NYPD Shooting Data

2024-03-24

Additional Libraries Used

Note that this analysis leverages the forecast library, so users must execute the command `install.packages("forecast")` prior to knitting this document.

Intro

In this analysis, we would like to investigate the frequency of shooting incidents in New York City.

Load the Data

Let's first load the data from https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD. Per the City of New York website, this dataset contains details, such as the location and time of occurrence, for every shooting incident that occurred in New York City from 2006 through the end of the prior calendar year. The City of New York website indicates that this data is made available for public use. Additional information about the dataset can be found on https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic.

```
rawData = read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD",
                   show_col_types = FALSE)
head(rawData)
```

```
## # A tibble: 6 x 21
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     LOC_OF_OCCUR_DESC PRECINCT
##          <dbl> <chr>      <time>     <chr>    <chr>                <dbl>
## 1    228798151 05/27/2021 21:30      QUEENS   <NA>                   105
## 2    137471050 06/27/2014 17:40      BRONX    <NA>                    40
## 3    147998800 11/21/2015 03:56      QUEENS   <NA>                   108
## 4    146837977 10/09/2015 18:30      BRONX    <NA>                    44
## 5     58921844 02/19/2009 22:58      BRONX    <NA>                    47
## 6    219559682 10/21/2020 21:36      BROOKLYN <NA>                    81
## # i 15 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>
```

Clean and Transform the Data

We cleaned the data by making the following changes:

1. Changed the `OCCUR_DATE` column from character to date
2. Coded `BORO` as a factor
3. Added columns of factors for the month, year, and hour of the incident

```
rawData = rawData %>%
    mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
```

```
    mutate(BORO = as.factor(as.character(BORO)))

rawData$MONTH = as.factor(month(rawData$OCCUR_DATE))
rawData$YEAR = as.factor(year(rawData$OCCUR_DATE))
rawData$HOUR = as.factor(hour(hms(as.character(rawData$OCCUR_TIME))))

levels(rawData$HOUR) = c("12a", "1a", "2a", "3a", "4a", "5a", "6a", "7a", "8a", "9a",
                         "10a", "11a", "12p", "1p", "2p", "3p", "4p", "5p", "6p", "7p",
                         "8p", "9p", "10p", "11p")

summary(rawData$MONTH)
```

```
##    1    2    3    4    5    6    7    8    9   10   11   12
## 1716 1340 1688 1983 2571 2829 3238 3156 2572 2279 1944 1996
```

```
summary(rawData$YEAR)
```

```
## 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021
## 2055 1887 1959 1828 1912 1939 1717 1339 1464 1434 1208  970  958  967 1948 2011
## 2022
## 1716
```

```
summary(rawData$HOUR)
```

```
##  12a   1a   2a   3a   4a   5a   6a   7a   8a   9a  10a  11a  12p   1p   2p   3p
## 2186 2081 1812 1633 1441  702  366  233  238  217  304  372  490  577  786  924
##   4p   5p   6p   7p   8p   9p  10p  11p
## 1034 1070 1247 1477 1684 1972 2162 2304
```

Subset the Data

We don't need all of the columns from the original dataset, so let's create a new, smaller dataframe to summarize incident counts by month and borough.

```
monthlyIncidents <- rawData %>% group_by(MONTH, BORO) %>%
  summarize(Incidents = n())
```

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

```
head(monthlyIncidents)
```

```
## # A tibble: 6 x 3
## # Groups:   MONTH [2]
##   MONTH BORO          Incidents
##   <fct> <fct>             <int>
## 1 1     BRONX               536
## 2 1     BROOKLYN            614
## 3 1     MANHATTAN           245
## 4 1     QUEENS              267
## 5 1     STATEN ISLAND        54
## 6 2     BRONX               381
```

Let's create a similar dataframe to summarize incident counts by year and borough.

```
yearlyIncidents <- rawData %>% group_by(YEAR, BORO) %>%
  summarize(Incidents = n())
```

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

```
head(yearlyIncidents)
```

```
## # A tibble: 6 x 3
## # Groups:   YEAR [2]
##   YEAR  BORO          Incidents
##   <fct> <fct>             <int>
## 1 2006  BRONX               568
## 2 2006  BROOKLYN            850
## 3 2006  MANHATTAN           288
## 4 2006  QUEENS              296
## 5 2006  STATEN ISLAND        53
## 6 2007  BRONX               533
```

