NYPD: Time and Space of Shooting Incidents

Data Description

The data for this project is from the New York City Open Data portal.
The dataset is the NYPD Shooting Incident Data (Historic) and contains information on shooting incidents in New York City.
The data is available in CSV format and can be downloaded from the following link:
https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic

Detailed description of the dataset was provided by the NYPD:
Details on data description: https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8/about_data

```
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(tidyr))
suppressPackageStartupMessages(library(lubridate))
suppressPackageStartupMessages(library(dplyr))
```

Data Import and Structure

```
df <- read.csv('https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD')
head(df, 2)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO LOC_OF_OCCUR_DESC PRECINCT
## 1    231974218 08/09/2021   01:06:00    BRONX                        40
## 2    177934247 04/07/2018   19:48:00 BROOKLYN                        79
##   JURISDICTION_CODE LOC_CLASSFCTN_DESC LOCATION_DESC STATISTICAL_MURDER_FLAG
## 1                 0                                                   false
## 2                 0                                                    true
##   PERP_AGE_GROUP PERP_SEX      PERP_RACE VIC_AGE_GROUP VIC_SEX VIC_RACE
## 1                                          18-24       M    BLACK
## 2          25-44        M WHITE HISPANIC         25-44       M    BLACK
##   X_COORD_CD Y_COORD_CD Latitude Longitude
## 1    1006343   234270.0 40.80967 -73.92019
## 2    1000083   189064.7 40.68561 -73.94291
##                                      Lon_Lat
## 1  POINT (-73.92019278899994 40.80967347200004)
## 2 POINT (-73.94291302299996 40.685609672000055)
```

```
names(df) <- tolower(names(df))
str(df)
```

```
## 'data.frame':    28562 obs. of  21 variables:
##  $ incident_key           : int  231974218 177934247 255028563 25384540 72616285 85875439 79780323 85
```

```
##  $ occur_date             : chr  "08/09/2021" "04/07/2018" "12/02/2022" "11/19/2006" ...
##  $ occur_time             : chr  "01:06:00" "19:48:00" "22:57:00" "01:50:00" ...
##  $ boro                   : chr  "BRONX" "BROOKLYN" "BRONX" "BROOKLYN" ...
##  $ loc_of_occur_desc      : chr  "" "" "OUTSIDE" "" ...
##  $ precinct               : int  40 79 47 66 46 42 71 69 75 69 ...
##  $ jurisdiction_code      : int  0 0 0 0 0 2 0 2 0 0 ...
##  $ loc_classfctn_desc     : chr  "" "" "STREET" "" ...
##  $ location_desc          : chr  "" "" "GROCERY/BODEGA" "PVT HOUSE" ...
##  $ statistical_murder_flag: chr  "false" "true" "false" "true" ...
##  $ perp_age_group         : chr  "" "25-44" "(null)" "UNKNOWN" ...
##  $ perp_sex               : chr  "" "M" "(null)" "U" ...
##  $ perp_race              : chr  "" "WHITE HISPANIC" "(null)" "UNKNOWN" ...
##  $ vic_age_group          : chr  "18-24" "25-44" "25-44" "18-24" ...
##  $ vic_sex                : chr  "M" "M" "M" "M" ...
##  $ vic_race               : chr  "BLACK" "BLACK" "BLACK" "BLACK" ...
##  $ x_coord_cd             : num  1006343 1000083 1020691 985107 1009854 ...
##  $ y_coord_cd             : num  234270 189065 257125 173350 247503 ...
##  $ latitude               : num  40.8 40.7 40.9 40.6 40.8 ...
##  $ longitude              : num  -73.9 -73.9 -73.9 -74 -73.9 ...
##  $ lon_lat                : chr  "POINT (-73.92019278899994 40.80967347200004)" "POINT (-73.94291302
```

Column Name	Description	API Field Name	Data Type
INCIDENT_KEY	Randomly generated persistent ID for each arrest	incident_key	Text
OCCUR_DATE	Exact date of the shooting incident	occur_date	Floating Timestamp
OCCUR_TIME	Exact time of the shooting incident	occur_time	Text
BORO	Borough where the shooting incident occurred	boro	Text
LOC_OF_OCCUR_DESC	Location where the shooting incident occurred	loc_of_occur_desc	Text
PRECINCT	Precinct where the shooting incident occurred	precinct	Number
JURISDICTION_CODE	Jurisdiction where the shooting incident occurred. Jurisdiction codes 0(Patrol), 1(Transit) and 2(Housing) represent NYPD whilst codes 3 and more represent non NYPD jurisdictions	jurisdiction_code	Number
LOC_CLASSFCTN_DESC	Classification of the location where the shooting incident occurred	loc_classfctn_desc	Text
LOCATION_DESC	Location of the shooting incident	location_desc	Text
STATISTICAL_MURDER_FLAG	Shooting resulted in the victim's death which would be counted as a murder	statistical_murder_flag	Checkbox
PERP_AGE_GROUP	Perpetrator's age within a category	perp_age_group	Text
PERP_SEX	Perpetrator's sex description	perp_sex	Text
PERP_RACE	Perpetrator's race description	perp_race	Text
VIC_AGE_GROUP	Victim's age within a category	vic_age_group	Text
VIC_SEX	Victim's sex description	vic_sex	Text
VIC_RACE	Victim's race description	vic_race	Text
X_COORD_CD	Midblock X-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet (FIPS 3104)	x_coord_cd	Text
Y_COORD_CD	Midblock Y-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet (FIPS 3104)	y_coord_cd	Text
Latitude	Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)	latitude	Number
Longitude	Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)	longitude	Number
Lon_Lat	Longitude and Latitude Coordinates for mapping	geocoded_column	Point

Column Name	Description	API Field Name	Data Type

Tidy and Transforming Data

We will proceed all the columns of the dataset one by one to study them and decide what to do with them. The important limitation is that due to the limited size of the assignment, I would like to focus on data related to time and space of the incidents.

Unique Identifiers, Dates, and Times

incendent_key a unique identifier for each incident (int)
While it should be unique, we will check if there are any duplicates.
There are duplicates by incident_key, as well as by key time and location.
Thus we can assume that for each incident there are multiple records, probably related to several victims.

```
length(unique(df$incident_key))
```

```
## [1] 22394
```

```
sum(duplicated(df$incident_key)) # 6168
```

```
## [1] 6168
```

```
sum(duplicated(df[c("incident_key", "occur_date", 'lon_lat', 'precinct')])) # 6168
```

```
## [1] 6168
```

```
# full duplicates
sum(duplicated(df)) # 0
```

```
## [1] 0
```

```
str(df)
```

```
## 'data.frame':    28562 obs. of  21 variables:
##  $ incident_key           : int  231974218 177934247 255028563 25384540 72616285 85875439 79780323 85
##  $ occur_date             : chr  "08/09/2021" "04/07/2018" "12/02/2022" "11/19/2006" ...
##  $ occur_time             : chr  "01:06:00" "19:48:00" "22:57:00" "01:50:00" ...
##  $ boro                   : chr  "BRONX" "BROOKLYN" "BRONX" "BROOKLYN" ...
##  $ loc_of_occur_desc      : chr  "" "" "OUTSIDE" "" ...
##  $ precinct               : int  40 79 47 66 46 42 71 69 75 69 ...
##  $ jurisdiction_code      : int  0 0 0 0 0 0 2 0 2 0 0 ...
##  $ loc_classfctn_desc     : chr  "" "" "STREET" "" ...
##  $ location_desc          : chr  "" "" "GROCERY/BODEGA" "PVT HOUSE" ...
##  $ statistical_murder_flag: chr  "false" "true" "false" "true" ...
##  $ perp_age_group         : chr  "" "25-44" "(null)" "UNKNOWN" ...
```

```
##  $ perp_sex              : chr  "" "M" "(null)" "U" ...
##  $ perp_race             : chr  "" "WHITE HISPANIC" "(null)" "UNKNOWN" ...
##  $ vic_age_group         : chr  "18-24" "25-44" "25-44" "18-24" ...
##  $ vic_sex               : chr  "M" "M" "M" "M" ...
##  $ vic_race              : chr  "BLACK" "BLACK" "BLACK" "BLACK" ...
##  $ x_coord_cd            : num  1006343 1000083 1020691 985107 1009854 ...
##  $ y_coord_cd            : num  234270 189065 257125 173350 247503 ...
##  $ latitude              : num  40.8 40.7 40.9 40.6 40.8 ...
##  $ longitude             : num  -73.9 -73.9 -73.9 -74 -73.9 ...
##  $ lon_lat               : chr  "POINT (-73.92019278899994 40.80967347200004)" "POINT (-73.94291302
```

