NYPD Project: The Best Time to Get Shot

What was **the best time to get shot** in City That Never Sleeps? We are about to find the answer by looking at the NYPD's data, and it might not be when you would expect.

We will discover when you were most likely to:

- Get shot
- Not get shot
- Die as a result
- Survive getting shot

This project is broken down into 4 steps.

1. Get the data
2. Convert the data into something useful
3. Present the data
4. Identify possible biases

Step 1: Import data in a way that's reproducible

1. Install all the packages used in this project and load the corresponding libraries
2. Use the tidyverse package to read the csv directly from the source
3. The data source is provided by the City of New York in the data section of their web site
4. The exact URL is currently `https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLO`

```
##
## The downloaded binary packages are in
##  /var/folders/h4/0yj9g1kx5bvgbvsm3b1209tr0000gn/T//RtmpcJ2KDg/downloaded_packages

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##
## Attaching package: 'scales'
##
##
## The following object is masked from 'package:purrr':
```

```
##
##     discard
##
##
## The following object is masked from 'package:readr':
##
##     col_factor
##
##
## Rows: 27312 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.
```

Step 2: Tidy and Transform your data

1. Filter out rows with UNKNOWN data or missing location information
2. Select only the columns used in this analysis
3. Create additional columns converting existing data into useful numeric data for:

- STATISTICAL_MURDER_FLAG
- OCCUR_TIME
- OCCUR_DATE

4. create summaries of the used data
5. group and split the data by months and hours for further examination
6. calculate the mean of each category
7. display the summary tables of data

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'year_month_occured'. You can override
## using the `.groups` argument.
```

The Summary Table by Year

```
## # A tibble: 6 x 4
##   year_occured shootings murders murder_rate
##   <chr>            <int>   <dbl>       <dbl>
## 1 2006              1524     317       0.208
## 2 2007              1359     233       0.171
## 3 2008              1414     236       0.167
## 4 2009              1096     224       0.204
## 5 2010              1045     214       0.205
## 6 2011               938     215       0.229
```

The Summary Table by Year and Month

```
## # A tibble: 6 x 6
## # Groups:   year_month_occured [1]
##   year_month_occured shootings murders month_occured year_occured murder_rate
##   <chr>                  <int>   <dbl>         <dbl> <chr>               <dbl>
## 1 2006-01                   99      22             1 2006                0.222
## 2 2006-01                   99      22             1 2006                0.222
## 3 2006-01                   99      22             1 2006                0.222
## 4 2006-01                   99      22             1 2006                0.222
## 5 2006-01                   99      22             1 2006                0.222
## 6 2006-01                   99      22             1 2006                0.222
```

The Summary Table grouped by Month

```
## # A tibble: 6 x 4
##   month_occured shootings murders murder_rate
##           <dbl>     <int>   <dbl>       <dbl>
## 1             1      1107     241       0.218
## 2             2       863     184       0.213
## 3             3      1093     215       0.197
## 4             4      1280     281       0.220
## 5             5      1572     367       0.233
## 6             6      1632     320       0.196
```

The Summary Table grouped by Hour