Let's create a similar dataframe to summarize incident counts by hour and borough.

```
hourlyIncidents <- rawData %>% group_by(HOUR, BORO) %>%
  summarize(Incidents = n())
```

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

```
head(hourlyIncidents)
```

```
## # A tibble: 6 x 3
## # Groups:   HOUR [2]
##   HOUR  BORO          Incidents
##   <fct> <fct>             <int>
## 1 12a   BRONX               684
## 2 12a   BROOKLYN            827
## 3 12a   MANHATTAN           321
## 4 12a   QUEENS              291
## 5 12a   STATEN ISLAND        63
## 6 1a    BRONX               609
```

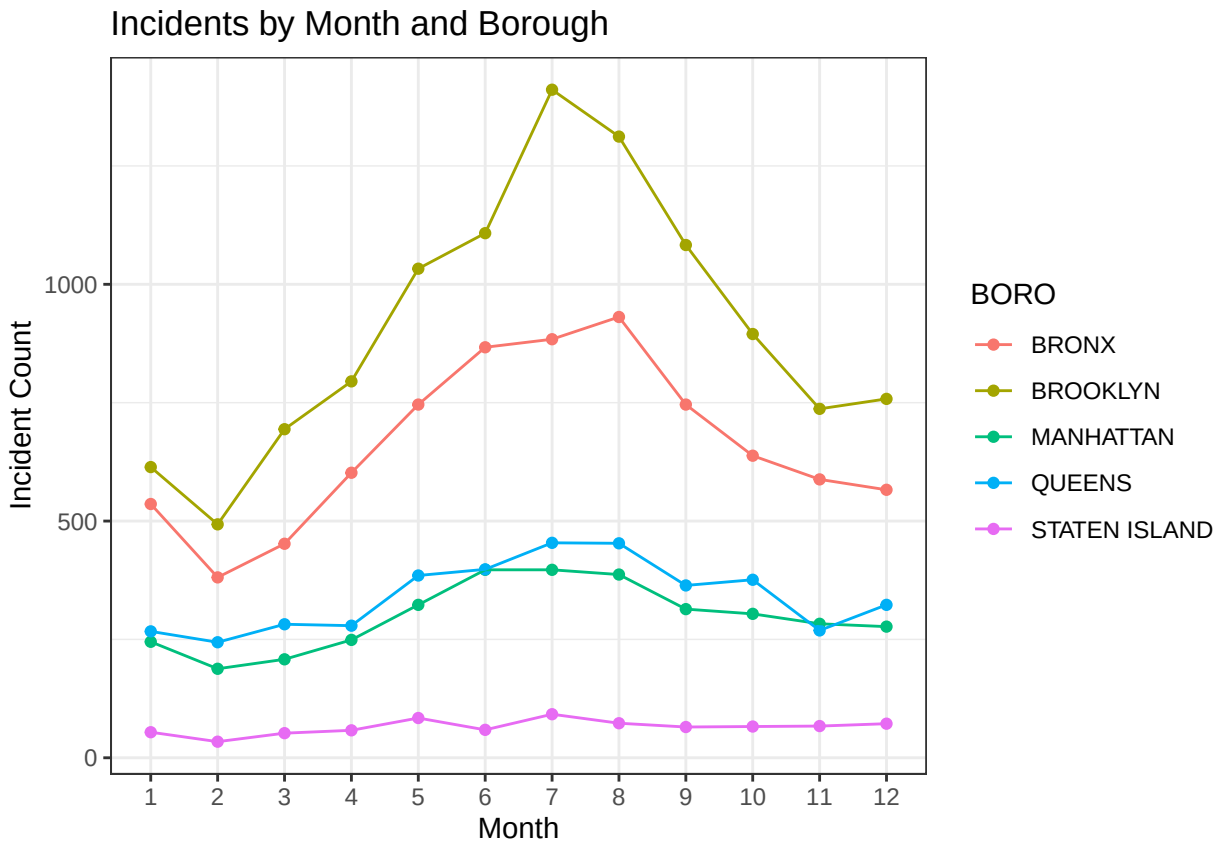
Visualize the Data - By Month and Borough

It would be interesting to understand if incident counts differ over time using time segments of months, years, and hours. Let's start with months and plot the relationship between incident counts and month for each borough. The plot shows that incident counts in Brooklyn are higher than the other boroughs and Staten Island has the fewest number of incidents. The incident counts could be misleading regarding the relative safety of each borough if the boroughs with higher populations have higher incident counts. It would be interesting to understand the incident count as a percentage of borough population. The population for each borough is not included in this data set, but another analysis could find this information from another data source and join the population into the data set. For now, understanding the relationship between population and incident count is out of scope for this analysis.

