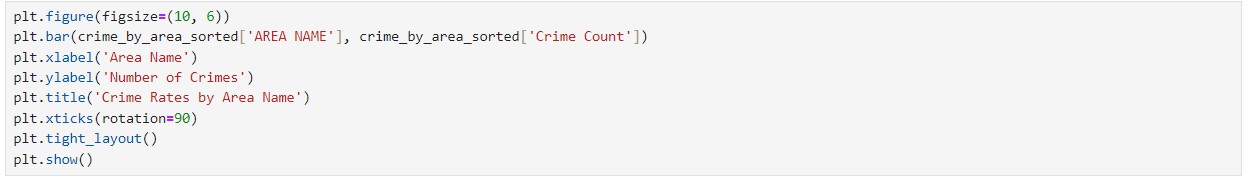
It sorts the "crime\_by\_area" DataFrame, which has the number of crimes in different parts of the world, by the "Crime Count" column in decreasing order ("ascending=False").It stores the sorted list of crimes in a new DataFrame called "crime\_by\_area\_sorted."The last thing it does is print the

"crime\_by\_area\_sorted" DataFrame. This shows the places with the highest crime rates at the top and the lowest crime rates at the bottom.To sum up, this code sorts and writes the places based on how many crimes they have, from most crimes to least crimes. It gives you a list of places ranked by how often crimes happen there.



Each row represents an area, and the 'Crime Count' column shows the number of reported crimes in that area. This output allows us to quickly identify and compare the areas with the highest and lowest crime rates based on the data in 'crime\_data' DataFrame. It provides valuable information for analyzing crime patterns across different geographical regions.

**Visualizing the Differences:**



This code makes a bar chart with these important parts:

It tells the chart what size fabric to use.

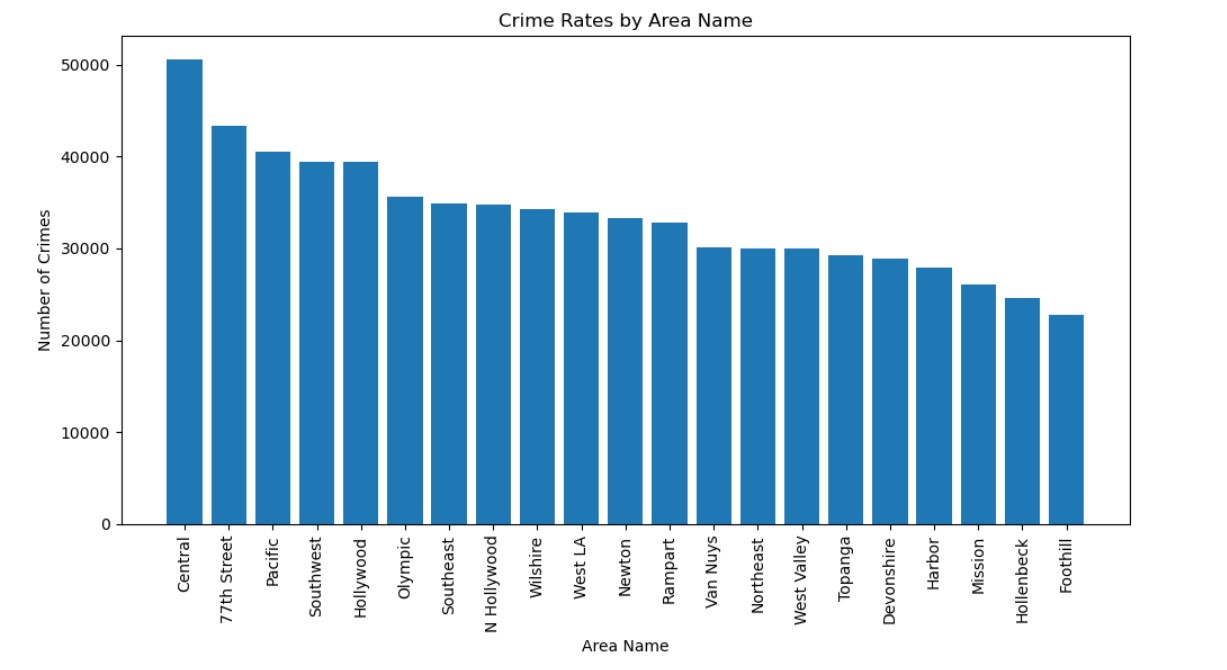
Areas are on the x-axis and crime rates are on the y-axis.

The map shows how many crimes happen in each place.

On the x-axis, "Area Name," and on the y-axis, "Number of Crimes," labels are added.It is called "Crime Rates by Area Name" on the map.

The names on the x-axis have been turned to make them easier to read. For a clean look, the layout has been changed.

Finally, the chart is shown.



The graph shows crime numbers broken down by the names of different areas. The area names are shown on the x-axis, and the number of events is shown on the y-axis.

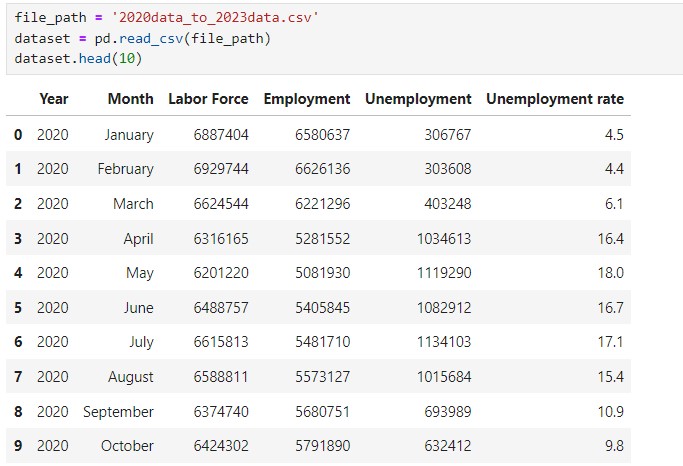
It's clear from the picture that:

The most crimes have been recorded in the "Central" area—nearly 50,000. There are also a lot of crimes in places like "Pacific," "Southwest," "Hollywood," and "77th Street." Each of these places has over 30,000 recorded crimes.

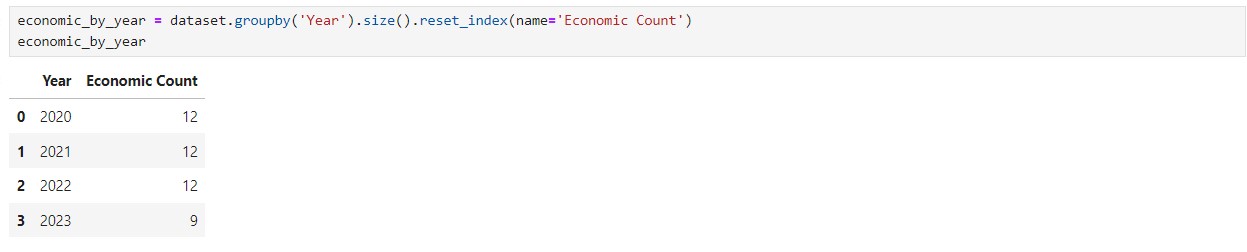
Graph: The number of crimes starts to go down as we move to the right. But the drop is slow, and the crime rate in each area is about the same as the crime rate in the places next to it. The crime rate in "Foothill" is the lowest of the places shown, but it's still over 10,000, which is pretty high.

This information can be used to look at patterns in crime, make sure that police resources are used wisely, or put community outreach programs at the top of the list in places with higher crime rates.

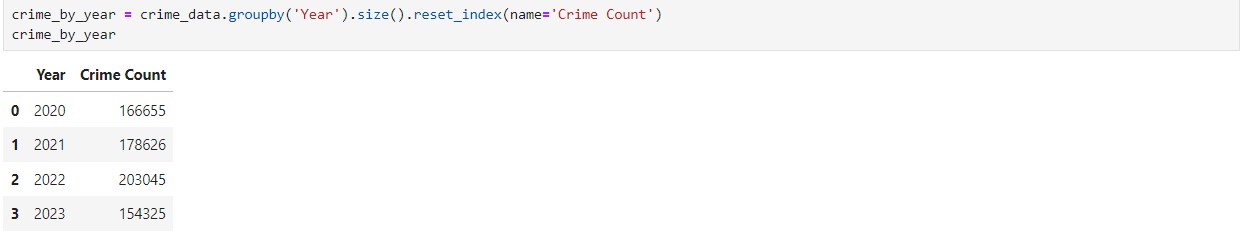
**Explored correlations between economic factors and crime rates:**



The table shows information about the work force for 2020. From March on, it shows that the jobless rate went up, reaching a high point of 18% in May. Because of the COVID-19 pandemic, there may be a link between this rise and the economy. The rate went down to 9.8% in October, showing some improvement, but it was still higher than it was in early 2020.



The number of economic data points for each year is shown in the table. There are 12 data points for 2020, 2021, and 2022. There are 9 data points for 2023.



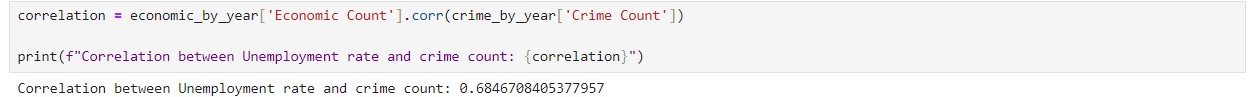
The table displays the total crime counts for each year. Specifically:

In 2020, there were 166,655 crimes.

In 2021, there were 178,626 crimes.

In 2022, there were 203,045 crimes.

In 2023, there were 154,325 crimes.



The data givesThere is a 0.685 link between the unemployment rate and the number of crimes. This shows a moderately positive connection, which means that as jobless rates rise, crime rates may also rise.

**Extracting the Day of the Week:**



The 'Date Rptd' field in the crime\_data DataFrame is being changed to a datetime style by this code. First, it makes a new column called "Day of Week." For each date in the "Date Rptd" column, it gives the day of the week as a number from 0 to 6, with 0 being Monday and 6 being Sunday. Basically, it takes the date of each reported crime and pulls out the day of the week.

