NYPD Shooting Incident Data Report

Andrew Simms

2022-05-21

Assignment Overview

Import, tidy, and analyze the NYPD Shooting data incident dataset obtained. Be sure your project is reproducible and contains some visualization and analysis. You may use the data to do any analysis that is of interest to you. You should include at least two visualizations and one model. Be sure to identify any bias possible in the data and in your analysis.

Importing Data

Data downloaded from data.gov: https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic

Descriptions of columns is here: https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8

Use the `read.csv` function to import csv data into the dataframe `df`.

```
df  <- read.csv("data/NYPD_Shooting_Incident_Data__Historic_.csv")
```

Wrangling

What data do we have?

```
str(df)
```

```
## 'data.frame':    23585 obs. of  19 variables:
##  $ INCIDENT_KEY           : int  24050482 77673979 203350417 80584527 90843766 92393427 73057167 2113
##  $ OCCUR_DATE             : chr  "08/27/2006" "03/11/2011" "10/06/2019" "09/04/2011" ...
##  $ OCCUR_TIME             : chr  "05:35:00" "12:03:00" "01:09:00" "03:35:00" ...
##  $ BORO                   : chr  "BRONX" "QUEENS" "BROOKLYN" "BRONX" ...
##  $ PRECINCT               : int  52 106 77 40 100 67 77 81 101 106 ...
##  $ JURISDICTION_CODE      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LOCATION_DESC          : chr  "" "" "" "" ...
##  $ STATISTICAL_MURDER_FLAG: chr  "true" "false" "false" "false" ...
##  $ PERP_AGE_GROUP         : chr  "" "" "" "" ...
##  $ PERP_SEX               : chr  "" "" "" "" ...
##  $ PERP_RACE              : chr  "" "" "" "" ...
##  $ VIC_AGE_GROUP          : chr  "25-44" "65+" "18-24" "<18" ...
##  $ VIC_SEX                : chr  "F" "M" "F" "M" ...
##  $ VIC_RACE               : chr  "BLACK HISPANIC" "WHITE" "BLACK" "BLACK" ...
```

```
##  $ X_COORD_CD             : num  1017542 1027543 995325 1007453 1041267 ...
##  $ Y_COORD_CD             : num  255919 186095 185155 233952 157134 ...
##  $ Latitude               : num  40.9 40.7 40.7 40.8 40.6 ...
##  $ Longitude              : num  -73.9 -73.8 -74 -73.9 -73.8 ...
##  $ Lon_Lat                : chr  "POINT (-73.87963173099996 40.86905819000003)" "POINT (-73.84392019
```

```
nrow(df)
```

```
## [1] 23585
```

Cleaning

Select columns with relevant data:

```
df <- df %>%
    select(OCCUR_DATE, OCCUR_TIME, BORO, VIC_RACE, VIC_SEX,
           STATISTICAL_MURDER_FLAG)
```

Check Integrity:

```
sum(is.na(df$OCCUR_DATE))
```

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

```
sum(is.na(df$OCCUR_TIME))
```

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

```
sum(is.na(df$BORO))
```

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

```
sum(is.na(df$VIC_RACE))
```

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

```
sum(is.na(df$VIC_SEX))
```

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

```
sum(is.na(df$STATISTICAL_MURDER_FLAG))
```

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

Ideas for interesting data analysis:

```
table(df["BORO"])
```

Shootings by borough

```
##
##         BRONX      BROOKLYN     MANHATTAN        QUEENS STATEN ISLAND
##          6701          9734          2922          3532           696
```

```
df$YEAR <- str_sub(df$OCCUR_DATE, -4)
table(df["YEAR"])
```

Shootings per year

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

```
df$HOUR <- str_sub(df$OCCUR_TIME, 1, 2)
table(df["HOUR"])
```

Shootings Time of Day

```
##
##   00   01   02   03   04   05   06   07   08   09   10   11   12   13   14   15
## 1908 1865 1622 1464 1292  636  301  198  190  177  248  315  415  442  685  770
##   16   17   18   19   20   21   22   23
##  874  909 1054 1235 1418 1717 1854 1996
```

```
df$MINUTE <- str_sub(df$OCCUR_TIME, 4, 5)
table(df["MINUTE"])
```

Shootings Time of Day Minute

```
##
##   00   01   02   03   04   05   06   07   08   09   10   11   12   13   14   15
## 1495  238  295  318  230  628  224  272  301  286  748  249  297  281  282  916
##   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
##  245  293  295  242  833  257  311  287  250  599  274  239  281  275 1580  196
##   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47
##  254  272  282  621  268  285  299  247  834  232  265  265  252  882  254  314
##   48   49   50   51   52   53   54   55   56   57   58   59
##  288  243  875  238  256  266  286  662  290  259  329  250
```

```
table(df["VIC_RACE"])
```

