NYPD Analysis

This project is part of course 5301 "Data Science as a Field" in the Master of Science in Data Science Degree Programm from the University of Colorado, Boulder. The goal is to show a basic R data science workflow using Rmd, including data cleaning and transformation steps, visualizations and modeling. The analysis will be conducted on the "NYPD Shooting Incidents Data (Historic)" dataset.

Importing the Data and Cleaning

Here is a link to the dataset: https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic

A description of all variables can be found here: https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
nypd <- read.csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")
summary(nypd)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:23568       Length:23568       Length:23568
##  1st Qu.: 55317014   Class :character   Class :character   Class :character
##  Median : 83365370   Mode  :character   Mode  :character   Mode  :character
##  Mean   :102218616
##  3rd Qu.:150772442
##  Max.   :222473262
##
##     PRECINCT      JURISDICTION_CODE LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Min.   :0.0000    Length:23568       Length:23568
##  1st Qu.: 44.00   1st Qu.:0.0000    Class :character   Class :character
##  Median : 69.00   Median :0.0000    Mode  :character   Mode  :character
##  Mean   : 66.21   Mean   :0.3323
##  3rd Qu.: 81.00   3rd Qu.:0.0000
##  Max.   :123.00   Max.   :2.0000
##                   NA's   :2
```

```
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
##  Length:23568       Length:23568       Length:23568       Length:23568
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_SEX            VIC_RACE          X_COORD_CD         Y_COORD_CD
##  Length:23568       Length:23568       Length:23568       Length:23568
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     Latitude       Longitude        Lon_Lat
##  Min.   :40.51   Min.   :-74.25   Length:23568
##  1st Qu.:40.67   1st Qu.:-73.94   Class :character
##  Median :40.70   Median :-73.92   Mode  :character
##  Mean   :40.74   Mean   :-73.91
##  3rd Qu.:40.82   3rd Qu.:-73.88
##  Max.   :40.91   Max.   :-73.70
##
```

```
head(nypd)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME          BORO PRECINCT JURISDICTION_CODE
## 1    201575314 08/23/2019   22:10:00        QUEENS      103                 0
## 2    205748546 11/27/2019   15:54:00         BRONX       40                 0
## 3    193118596 02/02/2019   19:40:00     MANHATTAN       23                 0
## 4    204192600 10/24/2019   00:52:00 STATEN ISLAND      121                 0
## 5    201483468 08/22/2019   18:03:00         BRONX       46                 0
## 6    198255460 06/07/2019   17:50:00      BROOKLYN       73                 0
##   LOCATION_DESC STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX      PERP_RACE
## 1                                 false
## 2                                 false            <18        M          BLACK
## 3                                 false          18-24        M WHITE HISPANIC
## 4     PVT HOUSE                    true          25-44        M          BLACK
## 5                                 false          25-44        M BLACK HISPANIC
## 6                                 false          45-64        M WHITE HISPANIC
##   VIC_AGE_GROUP VIC_SEX       VIC_RACE X_COORD_CD Y_COORD_CD Latitude Longitude
## 1         25-44       M          BLACK    1037451     193561 40.69781 -73.80814
## 2         25-44       F          BLACK    1006789     237559 40.81870 -73.91857
## 3         18-24       M BLACK HISPANIC     999347     227795 40.79192 -73.94548
## 4         25-44       F          BLACK     938149     171781 40.63806 -74.16611
## 5         18-24       M          BLACK    1008224     250621 40.85455 -73.91334
## 6         25-44       M          BLACK    1009650     186966 40.67983 -73.90843
##                                         Lon_Lat
## 1 POINT (-73.80814071699996 40.697805308000056)
## 2  POINT (-73.91857061799993 40.81869973000005)
## 3 POINT (-73.94547965999999 40.791916091000076)
## 4  POINT (-74.16610830199996 40.63806398200006)
## 5  POINT (-73.91333944399999 40.85454734900003)
## 6  POINT (-73.90842523899994 40.67982701600005)
```

Data Cleaning Steps: * Remove columns that are not needed * Change appropriate variables to factor and date types * Check for missing values and handle appropriately

Data Transformation Steps: * Transform `OCCUR_HOUR` variable into categorical variable differentiating only between `DAYTIME` and `NIGHTTIME`

