Module 6 – Final Project Report Analysis

Boston Crime Incident

(2018 - 2022)

Professor - Roy Wada

Group Number - 5

Submitted by:

Kumar Saransh

Nishanth Nagesh

Yunong Li

Northeastern University: College of Professional Studies

ALY 6015: Intermediate Analytics

14th December, 2024

Northeastern University
College of Professional Studies

Table Of CONTENTS

Introduction:

Our analysis utilizes a dataset sourced from the Boston Police Department's crime incident reports, which includes key details about crime incidents from 2015 to 2024. This report focuses on the years 2018, 2020, and 2022. We employ descriptive statistics, visualizations, and logistic regression modeling to uncover meaningful patterns.

About the Dataset:

Publisher: Department of Innovation and Technology
Location: Boston (all)

Description: The dataset used in this report is sourced from the Boston Police Department's (BPD) crime incident report system, which captures essential details about incidents to which BPD officers respond. Covering records from 2015 through 2024, the dataset focuses on documenting the type of crime, along with the time and location of each incident. This streamlined approach, part of a new system introduced in June 2015, reduces the number of fields, providing more concise information on each case.

Questions to Answer from the Dataset:

1. What types of offenses are more prevalent in each district?
2. If possible, which areas show the highest frequency of shootings?
3. Is there a trend for any particular months of the year where crimes occur?
4. Is there a trend in incidents depending on a day of the week?
5. How does time/hour of day affect the type of crimes reported?

Analytical Plans and Methods:

1. **Time Series Analysis:** this analyzes the crime incidence trend and patterns in time, month, day of the week, hour, etc., to see if criminal activities are temporally predisposed.
2. **Geospatial Analysis:** The latitudinal and longitudinal data provide information on heat maps or cluster analyses that were made to denote crime hotspots and how they relate to different districts or areas where reporting is from.
3. **Chi-Square Test of Independence:** This test is appropriate for various categorical variables on one or more dimensions. Such as the association between DISTRICT and SHOOTING.
4. **Logistic Regression:** this will model the probability of a shooting using the predictor variables that will be beneficial in pinning the causes leading to a serious incident.

Outcome Variables:

SHOOTING: A binary variable indicating whether a shooting occurred (1) or not (0).

Descriptive Statistics Tables

A) Descriptive Statistics for Districts by Year (2018-2022):

Table 1. Descriptive Statistics of Boston Crime Incidents for Districts by Year (2018-2022)

Variable	Overall N = 243,634[1]	2018 N = 98,888[1]	2020 N = 70,894[1]	2022 N = 73,852[1]
DISTRICT_NAME				
Brighton	16,569 (6.8%)	6,159 (6.2%)	4,756 (6.7%)	5,654 (7.7%)
Charlestown	5,095 (2.1%)	2,034 (2.1%)	1,600 (2.3%)	1,461 (2.0%)
Dorchester	31,085 (13%)	12,957 (13%)	8,992 (13%)	9,136 (12%)
Downtown	26,509 (11%)	10,976 (11%)	7,013 (9.9%)	8,520 (12%)
East Boston	10,660 (4.4%)	3,800 (3.8%)	3,141 (4.4%)	3,719 (5.0%)
External	315 (0.1%)	0 (0%)	246 (0.3%)	69 (<0.1%)
Hyde Park	13,803 (5.7%)	5,517 (5.6%)	4,177 (5.9%)	4,109 (5.6%)
Jamaica Plain	13,878 (5.7%)	5,480 (5.5%)	4,103 (5.8%)	4,295 (5.8%)
Mattapan	26,704 (11%)	11,560 (12%)	7,828 (11%)	7,316 (9.9%)
Roxbury	37,262 (15%)	16,340 (17%)	10,720 (15%)	10,202 (14%)
South Boston	19,053 (7.8%)	7,516 (7.6%)	5,440 (7.7%)	6,097 (8.3%)
South End	31,329 (13%)	12,454 (13%)	9,283 (13%)	9,592 (13%)
West Roxbury	11,372 (4.7%)	4,095 (4.1%)	3,595 (5.1%)	3,682 (5.0%)

[1]n (%)

Overall Trends:

- The total number of reported crime incidents decreased from 2018 to 2020 but increased slightly in 2022.
- The distribution of incidents across districts remained relatively consistent over the years, with Roxbury, Dorchester, and Mattapan consistently reporting the highest numbers.

District-Specific Observations:

- Dorchester, Roxbury, and Mattapan: These districts consistently reported the highest number of incidents, highlighting areas of concern for law enforcement and community safety.
- Brighton: This district saw a significant increase in reported incidents from 2020 to 2022.
- External: The category "External" shows a substantial increase in incidents from 2020 to 2022. This could be due to various factors such as changes in reporting methods or increased activity in specific crime categories.

B) Descriptive Statistics for DAY_OF_WEEK by Year (2018-2022):

Table 2. Descriptive Statistics of Boston Crime Incidents for Day of Week by Year (2018-2022)

Variable	2018 N = 98,888[1]	2020 N = 70,894[1]	2022 N = 73,852[1]
DAY_OF_WEEK			
Monday	14,272 (14%)	10,299 (15%)	10,698 (14%)
Tuesday	14,260 (14%)	10,217 (14%)	10,445 (14%)
Wednesday	14,557 (15%)	10,563 (15%)	10,825 (15%)
Thursday	14,426 (15%)	10,491 (15%)	10,847 (15%)
Friday	14,974 (15%)	10,722 (15%)	11,292 (15%)
Saturday	13,949 (14%)	9,753 (14%)	10,596 (14%)
Sunday	12,450 (13%)	8,849 (12%)	9,149 (12%)

Overall Trends:

- Crime incidents are relatively evenly distributed across the days of the week.
- There is a slight increase in reported crimes on weekends compared to weekdays.

Year-Specific Observations:

- 2018: Friday and Thursday had the highest number of reported incidents.
- 2020: Monday and Tuesday had the highest number of reported incidents.
- 2022: Friday and Thursday had the highest number of reported incidents.

C) Descriptive Statistics for UCR_PART by Year (2018-2022):

Table 3. Descriptive Statistics of Boston Crime Incidents for UCR Part by Year (2018-2022)

Variable	Overall N = 243,634[1]	2018 N = 98,888[1]	2020 N = 70,894[1]	2022 N = 73,852[1]
UCR_PART				
Other	145,179 (60%)	433 (0.4%)	70,894 (100%)	73,852 (100%)
Part One	18,005 (7.4%)	18,005 (18%)	0 (0%)	0 (0%)
Part Three	51,068 (21%)	51,068 (52%)	0 (0%)	0 (0%)
Part Two	29,382 (12%)	29,382 (30%)	0 (0%)	0 (0%)

[1]n (%)

Overall Trends:

- The majority of crime incidents are categorized as "Other" (60%).
- "Part Three" crimes account for 21% of the total incidents.
- "Part One" and "Part Two" crimes constitute a smaller proportion.

Year-Specific Observations:

- 2018: All crime categories were reported in 2018.
- 2020 and 2022: Only "Other" crimes were reported in these years. This could be due to changes in data collection methods or a shift in the types of crimes being reported.

D) Descriptive Statistics for Time of the Day by Year (2018-2022):

Table 4. Descriptive Statistics of Boston Crime Incidents for Hours by Year (2018-2022)

Variable	Overall N = 243,634[1]	2018 N = 98,888[1]	2020 N = 70,894[1]	2022 N = 73,852[1]
HOUR_BINNING				
12 a.m.- 4 a.m.	30,858 (13%)	11,290 (11%)	9,248 (13%)	10,320 (14%)
4 a.m. - 8 a.m.	15,557 (6.4%)	6,535 (6.6%)	4,390 (6.2%)	4,632 (6.3%)
8 a.m. - 12 p.m.	46,127 (19%)	19,162 (19%)	13,136 (19%)	13,829 (19%)
12 p.m. - 4 p.m.	53,994 (22%)	21,871 (22%)	16,081 (23%)	16,042 (22%)
4 p.m. - 8 p.m.	57,697 (24%)	23,798 (24%)	16,667 (24%)	17,232 (23%)
8 p.m. - 12 a.m.	39,401 (16%)	16,232 (16%)	11,372 (16%)	11,797 (16%)

[1]n (%)

Overall Trends:

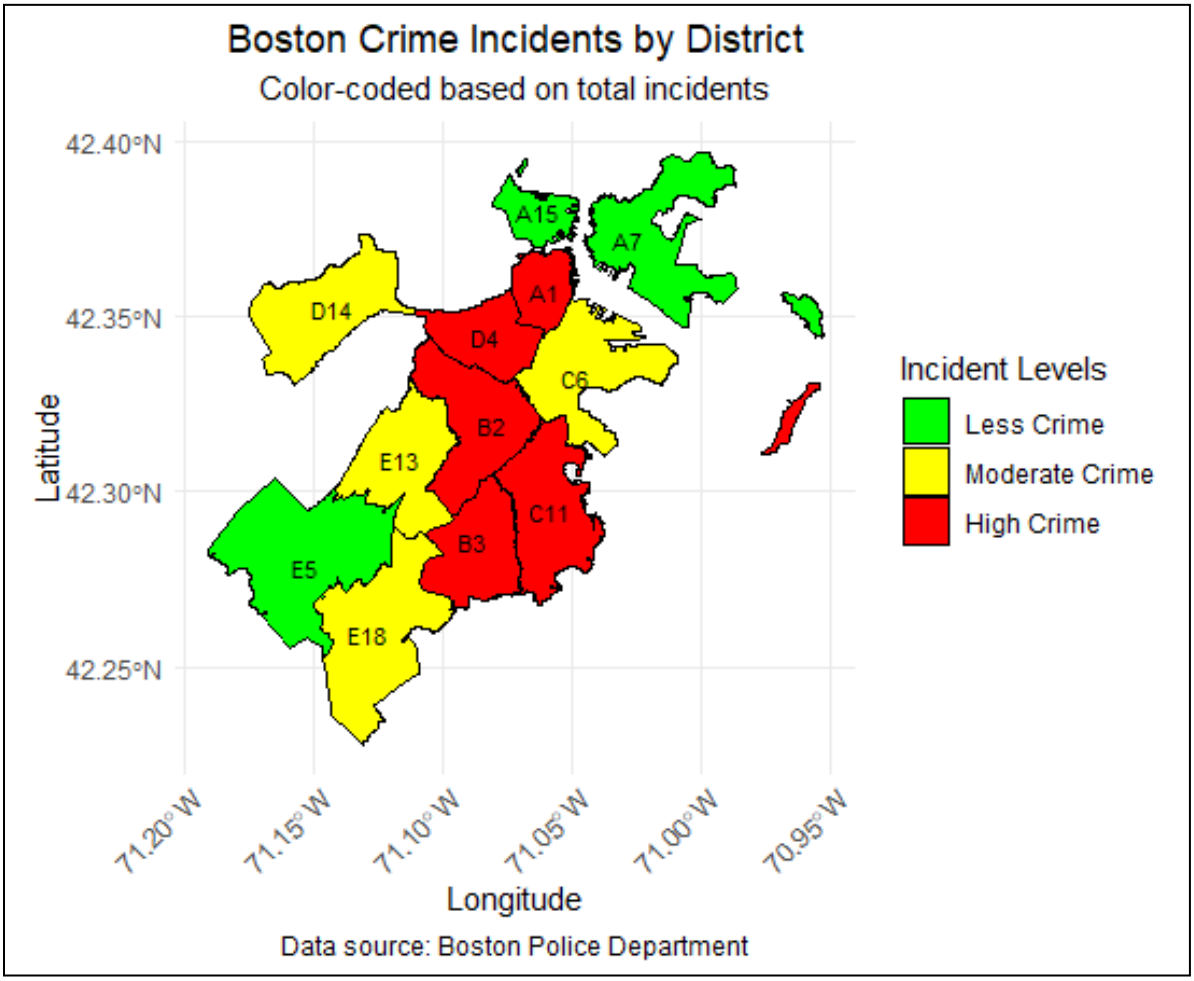
- Crime incidents are most frequent between 4 PM and 8 PM and 12 PM and 4 AM.
- The fewest number of incidents occur between 4 AM and 8 AM.
- The distribution of incidents across different hours of the day remains relatively consistent across the years.

Year-Specific Observations:

- 2018: The highest number of incidents occurred between 4 PM and 8 PM.
- 2020: The distribution of incidents across different hours was relatively even, with a slight peak between 12 PM and 4 PM.
- 2022: The highest number of incidents occurred between 4 PM and 8 PM.

Visualizations:

A) Boston Crime Incidents by District Using Color Coded Map from Year (2018-2022):



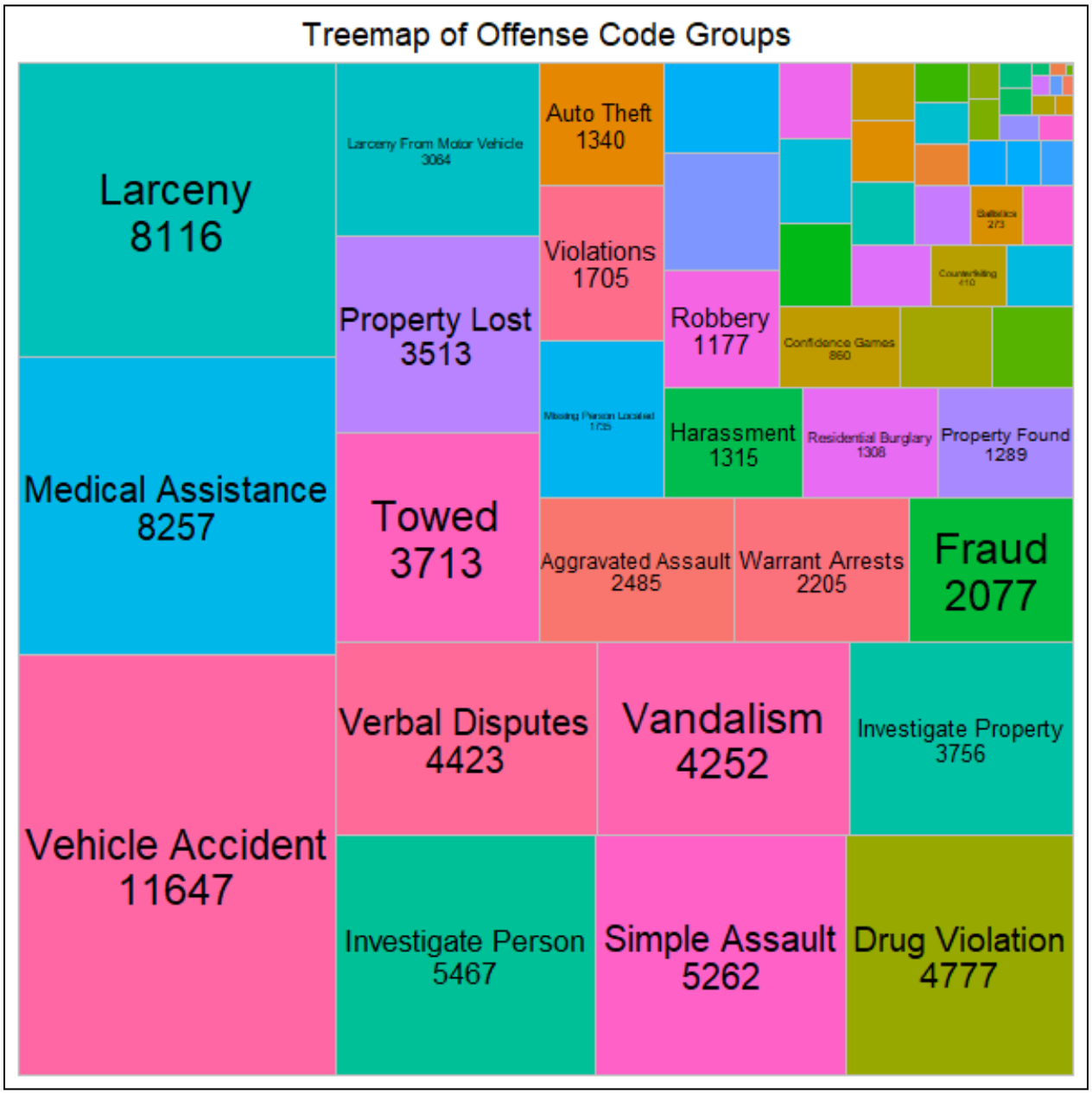
Overall Trends:

- Crime levels vary significantly across different districts in Boston.
- The districts with the highest crime levels are concentrated in the central and southern parts of the city.
- The districts with the lowest crime levels are located in the northern and eastern parts of the city.

Specific Observations:

- High Crime Districts: Districts with high crime levels include C6, C11, and B2. These districts are located in the central and southern parts of the city.
- Moderate Crime Districts: Districts with moderate crime levels include D4, D14, and A15. These districts are located in the northern and eastern parts of the city.
- Low Crime Districts: Districts with low crime levels include A1, A7, E5, and E18. These districts are located in the northern and eastern parts of the city.