Shooting Victims by Race

```
##
## AMERICAN INDIAN/ALASKAN NATIVE       ASIAN / PACIFIC ISLANDER
##                              9                            327
##                          BLACK                BLACK HISPANIC
##                          16869                         2245
##                        UNKNOWN                        WHITE
##                             65                          620
##                WHITE HISPANIC
##                           3450
```

```
table(df["VIC_SEX"])
```

Shooting Victims by Sex

```
##
##     F     M     U
##  2204 21370    11
```

```
table(df["STATISTICAL_MURDER_FLAG"])
```

Shooting defined as Murders

```
##
## false  true
## 19085  4500
```

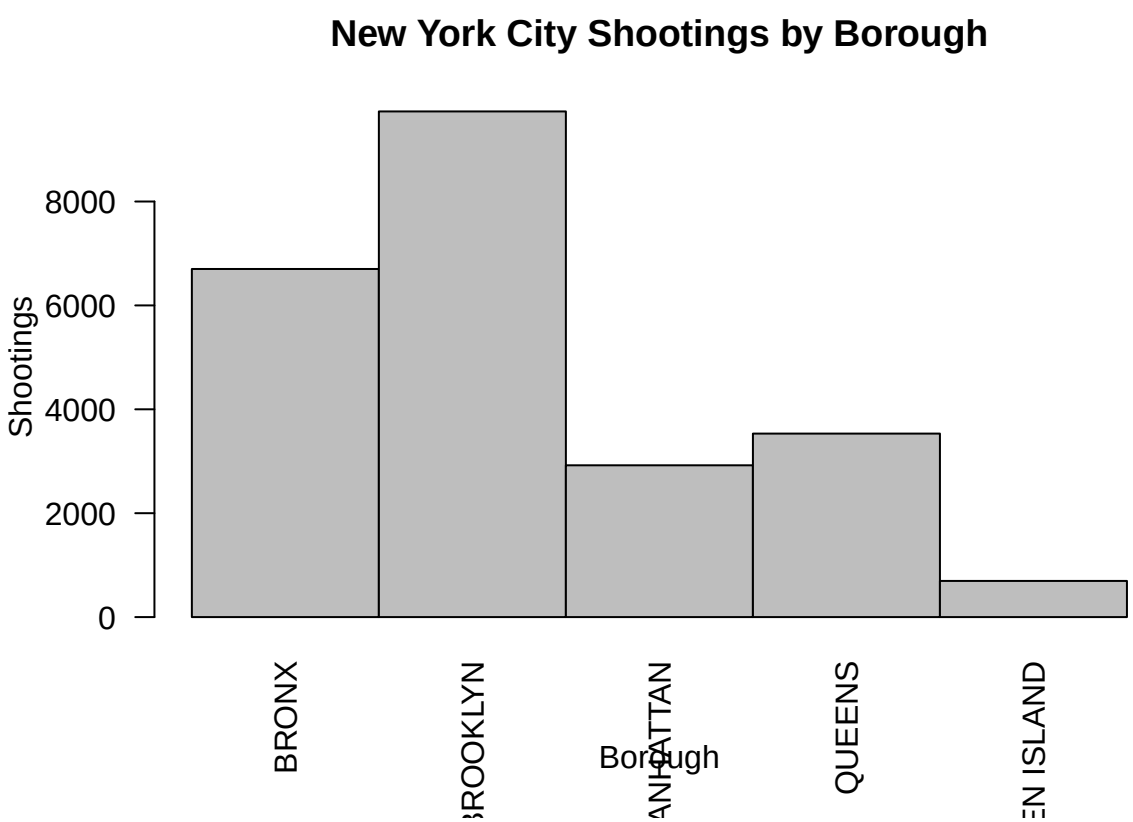
Tidying

I found the data to be clean and thus tidying the data was unnecessary.