There appears to be higher incident counts across the boroughs in the summer months (i.e., June - August). Note that this plot shows correlation, not causation.

```
ggplot(monthlyIncidents, aes(x=MONTH, y=Incidents)) +
  geom_point(aes(color = BORO)) +
  geom_line(aes(group = BORO, color=BORO)) +
  xlab("Month") +
  ylab("Incident Count") +
```

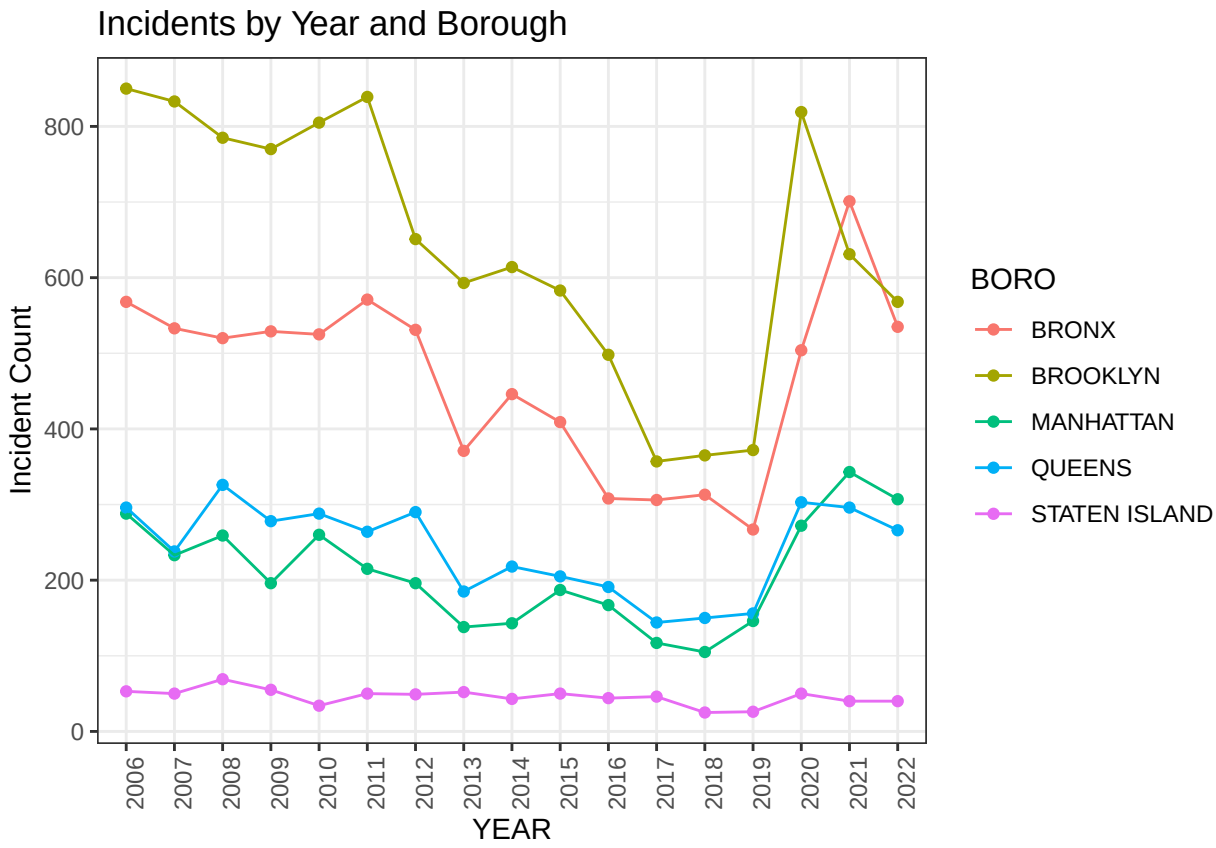
```
  ggtitle("Incidents by Month and Borough") +
  theme_bw()
```



Visualize the Data - By Year and Borough

Let's plot the relationship between incident counts and year for each borough. The number of incidents for all boroughs declined until 2019. Then, all of the boroughs except Staten Island experienced a sharp increase in incidents in 2020.