occure_date - character in format "MM/DD/YYYY" - we need to convert to date
occure_time - character in format "HH:MM:SS" - we need to convert to time (without date)
the detailed time of incindent may be of lesser interest, however its destribution by hour is something to be studied

```
df$occur_date <- as.Date(df$occur_date, format = "%m/%d/%Y")

df$occur_time <- as.POSIXct(df$occur_time, format = "%H:%M:%S")
df$hour_of_occurance <- as.numeric(format(df$occur_time, "%H"))

df <- subset(df, select = -c(occur_time))
```

Locations

boro character - we need to convert to factor it is the bigger area of the city

```
length(unique(df$boro))
```

```
## [1] 5
```

```
unique(df$boro)
```

```
## [1] "BRONX"         "BROOKLYN"      "MANHATTAN"     "QUEENS"
## [5] "STATEN ISLAND"
```

```
sum(is.na(df$boro)) # 0
```

```
## [1] 0
```

```
df$boro <- factor(df$boro)
```

loc_of_occur_desc character - there are to many missing values - remove this data

```
length(unique(df$loc_of_occur_desc))
```

```
## [1] 3
```

```r
unique(df$loc_of_occur_desc)
```

```
## [1] ""        "OUTSIDE" "INSIDE"
```

```r
df$loc_of_occur_desc <- factor(df$loc_of_occur_desc)

# count number of rows for each loc_of_occur_desc
df %>%
  group_by(loc_of_occur_desc) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  head(10)
```

```
## # A tibble: 3 x 2
##   loc_of_occur_desc count
##   <fct>             <int>
## 1 ""                25596
## 2 "OUTSIDE"          2506
## 3 "INSIDE"            460
```

```r
# delete this column
df <- subset(df, select = -c(loc_of_occur_desc))
```

precinct integer - but it is factor by nature - we would need to convert it to factor
However, this is very detailed location information, which can be skipped for this analysis. **jurisdiction_code** integer - but it is factor by nature - we need to convert to factor

```r
summary(df$precinct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    44.0    67.0    65.5    81.0   123.0
```

```r
df <- subset(df, select = -c(precinct))

length(unique(df$jurisdiction_code))
```

```
## [1] 4
```

```r
unique(df$jurisdiction_code)
```

```
## [1]  0  2  1 NA
```

```r
df$jurisdiction_code <- factor(df$jurisdiction_code)
sum(is.na(df$jurisdiction_code))
```

```
## [1] 2
```

loc_classfctn_desc character - we would need to convert to factor the location classification, however there are to many missing values
location_desc too many missing values - remove this data
coordinates
We will drop all the coordinates (x_coord_cd, y_coord_cd, latitude, longitude, lon_lat)
Analysis of detailed geographical data is out of scope of this project.

```
length(unique(df$loc_classfctn_desc))
```

```
## [1] 11
```

```
unique(df$loc_classfctn_desc)
```

```
##  [1] ""            "STREET"      "OTHER"       "PLAYGROUND"  "TRANSIT"
##  [6] "HOUSING"     "COMMERCIAL"  "DWELLING"    "VEHICLE"     "PARKING LOT"
## [11] "(null)"
```

```
# (null) and empty string should be treated as missing values
df$loc_classfctn_desc[df$loc_classfctn_desc == "(null)" | df$loc_classfctn_desc == ""] <- NA
df$loc_classfctn_desc <- factor(df$loc_classfctn_desc)

# count all the data for this one for each factor
df %>%
  group_by(loc_classfctn_desc) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  head(10)
```

```
## # A tibble: 10 x 2
##    loc_classfctn_desc count
##    <fct>              <int>
##  1 <NA>               25598
##  2 STREET              1886
##  3 HOUSING              460
##  4 DWELLING             243
##  5 COMMERCIAL           208
##  6 OTHER                 59
##  7 PLAYGROUND            41
##  8 VEHICLE               29
##  9 TRANSIT               23
## 10 PARKING LOT           15
```

```
df <- subset(df, select = -c(loc_classfctn_desc))

length(unique(df$location_desc))
```

```
## [1] 41
```

```
unique(df$location_desc)
```

```
##  [1] ""                           "GROCERY/BODEGA"
##  [3] "PVT HOUSE"                  "MULTI DWELL - APT BUILD"
##  [5] "MULTI DWELL - PUBLIC HOUS" "(null)"
##  [7] "BAR/NIGHT CLUB"             "COMMERCIAL BLDG"
##  [9] "FAST FOOD"                  "HOSPITAL"
## [11] "BEAUTY/NAIL SALON"          "LIQUOR STORE"
## [13] "CHAIN STORE"                "RESTAURANT/DINER"
## [15] "SMALL MERCHANT"             "GAS STATION"
## [17] "JEWELRY STORE"              "GYM/FITNESS FACILITY"
## [19] "STORE UNCLASSIFIED"         "SOCIAL CLUB/POLICY LOCATI"
## [21] "DRY CLEANER/LAUNDRY"        "NONE"
## [23] "VIDEO STORE"                "SUPERMARKET"
## [25] "VARIETY STORE"              "FACTORY/WAREHOUSE"
## [27] "CLOTHING BOUTIQUE"          "SHOE STORE"
## [29] "HOTEL/MOTEL"                "CANDY STORE"
## [31] "DEPT STORE"                 "BANK"
## [33] "TELECOMM. STORE"            "DRUG STORE"
## [35] "LOAN COMPANY"               "CHECK CASH"
## [37] "SCHOOL"                     "STORAGE FACILITY"
## [39] "PHOTO/COPY STORE"           "ATM"
## [41] "DOCTOR/DENTIST"
```

```
df <- subset(df, select = -c(location_desc))

# drop all the coordinates
df <- subset(df, select = -c(x_coord_cd, y_coord_cd, latitude, longitude, lon_lat))
```

Other data fields

Information on the age, sex, and race of the people involved is skipped in this analysis.
While I found this an extremely important topic, I believe it will be studied in other work.
Also, I do not have a background in US life, and my bias is that I am out of context.