Visualizations

Shootings by Borough

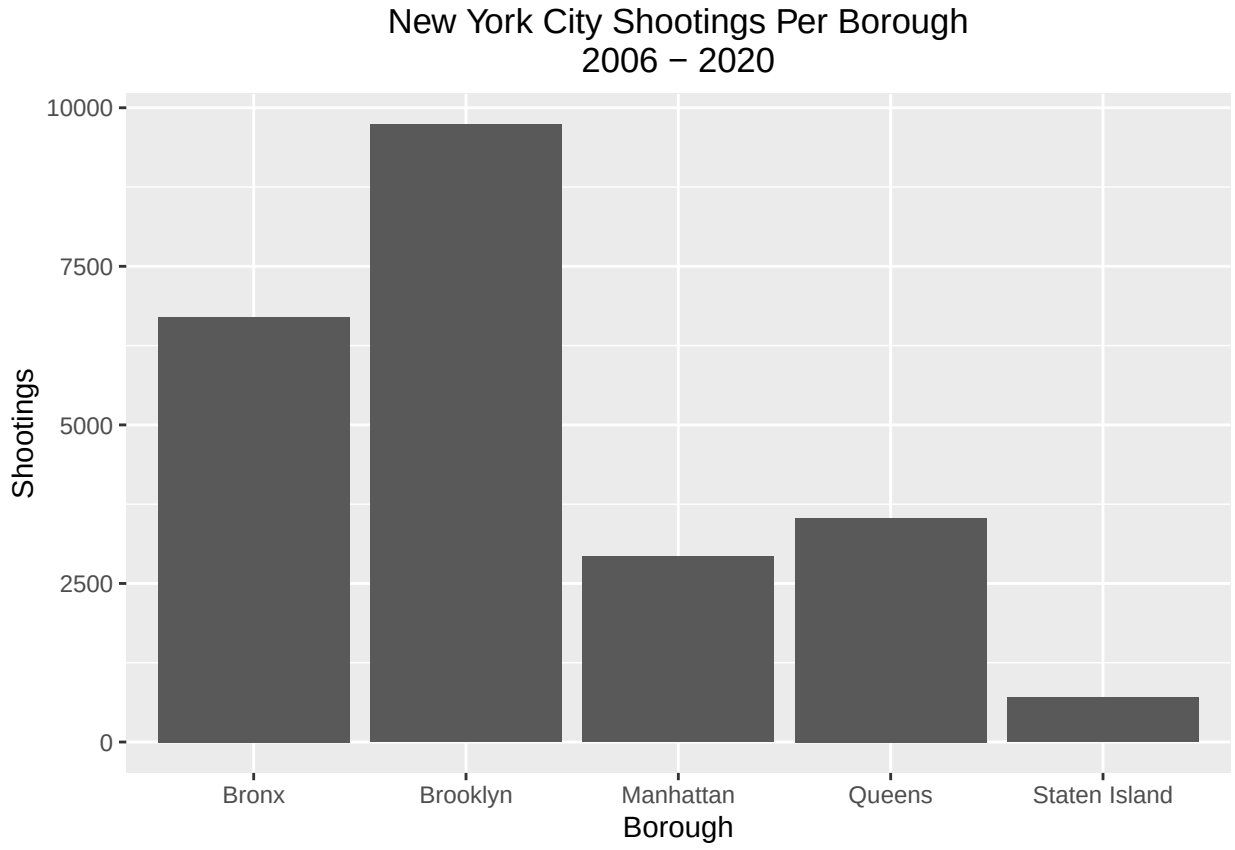
```
barplot(table(df["BORO"]), xlab = "Borough", ylab = "Shootings", space = 0,
        main = "New York City Shootings by Borough", las = 2)
```




```
boroTable <- table(df$BORO)
boroTable
```

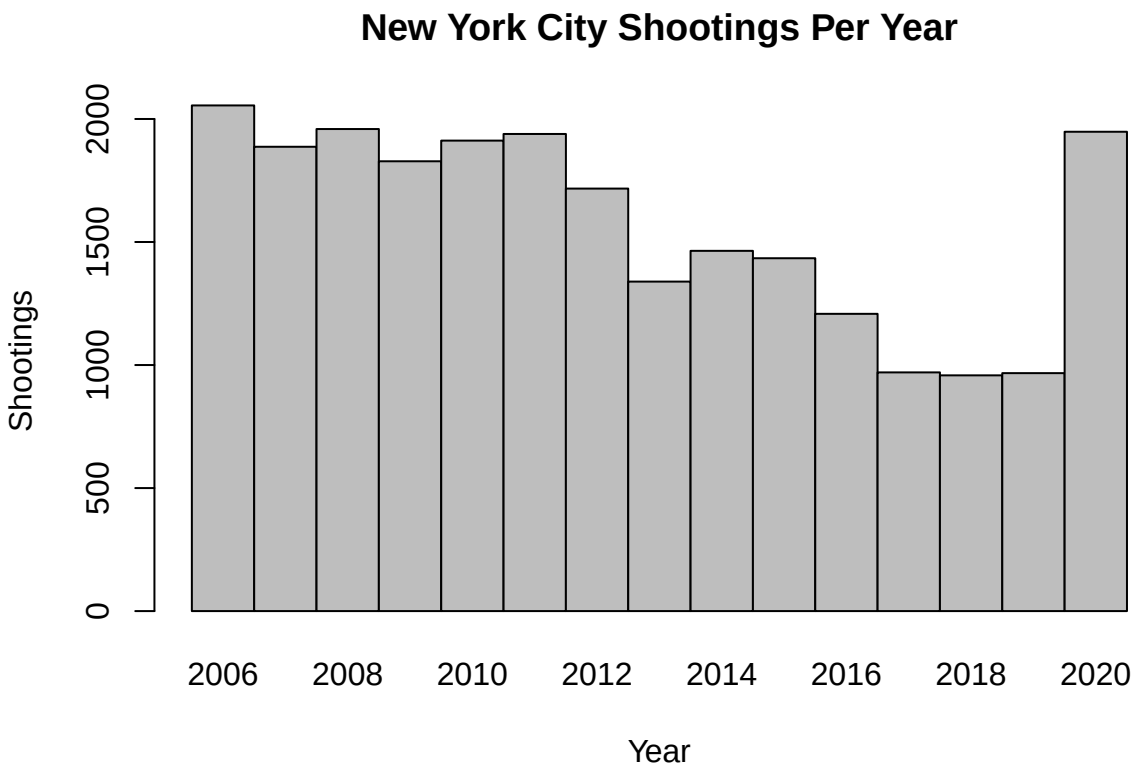
```
##
##         BRONX      BROOKLYN     MANHATTAN        QUEENS STATEN ISLAND
##          6701          9734          2922          3532           696
```

```
boros <- data.frame(
                    Borough = c("Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"),
                    Shootings = c(6701, 9734, 2922, 3532, 696)
)
theme_update(plot.title = element_text(hjust = 0.5))
ggplot(boros, aes(x = Borough, y = Shootings)) +
    geom_bar(stat = "identity") +
    ggtitle("New York City Shootings Per Borough\n2006 - 2020")
```

