NYPD Shooting Project

Ethan Tucker

2/28/2022

Question of Interest and Data Source

The data in this project is found here. The data is summarized in the next section (and in the conclusion) that details my tidying efforts. The data contains - among many other things - information regarding the victims of shooting incidents in New York City. I wanted to know how the victim statistics - age, race, sex - changed over time. Thanks so much for taking the time to read my project!

```
getwd()
```

```
## [1] "C:/Users/first/Desktop/NYPD_Shooting_Project"
```

```
untidyData <- read_csv("./Data/NYPD_Shooting_Incident_Data__Historic_.csv")
```

```
## Rows: 23585 Columns: 19
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): OCCUR_DATE, BORO, LOCATION_DESC, PERP_AGE_GROUP, PERP_SEX, PERP_R...
## 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.
```

```
## Remove unnecessary and potentially biased variables
shootingData <- data.frame(untidyData) %>%
                  select(-c(LOCATION_DESC, Lon_Lat, PERP_AGE_GROUP, PERP_SEX, PERP_RACE))

## Change categorical variables to factors
shootingData$INCIDENT_KEY <- parse_factor(as.character(untidyData$INCIDENT_KEY))
shootingData$BORO <- parse_factor(as.character(untidyData$INCIDENT_KEY))
shootingData$PRECINCT <- parse_factor(as.character(untidyData$PRECINCT))
shootingData$JURISDICTION_CODE <- parse_factor(as.character(untidyData$JURISDICTION_CODE))
shootingData$VIC_AGE_GROUP <- parse_factor(untidyData$VIC_AGE_GROUP)
shootingData$VIC_SEX <- as.factor(untidyData$VIC_SEX)
shootingData$VIC_RACE <- as.factor(untidyData$VIC_RACE)


## Change dates and times from character to date
shootingData$OCCUR_DATE <- parse_date(untidyData$OCCUR_DATE, format = "%m/%d/%Y")
shootingData$OCCUR_TIME <- parse_time(as.character(untidyData$OCCUR_TIME))

## Overview of the data
```

```
shootingData %>% glimpse()
```

```
## Rows: 23,585
## Columns: 14
## $ INCIDENT_KEY            <fct> 24050482, 77673979, 203350417, 80584527, 90843~
## $ OCCUR_DATE              <date> 2006-08-27, 2011-03-11, 2019-10-06, 2011-09-0~
## $ OCCUR_TIME              <time> 05:35:00, 12:03:00, 01:09:00, 03:35:00, 21:16~
## $ BORO                    <fct> 24050482, 77673979, 203350417, 80584527, 90843~
## $ PRECINCT                <fct> 52, 106, 77, 40, 100, 67, 77, 81, 101, 106, 71~
## $ JURISDICTION_CODE       <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ STATISTICAL_MURDER_FLAG <lgl> TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE~
## $ VIC_AGE_GROUP           <fct> 25-44, 65+, 18-24, <18, 18-24, <18, <18, 25-44~
## $ VIC_SEX                 <fct> F, M, F, M, M, M, M, M, M, M, F, M, M, M, M, M~
## $ VIC_RACE                <fct> BLACK HISPANIC, WHITE, BLACK, BLACK, BLACK, BL~
## $ X_COORD_CD              <dbl> 1017542, 1027543, 995325, 1007453, 1041267, 10~
## $ Y_COORD_CD              <dbl> 255918.9, 186095.0, 185155.0, 233952.0, 157133~
## $ Latitude                <dbl> 40.86906, 40.67737, 40.67489, 40.80880, 40.597~
## $ Longitude               <dbl> -73.87963, -73.84392, -73.96008, -73.91618, -7~
```

Summary of Dataset and Tidying:

The NYPD Shooting Incident (Historic) dataset contains `nrow(untidyData)` records on `ncol(untidyData)` variables. In my cleaning I removed the Lon_Lat variable which contained duplicate information (both longitude and latitude are variables by themselves), and the LOCATION_DESC variable which contained the officer's description of the location of the shooting. This is interesting information, but the descriptions are not consistent in nomenclature and are mostly missing. I have also chosen to remove the perpetrator variables (sex, race, age group), because these variables may contain bias from both the police filing the reports and witnesses who conveyed information to the police. My expectation is that minorities and men are overrepresented against reality in this dataset in the perpetrator variables. These variables are also awash with missing values, so removing them kill two birds with one stone. The resulting dataset which I have called shootingData contains the same number of observations on `ncol(shootingData)` variables. I have included a glimpse() of shootingData in the output of the cell above. The below chunk confirms that there are no remaining missing values in the dataset.

```
countMissing <- function(column){

        n.missing <- length(which(is.na(column)))
        return(n.missing)
}

toPrint <- rep("", ncol(shootingData))

for(i in 1:ncol(shootingData)){
  toPrint[i] <- paste("The number of missing values in the variable", names(shootingData)[i], "was:", co
}

toPrint
```

```
##  [1] "The number of missing values in the variable INCIDENT_KEY was: 0 ."
##  [2] "The number of missing values in the variable OCCUR_DATE was: 0 ."
##  [3] "The number of missing values in the variable OCCUR_TIME was: 0 ."
##  [4] "The number of missing values in the variable BORO was: 0 ."
##  [5] "The number of missing values in the variable PRECINCT was: 0 ."
##  [6] "The number of missing values in the variable JURISDICTION_CODE was: 0 ."
##  [7] "The number of missing values in the variable STATISTICAL_MURDER_FLAG was: 0 ."
```

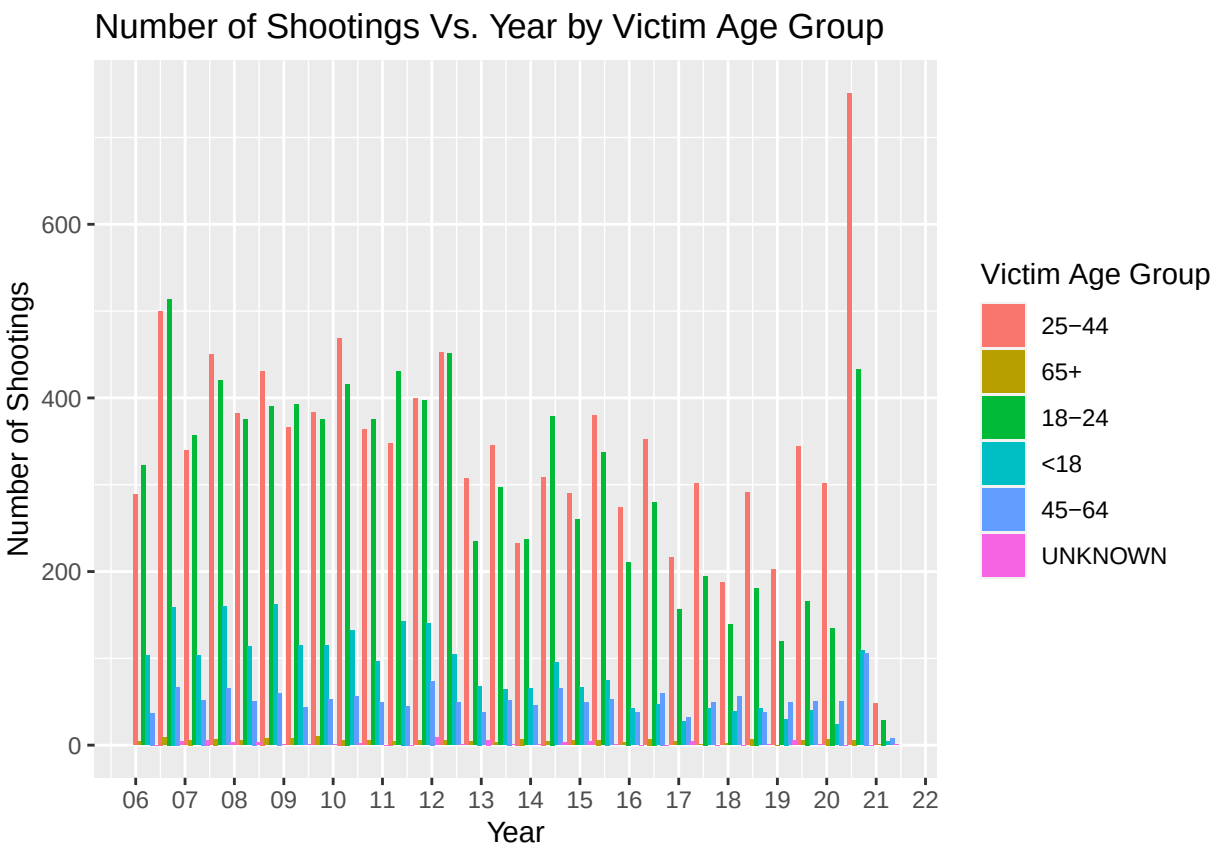
```
##  [8] "The number of missing values in the variable VIC_AGE_GROUP was: 0 ."
##  [9] "The number of missing values in the variable VIC_SEX was: 0 ."
## [10] "The number of missing values in the variable VIC_RACE was: 0 ."
## [11] "The number of missing values in the variable X_COORD_CD was: 0 ."
## [12] "The number of missing values in the variable Y_COORD_CD was: 0 ."
## [13] "The number of missing values in the variable Latitude was: 0 ."
## [14] "The number of missing values in the variable Longitude was: 0 ."
```