```
# Cleaning steps:
# - Remove unwanted columns for this analysis
# - Change variable types (factor and date)
nypd.cleaned <- nypd %>%
    select(-INCIDENT_KEY, -PRECINCT, -LOCATION_DESC, -JURISDICTION_CODE,
           -X_COORD_CD, -Y_COORD_CD, -Latitude, -Longitude, -Lon_Lat) %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
  mutate(STATISTICAL_MURDER_FLAG = as.logical(STATISTICAL_MURDER_FLAG)) %>%   mutate_at(c("BORO", "STATI
  separate(OCCUR_TIME, c("OCCUR_HOUR", "minute", "second"), sep = ":") %>% select(-minute, -second) %>%
  mutate(OCCUR_HOUR = as.numeric(OCCUR_HOUR)) %>%
  mutate(OCCUR_HOUR = case_when(OCCUR_HOUR < 6 ~ 'NIGHTTIME',
                                OCCUR_HOUR < 18 ~ 'NIGHTTIME',
                                TRUE ~ 'DAYTIME')) %>%
  mutate(OCCUR_HOUR = as.factor(OCCUR_HOUR))
head(nypd.cleaned)
```

```
##   OCCUR_DATE OCCUR_HOUR          BORO STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
## 1 2019-08-23    DAYTIME        QUEENS                   FALSE
## 2 2019-11-27  NIGHTTIME         BRONX                   FALSE            <18
## 3 2019-02-02    DAYTIME     MANHATTAN                   FALSE          18-24
## 4 2019-10-24  NIGHTTIME STATEN ISLAND                    TRUE          25-44
## 5 2019-08-22    DAYTIME         BRONX                   FALSE          25-44
## 6 2019-06-07  NIGHTTIME      BROOKLYN                   FALSE          45-64
##   PERP_SEX      PERP_RACE VIC_AGE_GROUP VIC_SEX       VIC_RACE
## 1                                 25-44       M          BLACK
## 2        M          BLACK         25-44       F          BLACK
## 3        M WHITE HISPANIC         18-24       M BLACK HISPANIC
## 4        M          BLACK         25-44       F          BLACK
## 5        M BLACK HISPANIC         18-24       M          BLACK
## 6        M WHITE HISPANIC         25-44       M          BLACK
```

The cleaned dataset has no `NA` values, although there is a large amount of missing information regarding the perpetrators (sex, age and race). For some part of this analysis (for example time-of-day comparisons or time-series visualizations, see below), the missing information does not matter and the data can still be included in the calculations. However, when demographic data is considered, the missing information should be excluded from the analysis. This is the case when, for example, race is used in some kind of statistic.

```
summary(nypd.cleaned)
```

```
##    OCCUR_DATE              OCCUR_HOUR                BORO
##  Min.   :2006-01-01   DAYTIME  : 9263   BRONX        :6700
##  1st Qu.:2008-12-30   NIGHTTIME:14305   BROOKLYN     :9722
##  Median :2012-02-26                     MANHATTAN    :2921
##  Mean   :2012-10-03                     QUEENS       :3527
##  3rd Qu.:2016-02-28                     STATEN ISLAND: 698
##  Max.   :2020-12-31
##
##  STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX           PERP_RACE
##  FALSE:19080                   :8459     : 8425   BLACK         :9855
##  TRUE : 4488             18-24  :5448   F:  334                 :8425
```

```
##                          25-44  :4613   M:13305   WHITE HISPANIC:1961
##                          UNKNOWN:3156   U: 1504   UNKNOWN       :1869
##                          <18    :1354             BLACK HISPANIC:1081
##                          45-64  : 481             WHITE         : 255
##                          (Other):  57             (Other)       : 122
##   VIC_AGE_GROUP   VIC_SEX                             VIC_RACE
##   <18    : 2525   F: 2195   AMERICAN INDIAN/ALASKAN NATIVE:    9
##   18-24  : 9000   M:21353   ASIAN / PACIFIC ISLANDER      :  320
##   25-44  :10287   U:   20   BLACK                         :16846
##   45-64  : 1536             BLACK HISPANIC                : 2244
##   65+    :  155             UNKNOWN                       :  102
##   UNKNOWN:   65             WHITE                         :  615
##                             WHITE HISPANIC                : 3432
```

According to the dataset, most incidents occur during the nighttime (i.e. from 6 pm to 6 am).