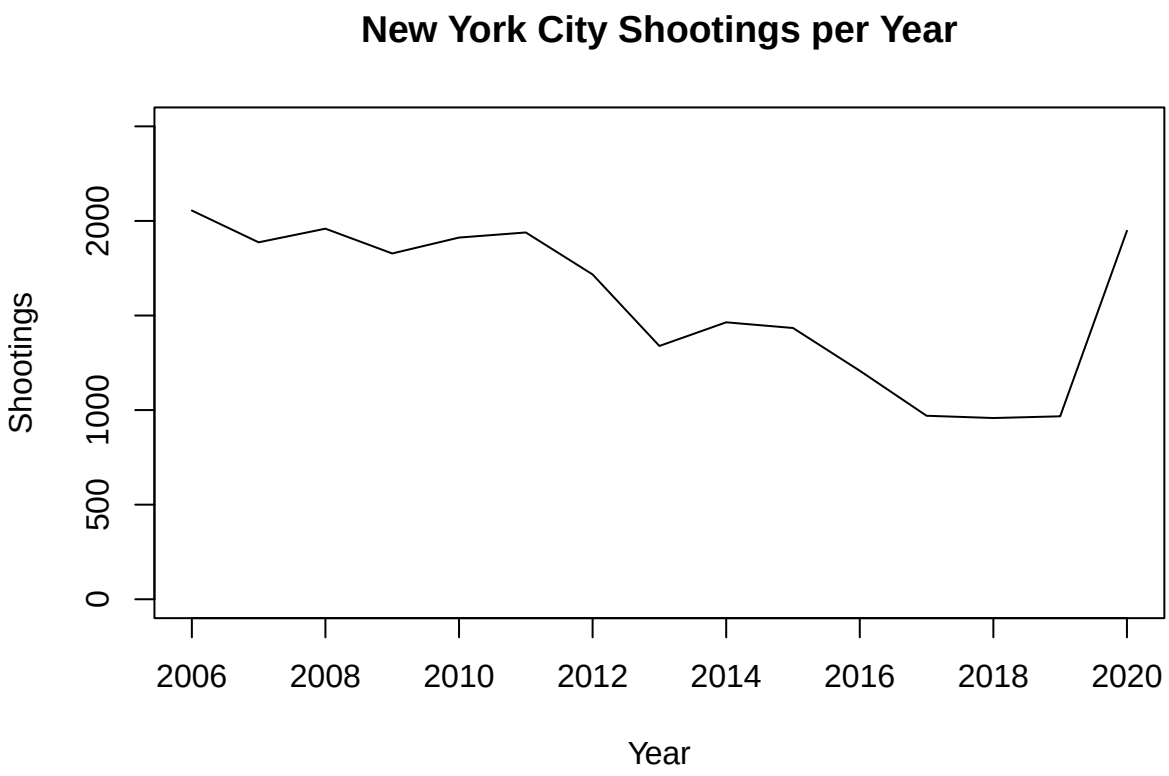


Shootings per year

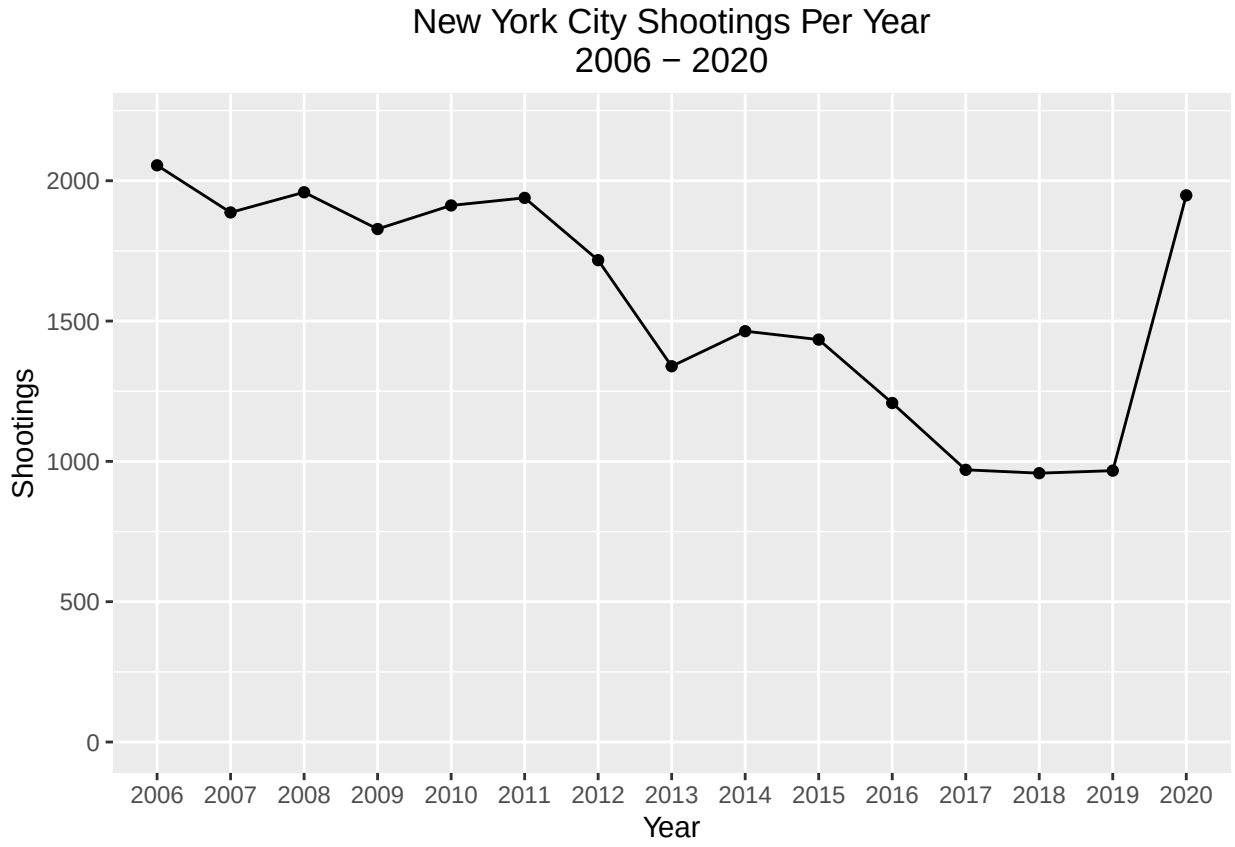
```
shootingsByYear = table(df$YEAR)


barplot(table(df["YEAR"]), xlab = "Year", ylab = "Shootings", space = 0,
        main = "New York City Shootings Per Year")
```




```
plot(names(shootingsByYear), as.vector(shootingsByYear), type = "l",
     xlab = "Year", ylab = "Shootings",
     main = "New York City Shootings per Year", ylim = c(0, 2500))
```