statistical_murder_flag we need to bring it back to bool

```
# Drop columns per_age_group, perp_sex, perp_race, vic_age_group, vic_sex, vic_race
df <- subset(df, select = -c(perp_age_group, perp_sex, perp_race, vic_age_group, vic_sex, vic_race))

length(unique(df$statistical_murder_flag))
```

```
## [1] 2
```

```
unique(df$statistical_murder_flag)
```

```
## [1] "false" "true"
```

```
# bring it to bool
df$statistical_murder_flag <- as.logical(df$statistical_murder_flag)
```

```
str(df)
```

```
## 'data.frame':    28562 obs. of  6 variables:
##  $ incident_key           : int  231974218 177934247 255028563 25384540 72616285 85875439 79780323 85
##  $ occur_date             : Date, format: "2021-08-09" "2018-04-07" ...
##  $ boro                   : Factor w/ 5 levels "BRONX","BROOKLYN",..: 1 2 1 2 1 1 2 2 2 2 ...
##  $ jurisdiction_code      : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 3 1 3 1 1 ...
##  $ statistical_murder_flag: logi  FALSE TRUE FALSE TRUE TRUE FALSE ...
##  $ hour_of_occurance      : num  1 19 22 1 1 21 22 23 15 15 ...
```

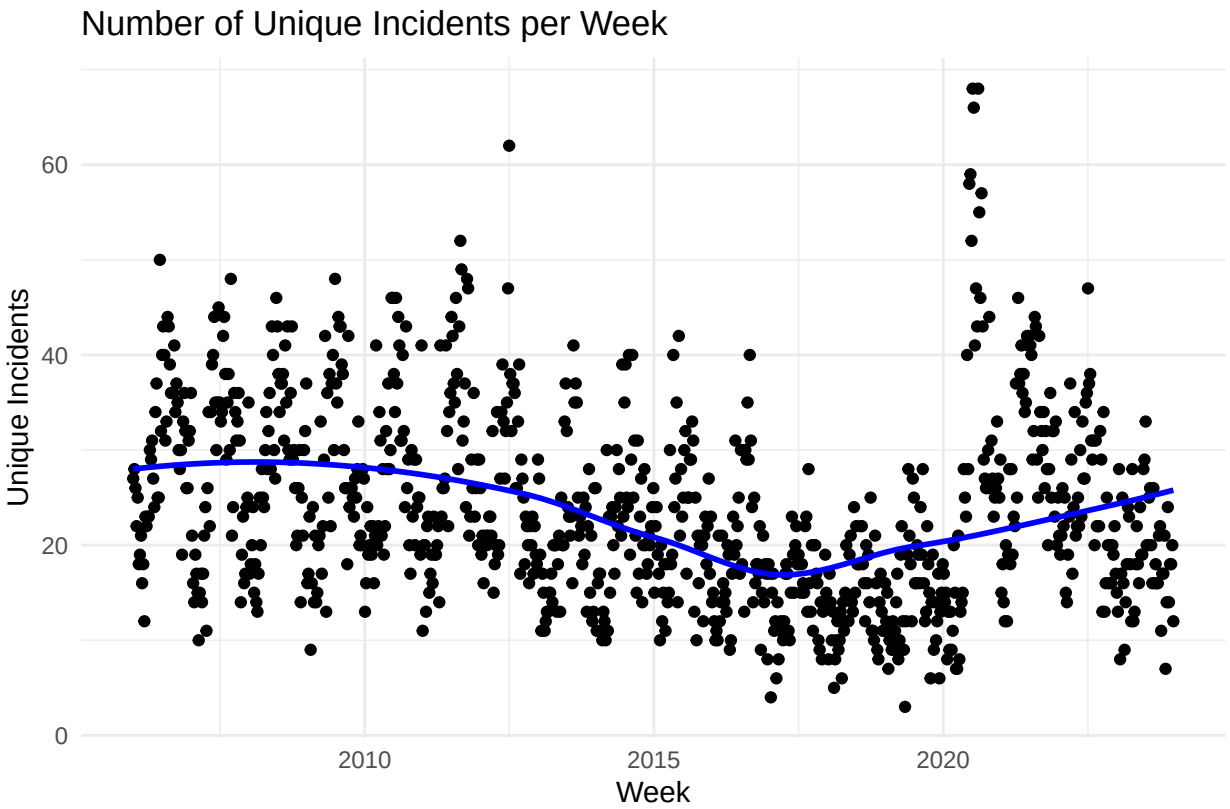
Vusyalising, Analyzing, and Modeling Data

Data and time related analysis

At first, we will study how the number of incidents depends on time.

```
# First, let's calculate unique incidents by week
df_weekly <- df %>%
  mutate(week = floor_date(occur_date, "week")) %>%
  group_by(week) %>%
  summarize(unique_incidents = n_distinct(incident_key))

# Scatter plot of unique incidents by week
p1 <- ggplot(df_weekly, aes(x = week, y = unique_incidents)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  labs(
    title = "Number of Unique Incidents per Week",
    x = "Week",
    y = "Unique Incidents",
    caption = "Source: Incident Data"
  ) +
  theme_minimal()

print(p1)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



First of all, **the overall number of incindents does not clearly decline over time**.
While there was decline in mid 2010s, it did not become a trend.
May be COVID-19 pandemic had an impact on the number of incidents - which spiked at 2020 and 2021.
In future years it will be interesting to see if the number of incidents will decline again.
There is clearly visible seasonability of the incidents, which we will study in more detail later.

```
# Calculate summary statistics
total_unique_incidents <- n_distinct(df$incident_key)
avg_weekly_incidents <- mean(df_weekly$unique_incidents)

# Print the summary statistics
cat("Total number of unique incidents:", total_unique_incidents, "\n")
```

```
## Total number of unique incidents: 22394
```

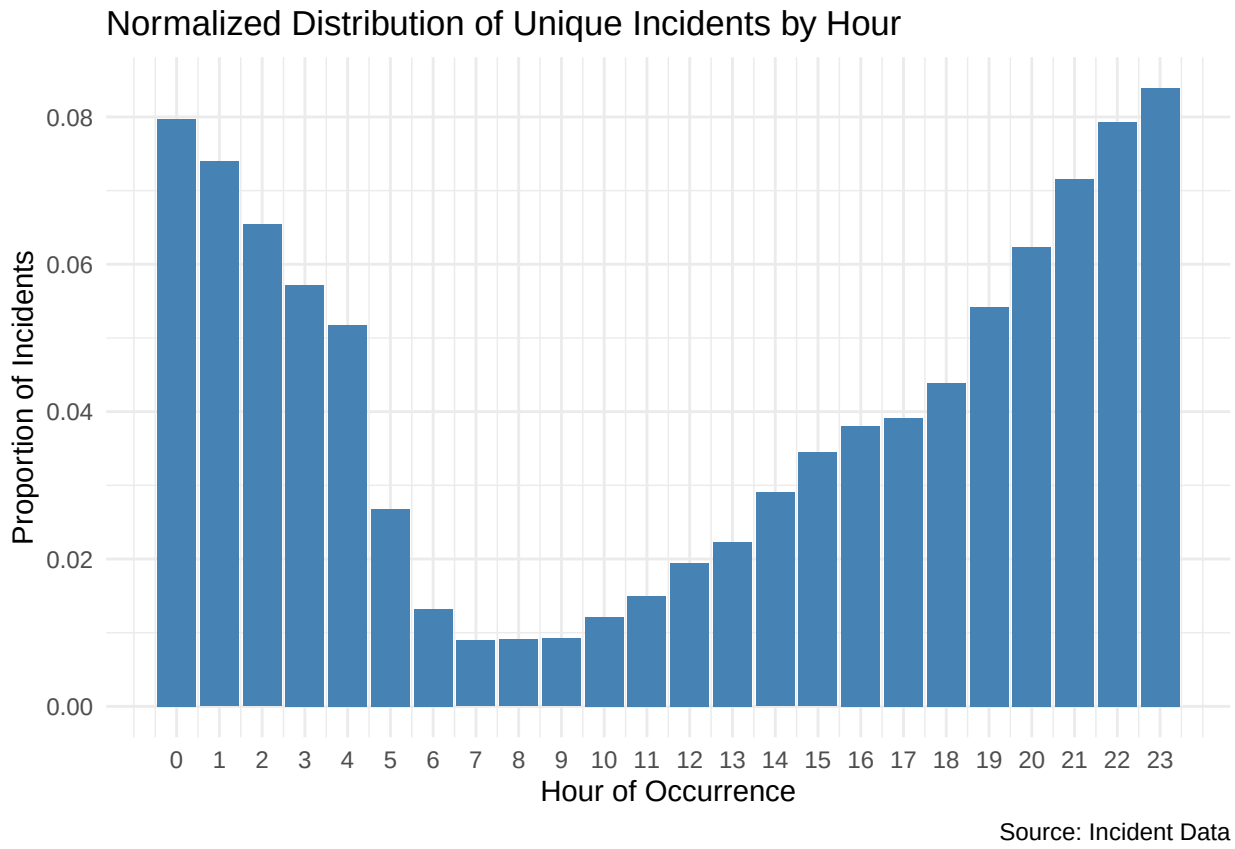
```
cat("Average number of unique incidents per week:", round(avg_weekly_incidents, 2), "\n")
```

