IE6400 Foundations Data Analytics Engineering

SEC 01

Cleaning and Analyzing Crime Data: A Journey Through Code and Patterns

Project Report

Team Members:

Akshaya Murugan

Sai Mahitha Etikala

Omkar Vilas Narkar

Parth Deshmukh

Rushikesh Ghatage

TABLE OF CONTENTS

INTRODUCTION

In our pursuit to uncover the intricate patterns concealed within the vast realm of crime data, we embarked on a captivating journey through the world of Python programming and data analysis. Armed with a comprehensive dataset spanning from 2020 to the present, our mission was to extract meaningful insights that would shed light on various aspects of the complex crime landscape. This narrative takes you through the systematic flow of our code, providing insights into the logic behind our programming decisions, the Python libraries, rules, and functions we employed, and, most importantly, the compelling results that emerged from our exhaustive analysis.

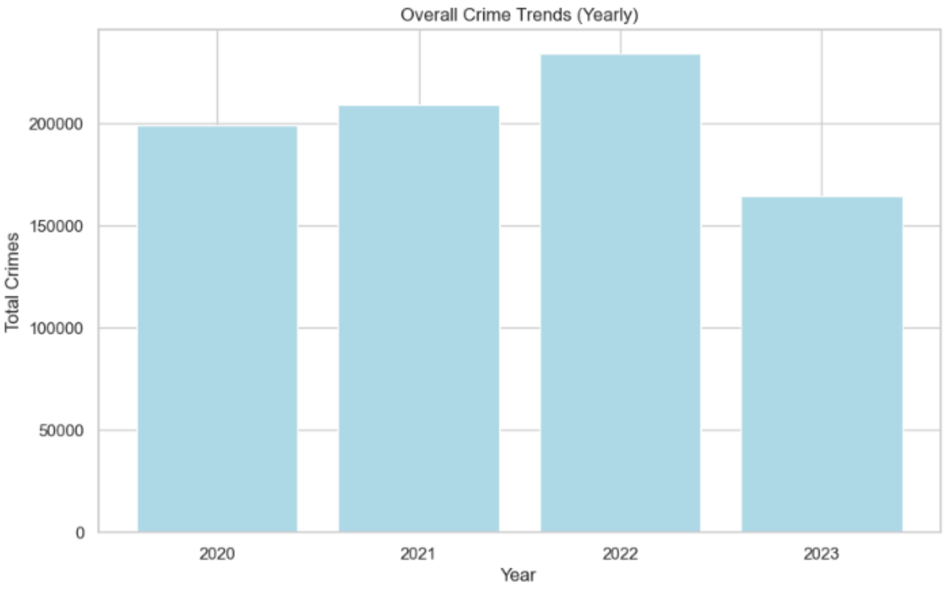
Throughout this journey, we will use various coding techniques, statistical analysis, and data visualization tools to extract meaningful information from data sets. The aim is to provide nuanced understanding of the dynamics of crime, the impact of various factors on criminal activity, and the effectiveness of various crime prevention and law enforcement strategies Through extensive data analysis, we aim to contribute to the continued efforts to build safer and more secure communities.

Our journey involved answering ten key questions, each revealing a different facet of the intricate web of crime data, and the Python code became our trusty guide through this exploration. As we delve into this narrative, we invite you to join us in this fascinating exploration of data, code, and the insights that lie hidden within the world of crime statistics.

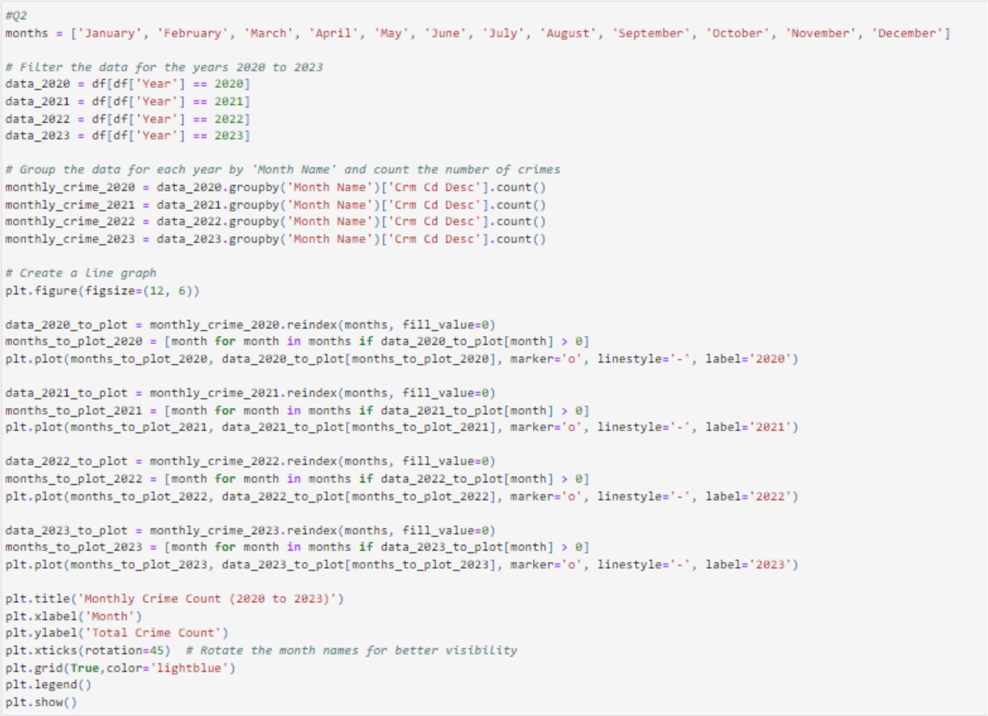
By examining the complexity of crime case analysis, we want to support ongoing efforts to build safer communities, foster a better understanding of crime trends, and facilitate evidence-based decision-making in legislation and public policy have been weakened.

