NYPD Shooting Incident Data 2006 - 2020 Report

Paul H Bartley

2022-06-19

```
# Import libraries, set options, and define an "mmdd" subclass of the "Date" class
library(tidyverse)
library(lubridate)
library(padr)
library(broom)
library(cowplot)
library(KernSmooth)
library(tseries)
library(zoo)
library(ggfortify)
options(dplyr.summarise.inform = FALSE)
options(warn=-1)
as.mmdd <- function(x, ...) UseMethod("as.mmdd")
as.mmdd.Date <- function(x, ...) structure(x, class = c("mmdd", "Date"))
as.Date.mmdd <- function(x, ...) structure(x, class = "Date")
format.mmdd <- function(x, format = "%m-%d", ...) format(as.Date(x), format = format, ...)
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_knit$set(root.dir = getwd())
```

```
# Download .csv file and read into R as data frame object
url_in <-
  'https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD'
download.file(url_in, destfile="./NYPD_Shooting_Incident_Data_Historic_.csv")
NYPD_Shooting_Incidents_Historic <-
  read_csv("./NYPD_Shooting_Incident_Data_Historic_.csv")
```

Overview

This is a summary, visual analysis, and modeling of the **NYPD_Shooting_Incident_Data_Historic_.csv** available at https://catalog.data.gov/dataset. The dataset contains a record of all citizen shooting incidents recorded by the New York City Police Department from 2006 to 2020.

- The data were collected daily from 01/01/2006 to 12/31/2021.
- The original dataset consists of 19 variables. To consolidate the dataset for this report, the variables for geographical coordinates have been removed (*X_COORD_CD, Y_COORD_CD, Latitude, Longitude, Lon_Lat*).

```
# Transform variable types and remove geographic coordinate columns
NYPD_Shooting_Incidents_Historic <- mutate_at(NYPD_Shooting_Incidents_Historic,
                                              vars(INCIDENT_KEY, BORO,
                                            PRECINCT, JURISDICTION_CODE,
                                            LOCATION_DESC, PERP_AGE_GROUP,
                                            PERP_SEX, PERP_RACE, VIC_AGE_GROUP,
                                            VIC_SEX, VIC_RACE), as.factor) %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
  select(-c(X_COORD_CD, Y_COORD_CD, Latitude, Longitude, Lon_Lat)) %>%
  filter(OCCUR_DATE > '2005-12-31') %>%
  filter(OCCUR_DATE < '2021-01-01')
```

```
# Display variable names
names(NYPD_Shooting_Incidents_Historic)
```

```
##  [1] "INCIDENT_KEY"            "OCCUR_DATE"
##  [3] "OCCUR_TIME"              "BORO"
##  [5] "PRECINCT"                "JURISDICTION_CODE"
##  [7] "LOCATION_DESC"           "STATISTICAL_MURDER_FLAG"
##  [9] "PERP_AGE_GROUP"          "PERP_SEX"
## [11] "PERP_RACE"               "VIC_AGE_GROUP"
## [13] "VIC_SEX"                 "VIC_RACE"
```

- The variable types are as follows.
  - `INCIDENT_KEY`, `BORO`, `PRECINCT`, `JURISDICTION_CODE`, `LOCATION_DESC`, `PERP_AGE_GROUP`, `PERP_SEX`, `PERP_RACE`, `VIC_AGE_GROUP`, `VIC_SEX`, and `VIC_RACE` are all categorical variables (`<fct>` class objects in R).
  - The `OCCUR_DATE` variable is in `year-month-day` format (a `<date>` class object in R).
  - The `OCCUR_TIME` variable is in `hour:minute:second` format (a `<time>` class object in R).
  - `STASTITICAL_MURDER_FLAG` is a Boolean variable (a `<lgl>` class object in R). This variable indicates whether the victim was killed (`TRUE`) or not killed (`FALSE`) in the shooting incident.
- The dataset has been ordered by `OCCUR_DATE` and `OCCUR_TIME`.

```
NYPD_Shooting_Incidents_Historic <-  NYPD_Shooting_Incidents_Historic[order(NYPD_Shooting_Incidents_Historic$OCCUR_DATE, NYPD_Shooting_Incidents_Historic$OCCUR_TIME),]
```

- There are a total of 23585 rows in this dataset.
- Each row contains an `INCIDENT_KEY` value (a 7-9 digit number).
- If a shooting incident has only one victim, then the record for that shooting incident is stored in one row and assigned a unique `INCIDENT_KEY` value.
- If a shooting incident has multiple victims, then the records are entered as one row per victim, with all of the related rows containing the same `INCIDENT_KEY` value.

Dataset Summary

```
# Display consolidated dataset summary
summary(NYPD_Shooting_Incidents_Historic)
```

```
##     INCIDENT_KEY     OCCUR_DATE          OCCUR_TIME                    BORO
##  173354054:   18   Min.   :2006-01-01   Length:23585      BRONX         :6701
##  23749375 :   12   1st Qu.:2008-12-31   Class1:hms        BROOKLYN      :9734
##  24717013 :   12   Median :2012-02-27   Class2:difftime   MANHATTAN     :2922
##  33478089 :   12   Mean   :2012-10-05   Mode  :numeric    QUEENS        :3532
##  33706902 :   12   3rd Qu.:2016-03-02                     STATEN ISLAND: 696
##  35803777 :   12   Max.   :2020-12-31
##  (Other)  :23507
##     PRECINCT     JURISDICTION_CODE                    LOCATION_DESC
##  75     : 1375   0   :19629        MULTI DWELL - PUBLIC HOUS: 4240
##  73     : 1284   1   :   54        MULTI DWELL - APT BUILD  : 2553
##  67     : 1101   2   : 3900        PVT HOUSE                :  857
##  79     :  921   NA's:    2        GROCERY/BODEGA           :  574
##  44     :  841                    BAR/NIGHT CLUB           :  562
##  47     :  818                    (Other)                  : 1218
##  (Other):17245                    NA's                     :13581
##  STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX             PERP_RACE
##  Mode :logical           18-24  :5508   F   :  335   BLACK         :10025
##  FALSE:19085             25-44  :4714   M   :13490   WHITE HISPANIC: 1988
##  TRUE :4500              UNKNOWN:3148   U   : 1499   UNKNOWN       : 1836
##                          <18    :1368   NA's: 8261   BLACK HISPANIC: 1096
##                          45-64  : 495                WHITE         :  255
##                          (Other):  57                (Other)       :  124
##                          NA's   :8295                NA's          : 8261
##  VIC_AGE_GROUP   VIC_SEX                              VIC_RACE
##  <18    : 2525   F: 2204   AMERICAN INDIAN/ALASKAN NATIVE:    9
##  18-24  : 9003   M:21370   ASIAN / PACIFIC ISLANDER      :  327
##  25-44  :10303   U:   11   BLACK                         :16869
##  45-64  : 1541             BLACK HISPANIC                : 2245
##  65+    :  154             UNKNOWN                       :   65
##  UNKNOWN:   59             WHITE                         :  620
##                            WHITE HISPANIC                : 3450
```

Data Transformations

```
# Group by OCCUR_DATE and create an INCIDENT_COUNT column
incident_count_per_day <- NYPD_Shooting_Incidents_Historic %>%
  select(OCCUR_DATE) %>%
  group_by(OCCUR_DATE) %>%
  summarize(INCIDENT_COUNT = n(), .groups = NULL)
```

```
# Create OCCUR_YEAR and INCIDENT_COUNT columns, group by OCCUR_YEAR
incident_count_per_year <- NYPD_Shooting_Incidents_Historic %>%
  select(OCCUR_DATE) %>%
  mutate(OCCUR_YEAR = format(OCCUR_DATE, "%Y")) %>%
```

```
  group_by(OCCUR_YEAR) %>%
  summarize(INCIDENT_COUNT = n(), .groups = NULL)
```

```
# Create OCCUR_YEAR and INCIDENT_COUNT columns, group by OCCUR_YEAR and BORO
incident_count_per_year_per_BORO <- NYPD_Shooting_Incidents_Historic %>%
  select(OCCUR_DATE, BORO) %>%
  mutate(OCCUR_YEAR = format(OCCUR_DATE, "%Y")) %>%
  group_by(OCCUR_YEAR, BORO) %>%
  summarize(INCIDENT_COUNT = n(), .groups = NULL)
```

```
# Group by OCCUR_DATE and BORO and create an INCIDENT_COUNT column
incident_count_per_day_per_BORO <- NYPD_Shooting_Incidents_Historic %>%
  select(OCCUR_DATE, BORO) %>%
  group_by(OCCUR_DATE, BORO) %>%
  summarize(INCIDENT_COUNT = n(), .groups = NULL)
```

```
# Filter only STATISTICAL_MURDER_FLAG == TRUE
shooting_death_incident_count_per_day <- NYPD_Shooting_Incidents_Historic %>%
  select(OCCUR_DATE, STATISTICAL_MURDER_FLAG) %>%
  filter(STATISTICAL_MURDER_FLAG == TRUE) %>%
  group_by(OCCUR_DATE) %>%
  summarize(INCIDENT_COUNT = n(), .groups = NULL)

shooting_death_incidents <- NYPD_Shooting_Incidents_Historic %>%
  select(STATISTICAL_MURDER_FLAG) %>%
  filter(STATISTICAL_MURDER_FLAG == TRUE)
```