```
## Average number of unique incidents per week: 23.85
```

Astonishingly, the average number of incidents per week is 24.
Which is **more than 3 shootings per day**.

```
# Calculate unique incidents by hour (normalized)
df_hourly <- df %>%
```

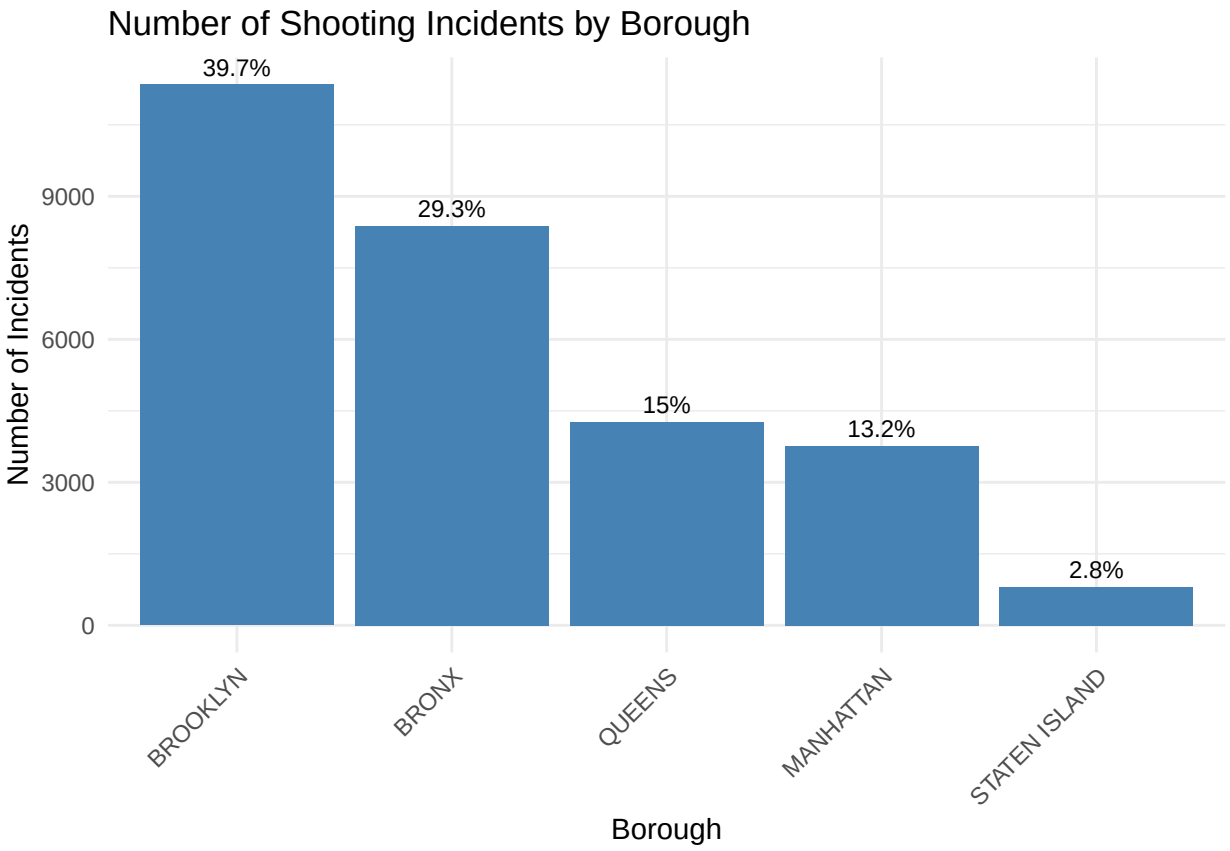
```r
  group_by(hour_of_occurance) %>%
  summarize(
    unique_incidents = n_distinct(incident_key),
    .groups = "drop"
  ) %>%
  mutate(normalized_incidents = unique_incidents / sum(unique_incidents))

# Bar chart of unique incidents by hour (normalized)
p3 <- ggplot(df_hourly, aes(x = hour_of_occurance, y = normalized_incidents)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(
    title = "Normalized Distribution of Unique Incidents by Hour",
    x = "Hour of Occurrence",
    y = "Proportion of Incidents",
    caption = "Source: Incident Data"
  ) +
  scale_x_continuous(breaks = 0:23) +
  theme_minimal()