**Grouping the Data and Analyzing:**



This piece of code groups the crime\_data DataFrame by the "Day of Week" and "Crm Cd Desc" fields, which most likely mean "Crime Code Description." It's keeping track of how many times each day of the week and type of crime showed up

As a result, crime\_type\_counts 'Crime Count' column in DataFrame will show how many times that type of crime was reported on that day of the week. Each row will have a different day and crime type mix. This can help you look at how crime changes from day to day.

**Visualizing the relationship between the day of the week and the frequency of certain types of crimes:**



code creates a bar chart to show the average crime frequency for each day of the week.

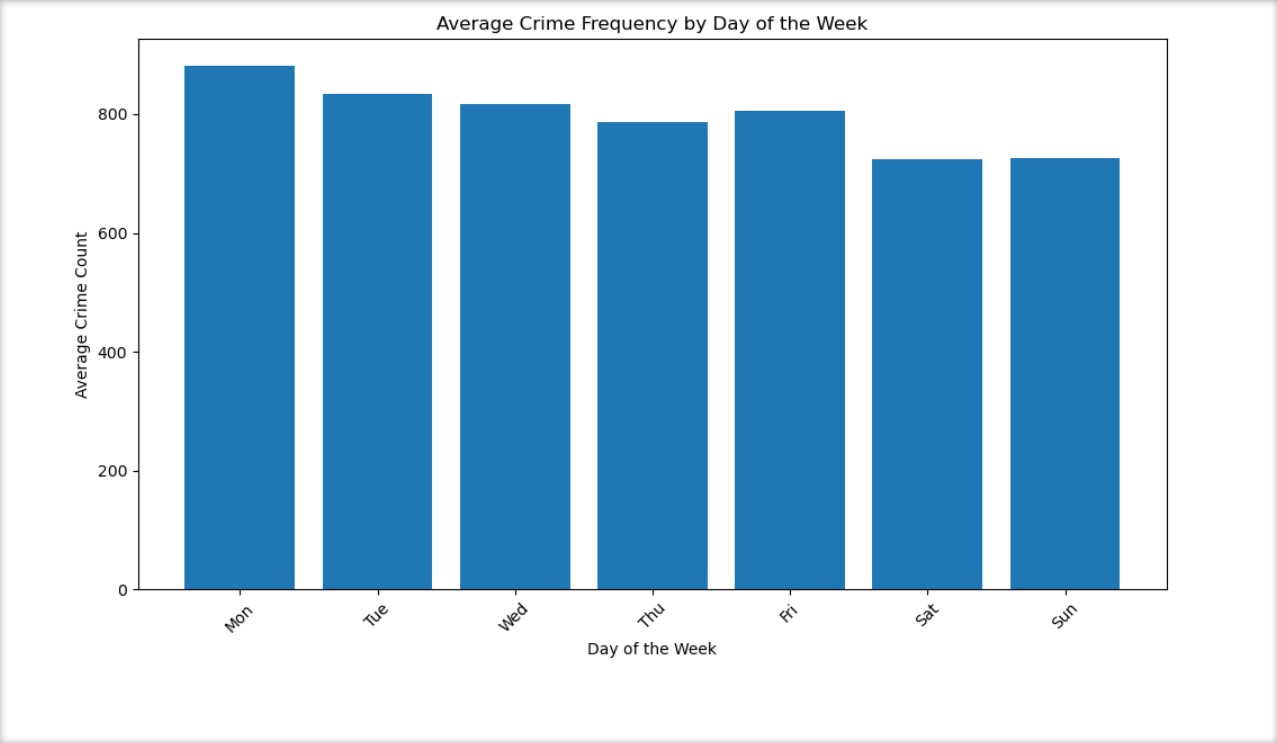
It calculates the average crime frequency for each day of the week and Sets up a canvas for the chart.

Defines the x-coordinates for the bars.

Creates the bar chart, with days of the week on the x-axis and average crime counts on the y-axis.

Labels the x-axis as "Day of the Week" and the y-axis as "Average Crime Count." Adds a title to the chart.

Rotates the x-axis labels for better readability and Ensures the layout is neat and readable.



The bar chart shows how often crimes happen on average each day of the week. Monday has just over

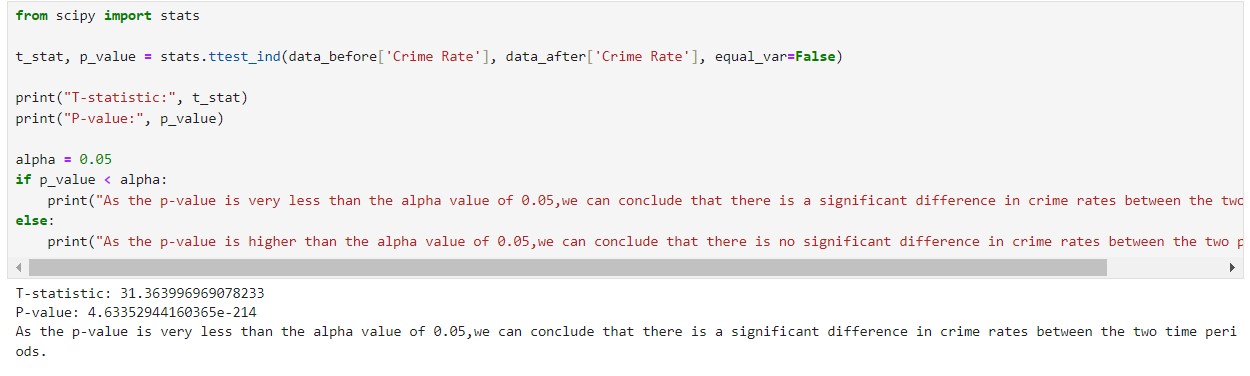
800 crimes, which is the most of any day. Between 700 and 750 people are counted every day from Tuesday to Saturday. On Sunday, there is a small drop. In general, crime rates stay pretty fixed during the week, with only small changes.

**Investigate any impact of major events or policy changes on crime rates:**



The George Floyd protests in Los Angeles began on May 27, 2020 and ended around June 13, 2020. Based on this information, performing a t-test would be suitable to find if there are any significant changes in crime rates before and after these events lasted, With the consideration of a time period of two months before and after.

The crime\_data DataFrame is split into two parts by the code: "before" and "after" certain times. Then, the average monthly crime rate for both times is found by adding up all the crimes that were reported every day and comparing them to a 30-day month.

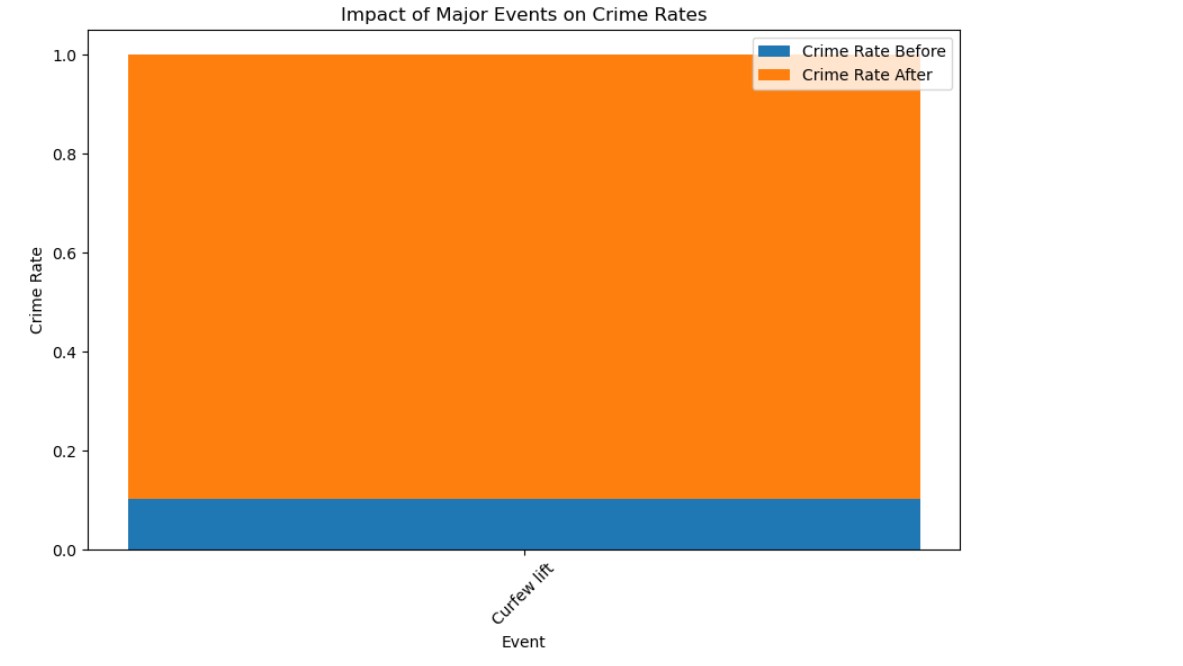


The visualization also can be done to the change in crime rates before and after a certain event using matplotlib.