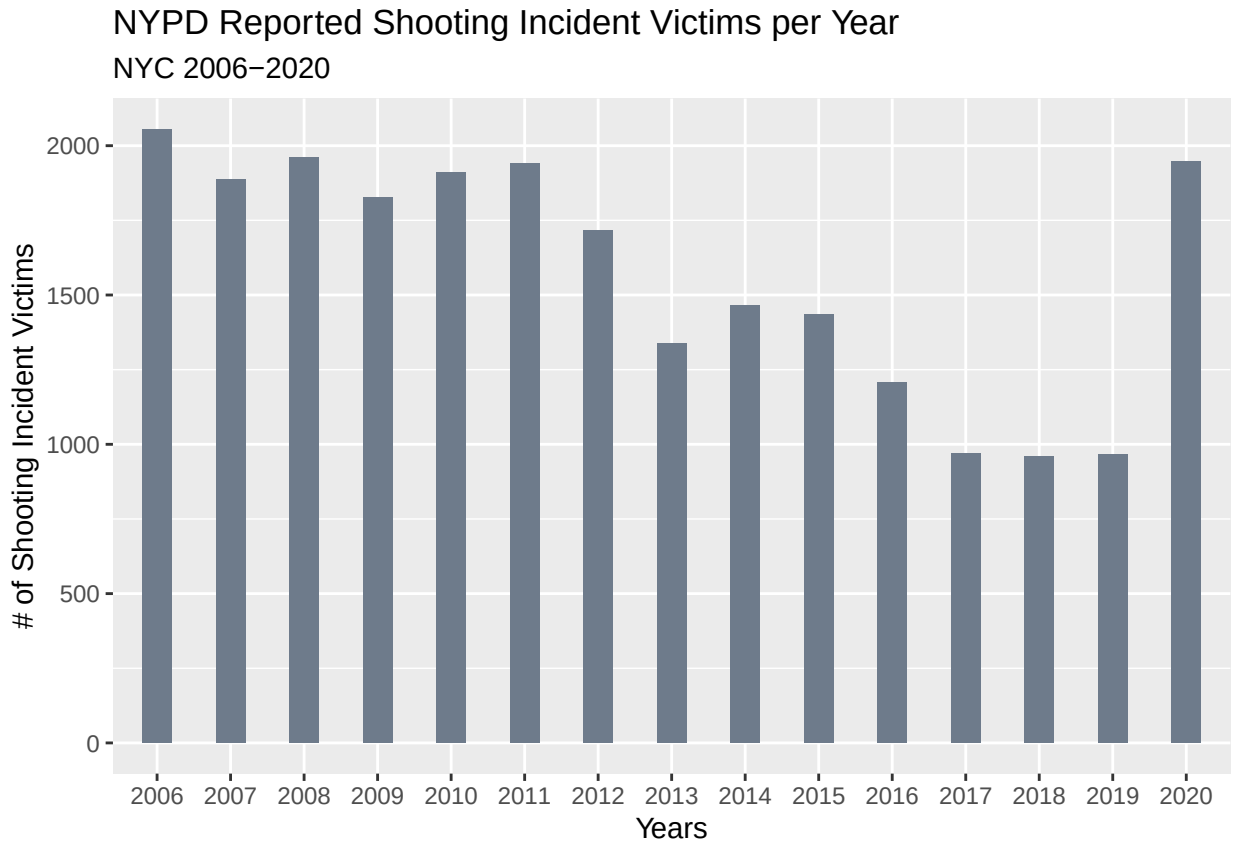
```
# Pad missing days and filter INCIDENT_COUNT == 0
incident_count_per_day_padded <- NYPD_Shooting_Incidents_Historic %>%
  select(OCCUR_DATE) %>%
  group_by(OCCUR_DATE) %>%
  summarize(INCIDENT_COUNT = n(), .groups = NULL) %>%
  pad(interval = "day") %>%
  fill_by_value(INCIDENT_COUNT, value = 0) %>%
  filter(INCIDENT_COUNT == 0)
```

```
# Add YEAR and MONTH columns and create a MONTH_DAY variable
incident_count_per_day_MONTH_DAY_YEAR <- NYPD_Shooting_Incidents_Historic %>%
  select(OCCUR_DATE) %>%
  group_by(OCCUR_DATE) %>%
  summarize(INCIDENT_COUNT = n(), .groups = NULL) %>%
  mutate(OCCUR_YEAR = year(OCCUR_DATE), OCCUR_MONTH = month(OCCUR_DATE)) %>%
  transform(MONTH_DAY = as.mmdd(OCCUR_DATE))
```

```
# Pad missing dates and add OCCUR_MONTH and OCCUR_YEAR variables
incident_count_per_day_padded_by_day <- NYPD_Shooting_Incidents_Historic %>%
  select(OCCUR_DATE) %>%
  group_by(OCCUR_DATE) %>%
  summarize(INCIDENT_COUNT = n(), .groups = NULL) %>%
  pad(interval = "day") %>%
  fill_by_value(INCIDENT_COUNT, value = 0) %>%
  mutate(OCCUR_YEAR = year(OCCUR_DATE), OCCUR_MONTH = month(OCCUR_DATE)) %>%
  select(OCCUR_DATE, OCCUR_MONTH, OCCUR_YEAR, INCIDENT_COUNT)
```

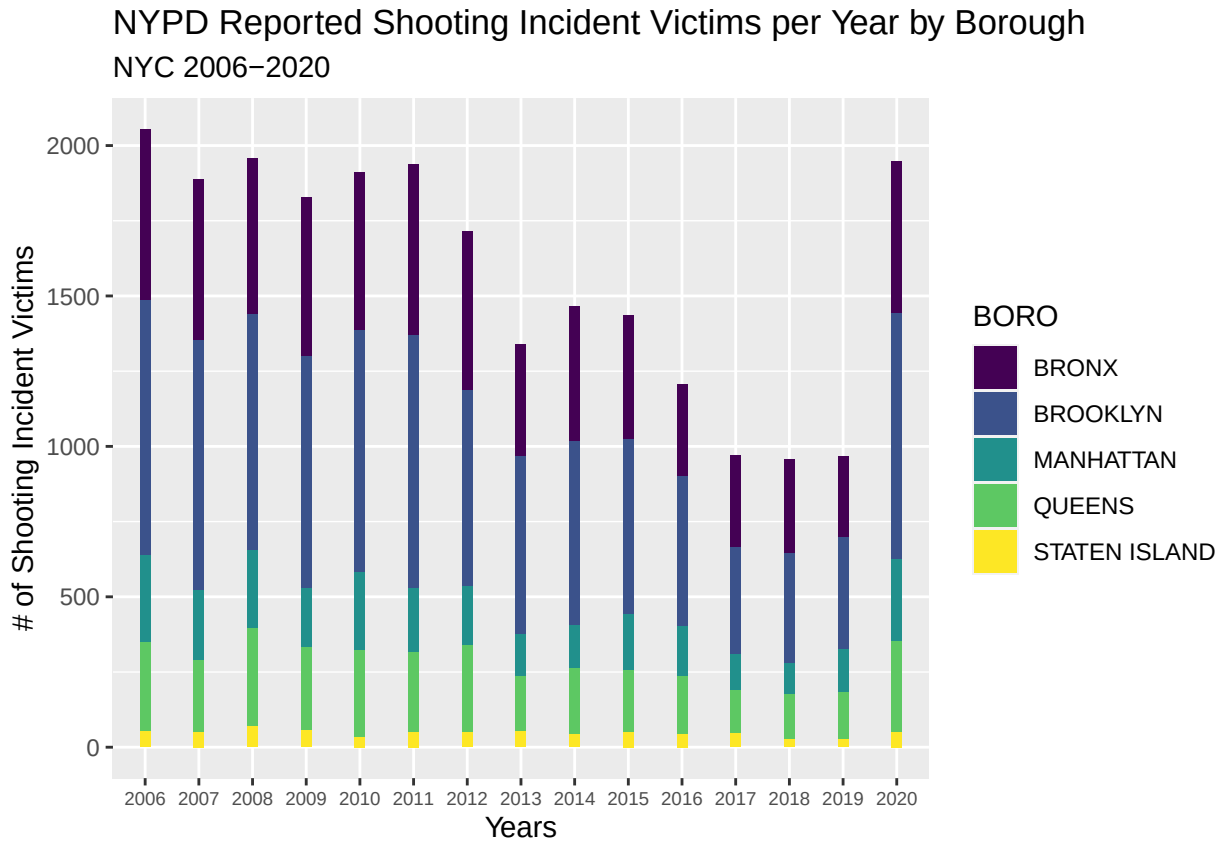
Data Visualization

```
ggplot(data = incident_count_per_year, aes(x = OCCUR_YEAR, y = INCIDENT_COUNT)) +
    geom_bar(stat = "identity", width = 0.4, fill = "lightsteelblue4") +
    labs(x = "Years",
         y = "# of Shooting Incident Victims",
         title = "NYPD Reported Shooting Incident Victims per Year",
         subtitle = "NYC 2006-2020")
```



