Week 3 Assignment: NYPD Shooting Incident Analysis

2024-05-28

Description:

Below is a high level data exploration and analysis using the NYPD Shooting Incident Historical dataset.

Data Source: Historic NYPD Shooting Incident was downloaded from: https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic

Description of Dataset: List of every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year.

This is a breakdown of every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year. This data is manually extracted every quarter and reviewed by the Office of Management Analysis and Planning before being posted on the NYPD website. Each record represents a shooting incident in NYC and includes information about the event, the location and time of occurrence. In addition, information related to suspect and victim demographics is also included.

*more information is available in the URL provided above.

Some of the questions I wish to answer are:

- Which Borough has the most incidents?
- Which Jurisdiction handles the most incidents?
- Can we tell which age group perpetrators belong to?
- Do the incidents occur mostly inside or outside?
- From the data available, can we predict the number of shooting incidents for the next year?

Some filtering and data processing done: NA's were excluded/filtered out from the counts

Potential Biases in the data: As I was working on this analysis, it is important for me to be mindful that there are potential biases in the data, and that possible recording/reporting and demographic biases are likely.

- Recording and reporting bias - are all incidents reported and recorded? Unreported or unrecorded incidents could skew the counts.
- Demographic bias - are there certain racial or age groups more likely to be stopped / arrested?

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(readr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tidyr)

# Install library for timeseries forecast used for modeling
if (!requireNamespace("forecast", quietly = TRUE)) {
  install.packages("forecast")
}
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(forecast)
```

```
filename <- "../data/NYPD_Shooting_Incident_Data__Historic_.csv"
nypd <- read_csv(filename)
```

```
## Rows: 28562 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (12): OCCUR_DATE, BORO, LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION...
## dbl   (7): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD...
## lgl   (1): STATISTICAL_MURDER_FLAG
## time  (1): OCCUR_TIME
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

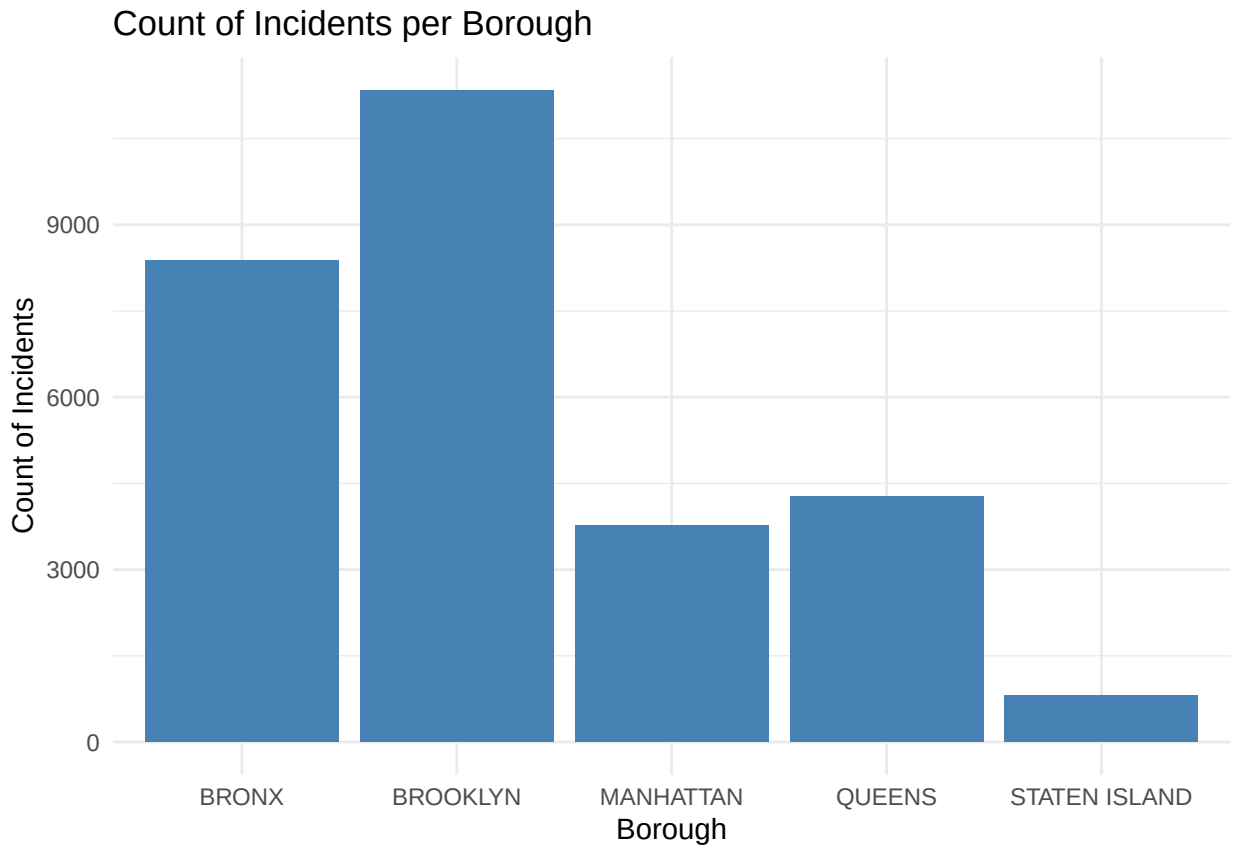
```
boro_counts <- nypd %>%
  group_by(BORO) %>%
  summarise(count = n())
```

```r
# Calculate the count of incidents per jurisdiction code
jurisdiction_counts <- nypd %>%
  group_by(JURISDICTION_CODE) %>%
  summarise(count = n())
```