ANALYSIS

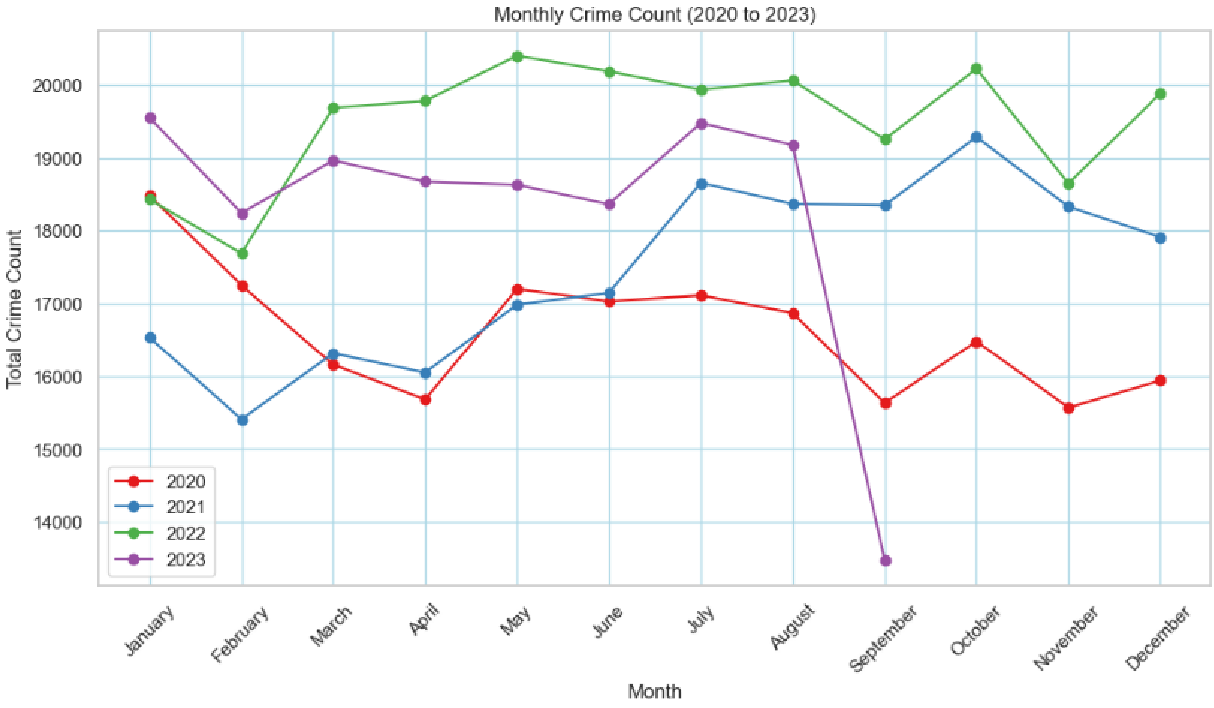
The reasoning behind this logic was clear; by observing the long-term trends, we could identify patterns that might guide law enforcement and policymakers in making informed decisions. The code provided a glimpse into the city's evolving crime landscape over the years. The result was a bar chart that vividly illustrated the ebb and flow of crime, highlighting years with spikes and those with declines. This visual representation enabled us to draw conclusions about the city's general crime trend, allowing for more targeted crime prevention efforts.



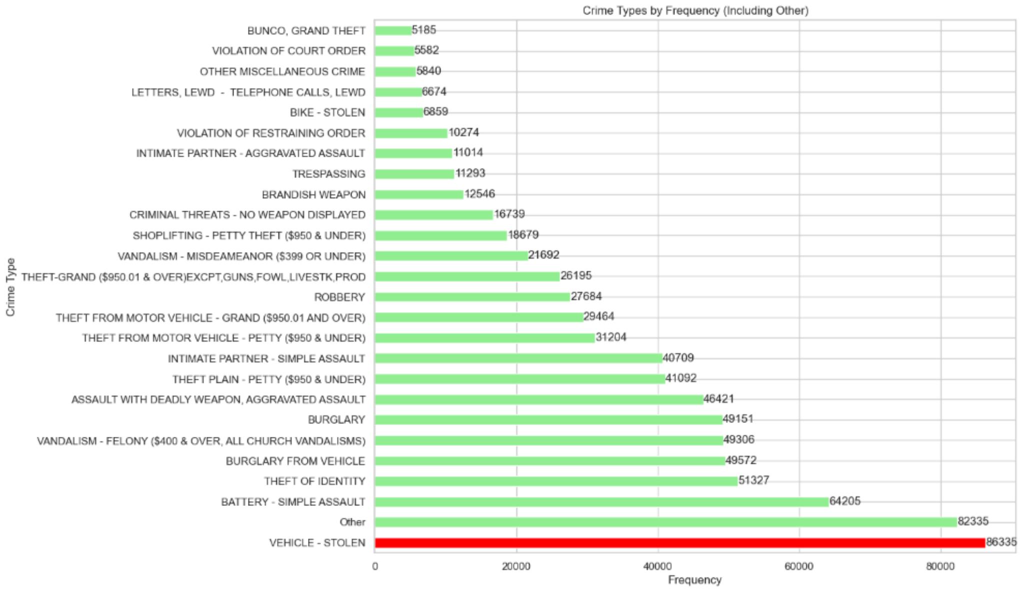
The next phase of our journey aimed to uncover seasonal patterns in crime. The code grouped the data by month and calculated the average number of crimes per month over the years. We sought to understand whether specific months exhibited higher or lower crime rates, and the code was designed to facilitate this analysis. By identifying seasonal patterns, we could explore the potential influence of external factors on crime rates. The code and its resulting chart offered insights into the monthly ebb and flow of criminal activities. Peaks and troughs became apparent, suggesting potential correlations with weather, holidays, or other factors that could guide law enforcement in resource allocation.