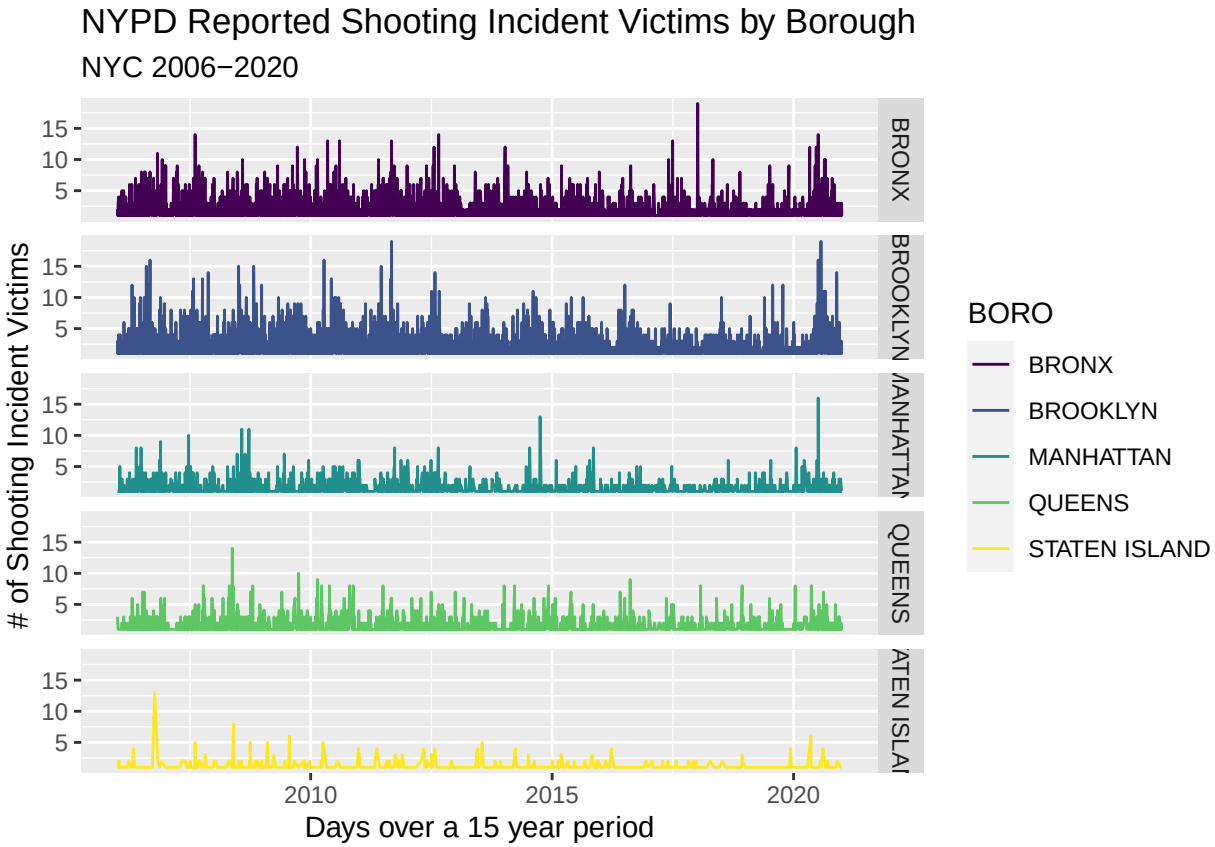
This graph shows the yearly number of shooting incidents in NYC from 2006 to 2020. A downward trend can be seen from 2011 to 2019, with a sharp increase in shooting incidents occurring in 2020.

```
ggplot(data = incident_count_per_year_per_BORO, aes(x = OCCUR_YEAR,
                                                    y = INCIDENT_COUNT,
                                                    fill = BORO)) +
    scale_fill_viridis_d() +
    geom_bar(stat = "identity", width = 0.2) +
    labs(x = "Years",
         y = "# of Shooting Incident Victims",
         title = "NYPD Reported Shooting Incident Victims per Year by Borough",
         subtitle = "NYC 2006-2020") +
    theme(axis.text.x = element_text(size = 7))
```



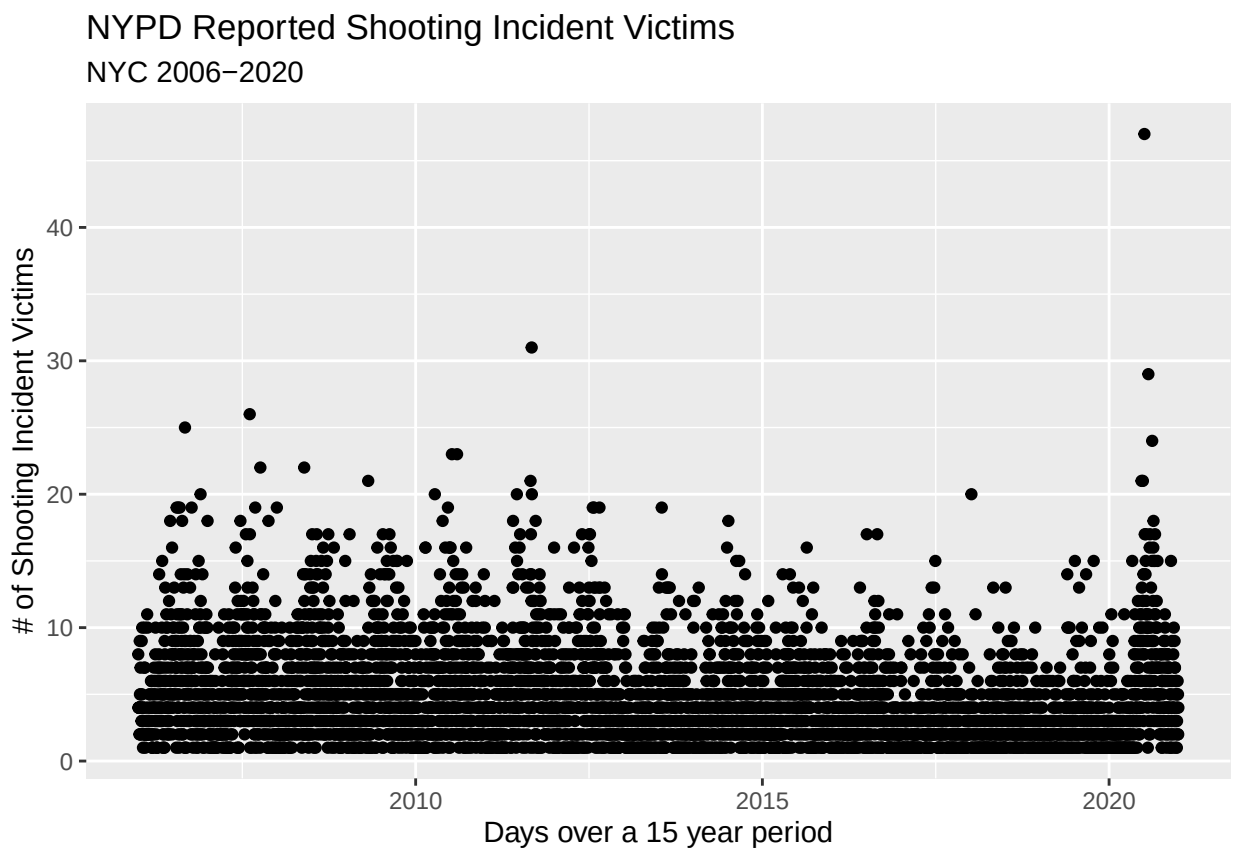
This expands on the previous graph with the per year proportion of shooting incidents shown for the five NYC boroughs.

```
ggplot(data = incident_count_per_day_per_BORO, aes(x = OCCUR_DATE,
                                                   y = INCIDENT_COUNT,
                                                   color = BORO)) +
  scale_color_viridis_d() +
  geom_line() +
  facet_grid(BORO~.) +
    labs(x = "Days over a 15 year period",
         y = "# of Shooting Incident Victims",
         title = "NYPD Reported Shooting Incident Victims by Borough",
         subtitle = "NYC 2006-2020")
```