For instance, consider another event when the curfew imposed by the LAPD was lifted on June 1, 2020



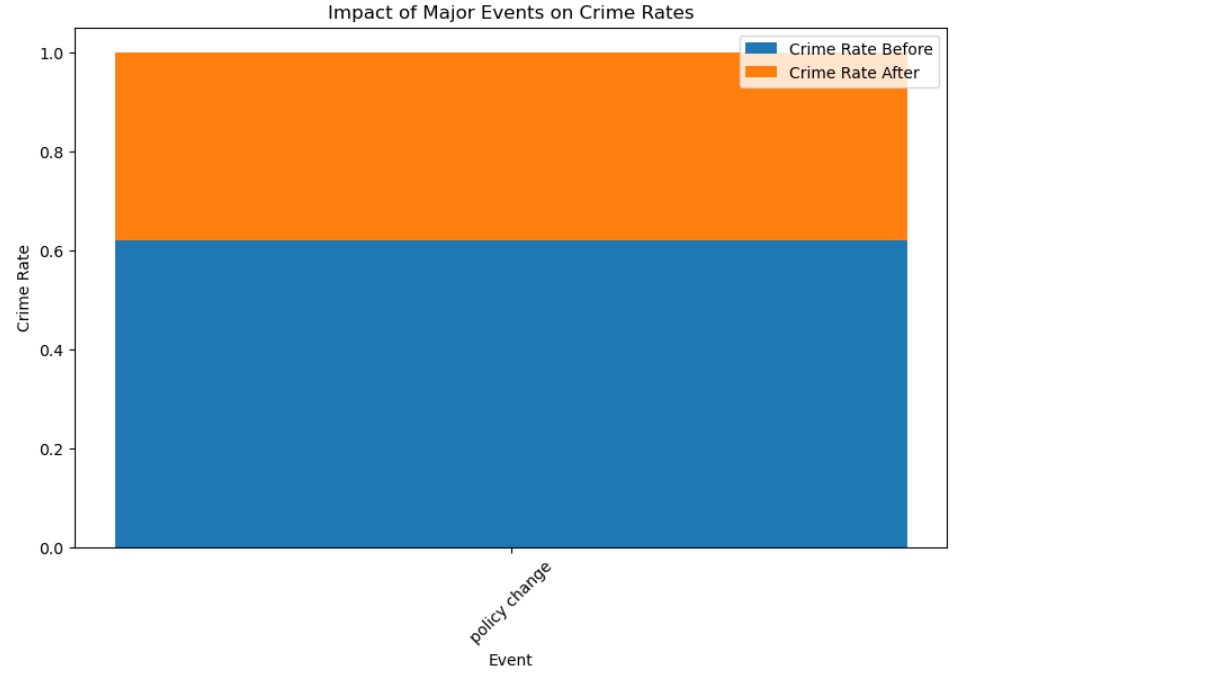
The code gets crime data from a CSV file and looks at how the "Curfew lift" event changed crime rates. The data is split into times before and after the event. Crime rates are then calculated for each time period and shown in a stacked bar chart. On the X-axis is the event, and on the Y-axis is the crime rate. Bars show the difference between the crime rate before and after the event.



The graph shows how the "Curfew lift" event changed the crime rate. Most of the crimes (shown in orange) happened after the event. A smaller number of crimes (shown in blue) happened before the curfew was lifted.



A "policy change" has an effect on crime rates, and the code shows this by splitting the data into times before and after the event and then showing the results as a stacked bar chart.



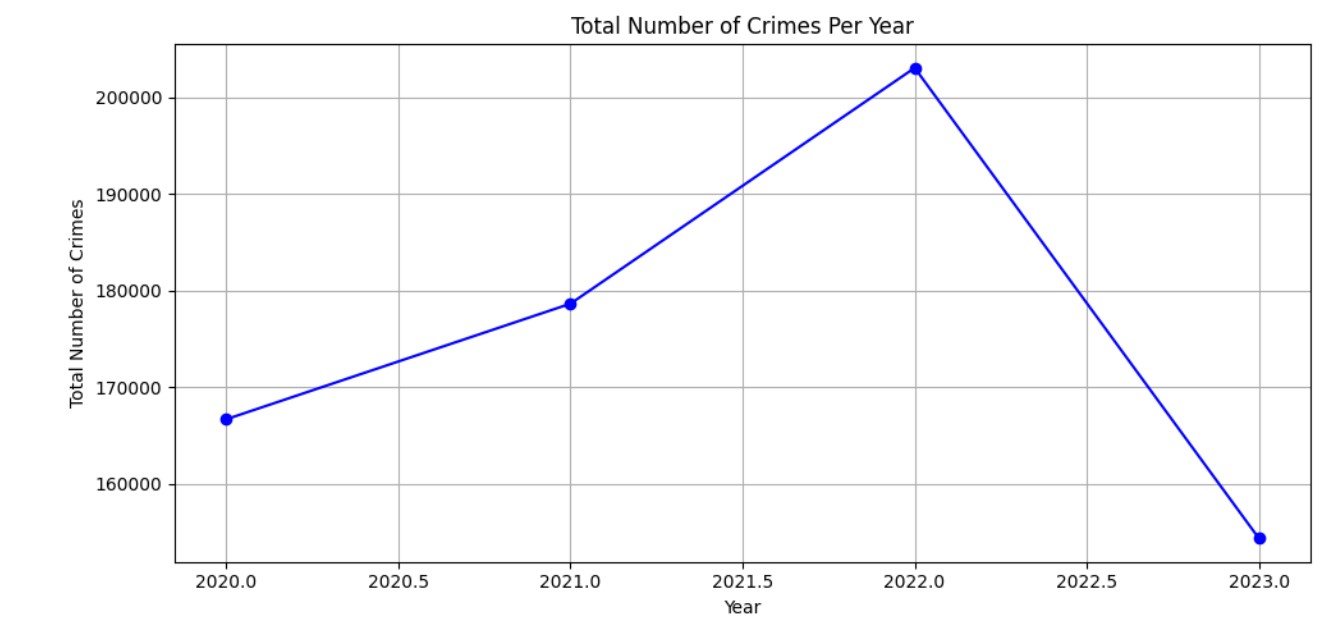
A "policy change" has an effect on crime numbers, as shown in the chart. The orange line shows the crime rate after the change, and the blue line shows the crime rate before the change. Based on the numbers, crime rates stayed pretty much the same before and after the event.

# Questions

**1. Overall Crime Trends:**



The above code processes a dataset containing crime reports. First, it converts the ‘Date Rptd’ column to a date format and extracts the ‘Year’. Then, it calculates the total number of crimes reported per year. Using this data, it creates a line graph where the x-axis represents years, the y-axis shows the total number of crimes reported, and each point on the line indicates the crime count for a specific year. The plot provides a visual overview of how the number of crimes changes each year, allowing easy comparison of crime rates over time.



The graph shows the total number of crimes reported each year from 2020 to 2023. The observations are as follows:

2020: About 1,700,000 crimes were reported.

2021: Crime numbers increased significantly, reaching almost 1,800,000.

2022: Crime rates remained high, peaking above 2,000,000, but started decreasing later in the year.

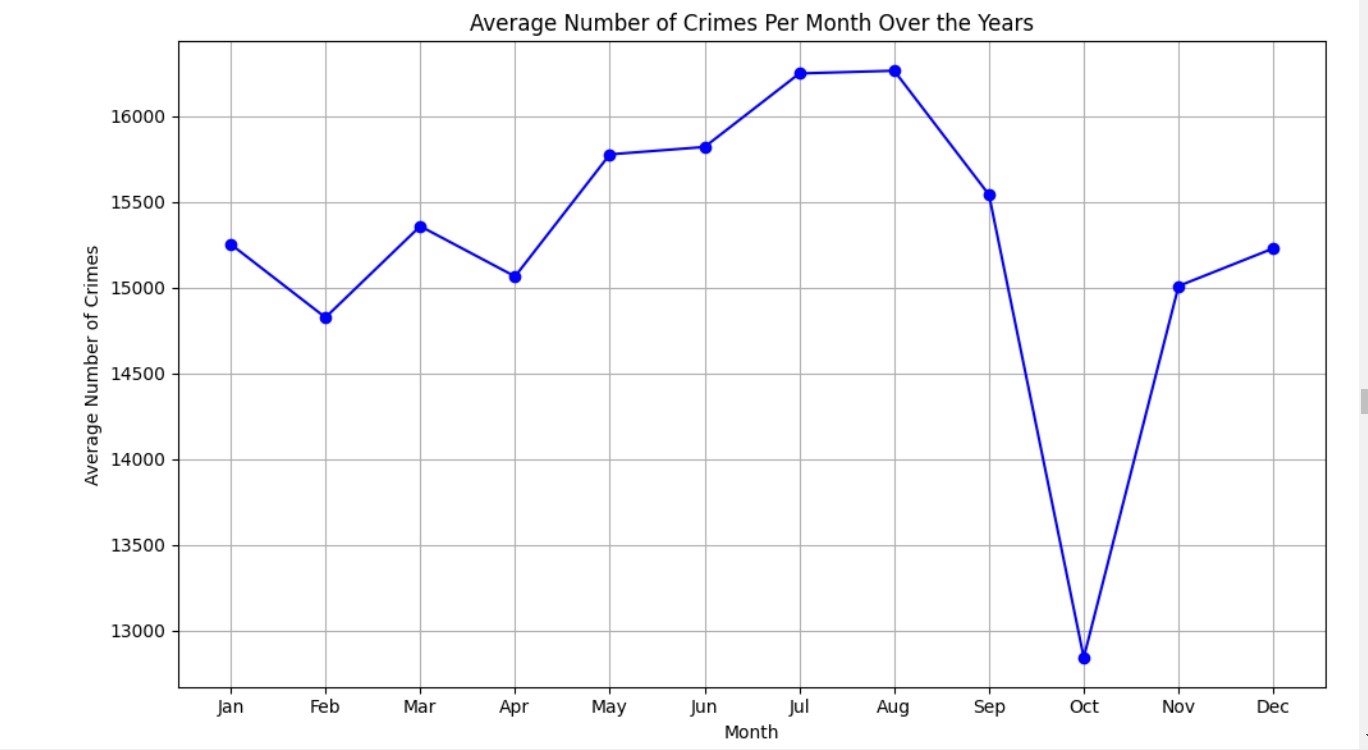
2023: There was a sharp decrease, bringing crime numbers back to around 1,600,000, similar to 2020 levels.

In simple terms, crime rates went up in 2021 and early 2022 but dropped significantly by 2023, returning to the levels seen in 2020.

1. **Seasonal Patterns:**



The above code looks at crime data, and organizes it by month and year. After then, a line graph with these averages for every month (January to December) is created. The average number of crimes is displayed on the y-axis, and months represented on the x-axis. The graph makes it simple to compare annual crime patterns and shows the months with the highest or lowest average crime rates.