Which Borough has the most incidents?:

- From the plot below, we can see the Brooklyn has the most number of incidents

```r
ggplot(boro_counts, aes(x = BORO, y = count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme_minimal() +
  labs(title = "Count of Incidents per Borough",
       x = "Borough",
       y = "Count of Incidents")
```

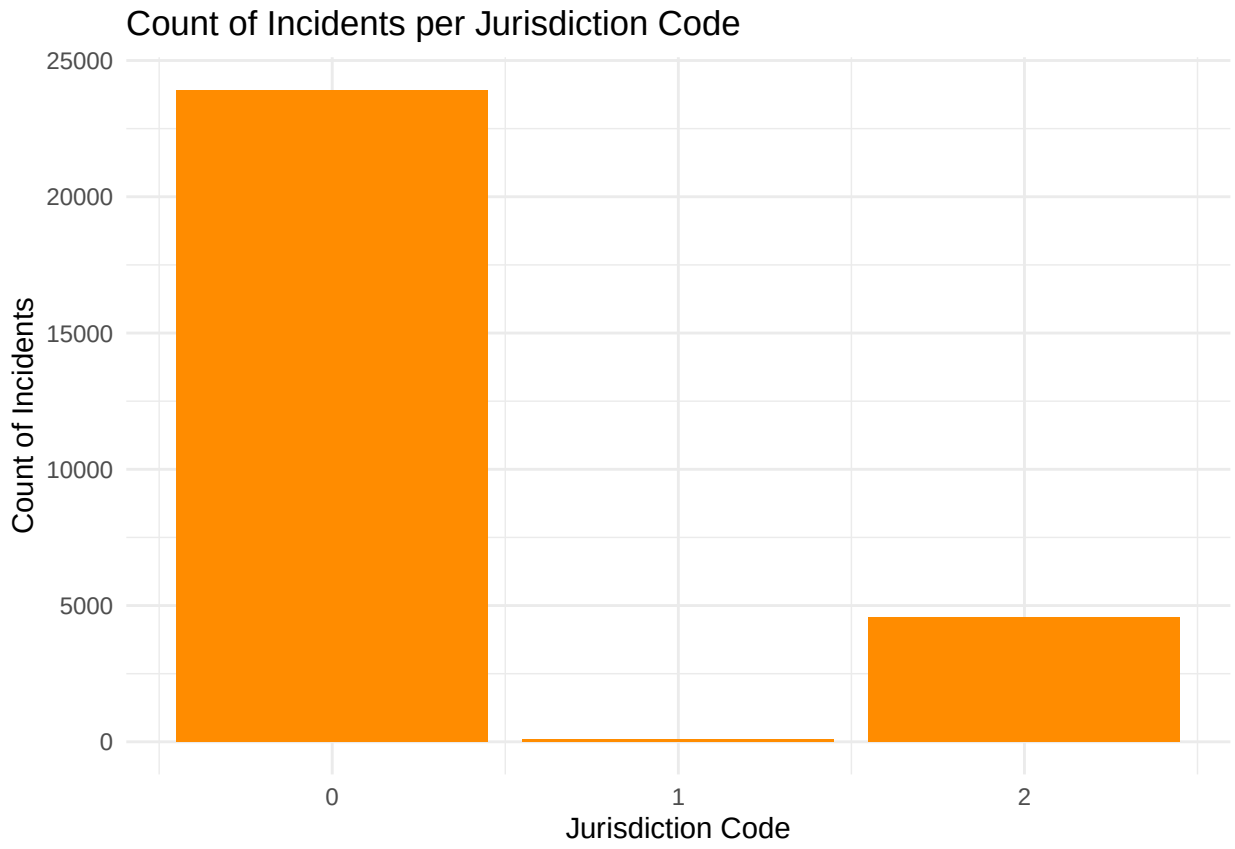


Which Jurisdiction handles the most incidents?:

- The Jurisdiction corresponding to Jurisdiction 0 appears to handle the most incidents.

```r
ggplot(jurisdiction_counts, aes(x = JURISDICTION_CODE, y = count)) +
  geom_bar(stat = "identity", fill = "darkorange") +
  theme_minimal() +
  labs(title = "Count of Incidents per Jurisdiction Code",
       x = "Jurisdiction Code",
       y = "Count of Incidents")
```

