MSDS NYPD Shooting Incidents Investigation and Analysis

J. Dean

2025-06-20

This document will be a light investigation and analysis on the “NYPD Shooting Incidents Data (Historic)” dataset. The packages used are the `tidyverse`, `gridExtra`, and `nnet` packages. All steps of the data science process will be displayed, starting with importing and tidying the data.

Importing and Tidying the Data

The data is a Non-Federal dataset from the US Data Catalog. First we import the data.

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

```
## Rows: 29744 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (12): OCCUR_DATE, BORO, LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION...
## dbl   (5): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, Latitude, Longitude
## num   (2): 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.
```

Then we tidy the data. The thought process behind our tidying decisions will be explained in the code below.

```
# Start by renaming the columns as desired.
# The 'date' variable will be made a 'date' type from the 'lubridate' package.

nypd_data <- nypd_data |>
  rename("date" = OCCUR_DATE, "time" = OCCUR_TIME, "borough" = BORO,
         "loc_occ" = LOC_OF_OCCUR_DESC, "precinct" = PRECINCT,
         "jurisdiction" = JURISDICTION_CODE, "loc_class" = LOC_CLASSFCTN_DESC,
         "loc_desc" = LOCATION_DESC, "victim_death" = STATISTICAL_MURDER_FLAG,
         "perp_age" = PERP_AGE_GROUP, "perp_sex" = PERP_SEX,
         "perp_race" = PERP_RACE, "victim_age" = VIC_AGE_GROUP,
         "victim_sex" = VIC_SEX, "victim_race" = VIC_RACE) |>
  mutate(date = mdy(date),

         # Unifying the missing or data points into one value,
         # which is chosen to be 'NA'.
```

```
       # The uncategorized data will be lumped in as well, so long as it does
       # not comprise a significant part of a variable's available data.
       loc_class = ifelse(loc_class %in% c("(null)", "OTHER"), NA, loc_class),
       loc_desc = ifelse(loc_desc == "(null)", NA, loc_desc),
       perp_age = ifelse(perp_age %in% c("(null)", "1020", "1028", "2021",
                                         "224", "940"), NA, perp_age),
       perp_sex = ifelse(perp_sex %in% c("(null)", "U"), NA, perp_sex),
       perp_race = ifelse(perp_race == "(null)", NA, perp_race),
       victim_age = ifelse(victim_age == "1022", NA, victim_age),
       victim_sex = ifelse(victim_sex == "U", NA, victim_sex),

       # Setting the classification variables to factor types.
       borough = as.factor(borough),
       loc_occ = as.factor(loc_occ),
       loc_class = as.factor(loc_class),
       loc_desc = as.factor(loc_desc),
       precinct = as.factor(precinct),
       jurisdiction = as.factor(jurisdiction),
       perp_age = as.factor(perp_age),
       perp_sex = as.factor(perp_sex),
       perp_race = as.factor(perp_race),
       victim_age = as.factor(victim_age),
       victim_sex = as.factor(victim_sex),
       victim_race = as.factor(victim_race)
       ) |>
# Excluding the data we will not be analyzing in this document.
select(-c(INCIDENT_KEY, X_COORD_CD, Y_COORD_CD, Latitude, Longitude, Lon_Lat))
```

Here is a quick summary of the data after tidying.

```
summary(nypd_data)
```

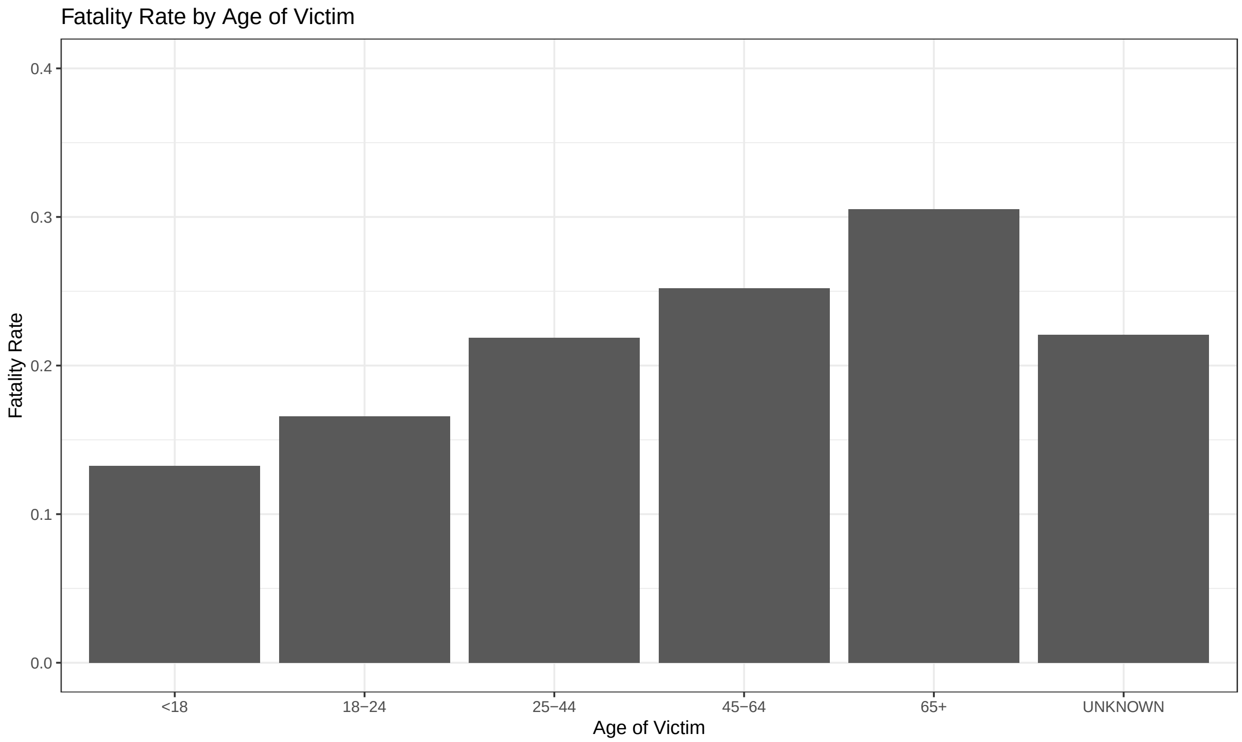
```
##       date                  time                   borough        loc_occ
##  Min.   :2006-01-01   Length:29744      BRONX        : 8834   INSIDE :  682
##  1st Qu.:2009-10-29   Class1:hms        BROOKLYN     :11685   OUTSIDE: 3466
##  Median :2014-03-25   Class2:difftime   MANHATTAN    : 3977   NA’s   :25596
##  Mean   :2014-10-31   Mode  :numeric    QUEENS       : 4426
##  3rd Qu.:2020-06-29                     STATEN ISLAND:  822
##  Max.   :2024-12-31
##
##     precinct     jurisdiction      loc_class
##  75     : 1680   0   :24957   STREET    : 2639
##  73     : 1561   1   :  109   HOUSING   :  643
##  67     : 1288   2   : 4676   DWELLING  :  341
##  44     : 1159   NA’s:    2   COMMERCIAL:  276
##  79     : 1073                PLAYGROUND:   67
##  47     : 1048                (Other)   :  101
##  (Other):21935                NA’s      :25677
##                         loc_desc     victim_death       perp_age     perp_sex
##  MULTI DWELL - PUBLIC HOUS: 5188   Mode :logical   <18     : 1805   F   :  461
##  MULTI DWELL - APT BUILD  : 3042   FALSE:23979     18-24  : 6630   M   :16845
##  PVT HOUSE                : 1010   TRUE :5765      25-44  : 6342   NA’s:12438
##  GROCERY/BODEGA           :  775                   45-64  :  775
##  BAR/NIGHT CLUB           :  695                   65+    :   67
##  (Other)                  : 1531                   UNKNOWN: 3148
##  NA's                     :17503                   NA's   :10977
##           perp_race        victim_age    victim_sex
##  BLACK         :12323   <18    : 3081   F    : 2891
##  WHITE HISPANIC: 2667   18-24  :10677   M    :26841
##  UNKNOWN       : 1838   25-44  :13563   NA's:   12
##  BLACK HISPANIC: 1487   45-64  : 2118
##  WHITE         :  305   65+    :  236
##  (Other)       :  186   UNKNOWN:   68
##  NA's          :10938   NA's   :    1
##                         victim_race
##  AMERICAN INDIAN/ALASKAN NATIVE:   13
##  ASIAN / PACIFIC ISLANDER      :  478
##  BLACK                         :20999
##  BLACK HISPANIC                : 2930
##  UNKNOWN                       :   72
##  WHITE                         :  741
##  WHITE HISPANIC                : 4511
```