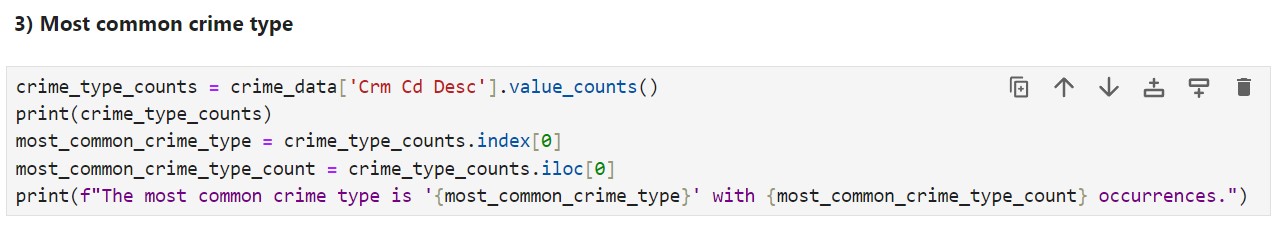


The graph shows the average number of crimes each month:

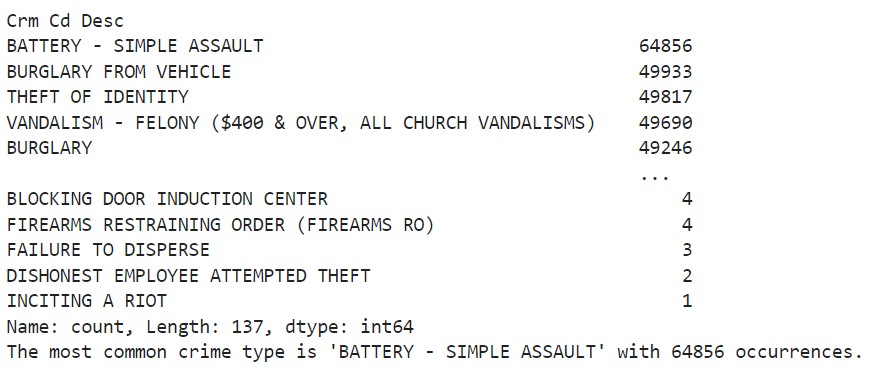
* + January: Around 15,000 crimes start the year.
  + February to April: There’s a slight difference in the crime rates, peaking in March at over 15,500, then slightly dropping in April.
  + May to August: We can see an increase in the crime rates in this time period. Where July and August have around 16,250 crimes.
  + September: There’s a decrease in the crime rates from 16,250 to 15,500 in this month.
  + October: A big drop, with crimes falling to just below 13,000.
  + November to December: Crimes rise again, reaching similar to the beginning of the year.

In short**,** crime stays steady most of the year, dips in September and October, then rises sharply towards the year’s end.

1. **Most Common Crime Type:**



The code counts the occurrences of each crime type and checks which type of crime appears most often in the data and tells us its name and how many times it occurs.

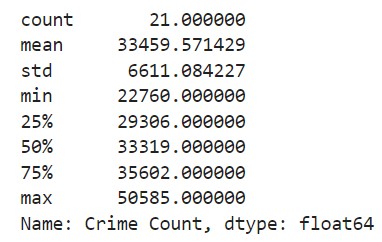


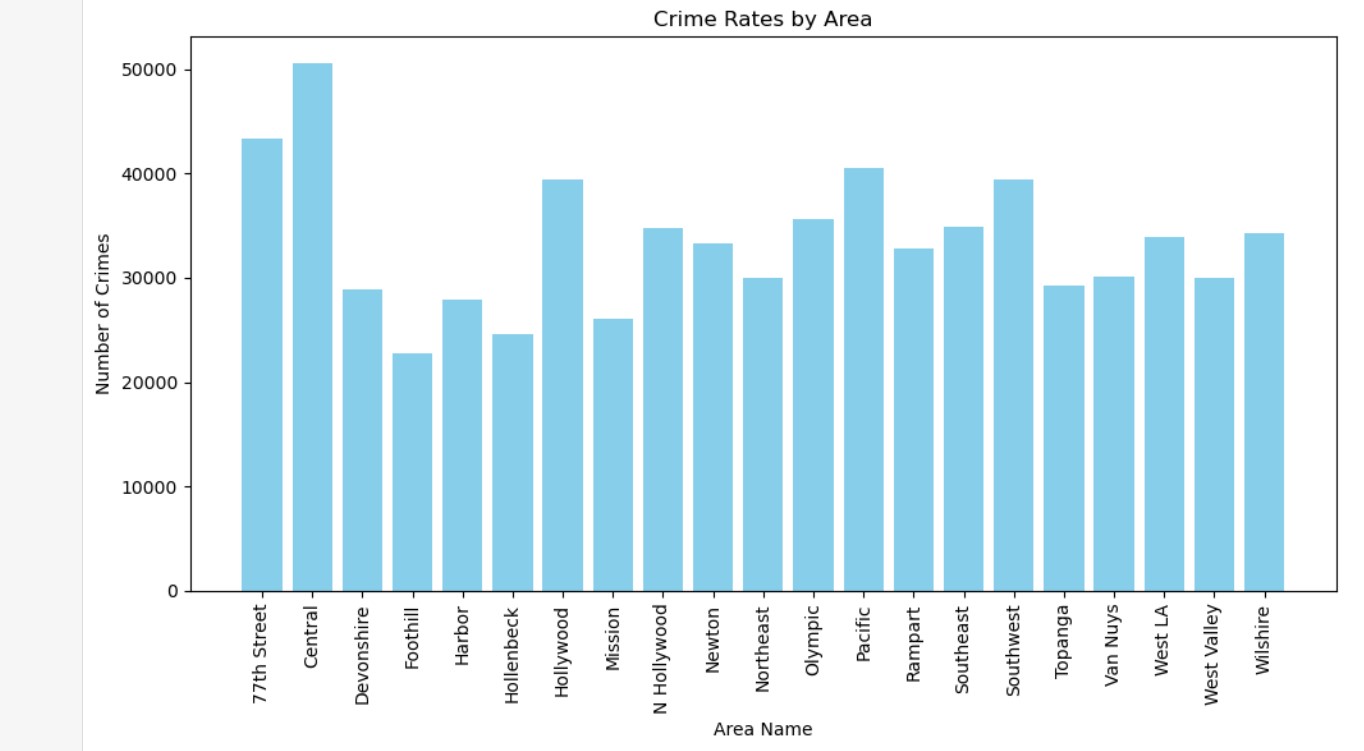
So the output given is the crime type and it’s count of occurrences where BATTERY - SIMPLE ASSAULT has the highest count i.e., 64856 whereas INCITING A RIOT has the least count with just 1. Therefore, BATTERY - SIMPLE ASSAULT is the most common crime type with 64856 occurrences.

1. **Regional Differences:**



This code groups data according to "Area Name," which stands for various cities or regions, in order to analyze and visualize crime rates. The number of crimes in each area is first obtained and saved in a DataFrame called "crime\_by\_area." We are provided with the mean, standard deviation, minimum and maximum values of the crime count statistics.The code creates a bar chart showing the total number of crimes in various areas. The height of each bar indicates the number of recorded crimes in that location, which each bar represents. The graph makes it simple to compare crime rates between areas.





Area name and the Crime count is displayed in the first code output.

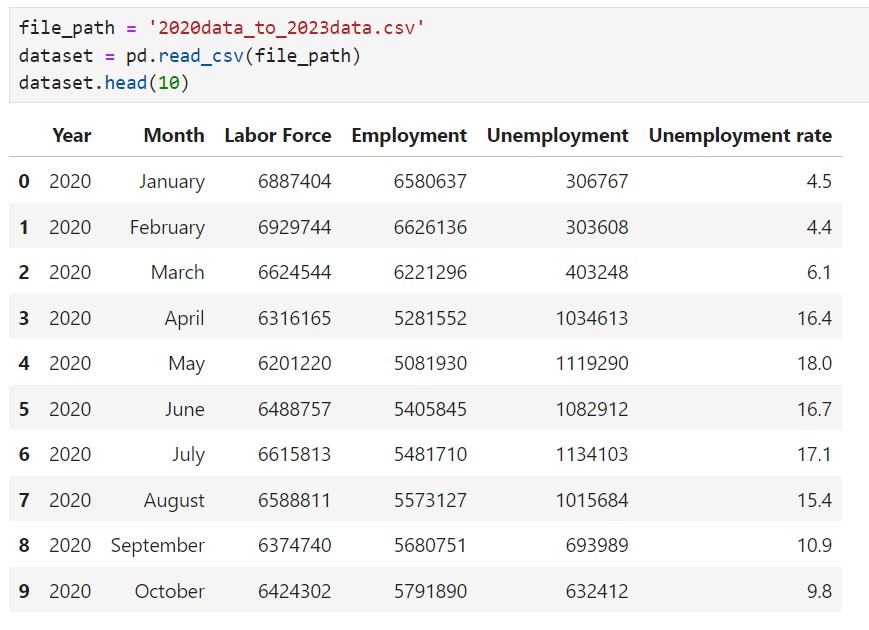
Then we get the values of mean, min, max etc in the second code output.