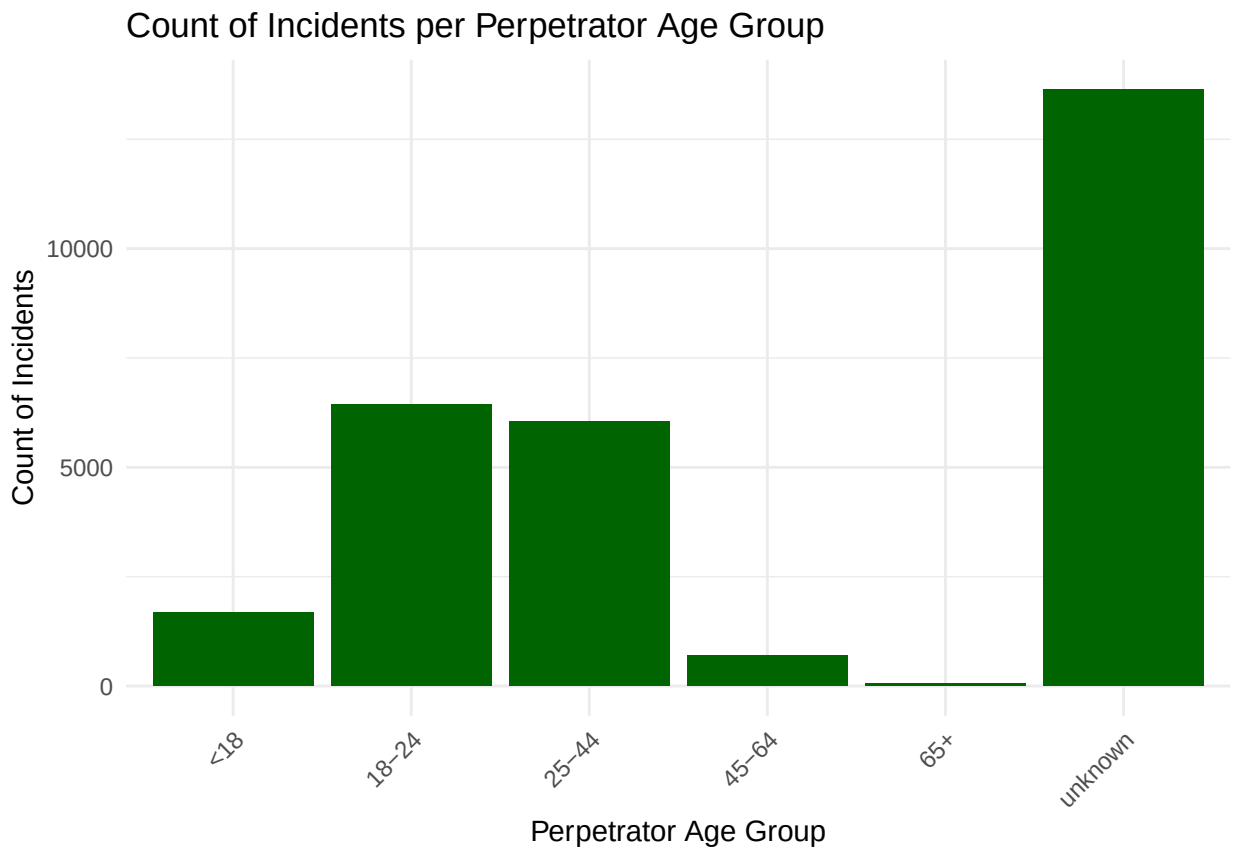
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_bar()`).
```



Can we tell which age group perpetrators belong to?

We have a large number of incidents with 'UNKNOWN', we cannot tell with absolute certainty which age group most perpetrators belong to. It is surprising in either case that there are a large number of perpetrators in the younger age bracket: 18-24, even in the <18 bracket. I am curious how they get access to guns/firearms.

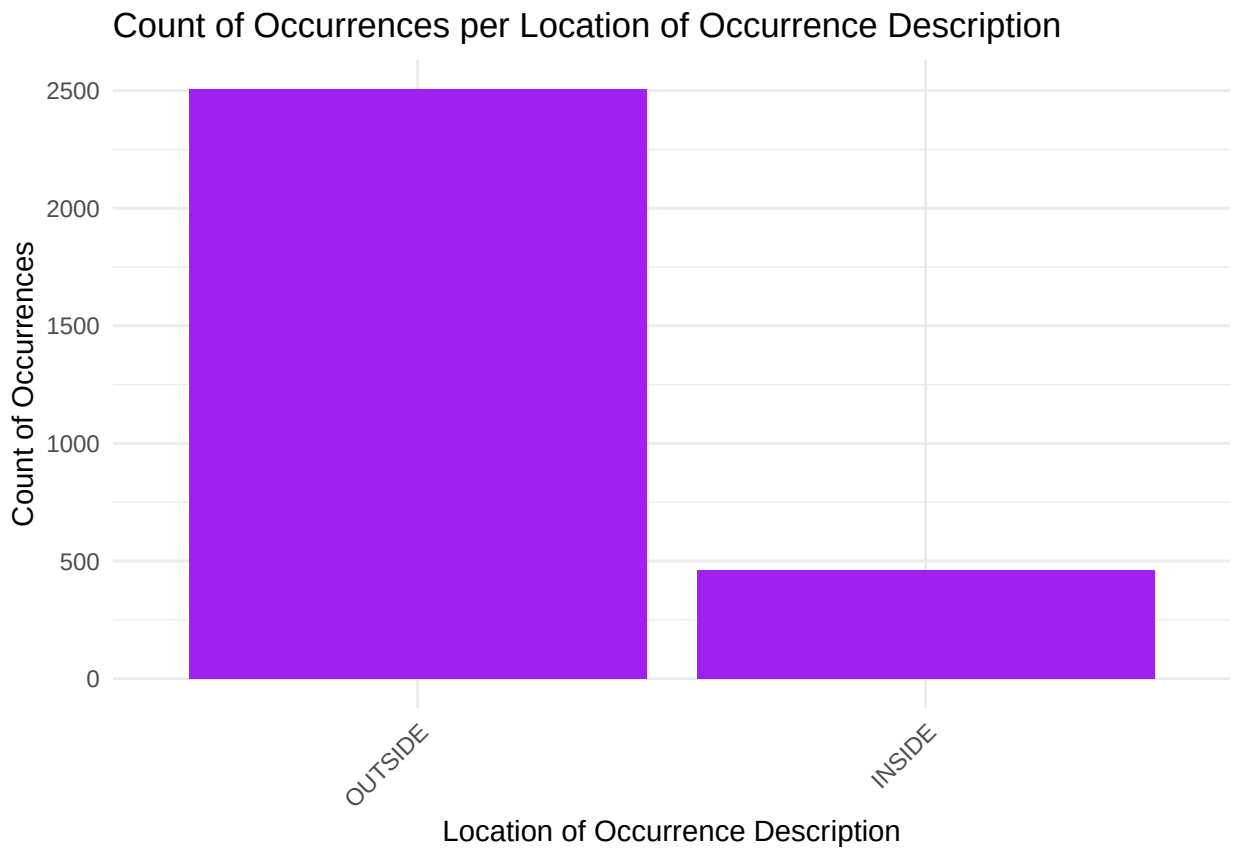
```
# Move all rows with invalid age group into an 'unknown' bucket
nypd <- nypd %>%
  mutate(PERP_AGE_GROUP = ifelse(PERP_AGE_GROUP %in% c('<18', '18-24', '25-44', '45-64', '65+'), PERP_A

age_group_counts <- nypd %>%
  group_by(PERP_AGE_GROUP) %>%
  summarise(count = n())

# Barplot of count of incidents per PERP_AGE_GROUP
ggplot(age_group_counts, aes(x = PERP_AGE_GROUP, y = count)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  theme_minimal() +
  labs(title = "Count of Incidents per Perpetrator Age Group",
       x = "Perpetrator Age Group",
       y = "Count of Incidents") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Do the incidents occur mostly inside or outside?