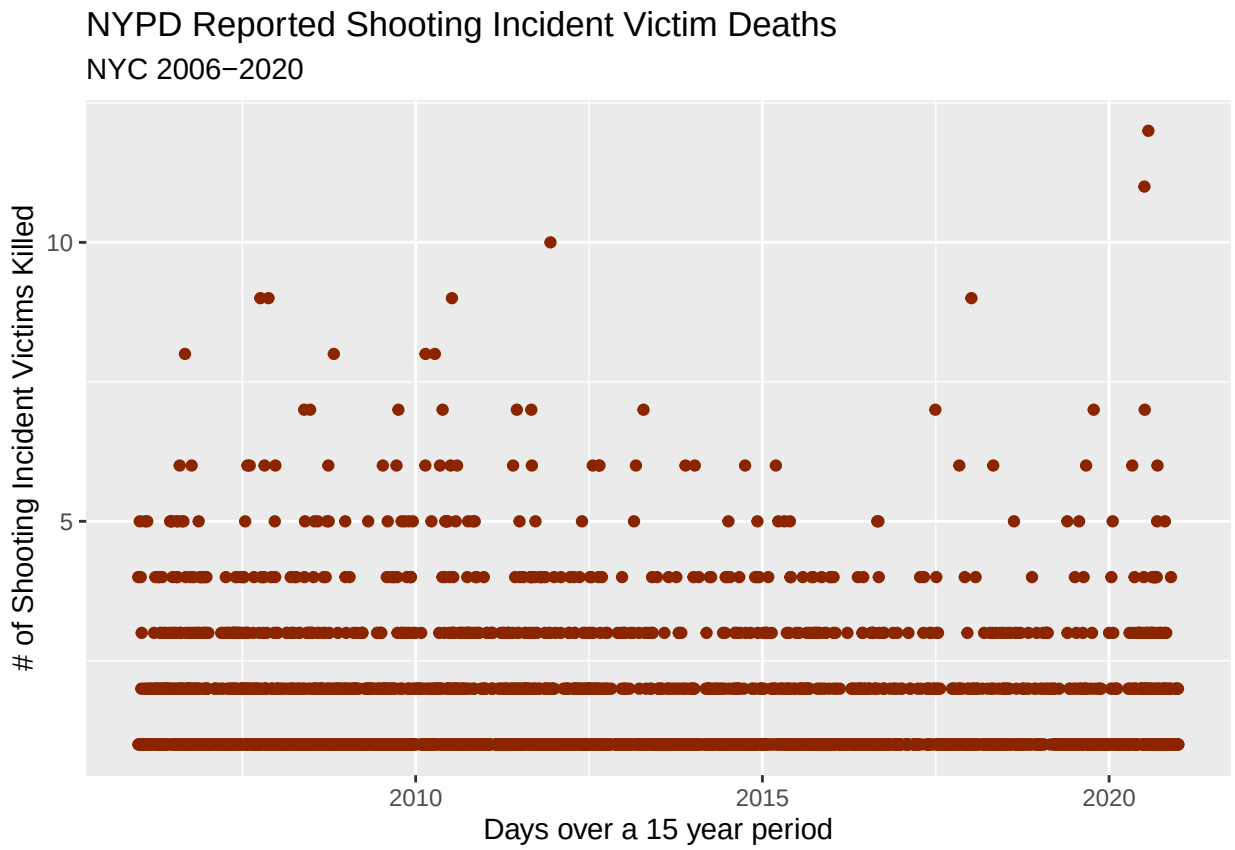
This graph displays how the number of shooting incidents compares across the five NYC boroughs on a finer time scale.

```
ggplot(data = incident_count_per_day, aes(x = OCCUR_DATE, y = INCIDENT_COUNT)) +
    geom_point() +
    labs(x = "Days over a 15 year period",
         y = "# of Shooting Incident Victims",
         title = "NYPD Reported Shooting Incident Victims",
         subtitle = "NYC 2006-2020")
```



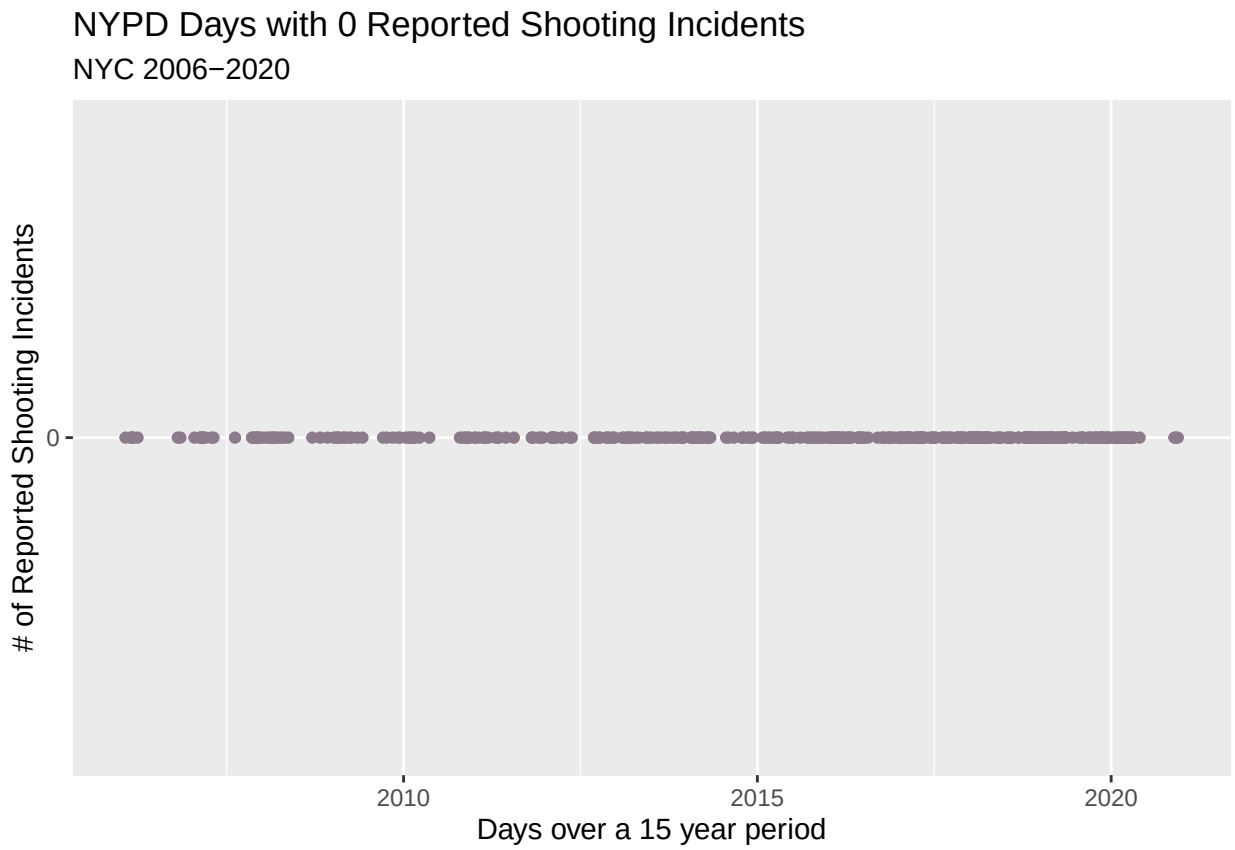
Each of the 23585 points in this graph represents a victim of a citizen shooting incident in NYC from 2006 to 2020. The graph is clearly periodic, indicating a yearly cycle where the number of shooting incidents rises and falls in sync with seasonal changes.

```
ggplot(data = shooting_death_incident_count_per_day, aes(x = OCCUR_DATE,
                                                         y = INCIDENT_COUNT)) +
    geom_point(color = "orangered4") +
    scale_y_continuous(breaks = seq(0, 15, by = 5)) +
    labs(x = "Days over a 15 year period",
         y = "# of Shooting Incident Victims Killed",
         title = "NYPD Reported Shooting Incident Victim Deaths",
         subtitle = "NYC 2006-2020")
```



This is a distillation of the previous graph. The 4500 points here represent NYC shooting deaths from 2006 to 2020.

```
ggplot(data = incident_count_per_day_padded, aes(x = OCCUR_DATE,
                                                 y = INCIDENT_COUNT)) +
    geom_point(color = "thistle4") +
    scale_y_continuous(breaks = 0) +
    labs(x = "Days over a 15 year period",
         y = "# of Reported Shooting Incidents",
         title = "NYPD Days with 0 Reported Shooting Incidents",
         subtitle = "NYC 2006-2020")
```



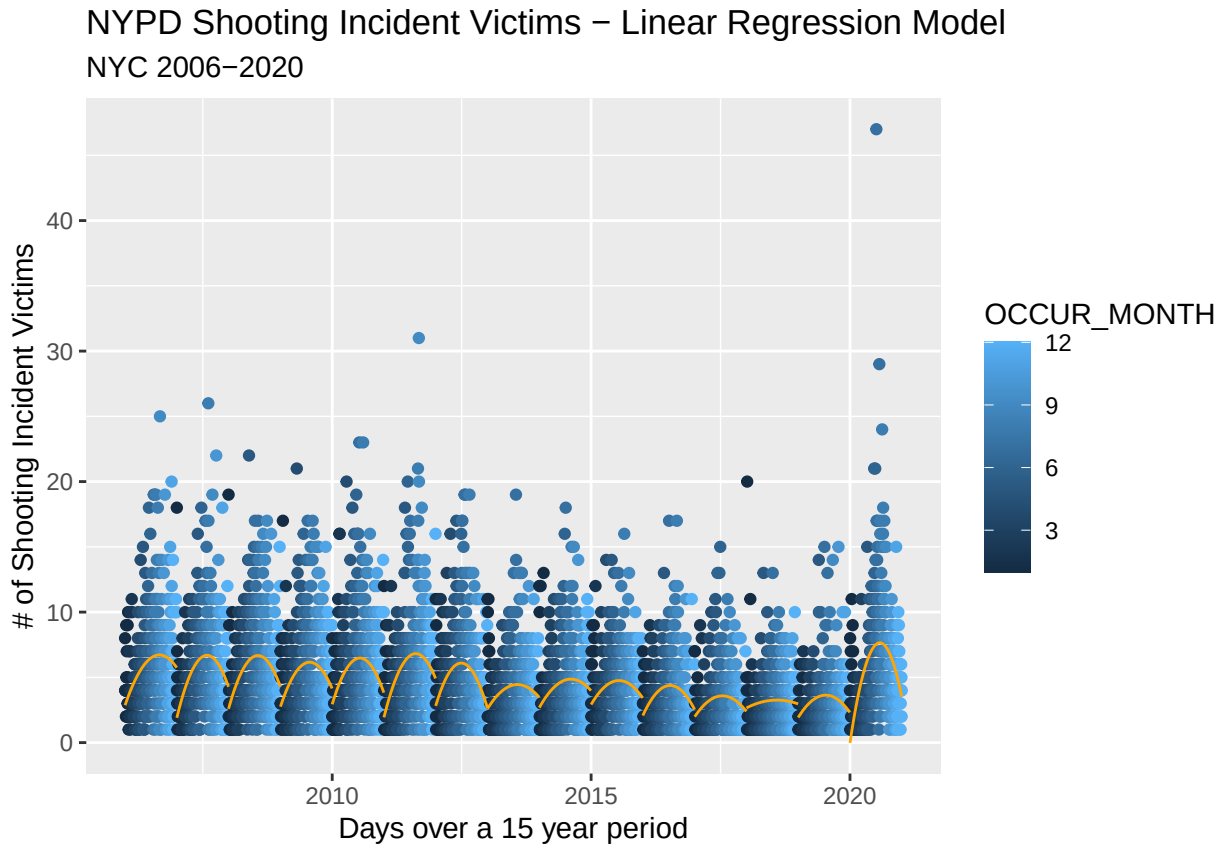
Here there are 425 points, each of which represents a day in the 2006 - 2020 time period where no shooting incidents were recorded by the NYPD.