```
## # A tibble: 6 x 4
##   hour_occured shootings murders murder_rate
##          <int>     <int>   <dbl>       <dbl>
## 1            0      1146     229       0.200
## 2            1      1105     217       0.196
## 3            2       947     173       0.183
## 4            3       884     186       0.210
## 5            4       761     181       0.238
## 6            5       397     111       0.280
```

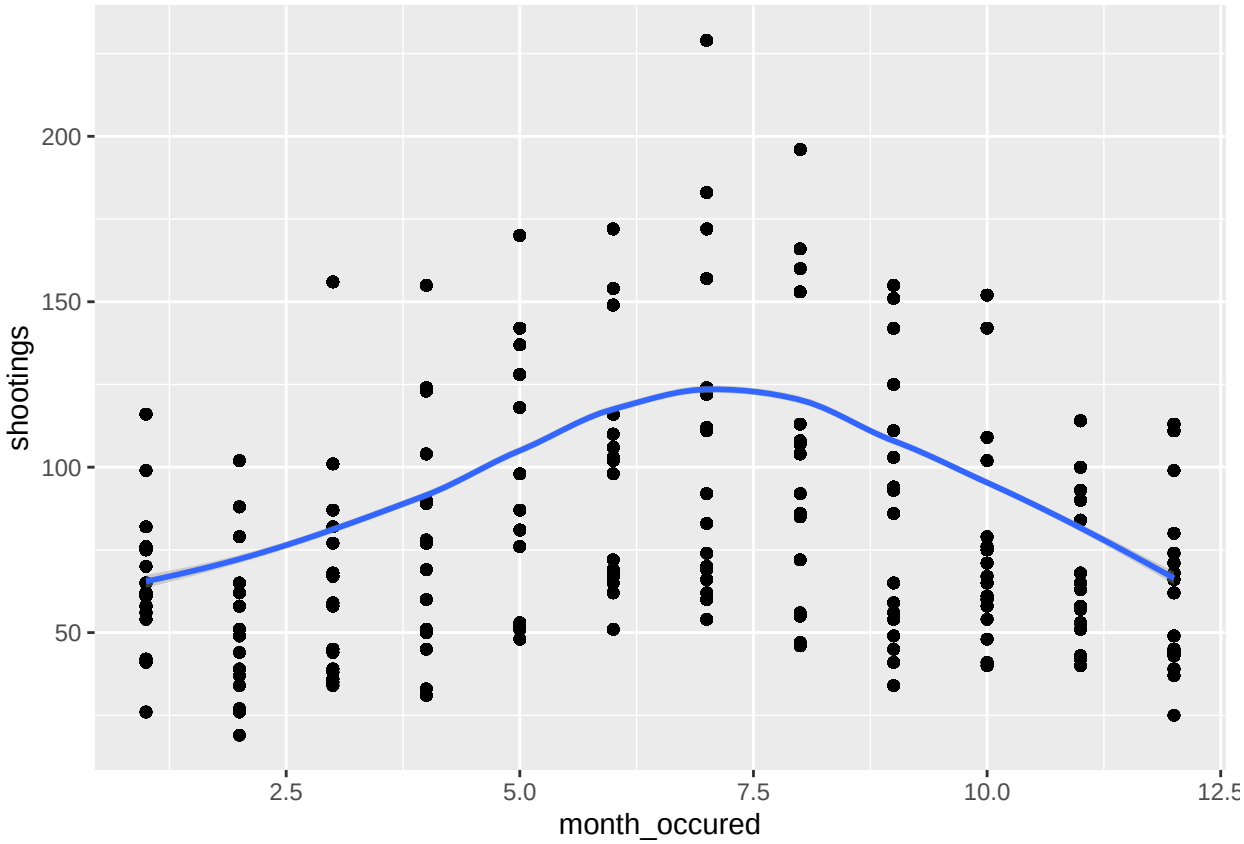
Step 3: Add Model, Visualizations, and Analysis

Model

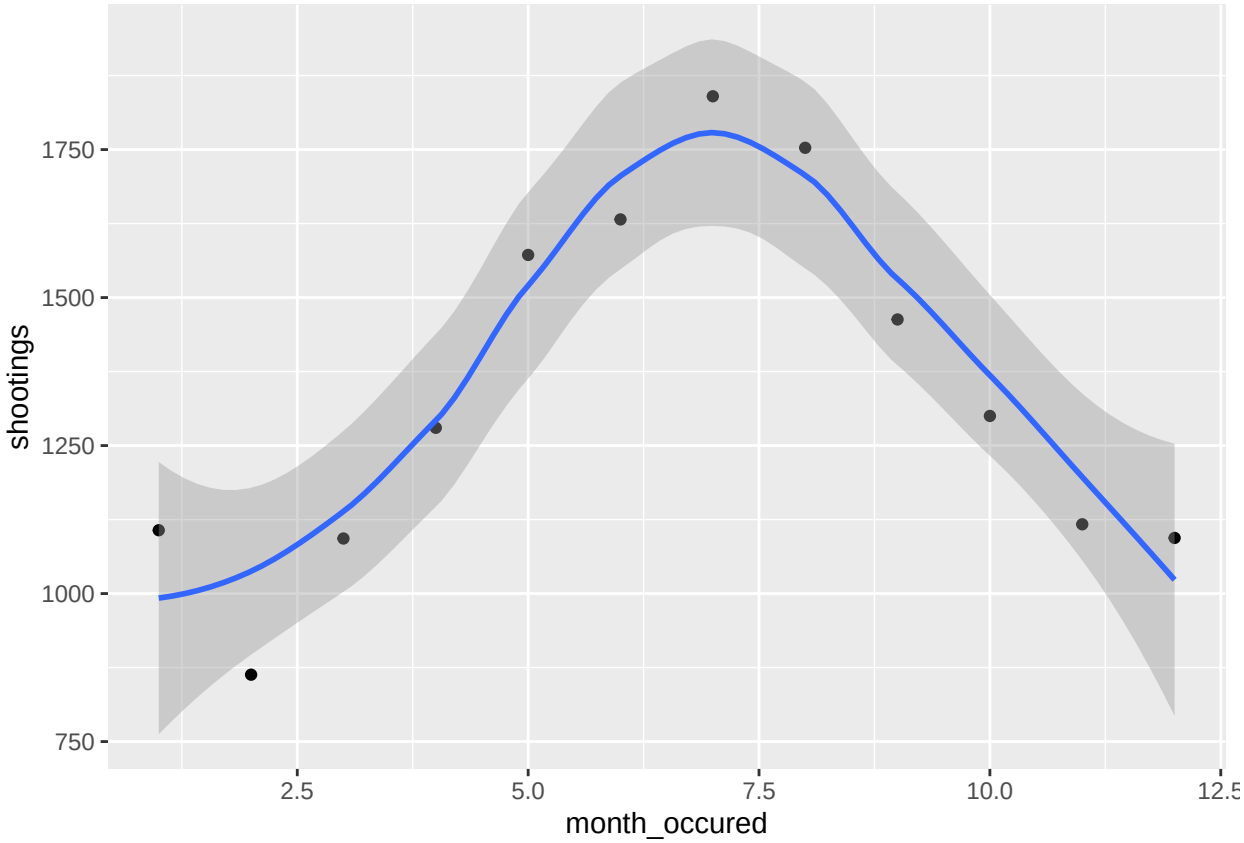
For a statistical model I'm used Local Polynomial Regression (called 'loess' in ggplot). The benefit is it can make a smooth curve that reflects the data without having to iterate over different degree polynomials and other formulas to best fit the data. The cost is it can be computationally intensive.