As you might be able to tell, there is a lot of missing data in this dataset. Some entire variables are mainly composed of `NA` values, such as `loc_occ` which is composed of over 80% missing data. There is also data that is poorly entered, such as age ranges claiming a perpetrator was 1020 years old, although this can be assumed to have the intention to record an age range of 10-20 years. The poorly entered data is sparse, usually only having one data point, and representing roughly zero percent of the data. These single data point outliers will be removed from consideration in this document. The questions of what to do with the missing data and addressing bias will be handled on a case-specific basis.

Understanding the Data

In order to understand the data, we must follow the cycle of `transform -> visualize -> model -> repeat`. I have already gone through the cycle, so I will just show the results. Let's first investigate whether the victim's age has any effect on the lethality of the shooting. Common sense dictates that the older the victim, the more lethal the shooting should be, but the data will show us. The following bar graph shows the `fatality` rate depending on the `victim_age`.

```r
nypd_data |>
  select(victim_age, victim_death)  |>
  # Removing missing data, <0.3% of the whole. Causes minimal bias.
  filter(is.na(victim_age) == FALSE) |>
  group_by(victim_age) |>
  summarize(fatality = sum(victim_death)/n()) |> #Calculating the fatality rate

  # Plotting the data.
  ggplot(aes(x = victim_age, y = fatality)) +
  geom_bar(stat = "identity") +

  # Themes, labels, and limits for the plot.
  theme_bw() +
  ylim(0, 0.4) + ylab("Fatality Rate") +
  xlab("Age of Victim") +
  ggtitle("Fatality Rate by Age of Victim")
```



As you can see, fatality rates follow a fairly linear trend: the older the victim, the more fatal the shooting. Due to time constraints we will not be creating and evaluating a linear model to this particular relation.