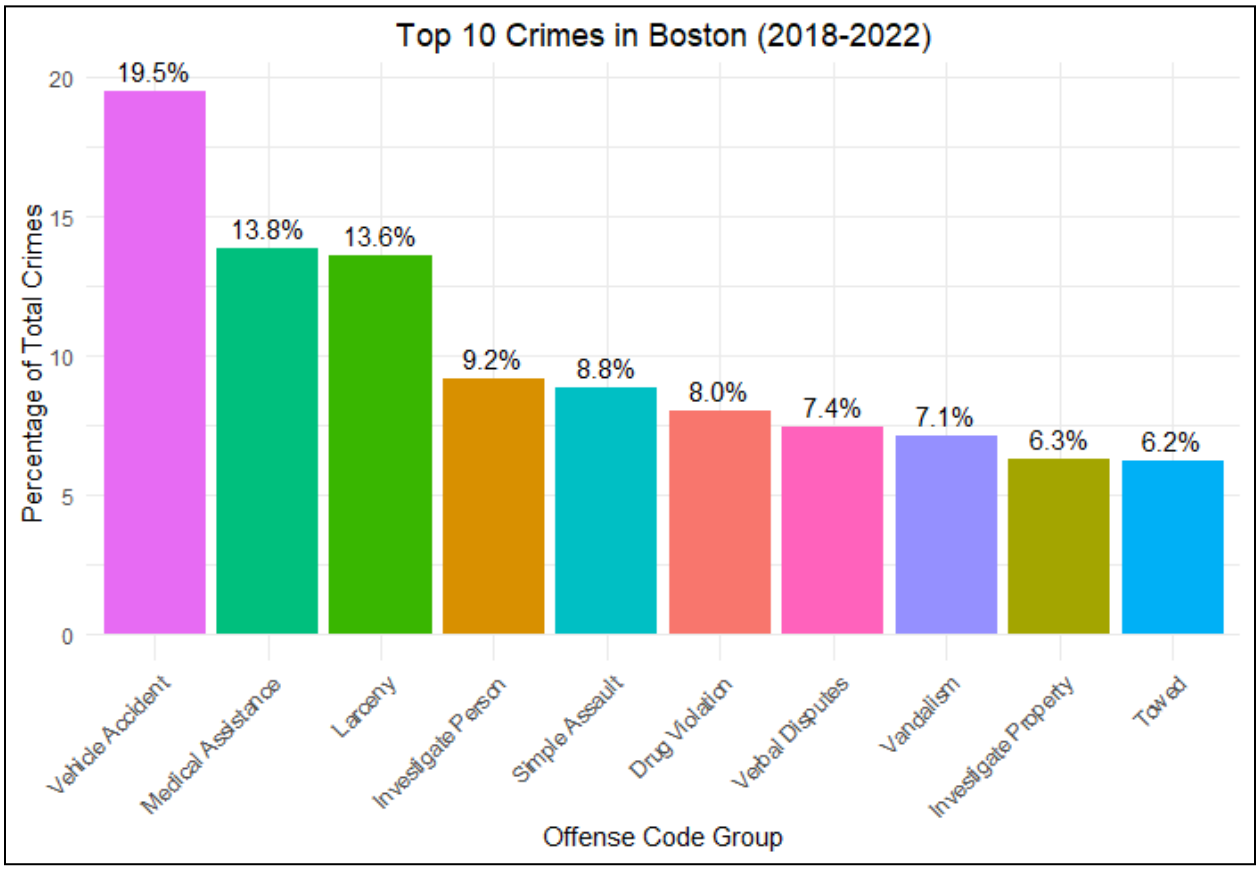
B) Tree-map of Offense Code Groups



Overall Observations:

- Vehicle Accident (11647), Larceny (8116) and Medical Assistance (8257) are the most frequent offense code groups, accounting for a significant portion of the total incidents.
- Drug Violation (4777), Harassment (1315), and Residential Burglary (1306) are relatively less frequent compared to other categories.
- Human Trafficking (1), Explosives (18), Prostitution (31) are very less in compared to other categories.

C) What are the top 10 crimes in Boston?



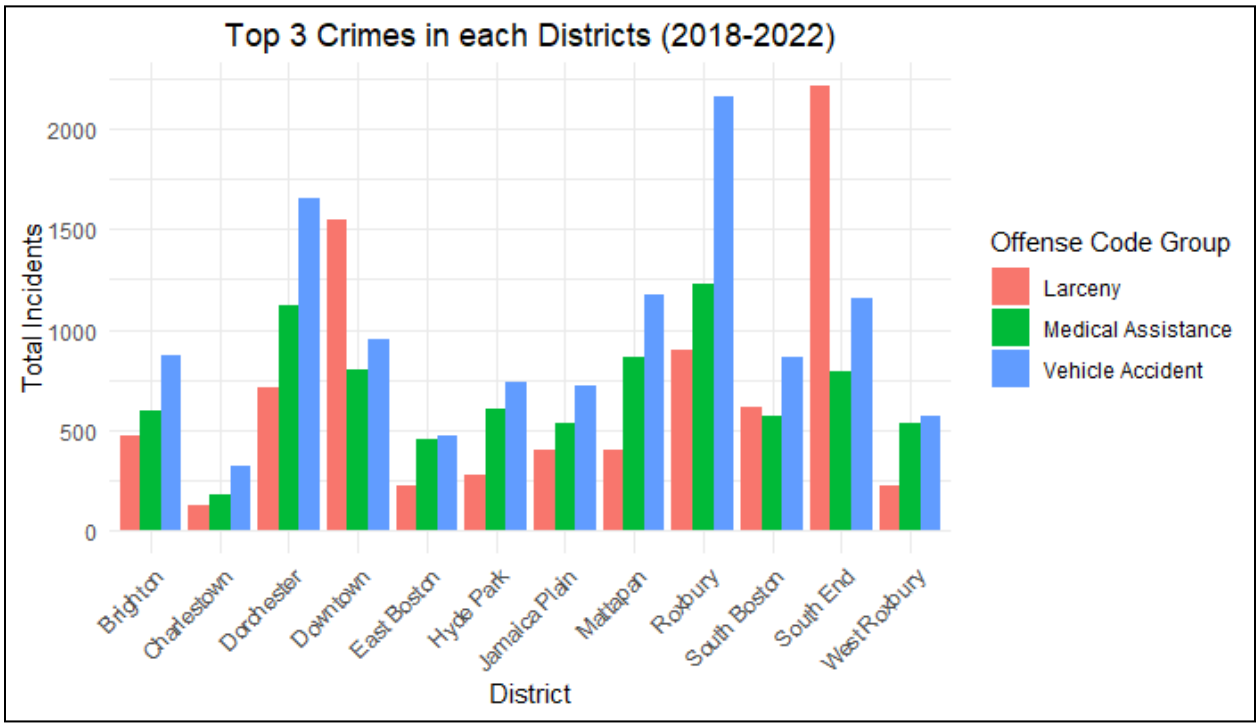
Overall Observations:

- Vehicle Accidents and Medical Assistance are the two most frequent crime categories, accounting for 19.5% and 13.8% of total crimes, respectively.
- Larceny, Investigate Person, and Simple Assault are also relatively common.
- Auto Theft and Violations are among the least frequent categories.

Specific Observations:

- Vehicle Accidents: This category likely includes a wide range of incidents, from minor fender benders to serious collisions.
- Medical Assistance: This category likely includes incidents where police officers were called to provide medical assistance, such as accidents or overdoses.
- Larceny: This category encompasses a wide range of property crimes, including theft, shoplifting, and burglary.
- Investigate Person: This category could include a variety of situations where police officers are investigating individuals, such as suspicious activity or disturbances.
- Simple Assault: This category includes incidents involving physical assault without the use of a weapon.

D) What types of offenses are more prevalent in each district?



Most Common Crimes by District

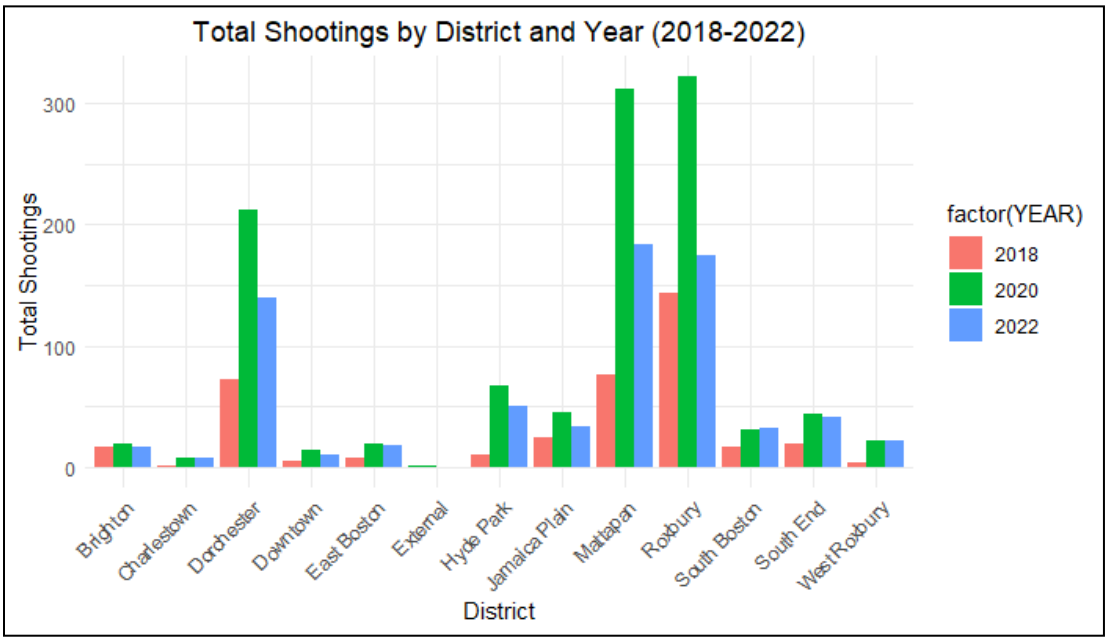
District	Most Common Offense	Number of Incidents
Brighton	Vehicle Accident	876
Charlestown	Vehicle Accident	319
Dorchester	Vehicle Accident	1,652
Downtown	Larceny	1,549
East Boston	Vehicle Accident	471
Hyde Park	Vehicle Accident	738
Jamaica Plain	Vehicle Accident	725
Mattapan	Vehicle Accident	1,175
Roxbury	Vehicle Accident	2,155
South Boston	Vehicle Accident	860
South End	Larceny	2,214
West Roxbury	Vehicle Accident	571

Most Common Crimes by District:

Vehicle Accidents are the most common offense in most districts, reinforcing the observation from the previous chart.
However, there are exceptions. In Downtown, Larceny is the most common offense, suggesting a higher concentration of property crimes in this area.
In South End, Larceny is also the most common offense, indicating a similar trend to Downtown.

Overall, the data suggests that vehicle accidents are a significant issue across Boston, while property crimes and investigations are also prevalent in certain districts.