Monthly Shootings

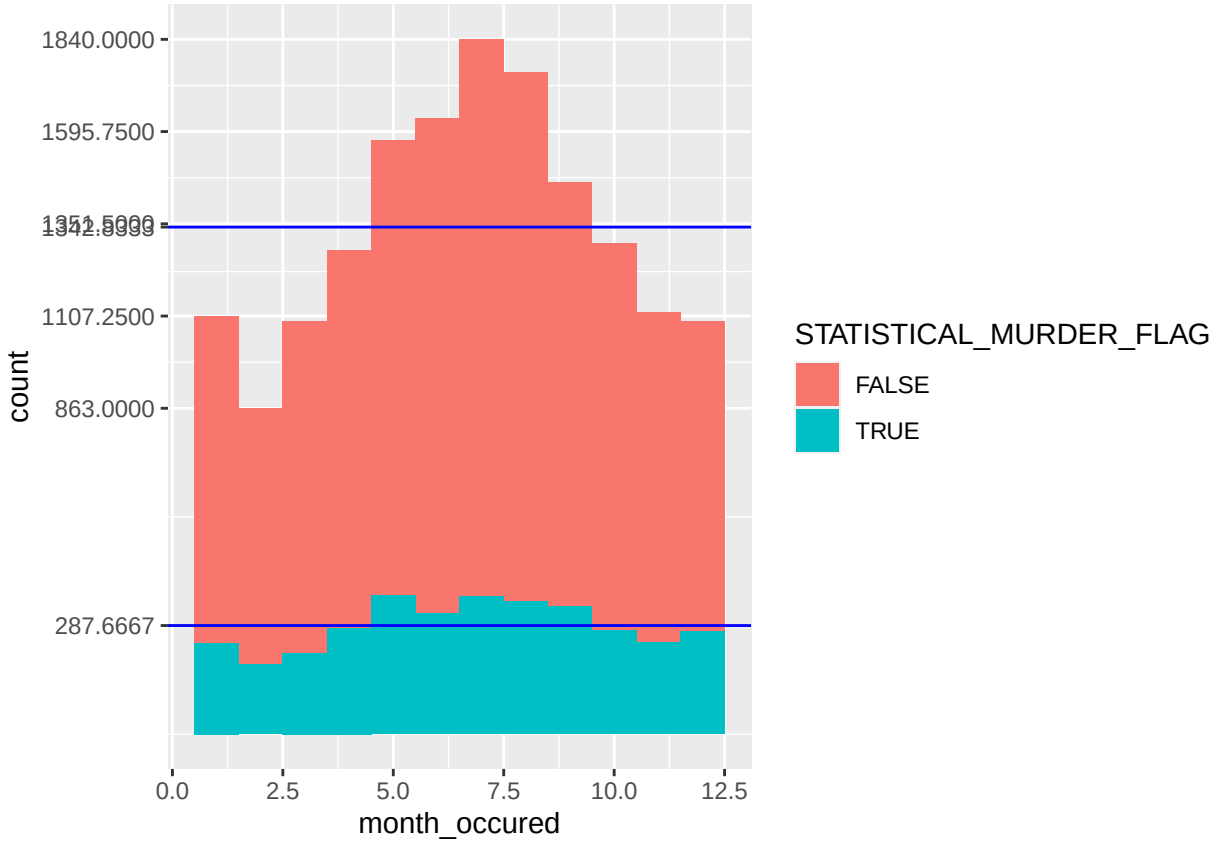
In order to discover if there is a monthly trend, I plotted a linear curve using local polynomial regression.



To amplify the curve I grouped the yearly data by month.



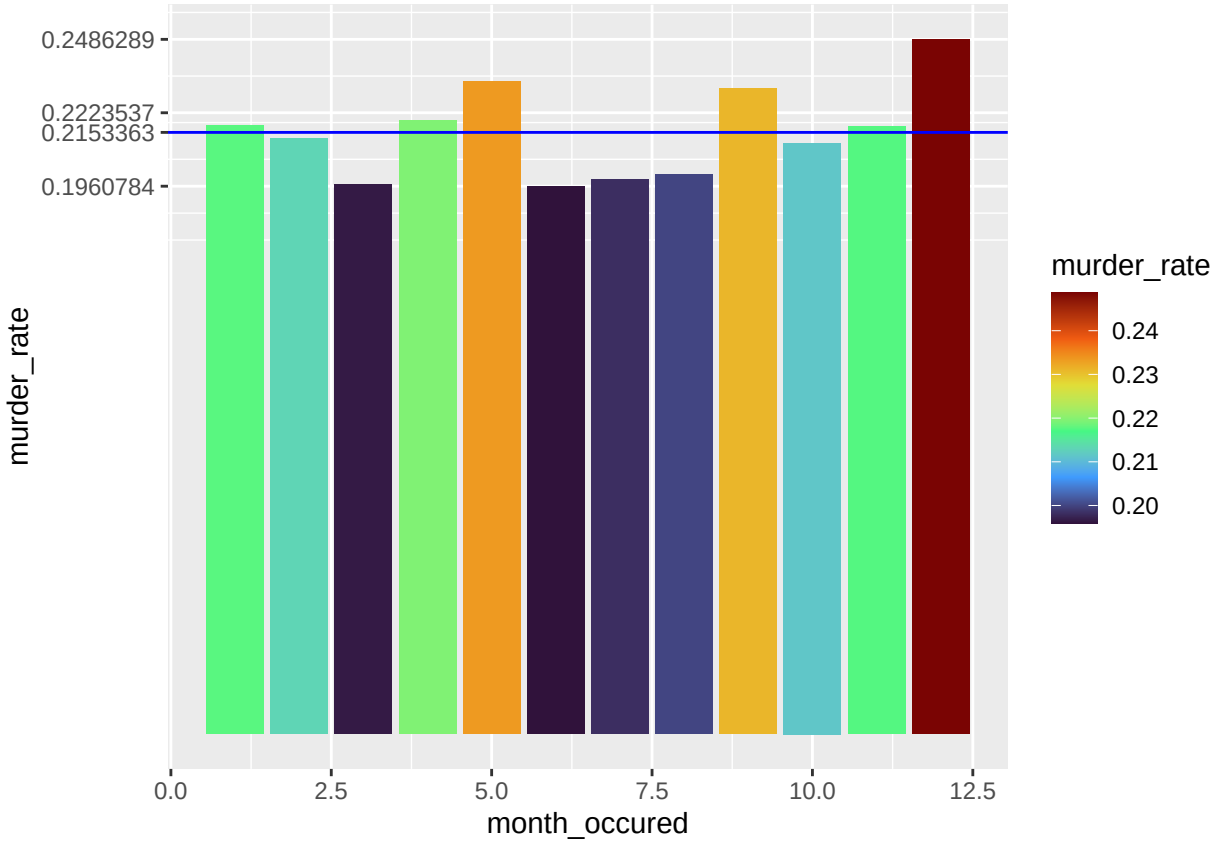
The monthly shootings are the lowest in February and escalate through the summer.



At a glance the murders seem to follow that trend.

Monthly Murder Rate

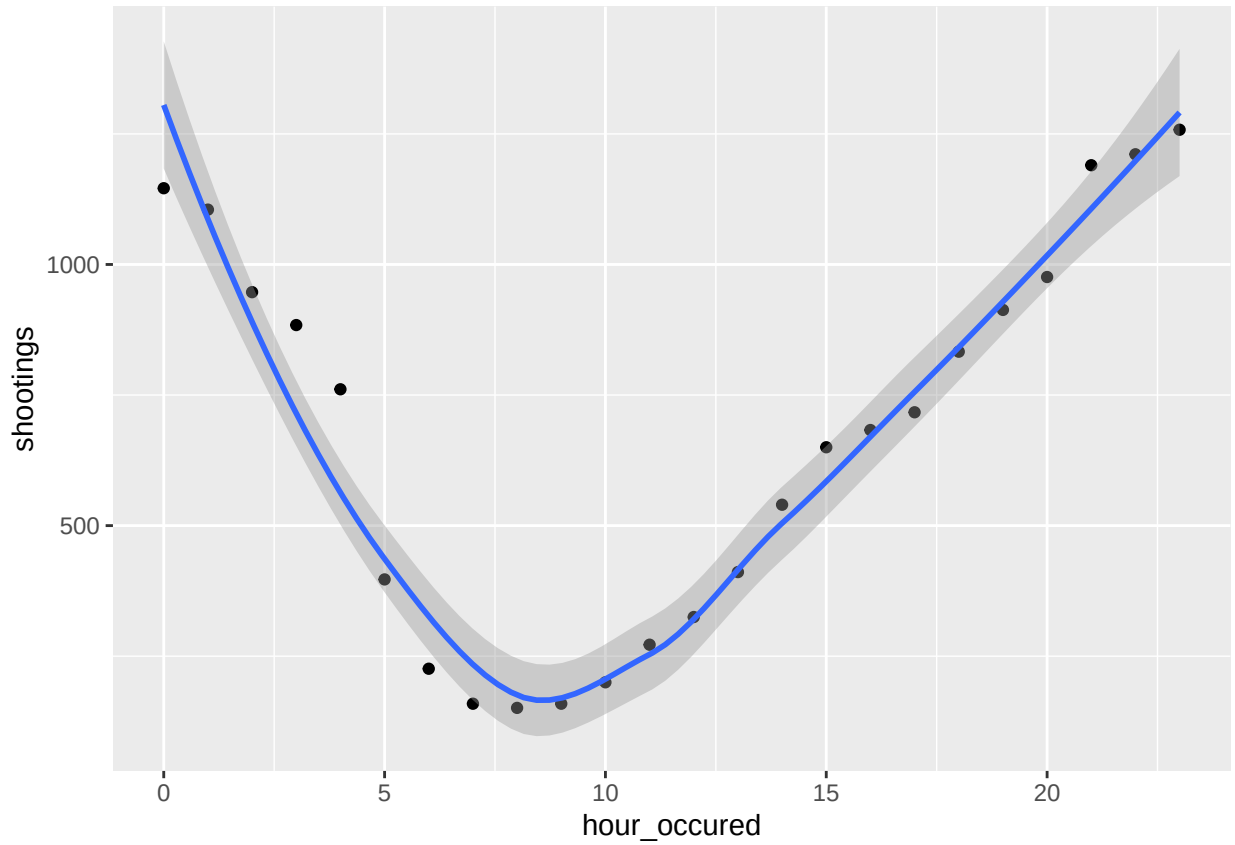
The rate of shootings to murders are pretty steady throughout the year. March, June, July, and August have slightly lower rate of shootings that result in Murder. May and September are slightly higher than the mean of .2215 and December is the highest rate of shootings to murders at .249.