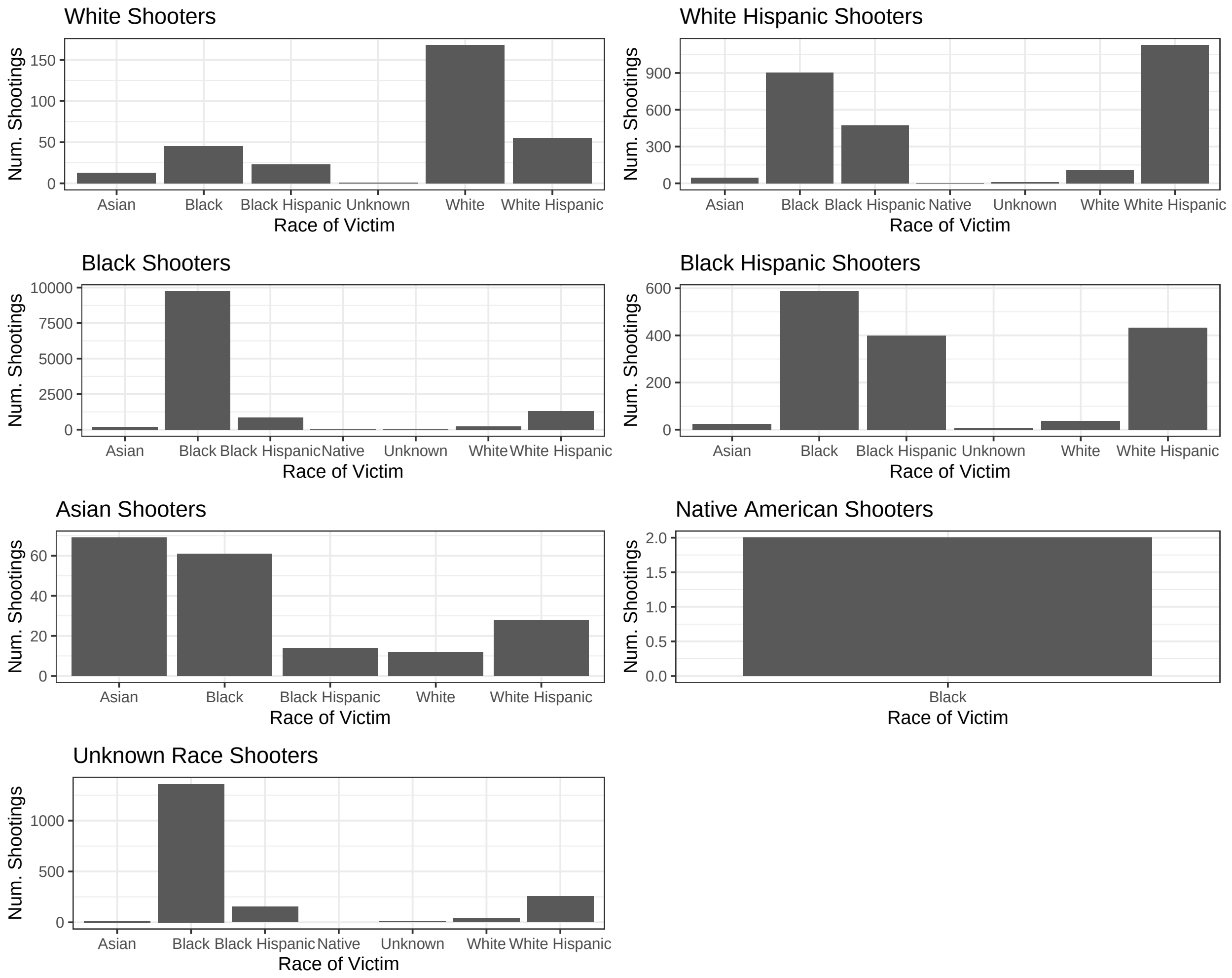
Race-on-Race shootings analysis

The phrase "Black-on-Black Crime" is a heavily loaded phrase that we have the potential to partially investigate using this dataset. We will first create a graph that displays the race of the victim based on one particular race of shooters, ie. what is the racial demographic of victims of white shooters? We repeat this process for all applicable races of shooters. Then we use the `grid.arrange` function from the `gridExtra` package to display all of these graphs together, allowing for simpler comparison.

```r
# Making a tibble specific for this "Race-on-Race" comparison.
race_on_race <- nypd_data |>

  # Eliminating all missing data here introduces bias that must be accounted for
  filter(is.na(perp_race)==FALSE, is.na(victim_race)==FALSE) |>
  select(perp_race, victim_race) |>

  # This large `mutate` function is simply a relabeling of our categories to
  # make our graphs easier to understand later.
  mutate(victim_race = ifelse(victim_race == "WHITE", "White",
                       ifelse(victim_race == "WHITE HISPANIC", "White Hispanic",
                       ifelse(victim_race == "BLACK", "Black",
                       ifelse(victim_race == "BLACK HISPANIC", "Black Hispanic",
                       ifelse(victim_race == "AMERICAN INDIAN/ALASKAN NATIVE",
                              "Native",
                       ifelse(victim_race == "ASIAN / PACIFIC ISLANDER", "Asian",
                       ifelse(victim_race == "UNKNOWN", "Unknown", victim_race))))))))
```

```
# Creating all the graphs and store them as variables.

# White Shooters
p_white <- race_on_race |>
  filter(perp_race == "WHITE") |>
  group_by(victim_race) |>
ggplot(aes(x=victim_race)) +
  geom_bar() + theme_bw() +
  ylab("Num. Shootings") + xlab("Race of Victim") +
  ggtitle("White Shooters")


# White Hispanic Shooters
p_white_hispanic <- race_on_race |>
  filter(perp_race == "WHITE HISPANIC") |>
  group_by(victim_race) |>
ggplot(aes(x=victim_race)) +
  geom_bar() + theme_bw() +
  ylab("Num. Shootings") + xlab("Race of Victim") +
  ggtitle("White Hispanic Shooters")


# Black Shooters
p_black <- race_on_race |>
  filter(perp_race == "BLACK") |>
  group_by(victim_race) |>
ggplot(aes(x=victim_race)) +
  geom_bar() + theme_bw() +
  ylab("Num. Shootings") + xlab("Race of Victim") +
  ggtitle("Black Shooters")


# Native American Shooters
p_native <- race_on_race |>
  filter(perp_race == "AMERICAN INDIAN/ALASKAN NATIVE") |>
  group_by(victim_race) |>
ggplot(aes(x=victim_race)) +
  geom_bar() + theme_bw() +
  ylab("Num. Shootings") + xlab("Race of Victim") +
  ggtitle("Native American Shooters")


# Asian Shooters
p_aapi <- race_on_race |>
  filter(perp_race == "ASIAN / PACIFIC ISLANDER") |>
  group_by(victim_race) |>
ggplot(aes(x=victim_race)) +
  geom_bar() + theme_bw() +
  ylab("Num. Shootings") + xlab("Race of Victim") +
  ggtitle("Asian Shooters")
```

```r
# Black Hispanic Shooters
p_black_hispanic <- race_on_race |>
  filter(perp_race == "BLACK HISPANIC") |>
  group_by(victim_race) |>
ggplot(aes(x=victim_race)) +
  geom_bar() + theme_bw() +
  ylab("Num. Shootings") + xlab("Race of Victim") +
  ggtitle("Black Hispanic Shooters")

# Shooters of unknown race
p_unknown <- race_on_race |>
  filter(perp_race == "UNKNOWN") |>
  group_by(victim_race) |>
ggplot(aes(x=victim_race)) +
  geom_bar() + theme_bw() +
  ylab("Num. Shootings") + xlab("Race of Victim") +
  ggtitle("Unknown Race Shooters")


# Putting all the graphs together using `grid.arrange` for an easier comparison.
grid.arrange(p_white, p_white_hispanic, p_black, p_black_hispanic, p_aapi,
             p_native, p_unknown, ncol=2)
```



Several things must be taken into account when analyzing this visual. The first is the scale of the y-axis. The scale is orders of magnitude different depending on which graph you look at. These unequal scales are necessary to properly illustrate the proportions that are the key point of this comparison. The bias present in this plot will be addressed near the end of this document.