- It looks like incidents are most likely to occur outside. But since we have a lot of NA's in this column, we cannot tell with absolute certainty if this is truly the case.

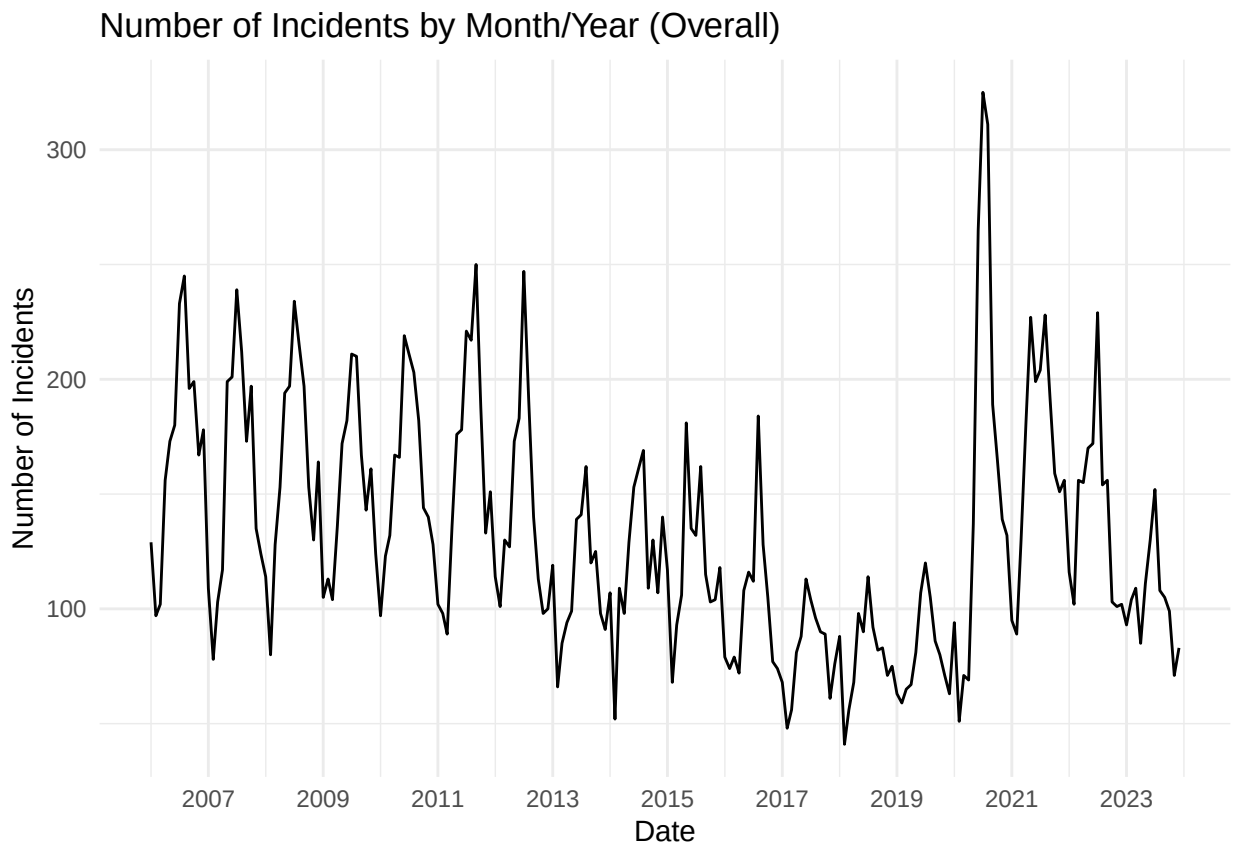
```
loc_counts <- nypd %>%
  filter(!is.na(LOC_OF_OCCUR_DESC)) %>%
  group_by(LOC_OF_OCCUR_DESC) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

ggplot(loc_counts, aes(x = reorder(LOC_OF_OCCUR_DESC, -count), y = count)) +
  geom_bar(stat = "identity", fill = "purple") +
  theme_minimal() +
  labs(title = "Count of Occurrences per Location of Occurrence Description",
       x = "Location of Occurrence Description",
       y = "Count of Occurrences") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Can we find trends in shooting incidents throughout the years?

- In the overall plot, we can observe a spike in shooting incidents from 2021 through 2022, which was around the pandemic.