Visualizations

I chose to make bar-charts of victim data, because victim data seems quite bias-free. I have plotted date on the x axis (binned by year) and shooting count on the y axis. The data are grouped into different colors by age group, sex, and race depending on the graph. An interesting comparison would be to plot this data against the national statistics to see how New York shooting violence has historically compared to the national averages.

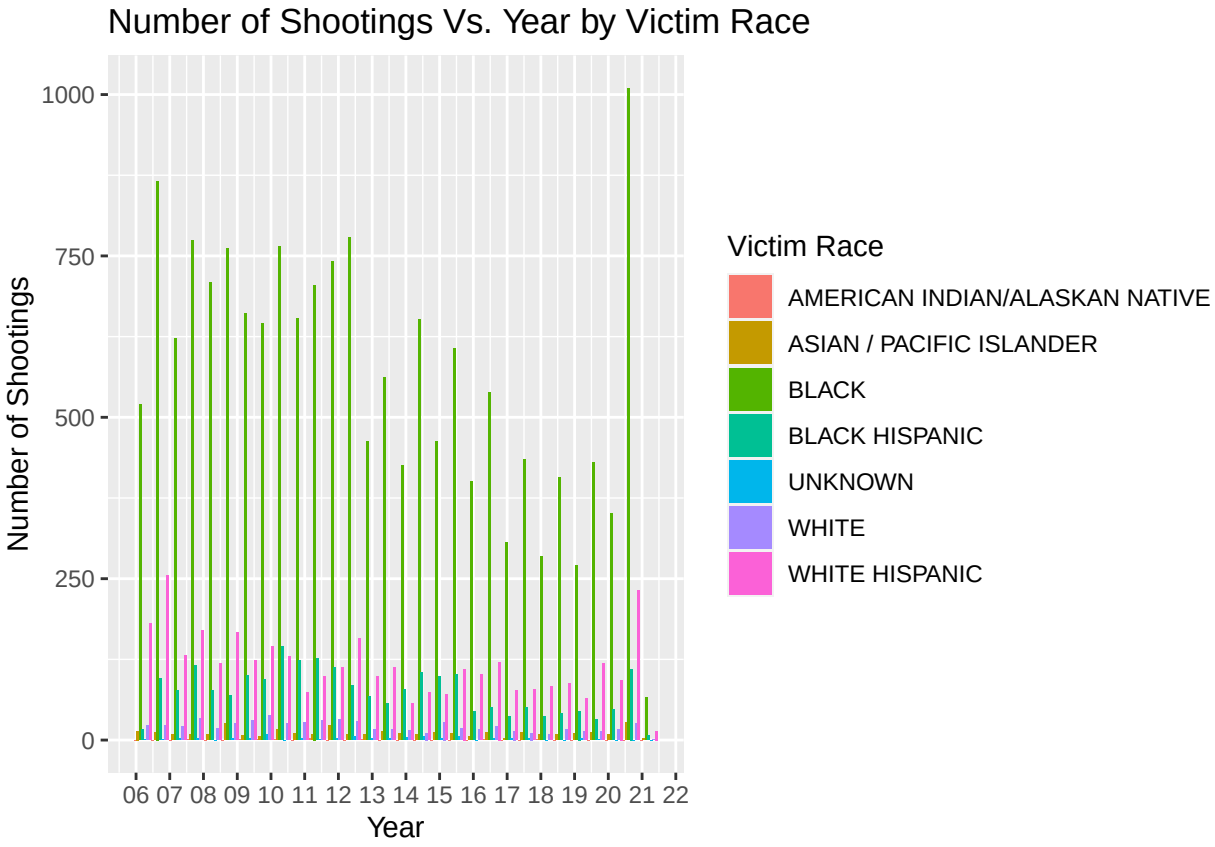
```
## Number of Shootings Vs. Year by Victim Age Group
shootingData %>%
  ggplot(aes(x = OCCUR_DATE, fill = VIC_AGE_GROUP)) +
  geom_histogram(position = "dodge") +
  scale_x_date(labels = date_format("%y"), breaks= date_breaks(width = "1 year")) +
  labs(x = "Year",
       y = "Number of Shootings",
       fill = "Victim Age Group",
       title = "Number of Shootings Vs. Year by Victim Age Group")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