Data Modeling

We're going to use the following question to guide our data modeling: ***How does the seasonal rise and fall of shooting incidents in NYC correlate with seasonal rise and fall of temperature in the NYC area?***

We'll begin by doing a simple linear regression model of the rising and falling number of shootings from year to year in NYC over the 15 year period.

```
ggplot(data = incident_count_per_day_MONTH_DAY_YEAR, aes(x = MONTH_DAY,
                                                         y = INCIDENT_COUNT,
                                                         group = OCCUR_YEAR,
                                                         color = OCCUR_MONTH)) +
    geom_point() +
    geom_smooth(formula = y ~ poly(x,2), method = "lm", se = FALSE,
                color = "orange", size = .5) +
    labs(x = "Days over a 15 year period",
         y = "# of Shooting Incident Victims",
         title = "NYPD Shooting Incident Victims - Linear Regression Model",
         subtitle = "NYC 2006-2020")
```

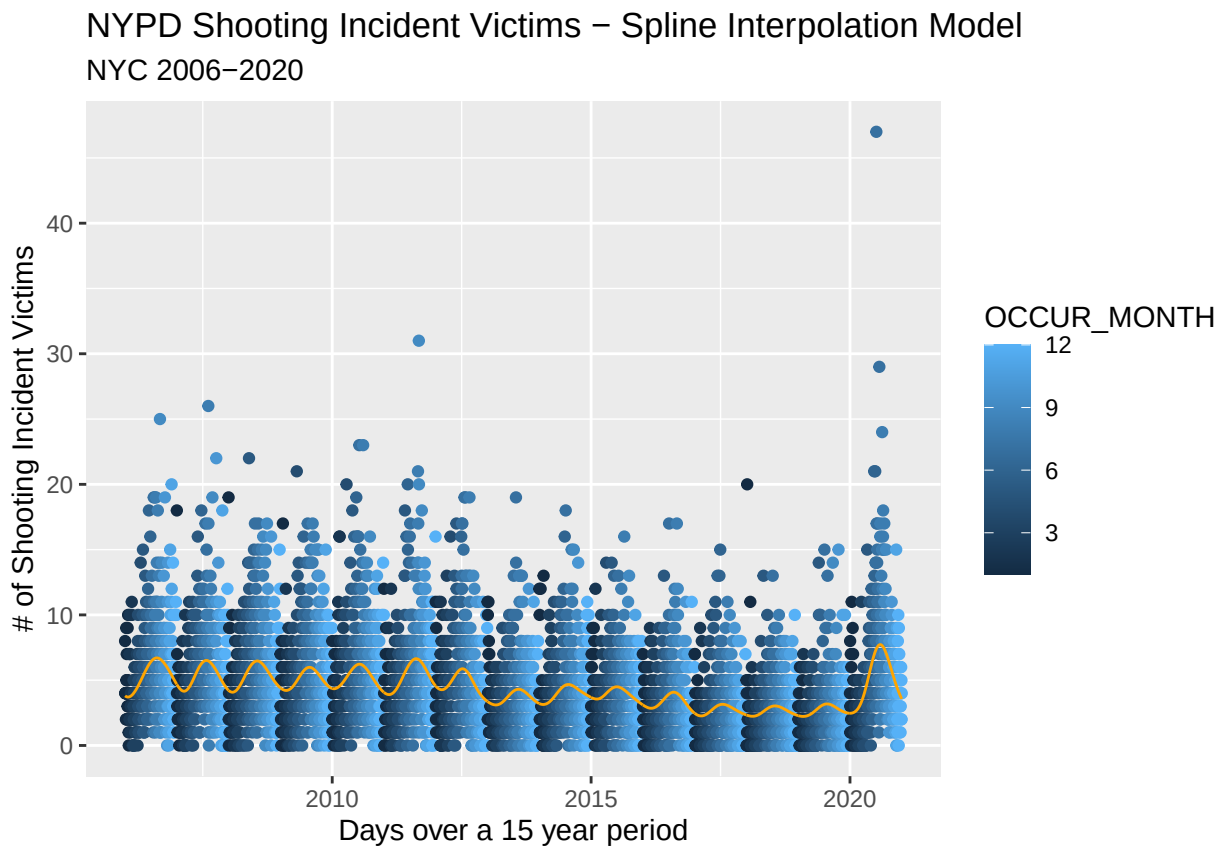


This point graph shows the total number of shooting victims in NYC from 2006 to 2020 with the addition of 15 regression lines to model the seasonal rise and fall in the number of shooting incidents. Each line is plotted with the coefficients of a second-order polynomial equation (`y = b0 + b1x + b2x^2`) calculated from that year's data. The lines approximate the expected (i.e., mean) # of shooting victims on that day of the year, in that year. Despite the quadratic curvature, this method does not fit the data with statistical significance (p-values for first- and second-order coefficients are over .05).

Next we'll look at a spline interpolation model of the daily mean # of NYC shooting incidents over the same period. We'll use the `KernSmooth` package as detailed in the R Cookbook.

```
# Set up a spline interpolation of the mean # of daily shooting incidents
t <- seq(from = 1, to = nrow(incident_count_per_day_padded_by_day))
y1 <- incident_count_per_day_padded_by_day$INCIDENT_COUNT
gridsize <- nrow(incident_count_per_day_padded_by_day)
bw1 <- dpill(t, y1, gridsize = gridsize)
lp1 <- locpoly(x = t, y = y1, bandwidth = bw1, gridsize = gridsize)
smooth1 <- lp1$y
incident_count_per_day_padded_by_day$INCIDENT_COUNT_SMOOTH = smooth1
incident_count_per_day_padded_by_day$t = t
```

```
ggplot(data = incident_count_per_day_padded_by_day, aes(x = OCCUR_DATE,
      y = INCIDENT_COUNT, group = OCCUR_YEAR, color = OCCUR_MONTH)) +
    geom_point() +
    geom_line(aes(x = OCCUR_DATE, y = smooth1), color = "orange") +
    labs(x = "Days over a 15 year period",
         y = "# of Shooting Incident Victims",
         title = "NYPD Shooting Incident Victims - Spline Interpolation Model",
         subtitle = "NYC 2006-2020")
```



As can be seen, qualitatively the spline interpolation more closely models the seasonal rise and fall of shooting incidents in NYC.

Correlation between NYPD shooting incident data and NYC temperature data

We'll now import another dataset containing daily temperature readings from 2006 to 2020 collected at JFK International Airport in NYC. Custom climate data sets like this can be specified and downloaded for free at https://www.ncdc.noaa.gov.

```
# Import daily NYC temperature data from 2006 - 2020
NYC_weather <- read_csv('./NYC_weather_2006-2020.csv')

NYC_temperatures <- NYC_weather %>%
  select(DATE, TMAX) %>%
  rename(OCCUR_DATE = 'DATE')
```

1 Compare the NYPD shooting incident and NYC temperature data

- Merge the NYPD shooting and NYC temperature data into one data frame.

