NYPD Shooting Incident Data - Data Science Project

Instructions for the NYPD Shooting Incidents Project:

The instructions for this data project, as posted in Coursera were as follows. The sections of this document follow this order and topic for simplicity.

1. Step 1 - Start an Rmd Document:
   Start an Rmd document that describes and imports the shooting project dataset in a reproducible manner.

2. Project Step 2 - Tidy and Transform Your Data:
   Add to your Rmd document a summary of the data and clean up your dataset by changing appropriate variables to factor and date types and getting rid of any columns not needed. Show the summary of your data to be sure there is no missing data. If there is missing data, describe how you plan to handle it.

3. Project Step 3 - Add Visualizations and Analysis:
   Add at least two different visualizations & some analysis to your Rmd. Does this raise additional questions that you should investigate?

4. Project Step 4 - Add Bias Identification:
   Write the conclusion to your project report and include any possible sources of bias. Be sure to identify what your personal bias might be and how you have mitigated that.

Step 1 - Importing and Describing Data Set:

To begin this project, I first loaded the tidyverse and lubridate packages. These packages include functions for data wrangling and simplified data/time coding respectively.

```
library( tidyverse )
library( lubridate )
```

I next loaded the data from the csv file, which I download from 'data.gov' in the code below. I then used the **glimpse()** and **summary()** functions to get a first look at the structure of this data and brief characteristics of each column.

```
# Read the NYPD csv file from data.gov:
data_url <- paste( "https://data.cityofnewyork.us/api/views/833y-fsy8/",
"rows.csv?accessType=DOWNLOAD", sep = "" )
raw_data <- read_csv( data_url )

glimpse(raw_data) # See the columns (and types) of the data.
```

```
## Rows: 28,562
## Columns: 21
## $ INCIDENT_KEY            <dbl> 244608249, 247542571, 84967535, 202853370, 270~
## $ OCCUR_DATE              <chr> "05/05/2022", "07/04/2022", "05/27/2012", "09/~
## $ OCCUR_TIME              <time> 00:10:00, 22:20:00, 19:35:00, 21:00:00, 21:00~
## $ BORO                    <chr> "MANHATTAN", "BRONX", "QUEENS", "BRONX", "BROO~
## $ LOC_OF_OCCUR_DESC       <chr> "INSIDE", "OUTSIDE", NA, NA, NA, NA, NA, NA, N~
## $ PRECINCT                <dbl> 14, 48, 103, 42, 83, 23, 113, 77, 48, 49, 73, ~
## $ JURISDICTION_CODE       <dbl> 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ LOC_CLASSFCTN_DESC      <chr> "COMMERCIAL", "STREET", NA, NA, NA, NA, NA, NA~
## $ LOCATION_DESC           <chr> "VIDEO STORE", "(null)", NA, NA, NA, "MULTI DW~
```

```
## $ STATISTICAL_MURDER_FLAG <lgl> TRUE, TRUE, FALSE, FALSE, FALSE, FALSE, TRUE, ~
## $ PERP_AGE_GROUP          <chr> "25-44", "(null)", NA, "25-44", "25-44", NA, N~
## $ PERP_SEX                <chr> "M", "(null)", NA, "M", "M", NA, NA, NA, NA, "~
## $ PERP_RACE               <chr> "BLACK", "(null)", NA, "UNKNOWN", "BLACK", NA,~
## $ VIC_AGE_GROUP           <chr> "25-44", "18-24", "18-24", "25-44", "25-44", "~
## $ VIC_SEX                 <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
## $ VIC_RACE                <chr> "BLACK", "BLACK", "BLACK", "BLACK", "BLACK", "~
## $ X_COORD_CD              <dbl> 986050, 1016802, 1048632, 1014493, 1009149, 99~
## $ Y_COORD_CD              <dbl> 214231.0, 250581.0, 198262.0, 242565.0, 190104~
## $ Latitude                <dbl> 40.75469, 40.85440, 40.71063, 40.83242, 40.688~
## $ Longitude               <dbl> -73.99350, -73.88233, -73.76777, -73.89071, -7~
## $ Lon_Lat                 <chr> "POINT (-73.9935 40.754692)", "POINT (-73.8823~
```

```r
summary(raw_data) # Characteristics of the columns
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:28562       Length:28562      Length:28562
##  1st Qu.: 65439914   Class :character   Class1:hms        Class :character
##  Median : 92711254   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :127405824                      Mode  :numeric
##  3rd Qu.:203131993
##  Max.   :279758069
##
##  LOC_OF_OCCUR_DESC     PRECINCT     JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:28562       Min.   :  1.0   Min.   :0.0000    Length:28562
##  Class :character   1st Qu.: 44.0   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 67.0   Median :0.0000    Mode  :character
##                     Mean   : 65.5   Mean   :0.3219
##                     3rd Qu.: 81.0   3rd Qu.:0.0000
##                     Max.   :123.0   Max.   :2.0000
##                                     NA's   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:28562       Mode :logical           Length:28562
##  Class :character   FALSE:23036             Class :character
##  Mode  :character   TRUE :5526              Mode  :character
##
##
##
##
##    PERP_SEX          PERP_RACE          VIC_AGE_GROUP        VIC_SEX
##  Length:28562       Length:28562       Length:28562       Length:28562
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude
##  Length:28562       Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character   1st Qu.:1000068   1st Qu.:182912   1st Qu.:40.67
##  Mode  :character   Median :1007772   Median :194901   Median :40.70
##                     Mean   :1009424   Mean   :208380   Mean   :40.74
##                     3rd Qu.:1016807   3rd Qu.:239814   3rd Qu.:40.82
##                     Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                        NA's   :59
##    Longitude        Lon_Lat
##  Min.   :-74.25   Length:28562
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
```

```
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA's   :59
```