The first thing that may stand out as potential for further investigation is that the majority of shootings are intraracial, ie. White shooters mainly shoot White people. However, in the case of Black Hispanic shooters, the majority of victims are Black people, followed by White Hispanic, and only then fellow Black Hispanic people. Another point of note is that Black victims make up a fairly sizable percentage of victims of all races of shooters. However, White shooters have shot slightly more White Hispanic people than Black people, and Native American people have been recorded as shooting people only two times in the 18 years this dataset covers.

This visualization brings plenty of questions and threads for further investigation. Unfortunately due to time constraints, we will be unable to explore most of these in this document.

Shooter's age Analysis

We will only focus on one question for now. Since over 40% of the `perp_race` variable consists of missing values, can we use the victim's race to predict the race of the shooter? This question holds significant potential for stereotyping, bias, and racism, so we will not be tackling it in this document. I am not currently qualified to tackle this as a fledgling data scientist. However, we will focus on something similar:

since over 30% of the shootings were carried out by shooters of unknown age, can we predict the shooter's age based on the age and race of each victim? While a slightly less effective question in terms of drawing meaningful conclusions, it is most certainly less potent and will do just well for our purposes.

Since I am a fledgling data scientist, and due to time constraints, we will be using a logistic regression model to tackle this question. The variables we are working with are all categorical, and only two of which have natural ordering: the ages of the victim and perpetrator, whereas the victim's race and sex have no natural ordering. This means we will use a multinomial regression model, using the `multinom` function of the `nnet` package.

First we transform the data into a useful form.

```r
age_pred_data <- nypd_data |>

  # Select the variables we want to use, and remove the missing data.
  # This leaves us with 15,528 data points.
  select(perp_age, victim_race, victim_age) |>
  filter(is.na(perp_age)==FALSE,
         is.na(victim_race)==FALSE,
         is.na(victim_age)==FALSE) |>

  # This large `mutate` function is simply a relabeling of our categories to
  # make our analysis and graphs easier to understand later.
  mutate(victim_age = ifelse(victim_age == "UNKNOWN", "Unknown", victim_age),
         perp_age = ifelse(perp_age == "UNKNOWN", "Unknown", perp_age),
         victim_race = ifelse(victim_race == "WHITE", "White",
                       ifelse(victim_race == "WHITE HISPANIC", "White Hispanic",
                       ifelse(victim_race == "BLACK", "Black",
                       ifelse(victim_race == "BLACK HISPANIC", "Black Hispanic",
                       ifelse(victim_race == "AMERICAN INDIAN/ALASKAN NATIVE",
                              "Native",
                       ifelse(victim_race == "ASIAN / PACIFIC ISLANDER", "Asian",
                       ifelse(victim_race == "UNKNOWN", "Unknown", victim_race))))))),
         victim_race = as.factor(victim_race),
         victim_age = as.factor(victim_age),
         perp_age = as.factor(perp_age))
summary(age_pred_data)
```

```
##     perp_age            victim_race      victim_age
##  1      :1805   Asian          :  343   1      :2113
##  2      :6629   Black          :12661   2      :6650
##  3      :6342   Black Hispanic: 1924   3      :8359
##  4      : 775   Native         :    9   4      :1410
##  5      :  67   Unknown        :   53   5      : 171
##  Unknown:3148   White          :  574   Unknown:  63
##                 White Hispanic: 3202
```

Then we create a model and look at it's metrics.

```r
age_model <- multinom(perp_age ~ victim_race + victim_age, data = age_pred_data)
```

```
## # weights:  78 (60 variable)
## initial  value 33624.158200
## iter  10 value 25487.647172
## iter  20 value 25246.228056
## iter  30 value 25144.110961
## iter  40 value 25108.639116
## iter  50 value 25103.822166
## iter  60 value 25102.852900
## iter  70 value 25102.636625
## final  value 25102.635288
## converged
```

```
summary(age_model)
```

```
## Call:
## multinom(formula = perp_age ~ victim_race + victim_age, data = age_pred_data)
##
## Coefficients:
##         (Intercept) victim_raceBlack victim_raceBlack Hispanic
## 2         0.2905287        0.0964909               0.008333205
## 3        -0.9383021        0.2612409               0.117740322
## 4        -2.8772740       -0.4003104              -0.556964666
## 5       -29.0673209       14.1775431              14.503992057
## Unknown  -0.8030127        0.5779537               0.136049856
##         victim_raceNative victim_raceUnknown victim_raceWhite
## 2              -1.0539578          0.6358983        0.5348287
## 3               0.3601474          0.7506838        1.0950589
## 4             -12.8844488          0.6387714        1.1457989
## 5              -0.3898977         -0.8526977       16.9681526
## Unknown         1.4609173          0.9553728        0.9258967
##         victim_raceWhite Hispanic victim_age2 victim_age3 victim_age4
## 2                      0.08026179    1.088761    1.315890   0.9807670
## 3                      0.23605363    1.572630    2.796951   2.5057401
## 4                     -0.28394285    1.220447    3.120646   4.0182296
## 5                     14.54903771    8.848268   11.769211  13.1018425
## Unknown                0.30997608    1.026053    1.288761   0.8159409
##         victim_age5 victim_ageUnknown
## 2        0.35551277         1.3455853
## 3        1.45130428         3.3991212
## 4        2.80190563         3.5028644
## 5       13.63375953         0.7422478
## Unknown -0.03623767         0.1508755
##
## Std. Errors:
##         (Intercept) victim_raceBlack victim_raceBlack Hispanic
## 2         0.2082910        0.2031643                 0.2154904
## 3         0.2178681        0.2074958                 0.2205478
## 4         0.3404929        0.2664563                 0.2958109
## 5         0.1518531        0.1994248                 0.3238583
## Unknown   0.2496281        0.2431575                 0.2578799
##         victim_raceNative victim_raceUnknown victim_raceWhite
## 2            1.445450e+00       7.987391e-01        0.3018523
## 3            1.233464e+00       7.970015e-01        0.3015923
## 4            3.501955e-06       9.572029e-01        0.3570739
## 5            1.814768e-08       2.406895e-08        0.2753284
## Unknown      1.163362e+00       8.748562e-01        0.3419545
##         victim_raceWhite Hispanic victim_age2 victim_age3 victim_age4
## 2                       0.2104907  0.06948801  0.07486074   0.1305724
## 3                       0.2150642  0.08611105  0.08820827   0.1353416
## 4                       0.2800868  0.24475737  0.22787910   0.2518044
## 5                       0.2615520  0.57073367  0.23485296   0.2502133
## Unknown                 0.2514407  0.08017168  0.08513218   0.1491560
##         victim_age5 victim_ageUnknown
## 2         0.2593422      7.869157e-01
## 3         0.2620207      7.572603e-01
## 4         0.3857086      9.113905e-01
## 5         0.3303408      8.009328e-06
## Unknown   0.3329208      1.030815e+00
##
## Residual Deviance: 50205.27
## AIC: 50325.27
```