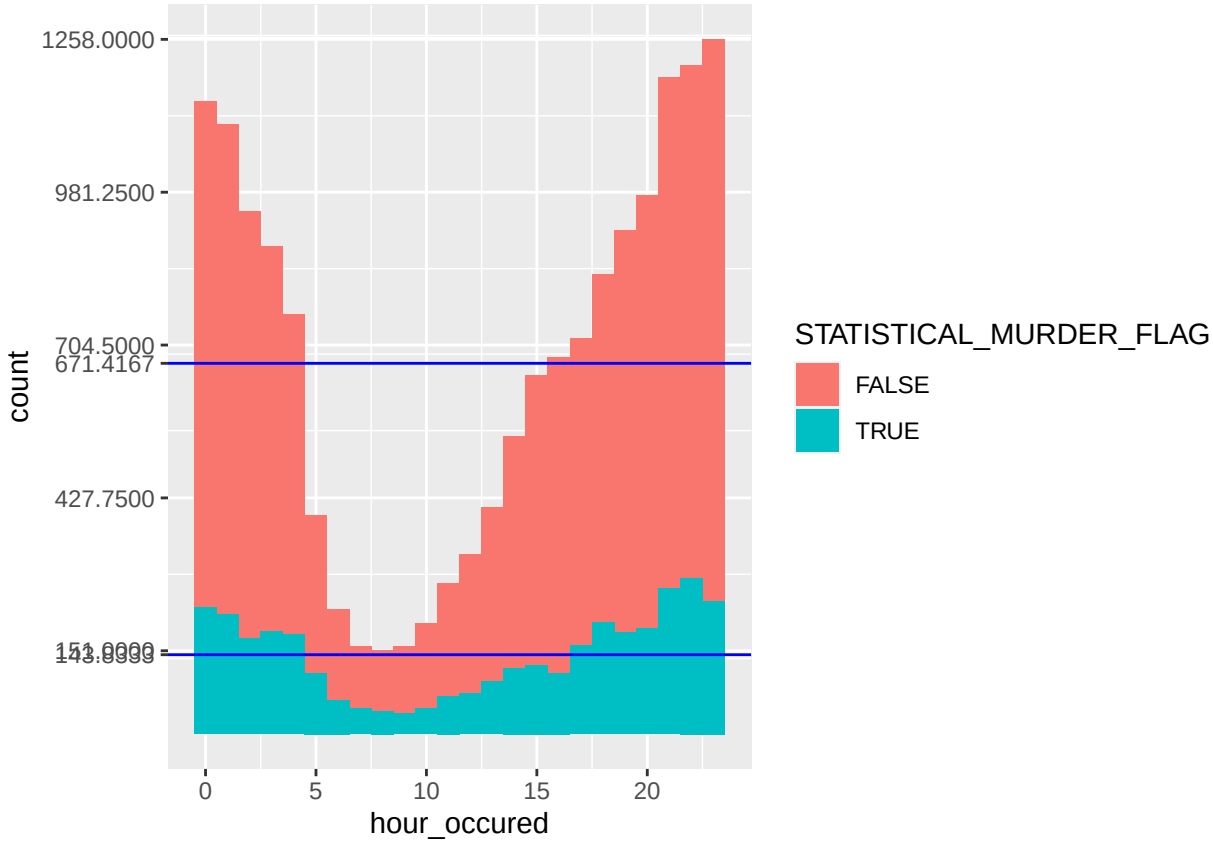
If you were shot in March or June, you were least likely to die. If you were shot in December you were most likely to die. While the murder rate didn't fluctuate dramatically, it also didn't follow the shooting volume.

Hourly Shootings

With hourly shootings we grouped the data by hour and plotted a curve of the shootings using local polynomial regression (or Loess). Again we notice a pretty smooth curve in the data.

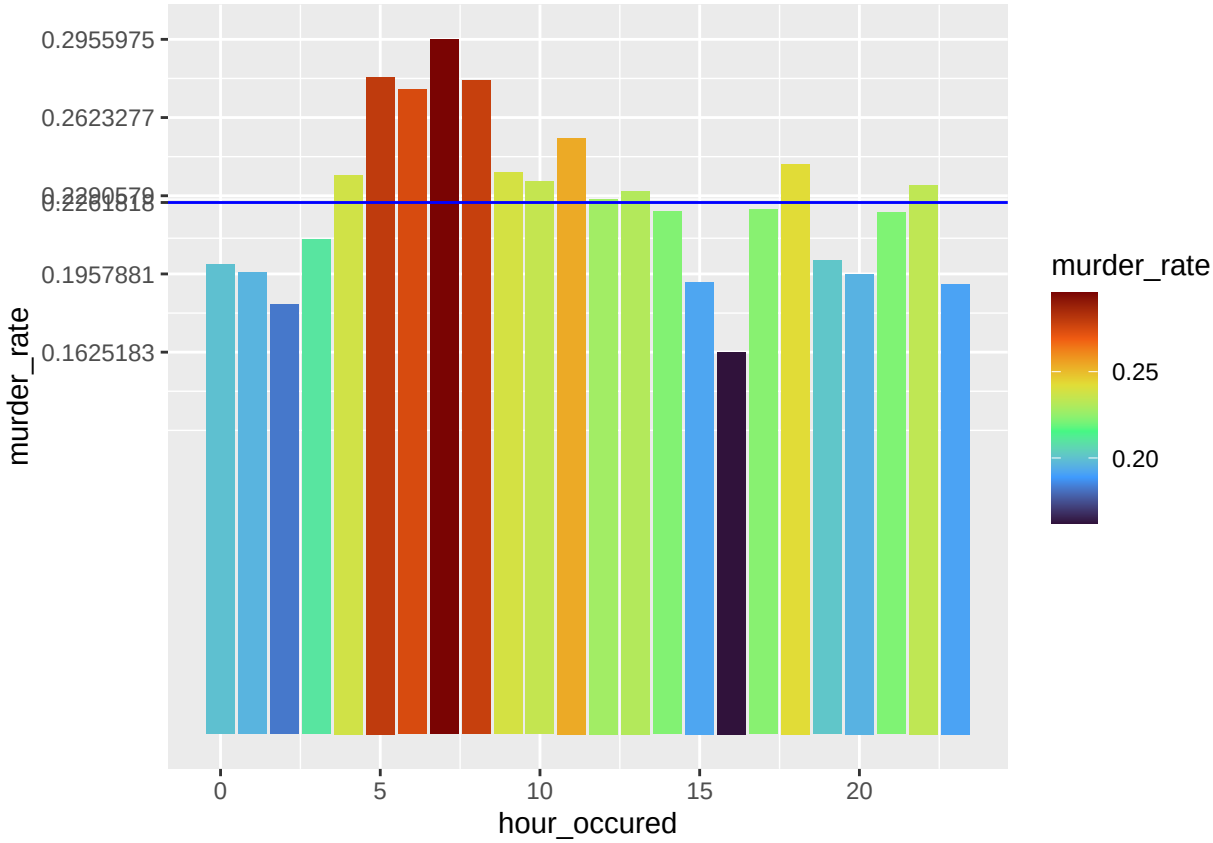


The number of shootings per hour is the lowest between 8:00-9:00am and escalates until midnight. From midnight it stays above average until 6-7 where it begins to drop rapidly. The number of murders mostly follow that trend as well.