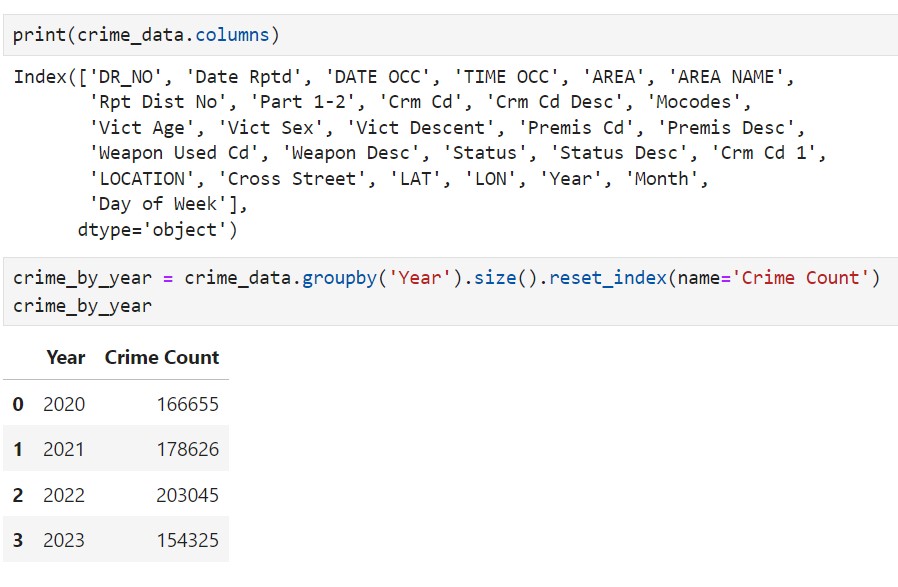
Crime rates by area are shown in the graph. With nearly 50,000 crimes, "Central" has the highest crime rate. Additionally, "Pacific," "Southwest," "Hollywood," and "77th Street" each have more than 30,000 crimes. Though it has the fewest, "Foothill" nevertheless has more than 10,000. This data helps in community interventions and the distribution of law enforcement resources.

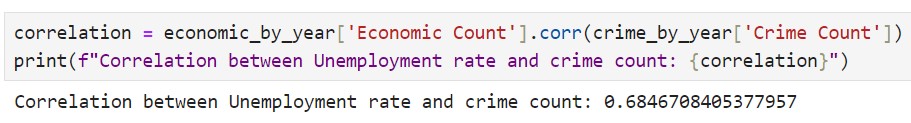
1. **Correlation with Economic Factors:**



The economic data spans the years 2020–2023. The dataset's first ten rows are being printed. The dataset is then grouped by "Year," and the index is reset to "Economic\_Count." We are printing the crime data columns. The dataset is then grouped by "Year," and the index is reset to "Crime\_Count." To determine how economic factors and crime rates are related, we are using correlation analysis.





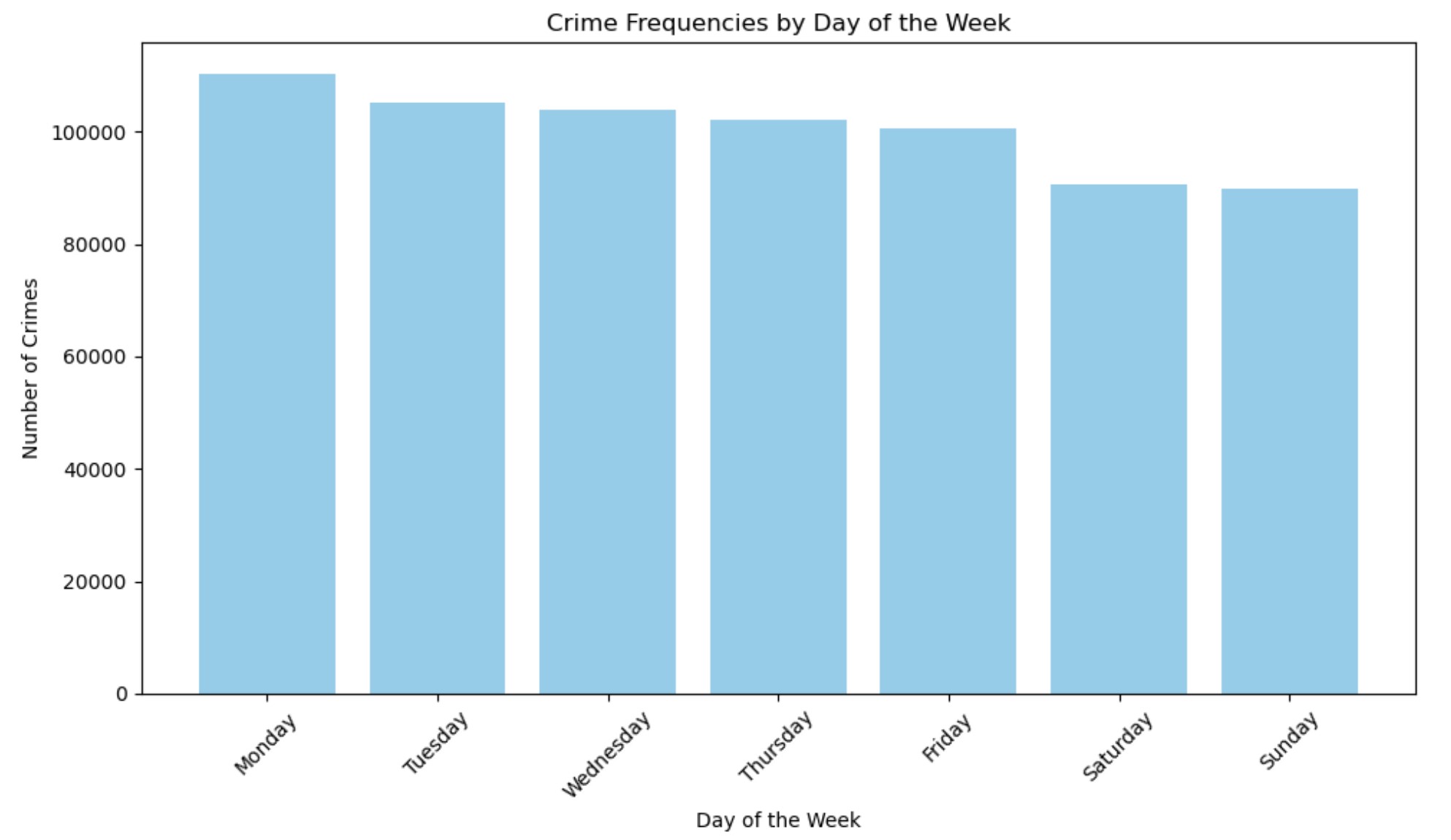


From all the output which we get, we use correlation analysis to measure the relationship between the number of economic events and number of crimes.The output value of around 0.685 indicates that there is a positive relationship between the number of crimes and the unemployment rate. Put more simply, there is a tendency for the number of crimes to rise along with the unemployment rate. Although it doesn't necessarily suggest causality, this positive connection suggests that there could be a relationship between economic issues like unemployment and crime rates.

1. **Day of the Week Analysis:**

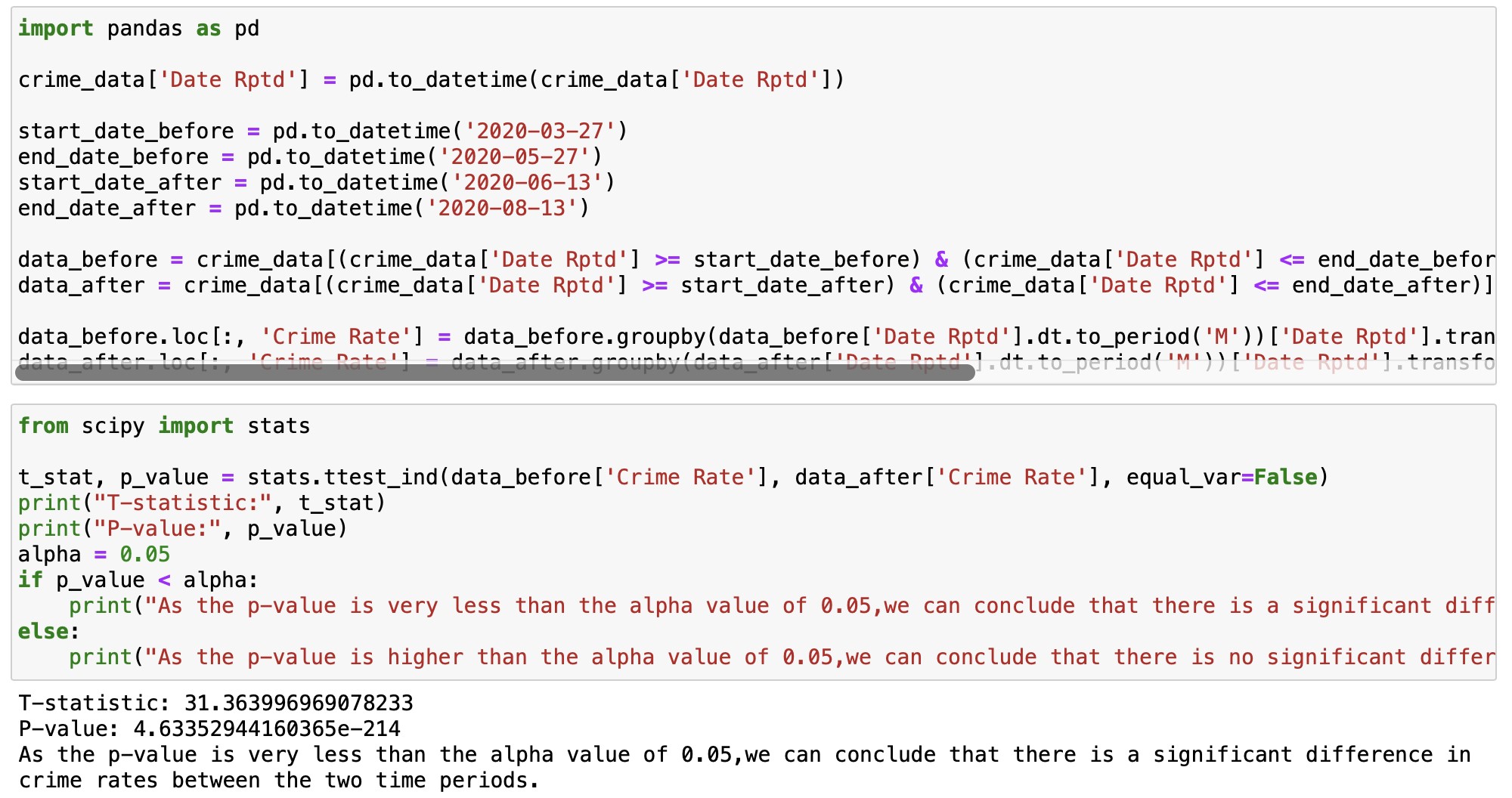


* + The code starts by converting the 'Date Rptd' column in the crime\_data\_2020\_to\_Present Data Frame into a datetime format. This is important because it allows for the extraction of various date-related information.
  + After converting the date into a datetime format, the code creates a new column called 'Day of Week' in the Data Frame. This new column stores the day of the week for each reported date. This step enables the grouping of crime data by the day of the week.
  + The code groups the crime data by the 'Day of Week' column. It calculates the number of crimes reported on each day of the week and stores this information in the Data Frame. This provides a count of crimes for each day of the week.
  + The code then proceeds to create a bar chart to visualize the frequency of crimes reported for each day of the week. Here is what the visualization does

 :

The resulting bar chart visually presents the distribution of reported crimes across the days of the week. It can help in identifying patterns or trends in crime reporting, such as whether certain days tend to have more or fewer reported incidents. This information can be valuable for law enforcement and policy analysis.