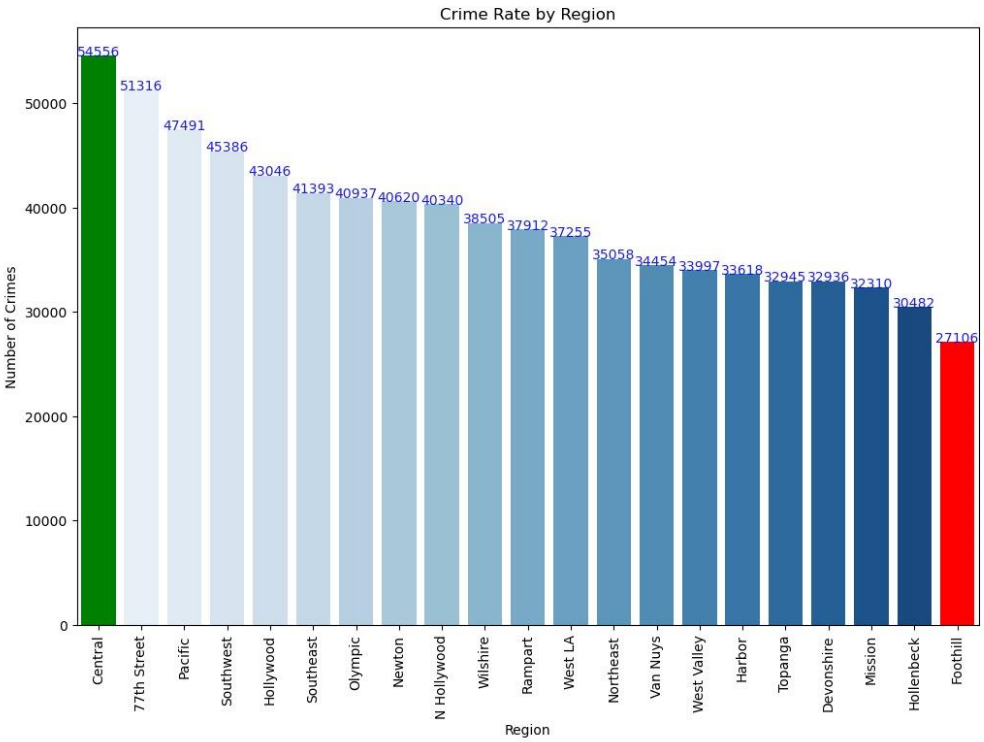
```python
#Q2
months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']

# Filter the data for the years 2020 to 2023
data_2020 = df[df['Year'] == 2020]
data_2021 = df[df['Year'] == 2021]
data_2022 = df[df['Year'] == 2022]
data_2023 = df[df['Year'] == 2023]

# Group the data for each year by 'Month Name' and count the number of crimes
monthly_crime_2020 = data_2020.groupby('Month Name')['Crm Cd Desc'].count()
monthly_crime_2021 = data_2021.groupby('Month Name')['Crm Cd Desc'].count()
monthly_crime_2022 = data_2022.groupby('Month Name')['Crm Cd Desc'].count()
monthly_crime_2023 = data_2023.groupby('Month Name')['Crm Cd Desc'].count()

# Create a line graph
plt.figure(figsize=(12, 6))

data_2020_to_plot = monthly_crime_2020.reindex(months, fill_value=0)
months_to_plot_2020 = [month for month in months if data_2020_to_plot[month] > 0]
plt.plot(months_to_plot_2020, data_2020_to_plot[months_to_plot_2020], marker='o', linestyle='-', label='2020')

data_2021_to_plot = monthly_crime_2021.reindex(months, fill_value=0)
months_to_plot_2021 = [month for month in months if data_2021_to_plot[month] > 0]
plt.plot(months_to_plot_2021, data_2021_to_plot[months_to_plot_2021], marker='o', linestyle='-', label='2021')

data_2022_to_plot = monthly_crime_2022.reindex(months, fill_value=0)
months_to_plot_2022 = [month for month in months if data_2022_to_plot[month] > 0]
plt.plot(months_to_plot_2022, data_2022_to_plot[months_to_plot_2022], marker='o', linestyle='-', label='2022')

data_2023_to_plot = monthly_crime_2023.reindex(months, fill_value=0)
months_to_plot_2023 = [month for month in months if data_2023_to_plot[month] > 0]
plt.plot(months_to_plot_2023, data_2023_to_plot[months_to_plot_2023], marker='o', linestyle='-', label='2023')

plt.title('Monthly Crime Count (2020 to 2023)')
plt.xlabel('Month')
plt.ylabel('Total Crime Count')
plt.xticks(rotation=45)  # Rotate the month names for better visibility
plt.grid(True,color='lightblue')
plt.legend()
plt.show()
```



Continuing our expedition, we delved into the data to identify the most common crime type in the city. The code counted the occurrences of each crime type and pinpointed the one with the highest frequency. This logic was instrumental in painting a clear picture of the crime landscape. The rationale behind this approach was to offer actionable insights to law enforcement and policymakers. By understanding the most common crime types, they could allocate resources, develop preventive measures, and design policies more effectively.