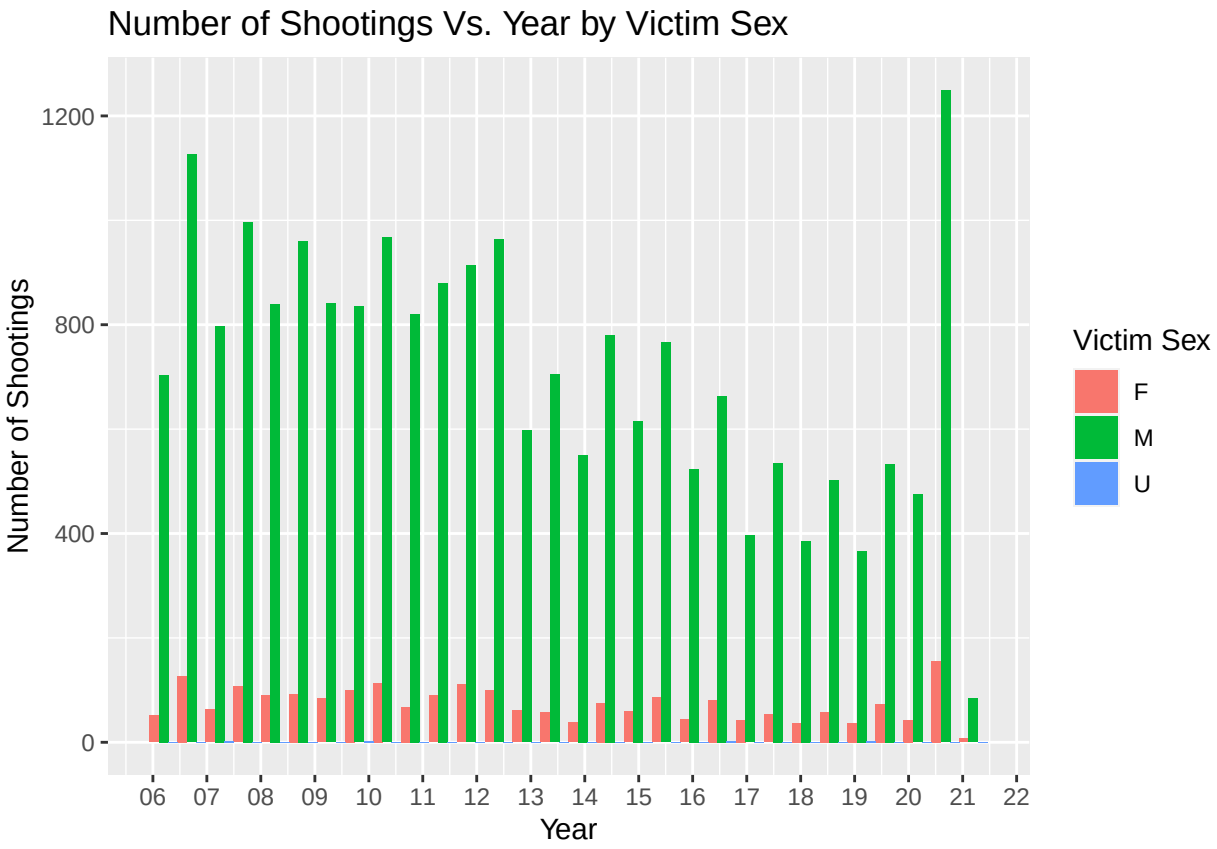
```
## Number of Shootings Vs. Year by Victim Race
shootingData %>%
  ggplot(aes(x = OCCUR_DATE, fill = VIC_RACE)) +
  geom_histogram(position = "dodge") +
  scale_x_date(labels = date_format("%y"), breaks= date_breaks(width = "1 year")) +
  labs(x = "Year",
       y = "Number of Shootings",
       fill = "Victim Race",
       title = "Number of Shootings Vs. Year by Victim Race")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
## Number of Shootings Vs. Year by Victim Race
shootingData %>%
  ggplot(aes(x = OCCUR_DATE, fill = VIC_SEX)) +
  geom_histogram(position = "dodge") +
  scale_x_date(labels = date_format("%y"), breaks= date_breaks(width = "1 year")) +
  labs(x = "Year",
       y = "Number of Shootings",
       fill = "Victim Sex",
       title = "Number of Shootings Vs. Year by Victim Sex")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Analysis