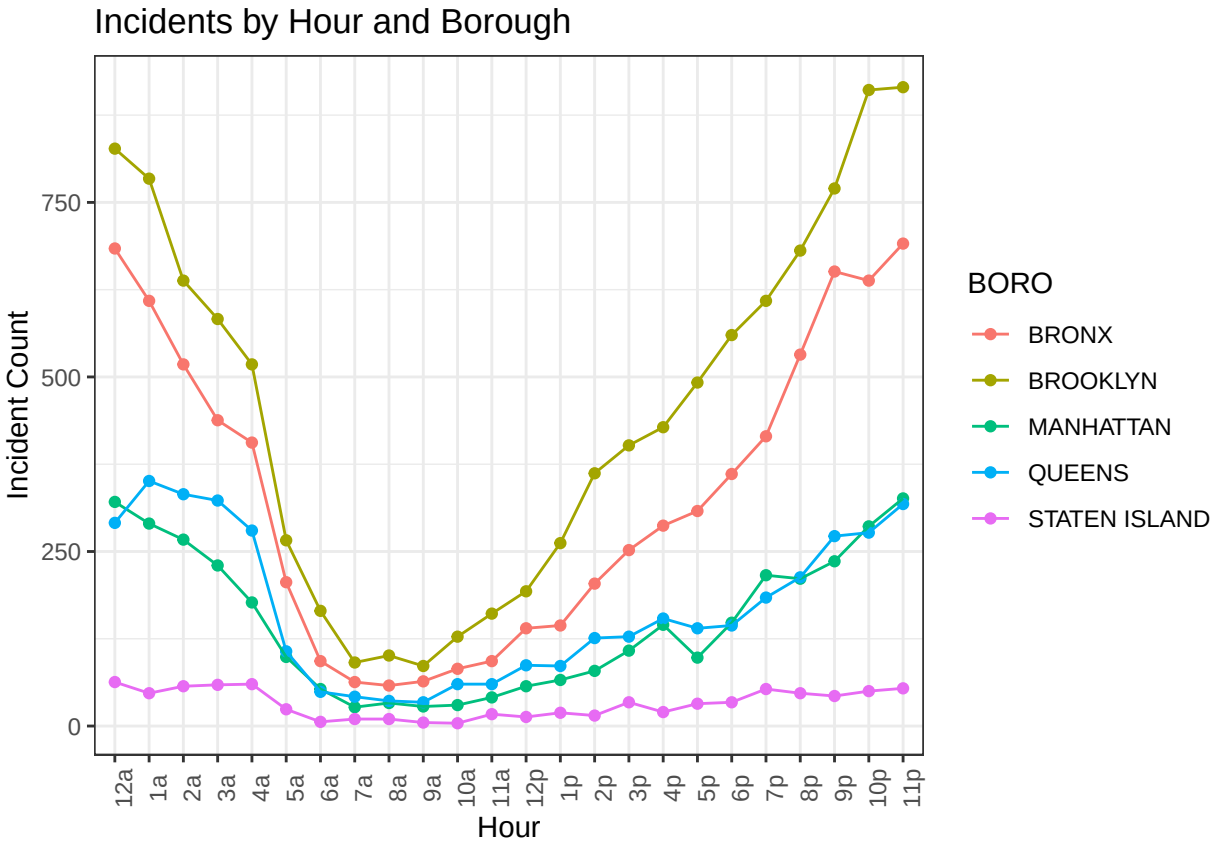
```
ggplot(yearlyIncidents, aes(x=YEAR, y=Incidents)) +
  geom_point(aes(color = BORO)) +
  geom_line(aes(group = BORO, color=BORO)) +
  xlab("YEAR") +
  ylab("Incident Count") +
  ggtitle("Incidents by Year and Borough") +
  theme_bw() +
  theme(axis.text.x = element_text(angle=90))
```