```
nypd.timeofday <- nypd.cleaned %>%
  group_by(OCCUR_HOUR) %>%
  summarize(SUM_CASES = n()) %>%
  ungroup()
nypd.timeofday
```

```
## # A tibble: 2 x 2
##   OCCUR_HOUR SUM_CASES
##   <fct>          <int>
## 1 DAYTIME         9263
## 2 NIGHTTIME      14305
```

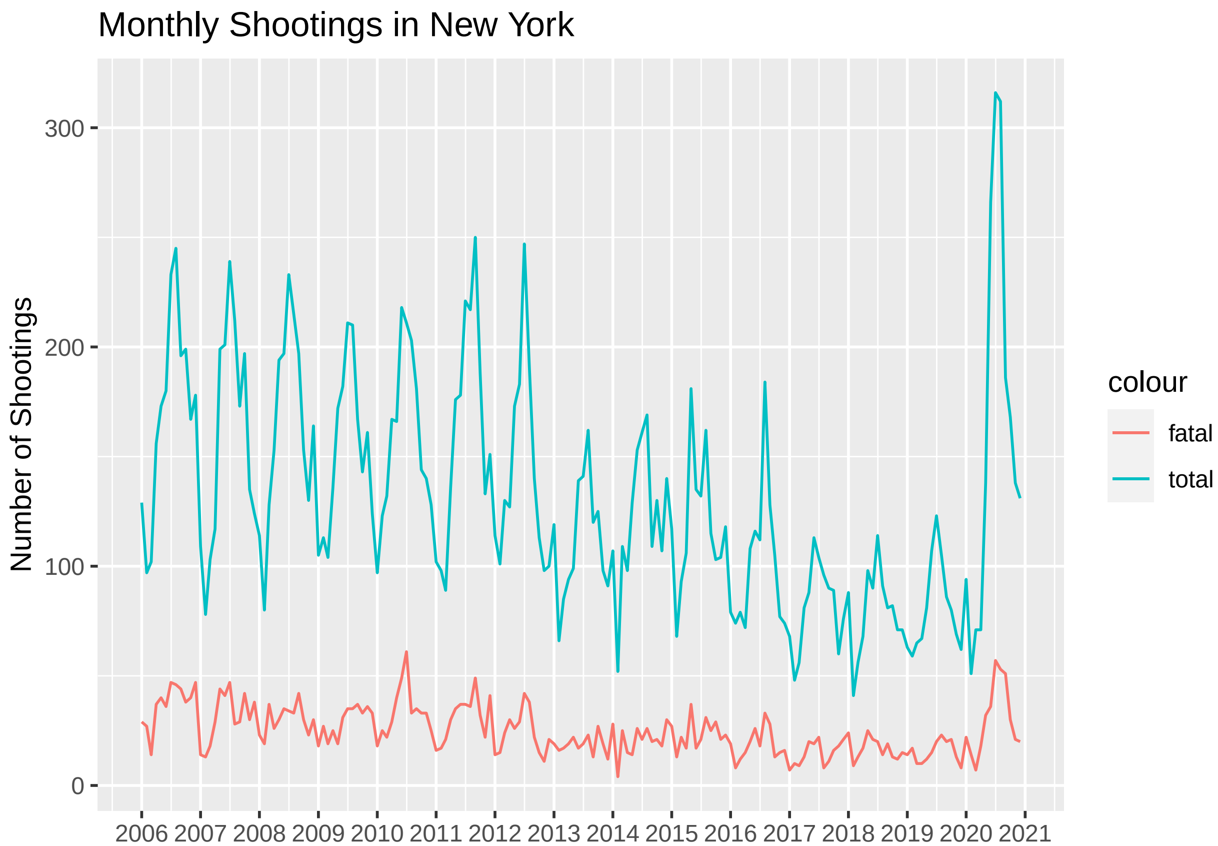
We can see that - when omitting cases where there is not enough information on the perpetrator's race - most shootings are between people of the same race. There are only half as many "interracial" shootings.