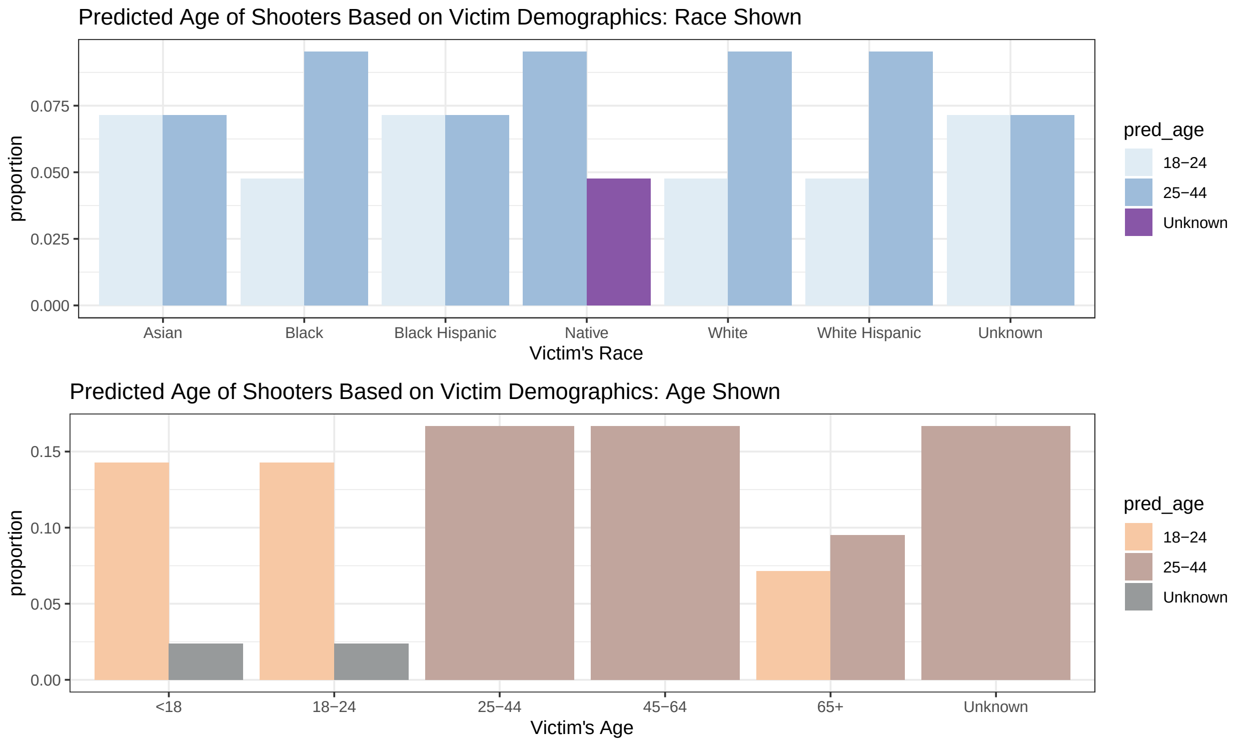
Then we visualize the fitted model for the predicted age of shooters based on some of the victim's demographics. Unfortunately I cannot find a way to display all variables in one plot, so I will create two plots and use `grid.arrange` to show them side-by-side.

```
# Create prediction data.
plot_data <- as_tibble(expand.grid(perp_age = c("1", "2", "3", "4", "5",
                                                "Unknown"),
                         victim_race = c("Asian", "Black", "Black Hispanic",
                                         "Native", "White", "White Hispanic",
                                         "Unknown"),
                         victim_age = c("1", "2", "3", "4", "5", "Unknown")))

pred_data <- plot_data |>
  bind_cols(predict(age_model, plot_data, type = "class"))
```

```
## New names:
## * ‘‘ -> ‘...4‘
```

```
# Reformatting for easier plotting.
pred_data <- pred_data |>
  rename(pred_age = "...4") |>
  mutate(victim_age = as.character(victim_age),    # Un-factor the variables
         pred_age = as.character(pred_age),
         victim_age = ifelse(victim_age == "1", "<18",
                      ifelse(victim_age == "2", "18-24",
                      ifelse(victim_age == "3", "25-44",
                      ifelse(victim_age == "4", "45-64",
                      ifelse(victim_age == "5", "65+", victim_age))))),
         pred_age = ifelse(pred_age == "1", "<18",
                      ifelse(pred_age == "2", "18-24",
                      ifelse(pred_age == "3", "25-44",
                      ifelse(pred_age == "4", "45-64",
                      ifelse(pred_age == "5", "65+", pred_age))))),
         victim_age = as.factor(victim_age),   # Re-factor the variables
         pred_age = as.factor(pred_age))

# Plot for predicted age of shooters including victim_race.
p_race <- pred_data |>
```

```
  ggplot(aes(victim_race, fill = pred_age)) +
  geom_bar(aes(y = after_stat(count)/sum(after_stat(count))), position="dodge") +
  scale_fill_brewer(palette = "BuPu") +
  theme_bw() +
  ggtitle("Predicted Age of Shooters Based on Victim Demographics: Race Shown") +
  xlab("Victim's Race") +
  ylab("proportion")

# Plot for predicted age of shooters including victim_age.
p_age <- pred_data |>
  ggplot(aes(victim_age, fill = pred_age)) +
  geom_bar(aes(y = after_stat(count)/sum(after_stat(count))), position="dodge") +
  scale_fill_manual(values = c("#F7C8A4FF", "#C1A59DFF", "#979A9BFF",
                               "#7795A2FF", "#497889FF")) +
  theme_bw() +
  ggtitle("Predicted Age of Shooters Based on Victim Demographics: Age Shown") +
  xlab("Victim's Age") +
  ylab("proportion")

grid.arrange(p_race, p_age)
```



The y-axis `proportion` represents the proportion of shootings out of the 300 sample data points used for prediction.