Visualize the Data - By Hour and Borough

Let's plot the relationship between incident counts and hour for each borough. The number of incidents is higher during the late evening and early morning hours. The incident count in Brooklyn shows the greatest difference between mid-day and late-night hours, but the Staten Island counts remain relatively flat.

```
ggplot(hourlyIncidents, aes(x=HOUR, y=Incidents)) +
  geom_point(aes(color = BORO)) +
  geom_line(aes(group = BORO, color=BORO)) +
  xlab("Hour") +
  ylab("Incident Count") +
  ggtitle("Incidents by Hour and Borough") +
  theme_bw() +
  theme(axis.text.x = element_text(angle=90))
```



Model the Data

Let's create a linear regression model predicting `Incidents` using `MONTH and BORO`. We can see from the summary that the predictors for `BORO` are statistically significant based on the low p-values. However, some of the month values are not statistically significant. The model also confirms that we see lower predicted values of incidents for the Bronx, Manhattan, Queens, and Staten Island boroughs when compared to Brooklyn. The model also confirms what we saw in the plot with June-August having the highest incidents per month. The high R-squared means that the linear regression model explains roughly 90% of the variability we see in the incident data.

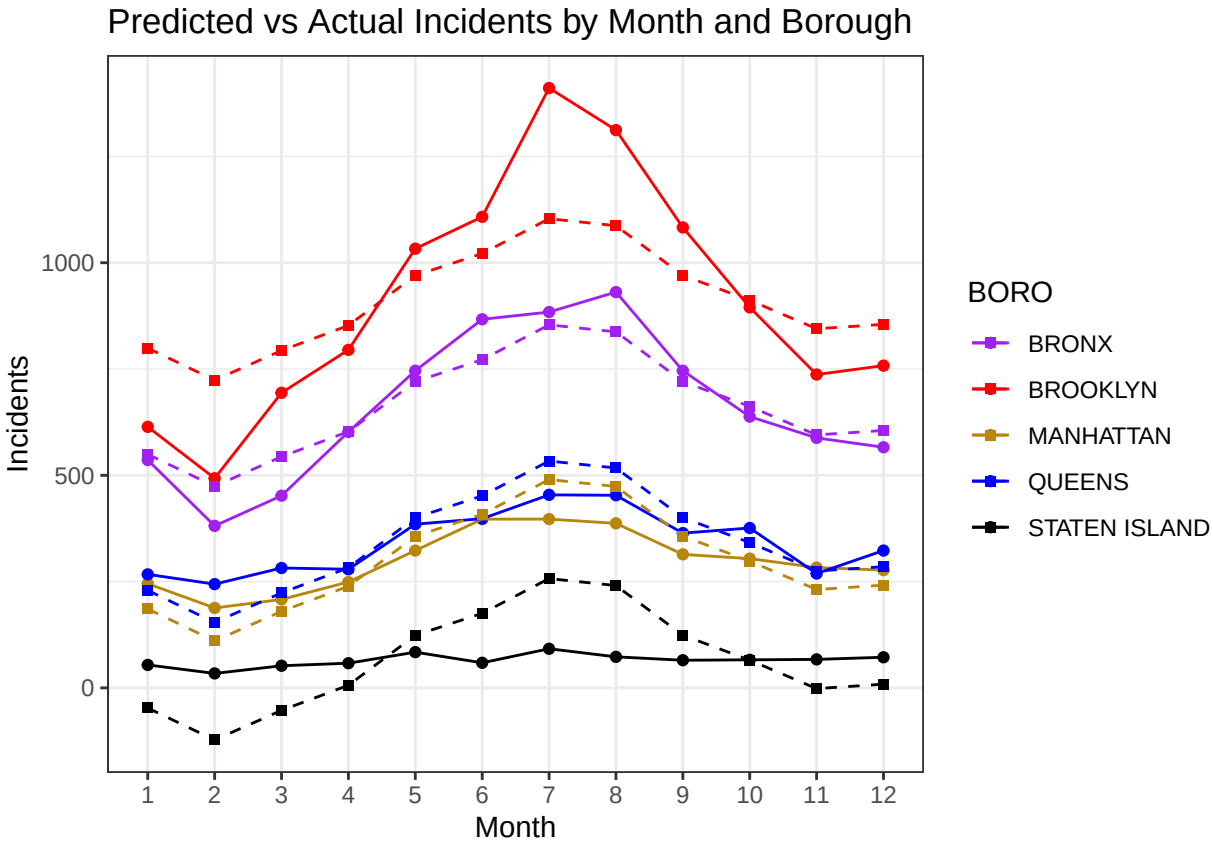
```
lm.mod = lm(Incidents ~ MONTH + BORO, data=monthlyIncidents)
lm.pred = predict(lm.mod)
monthlyIncidents$PRED = lm.pred

summary(lm.mod)
```

```
##
## Call:
## lm(formula = Incidents ~ MONTH + BORO, data = monthlyIncidents)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -230.883  -57.829   -2.192   58.658  307.517
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         549.42      56.47   9.730 1.54e-12 ***
## MONTH2              -75.20      69.16  -1.087 0.282775
```

```
## MONTH3               -5.60      69.16  -0.081 0.935827
## MONTH4               53.40      69.16   0.772 0.444136
## MONTH5              171.00      69.16   2.473 0.017345 *
## MONTH6              222.60      69.16   3.219 0.002419 **
## MONTH7              304.40      69.16   4.402 6.75e-05 ***
## MONTH8              288.00      69.16   4.165 0.000143 ***
## MONTH9              171.20      69.16   2.476 0.017222 *
## MONTH10             112.60      69.16   1.628 0.110617
## MONTH11              45.60      69.16   0.659 0.513082
## MONTH12              56.00      69.16   0.810 0.422426
## BOROBROOKLYN        249.67      44.64   5.593 1.33e-06 ***
## BOROMANHATTAN      -363.75      44.64  -8.149 2.47e-10 ***
## BOROQUEENS         -320.25      44.64  -7.174 6.36e-09 ***
## BOROSTATEN ISLAND  -596.75      44.64 -13.368  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 109.3 on 44 degrees of freedom
## Multiple R-squared:  0.9204, Adjusted R-squared:  0.8933
## F-statistic: 33.94 on 15 and 44 DF,  p-value: < 2.2e-16
```