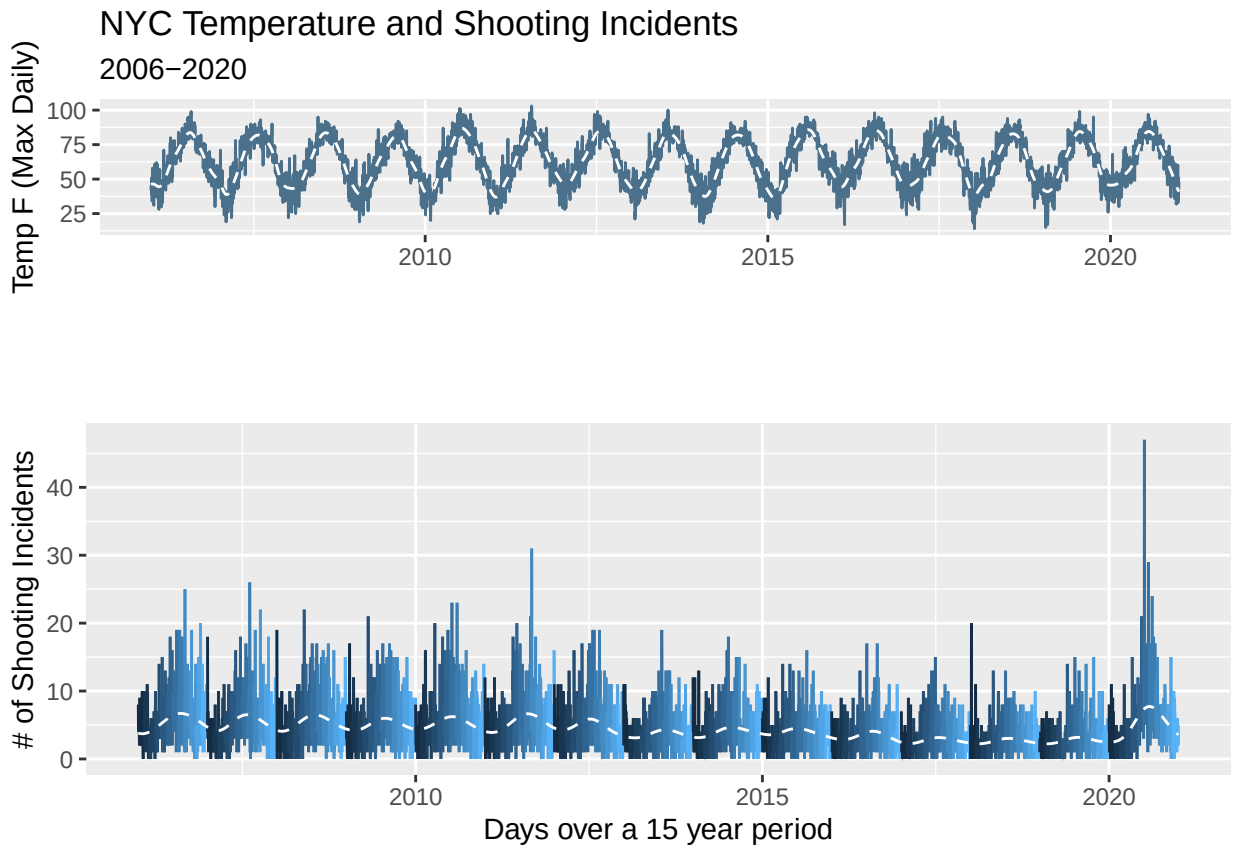
```
# Join temperature data and shooting incident data into one data frame
incident_count_with_temp_per_day <-
  full_join(incident_count_per_day_padded_by_day, NYC_temperatures, by = NULL)
```

- Run the `KernSmooth` spline interpolation model on the NYC temperature data.

```
# Calculate a spline interpolation model for NYC temperature data
y2 <- incident_count_with_temp_per_day$TMAX
bw2 <- dpill(t, y2, gridsize = gridsize)
lp2 <- locpoly(x = t, y = y2, bandwidth = bw2, gridsize = gridsize)
smooth2 <- lp2$y
incident_count_with_temp_per_day$TMAX_SMOOTH = smooth2
```

- Plot both the temperature and shooting incident data together.

```
temp_plot = ggplot(data = incident_count_with_temp_per_day, aes(x = OCCUR_DATE,
                                                                y = TMAX)) +
  geom_line(color = "skyblue4") +
  geom_line(aes(x = OCCUR_DATE, y = smooth2), linetype = 2, color = "white") +
  labs(x = "",
         y = "Temp F (Max Daily)",
         title = "NYC Temperature and Shooting Incidents",
         subtitle = "2006-2020") +
  theme(legend.position="none")

shootings_plot = ggplot(data = incident_count_with_temp_per_day, aes(x = OCCUR_DATE,
                      y = INCIDENT_COUNT, color = OCCUR_MONTH)) +
  geom_line() +
  geom_line(aes(x = OCCUR_DATE, y = smooth1), linetype = 2, color = "white") +
  labs(x = "Days over a 15 year period",
         y = "# of Shooting Incidents",
         title = "",
         subtitle = "") +
  theme(legend.position="none")

plot_grid(temp_plot, shootings_plot, ncol = 1, rel_heights = c(3,5))
```

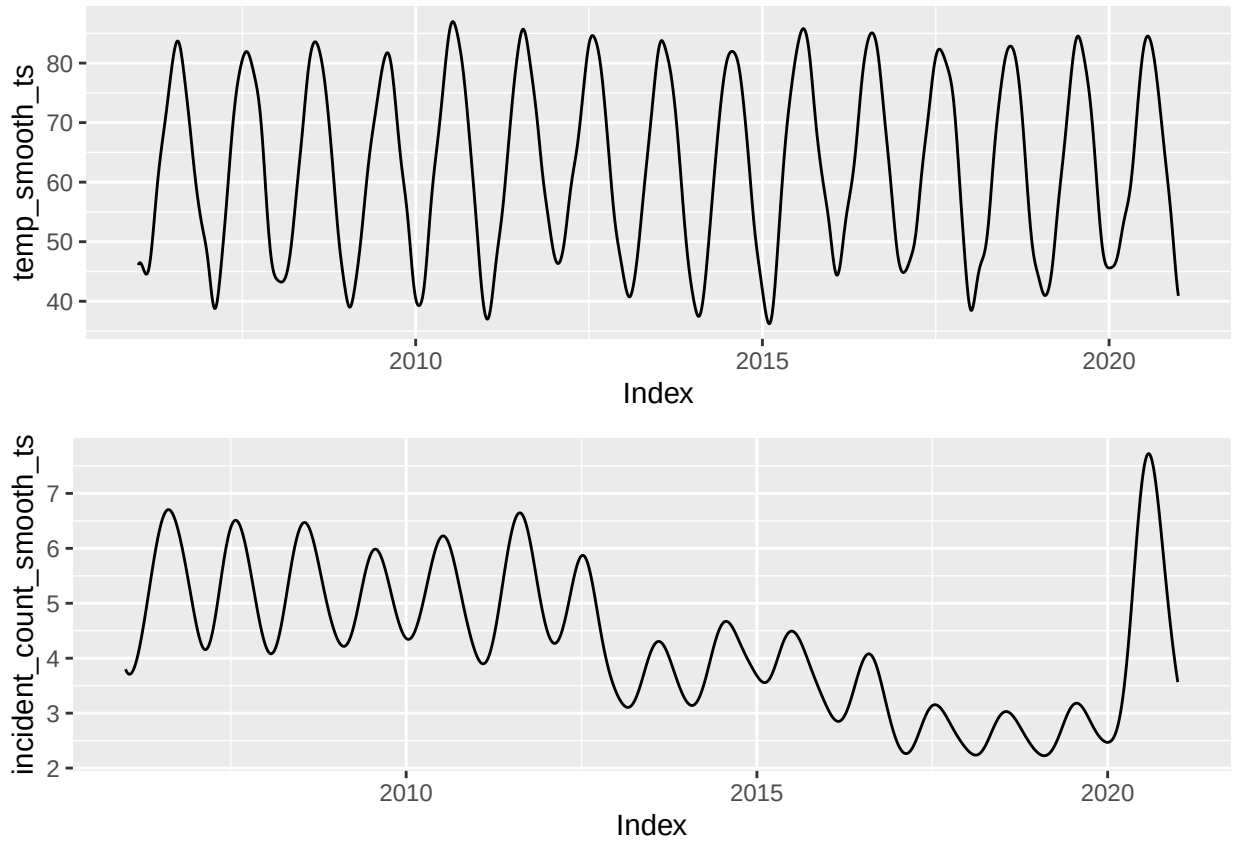


Above we see nearly time-aligned plots of the NYC temperature and shooting incident data. The dashed lines show the interpolated mean values over the 15-year period. The sinusoidal pattern in the temperature data is relatively stationary whereas the pattern in the shooting incident data is sinusoidal but non-stationary - exhibiting a decreasing and then sharply increasing trend.