print(p3)
```



Clearly the safest time is between 6 and 8 am. With peak of incidents around midnight.

Place and jurisdiction related analysis

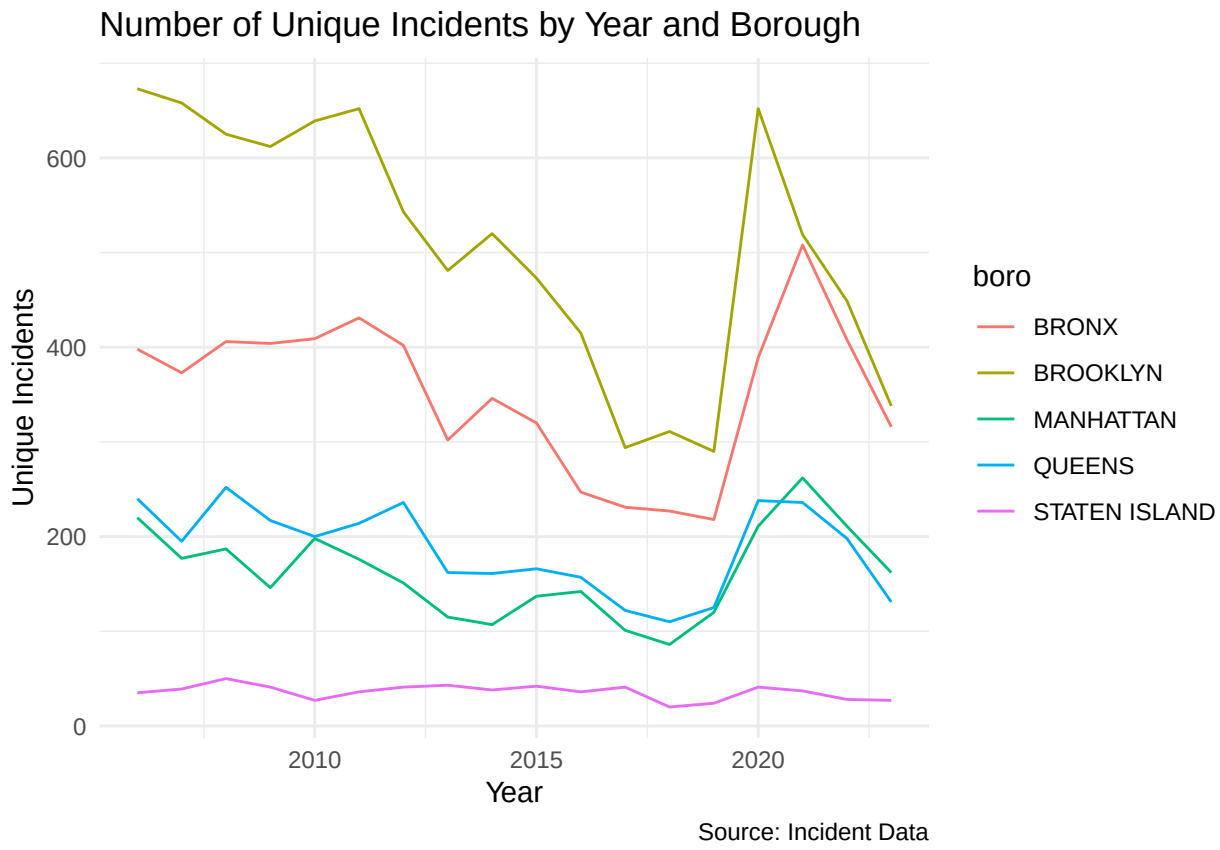
```
boro_counts <- df %>%
  count(boro) %>%
  mutate(percentage = n/sum(n) * 100)

ggplot(boro_counts, aes(x = reorder(boro, -n), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = paste0(round(percentage, 1), "%")),
            vjust = -0.5, size = 3) +
  labs(title = "Number of Shooting Incidents by Borough",
       x = "Borough", y = "Number of Incidents") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Brooklyn is the most dangerous borough with 40% of all incidents.
Staten Island is the safest borough with only 3% of all incidents.
It would be important to compare the number of incidents with the population of each borough.

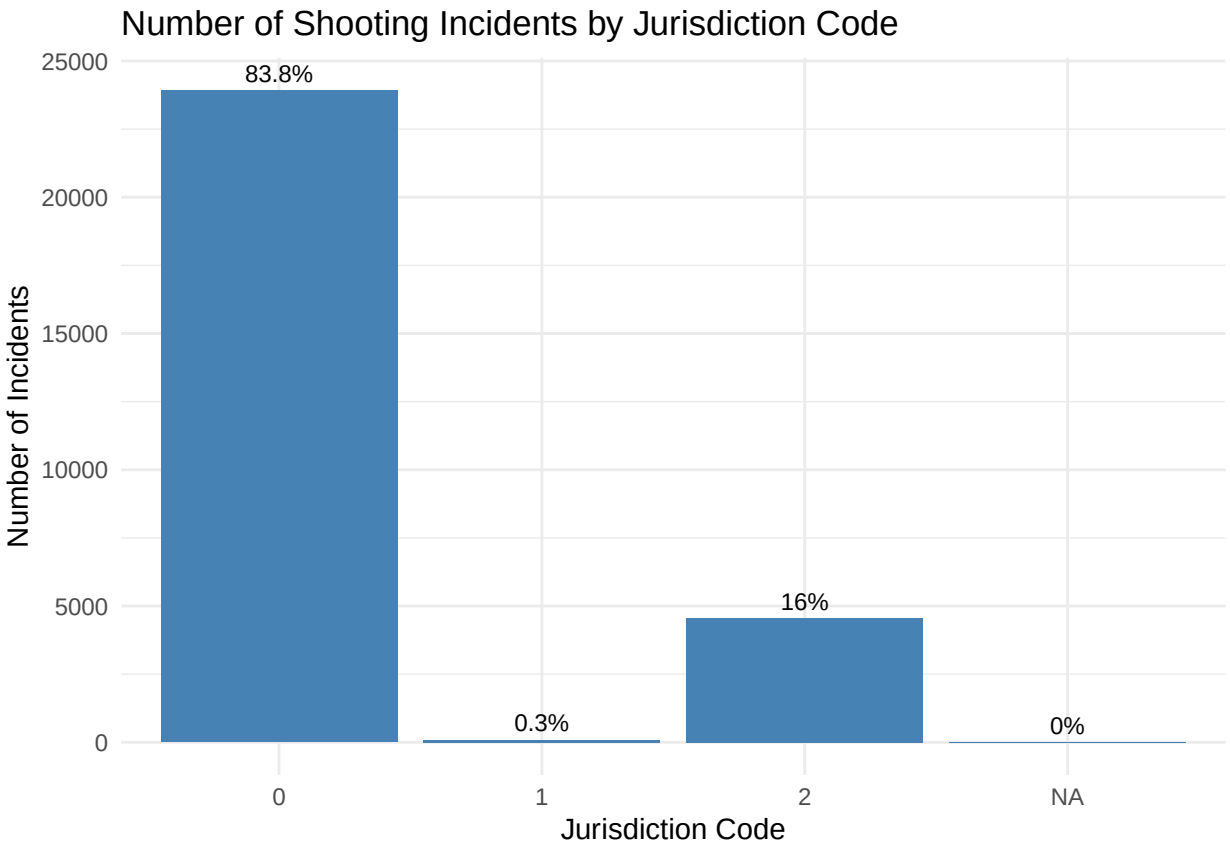
```
df_boro_yearly <- df %>%
  mutate(year = year(occur_date)) %>%
  group_by(year, boro) %>%
  summarize(unique_incidents = n_distinct(incident_key), .groups = "drop")
```

```r
ggplot(df_boro_yearly, aes(x = year, y = unique_incidents, color = boro)) +
  geom_line() +
  labs(
    title = "Number of Unique Incidents by Year and Borough",
    x = "Year",
    y = "Unique Incidents",
    caption = "Source: Incident Data"
  ) +
  theme_minimal()
```



Pre-COVID dynamics was positive or Brooklyn and Bronx, while rather steady for Manhattan, Queens, and Staten Island.

```r
jurisdiction_counts <- df %>%
  count(jurisdiction_code) %>%
  mutate(percentage = n/sum(n) * 100)

ggplot(jurisdiction_counts, aes(x = as.factor(jurisdiction_code), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = paste0(round(percentage, 1), "%")),
            vjust = -0.5, size = 3) +
  labs(title = "Number of Shooting Incidents by Jurisdiction Code",
       x = "Jurisdiction Code", y = "Number of Incidents") +
  theme_minimal()
```



Jurisdiction codes 0(Patrol), 1(Transit) and 2(Housing).
Clearly the majority of incidents are in the Patrol jurisdiction with over 80$% of all incidents.

Modeling Data

We will focus our attnetion on time-related dependancies for this report.
First with relation to hour of the day.