Step 2 - Tidying and Transforming Data:

I tidied and transformed the data in the following steps:

- Consolidation the date and time information from two columns into a single column that was formatted using the *lubridate* coding.
- Removing the 'Log_Lat' Column which contains information that is duplicated in in the 'Longitude' and 'Latitude' columns.
- Casting the 'PERP_SEX' and 'VIC_SEX' columns as factors after recoding "(null)" entries as NA.
- Casting the 'PERP_RACE' and 'VIC_RACE' columns as factors after again recoding "(null)" entries as NA.
- Casting the 'PERP_AGE_GROUP' and 'VIC_AGE_GROUP' columns as factors after recoding a variety of anomalous entries as NA.
- Casting the 'BORO' and 'JURISDICTION_CODE' columns as factors.
- The character columns of 'LOC_OF_OCCUR_DESC', 'LOC_CLASSFCTN_DESC', and 'LOCATION_DESC' were left in their original forms. While not used in analysis in this document, this could be useful for later analysis. These columns have many missing values or NA entries. Due to incompleteness of these fields, I plan to use the information as an extension of another analysis as opposed to trying to plot or group_by/summarize any of these columns.
- The 'X_COORD_CD' and 'Y_COORD_CD' columns were deleted, as I can already use the latitude and longitude information for plotting.

```r
#~~~# Create a copy of raw_data to start tidying/transforming:
clean_data <- raw_data

#~~~# Code the date and time in the same column using mutate and the lubridate
#date/time coding.
clean_data <- mutate( .data = raw_data,
    occour_date_time = mdy_hms( paste(OCCUR_DATE, OCCUR_TIME) ) )

# Remove original date and time columns from the clean data.
clean_data <- select( .data = clean_data, c(-OCCUR_DATE, -OCCUR_TIME) )
```

```r
#~~~# Lon_lat looks like it contains the same info as Latitude and Longitude columns,
# so there is duplicate information here that can be removed. Determining if there
# are the same NAs in both the individual and combined columns.
sum( is.na(clean_data$Latitude) )
sum( is.na(clean_data$Longitude) )
sum( is.na(clean_data$Lon_Lat) )
all( which( is.na(clean_data$Latitude) ) == which( is.na(clean_data$Longitude) ) )
na_indices <- which( is.na(clean_data$Longitude) )
clean_data$Lon_Lat[na_indices]
all( is.na(clean_data$Lon_Lat[na_indices]) )
```

```
## [1] 59
## [1] 59
## [1] 59
## [1] TRUE
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
## [51] NA NA NA NA NA NA NA NA NA
## [1] TRUE
```

```r
# The NAs are identical between the three columns; no info lost by removing the
# duplicate column.
clean_data <- select( .data = clean_data, -Lon_Lat )

#~~# Transforming the 'SEX' Columns
unique( clean_data$PERP_SEX ) # Note the "(null)" entries
```

```
## [1] "M"      "(null)" NA       "F"      "U"
```

```r
# Reassign the "(null)" entries as NA
null_idxs <- which( clean_data$PERP_SEX == "(null)" )
clean_data$PERP_SEX[null_idxs] <- NA
rm( null_idxs ) # Removing indices that are no longer needed

unique( clean_data$VIC_SEX ) # No obvious needs for reassignment.