```
set.seed(0)

# Convert OCCUR_DATE to Date format
nypd$OCCUR_DATE_MM <- as.Date(nypd$OCCUR_DATE, format="%m/%d/%Y")

# Aggregate the number of incidents by month/year
nypd$OCCUR_DATE_MM <- floor_date(nypd$OCCUR_DATE_MM, "month")
incidents_by_month <- nypd %>%
  group_by(OCCUR_DATE_MM) %>%
  summarise(Incidents = n())

# Plot overall time series of incidents by month/year
ggplot(incidents_by_month, aes(x = OCCUR_DATE_MM, y = Incidents)) +
  geom_line() +
  labs(title = 'Number of Incidents by Month/Year (Overall)', x = 'Date', y = 'Number of Incidents') +
  scale_x_date(date_breaks = "2 years", date_labels = "%Y") +
  theme_minimal()
```



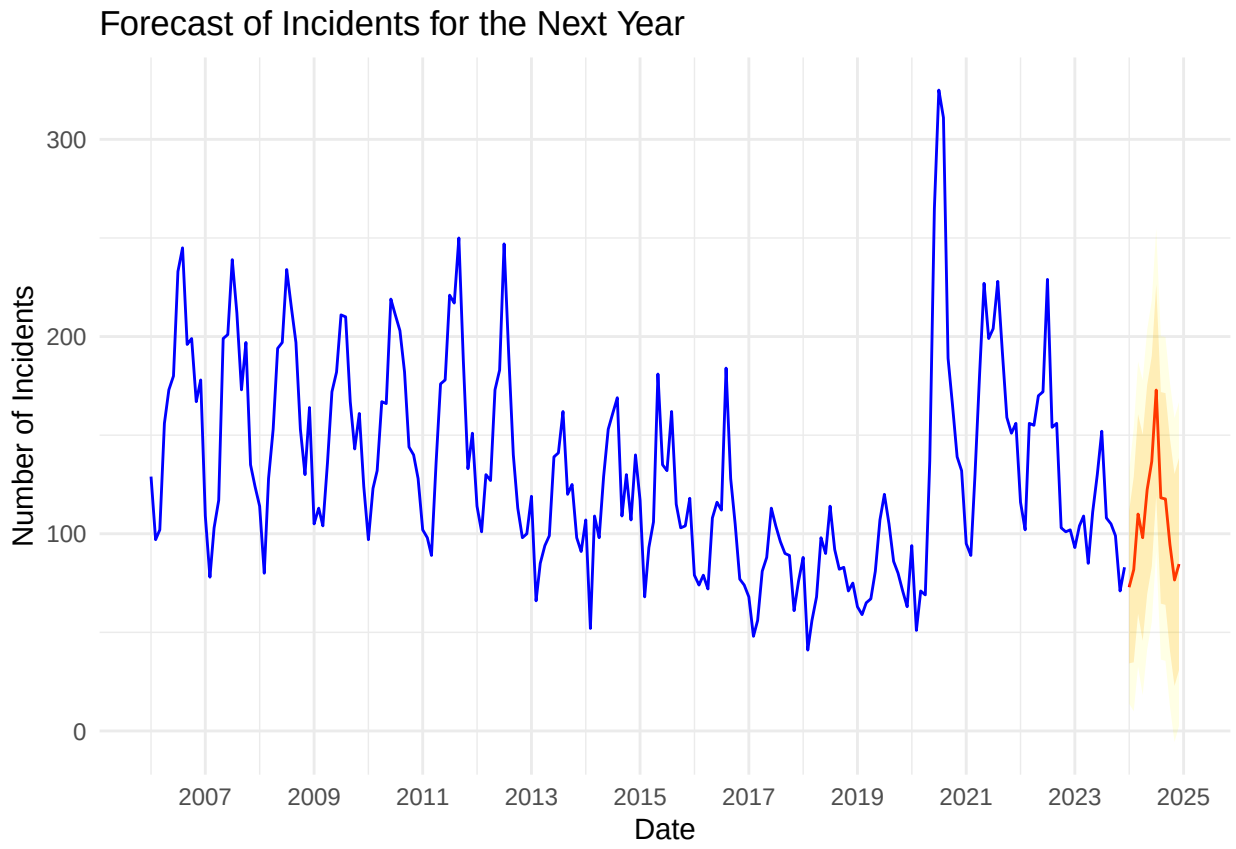
Can we generate a Time Series model that will predict the number of shooting incidents for the next 12 months?

Below is a code which generates a model using ARIMA, which uses data from the past and uses the pattern it finds to predict the future.