Well, our model predicts that, in any shooting, the shooter will be 18-44 years old, or of unknown age. According to our model, there is no other age that commits shootings. This is quite unrepresentative of our data, as shown here:

```
nypd_data |>
  select(perp_age) |>
  filter(is.na(perp_age) == FALSE) |>
  count(perp_age)
```

```
## # A tibble: 6 x 2
##   perp_age     n
##   <fct>    <int>
## 1 <18       1805
## 2 18-24     6630
## 3 25-44     6342
## 4 45-64      775
## 5 65+         67
## 6 UNKNOWN   3148
```

This could be indicative of several things, but I do not have the knowledge to dig deeper at this issue. The model will remain as is and we will move on to what conclusions we can draw, however tentative.

Conclusions based on age The model predicts that, if a victim is between ages 25 and 64, or of unknown age, then the shooter is definitely age 25 to 44. Another conclusion we can draw from the model is that the predicted age of shooters seems to loosely mirror that of the victims, forming a range that is centered on the shooter's age. Except in the case of shooters of unknown age.

Conclusions based on race The model predicts that, if a victim is Black, Native American, White or White Hispanic, then they are twice as likely to have been shot by someone ages 25-44 than any other age. The Asian, Black Hispanic, and unknown race victims are equally as likely to have been shot by someone ages 18-24 as someone ages 25-44. Interestingly only the Native American victims are likely to have been shot by someone of unknown age.

Bias

Bias is inevitable when working with data. The main source of bias present in this document is the missing data and how it was handled. In each case, the listwise deletion method was chosen. How does this bias our conclusions?

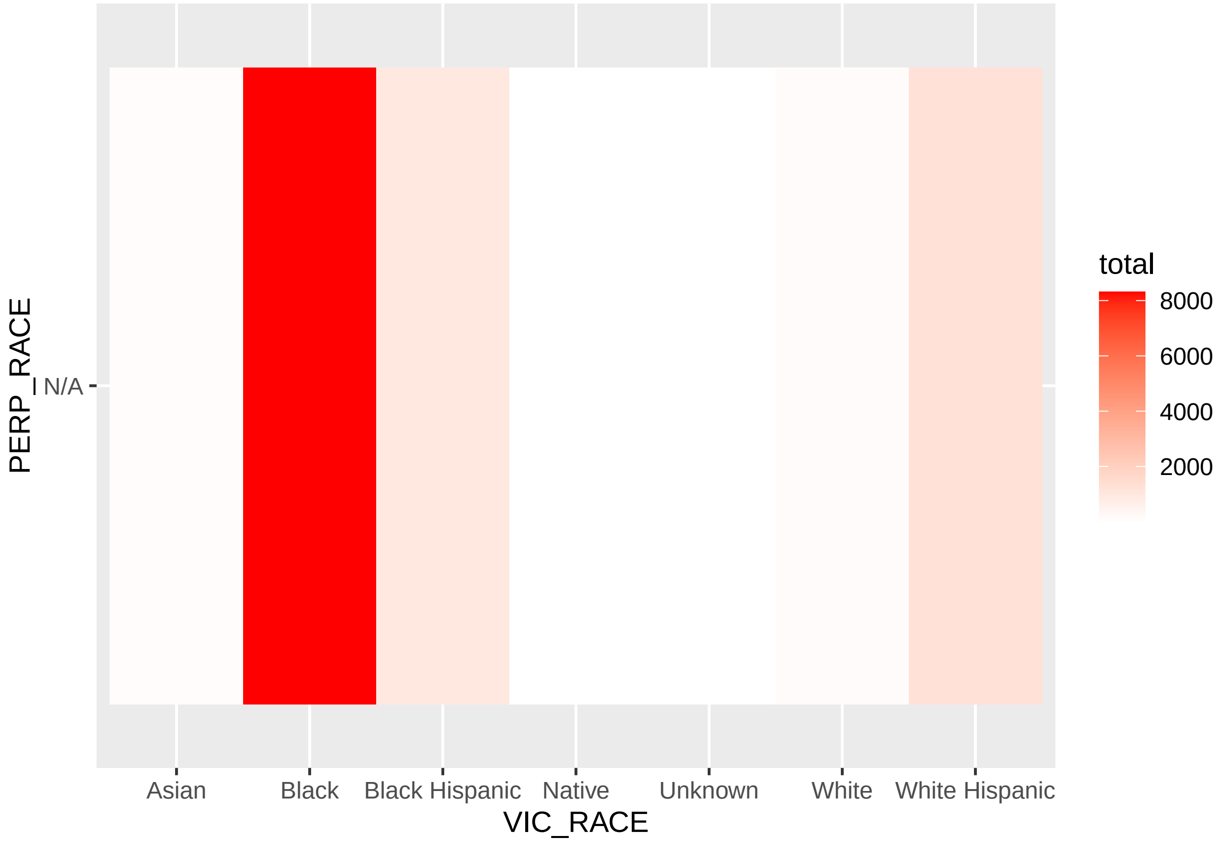
To determine how much bias is present, we will investigate the original dataset using a heatmap of the variables used for each plot. Note that the comparison between the fatality of a shooting and the victim's age only required a deletion of less than 0.3 of the dataset, thus barely influencing the conclusions that can be drawn.