I did some data manipulation to effectively re-create the above visualizations as tables. The one change I made was that instead of reporting the total shooting count for each analysis, I decided to report the proportion of shootings that year per category. This would also make an interesting plot!

```
age_table <- shootingData %>%
                group_by(year = lubridate::floor_date(OCCUR_DATE, "year")) %>%
                  mutate(count = n()) %>%
                    group_by(year, VIC_AGE_GROUP) %>%
                      summarise(year = year, proportion = n()/count) %>%
                        arrange(-desc(year)) %>%
                          distinct() %>%
                            ungroup()
```

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

```
race_table <- shootingData %>%
                group_by(year = lubridate::floor_date(OCCUR_DATE, "year")) %>%
                  mutate(count = n()) %>%
                    group_by(year, VIC_RACE) %>%
                      summarise(year = year, proportion = n()/count) %>%
                        arrange(-desc(year)) %>%
                          distinct() %>%
                            ungroup()
```

```
## `summarise()` has grouped output by 'year', 'VIC_RACE'. You can override using the `.groups` argumen
```

```
sex_table <- shootingData %>%
                group_by(year = lubridate::floor_date(OCCUR_DATE, "year")) %>%
                  mutate(count = n()) %>%
                    group_by(year, VIC_SEX) %>%
                      summarise(year = year, proportion = n()/count) %>%
                        arrange(-desc(year)) %>%
                          distinct() %>%
                            ungroup()
```

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

```
age_table ; race_table ; sex_table
```

```
## # A tibble: 89 x 3
##    year       VIC_AGE_GROUP proportion
##    <date>     <fct>             <dbl>
##  1 2006-01-01 25-44           0.396
##  2 2006-01-01 65+             0.00633
##  3 2006-01-01 18-24           0.413
##  4 2006-01-01 <18             0.128
##  5 2006-01-01 45-64           0.0540
##  6 2006-01-01 UNKNOWN         0.00243
##  7 2007-01-01 25-44           0.398
##  8 2007-01-01 65+             0.00636
##  9 2007-01-01 18-24           0.397
## 10 2007-01-01 <18             0.138
## # ... with 79 more rows
```

```
## # A tibble: 96 x 3
##    year       VIC_RACE                         proportion
##    <date>     <fct>                                 <dbl>
##  1 2006-01-01 ASIAN / PACIFIC ISLANDER            0.0127
##  2 2006-01-01 BLACK                               0.692
##  3 2006-01-01 BLACK HISPANIC                      0.0555
##  4 2006-01-01 UNKNOWN                             0.000973
##  5 2006-01-01 WHITE                               0.0224
##  6 2006-01-01 WHITE HISPANIC                      0.217
##  7 2007-01-01 AMERICAN INDIAN/ALASKAN NATIVE      0.000530
##  8 2007-01-01 ASIAN / PACIFIC ISLANDER            0.00954
##  9 2007-01-01 BLACK                               0.709
## 10 2007-01-01 BLACK HISPANIC                      0.0991
## # ... with 86 more rows
```

```
## # A tibble: 35 x 3
##    year       VIC_SEX proportion
##    <date>     <fct>        <dbl>
##  1 2006-01-01 F          0.0886
##  2 2006-01-01 M          0.911
##  3 2007-01-01 F          0.0859
##  4 2007-01-01 M          0.912
##  5 2007-01-01 U          0.00212
##  6 2008-01-01 F          0.0929
##  7 2008-01-01 M          0.907
##  8 2009-01-01 F          0.0990
##  9 2009-01-01 M          0.900
```

```
## 10 2009-01-01 U          0.00109
## # ... with 25 more rows
```