```

```
## [1] "M" "F" "U"
```

```r
# Checking that cleanup was successful
unique( clean_data$PERP_SEX )
```

```
## [1] "M" NA  "F" "U"
```

```r
# Casting the sex data as factors:
clean_data <- mutate( .data = clean_data,
    Perp_Sex = as.factor( PERP_SEX ) )
clean_data <- mutate( .data = clean_data,
    Vic_Sex = as.factor( VIC_SEX ) )

# Removing the original 'SEX' columns:
clean_data <- select( .data = clean_data, c(-PERP_SEX, -VIC_SEX) )

#~~# Transforming the 'RACE' Columns
unique( clean_data$PERP_RACE ) # Note the NA and "(null)" entries.
```

```
## [1] "BLACK"                          "(null)"
## [3] NA                               "UNKNOWN"
## [5] "WHITE HISPANIC"                 "BLACK HISPANIC"
## [7] "ASIAN / PACIFIC ISLANDER"       "WHITE"
## [9] "AMERICAN INDIAN/ALASKAN NATIVE"
```

```r
unique( clean_data$VIC_RACE ) # No obvious needs for recoding.
```

```
## [1] "BLACK"                          "WHITE"
## [3] "WHITE HISPANIC"                 "BLACK HISPANIC"
## [5] "ASIAN / PACIFIC ISLANDER"       "UNKNOWN"
## [7] "AMERICAN INDIAN/ALASKAN NATIVE"
```

```r
# Reassigning "(null)" entries as NA
null_idxs <- which( clean_data$PERP_RACE == "(null)" )
clean_data$PERP_RACE[null_idxs] <- NA
rm( null_idxs ) # Removing indices that are no longer needed

# Checking that reassignment was successful:
unique( clean_data$PERP_RACE )
```

```
## [1] "BLACK"                          NA
## [3] "UNKNOWN"                        "WHITE HISPANIC"
## [5] "BLACK HISPANIC"                 "ASIAN / PACIFIC ISLANDER"
## [7] "WHITE"                          "AMERICAN INDIAN/ALASKAN NATIVE"
```

```r
# Casting the 'RACE' columns as factors:
clean_data <- mutate( .data = clean_data, Perp_Race = as.factor( PERP_RACE ) )
clean_data <- mutate( .data = clean_data, Vic_Race = as.factor( VIC_RACE ) )

# Removing the original 'RACE' columns:
clean_data <- select( .data = clean_data, c(-PERP_RACE, -VIC_RACE) )

#~~~# Transforming the 'AGE' columns:
unique(clean_data$PERP_AGE_GROUP) # Note the: (null), 1020, 940, 224, 1028
```

```
##  [1] "25-44"   "(null)"  NA        "18-24"   "45-64"   "UNKNOWN" "<18"
##  [8] "65+"     "1020"    "940"     "224"     "1028"
```

```r
unique(clean_data$VIC_AGE_GROUP) # Note the: 1022
```

```
## [1] "25-44"   "18-24"   "45-64"   "65+"     "<18"     "UNKNOWN" "1022"
```

```r
# Reassigning anomalous entries in PERP_AGE
anom_idxs = which( clean_data$PERP_AGE_GROUP %in% c("(null)", "1020", "940",
        "224", "1028") )
clean_data$PERP_AGE_GROUP[anom_idxs] <- NA
rm( anom_idxs ) # Removing indices that are longer needed.

# Reassigning anomalous entries in VIC_AGE
anom_idxs = which( clean_data$VIC_AGE_GROUP == "1022" )
clean_data$VIC_AGE_GROUP[anom_idxs] <- NA
rm( anom_idxs ) # Removing indices that are longer needed

# Checking for success in reassignment:
unique(clean_data$PERP_AGE_GROUP)
```

```
## [1] "25-44"   NA        "18-24"   "45-64"   "UNKNOWN" "<18"     "65+"
```

```r
unique(clean_data$VIC_AGE_GROUP)
```