```
nypd.racial <- nypd.cleaned %>%
  filter(PERP_RACE != "") %>%
  mutate(INTERRACIAL = if_else(as.character(PERP_RACE)!=as.character(VIC_RACE),
         T, F)) %>%
  group_by(INTERRACIAL) %>%
  summarize(SUM_CASES = n()) %>%
  ungroup()
nypd.racial
```

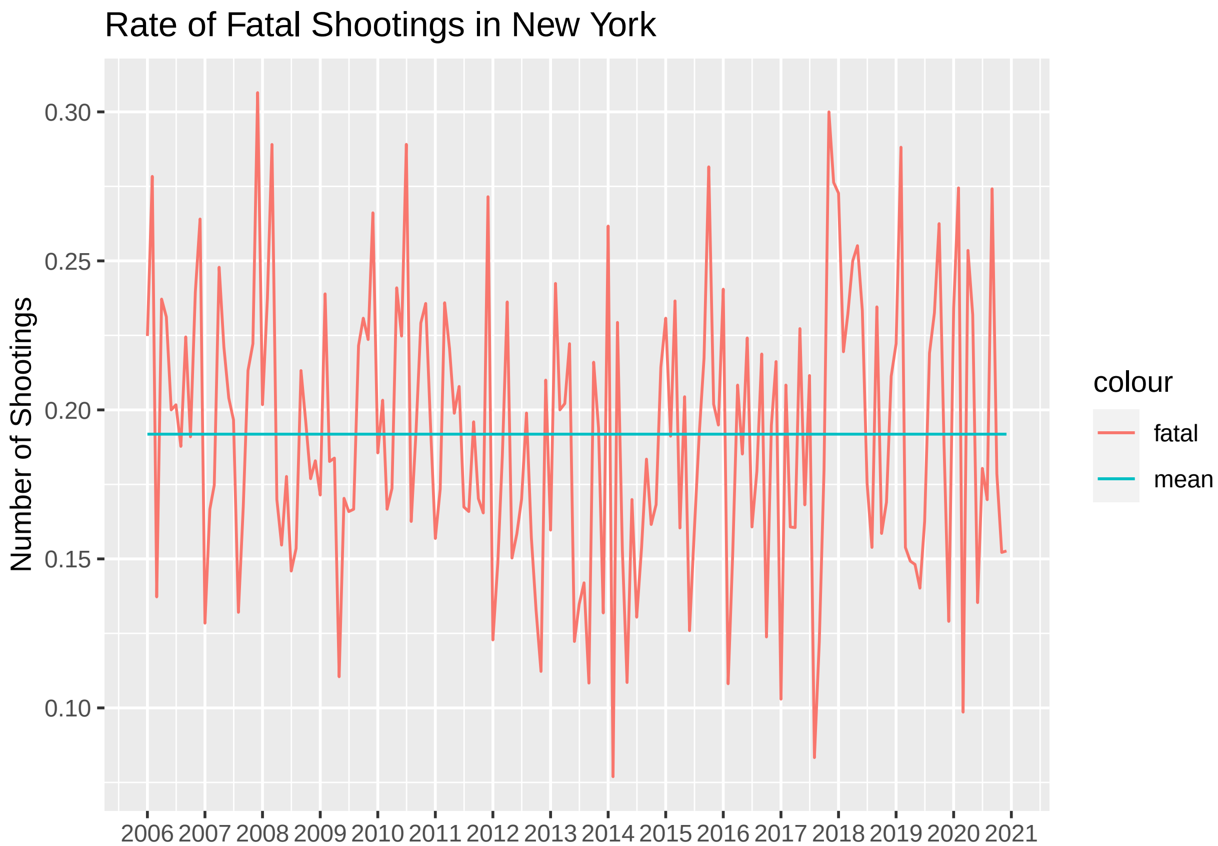
```
## # A tibble: 2 x 2
##   INTERRACIAL SUM_CASES
##   <lgl>           <int>
## 1 FALSE            9157
## 2 TRUE             5986
```

Visualizations

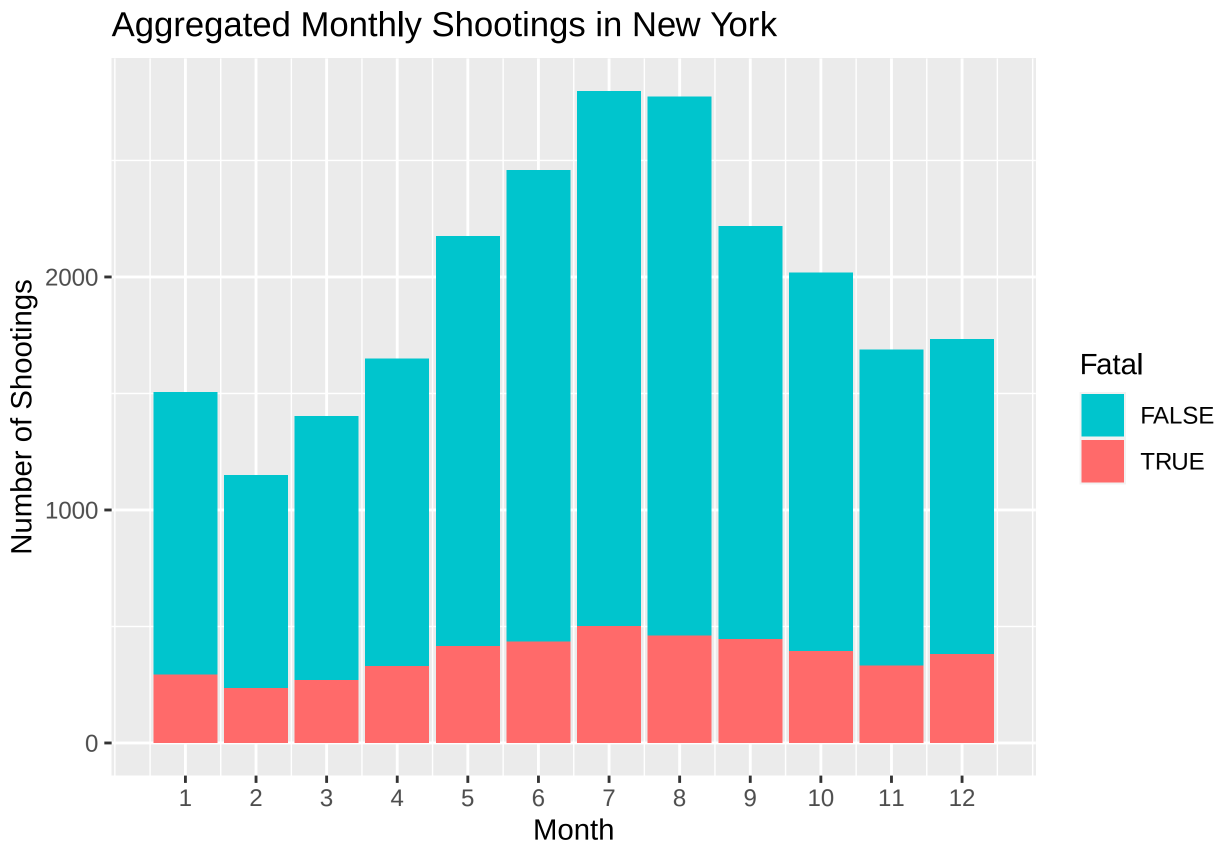
The following plot shows the monthly number of shootings:



Interestingly, we can see a seasonal pattern and an exceptionally high amount of shootings in the middle of 2020 (probably due to the COVID-19 crisis?).



Regarding the frequency of fatal outcomes of shootings, the rate has a large variance but no clear pattern is visible. In total, roughly 20% of all shooting incidents are fatal for the victim.



boxplot-1.pdf

Another interesting question is to further investigate the seasonal pattern that was present in the time series plot from above. From the dataset, it can be seen that the total number of shootings per month is much larger in warmer months than it is in colder months, with the peak being July and the bottom being February. The reason for that can only be guessed, but it could be that in warmer months people tend to go outside more, meet more and, ultimately, have more conflicts than in winter time.

Modeling Shooting Incidents

The goal of my model will be to predict whether a shooting incident will be fatal or not. For this, I will use logistic regression. The data will be split into a training set used to fit the model, and a test set. Since incidents without fatal outcomes are 4 times more likely, undersampling will be used to avoid issues with imbalanced classes. Undersampling is used since there are enough data points available and it is easy to apply, although other techniques might yield better results.