1. **Major Events:**
2. George Floyd Protests (May 27, 2020 - June 13, 2020):



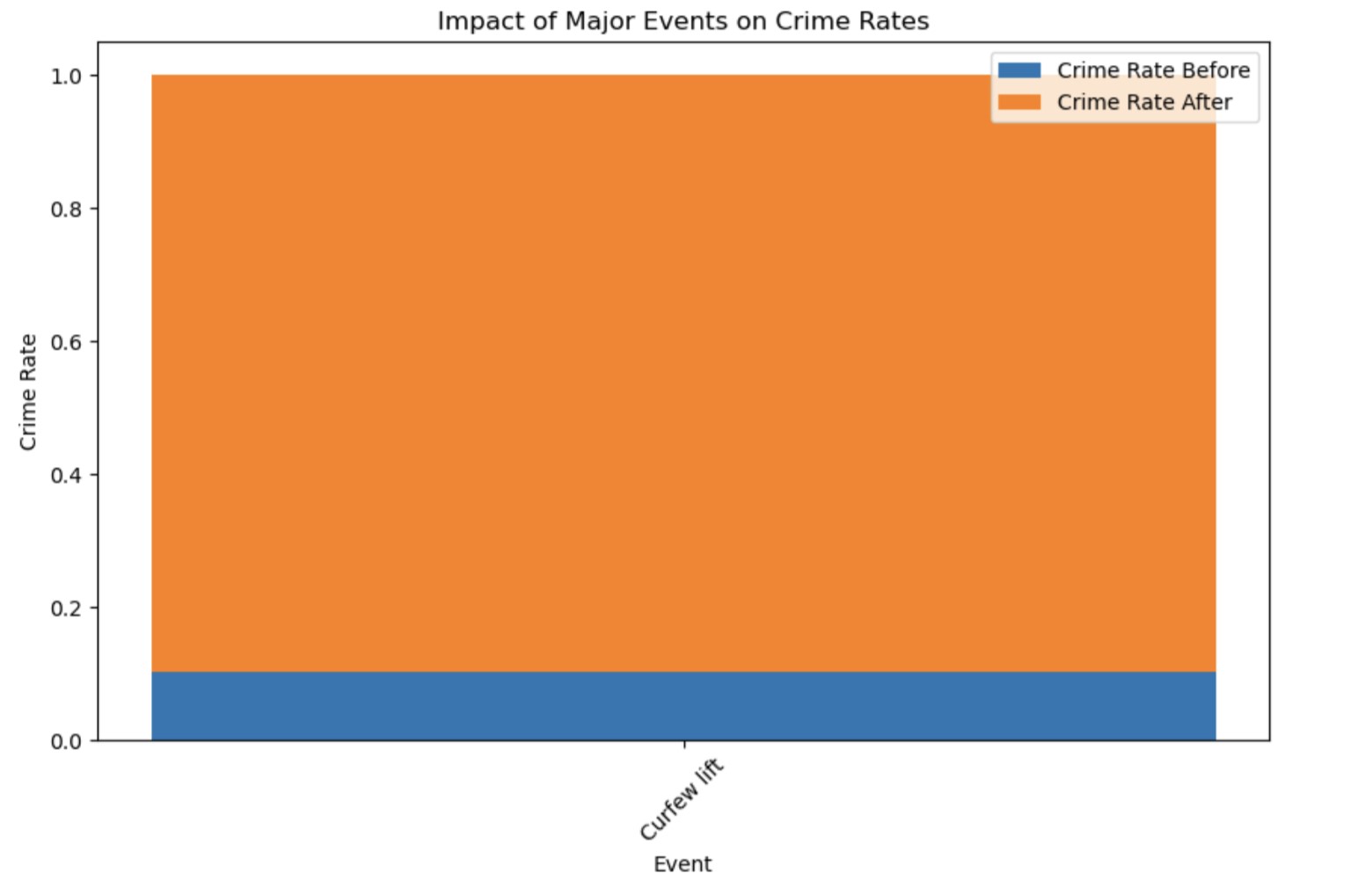
* + The provided code demonstrates an analysis of crime rates before and after specific events, using statistical tests and visualizations.
  + The code first prepares the data by dividing it into two time periods, two months before and after the protests.
  + It calculates the crime rates for both periods and conducts a t-test to assess if there's a significant difference.
  + The t-test result shows a very high t-statistic (31.36) and an extremely low p-value (4.63e-214), indicating a highly significant difference in crime rates.
  + The conclusion is that there is a significant increase in crime rates during and after the George Floyd protests.

1. Lifting of Curfew (June 1, 2020):

For this event, the code calculates crime rates before and after the curfew imposed by the LAPD was lifted on June 1, 2020.



The plot illustrates a significant increase in the crime rate after the curfew was lifted. This visualization supports the conclusion that lifting the curfew had an impact on crime rates.

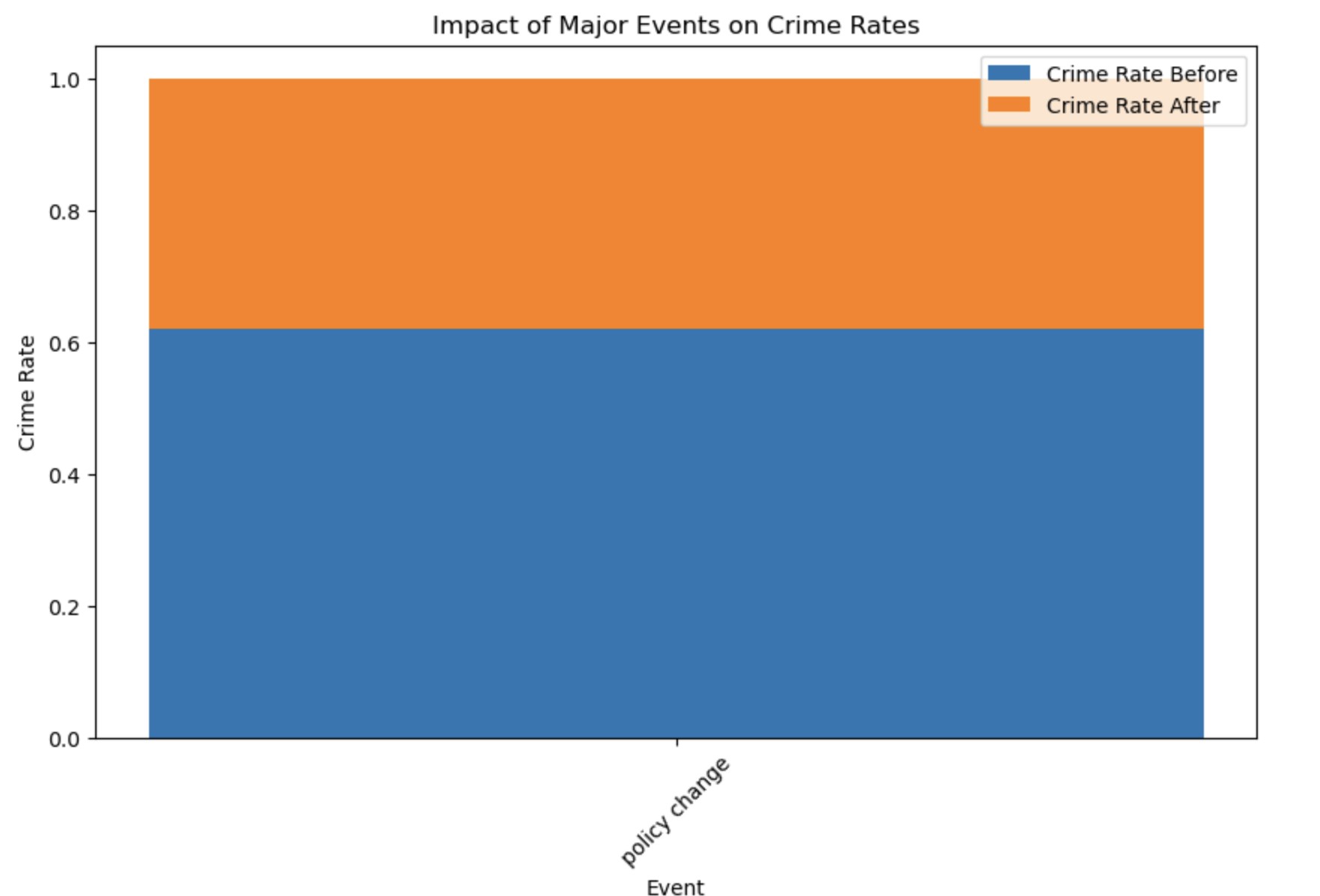


1. Policy Change by Los Angeles County District Attorney (June 2, 2022):
   * The code analyzes the impact of a policy change announced by Los Angeles County District Attorney George Gascón on June 2, 2022.
   * - It calculates crime rates before and after the policy change and produces a bar chart to visualize the results.