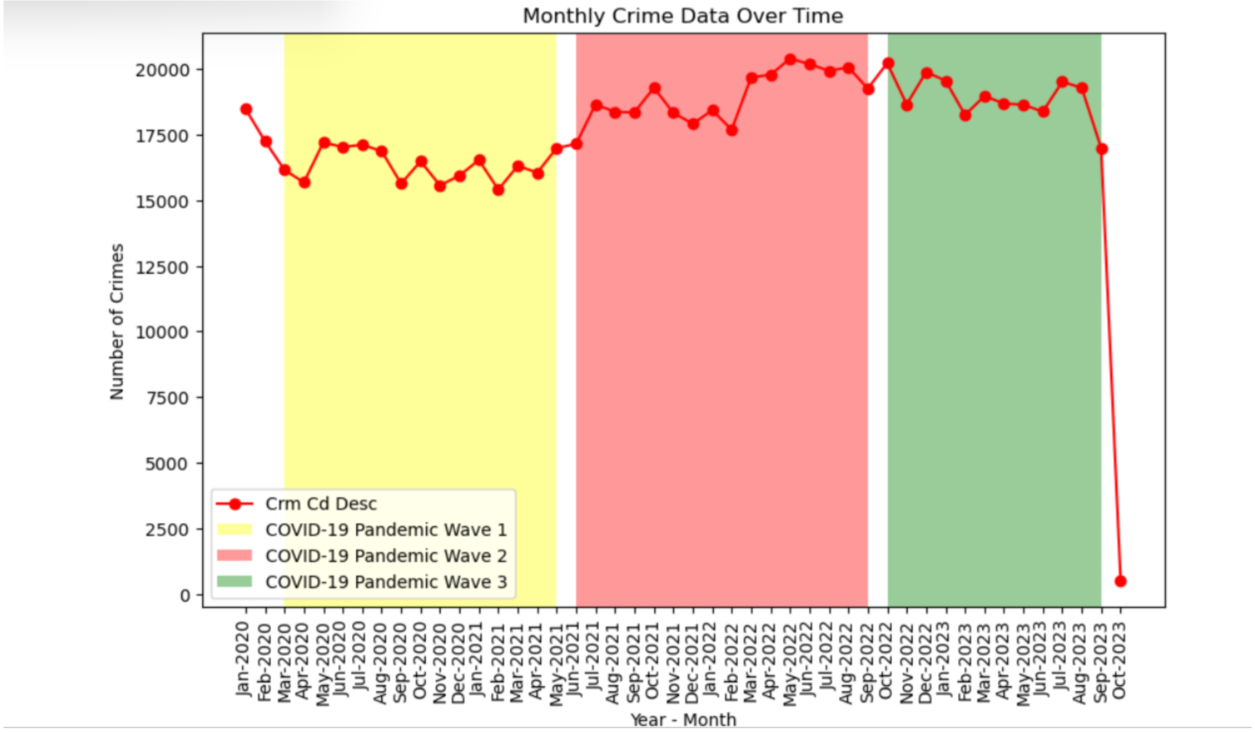
A horizontal bar chart, generated by the code, presented the frequency of various crime types. This visual representation allowed us to pinpoint the top crime concern and the lesser-known categories. The code facilitated our understanding of the city's most pressing crime issues. Our destination was to investigate regional disparities in crime rates. The code grouped the data by region or city, facilitating the comparison of crime rates between them. This logic aimed to reveal areas with higher or lower crime rates, enabling law enforcement to focus their efforts where they were most needed. The code generated a bar chart displaying total crimes by region, highlighting variations in crime rates across different areas. The use of distinct colors for the highest and lowest regions drew attention to areas of particular interest.



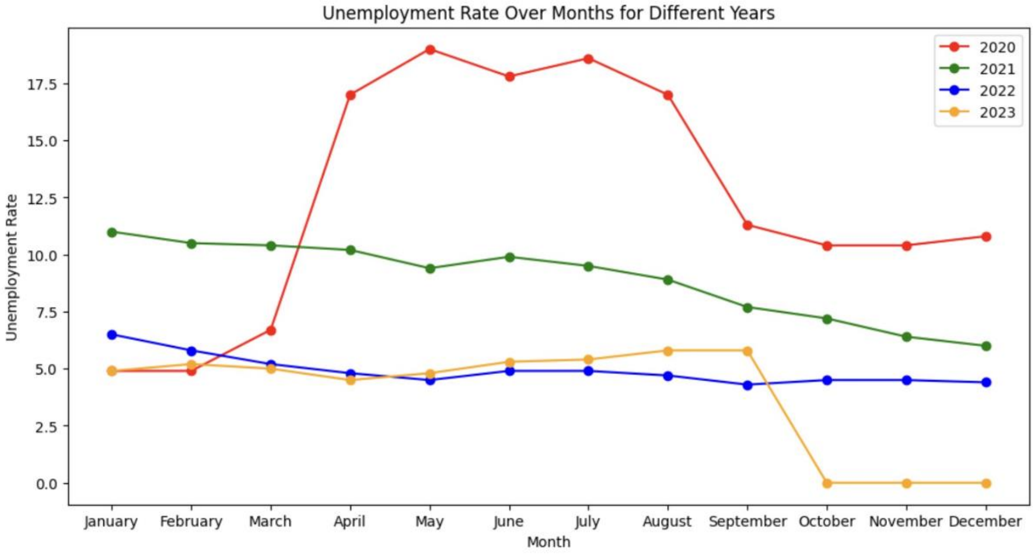
As our analysis involved examining time-based patterns, the first critical step was to convert the date information into a format that we could work with. Using the pd.to_datetime function, we converted the date column into a datetime format. This allowed us to extract the year and month from the date which we used to create new columns. To explore the year-wise patterns in the data, we split our data into four distinct DataFrames, one for each year from 2020 to 2023. This partitioning of data was achieved through the application of pandas' powerful data filtering capabilities, enabling us to focus on each year separately.