Bias in the 'Race-on-Race shootings' figure Let's determine how much bias is present in the 'Race-on-Race shootings' figure. First we import the original dataset, and do some tidying. Then we visualize how our missing data is represented throughout the dataset.

```
bias_test_data <- read.csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNL

# Tidying
bias_test_tidy <- bias_test_data |>
  # This large `mutate` function is simply a relabeling of our categories to
  # make our graphs easier to understand later.
  mutate(VIC_RACE = ifelse(VIC_RACE == "WHITE", "White",
                       ifelse(VIC_RACE == "WHITE HISPANIC", "White Hispanic",
                       ifelse(VIC_RACE == "BLACK", "Black",
                       ifelse(VIC_RACE == "BLACK HISPANIC", "Black Hispanic",
                       ifelse(VIC_RACE == "AMERICAN INDIAN/ALASKAN NATIVE",
                              "Native",
                       ifelse(VIC_RACE == "ASIAN / PACIFIC ISLANDER", "Asian",
                       ifelse(VIC_RACE == "UNKNOWN", "Unknown", VIC_RACE))))))),
         PERP_RACE = ifelse(PERP_RACE == "WHITE", "White",
                       ifelse(PERP_RACE == "WHITE HISPANIC", "White Hispanic",
                       ifelse(PERP_RACE == "BLACK", "Black",
                       ifelse(PERP_RACE == "BLACK HISPANIC", "Black Hispanic",
                       ifelse(PERP_RACE == "AMERICAN INDIAN/ALASKAN NATIVE",
                              "Native",
                       ifelse(PERP_RACE == "ASIAN / PACIFIC ISLANDER", "Asian",
                       ifelse(PERP_RACE == "UNKNOWN", "Unknown", PERP_RACE))))))) |>
  # Unifying the missing data and labeling it "N/A"
  mutate(PERP_RACE = ifelse(PERP_RACE %in% c("", " ", "(null)"), NA, PERP_RACE),
         PERP_RACE = ifelse(is.na(PERP_RACE == TRUE), "N/A", PERP_RACE),

         # This unification is useful for the next bias investigation
         PERP_AGE_GROUP = ifelse(PERP_AGE_GROUP %in%
                                 c("", " ", "(null)", "1020", "1028", "2021",
                                   "224", "940"), NA, PERP_AGE_GROUP),
         PERP_AGE_GROUP = ifelse(is.na(PERP_AGE_GROUP)==TRUE, "N/A", PERP_AGE_GROUP),
         VIC_AGE_GROUP = ifelse(VIC_AGE_GROUP == "1022", NA, VIC_AGE_GROUP),
         VIC_AGE_GROUP = ifelse(is.na(VIC_AGE_GROUP)==TRUE, "N/A", VIC_AGE_GROUP),

         # 'Factoring' the data
         PERP_RACE = as.factor(PERP_RACE),
         VIC_RACE = as.factor(VIC_RACE),
         PERP_AGE_GROUP = as.factor(PERP_AGE_GROUP),
         VIC_AGE_GROUP = as.factor(VIC_AGE_GROUP))

# Reformatting
bias_test_race_on_race <- bias_test_tidy |>
  select(PERP_RACE, VIC_RACE) |>
  group_by(PERP_RACE, VIC_RACE) |>
  summarize(total = n()) |>
  ungroup() |>
  filter(PERP_RACE == "N/A")
```

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

```
# Plotting
bias_test_race_on_race |>
  ggplot(aes(x=VIC_RACE, y=PERP_RACE, fill=total)) +
  geom_tile() +
  scale_fill_gradient(low="white", high = "red")
```



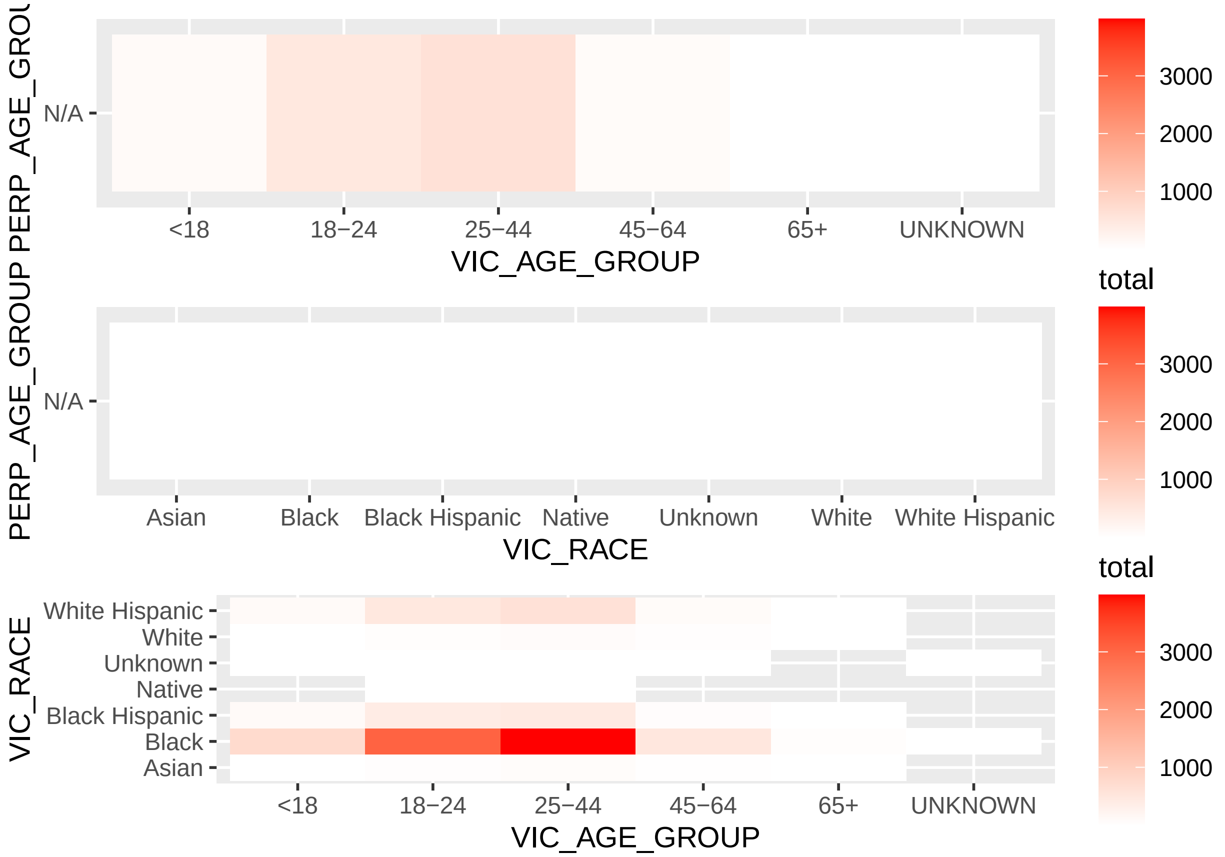
There is a severe amount of contrast in this heatmap. This indicates that the missing data is not evenly spread across the variables we looked at. This means the missing data does not satisfy the MCAR (missing completely at random) requirement that is often used when determining the validity of listwise deletion. This indicates that our data, or at least the two variables `PERP_RACE` and `VIC_RACE`, are biased to a strong degree. The conclusions may not be wholly representative of all shootings in NYC. Thus, the conclusions drawn can only be circumstantially useful.