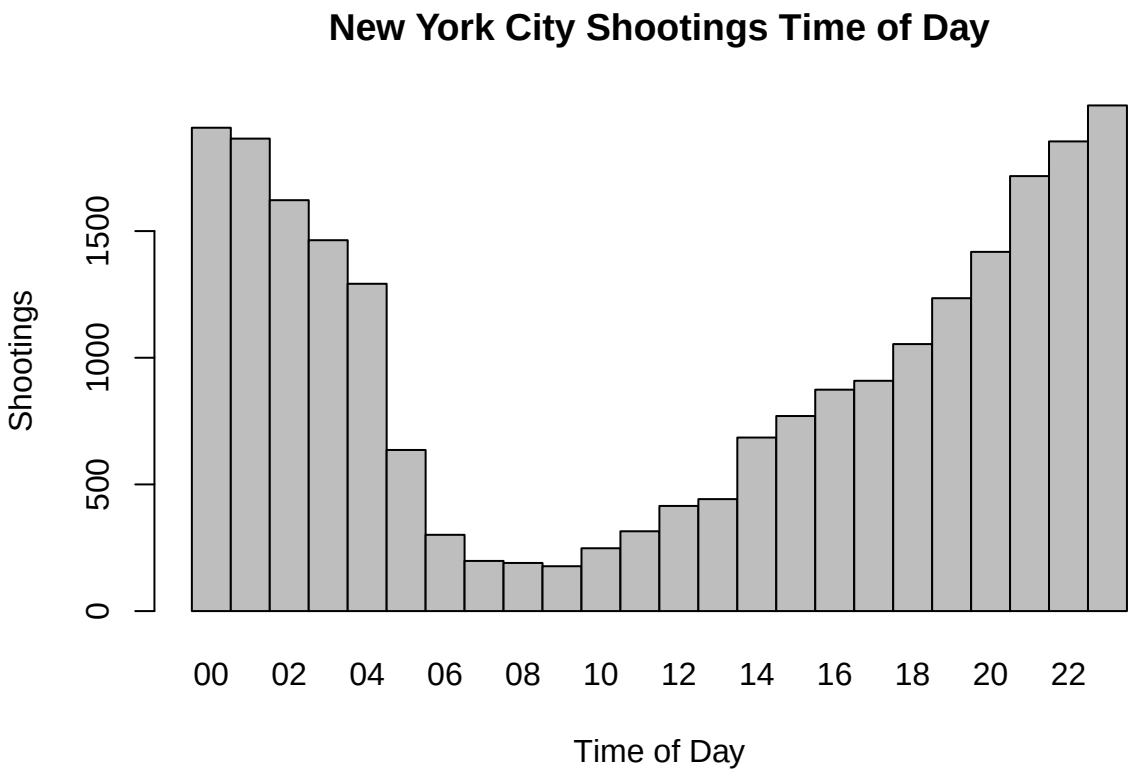

```
shoot_df <- data.frame(
  Shootings = as.vector(shootingsByYear), Year = names(shootingsByYear)
)

ggplot(data = shoot_df, aes(x = Year, y = Shootings, group = 1)) +
    geom_line() + geom_point() + ylim(0, 2200) +
    ggtitle("New York City Shootings Per Year\n2006 - 2020")
```



Analysis Shootings per year declined steadily from 2011 to 2019. Shootings increased significantly in 2020. More analysis is needed to understand if the uptick in shootings was due to the COVID-19 pandemic or some other unrelated cause.

Shootings Time of Day

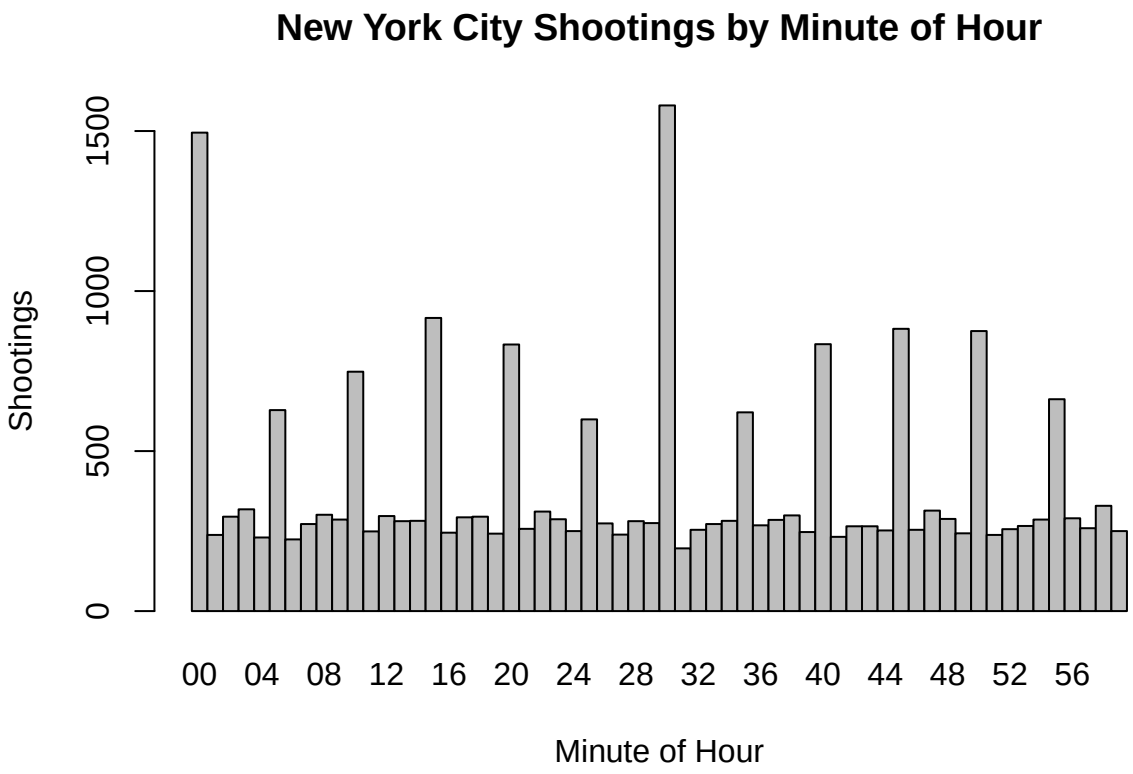
```
barplot(table(df["HOUR"]), xlab = "Time of Day", ylab = "Shootings", space = 0,
        main = "New York City Shootings Time of Day")
```



Analysis Shootings are mostly likely to happen during the night time hours. The safest hours are between 6 and 10 am. Shootings steadily increase from 9 am to midnight, then steadily decline from midnight to 8 am.