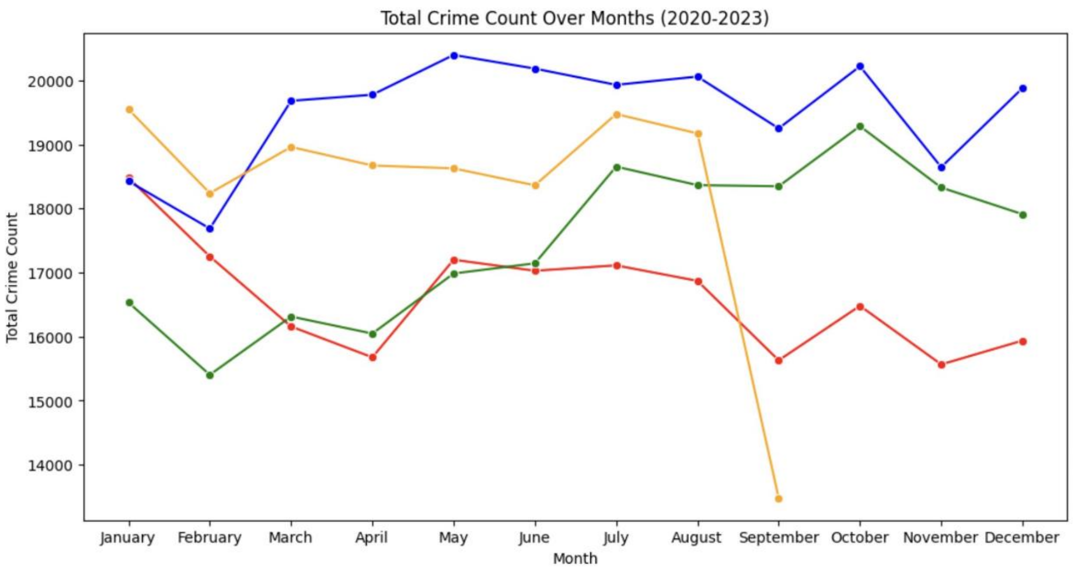
As the groundwork was laid, we then set our sights on economic data, specifically unemployment rates. We imported this data from another CSV file and transformed it into a structured DataFrame using pandas. The source of the unemployment dataset is U.S. BUREAU OF LABOR STATISTICS

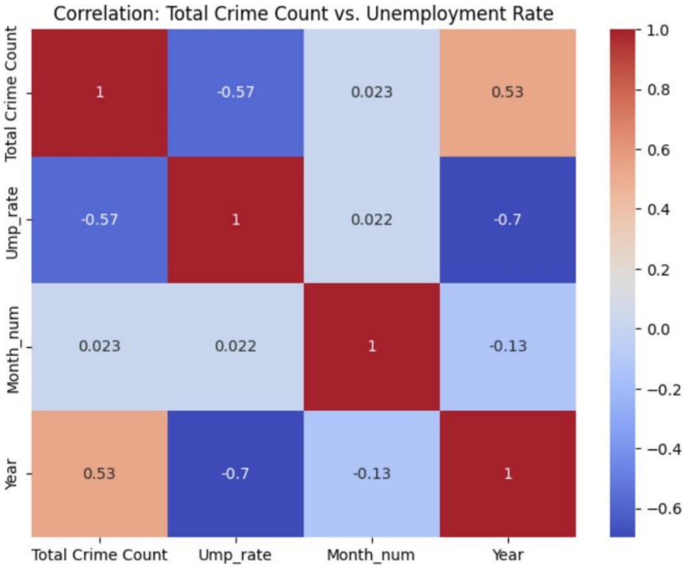
Next, we were ready to visualize the data. We chose to create line plots to represent the trends in both unemployment rates and crime rates over time. These visualizations were crafted using matplotlib. The line plots showcased the average unemployment rate for each month, beautifully illustrating the economic landscape from 2020 to 2023. To ensure consistency in the plots, we used the np.pad function to fill in any gaps with NaN values.