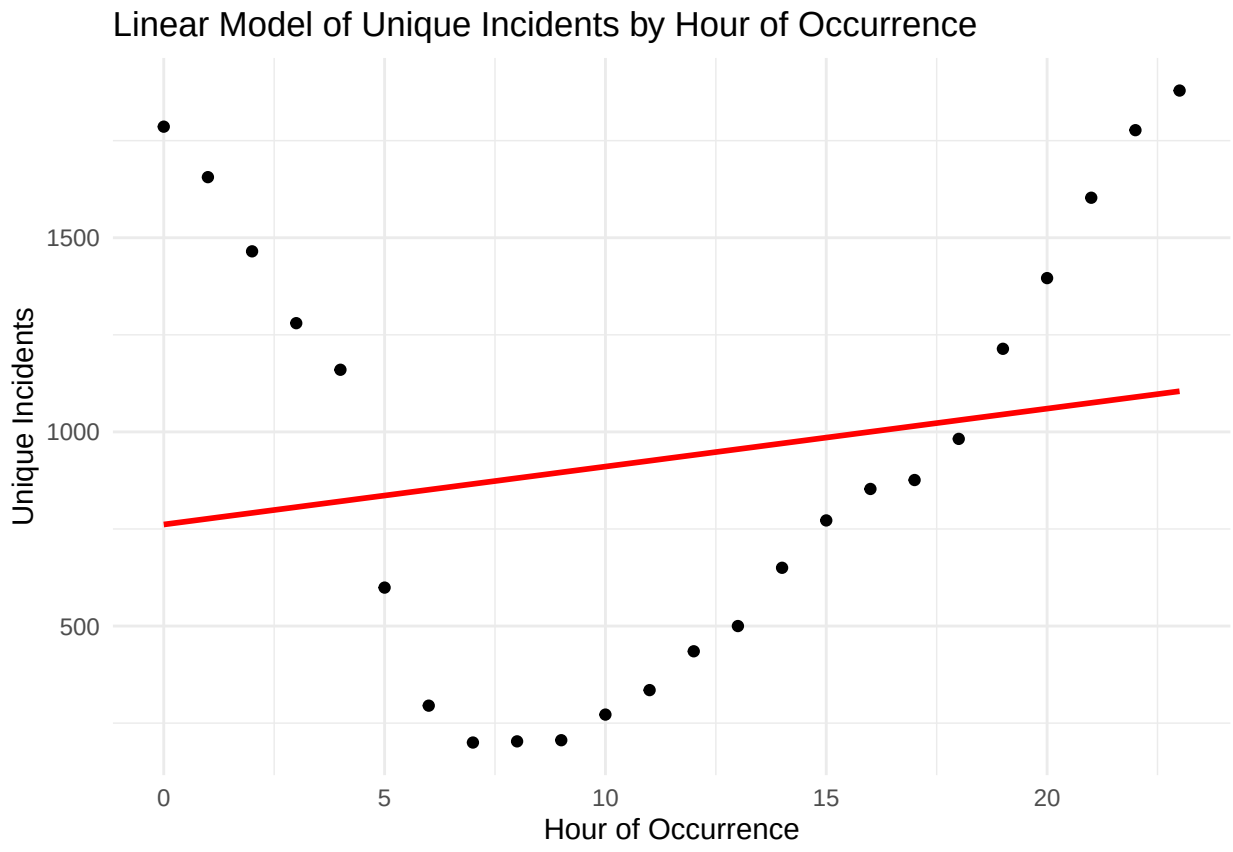
```
df_hourly <- df %>%
  group_by(hour_of_occurance) %>%
  summarize(unique_incidents = n_distinct(incident_key), .groups = "drop")


model_1 <- lm(unique_incidents ~ hour_of_occurance, data = df_hourly)
summary(model_1)
```

```
##
## Call:
## lm(formula = unique_incidents ~ hour_of_occurance, data = df_hourly)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -689.8 -518.2 -143.2  487.4 1024.5
##
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         761.49     227.44   3.348  0.00291 **
## hour_of_occurance    14.92      16.94   0.881  0.38807
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 574.6 on 22 degrees of freedom
## Multiple R-squared:  0.03405,    Adjusted R-squared:  -0.009862
## F-statistic: 0.7754 on 1 and 22 DF,  p-value: 0.3881
```

```r
ggplot(df_hourly, aes(x = hour_of_occurance, y = unique_incidents)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(
    title = "Linear Model of Unique Incidents by Hour of Occurrence",
    x = "Hour of Occurrence",
    y = "Unique Incidents"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



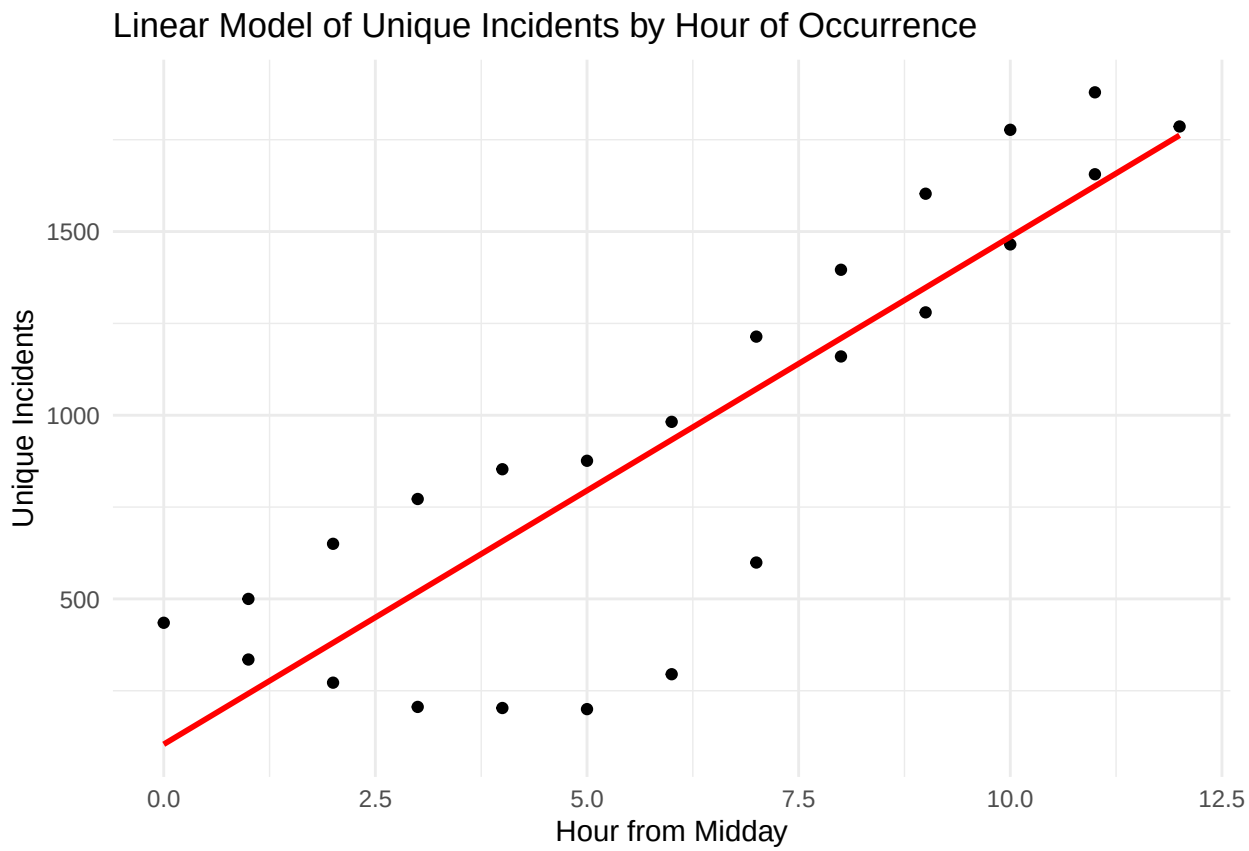
The linear correlation with just hour of day is neglactble.
However, if we take how far the hour is from midday, we can see a clear linear correlation.