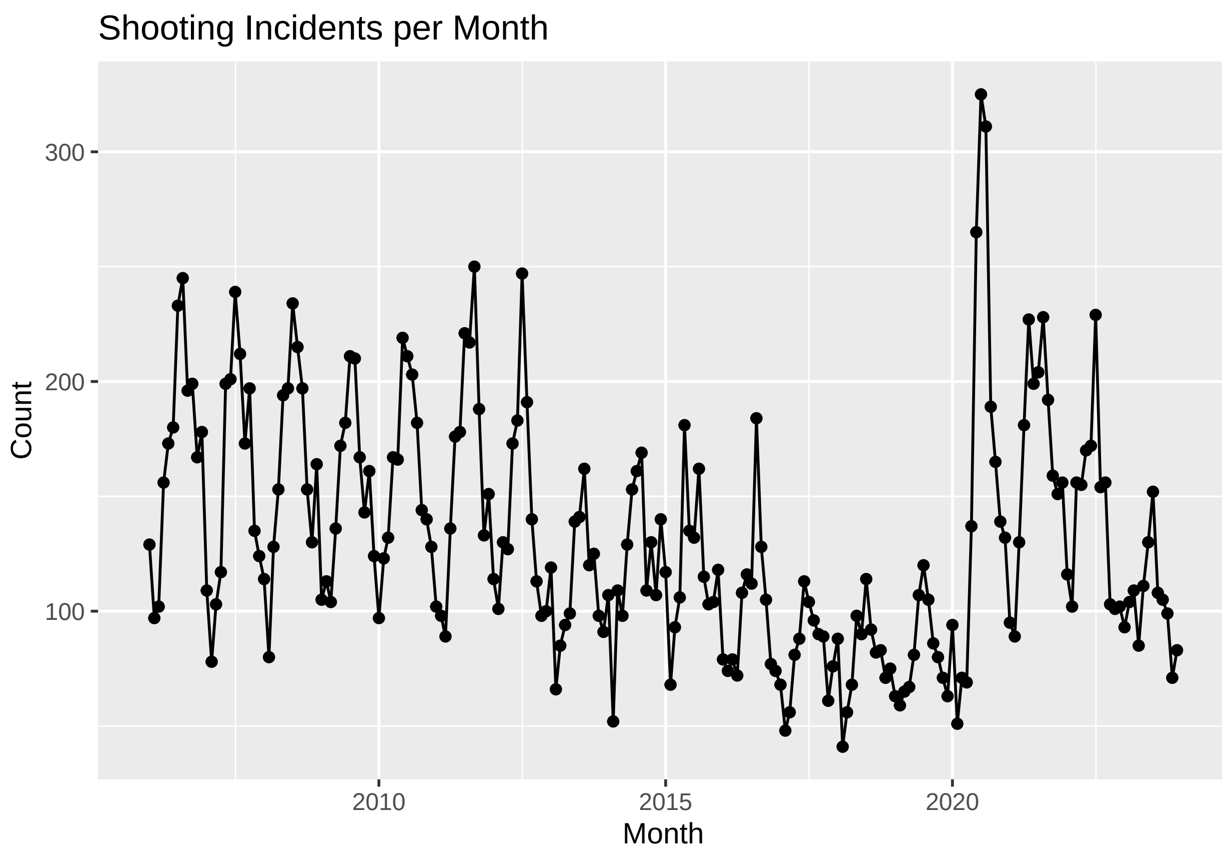
```
## [1] "25-44"   "18-24"   "45-64"   "65+"     "<18"     "UNKNOWN" NA
```

```r
# Casting the 'AGE' data as factors:
clean_data <- mutate( .data = clean_data, Perp_Age = as.factor( PERP_AGE_GROUP ) )
clean_data <- mutate( .data = clean_data, Vic_Age = as.factor( VIC_AGE_GROUP ) )

# Removing the original 'AGE' columns:
clean_data <- select( .data = clean_data, c(-PERP_AGE_GROUP, -VIC_AGE_GROUP) )

#~~~# Casting 'BORO' and 'JURISDICTION_CODE' as factors:
clean_data <- mutate( .data = clean_data, Boro = as.factor( BORO ) )
clean_data <- mutate( .data = clean_data,
    Jurisdiction_Code = as.factor( JURISDICTION_CODE ) )

# Removing original 'BORO' and 'JURISDICTION_CODE' columns:
clean_data <- select( .data = clean_data, c(-BORO, -JURISDICTION_CODE) )

#~~~# Removing original 'X_COORD_CD' and 'Y_COORD_CD' columns:
clean_data <- select( .data = clean_data, c(-X_COORD_CD, -Y_COORD_CD) )
```