Shootings Time of Day Minute

```
barplot(table(df["MINUTE"]), xlab = "Minute of Hour", ylab = "Shootings", space = 0,
        main = "New York City Shootings by Minute of Hour")
```



Analysis Shootings are typically reported in 5 minute intervals around the hour with the most common times being 00 and 30. This is most likely due to exact shooting times being unknown and law enforcement estimating approximate shooting times in official documentation.

Models

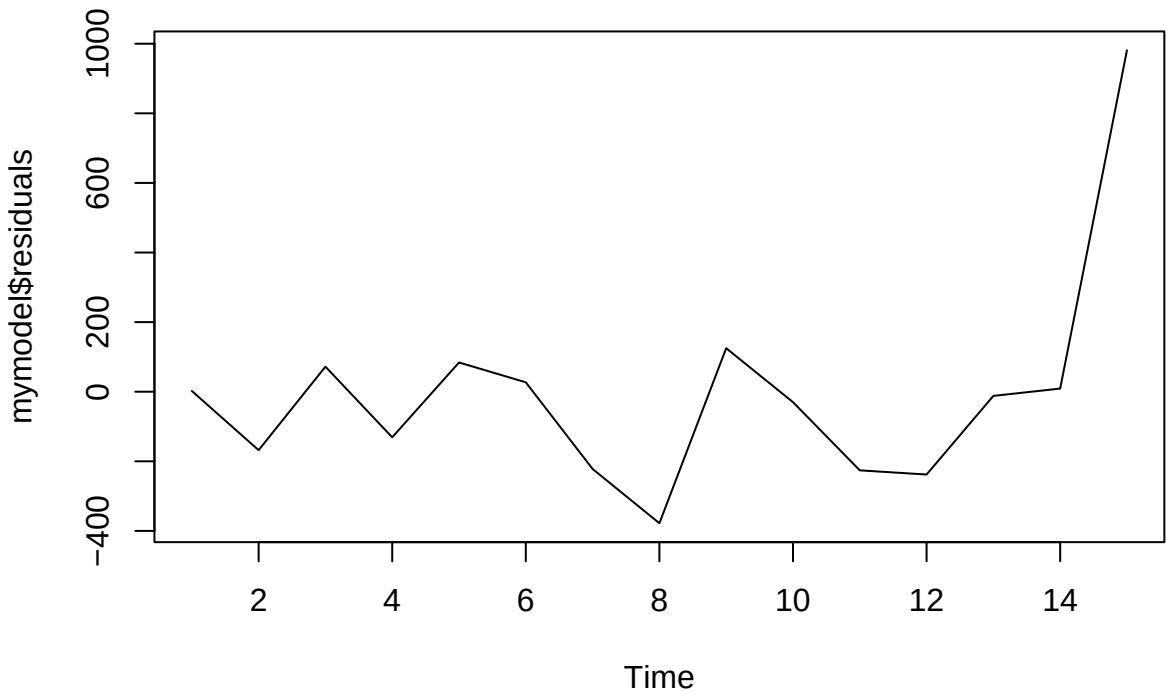
ARIMA

Build a model using the `auto.arima()` function to predict future yearly murder statistics: https://otexts.com/fpp2/arima-r.html