```r
df_hourly$hour_from_midday <- abs(df_hourly$hour_of_occurance - 12)
model_2 <- lm(unique_incidents ~ hour_from_midday, data = df_hourly)
summary(model_2)
```

```
##
## Call:
## lm(formula = unique_incidents ~ hour_from_midday, data = df_hourly)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -638.08  -77.76   64.99  253.81  330.78
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        104.22     120.85   0.862    0.398
## hour_from_midday   138.14      17.41   7.934 6.78e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 297.5 on 22 degrees of freedom
## Multiple R-squared:  0.741,  Adjusted R-squared:  0.7292
## F-statistic: 62.94 on 1 and 22 DF,  p-value: 6.778e-08
```

```r
# Plot the model

ggplot(df_hourly, aes(x = hour_from_midday, y = unique_incidents)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(
    title = "Linear Model of Unique Incidents by Hour of Occurrence",
    x = "Hour from Midday",
    y = "Unique Incidents"
  ) +
  theme_minimal()
```

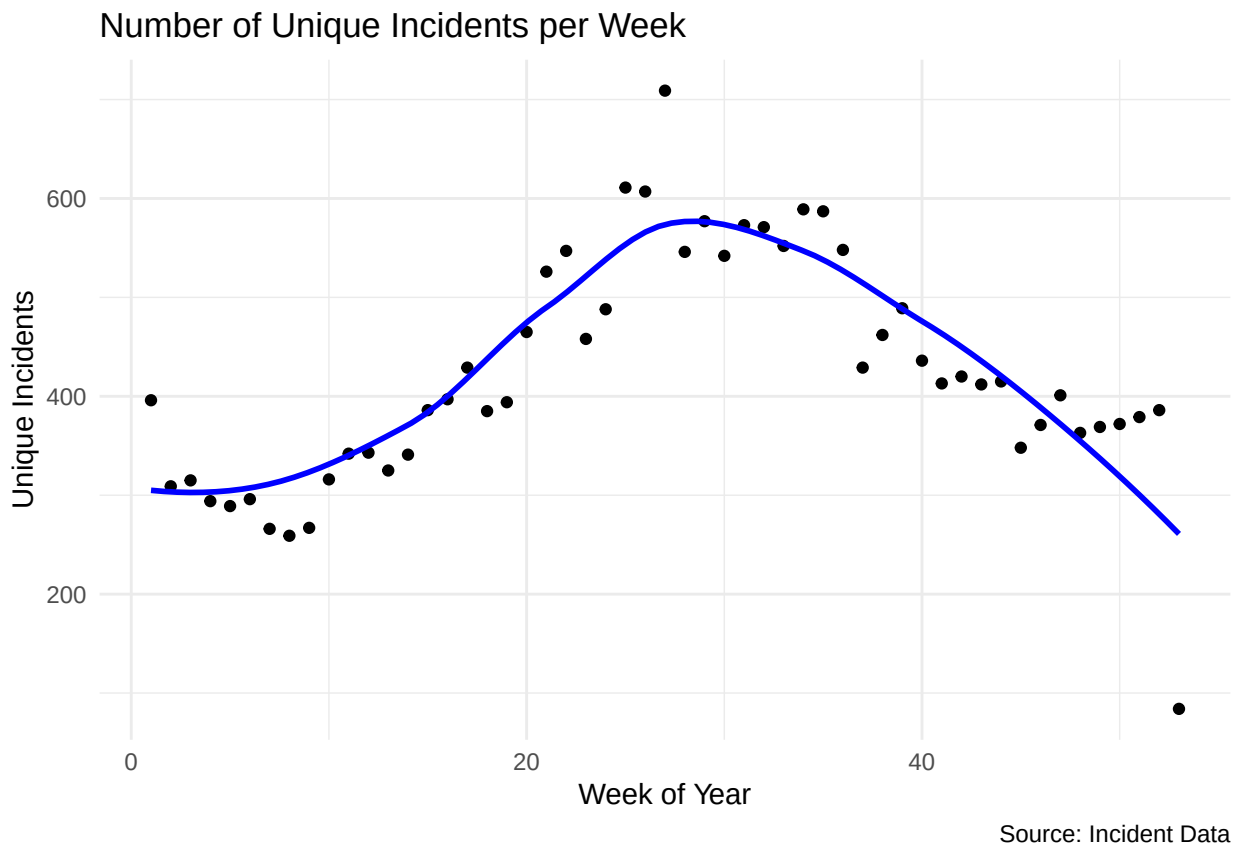
```
## `geom_smooth()` using formula = 'y ~ x'
```



Another interesting aspect is seasonability on longer time scale.
Week and month of the year.

```
df$week_of_year <- week(df$occur_date)

df_by_week <- df %>%
  group_by(week_of_year) %>%
  summarize(unique_incidents = n_distinct(incident_key), .groups = "drop")

p4 <- ggplot(df_by_week, aes(x = week_of_year, y = unique_incidents)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  labs(
    title = "Number of Unique Incidents per Week",
    x = "Week of Year",
    y = "Unique Incidents",
    caption = "Source: Incident Data"
  ) +
  theme_minimal()