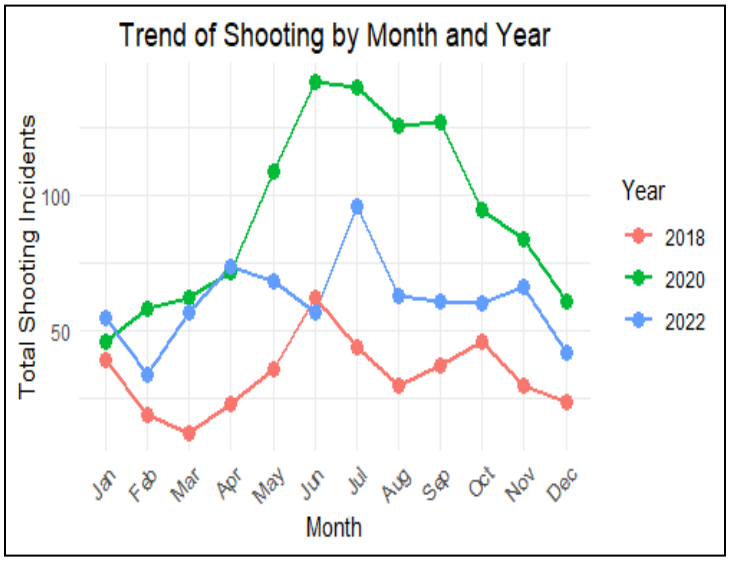
E) Which areas show the highest frequency of shooting?



Overall Trends:

- There was a noticeable increase in shootings in 2020 compared to 2018 and 2022. This could be attributed to various factors, including the COVID-19 pandemic and its associated social and economic disruptions.
- Districts, such as Roxbury and Mattapan, consistently experience higher levels of shootings compared to others. Whereas Downtown, Charlestown and East Boston typically have lower shooting rates compared to other districts.

F) Is there a trend for any particular months of the year where Shooting occurs?



Seasonal Variation: There appears to be a seasonal pattern in shooting incidents, with higher numbers occurring during the warmer months (May-September) and lower numbers during the colder months (November-March).

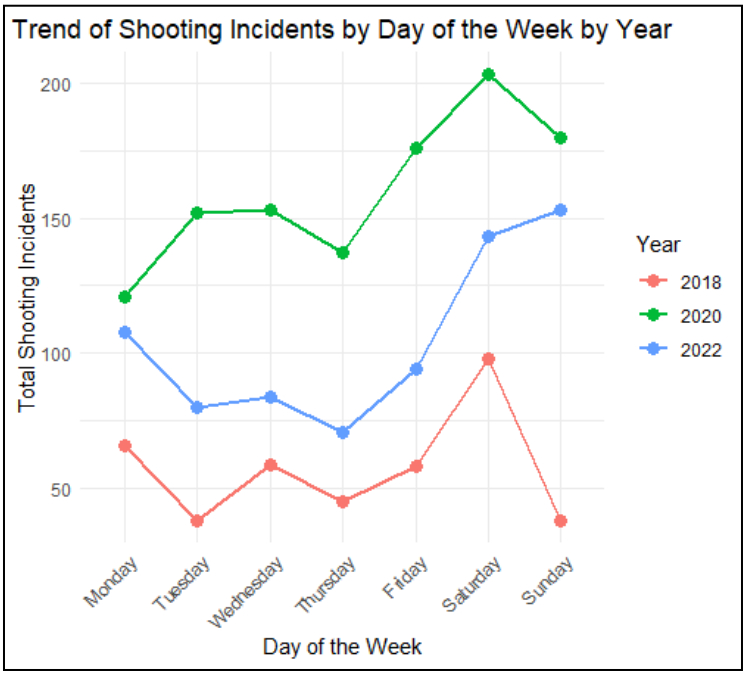
Year-to-Year Fluctuation: The number of shooting incidents fluctuates from year to year, with 2020 showing a significant spike, particularly during the summer months. This could be attributed to various factors, including the COVID-19 pandemic and its associated social and economic disruptions.

G) Is there a trend for any particular Days of the Week where Shooting occurs?

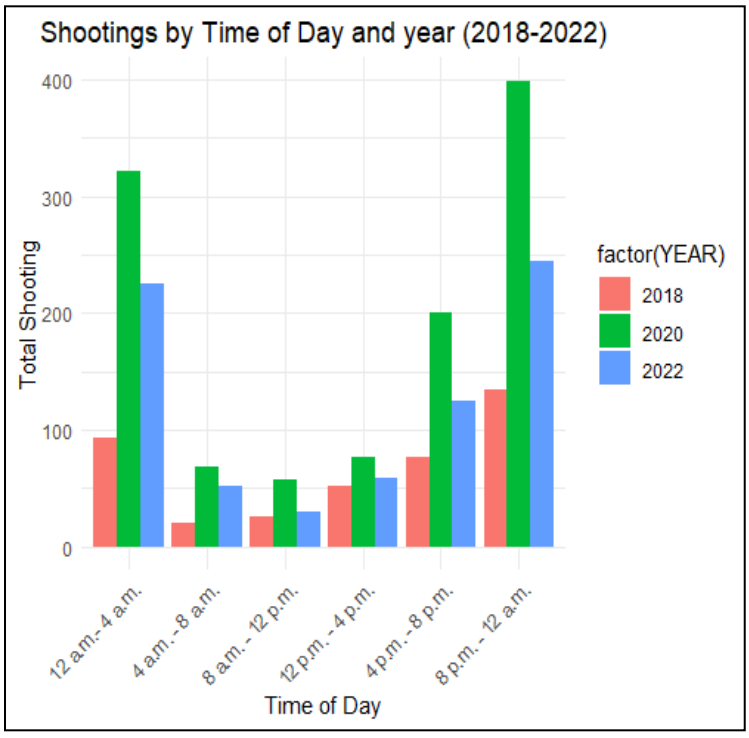
Overall Trends:

Friday is the most dangerous day: In all three years, Friday consistently had the highest number of shooting incidents. This suggests that there might be specific factors, such as social gatherings or other events that contribute to increased violence on Fridays.

Weekends are more dangerous: Both Saturday and Sunday generally have higher shooting incidents compared to weekdays. This could be due to factors like increased alcohol consumption, social gatherings, and relaxed routines.



H) Is there a trend for any particular Time of the Day where Shooting occurs?



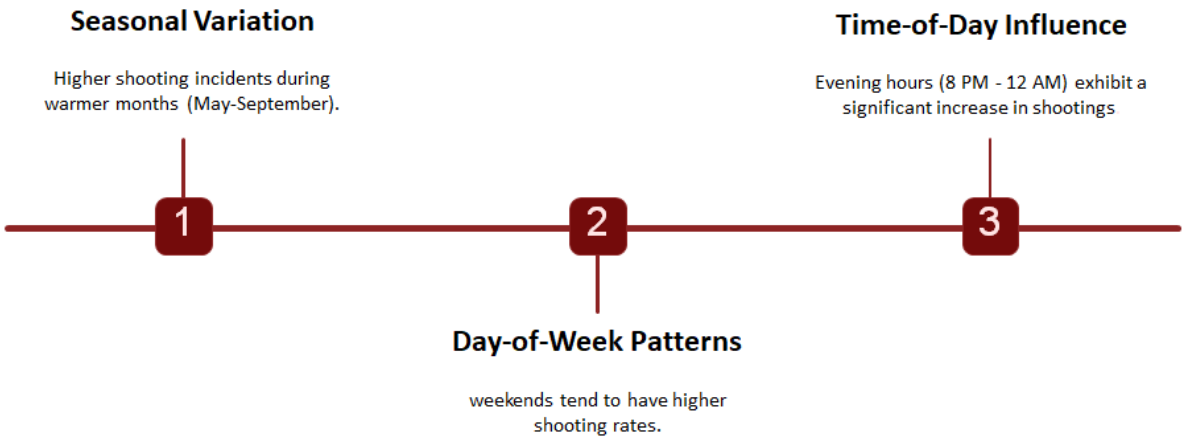
Overall Trends:

Evening Peak: The highest number of shootings consistently occurs between 8 PM and 12 AM, suggesting that evening hours are a critical period for increased gun violence.

Morning Lows: The lowest number of shootings occurs between 4 AM and 8 AM, indicating a significant decrease in violent activity during the early morning hours.

Year-to-Year Fluctuation: The number of shootings fluctuates across different time periods and years. 2020, in particular, saw a significant spike in shootings, especially during the evening hours.

Shooting Trends: Temporal Patterns



Chi-Square Test for Association between District and Shooting:

The test evaluates if the distribution of shootings across districts is independent or if certain districts are more prone to shootings.

Framing Hypothesis:

- H0: There is NO association between the district and the occurrence of shootings (The distribution of shootings is independent of the district).
- H1: There is an association between the district and the occurrence of shootings (The distribution of shootings depends on the district).

Test Statistic (X^2)	DF	P-value	Decision
1258.6	**12**	**< 2.2e-16**	**Reject the null hypothesis**

Decision:

Since the p-value is extremely small ($2.2e{-}16 < 0.05$), we reject the null hypothesis.
This means there is a statistically significant association between districts and shootings.

Interpretation:

The result indicates that shootings are not evenly distributed across districts. Some districts may experience higher or lower frequencies of shootings compared to others. This insight can guide resource allocation and targeted interventions by law enforcement in specific districts.

Logistic Regression Modeling:

Two logistic regression models were developed:

Model-1: A full model using all available predictors

Model-2: A reduced model using selected predictors

Logistic regression was used to predict the likelihood of a shooting incident (SHOOTING) based on factors like DISTRICT, DAY_OF_WEEK, MONTH, HOUR_BINNING, and OFFENSE_CODE_GROUP. This method is appropriate because the outcome variable is binary (shooting occurred or not). The purpose is to identify significant predictors and assess the model's accuracy in classifying incidents, which can support resource allocation and policy decisions.