```
head(nypd.cleaned)
```

```
##   OCCUR_DATE OCCUR_HOUR          BORO STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
## 1 2019-08-23    DAYTIME        QUEENS                   FALSE
## 2 2019-11-27  NIGHTTIME         BRONX                   FALSE            <18
## 3 2019-02-02    DAYTIME     MANHATTAN                   FALSE          18-24
## 4 2019-10-24  NIGHTTIME STATEN ISLAND                    TRUE          25-44
## 5 2019-08-22    DAYTIME         BRONX                   FALSE          25-44
## 6 2019-06-07  NIGHTTIME      BROOKLYN                   FALSE          45-64
##   PERP_SEX      PERP_RACE VIC_AGE_GROUP VIC_SEX       VIC_RACE
## 1                                 25-44       M          BLACK
## 2        M          BLACK         25-44       F          BLACK
## 3        M WHITE HISPANIC         18-24       M BLACK HISPANIC
## 4        M          BLACK         25-44       F          BLACK
```

```
## 5        M BLACK HISPANIC         18-24       M           BLACK
## 6        M WHITE HISPANIC         25-44       M           BLACK
```

```
# train-test split
bound = floor(nrow(nypd.cleaned)*0.8)
df = nypd.cleaned[sample(nrow(nypd.cleaned)), ]
df_train = df[1:bound, ]
df_test = df[(bound+1):nrow(df), ]

# balance training set
df_train_notfatal = df_train[df_train$STATISTICAL_MURDER_FLAG==F,]
df_train_fatal = df_train[df_train$STATISTICAL_MURDER_FLAG==T,]
df_train_notfatal = df_train_notfatal[sample(nrow(df_train_fatal)),]
df_train_balanced = rbind(df_train_fatal, df_train_notfatal)
df_train_balanced = df_train_balanced[sample(nrow(df_train_balanced)), ]

cat("Train Set Dimensions: \t(", dim(df_train_balanced)[1],
    ",", dim(df_train_balanced)[2], ")",
    "\nTest Set Dimensions: \t(", dim(df_test)[1], " ,", dim(df_test)[2], ")")
```

```
## Train Set Dimensions:    ( 7182 , 10 )
## Test Set Dimensions:     ( 4714  , 10 )
```

Here is how the model is trained. All non-significant predictors were removed. Note: You will learn more about generalized linear models in DTSA 5013 (I have already taken this course).

```
#nypd.cleaned$PERP_RACE
nypd.mod = glm(STATISTICAL_MURDER_FLAG ~
                 PERP_AGE_GROUP + VIC_AGE_GROUP,
               family="binomial", df_train_balanced)
summary(nypd.mod)
```

```
##
## Call:
## glm(formula = STATISTICAL_MURDER_FLAG ~ PERP_AGE_GROUP + VIC_AGE_GROUP,
##     family = "binomial", data = df_train_balanced)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.82927  -1.16868   0.03501   1.12948   2.17848
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.38643    0.09033  -4.278 1.89e-05 ***
## PERP_AGE_GROUP<18      0.06869    0.10913   0.629 0.529092
## PERP_AGE_GROUP18-24    0.10076    0.06352   1.586 0.112661
## PERP_AGE_GROUP25-44    0.38805    0.06451   6.015 1.79e-09 ***
## PERP_AGE_GROUP45-64    0.54968    0.15885   3.460 0.000539 ***
## PERP_AGE_GROUP65+      0.88695    0.51727   1.715 0.086401 .
## PERP_AGE_GROUPUNKNOWN -1.88862    0.12268 -15.395  < 2e-16 ***
## VIC_AGE_GROUP18-24     0.26508    0.09270   2.859 0.004244 **
## VIC_AGE_GROUP25-44     0.50023    0.09161   5.460 4.75e-08 ***
## VIC_AGE_GROUP45-64     0.65895    0.12516   5.265 1.40e-07 ***
## VIC_AGE_GROUP65+       1.46364    0.32466   4.508 6.54e-06 ***
## VIC_AGE_GROUPUNKNOWN   0.29756    0.39548   0.752 0.451816
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9956.4  on 7181  degrees of freedom
## Residual deviance: 9383.1  on 7170  degrees of freedom
## AIC: 9407.1
##
## Number of Fisher Scoring iterations: 4
```

The evaluation shows a relatively low F1-score, meaning that it is hard to predict whether a shooting will be fatal or not, given this dataset. The metrics on the test dataset show that only about 63% of incidents are predicted correctly, with a relatively large false positive rate (which may not be too bad in a real-world application on predictive crime). Out of the fatal shootings, roughly half of them are predicted as such.