Hourly Murder Rate

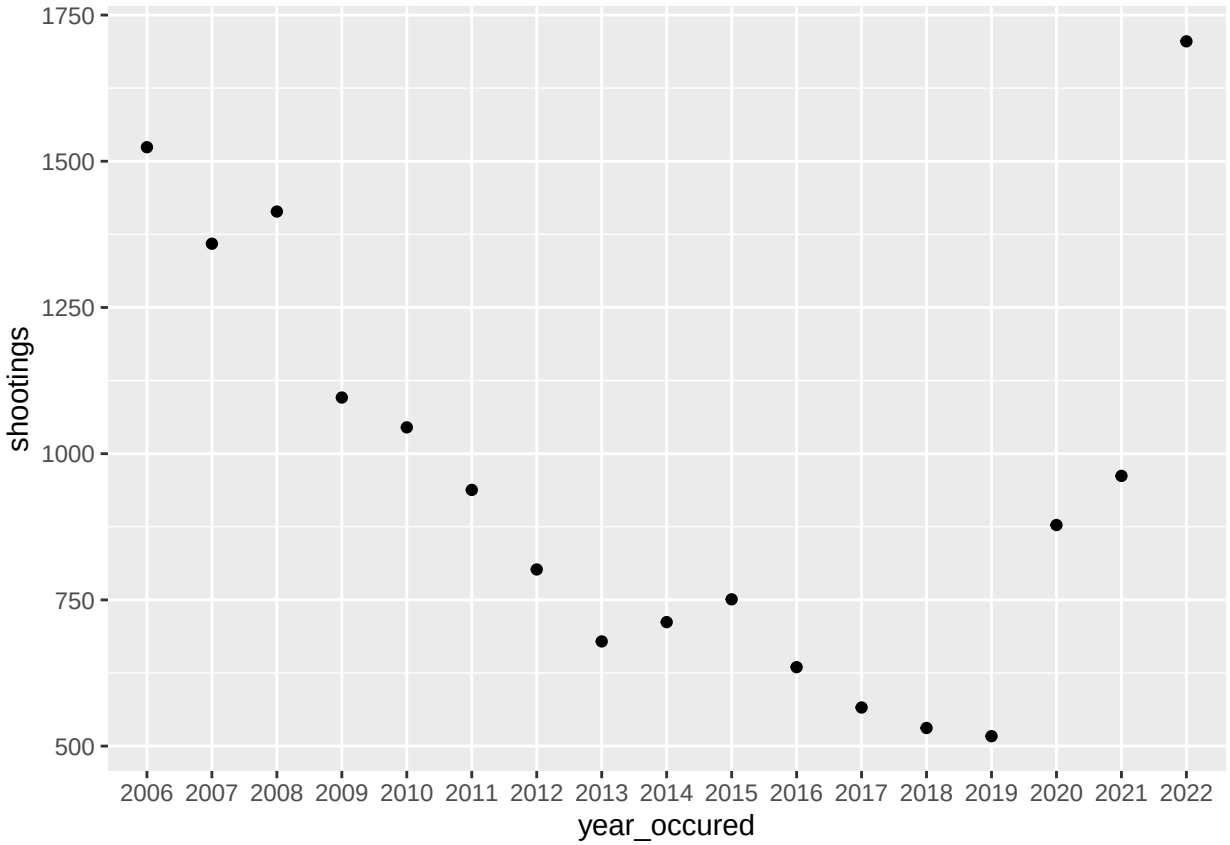
The hours with the highest murder rate start at 5:00 AM, 6:00 AM, 7:00 AM, and 8:00 AM, with 7:00 AM - 8:00 AM having the highest rate of murders at .296 in spite of the low shooting volume. The lowest rate of murders happens between 4:00 PM - 5:00 PM which corresponds to a shooting volume that is only slightly above the mean.



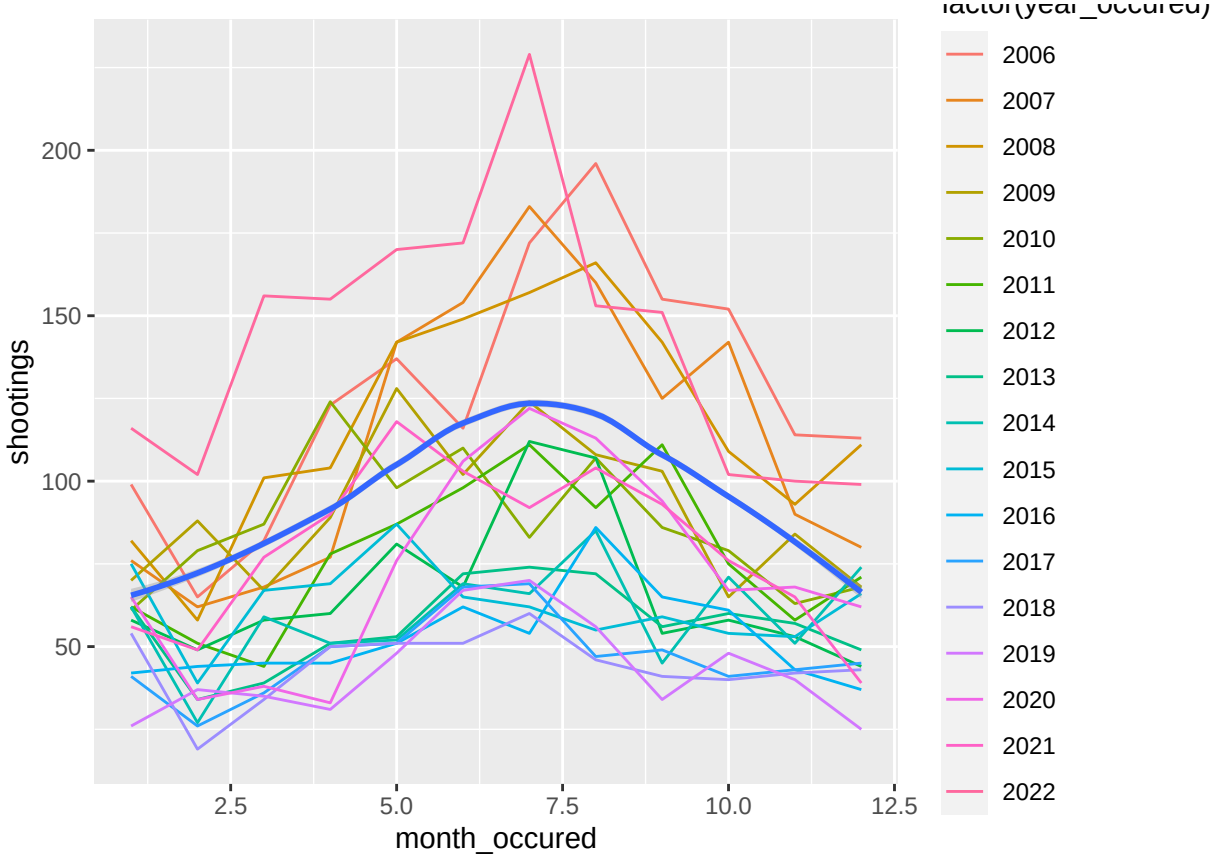
Step 4: Add Bias Identification

Data Bias

In the 17 years of data available, the number of shootings per year fluctuated greatly. In 2022 there were over 3 times the number of shootings as in 2019. The risk is that one of the more extreme years could skew the results. Below is a point chart of the number of shootings for each year.