Bias in Shooter's age Analysis Next we will look at the bias present in the 'Shooter's age analysis'. We will start by using the `bias_test_tidy` tibble created in the previous code chunk.

```
# Reformatting
bias_test_shooter_age <- bias_test_tidy |>
  select(PERP_AGE_GROUP, VIC_AGE_GROUP, VIC_RACE) |>
  group_by(PERP_AGE_GROUP, VIC_AGE_GROUP, VIC_RACE) |>
  summarize(total = n()) |>
  ungroup() |>
  filter(PERP_AGE_GROUP == "N/A")
```

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

```
# Plotting using grid.arrange
p_bias1 <- bias_test_shooter_age |>
  ggplot(aes(x=VIC_AGE_GROUP, y=PERP_AGE_GROUP, fill=total)) +
  geom_tile() +
  scale_fill_gradient(low="white", high="red")

p_bias2 <- bias_test_shooter_age |>
  ggplot(aes(x=VIC_RACE, y=PERP_AGE_GROUP, fill=total)) +
  geom_tile() +
  scale_fill_gradient(low="white", high="red")

p_bias3 <- bias_test_shooter_age |>
  ggplot(aes(x=VIC_AGE_GROUP, y=VIC_RACE, fill=total)) +
  geom_tile() +
  scale_fill_gradient(low="white", high="red")

grid.arrange(p_bias1, p_bias2, p_bias3)
```



There is almost zero contrast in the heatmap between `VIC_RACE` and `PERP_AGE_GROUP`, and there is little contrast in the heatmap between `VIC_AGE_GROUP` and `PERP_AGE_GROUP`. However, there is significant contrast on the heatmap between the two predictive variables, `VIC_AGE_GROUP` and `VIC_RACE`. This indicates that the model we created is missing "pieces to the puzzle"; there is significant bias generated by the listwise deletion of missing data when our model was formed. Modelling only enhances any bias present in the data, so it is reasonable to conclude that our model is not capturing the whole picture. This may be why our model predicted that shooters are only between the ages of 18-44 or of unknown age when the dataset as a whole did not suggest that.

Conclusion

This has been a light investigation and analysis of the “NYPD Shooting Incident Data” dataset. While it is disappointing that no strong conclusions were able to be drawn as a result of strong bias present in the data, this document serves a purpose for learning what can cause bias and how it can manifest in the modeling process. Due to time constraints, I went through the data science analysis process first and only considered potential bias after all conclusions were drawn. Next time, bias mitigation strategies will be in the forefront of my mind as I start the data science process.

```
## function (package = NULL)
## {
##     z <- list()
##     z$R.version <- R.Version()
##     z$platform <- z$R.version$platform
##     if (nzchar(.Platform$r_arch))
##         z$platform <- paste(z$platform, .Platform$r_arch, sep = "/")
##     sp <- 8 * .Machine$sizeof.pointer
##     if (sp != 64)
##         z$platform <- paste0(z$platform, " (", sp, "-bit)")
##     z$locale <- Sys.getlocale()
##     z$tzone <- Sys.timezone()
##     z$tzcode_type <- .Call(C_tzcode_type)
##     z$running <- osVersion
##     z$RNGkind <- RNGkind()
##     if (is.null(package)) {
##         package <- grep("^package:", search(), value = TRUE)
##         keep <- vapply(package, function(x) x == "package:base" ||
##             !is.null(attr(as.environment(x), "path")), NA)
##         package <- .rmpkg(package[keep])
##     }
##     pkgDesc <- lapply(package, packageDescription, encoding = NA)
##     if (length(package) == 0)
##         stop("no valid packages were specified")
##     basePkgs <- sapply(pkgDesc, function(x) !is.null(x$Priority) &&
##         x$Priority == "base")
##     z$basePkgs <- package[basePkgs]
##     if (any(!basePkgs)) {
##         z$otherPkgs <- pkgDesc[!basePkgs]
##         names(z$otherPkgs) <- package[!basePkgs]
##     }
##     loadedOnly <- loadedNamespaces()
##     loadedOnly <- loadedOnly[!(loadedOnly %in% package)]
##     if (length(loadedOnly)) {
##         names(loadedOnly) <- loadedOnly
##         pkgDesc <- c(pkgDesc, lapply(loadedOnly, packageDescription,
##             encoding = NA))
##         z$loadedOnly <- pkgDesc[loadedOnly]
##     }
##     z$matprod <- as.character(options("matprod"))
##     es <- extSoftVersion()
##     z$BLAS <- es[["BLAS"]]
##     z$LAPACK <- La_library()
##     z$LA_version <- La_version()
##     l10n <- l10n_info()
##     if (!is.null(l10n[["system.codepage"]]))
##         z$system.codepage <- l10n[["system.codepage"]]
##     if (!is.null(l10n[["codepage"]]))
##         z$codepage <- l10n[["codepage"]]
##     class(z) <- "sessionInfo"
##     z
## }
## <bytecode: 0x000002208de645b0>
## <environment: namespace:utils>
```