```
predictions = round(predict(nypd.mod, df_test, type='response'))
sum(predictions)
```

```
## [1] 2483
```

```
truth = as.integer(as.logical(df_test$STATISTICAL_MURDER_FLAG))

tp = 0
tn = 0
fp = 0
fn = 0
for(i in 1:length(predictions)) {
  # truth is negative
  if(truth[i] == 0){
      if(predictions[i] == 0){tn = tn+1}
      else {fp = fp+1}
  }
  # truth is positive
  if(truth[i] == 1){
      if(predictions[i] == 1){tp = tp+1}
      else {fn = fn+1}
  }
}

cat('True Positive:', tp, 'False Positive:', fp,
    '\nFalse Negative:', fn, 'True Negative:', tn)
```

```
## True Positive: 600 False Positive: 1883
## False Negative: 297 True Negative: 1934
```

```
accuracy = (tp+tn)/(tp+tn+fp+fn)
precision = tp/(tp+fp)
recall = tp/(tp+fn)
f.score = 2*precision*recall/(precision+recall)

cat('Accuracy:', round(accuracy, 2),
    'Precision:', round(precision, 2),
    'Recall:', round(recall, 2),
    'F-score:', round(f.score, 2))
```

```
## Accuracy: 0.54 Precision: 0.24 Recall: 0.67 F-score: 0.36
```

Bias Identification

As a European citizen, I am biased towards being in favor of stricter firearm regulation laws. This might lead to analyzing the dataset in a way that supports such policies. I tried to mitigate the bias by considering all aspects of the dataset and not only pick the ones that support my beliefs for this analysis.

R Session Info

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.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] lubridate_1.7.10 forcats_0.5.1    stringr_1.4.0    dplyr_1.0.6
##  [5] purrr_0.3.4      readr_1.4.0      tidyr_1.1.3      tibble_3.1.2
##  [9] ggplot2_3.3.3    tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.23         haven_2.4.1       colorspace_2.0-1
##  [5] vctrs_0.3.8       generics_0.1.0    htmltools_0.5.1.1 yaml_2.2.1
##  [9] utf8_1.2.1        rlang_0.4.11      pillar_1.6.1      glue_1.4.2
## [13] withr_2.4.2       DBI_1.1.1         dbplyr_2.1.1      modelr_0.1.8
## [17] readxl_1.3.1      lifecycle_1.0.0   munsell_0.5.0     gtable_0.3.0
## [21] cellranger_1.1.0  rvest_1.0.0       evaluate_0.14     labeling_0.4.2
## [25] knitr_1.33        fansi_0.5.0       highr_0.9         broom_0.7.6
## [29] Rcpp_1.0.6        scales_1.1.1      backports_1.2.1   jsonlite_1.7.2
## [33] farver_2.1.0      fs_1.5.0          hms_1.1.0         digest_0.6.27
## [37] stringi_1.6.2     grid_4.1.0        cli_2.5.0         tools_4.1.0
## [41] magrittr_2.0.1    crayon_1.4.1      pkgconfig_2.0.3   ellipsis_0.3.2
## [45] xml2_1.3.2        reprex_2.0.0      assertthat_0.2.1  rmarkdown_2.9
## [49] httr_1.4.2        rstudioapi_0.13   R6_2.5.0          compiler_4.1.0
```