The plot shows that the linear model, shown with dashed lines, follows the pattern of the actual incident data, shown in solid lines. In other words, the linear prediction model captures that incidents were higher in the summer months and correctly predicted Brooklyn having the highest incident count followed by the Bronx, Queens, Manhattan, and Staten Island. However, the prediction lines show sizeable prediction errors for Brooklyn and Staten Island. The linear prediction model seems well-fitting for the incident counts in the Bronx, Manhattan, and Queens.

```
ggplot(monthlyIncidents, aes(x=MONTH, y=Incidents, group = BORO, color=BORO)) +
  geom_point() +
  geom_line() +
  geom_point(aes(y=PRED, color = BORO), shape=15)+
  geom_line(aes(y=PRED), lty=2) +
  xlab("Month") +
  ylab("Incidents") +
  ggtitle("Predicted vs Actual Incidents by Month and Borough") +
  scale_color_manual(values=c("purple","red", "darkgoldenrod", "blue", "black")) +
  theme_bw()
```



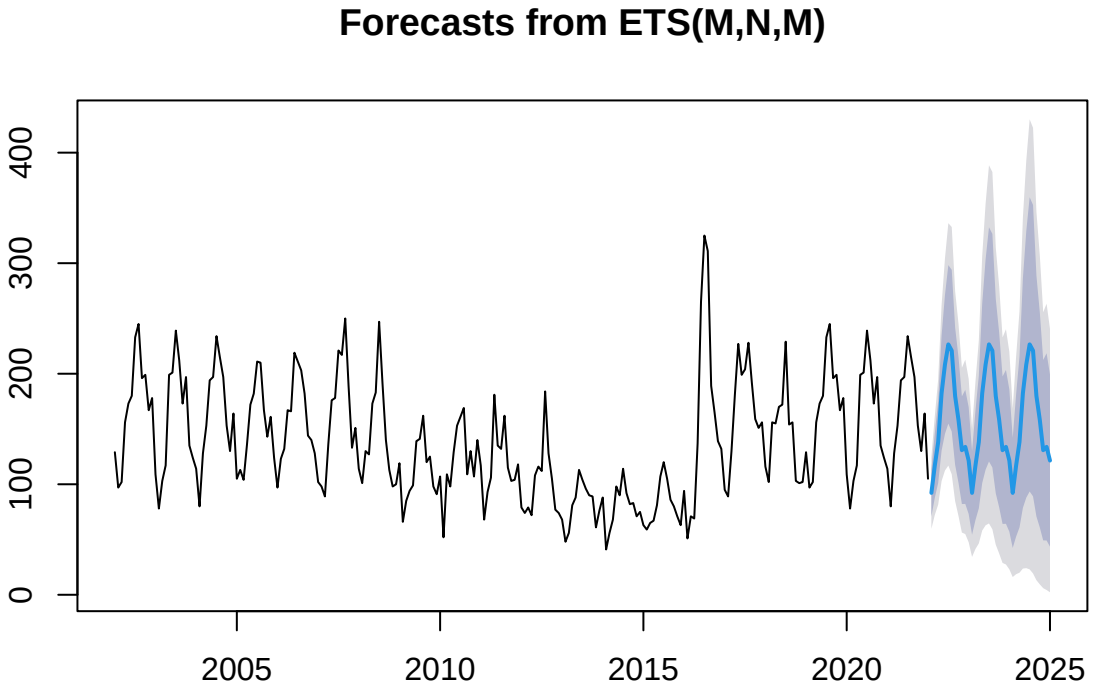
Forecast the Data

Let's use the historical data for forecast shootings for the next three years. We can see that the forecasted number of incidents is similar to the last three years of historical data. We can also see that the level of uncertainty grows the further out into the future we forecast.

```
monthly.fc <- rawData %>% group_by(YEAR, MONTH) %>%
  summarize(Incidents = n())
```

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

```
fc = forecast(ts(monthly.fc$Incidents, start=c(2002), end=c(2022), frequency=12), h=36)
plot(fc)
```



Commentary

The data shows that incident counts differ across time. We saw that incident counts were highest in the summer months and during the over night hours. We also saw that incident counts had a downward trend across all boroughs until 2020, when there was a sharp increase in incidents.