Model

While a linear model is certainly not appropriate for this shooting data considering the massive spike during COVID, it can give us some insight into what might have been the shooting statistics had COVID not happened. I will do a model that predicts the number of shootings by year for each age group in age_table.

```
## Model for ages less than 18
ages.lessthan18 <- age_table %>%
                    filter(VIC_AGE_GROUP == unique(age_table$VIC_AGE_GROUP)[4])
lmod.lessthan18 <- lm(proportion ~ year, data = ages.lessthan18)
summary(lmod.lessthan18)
```

```
##
## Call:
## lm(formula = proportion ~ year, data = ages.lessthan18)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.023589 -0.002769 -0.001385  0.004304  0.027378
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.477e-01  3.054e-02  11.387 3.91e-08 ***
## year        -1.565e-05  1.934e-06  -8.088 1.98e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01182 on 13 degrees of freedom
## Multiple R-squared:  0.8342, Adjusted R-squared:  0.8215
## F-statistic: 65.41 on 1 and 13 DF,  p-value: 1.984e-06
```

```
## Model for ages 18 - 24
ages.18to24 <- age_table %>%
                filter(VIC_AGE_GROUP == unique(age_table$VIC_AGE_GROUP)[3])
lmod.18to24 <- lm(proportion ~ year, data = ages.18to24)
summary(lmod.18to24)
```

```
##
## Call:
## lm(formula = proportion ~ year, data = ages.18to24)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.045937 -0.021358  0.002677  0.022123  0.046943
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.237e-01  7.148e-02  10.125 1.56e-07 ***
## year        -2.216e-05  4.528e-06  -4.893 0.000294 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.02767 on 13 degrees of freedom
## Multiple R-squared:  0.6481, Adjusted R-squared:  0.621
## F-statistic: 23.94 on 1 and 13 DF,  p-value: 0.0002939
```

```
## Model for ages 25 - 44
ages.25to44 <- age_table %>%
               filter(VIC_AGE_GROUP == unique(age_table$VIC_AGE_GROUP)[1])
lmod.25to44 <- lm(proportion ~ year, data = ages.25to44)
summary(lmod.25to44)
```

```
##
## Call:
## lm(formula = proportion ~ year, data = ages.25to44)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.061791 -0.006449  0.002039  0.018410  0.029458
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.477e-02  7.021e-02  -0.210    0.837
## year         2.925e-05  4.448e-06   6.575 1.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02719 on 13 degrees of freedom
## Multiple R-squared:  0.7688, Adjusted R-squared:  0.751
## F-statistic: 43.23 on 1 and 13 DF,  p-value: 1.784e-05
```

```
## Model for ages 45 - 64
ages.45to64 <- age_table %>%
               filter(VIC_AGE_GROUP == unique(age_table$VIC_AGE_GROUP)[5])
lmod.45to64 <- lm(proportion ~ year, data = ages.45to64)
summary(lmod.45to64)
```

```
##
## Call:
## lm(formula = proportion ~ year, data = ages.45to64)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0188154 -0.0056872  0.0002752  0.0065819  0.0128451
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.692e-02  2.378e-02  -2.815   0.0146 *
## year         8.626e-06  1.506e-06   5.727 6.97e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009206 on 13 degrees of freedom
## Multiple R-squared:  0.7161, Adjusted R-squared:  0.6943
## F-statistic:  32.8 on 1 and 13 DF,  p-value: 6.972e-05
```

```
## Model for ages 65 +
ages.greaterthan65 <- age_table %>%
```

```
                filter(VIC_AGE_GROUP == unique(age_table$VIC_AGE_GROUP)[2])
lmod.greaterthan65 <- lm(proportion ~ year, data = ages.greaterthan65)
summary(lmod.greaterthan65)
```

```
##
## Call:
## lm(formula = proportion ~ year, data = ages.greaterthan65)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0024437 -0.0004221  0.0001165  0.0005790  0.0027656
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.845e-03  3.532e-03   1.655    0.122
## year        4.840e-08  2.238e-07   0.216    0.832
##
## Residual standard error: 0.001368 on 13 degrees of freedom
## Multiple R-squared:  0.003586,   Adjusted R-squared:  -0.07306
## F-statistic: 0.04679 on 1 and 13 DF,  p-value: 0.8321
```