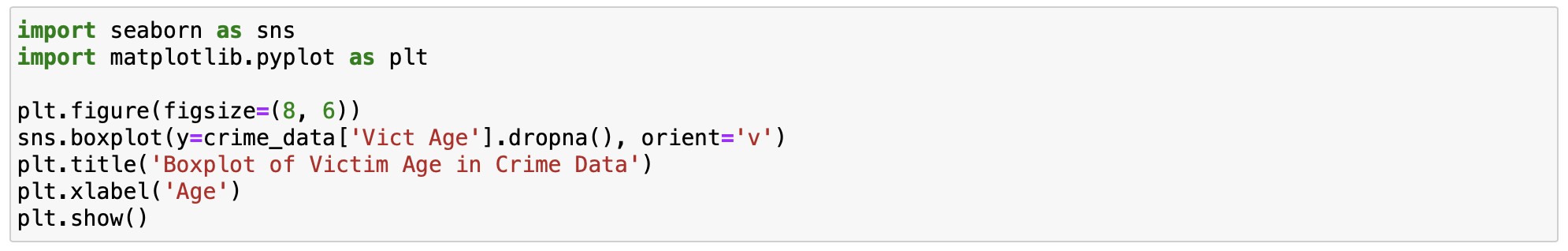
The plot shows a decrease in the crime rate after the policy change, but the decrease is not significant.

This suggests that the policy change may not have had a substantial impact on crime rates

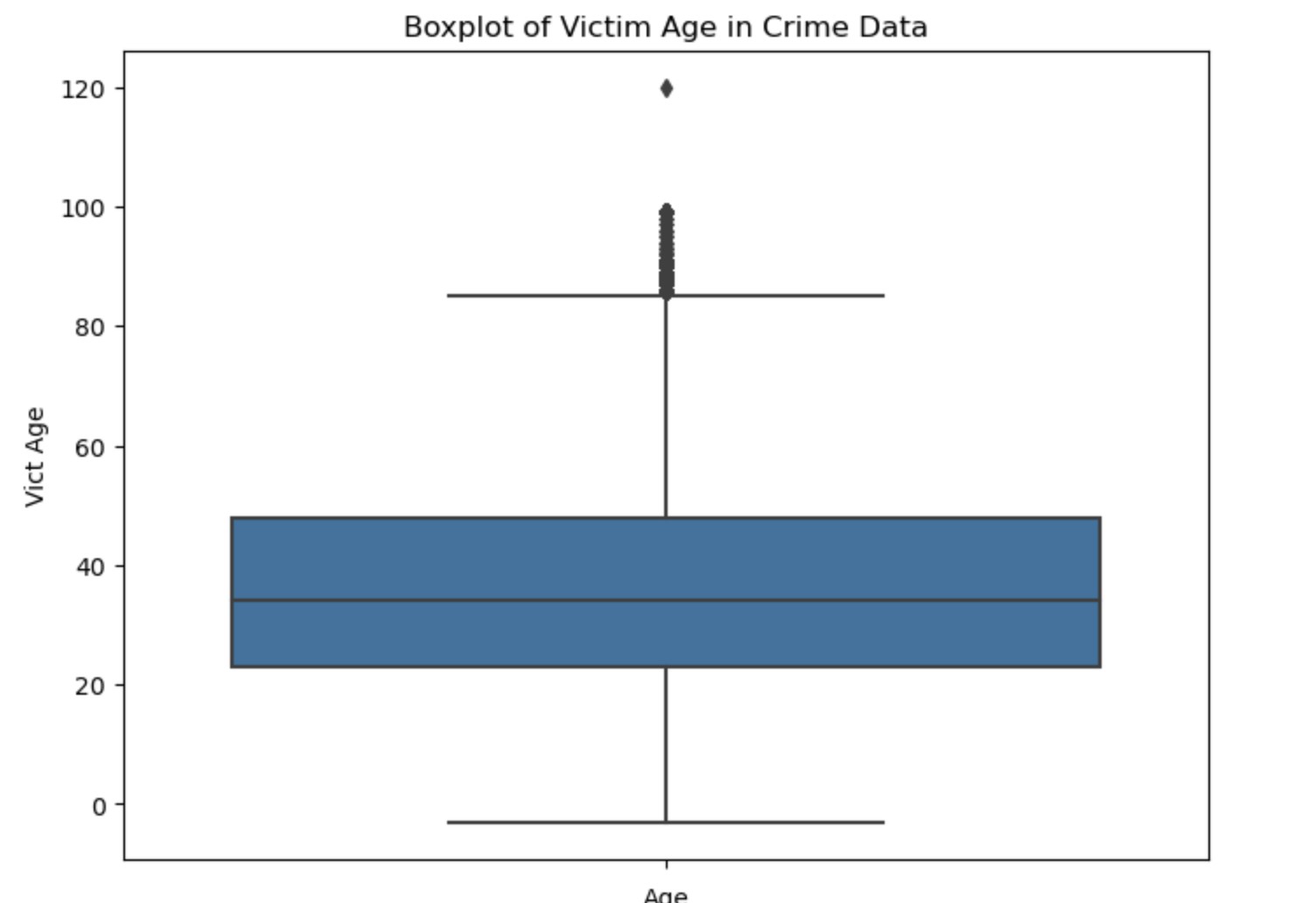


In summary, the code performs statistical tests and visualizations to assess the impact of major events on crime rates. It provides evidence that the George Floyd protests, and the lifting of the curfew had a significant impact on crime rates, while the policy change by the Los Angeles County District Attorney did not appear to have a significant effect on crime rates, resulting in a decrease that was not statistically significant. These analyses can be valuable for policymakers and law enforcement agencies to understand the relationship between events and crime rates.

1. **Outliers and Anomalies:**

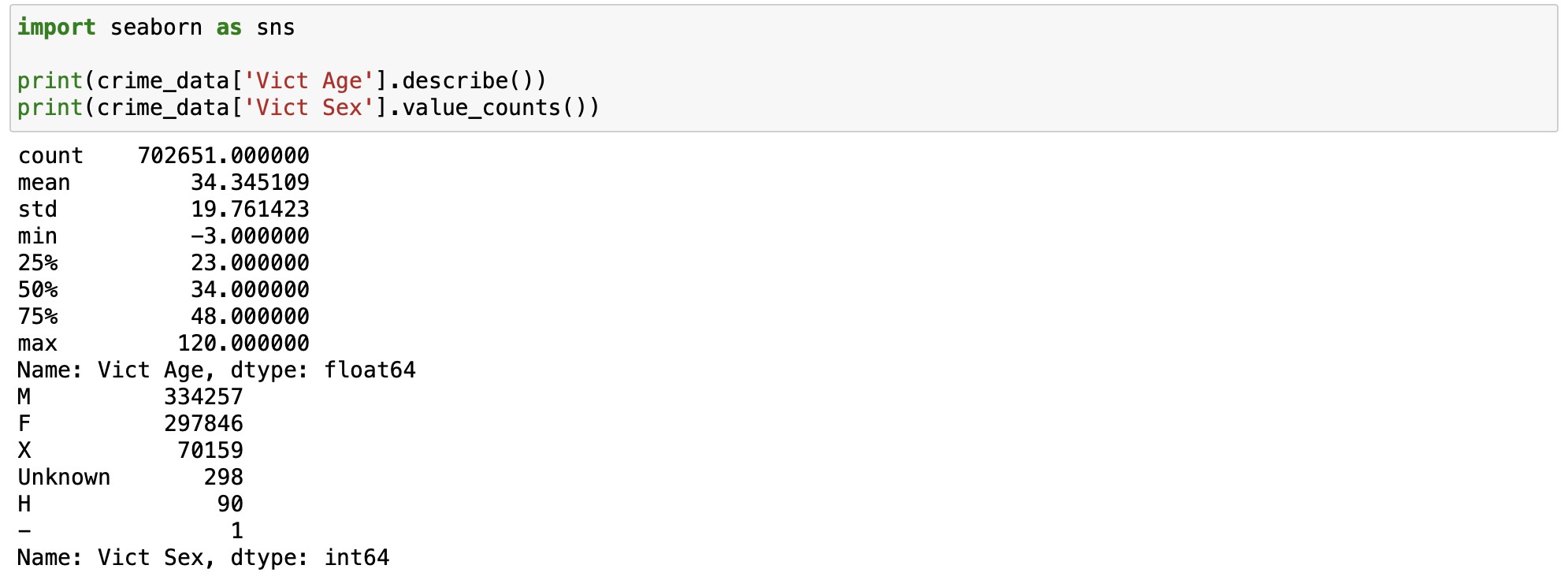


* + The code demonstrates how to use Seaborn and Matplotlib in Python to create and display a boxplot. This boxplot visualizes the distribution of victim ages in a dataset named 'crime\_data.' The key components and steps in the code include:
  + Importing Seaborn and Matplotlib for data visualization.
  + Specifying the plot's figure size to ensure readability.
  + Creating a vertical boxplot that represents the distribution of victim ages, with data cleansing to remove missing values.
  + Adding a title and labeling the x-axis of the plot.
  + Displaying the boxplot, which showcases the interquartile range and identifies potential outliers in victim ages.



In essence, the code generates a boxplot to provide insights into the distribution of victim ages within crime data, aiding the identification of potential age-related anomalies or outliers as the above boxplot.