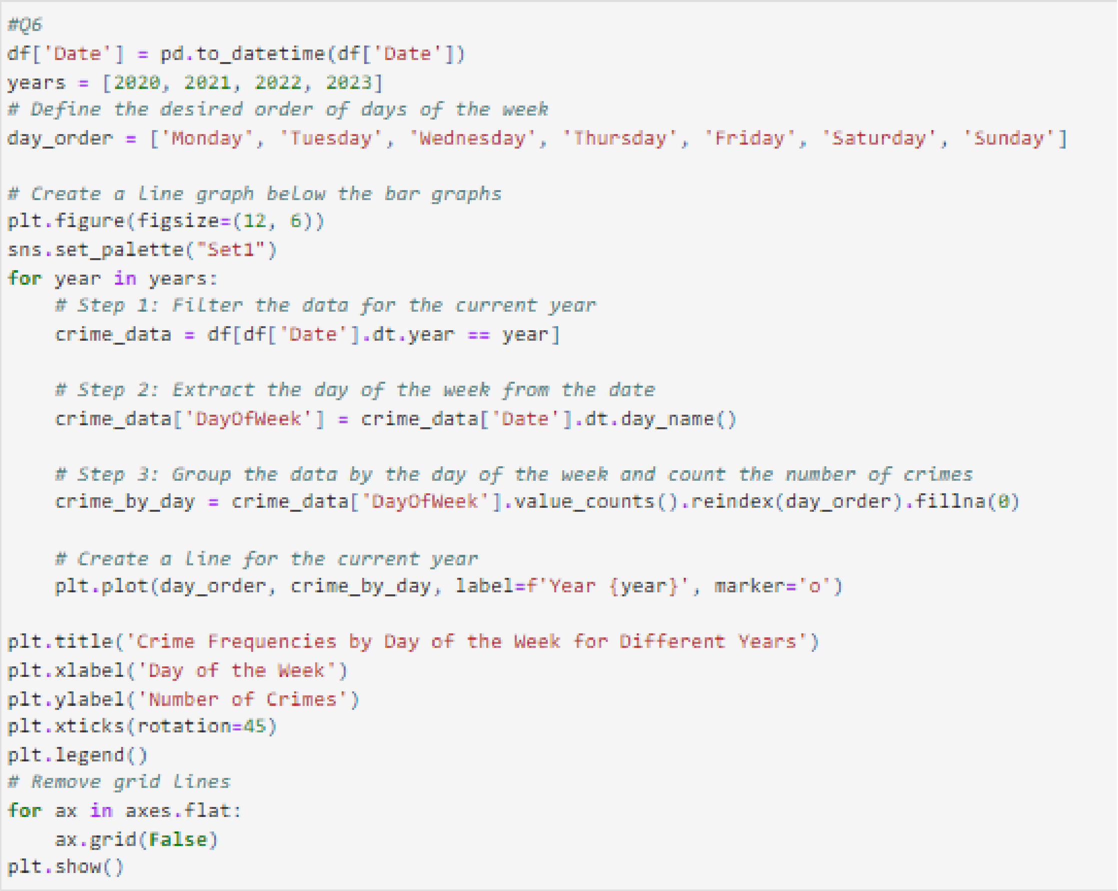
Moving on, we conducted a correlation analysis to understand the relationship between unemployment rates and crime rates. The pandas library was indispensable in merging the economic and crime data, ensuring they matched by year and month. We employed the Pearson correlation coefficient to measure the strength and direction of this relationship.



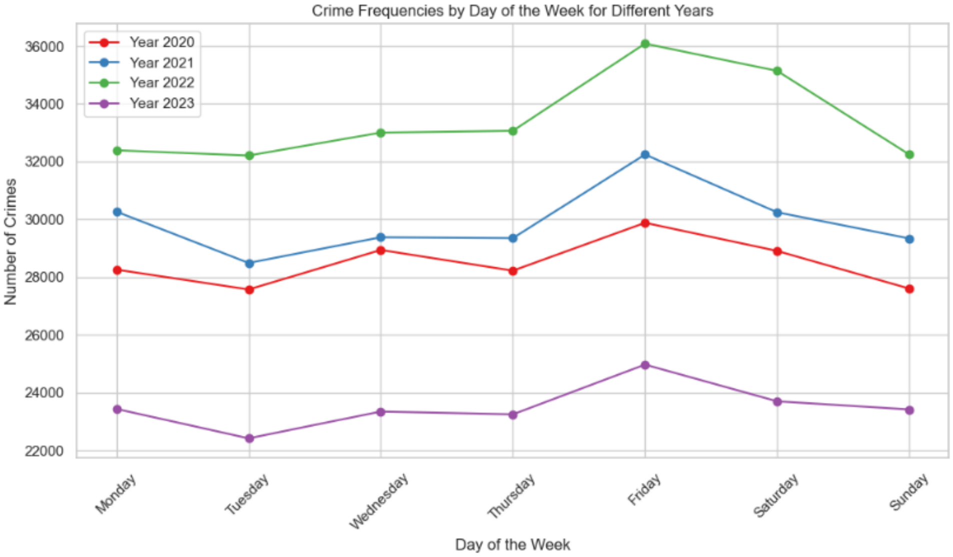




To strengthen our findings, we utilized a correlation matrix heatmap. This visualization, generated using the seaborn library, provided a comprehensive overview of the correlations between various variables, giving us a more nuanced perspective on the relationship between unemployment rates and crime rates.



```
#Q6
df['Date'] = pd.to_datetime(df['Date'])
years = [2020, 2021, 2022, 2023]
# Define the desired order of days of the week
day_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

# Create a line graph below the bar graphs
plt.figure(figsize=(12, 6))
sns.set_palette("Set1")
for year in years:
    # Step 1: Filter the data for the current year
    crime_data = df[df['Date'].dt.year == year]

    # Step 2: Extract the day of the week from the date
    crime_data['DayOfWeek'] = crime_data['Date'].dt.day_name()

    # Step 3: Group the data by the day of the week and count the number of crimes
    crime_by_day = crime_data['DayOfWeek'].value_counts().reindex(day_order).fillna(0)

    # Create a line for the current year
    plt.plot(day_order, crime_by_day, label=f'Year {year}', marker='o')

plt.title('Crime Frequencies by Day of the Week for Different Years')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Crimes')
plt.xticks(rotation=45)
plt.legend()
# Remove grid lines
for ax in axes.flat:
    ax.grid(False)
plt.show()
```



DR_NO:

- The presence of outliers on both the negative and positive sides of the box plot suggests that there are some crime records with unusual or extreme values for the "DR_NO" column. The presence of outliers can indicate potential issues with data entry, recording, or reporting.