Comparison of Logistic Regression Models:

Comparison of Logistic Regression Models

Model	Adj. R^2	AIC	BIC
Model-1	0.0766	16,386.77	16,577.66
Model-2	0.1985	14,380.10	15,334.54

Adjusted R-squared: The second model has a higher adjusted R-squared (0.1985) compared to the first model (0.0766). This indicates that the second model explains more of the variance in the data after adjusting for the number of predictors.

AIC and BIC: Both AIC and BIC are lower for the second model. Lower values of these information criteria generally indicate a better-fitting model.

Model Evaluation for Train Set:

Confusion matrices for training sets:

		Actual Values	
		0	**1**
Predicted Values	**0**	True Positive 168982	False Positive 7
	1	False Negative 1512	True Negative 43

Performance matrices for training sets:

- **Accuracy:**
  (TP + TN) / (TP + FP + FN + TN) = 99.10%
- **Precision:**
  TP / (TP + FP) = 99.99%
- **Specificity:**
  TP / (TP + FN) = 99.91%
- **Sensitivity:**
  TN / (TN + FP) = 86%

Model Evaluation for Test Set:

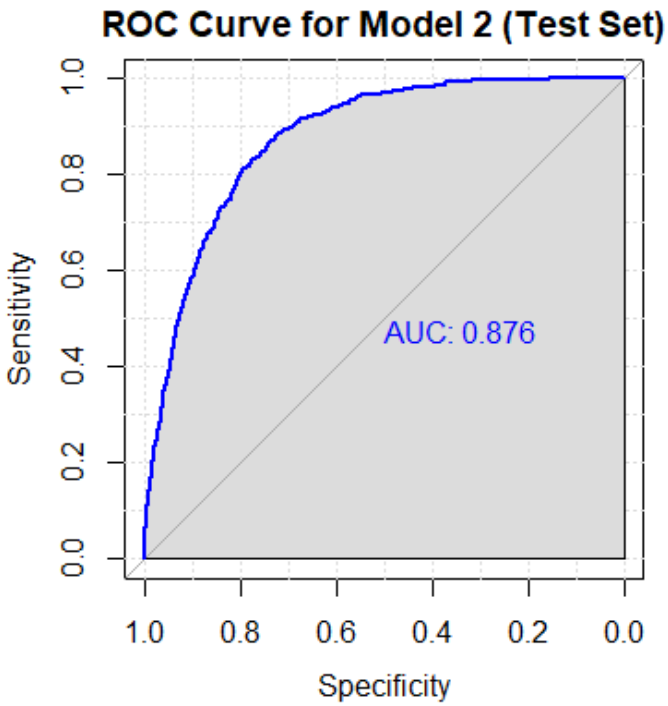
Confusion matrices for test sets:

		Actual Values	
		0	**1**
Predicted Values	**0**	True Positive 72387	False Positive 1
	1	False Negative 686	True Negative 16

Performance matrices for test sets:

- **Accuracy:**
  (TP + TN) / (TP + FP + FN + TN) = 99.06%
- **Precision:**
  TP / (TP + FP) = 99.99%
- **Specificity:**
  TP / (TP + FN) = 99.06%
- **Sensitivity:**
  TN / (TN + FP) = 94.11%

ROC and AUC for testing sets:



Good Discriminative Power:
The ROC curve shows a strong upward trajectory, indicating that the model has good discriminatory power between positive and negative classes.

AUC of 0.876:
This AUC score suggests that the model has a high level of accuracy in distinguishing between positive and negative instances. A value closer to 1 indicates better performance.

Report Summary:

Overall Crime Trends:
Consistent Patterns: Certain districts (Roxbury, Mattapan) consistently have higher crime rates, while others (Charlestown, East Boston) have lower rates.
Seasonal Variations: Crime rates tend to increase during warmer months, particularly in the evening hours.
Day-of-Week Patterns: Fridays and weekends tend to have higher crime rates.

Specific Offense Trends:
Vehicle Accidents and Medical Assistance: These are the most frequent offense categories, highlighting the importance of traffic safety and emergency response.
Larceny and Property Crimes: These offenses are prevalent in certain districts, particularly in urban areas.
Drug Violations and Violent Crimes: While less frequent, these offenses can have significant societal impacts.

Model Performance:
Model 2: Shows strong discriminatory power with an AUC of 0.876, indicating good performance in distinguishing between positive and negative classes.

References:

City of Boston. (n.d.). Crime incident reports: August 2015 to date (source new system). Retrieved November 29, 2024, from https://data.boston.gov/dataset/crime-incident-reports-august-2015-to-date-source-new-system
City of Boston. (n.d.). *City council districts 2023-2032*. Retrieved November 29, 2024, from https://data.boston.gov/dataset/city-council-districts-2023-2032/resource/9255bfcb-857d-4e2a-8040-eac31d106c98
Boston Planning & Development Agency. (n.d.). Neighborhoods. Retrieved November 29, 2024, from http://www.bostonplans.org/neighborhoods
Sjoberg, D. (n.d.). *gtsummary*. Retrieved from https://www.danieldsjoberg.com/gtsummary/
R-Bloggers. (2015, September). How to perform a logistic regression in R. Retrieved from https://www.r-bloggers.com/2015/09/how-to-perform-a-logistic-regression-in-r/
GraphPad. (n.d.). Logistic regression and ROC curves. Retrieved from https://www.graphpad.com/guides/prism/latest/curve-fitting/reg_logistic_roc_curves.htm
GeeksforGeeks. (n.d.). Confusion matrix in R. Retrieved from https://www.geeksforgeeks.org/confusion-matrix-in-r/

```
# Load necessary libraries
library(dplyr)
library(DataExplorer)
library(ggplot2)
library(readr)
library(gtsummary)
library(flextable)
library(gt)

# Read each CSV file into R
data_2018 <- read.csv("Z:/NEU/Intermediate Analytics/Module 2/Final Project - Proposal/crime report (2018-2022)/Crime Incident Reports - 2018.csv")
data_2020 <- read.csv("Z:/NEU/Intermediate Analytics/Module 2/Final Project - Proposal/crime report (2018-2022)/Crime Incident Reports - 2020.csv")
data_2022 <- read.csv("Z:/NEU/Intermediate Analytics/Module 2/Final Project - Proposal/crime report (2018-2022)/Crime Incident Reports - 2022.csv")

#-----------------------------------------------------------------------------------------------------------------------

# Displaying columns with blank ("") values in Plot
blank_percentage <- sapply(data_2018, function(x) sum(x == "", na.rm = TRUE)) / nrow(data_2018) * 100
blank_df <- data.frame(Column = names(blank_percentage), Blank_Percentage = blank_percentage)
blank_df <- subset(blank_df, Blank_Percentage > 0)

ggplot(blank_df, aes(x = reorder(Column, -Blank_Percentage), y = Blank_Percentage)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme_minimal() +
  labs(title = "Percentage of Blank Values in Each Column",
       x = "Columns",
       y = "Percentage of Blank Values") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Convert blank ("") values to NA in the entire dataframe
data_2018[data_2018 == ""] <- NA

data_2018$SHOOTING <- ifelse(data_2018$SHOOTING == "Y", 1, 0)
head(data_2018$SHOOTING)

#-----------------------------------------------------------------------------------------------------------------------

# Combine the datasets into one using rbind
combined_data <- bind_rows(data_2018, data_2020, data_2022)

# Save the combined data to a new CSV file
write.csv(combined_data, "Z:/NEU/Intermediate Analytics/Module 2/Final Project - Proposal/crime report (2018-2022)/Combined_Crime_Incident_Reports_2018_2022.csv", row.names = FALSE)
```

```
# Output a message to confirm the file was saved
cat("Combined CSV file saved successfully.")

#-------------------------------------------------------------------------------------------------------------------

#column names
column_names <- names(combined_data)
print(column_names)

# Create SERIOUS_CRIME variable
combined_data$SERIOUS_CRIME <- as.numeric(combined_data$UCR_PART == "Part One")

#-------------------------------------------------------------------------------------------------------------------

# Data checking

# Displaying columns with blank ("") values in Plot
blank_percentage <- sapply(combined_data, function(x) sum(x == "", na.rm = TRUE)) /
nrow(combined_data) * 100
blank_df <- data.frame(Column = names(blank_percentage), Blank_Percentage = blank_percentage)
blank_df <- subset(blank_df, Blank_Percentage > 0)

ggplot(blank_df, aes(x = reorder(Column, -Blank_Percentage), y = Blank_Percentage)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme_minimal() +
  labs(title = "Percentage of Blank Values in Each Column",
       x = "Columns",
       y = "Percentage of Blank Values") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Convert blank ("") values to NA in the entire dataframe
combined_data[combined_data == ""] <- NA


# View missing data patterns
plot_missing(combined_data)

#-------------------------------------------------------------------------------------------------------------------

# Data cleaning

# Checking for NA values in each column
colSums(is.na(combined_data))

# Remove rows with 50% or more NA values
threshold <- 0.5
combined_data <- combined_data %>%
  filter(rowMeans(is.na(.)) < threshold)
```

```
# Function to calculate the mode
calculate_mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

# Threshold for NA proportion
na_threshold <- 0.3

# Iterate over columns to clean data
df1 <- combined_data %>%
  mutate(across(everything(), ~ {
    # Calculate proportion of missing values
    na_proportion <- sum(is.na(.)) / n()

    if (na_proportion < na_threshold) {
      if (is.numeric(.)) {
        replace(., is.na(.), median(., na.rm = TRUE))
      } else {
        replace(., is.na(.), calculate_mode(na.omit(.)))
      }
    } else {
      . # Leave the column as is if NA proportion >= 30%
    }
  }))

# View missing data patterns
plot_missing(df1)

head(df1$SHOOTING)
# Print the mode of shooting column
print(calculate_mode(df1$SHOOTING))
# Replace NA values in df1$SHOOTING with 0
df1$SHOOTING[is.na(df1$SHOOTING)] <- 0

head(df1$SERIOUS_CRIME)
# Print the mode of SERIOUS_CRIME column
print(calculate_mode(df1$SERIOUS_CRIME))
# Replace NA values in df1$SERIOUS_CRIME with 0
df1$SERIOUS_CRIME[is.na(df1$SERIOUS_CRIME)] <- 0

head(df1$OFFENSE_CODE_GROUP)
print(calculate_mode(df1$OFFENSE_CODE_GROUP))
# Replace NA values in df1$OFFENSE_CODE_GROUP with "Other"
df1$OFFENSE_CODE_GROUP[is.na(df1$OFFENSE_CODE_GROUP)] <- "Other"

head(df1$UCR_PART)
```

```
print(calculate_mode(df1$UCR_PART))
# Replace NA values in df1$UCR_PART with "Other"
df1$UCR_PART[is.na(df1$UCR_PART)] <- "Other"

#-----------------------------------------------------------------------------------------------------------------------

# Create a mapping of district codes to district names
district_mapping <- c(
  "D4" = "South End",
  "A7" = "East Boston",
  "D14" = "Brighton",
  "B3" = "Mattapan",
  "A1" = "Downtown",
  "C6" = "South Boston",
  "A15" = "Charlestown",
  "E5" = "West Roxbury",
  "E18" = "Hyde Park",
  "B2" = "Roxbury",
  "C11" = "Dorchester",
  "E13" = "Jamaica Plain",
  "External" = "External"
)

# Create the new column DISTRICT_NAME
df1$DISTRICT_NAME <- district_mapping[df1$DISTRICT]

head(df1)

# Create a mapping of month numbers to month names
month_mapping <- c(
  "1" = "Jan",
  "2" = "Feb",
  "3" = "Mar",
  "4" = "Apr",
  "5" = "May",
  "6" = "Jun",
  "7" = "Jul",
  "8" = "Aug",
  "9" = "Sep",
  "10" = "Oct",
  "11" = "Nov",
  "12" = "Dec"
)

# Replace month numbers with month names
df1 <- df1 %>%
  mutate(MONTH = case_when(
    as.character(MONTH) %in% names(month_mapping) ~ month_mapping[as.character(MONTH)],
```

```
   is.na(MONTH) ~ "Unknown",
   TRUE ~ as.character(MONTH)
  ))

# Create MONTH_ID column
month_mapping <- c("Jan" = 1, "Feb" = 2, "Mar" = 3, "Apr" = 4, "May" = 5, "Jun" = 6,
                   "Jul" = 7, "Aug" = 8, "Sep" = 9, "Oct" = 10, "Nov" = 11, "Dec" = 12)
df1$MONTH_ID <- month_mapping[df1$MONTH]

# Create DAY_ID column
day_mapping <- c("Monday" = 1, "Tuesday" = 2, "Wednesday" = 3, "Thursday" = 4,
                 "Friday" = 5, "Saturday" = 6, "Sunday" = 7)
df1$DAY_ID <- day_mapping[df1$DAY_OF_WEEK]


# Create the HOUR_BINNING column
df1 <- df1 %>%
  mutate(HOUR_BINNING = cut(HOUR,
                            breaks = c(-1, 4, 8, 12, 16, 20, 24),
                            labels = c("12 a.m.- 4 a.m.", "4 a.m. - 8 a.m.", "8 a.m. - 12 p.m.",
                                       "12 p.m. - 4 p.m.", "4 p.m. - 8 p.m.", "8 p.m. - 12 a.m."),
                            include.lowest = TRUE,
                            right = FALSE))

#-----------------------------------------------------------------------------------------------------------------------

names(df1)

# 1. Create summary table by Year for DISTRICT
district_summary <- df1 %>%
  select(YEAR, DISTRICT_NAME) %>%
  tbl_summary(by = YEAR,
              statistic = list(
                all_continuous() ~ "{mean} ({sd})",
                all_categorical() ~ "{n} ({p}%)"
              ),
              digits = all_continuous() ~ 2) %>%
  add_overall() %>%
  modify_header(label = "**Variable**") %>%
  modify_caption("Table 1. Descriptive Statistics of Boston Crime Incidents for Districts by Year (2018-
2022)")
print(district_summary)

  # Convert gtsummary object to flextable and assign to a new variable
district_summary_flextable <- as_flex_table(district_summary)

  # Add formatting to flextable
district_summary_flextable <- district_summary_flextable %>%
```

```
  fontsize(size = 10, part = "all") %>%
  set_caption("Table 1. Descriptive Statistics of Boston Crime Incidents for Districts by Year (2018-2022)")
print(district_summary_flextable)

  # Export flextable to Word document
save_as_docx(district_summary_flextable, path = "district_summary_by_year.docx")

#-------------------------------------------------------------------------------------------------------------------

# 2. Create summary table by Year for DAY_OF_WEEK
day_of_week_summary <- df1 %>%
  select(YEAR, DAY_OF_WEEK) %>%
  tbl_summary(by = YEAR,
              statistic = list(
                all_continuous() ~ "{mean} ({sd})",
                all_categorical() ~ "{n} ({p}%)"
              ),
              digits = all_continuous() ~ 2) %>%
  add_overall() %>%
  modify_header(label = "**Variable**") %>%
  modify_caption("Table 2. Descriptive Statistics of Boston Crime Incidents for Day of Week by Year
(2018-2022)")
print(day_of_week_summary)

  # Convert gtsummary object to flextable and assign to a new variable
day_of_week_summary_flextable <- as_flex_table(day_of_week_summary)

  # Add formatting to flextable
day_of_week_summary_flextable <- day_of_week_summary_flextable %>%
  fontsize(size = 10, part = "all") %>%
  set_caption("Table 2. Descriptive Statistics of Boston Crime Incidents for Day of Week by Year (2018-
2022)")
print(day_of_week_summary_flextable)

  # Export flextable to Word document
save_as_docx(day_of_week_summary_flextable, path = "day_of_week_summary_by_year.docx")

#-------------------------------------------------------------------------------------------------------------------