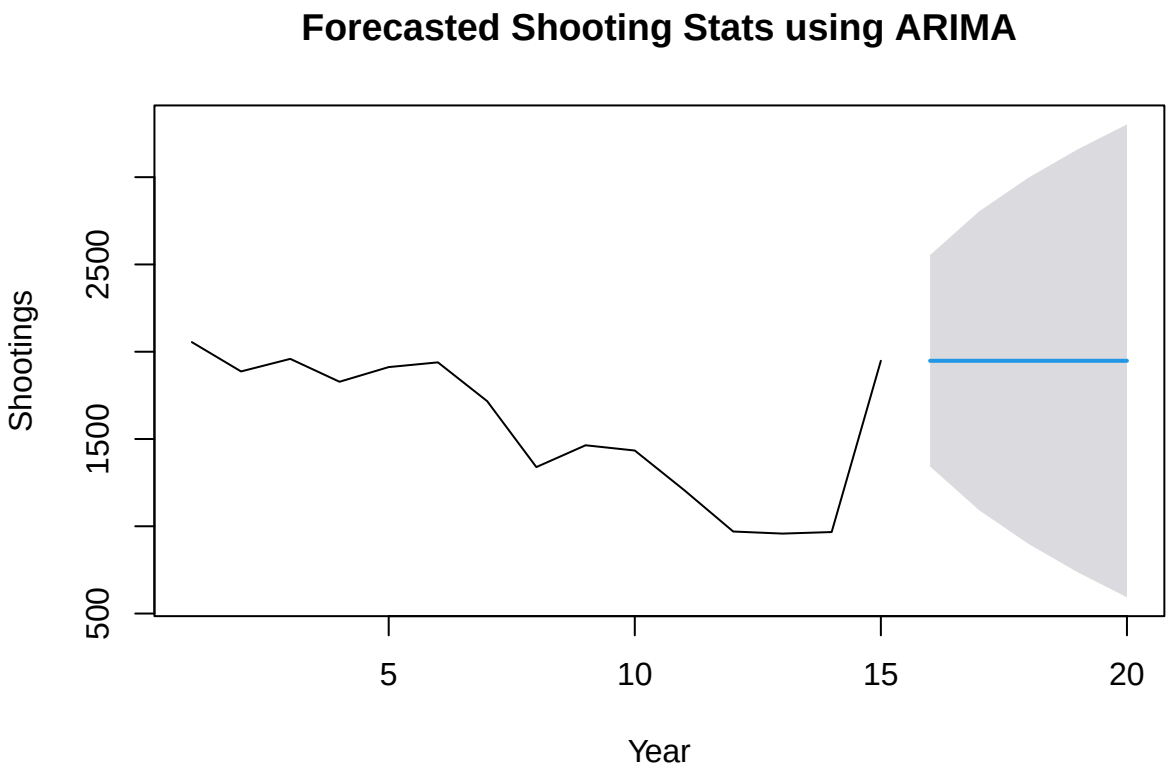
```
shootingsByYear = table(df$YEAR)
mymodel <- auto.arima(as.vector(shootingsByYear))
mymodel
```

```
## Series: as.vector(shootingsByYear)
## ARIMA(0,1,0)
##
## sigma^2 = 95526:  log likelihood = -100.14
## AIC=202.27   AICc=202.6   BIC=202.91
```

```
plot.ts(mymodel$residuals)
```




```
myforecast <- forecast(mymodel, level=c(95), h = 5)
plot(myforecast, xlab="Year", ylab="Shootings", main="Forecasted Shooting Stats using ARIMA")
```



Analysis This model predicts flat shooting statistics in the future. Unfortunately, I do not believe there is enough data for the ARIMA model to accurately predict future yearly shooting statistics. More research, and more granular data is necessary to improve this model.

Bias

As this data analysis is focused on shooting times and dates, I do not believe there are any sources of bias.

R Markdown Session Information

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

```
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forecast_8.16    forcats_0.5.1    stringr_1.4.0    dplyr_1.0.7
##  [5] purrr_0.3.4      readr_2.0.1      tidyr_1.1.3      tibble_3.1.4
##  [9] ggplot2_3.3.5    tidyverse_1.3.1  rmarkdown_2.11   nvimcom_0.9-92.3
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7        lubridate_1.7.10  lattice_0.20-44   zoo_1.8-10
##  [5] assertthat_0.2.1  digest_0.6.27     lmtest_0.9-40     utf8_1.2.2
##  [9] R6_2.5.1          cellranger_1.1.0  backports_1.2.1   reprex_2.0.1
## [13] evaluate_0.14     highr_0.9         httr_1.4.2        pillar_1.6.2
## [17] rlang_1.0.2       curl_4.3.2        readxl_1.3.1      rstudioapi_0.13
## [21] TTR_0.24.3        fracdiff_1.5-1    labeling_0.4.2    munsell_0.5.0
## [25] broom_0.7.9       compiler_4.1.1    modelr_0.1.8      xfun_0.31
## [29] pkgconfig_2.0.3   urca_1.3-0        htmltools_0.5.2   nnet_7.3-16
## [33] tidyselect_1.1.1  quadprog_1.5-8    fansi_0.5.0       crayon_1.4.1
## [37] tzdb_0.1.2        dbplyr_2.1.1      withr_2.5.0       grid_4.1.1
## [41] nlme_3.1-152      jsonlite_1.7.2    gtable_0.3.0      lifecycle_1.0.0
## [45] DBI_1.1.1         magrittr_2.0.1    scales_1.1.1      quantmod_0.4.20
## [49] cli_3.3.0         stringi_1.7.4     farver_2.1.0      tseries_0.10-51
## [53] fs_1.5.0          timeDate_3043.102 xml2_1.3.2        xts_0.12.1
## [57] ellipsis_0.3.2    generics_0.1.0    vctrs_0.3.8       tools_4.1.1
## [61] glue_1.6.2        hms_1.1.0         parallel_4.1.1    fastmap_1.1.0
## [65] yaml_2.2.1        colorspace_2.0-2  rvest_1.0.1       knitr_1.34
## [69] haven_2.4.3
```