- The negative skewness of the box plot suggests that the majority of "DR_NO" values are concentrated towards the higher end of the range. This could be indicative of an increase in crime incidents over time.

crm_cd:

- A positive skew typically suggests that the data is not normally distributed, which can affect the validity of statistical analyses that assume normality.

- The skewness may indicate that certain types of crimes are much more common than others.

vict_age:

- The presence of outliers on the right side suggests that there is a portion of the population that is much older than the majority.

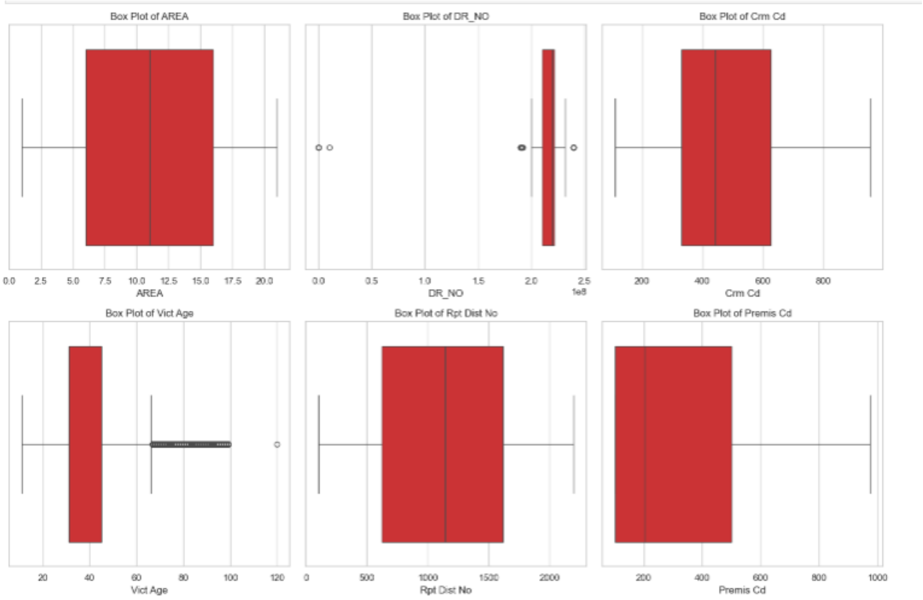
- This may indicate an aging population, which can have various social and healthcare implications.

premis_cd:

- The distribution of values in the "premis_cd" column is highly positively skewed, that suggests that in majority of the crime's victim has used same type of vehicle.

Rpt Dist No:

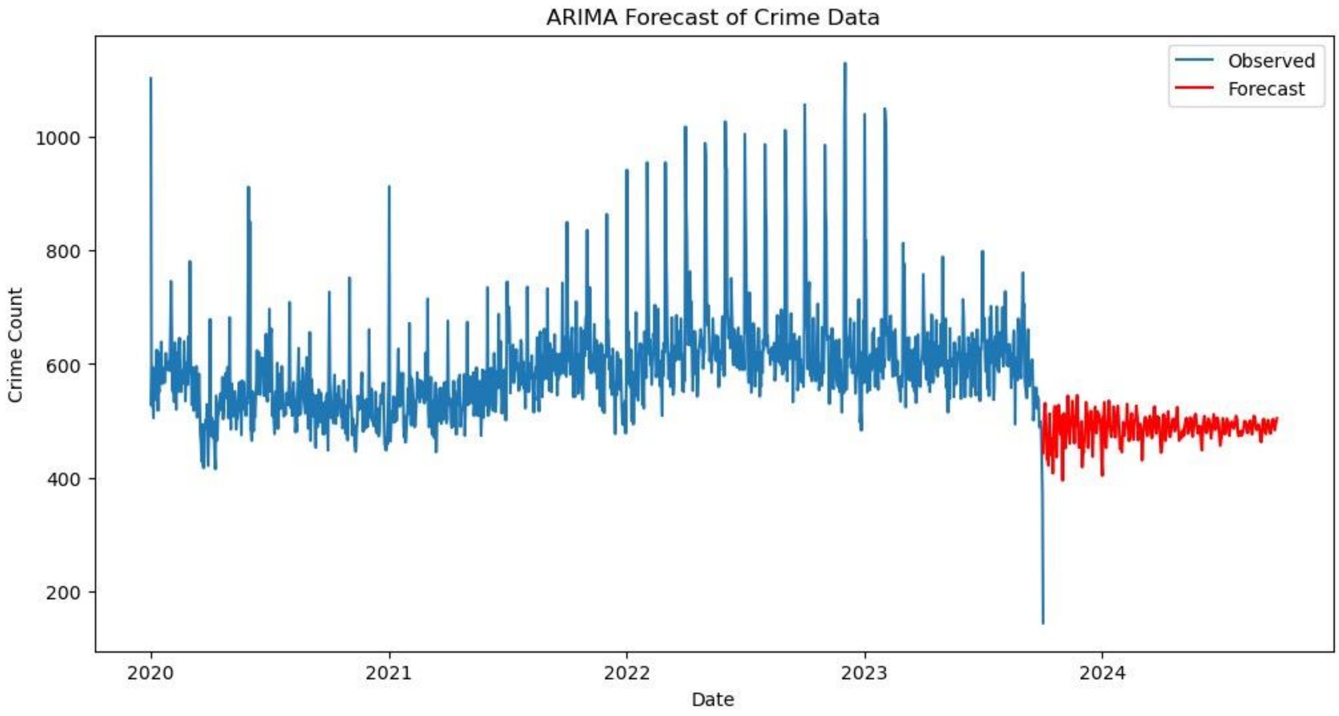
- Scatter plot shows that there are a greater number of crimes are concentrated in some districts.