I chose to include the data for all years for two reasons. First, the difference between 2018 and 2019 or 2022 and 2006 wasn't dramatic. Secondly, the results didn't change dramatically when filtered out. To visualize this, a version of the initial table with color coded lines for the individual years is displayed below.



Personal Bias

This analysis focused on timing. I had my own personal assumption that most shootings occur at night, and the data supports this. I also assumed these same hours in the evening with high numbers of shootings would be more likely for a shooting to result in a fatality due to overwhelmed first responders and emergency rooms. This assumption was incorrect according to the data.

The Geographic distribution could also be a source of personal bias for anyone familiar with New York City. While I have visited a couple of times, it wasn't interesting enough to learn about the reputations of different neighborhoods.

Additionally, this data included age, sex, and race data in the PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP, VIC_SEX, VIC_RACE columns. Some records were based on incomplete reports where this information was listed as "UNKNOWN". While the values of these columns didn't factor into the analysis, records based on incomplete reports were removed.

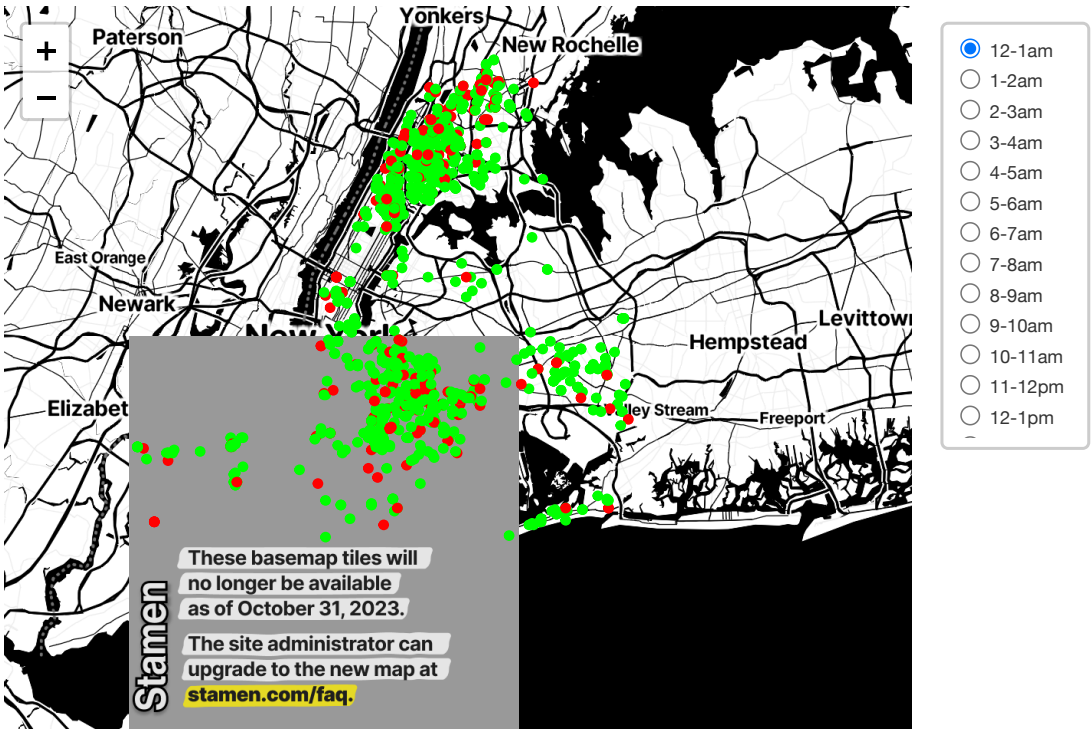
Bonus Charts and Analysis

Here are some maps with the hourly and monthly shootings plotted on them. Overall, the deaths seem pretty evenly distributed among the shootings. There doesn't seem to be any big obvious red clusters where I could easily speculate that one part of the city is under-served.

Mostly, I like maps and wanted to plot the data to locations. That's why I included them.

Geographic distribution of Hourly Shootings

With more data, this map could be overlayed on traffic patterns and the locations of critical resources such as hospitals. Perhaps there's a correlation between deaths and proximity to emergency rooms, or traffic congestion in the routes to reach them.



Geographic distribution of Monthly Shootings

With more data, this could correlate to something like road conditions. Mostly, this is here because I like maps.