Step 3 - Analysis and Visualization of Data:

With the data set tidied and transformed, I was curious if there were any obvious patterns over time in the shooting data. This is initially as exploratory data analysis, as opposed to directly trying to answer a question. The first plot I made was to plot shooting incidents per month versus time. The number of incidents per month was found using the **group_by()** and **summarize()** functions from the tidyverse package. This plot is just below. We can see in the plot that there appears to be a seasonal pattern in the shooting data with yearly low values in the winter; but this will be clarified in a later plot too.

```
ana_data <- mutate( .data = clean_data,
    Month = floor_date(occour_date_time, "month") )

# dev.new() # Pop out a new figure window
ana_data %>%
    group_by( Month ) %>%
    summarize( Count = n() ) %>%
    ggplot( aes( x = Month, y = Count) ) + geom_line() + geom_point() +
    labs( title = "Shooting Incidents per Month")
```

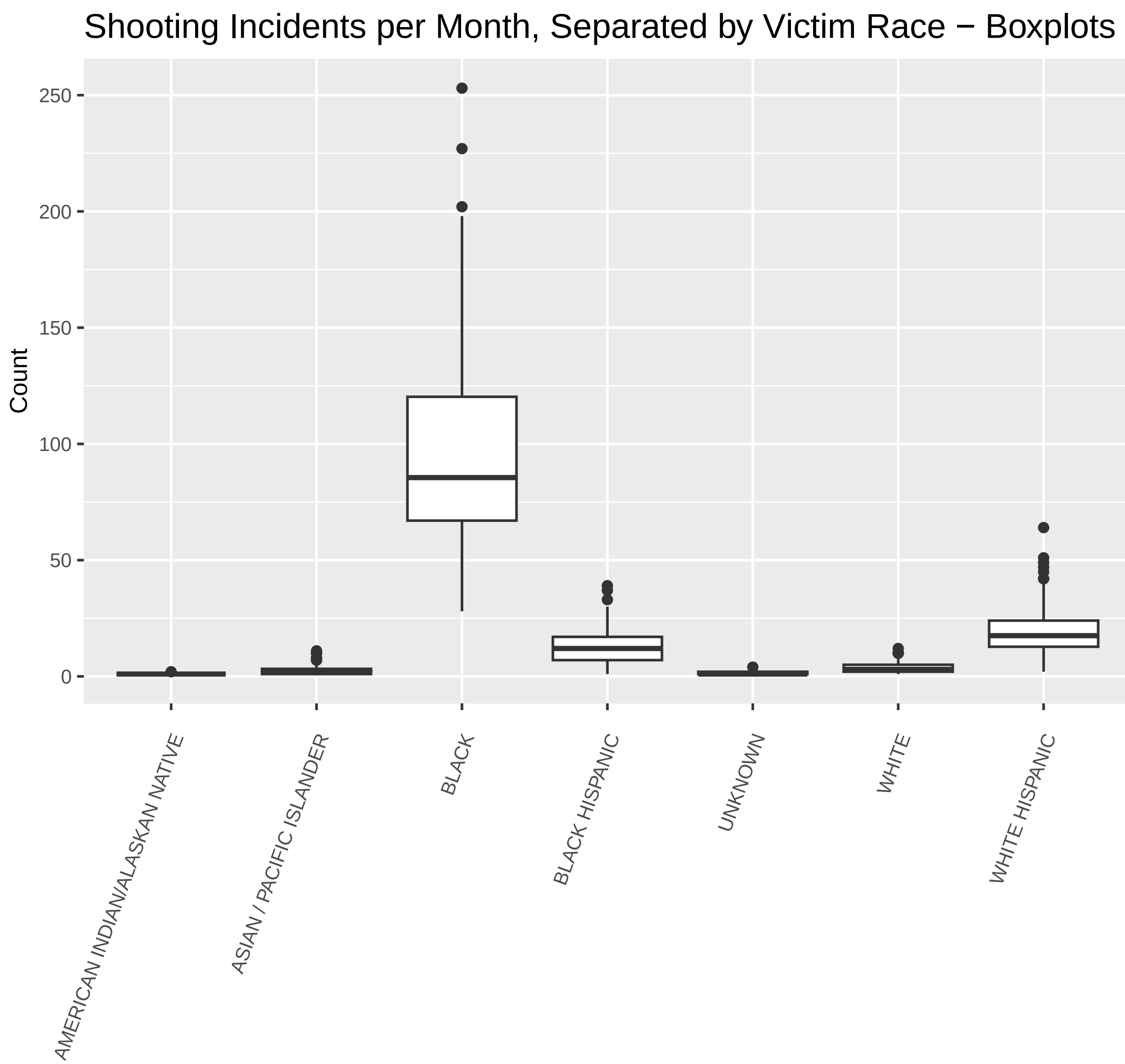


The second plot I made, was to see if there is a difference in number shooting victims with respect to race. Below I have used box plots for each victim race category, with shooting incidents per month on the y-axis. This is the exact same data (shooting incidents per month) as the first plot above, but split into bins for each victim race. For readers not familiar with boxplots; the center line inside the box represents the median number, and the top and bottom of the box representing the 25th and 75th percentiles. Further whiskers and dots represent values of 1.5 times the inter-quartile-range and outliers in the data respectively. This plot shows there is a large difference in both the median numbers of shootings per month with respect to race, and a large amount of variance in numbers of shootings per month too.

```
#dev.new() # Pop out a new figure window
ana_data %>%
    group_by( Vic_Race, Month ) %>%
    summarize( Count = n() ) %>%
    ggplot() + geom_boxplot( aes(x = Vic_Race, y = Count) ) +
        theme( text = element_text(size = 10),
            plot.title = element_text(size=14),
```

```
        axis.text.x = element_text(angle = 70, vjust = 0.95, hjust=1),
        axis.title.x=element_blank() ) +
    labs( title = "Shooting Incidents per Month, Separated by Victim Race - Boxplots")
```