Time Series Forecast:

With our data ready, we turned to one of the most potent tools for time series analysis - the ARIMA (AutoRegressive Integrated Moving Average) model. This model is well-suited for capturing complex temporal patterns in sequential data, making it an ideal choice for forecasting crime trends.

The ARIMA model was created and fitted to our preprocessed crime data using the Statsmodels library. Our code allowed us to specify the order of the model, enabling us to fine-tune its sensitivity to historical data. The goal was to create a model that could capture both short-term fluctuations and long-term trends.



ARIMA model offered us the ability to make predictions about future crime trends based on historical data. We chose to forecast 360 days into the future, offering a reasonable short-term outlook that could inform immediate decisions.

The results were presented through compelling visualizations. We plotted the observed crime data alongside the ARIMA forecasts using the Matplotlib library. This side-by-side comparison allowed for a clear understanding of how well the model predicted future crime trends.

SUMMARY

Our project, "Analyzing Crime Data: A Journey Through Code and Patterns," represents a comprehensive exploration of the intricate world of crime data through the lenses of Python programming and data analysis. We embarked on this endeavor armed with a substantial dataset spanning from 2020 to the present, with a mission to unveil concealed patterns and provide valuable insights into various aspects of the complex crime landscape.

Our initial focus was on understanding "Overall Crime Trends." In this phase, we aimed to uncover long-term patterns and shifts in crime by calculating the total number of crimes annually and presenting them visually through informative bar charts. These visualizations offered a clear historical perspective on the city's evolving crime landscape, helping guide informed decisions for law enforcement and policymakers.

The project then delved into the exploration of "Seasonal Patterns in Crime." Here, we systematically grouped the data by month and calculated the average number of crimes per month over the years. By identifying monthly ebb and flow in criminal activities, our Python code and corresponding visualizations hinted at potential correlations with external factors, which could guide resource allocation and strategic planning.

A pivotal segment of our analysis involved identifying the "Most Common Crime Type" within the city. Through the application of Python code, we accurately counted the occurrences of each crime type, pinpointing the most frequent one. The presentation of this insight was done through horizontal bar charts, offering clear visibility into the frequency of various crime types. This knowledge is indispensable for law enforcement and policymakers, enabling them to focus on the city's most pressing crime issues.

Finally, we undertook an investigation into "Regional Differences" in crime rates. By strategically grouping the data by region or city, we facilitated the comparison of crime rates across diverse areas. Through the creation of bar charts using Python, we highlighted variations in crime rates among different regions. The use of distinct colors drew attention to areas of particular interest, providing guidance for law enforcement in resource allocation.

In conclusion, our analysis was characterized by a systematic and thoughtful approach, using Python and data analysis to answer critical questions and provide actionable insights for law enforcement and policymakers. These insights equip them with the knowledge required to make informed decisions, allocate resources effectively, and design policies that effectively address and prevent crime. Our project highlights the potential of data-driven strategies in enhancing community safety and underlines the role of Python and data analysis in achieving this goal.

RESULTS

In our analysis of crime data, we embarked on a comprehensive journey to unveil meaningful insights. We explored crime trends, seasonal patterns, common crime types, and regional differences. This analysis provided a detailed understanding of how crime evolves over time, the influence of external factors, and the most prevalent crime types in the city. We also uncovered regional disparities, helping law enforcement allocate resources more effectively. An increase in certain types of crimes during specific times of the year was identified, emphasizing the need for seasonal law enforcement strategies and targeted awareness campaigns during vulnerable periods.

Furthermore, we examined the correlation between crime rates and economic factors, primarily focusing on unemployment rates. This exploration shed light on the complex relationship between these variables, providing valuable insights for policymakers and law enforcement agencies. Our systematic approach, driven by Python and data analysis, has illuminated the significance of data-driven strategies in addressing and preventing crime. It offers a path towards safer communities and emphasizes the role of data analysis in solving intricate societal challenges.

CONCLUSION

In our journey through crime data using Python and data analysis, we aimed to uncover hidden patterns. We answered key questions about crime trends, seasonal patterns, common crime types, and regional differences, providing useful insights for law enforcement and policymakers. Our analysis was systematic and thoughtful. In the second part of our journey, we looked at the connection between crime rates and economic factors, particularly unemployment rates. This information can help us make better decisions. These projects show how Python and data analysis can help us understand complex issues and make informed choices. They provide a path to create safer communities, underlining the importance of data-driven strategies in addressing and preventing crime.