1. **Demographic Factors:**
2. Descriptive Statistics for Victim Age and Gender:



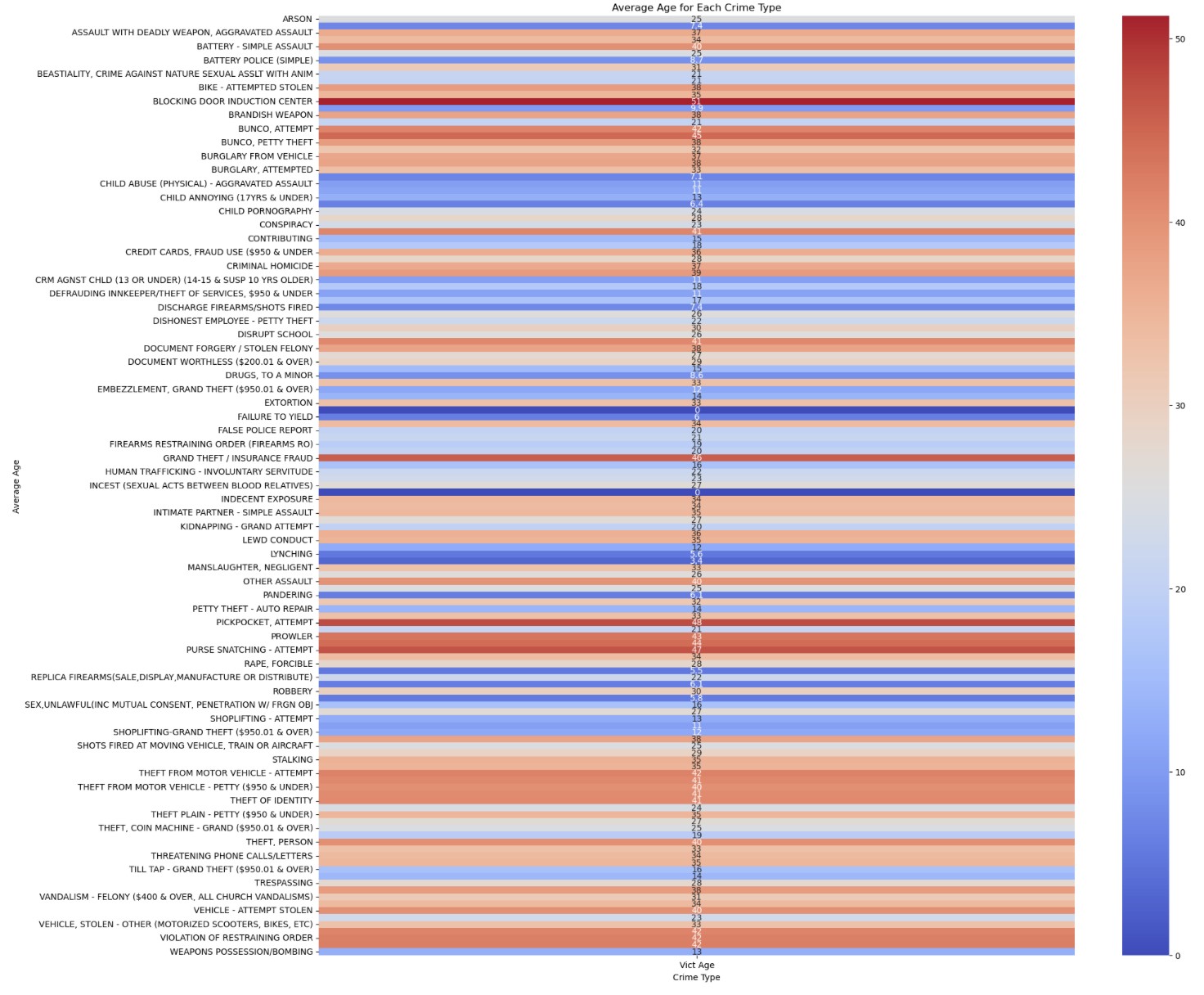
* + The code begins by displaying descriptive statistics for the 'Vict Age' and 'Vict Sex' columns from the dataset.
  + The `describe()` function provides statistics such as the count, mean, standard deviation, minimum, 25th percentile , median , 75th percentile , and maximum values for victim age.
  + It also displays the counts of each category of victim gender

1. Analyzing Age and Crime Type Correlations:



* + The code generates a heatmap to visualize the average victim age for each type of crime ('Crm Cd

Desc').



* + The heatmap uses a color scale (cmap='coolwarm') to represent average ages, with cooler colors indicating lower ages and warmer colors indicating higher ages.
  + Each row of the heatmap represents a different crime type, and the values in the cells represent the average victim age for that particular crime type.
  + This analysis helps understand whether certain types of crimes tend to have older or younger victims.

1. Analyzing Gender and Crime Type Correlations:



Similarly, the code creates a heatmap to visualize the distribution of victim genders for each type of crime.



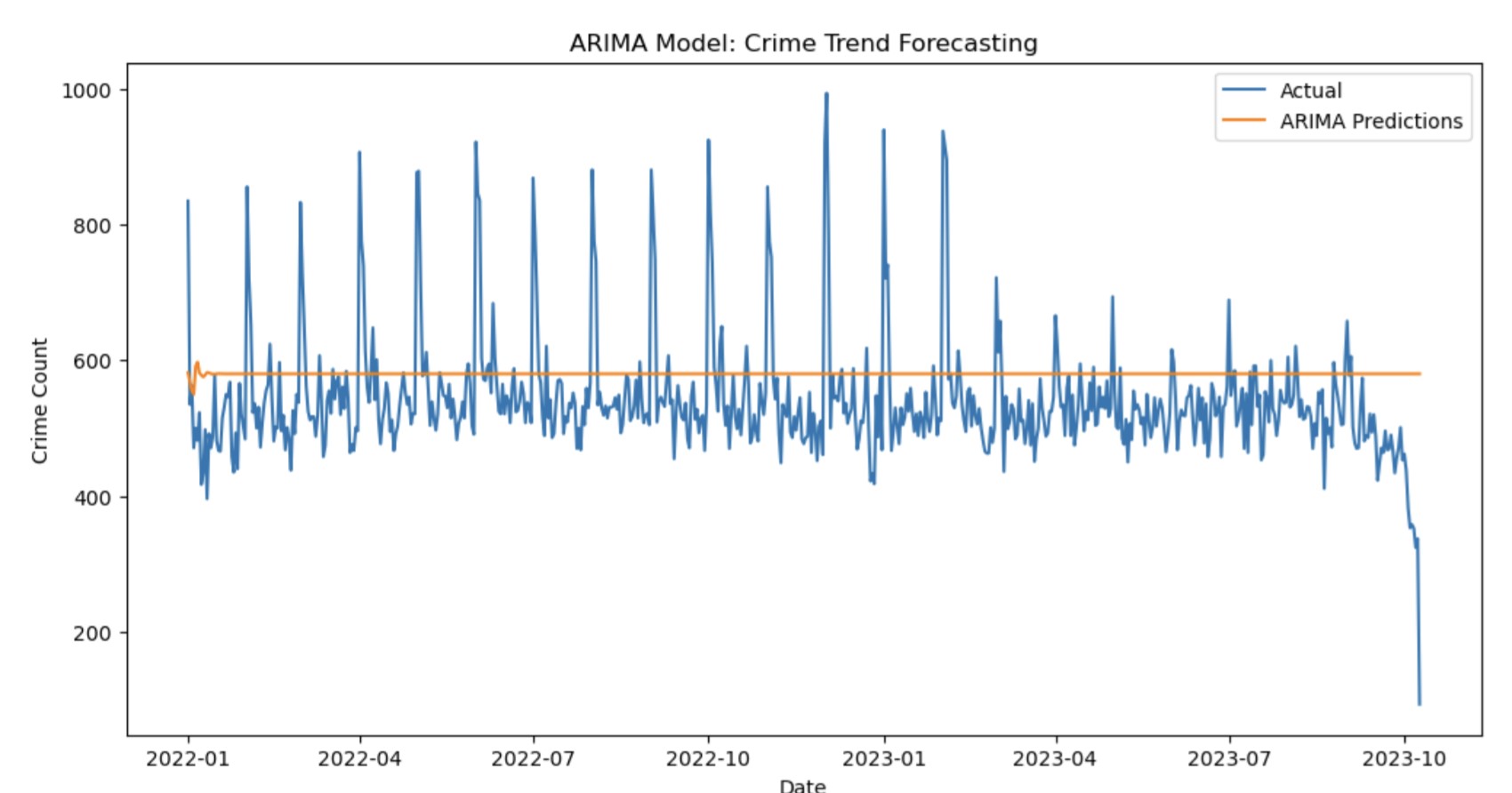
* The heatmap shows how different gender categories are distributed across various crime types.
* The values in the cells represent the counts of victims of each gender category for a particular crime type.
* This analysis helps identify whether certain crimes are more likely to involve victims of a particular gender.

**10. Predicting Future Trends:**



The provided code is about using an ARIMA (Autoregressive Integrated Moving Average) model for forecasting future trends in crime count based on historical crime data.

1. Data Preparation:
   * The code starts by setting a date format and converting the 'DATE OCC' column in the dataset into a datetime format.
   * It sets the 'DATE OCC' column as the index for the dataset. This is a common practice in time series analysis.
2. Resampling Data:
   * The code resamples the crime data to a daily frequency ('D'). This is done to create a time series of daily crime counts.
3. Splitting Data into Train and Test Sets:
   * The code splits the time series data into a training set, which includes data up to January 1, 2022, and a test set, which includes data from January 1, 2022, onward.
4. ARIMA Modeling:
   * The ARIMA model (AutoRegressive Integrated Moving Average) is used to capture the temporal dependencies and trends in the crime data.
   * The model is initialized with an order of (5, 1, 0), which represents the number of autoregressive (AR) terms, differences (I), and moving average (MA) terms, respectively. ● The model is fitted to the training data using the `model.fit()` function.
5. Making Predictions:
   * The model is used to make predictions on the test data using `model\_fit.forecast()` for the same number of time steps as in the test set.
6. Evaluation:
   * The code calculates the Root Mean Squared Error (RMSE) to measure the accuracy of the ARIMA model's predictions. The RMSE quantifies the difference between the predicted and actual crime counts. In this case, the RMSE is approximately 96.32.
7. Plotting the Results:



* + The code generates a plot to visualize the actual crime counts in the test set and the ARIMA model's predictions.
  + The x-axis represents the date, and the y-axis represents the crime count.
  + This plot provides a visual comparison of the model's predictions with the actual crime counts. In summary, this code demonstrates the use of time series analysis with an ARIMA model to forecast future crime trends. It uses historical crime data to train the model and then makes predictions for a specified period in the future. The RMSE value is a measure of how well the model's predictions align with the actual data, and the plot visually illustrates the model's performance in forecasting crime trends.