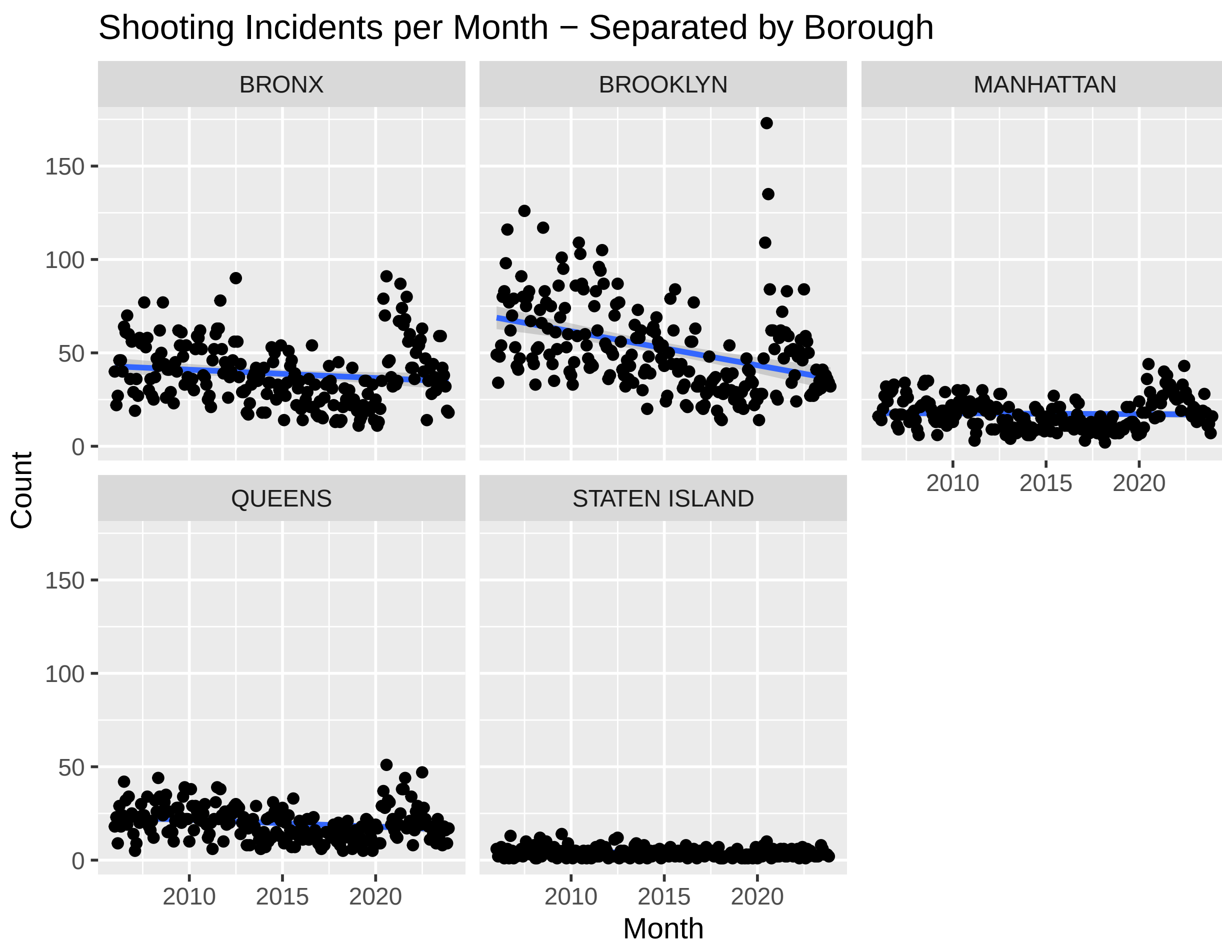
```
## `summarise()` has grouped output by 'Vic_Race'. You can override using the
## `.groups` argument.
```



The third plot I was interested in making was how shooting incidents varied over the boroughs of the city. I added trend lines to these plots to indicate if shooting incidents were largely increasing or decreasing with respect to time for each borough.

```
#dev.new() # Pop out a new figure window
ana_data %>%
    group_by( Month, Boro ) %>%
    summarize( Count = n() ) %>%
    ggplot( aes( x = Month, y = Count) ) + geom_smooth( method = "lm" ) +
    geom_point() + facet_wrap( ~ Boro ) +
    labs( title = "Shooting Incidents per Month - Separated by Borough")
```

```
## `summarise()` has grouped output by 'Month'. You can override using the
## `.groups` argument.
## `geom_smooth()` using formula = 'y ~ x'
```