As mentioned above, the plots could be misleading because the incident count is much higher in Brooklyn compared to the other boroughs. This could mean that Brooklyn is more dangerous. It could also mean that all boroughs have a similar ratio of incidents to population, but the incident count is higher in Brooklyn because the population is higher. Without data on the population within each borough, we cannot speak to the safety of the boroughs.

As with any manually entered data, there could be human error in the data that skews the results. Additionally, there could be bias in the way the data was collected (e.g., some specific subset of the data were not reported).

Session Info

```
## R version 4.3.3 (2024-02-29)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.4.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## 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/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
```

```
##
## other attached packages:
##  [1] forecast_8.22.0 lubridate_1.9.3 forcats_1.0.0   stringr_1.5.1
##  [5] dplyr_1.1.4     purrr_1.0.2     readr_2.1.5     tidyr_1.3.1
##  [9] tibble_3.2.1    ggplot2_3.5.0   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.3    stringi_1.8.3     lattice_0.22-5
##  [5] hms_1.1.3         digest_0.6.33     magrittr_2.0.3    evaluate_0.23
##  [9] grid_4.3.3        timechange_0.3.0  fastmap_1.1.1     nnet_7.3-19
## [13] fansi_1.0.6       scales_1.3.0      cli_3.6.2         crayon_1.5.2
## [17] rlang_1.1.2       bit64_4.0.5       munsell_0.5.0     withr_2.5.2
## [21] yaml_2.3.8        tools_4.3.3       parallel_4.3.3    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_0.10-55   bit_4.0.5
## [33] vroom_1.6.5       urca_1.3-3        pkgconfig_2.0.3   pillar_1.9.0
## [37] gtable_0.3.4      quantmod_0.4.26   glue_1.6.2        Rcpp_1.0.12
## [41] highr_0.10        xfun_0.41         lmtest_0.9-40     tidyselect_1.2.1
## [45] rstudioapi_0.15.0 knitr_1.45        farver_2.1.1      nlme_3.1-164
## [49] htmltools_0.5.7   labeling_0.4.3    xts_0.13.2        rmarkdown_2.26
## [53] timeDate_4032.109 fracdiff_1.5-3    compiler_4.3.3    quadprog_1.5-8
## [57] TTR_0.24.4
```