# 3. Create summary table by Year for UCR_PART
ucr_part_summary <- df1 %>%
  select(YEAR, UCR_PART) %>%
  tbl_summary(by = YEAR,
              statistic = list(
                all_continuous() ~ "{mean} ({sd})",
                all_categorical() ~ "{n} ({p}%)"
              ),
              digits = all_continuous() ~ 2) %>%
```

```
  add_overall() %>%
  modify_header(label = "**Variable**") %>%
  modify_caption("Table 3. Descriptive Statistics of Boston Crime Incidents for UCR Part by Year (2018-
2022)")
print(ucr_part_summary)

# Convert gtsummary object to flextable and assign to a new variable
ucr_part_summary_flextable <- as_flex_table(ucr_part_summary)

# Add formatting to flextable
ucr_part_summary_flextable <- ucr_part_summary_flextable %>%
  fontsize(size = 10, part = "all") %>%
  set_caption("Table 3. Descriptive Statistics of Boston Crime Incidents for UCR Part by Year (2018-
2022)")
print(ucr_part_summary_flextable)

# Export flextable to Word document
save_as_docx(ucr_part_summary_flextable, path = "ucr_part_summary_by_year.docx")

#-------------------------------------------------------------------------------------------------------------------------

# 4. Create summary table by Year for MONTH
month_summary <- df1 %>%
  select(YEAR, MONTH) %>%
  tbl_summary(by = YEAR,
              statistic = list(
                all_continuous() ~ "{mean} ({sd})",
                all_categorical() ~ "{n} ({p}%)"
              ),
              digits = all_continuous() ~ 2) %>%
  add_overall() %>%
  modify_header(label = "**Variable**") %>%
  modify_caption("Table 4. Descriptive Statistics of Boston Crime Incidents for Months by Year (2018-
2022)")
print(month_summary)

# Convert gtsummary object to flextable and assign to a new variable
month_summary_flextable <- as_flex_table(month_summary)

# Add formatting to flextable
month_summary_flextable <- month_summary_flextable %>%
  fontsize(size = 10, part = "all") %>%
  set_caption("Table 4. Descriptive Statistics of Boston Crime Incidents for Months by Year (2018-2022)")
print(month_summary_flextable)

# Export flextable to Word document
save_as_docx(month_summary_flextable, path = "month_summary_by_year.docx")
```

```
#-----------------------------------------------------------------------------------------------------------------------

# 5. Create summary table by Year for Hour
hour_summary <- df1 %>%
  select(YEAR, HOUR_BINNING) %>%
  tbl_summary(by = YEAR,
              statistic = list(
                all_continuous() ~ "{mean} ({sd})",
                all_categorical() ~ "{n} ({p}%)"
              ),
              digits = all_continuous() ~ 2) %>%
  add_overall() %>%
  modify_header(label = "**Variable**") %>%
  modify_caption("Table 5. Descriptive Statistics of Boston Crime Incidents for Hours by Year (2018-
2022)")

# Convert gtsummary object to flextable and assign to a new variable
hour_summary_flextable <- as_flex_table(hour_summary)

# Add formatting to flextable
hour_summary_flextable <- hour_summary_flextable %>%
  fontsize(size = 10, part = "all") %>%
  set_caption("Table 5. Descriptive Statistics of Boston Crime Incidents for Hours by Year (2018-2022)")
print(hour_summary_flextable)

# Export flextable to Word document
save_as_docx(hour_summary_flextable, path = "hour_summary_by_year.docx")

#-----------------------------------------------------------------------------------------------------------------------

library(sf) # Library For handling spatial data

# Load Boston district shapefile (replace with actual file path)
boston_map <- st_read("Z:/NEU/Intermediate Analytics/Module 4/Final Project - milestone 2/Boston
Map shape file/police_districts/Police_Districts.shp")

# Prepare incident data by district
district_incidents <- df1 %>%
  group_by(DISTRICT) %>%
  summarise(Total_Incidents = n()) %>%
  mutate(
    Incident_Percentage = (Total_Incidents / sum(Total_Incidents)) * 100,
    Color_Code = case_when(
      Incident_Percentage <= 5 ~ "Green",
      Incident_Percentage > 5  & Incident_Percentage <= 10 ~ "Yellow",
      Incident_Percentage > 10 ~ "Red"
    )
  )
```

```
district_incidents

# Join the incident data with the Boston map
boston_map_data <- boston_map %>%
  left_join(district_incidents, by = c("DISTRICT" = "DISTRICT"))

# Ensure correct factor ordering for Color_Code
boston_map_data$Color_Code <- factor(boston_map_data$Color_Code,
                                     levels = c("Green", "Yellow", "Red"))

# Plot the Boston map with corrected color levels
ggplot(boston_map_data) +
  geom_sf(aes(fill = Color_Code), color = "black", size = 0.2) +
  geom_sf_text(aes(label = ID), color = "black", size = 3) +  # Add district labels
  scale_fill_manual(
    values = c("Green" = "green", "Yellow" = "yellow", "Red" = "red"),
    name = "Incident Levels",
    labels = c("Less Crime", "Moderate Crime", "High Crime")
  ) +
  labs(
    title = "Boston Crime Incidents by District",
    x = "Longitude",
    y = "Latitude",
    subtitle = "Color-coded based on total incidents",
    caption = "Data source: Boston Police Department"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5),  # Center-align title
    plot.subtitle = element_text(hjust = 0.5),  # Center-align subtitle
    plot.caption = element_text(hjust = 0.5),  # Center-align caption
    legend.position = "right"  # Move legend to the right
  )

#-------------------------------------------------------------------------------------------------------------------------

# Prepare shooting data by district
district_shootings <- df1 %>%
  group_by(DISTRICT) %>%
  summarise(Total_Shootings = sum(SHOOTING)) %>%
  mutate(
    Shooting_Percentage = (Total_Shootings / sum(Total_Shootings)) * 100,
    Color_Code = case_when(
      Shooting_Percentage <= 5 ~ "Green",
      Shooting_Percentage > 5 & Shooting_Percentage <= 10 ~ "Yellow",
      Shooting_Percentage > 10 ~ "Red"
    )
```

```
  )
district_shootings

# Join the shooting data with the Boston map
boston_map_data <- boston_map %>%
  left_join(district_shootings, by = c("DISTRICT" = "DISTRICT"))

# Ensure correct factor ordering for Color_Code
boston_map_data$Color_Code <- factor(boston_map_data$Color_Code,
                                     levels = c("Green", "Yellow", "Red"))

# Plot the Boston map with shooting data
ggplot(boston_map_data) +
  geom_sf(aes(fill = Color_Code), color = "black", size = 0.2) +
  geom_sf_text(aes(label = ID), color = "black", size = 3) +  # Add district labels
  scale_fill_manual(
    values = c("Green" = "green", "Yellow" = "yellow", "Red" = "red"),
    name = "Shooting Levels",
    labels = c("Less Shootings", "Moderate Shootings", "High Shootings")
  ) +
  labs(
    title = "Boston Shootings by District",
    x = "Longitude",
    y = "Latitude",
    subtitle = "Color-coded based on total shootings",
    caption = "Data source: Boston Police Department"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    plot.caption = element_text(hjust = 0.5),
    legend.position = "right"
  )

#-------------------------------------------------------------------------------------------------------------------

# Question 1: What are the top 10 crimes in Boston?

# create Treemap of Offense Code Groups
library(treemapify)

# Filter out 'Other' from OFFENSE_CODE_GROUP
crime_filtered <- df1 %>%
  filter(OFFENSE_CODE_GROUP != "Other")

# Rename the specific value in the OFFENSE_CODE_GROUP column
```

```
crime_filtered <- crime_filtered %>%
  mutate(OFFENSE_CODE_GROUP = ifelse(OFFENSE_CODE_GROUP == "Motor Vehicle Accident
Response",
                                     "Vehicle Accident",
                                     OFFENSE_CODE_GROUP))
# Prepare data for treemap
offense_group_summary <- crime_filtered %>%
  group_by(OFFENSE_CODE_GROUP) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
offense_group_summary

# Create treemap
ggplot(offense_group_summary, aes(area = count, fill = OFFENSE_CODE_GROUP, label =
paste(OFFENSE_CODE_GROUP, count, sep = "\n"))) +
  geom_treemap() +
  geom_treemap_text(fontface = "plain", colour = "black", place = "centre", grow = FALSE) +
  labs(title = "Treemap of Offense Code Groups") +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5)  # Center-align title
  )

# Find top 10 crimes
top_10_crimes <- crime_filtered %>%
  group_by(OFFENSE_CODE_GROUP) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  top_n(10, count) %>%
  mutate(percentage = count / sum(count) * 100)
top_10_crimes

# Plot top 10 crimes with colors based on OFFENSE_CODE_GROUP
ggplot(top_10_crimes, aes(x = reorder(OFFENSE_CODE_GROUP, -count), y = percentage, fill =
OFFENSE_CODE_GROUP)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = sprintf("%.1f%%", percentage)), vjust = -0.5) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none", # Hides the legend for a cleaner look (optional)
    plot.title = element_text(hjust = 0.5)  # Center-align title
  ) +
  labs(
    title = "Top 10 Crimes in Boston (2018-2022)",
    x = "Offense Code Group",
    y = "Percentage of Total Crimes"
  )
```

```
#-----------------------------------------------------------------------------------------------------------------------

# Question 2: What types of offenses are more prevalent in each district?

# Aggregate each district's most common offenses
district_crime <- crime_filtered %>%
  group_by(DISTRICT_NAME, OFFENSE_CODE_GROUP) %>%
  summarise(counts = n()) %>%
  arrange(DISTRICT_NAME, desc(counts))

# For each district, get the offense with the maximum count
most_common_crime_by_district <- district_crime %>%
  group_by(DISTRICT_NAME) %>%
  slice_max(order_by = counts, n = 1)
print(most_common_crime_by_district)

# Create a flextable from the summary
most_common_crime_table <- flextable(most_common_crime_by_district) %>%
  set_header_labels(
    DISTRICT_NAME = "District",
    OFFENSE_CODE_GROUP = "Most Common Offense",
    counts = "Number of Incidents"
  ) %>%
  theme_vanilla() %>%  # Apply a clean table style
  autofit() %>%        # Adjust column widths to fit content
  add_header_lines("Most Common Crimes by District") %>%
  align(i = 1, part = "header", align = "center")  # Center-align the title


# Print the flextable
most_common_crime_table

# Find top 3 crimes
top_3_crimes <- crime_filtered %>%
  group_by(OFFENSE_CODE_GROUP) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  top_n(3, count) %>%
  mutate(percentage = count / sum(count) * 100)
top_3_crimes

# Filter data for top 3 crimes in each district
top_3_crimes_in_districts <- crime_filtered %>%
  filter(OFFENSE_CODE_GROUP %in% top_3_crimes$OFFENSE_CODE_GROUP) %>%
  group_by(DISTRICT_NAME, OFFENSE_CODE_GROUP) %>%
  summarise(count = n(), .groups = "drop")