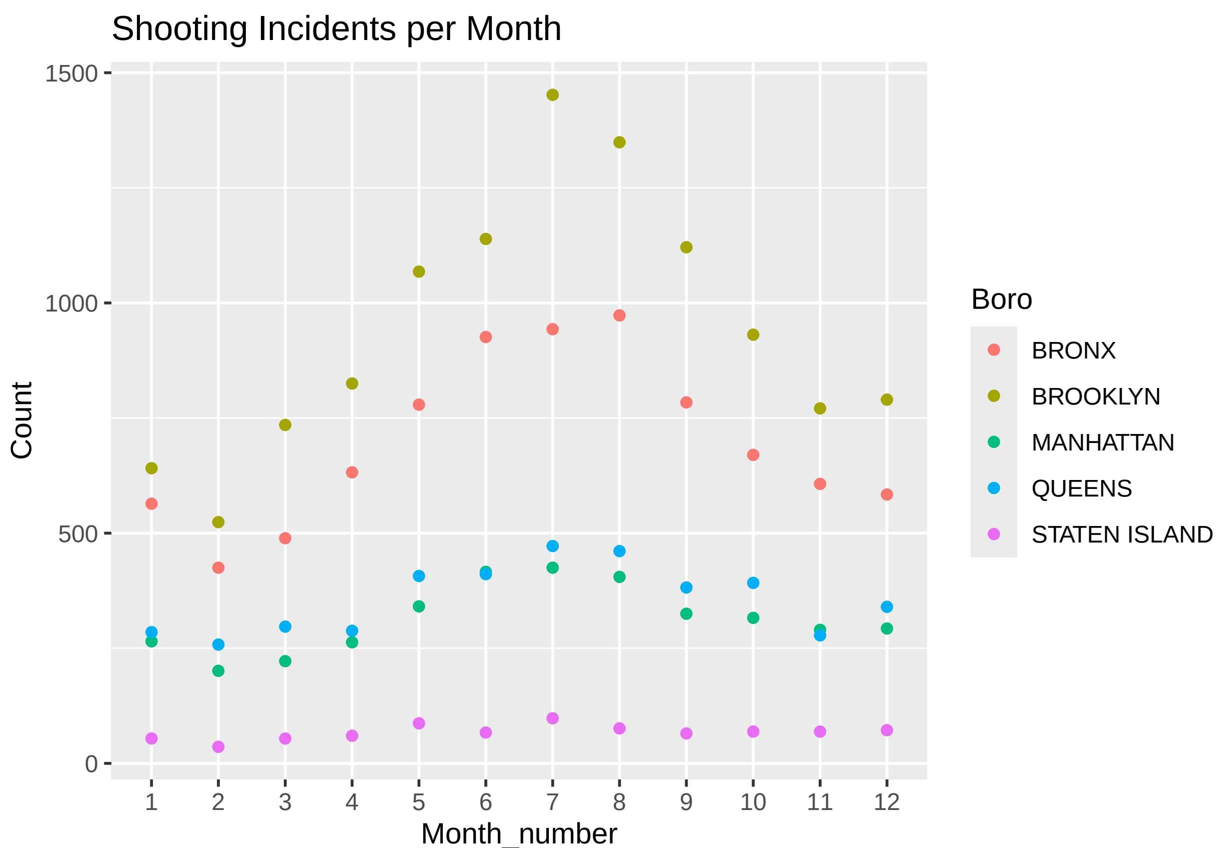
Step 3.5 - Reanalysis of Shooting Dependence on Month and Modeling

This section does not neatly fit into the steps as laid out in the week 3 assignment. I wanted to more clearly plot the dependence of shooting incidence on the month, as well as try to fit a more defined linear model of the number of shooting incidents as a function of the month and borough in the city.\

A plot to more clearly show the dependence of month on the number of shooting incidents is shown below:

```
ana_data %>%
    mutate( Month_number = as.factor( month(Month) ) ) %>%
    group_by( Boro, Month_number ) %>%
    summarize( Count = n() ) %>%
    ggplot( aes( x = Month_number, y = Count, color = Boro) ) +
    geom_point() + labs( title = "Shooting Incidents per Month")
```

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



Given that the exploratory data analysis has shown strong relationships between victim race, borough, and month on the number of shooting incidents, I next fit a linear model to the data. This linear model uses these three variables as predictors, as well as using the year to allow for shooting incidents trending in time and using number of shooting incidents per month as the response variable. Please see the code and output below:

```
ana_data2 <- ana_data %>%
    mutate( Month_Number = as.factor( month(Month) ) ) %>%
    group_by( Month_Number, Vic_Race, Boro ) %>%
    summarize( Count = n() )
```