2 Convert the data into time series objects in R and further compare graphs

```
# Create incident_count_smooth data frame
incident_count_smooth <- incident_count_with_temp_per_day %>%
  select(OCCUR_DATE, INCIDENT_COUNT_SMOOTH)

#Create incident_count_smooth_ts time series object
incident_count_smooth_ts <- read.zoo(incident_count_smooth)

# Define the start, end, and frequency for the time series object
incident_count_smooth_ts_zoo <- zooreg(incident_count_smooth_ts,
    start = c(2006, 1), end = c(2020, 12), frequency = 365)

# Create temp_smooth data frame
temp_smooth <- incident_count_with_temp_per_day %>%
  select(OCCUR_DATE, TMAX_SMOOTH)

#Create temp_smooth_ts time series object
temp_smooth_ts <- read.zoo(temp_smooth)
```

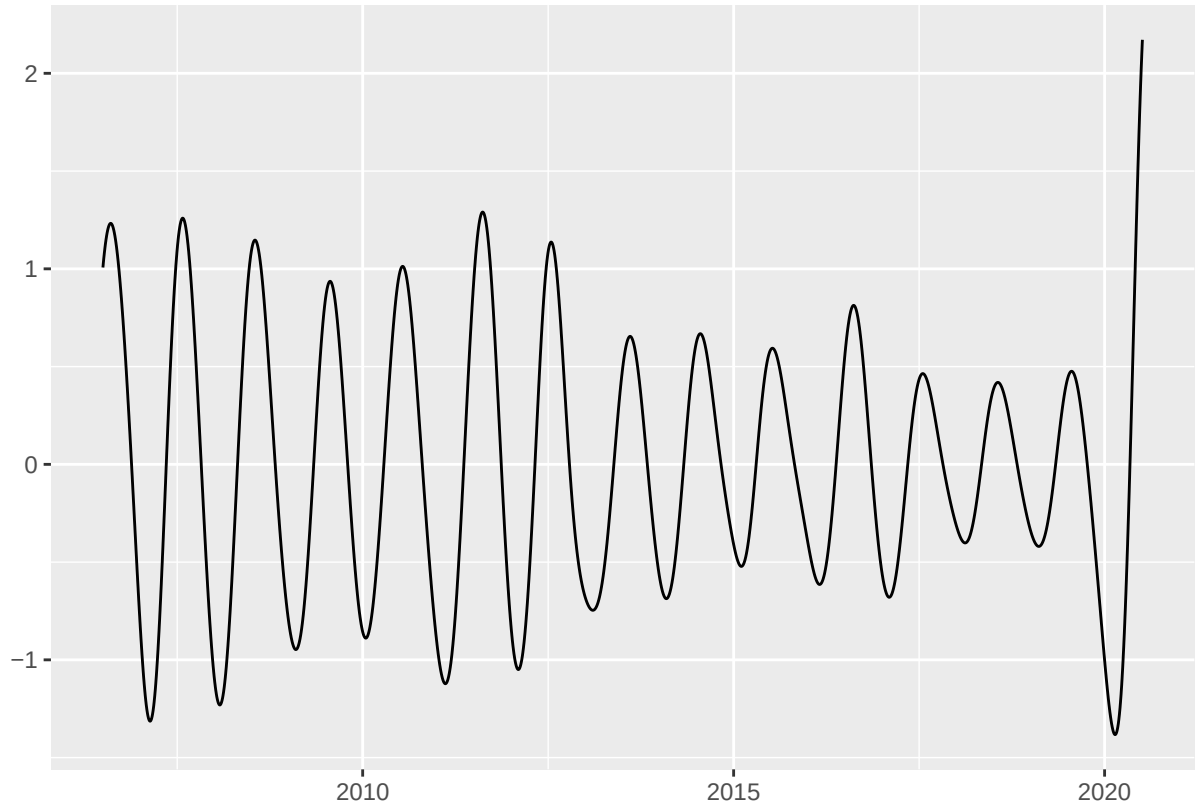
```r
# Define the start, end, and frequency for the time series object
temp_smooth_ts_zoo <- zooreg(temp_smooth_ts,
    start = c(2006, 1), end = c(2020, 12), frequency = 365)

# Define plot of the smoothed incident_count_smooth_ts_zoo time series object
shootings_plot2 = autoplot(incident_count_smooth_ts)

# Define plot of the smoothed temp_smooth_ts_zoo time series object
temp_plot2 = autoplot(temp_smooth_ts)

# Plot both graphs in alignment with each other
plot_grid(temp_plot2, shootings_plot2, ncol = 1)
```



Here we can clearly see the seasonality of the sinusoidal patterns in both graphs, with the non-stationary trend in the shooting incident graph visible toward the latter half of the time period. We are interested in calculating the cross-correlation between these two time series, so we will de-trend the data to make both time series stationary.

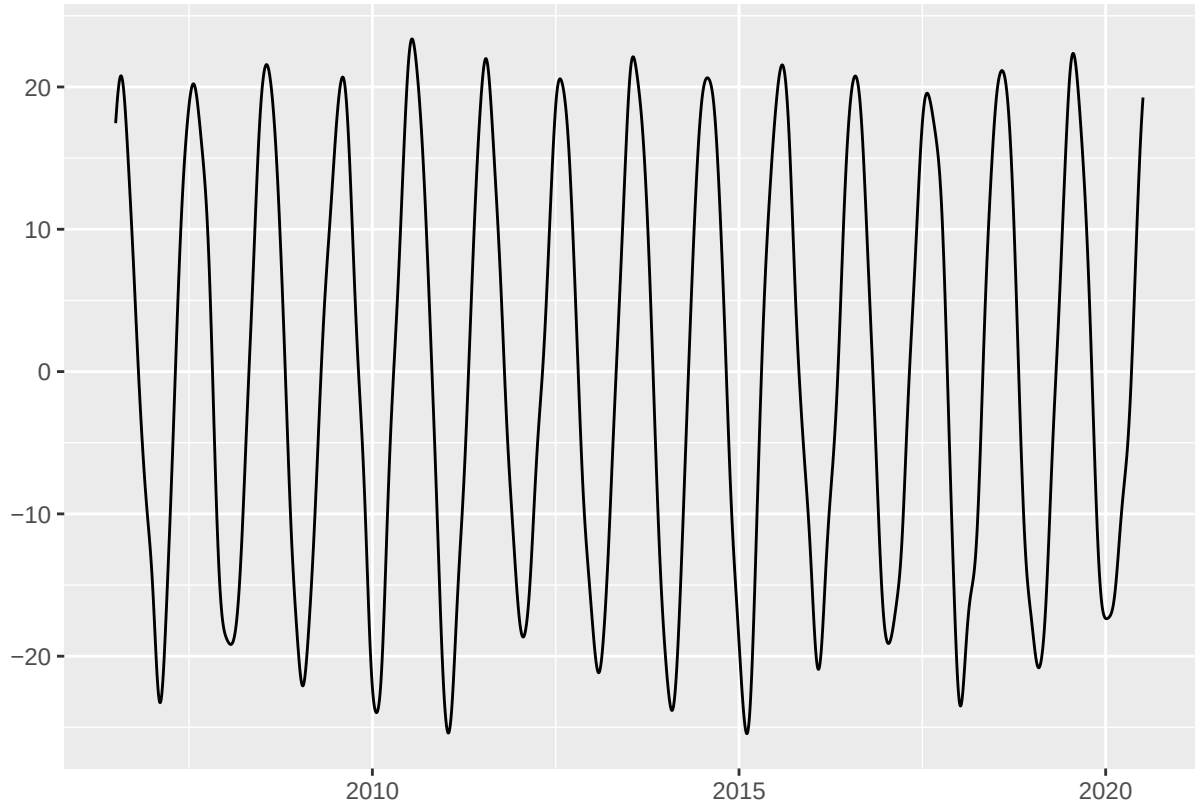
3 Perform data transformations to de-trend the temperature and shooting incident data

```
# Decompose shooting incident data to extract trend
incident_count_smooth_ts_zoo_decompose <- decompose(incident_count_smooth_ts_zoo)

# Define time series containing shooting incident data trend
incident_count_smooth_ts_zoo_decompose_trend <-
  incident_count_smooth_ts_zoo_decompose$trend

# De-trend the shooting incident data by subtracting the trend
incident_count_smooth_ts_zoo_detrend <- incident_count_smooth_ts_zoo -
  incident_count_smooth_ts_zoo_decompose_trend

# Remove leading and trailing NA values added in trend decomposition
incident_count_smooth_ts_zoo_detrend_na_remove <-
  na.remove(incident_count_smooth_ts_zoo_detrend)

# Plot the de-trended shooting incident data
shootings_plot3 = autoplot(incident_count_smooth_ts_zoo_detrend_na_remove)
autoplot(incident_count_smooth_ts_zoo_detrend_na_remove)
```



Shown above is a graph of the de-trended shooting incident data. Notice that some of the data at the beginning and end have been removed. This is an artifact of the trend detection algorithm (from the `decompose` function, which is part of the `stats` package in R). To remove the trend from the data, the trend is first extracted and then simply subtracted from the smoothed time series.