```
# Create a time series object
incidents_ts <- ts(incidents_by_month$Incidents, start = c(year(min(incidents_by_month$OCCUR_DATE_MM)),

# Fit an ARIMA model
fit <- auto.arima(incidents_ts)
```

```
# Forecast the next year (12 months)
  forecast_next_year <- forecast(fit, h = 12)

  # Convert forecast to data frame
  forecast_df <- data.frame(
    Date = seq(as.Date(tail(incidents_by_month$OCCUR_DATE_MM, 1)) %m+% months(1), by = "month", length.
    Point_Forecast = forecast_next_year$mean,
    Lo80 = forecast_next_year$lower[,1],
    Hi80 = forecast_next_year$upper[,1],
    Lo95 = forecast_next_year$lower[,2],
    Hi95 = forecast_next_year$upper[,2]
  )

  # Plot the forecast
  ggplot() +
```

```
    geom_line(data = incidents_by_month, aes(x = OCCUR_DATE_MM, y = Incidents), color = 'blue') +
    geom_line(data = forecast_df, aes(x = Date, y = Point_Forecast), color = 'red') +
    geom_ribbon(data = forecast_df, aes(x = Date, ymin = Lo80, ymax = Hi80), alpha = 0.2, fill = 'orang
    geom_ribbon(data = forecast_df, aes(x = Date, ymin = Lo95, ymax = Hi95), alpha = 0.1, fill = 'yellow
    scale_x_date(date_breaks = "2 years", date_labels = "%Y") +
    labs(title = 'Forecast of Incidents for the Next Year', x = 'Date', y = 'Number of Incidents') +
    theme_minimal()
```




```
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: x86_64-apple-darwin20
## Running under: macOS Ventura 13.6.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRlapack.dylib;  LAPACK
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
```

```
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] forecast_8.22.0 tidyr_1.3.1     lubridate_1.9.3 readr_2.1.5
## [5] ggplot2_3.5.1   dplyr_1.1.4
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.3    lattice_0.22-6    hms_1.1.3
##  [5] digest_0.6.35     magrittr_2.0.3    evaluate_0.23     grid_4.4.0
##  [9] timechange_0.3.0  fastmap_1.1.1     nnet_7.3-19       purrr_1.0.2
## [13] fansi_1.0.6       scales_1.3.0      cli_3.6.2         crayon_1.5.2
## [17] rlang_1.1.3       bit64_4.0.5       munsell_0.5.1     withr_3.0.0
## [21] yaml_2.3.8        tools_4.4.0       parallel_4.4.0    tzdb_0.4.0
## [25] colorspace_2.1-0  curl_5.2.1        vctrs_0.6.5       R6_2.5.1
## [29] zoo_1.8-12        lifecycle_1.0.4   tseries_0.10-56   bit_4.0.5
## [33] vroom_1.6.5       pkgconfig_2.0.3   urca_1.3-3        pillar_1.9.0
## [37] gtable_0.3.5      glue_1.7.0        quantmod_0.4.26   Rcpp_1.0.12
## [41] highr_0.10        xfun_0.43         tibble_3.2.1      lmtest_0.9-40
## [45] tidyselect_1.2.1  rstudioapi_0.16.0 knitr_1.46        farver_2.1.2
## [49] nlme_3.1-164      htmltools_0.5.8.1 labeling_0.4.3    rmarkdown_2.26
## [53] xts_0.13.2        timeDate_4032.109 fracdiff_1.5-3    compiler_4.4.0
## [57] quadprog_1.5-8    TTR_0.24.4
```