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

```
shooting_incidents_model <- lm( data = ana_data2,
    Count ~ Month_Number + Vic_Race + Boro )
summary( shooting_incidents_model )
```

```
##
## Call:
## lm(formula = Count ~ Month_Number + Vic_Race + Boro, data = ana_data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -221.04  -56.57    4.03   37.71  746.36
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          -8.728     43.779  -0.199 0.842113
## Month_Number2                       -13.918     30.017  -0.464 0.643216
```

```
## Month_Number3                        -2.869     29.344  -0.098 0.922176
## Month_Number4                         3.709     29.744   0.125 0.900844
## Month_Number5                        28.748     29.596   0.971 0.332131
## Month_Number6                        35.386     29.290   1.208 0.227918
## Month_Number7                        59.954     29.248   2.050 0.041220 *
## Month_Number8                        46.603     29.508   1.579 0.115279
## Month_Number9                        29.552     29.767   0.993 0.321596
## Month_Number10                       15.630     29.767   0.525 0.599903
## Month_Number11                        1.995     29.508   0.068 0.946130
## Month_Number12                        6.716     29.753   0.226 0.821553
## Vic_RaceASIAN / PACIFIC ISLANDER     28.264     40.222   0.703 0.482776
## Vic_RaceBLACK                       372.668     39.842   9.354  < 2e-16 ***
## Vic_RaceBLACK HISPANIC               82.002     39.842   2.058 0.040409 *
## Vic_RaceUNKNOWN                      11.868     41.814   0.284 0.776727
## Vic_RaceWHITE                        47.552     39.842   1.194 0.233584
## Vic_RaceWHITE HISPANIC              106.802     39.842   2.681 0.007741 **
## BoroBROOKLYN                         40.748     18.308   2.226 0.026758 *
## BoroMANHATTAN                       -70.199     18.842  -3.726 0.000231 ***
## BoroQUEENS                          -57.623     18.285  -3.151 0.001784 **
## BoroSTATEN ISLAND                  -134.506     20.113  -6.688 1.06e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 109.2 on 310 degrees of freedom
## Multiple R-squared:  0.6214, Adjusted R-squared:  0.5957
## F-statistic: 24.22 on 21 and 310 DF,  p-value: < 2.2e-16
```

The summary of this model shows that they are significant values (Months/Races/Boroughs) in each predictor with respect to the number of shootings per month. But I also believe that further modifications would be needed (e.g. using a glm, or an ordinal model, etc.) to refine this model for better predictions. To keep this presentation shorter, I will same model refinement for another project.

Step 4 - Consideration of Bias:

This data set is very interesting, and I think there are many possible sources of bias we could observe. From the viewpoint of data collection, I think it is possible that some areas of New York City may have an easier time documenting shooting incidents than others. For example some neighborhoods may be more willing to report crimes to the police than other neighborhoods. Another possible source of bias in data collection could be differing definitions of "shooting incidents" between different police precincts in the city. For example, perhaps one precinct would classify an accidental gun discharge as a shooting incident, but another precinct would not. Both of these ideas of bias in data collection are hypothetical, as I am unsure if they occur for this particular dataset.\

For my own approach to this data, I have assumed that the latitude and longitude are equally good for geolocating an incident, as the x and y coordinates in the raw data. If these systems do differ, then any plots I make would be skewed compared someone using the x and y coordinates. Another source of bias I could be introducing was casting some victim age codes as NA. I believe that some of these codes were typos, or perhaps errors in the data, and cast them as NA. If instead these codes had meaning then I could have biased results. I think my biggest source of bias in analysis of this data, is my own lack of knowledge about the New York region, and NYPD data in particular.

Appendix - Session Information

To wrap up this document, I include session info for the R code, i.e. calling out the current versions of packages for reproducibility.

```r
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Denver
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.3 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
##  [5] purrr_1.0.2     readr_2.1.5     tidyr_1.3.1     tibble_3.2.1
##  [9] ggplot2_3.5.1   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.3    lattice_0.22-6    stringi_1.8.4
##  [5] hms_1.1.3         digest_0.6.37     magrittr_2.0.3    evaluate_1.0.0
##  [9] grid_4.4.1        timechange_0.3.0  fastmap_1.2.0     Matrix_1.7-0
## [13] mgcv_1.9-1        fansi_1.0.6       scales_1.3.0      cli_3.6.3
## [17] rlang_1.1.4       crayon_1.5.3      splines_4.4.1     bit64_4.0.5
## [21] munsell_0.5.1     withr_3.0.1       yaml_2.3.10       tools_4.4.1
## [25] parallel_4.4.1    tzdb_0.4.0        colorspace_2.1-1  curl_5.2.2
## [29] vctrs_0.6.5       R6_2.5.1          lifecycle_1.0.4   bit_4.0.5
## [33] vroom_1.6.5       pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.5
## [37] glue_1.7.0        xfun_0.47         tidyselect_1.2.1  highr_0.11
## [41] rstudioapi_0.16.0 knitr_1.48        farver_2.1.2      nlme_3.1-164
## [45] htmltools_0.5.8.1 rmarkdown_2.28    labeling_0.4.3    compiler_4.4.1
```