Comments on models

The models show that on average the ages of shooting victims have been increasing. That's probably a good thing because fewer kids are getting shot. The model for shooting victims over the age of 65 is not statistically significant.

Conclusion and Bias Identification:

In this project I imported a dataset containing NYPD records of shooting incidents between `min(shootingData$OCCUR_DATE)` and `max(shootingData$OCCUR_DATE)`. I cleaned the data by adjusting the variable classes, and by removing some oronious and potentially biased variables. I created three visualizations and three analysis tables summarizing shooting incidents vs. year according to a couple different victim groupings. I ran some models predicting the proportion of shootings by age group by year, and found that on average the age of shooting victims was increasing between 2006 and 2020. An important caviat is that the linear models do not accurately predict the shooting statistics from 2020 during the COVID pandemic.

The data are inputted by the New York Police Department, which notoriously had "stop and frisk" as their modus operandi for several years while Rudy Guiliani was mayor. That time period falls within this dataset. As such it is likely that minorities are inaccurately reported, especially in the perpetrator variables. To help circumvent this problem, the analyses, visualizations, and models I made only used the victim columns which (to me) seem less likely to have racial / sexist biases. The reason for this belief is because victims tend to go to hospitals at which the police take their statements, instead of having their race / sex misconstrued by witnesses and police. That said, another potential source of bias relates to police potentially not responding to shooting calls in minority neighborhoods, thus deflating those shooting statistics. This seems to be minor if extant, because shootings are always a high priority for police and is responsible for large swathes of NYPD funding (gear, etc.). My personal bias lies in that I am overly suspiscious of any data that includes race but not social class. I particularly don't believe police reports, and think that some of these shootings were perpetrated by police rather than "Unknown" perpetrators. Perhaps that's wrong. Regardless, that doesn't impact the validity of the victim data.

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
```

```
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] scales_1.1.1    forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7
##  [5] purrr_0.3.4     readr_2.1.1     tidyr_1.1.4     tibble_3.1.6
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.24         haven_2.4.3       colorspace_2.0-2
##  [5] vctrs_0.3.8       generics_0.1.1    htmltools_0.5.1.1 yaml_2.2.1
##  [9] utf8_1.2.2        rlang_0.4.11      pillar_1.6.4      glue_1.4.2
## [13] withr_2.4.3       DBI_1.1.1         bit64_4.0.5       dbplyr_2.1.1
## [17] modelr_0.1.8      readxl_1.3.1      lifecycle_1.0.1   munsell_0.5.0
## [21] gtable_0.3.0      cellranger_1.1.0  rvest_1.0.2       evaluate_0.14
## [25] labeling_0.4.2    knitr_1.36        tzdb_0.2.0        parallel_4.1.2
## [29] fansi_0.5.0       highr_0.9         broom_0.7.10      Rcpp_1.0.7
## [33] backports_1.3.0   vroom_1.5.7       jsonlite_1.7.2    farver_2.1.0
## [37] bit_4.0.4         fs_1.5.0          hms_1.1.1         digest_0.6.27
## [41] stringi_1.7.3     grid_4.1.2        cli_3.1.0         tools_4.1.2
## [45] magrittr_2.0.1    crayon_1.4.2      pkgconfig_2.0.3   ellipsis_0.3.2
## [49] xml2_1.3.3        reprex_2.0.1      lubridate_1.8.0   assertthat_0.2.1
## [53] rmarkdown_2.11    httr_1.4.2        rstudioapi_0.13   R6_2.5.1
## [57] compiler_4.1.2
```