print(p4)
```

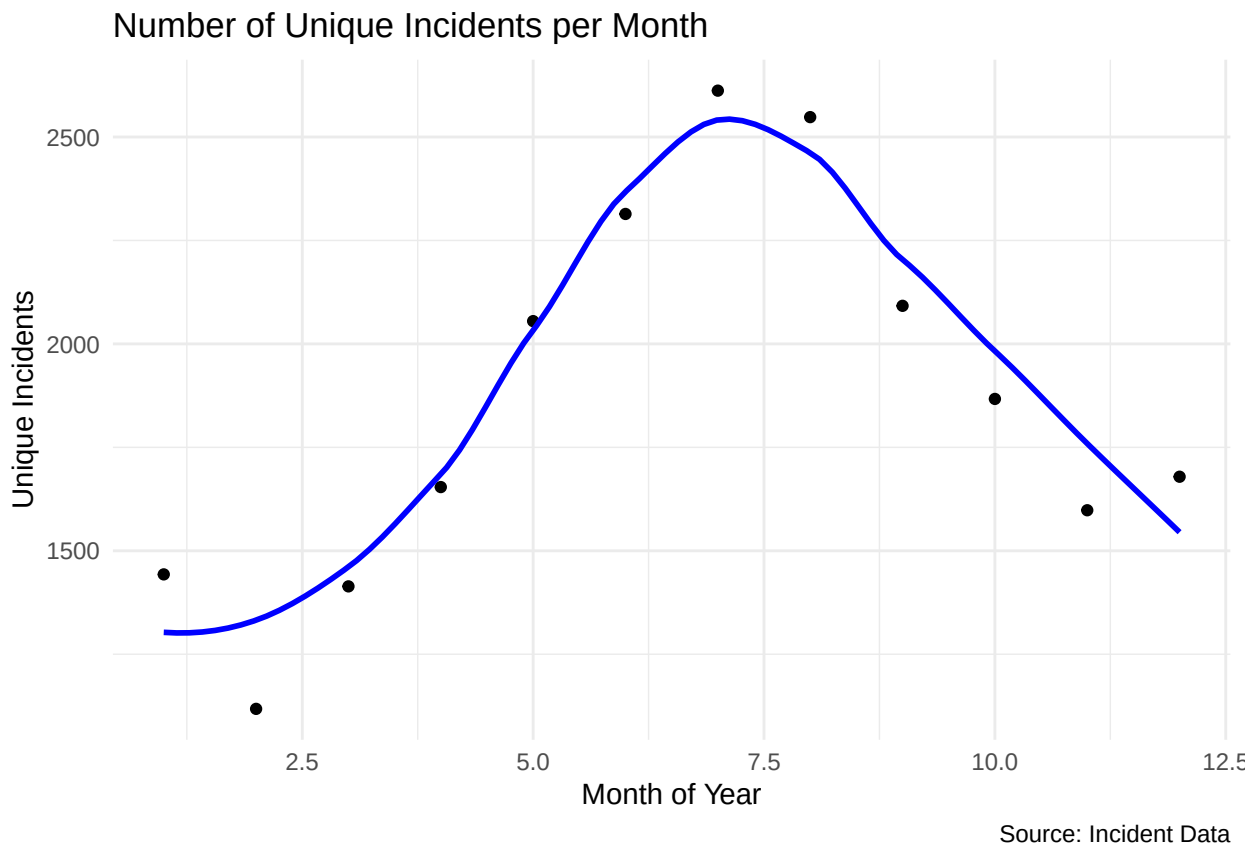
```
## `geom_smooth()` using formula = 'y ~ x'
```



There is a clear correlation between the number of incidents and the week of the year.
There is also a smaller sized dependancy likely related to week of the month.

```
df$month_of_year <- month(df$occur_date)

df_by_month <- df %>%
  group_by(month_of_year) %>%
  summarize(unique_incidents = n_distinct(incident_key), .groups = "drop")

p5 <- ggplot(df_by_month, aes(x = month_of_year, y = unique_incidents)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  labs(
    title = "Number of Unique Incidents per Month",
    x = "Month of Year",
    y = "Unique Incidents",
    caption = "Source: Incident Data"
  ) +
  theme_minimal()

print(p5)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



It is slightly smoothed for months.
The peak is in July and August, while the lowest number of incidents is in winter.
There is a small increase in the number of incidents around December and January.
Probably related to holidays.

Let's try to model the number of incendents by how far the week is from the middle of the year.

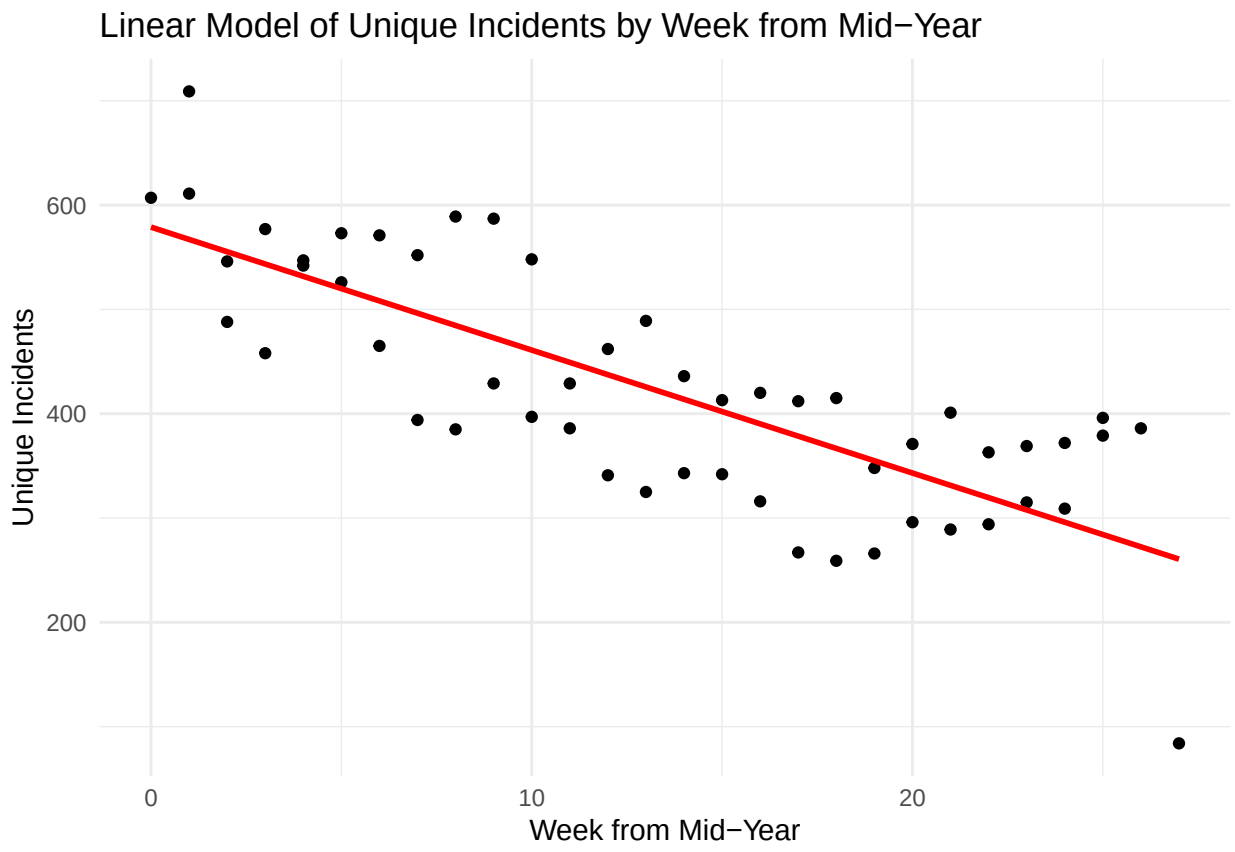
```
mid_week <- 26
df_by_week$week_from_mid_year <- abs(df_by_week$week_of_year - mid_week)
model_3 <- lm(unique_incidents ~ week_from_mid_year, data = df_by_week)

summary(model_3)
```

```
##
## Call:
## lm(formula = unique_incidents ~ week_from_mid_year, data = df_by_week)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -176.61  -63.22   10.93   53.05  141.90
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         578.884     20.046  28.878  < 2e-16 ***
## week_from_mid_year  -11.788      1.308  -9.015 3.92e-12 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73.16 on 51 degrees of freedom
## Multiple R-squared:  0.6144, Adjusted R-squared:  0.6068
## F-statistic: 81.26 on 1 and 51 DF,  p-value: 3.919e-12
```

```r
# Plot the model
ggplot(df_by_week, aes(x = week_from_mid_year, y = unique_incidents)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(
    title = "Linear Model of Unique Incidents by Week from Mid-Year",
    x = "Week from Mid-Year",
    y = "Unique Incidents"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Bias

Bias from Data

Personal Bias

Write the conclusion to your project report and include any possible sources of bias. Be sure to identify what your personal bias might be and how you have mitigated that.

(My bias may be that I do not know the situation and can consider something as less important)

Conclusion