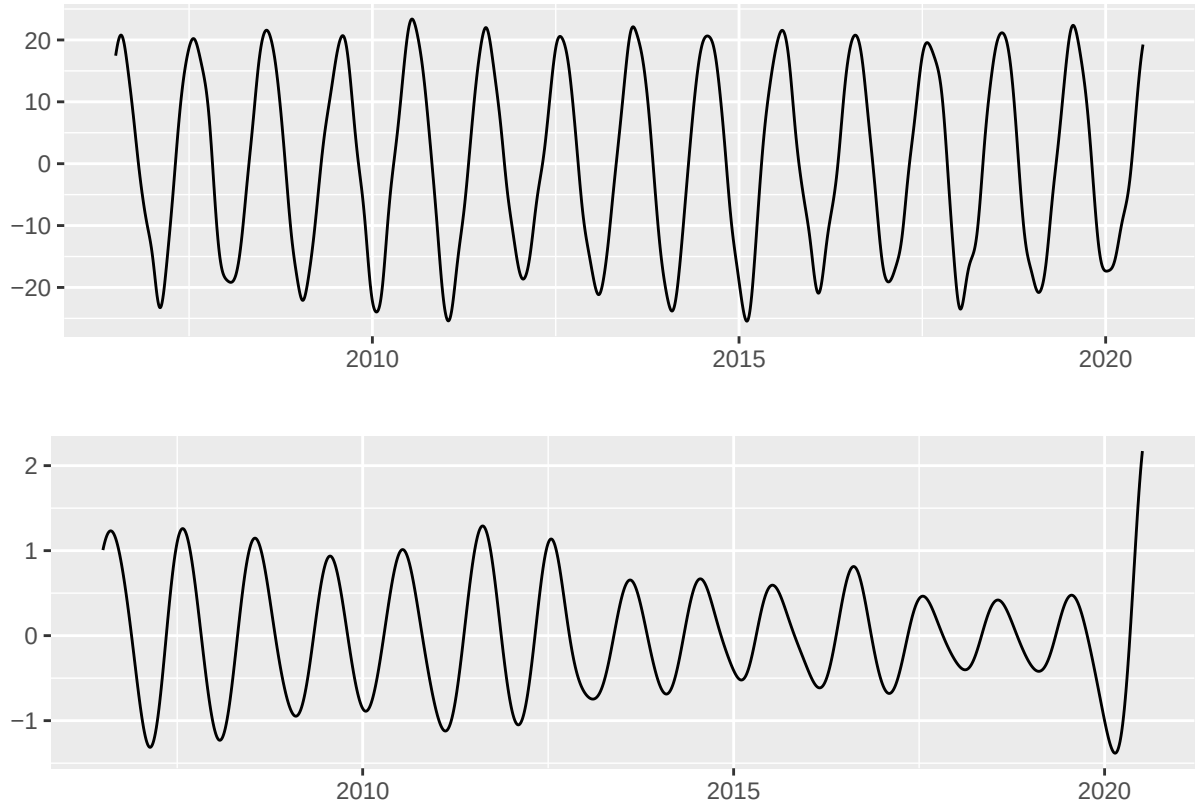
```
# Decompose temperature data to extract trend
temp_smooth_ts_zoo_decompose <- decompose(temp_smooth_ts_zoo)

# Define time series containing temperature data trend
temp_smooth_ts_zoo_decompose_trend <- temp_smooth_ts_zoo_decompose$trend

# De-trend the temperature data by subtracting the trend
temp_smooth_ts_zoo_detrend <- temp_smooth_ts_zoo - temp_smooth_ts_zoo_decompose_trend

# Remove leading and trailing NA values added in trend decomposition
temp_smooth_ts_zoo_detrend_na_remove <- na.remove(temp_smooth_ts_zoo_detrend)

# Plot the de-trended temperature data
temp_plot3 = autoplot(temp_smooth_ts_zoo_detrend_na_remove)
autoplot(temp_smooth_ts_zoo_detrend_na_remove)
```



Here we see a graph of the de-trended temperature data. The temperature data was already fairly stationary, but for the purposes of calculating the cross-correlation, it's best to use the same de-trending algorithm for both time series. Notice also that some of the data in the beginning and end have been removed.

4 Compare the de-trended shooting incident and temperature time series data

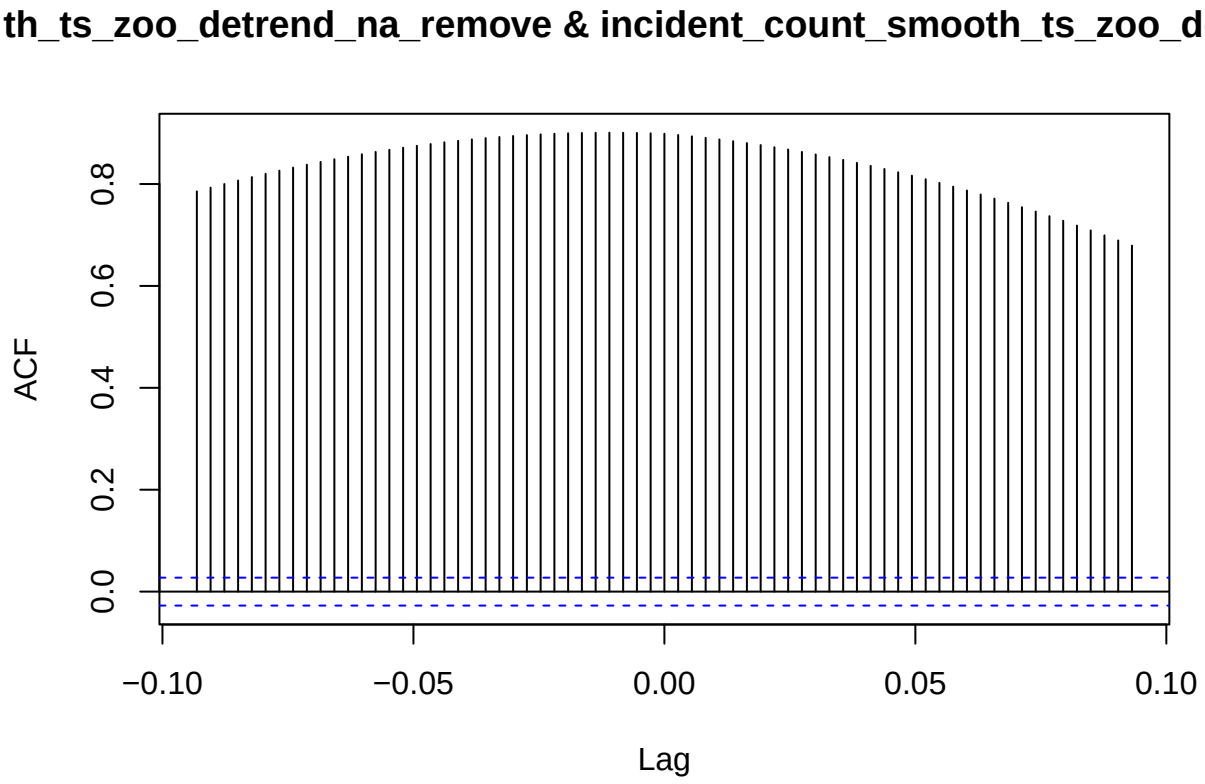
```
# Plot both detrended graphs in alignment with each other
plot_grid(temp_plot3, shootings_plot3, ncol = 1)
```



With both time series de-trended and centered around 0, we can visually assess how their sinusoidal patterns compare with each other. If you look closely, you will observe that the shooting incident peaks and troughs occur slightly later in time than the temperature peaks and troughs. This lag is especially pronounced around the 2020 mark.

5 Calculate cross-correlation lag between the temperature and shooting incident data and plot results

```
# Calculate lag/lead time for max correlation
# Find_Max_CCF function source: https://stackoverflow.com/questions/59389478/how-to-find-
# the-maximum-lag-in-a-list-of-cross-correlations-in-r-ccf
Find_Max_CCF<- function(a,b)
{
  d <- ccf(a, b, plot = FALSE)
  cor = d$acf[,,1]
  lag = d$lag[,,1]
  res = data.frame(cor,lag)
  res_max = res[which.max(res$cor),]
  return(res_max)
}

ccf(temp_smooth_ts_zoo_detrend_na_remove, incident_count_smooth_ts_zoo_detrend_na_remove)
```



In this cross correlation lag graph, we see the correlation values between the temperature and shooting incident data calculated for different amounts of lag. At 0.00 on the x-axis, the cross correlation value is calculated for no lag between the two time series.

Notice that the cross correlation value peaks at about -.011 on the x-axis. We can calculate the precise x-axis value for the peak correlation using the `Find_Max_CCF` function defined above.

```
# Run the Find_Max_CCF function to compute the lag/lead time for max x-correlation
Find_Max_CCF(temp_smooth_ts_zoo_detrend_na_remove, incident_count_smooth_ts_zoo_detrend_na_remove)
```

```
##          cor        lag
## 31 0.9008067 -0.0109589
```

From the `Find_Max_CCF` function output, we see that the peak correlation between the two time series is .9008067. This is a qualitatively "high" correlation (1 being the maximum).

We also see a lag value of -0.0109589, which represents (-0.0109589 * 365) or -3.999999 days of lag between the NYC temperature sinusoidal pattern and the shooting data sinusoidal pattern. This indicates that the highest temporal correlation is at approximately -4 days of lag* between the temperature and shooting incident data.

> *Note: Negative lag values represent a temporal lead. Hence, the temperature data "leads" the shooting incident data by approximately 4 days.

Potential bias in this report

There is some potential bias in the analysis and data visualizations I have presented in this report. My personal view on the subject of gun violence is that gun violence is a problem in NYC and in the broader U.S., and despite my attempts at being neutral on the subject, my personal viewpoint may have come through in the way I chose to depict the data in this report.

Other sources of bias:

- Only the maximum daily temperature was used from the NYC temperature data.
- The `KernSmooth` algorithm has removed the higher frequency information from both time series. If the higher frequency information were left in, the cross-correlation value would be lower.
- Subtracting the trend introduces additional artifacts in the time series data.

References

https://catalog.data.gov/dataset
https://www.ncdc.noaa.gov
https://cran.r-project.org/web/packages/KernSmooth/index.html
https://rc2e.com/timeseriesanalysis#recipe-id195
https://en.wikipedia.org/wiki/Cross-correlation
https://online.stat.psu.edu/stat510/lesson/8/8.2
https://stackoverflow.com/questions/59389478/how-to-find-the-maximum-lag-in-a-list-of-cross-correlations-in-r-ccf