# Plot the data
```

```
ggplot(top_3_crimes_in_districts, aes(x = DISTRICT_NAME, y = count, fill = OFFENSE_CODE_GROUP)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Top 3 Crimes in each Districts (2018-2022)",
    x = "District",
    y = "Total Incidents",
    fill = "Offense Code Group"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)  # Center-align title
  )

#-----------------------------------------------------------------------------------------------------------------------

# Question 3: Is there a trend in total incident/shooting depending on a day of the week?

df1$DAY_OF_WEEK <- factor(
  df1$DAY_OF_WEEK,
  levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
)

# Summarize incidents by DAY_OF_WEEK and YEAR
incidents_by_day <- df1 %>%
  group_by(DAY_OF_WEEK, YEAR) %>%
  summarise(Total_Incidents = n(), .groups = "drop") %>%
  arrange(DAY_OF_WEEK, YEAR)

# Create the line plot
ggplot(incidents_by_day, aes(x = DAY_OF_WEEK, y = Total_Incidents, group = YEAR, color =
factor(YEAR))) +
  geom_line(size = 1) +
  labs(
    title = "Trend of Incidents by Day of the Week by Year",
    x = "Day of the Week",
    y = "Total Incidents",
    color = "Year"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)  # Center-align title
  )

# Summarize shooting incidents by DAY_OF_WEEK and YEAR
shootings_by_day <- df1 %>%
  filter(SHOOTING == 1) %>%  # Filter for shooting incidents
```

```
  group_by(DAY_OF_WEEK, YEAR) %>%
  summarise(Total_Shootings = n(), .groups = "drop") %>%
  arrange(DAY_OF_WEEK, YEAR)

# Create the line plot for shootings
ggplot(shootings_by_day, aes(x = DAY_OF_WEEK, y = Total_Shootings, group = YEAR, color =
factor(YEAR))) +
  geom_line(size = 1) +
  geom_point(size = 3) +  # Add points for better visibility
  labs(
    title = "Trend of Shooting Incidents by Day of the Week by Year",
    x = "Day of the Week",
    y = "Total Shooting Incidents",
    color = "Year"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)  # Center-align title
  ) +
  scale_y_continuous(breaks = scales::pretty_breaks())  # Ensure appropriate y-axis breaks

#-----------------------------------------------------------------------------------------------------------------------

# Question 4: Which areas show the highest frequency of shooting?

# Subset analysis by district
district_analysis <- df1 %>%
  group_by(YEAR, DISTRICT_NAME) %>%
  summarise(
    Total_Incidents = n(),
    Shootings = sum(SHOOTING, na.rm = TRUE)
  ) %>%
  arrange(YEAR, desc(Total_Incidents))
print(district_analysis)

# Visualization: Shootings by year and district
ggplot(district_analysis, aes(x = DISTRICT_NAME, y = Shootings, fill = factor(YEAR))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Total Shootings by District and Year (2018-2022)", x = "District", y = "Total Shootings") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5)  # Center-align title
  )

#-----------------------------------------------------------------------------------------------------------------------

# Question 5: How does time/hour of day affect the type of crimes reported?
```

```r
# Summarize incidents by HOUR and YEAR
crime_by_hour <- df1 %>%
  group_by(HOUR, YEAR) %>%
  summarise(Total_Incidents = n(), .groups = "drop") %>%
  arrange(HOUR, YEAR)

# Create the line plot
ggplot(crime_by_hour, aes(x = HOUR, y = Total_Incidents, group = YEAR, color = factor(YEAR))) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(
    title = "Total Incidents by Hour of the Day by Year (2018-2022)",
    x = "Hour of the Day",
    y = "Total Incidents",
    color = "Year"
  ) +
  scale_x_continuous(
    breaks = seq(0, 24, by = 2),  # Set x-axis breaks from 0 to 24 with steps of 2
    limits = c(0, 24)           # Ensure the axis spans from 0 to 24
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)  # Center-align title
  )

# Subset analysis by time of day
time_analysis <- df1 %>%
  group_by(YEAR, HOUR_BINNING) %>%
  summarise(
    Total_Incidents = n(),
    Shootings = sum(SHOOTING, na.rm = TRUE)
  ) %>%
  arrange(YEAR, desc(Total_Incidents))
print(time_analysis)

# Visualization: Shootings by time of day
ggplot(time_analysis, aes(x = HOUR_BINNING, y = Shootings, fill = factor(YEAR))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Shootings by Time of Day and year (2018-2022)", x = "Time of Day", y = "Total Shooting") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5)  # Center-align title
  )

#-----------------------------------------------------------------------------------------------------------------------
```

```
# Question 6: Is there a trend for any particular months of the year where crimes occur?

# Summarize incidents by MONTH and YEAR
incidents_by_month <- df1 %>%
  group_by(MONTH, YEAR) %>%
  summarise(Total_Incidents = n(), .groups = "drop") %>%
  arrange(MONTH, YEAR)

# Create the line plot
ggplot(incidents_by_month, aes(x = MONTH, y = Total_Incidents, group = YEAR, color = factor(YEAR))) +
  geom_line(size = 1) +
  geom_point(size = 2)+
    labs(
    title = "Trend of Incidents by Month by Year",
    x = "Month",
    y = "Total Incidents",
    color = "Year"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)  # Center-align title
  )

# Summarize shooting incidents by MONTH and YEAR
shootings_by_month <- df1 %>%
  filter(SHOOTING == 1) %>%  # Filter for shooting incidents
  group_by(MONTH, YEAR) %>%
  summarise(Total_Shootings = n(), .groups = "drop") %>%
  arrange(MONTH, YEAR)

# Create the line plot for shootings
ggplot(shootings_by_month, aes(x = MONTH, y = Total_Shootings, group = YEAR, color = factor(YEAR)))
+
  geom_line(size = 1) +
  geom_point(size = 3) +  # Increased point size for better visibility
  labs(
    title = "Trend of Shooting by Month and Year",
    x = "Month",
    y = "Total Shooting Incidents",
    color = "Year"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)  # Center-align title
  ) +
  scale_x_discrete(limits = month.abb) +  # Use month abbreviations on x-axis
```

```
  scale_y_continuous(breaks = scales::pretty_breaks())  # Ensure appropriate y-axis breaks

#-----------------------------------------------------------------------------------------------------------------------

# Chi-Square Test: Association between DISTRICT and OFFENSE_CODE_GROUP
chi_square_test <- chisq.test(table(df1$DISTRICT_NAME, df1$OFFENSE_CODE_GROUP))
print(chi_square_test)

# Chi-square test for association between district and shootings
chi_square_test <- chisq.test(table(df1$DISTRICT, df1$SHOOTING))
print(chi_square_test)

# ANOVA: Compare mean differences in HOUR across different DISTRICTS
anova_model <- aov(HOUR ~ DISTRICT_NAME, data = df1)
summary(anova_model)

#-----------------------------------------------------------------------------------------------------------------------

# Load necessary libraries
library(corrplot)

# Select numeric columns for correlation analysis
numeric_columns <- df1 %>%
  select_if(is.numeric)  # Select only numeric columns

# Compute correlation matrix
correlation_matrix <- cor(numeric_columns, use = "complete.obs")

# View the correlation matrix
print("Correlation Matrix:")
print(correlation_matrix)

# Visualize the correlation matrix using corrplot
corrplot(correlation_matrix, method = "color",
         col = colorRampPalette(c("red", "white", "green"))(200),
         addCoef.col = "black", number.cex = 0.8, tl.col = "black", tl.cex = 0.8)

#-----------------------------------------------------------------------------------------------------------------------

# Load required libraries
library(caret)
library(pROC)
library(broom)
library(MASS)

# Split the data into training and testing sets
set.seed(123)
train_index <- createDataPartition(df1$SHOOTING, p = 0.7, list = FALSE)
```

```
train_data <- df1[train_index, ]
test_data <- df1[-train_index, ]

head(df1)
names(df1)

# Model-1: Full model with all predictors
model1 <- glm(SHOOTING ~ DISTRICT + REPORTING_AREA + YEAR + HOUR + OFFENSE_CODE + DAY_ID +
MONTH_ID,
              data = train_data, family = "binomial")
summary(model1)

# Model-2: Reduced model with selected predictors
model2 <- glm(SHOOTING ~ DISTRICT + DAY_OF_WEEK + MONTH + HOUR_BINNING +
OFFENSE_CODE_GROUP,
              data = train_data, family = "binomial")
summary(model2)

#-----------------------------------------------------------------------------------------------------------------------

# Compare models using flextable:

# Function to calculate adjusted R-squared for logistic regression
logistic_pseudo_r2 <- function(model) {
  1 - model$deviance / model$null.deviance
}

# Function to get model statistics
get_model_stats <- function(model) {
  glance_data <- glance(model)
  tibble(
    Model = deparse(substitute(model)),
    `Adj R-squared` = round(logistic_pseudo_r2(model), 4),
    AIC = round(glance_data$AIC, 2),
    BIC = round(glance_data$BIC, 2)
  )
}

# Combine statistics for both models
model_comparison <- bind_rows(
  Model1 = get_model_stats(model1),
  Model2 = get_model_stats(model2)
)

# Create flextable
comparison_table <- flextable(model_comparison) %>%
  theme_vanilla() %>%
  autofit() %>%
```

```
  bold(part = "header") %>%
  set_caption("Comparison of Logistic Regression Models")

# Print the table
comparison_table

# Export flextable to Word document
save_as_docx(comparison_table, path = "Comparison of Logistic Regression Models.docx")

#-------------------------------------------------------------------------------------------------------------------

# Function to create confusion matrix for Model-2
create_confusion_matrix <- function(model, data) {
  predictions <- predict(model, newdata = data, type = "response")
  predicted_classes <- ifelse(predictions > 0.5, 1, 0)
  conf_matrix <- table(Actual = data$SHOOTING, Predicted = predicted_classes)
  return(conf_matrix)
}

# Confusion matrices for training set
train_cm_model2 <- create_confusion_matrix(model2, train_data)

# Print confusion matrices (Training Set)
print("Confusion Matrix for Model 2 (Training Set):")
print(train_cm_model2)

# Confusion matrices for test set
test_cm_model2 <- create_confusion_matrix(model2, test_data)

# Print confusion matrices (Test Set)
print("Confusion Matrix for Model 2 (Test Set):")
print(test_cm_model2)

#-------------------------------------------------------------------------------------------------------------------

# Make predictions on the test set
test_pred <- predict(model2, newdata = test_data, type = "response")

# Create ROC curve
roc_obj <- roc(test_data$SHOOTING, test_pred)

# Plot ROC curve
plot(roc_obj, main = "ROC Curve for Model 2 (Test Set)",
     col = "blue", lwd = 2,
     print.auc = TRUE, auc.polygon = TRUE, grid = TRUE)
```