The NYPD Shootout incident

In this report, I am going to analyze the NYPD Shootout incident that was reported from the year 2006 to 2022.

This data set consists of 27132 observations with 21 variables including incidents that happened in cities, perpetrator's details, victim's details, date, time of the incident, and many more.

For the report, I will be using various libraries and techniques to tidy the data, clean, visualization, and model for better analysis.

1. Importing the Data:

Firstly, importing the required packages and loading the libraries and then reading the data using csv file. Reading the csv file.

```
data <- read_csv('~/Downloads/NYPD_Shooting_Incident_Data__Historic_.csv', show_col_types = FALSE)

view(data)
```

Checking if data set is in data.frame format.

```
is.data.frame(data)
```

```
## [1] TRUE
```

By using the head, glimpse, and summary functions, we will get to know some information about the data like how many columns(including column names) and rows, data types, and statistical values on each column (not for the character data type). This data has 21 columns and 27132 rows.

```
head(data)
```

```
## # A tibble: 6 x 21
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     LOC_OF_OCCUR_DESC PRECINCT
##          <dbl> <chr>      <time>     <chr>    <chr>                <dbl>
## 1    228798151 05/27/2021 21:30      QUEENS   <NA>                   105
## 2    137471050 06/27/2014 17:40      BRONX    <NA>                    40
## 3    147998800 11/21/2015 03:56      QUEENS   <NA>                   108
## 4    146837977 10/09/2015 18:30      BRONX    <NA>                    44
## 5     58921844 02/19/2009 22:58      BRONX    <NA>                    47
## 6    219559682 10/21/2020 21:36      BROOKLYN <NA>                    81
## # i 15 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>
```

```
glimpse(data)
```

```
## Rows: 27,312
## Columns: 21
## $ INCIDENT_KEY            <dbl> 228798151, 137471050, 147998800, 146837977, 58~
## $ OCCUR_DATE              <chr> "05/27/2021", "06/27/2014", "11/21/2015", "10/~
## $ OCCUR_TIME              <time> 21:30:00, 17:40:00, 03:56:00, 18:30:00, 22:58~
## $ BORO                    <chr> "QUEENS", "BRONX", "QUEENS", "BRONX", "BRONX",~
## $ LOC_OF_OCCUR_DESC       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ PRECINCT                <dbl> 105, 40, 108, 44, 47, 81, 114, 81, 105, 101, 2~
## $ JURISDICTION_CODE       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 2~
## $ LOC_CLASSFCTN_DESC      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ LOCATION_DESC           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "MULTI DWE~
## $ STATISTICAL_MURDER_FLAG <lgl> FALSE, FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, ~
## $ PERP_AGE_GROUP          <chr> NA, NA, NA, NA, "25-44", NA, NA, NA, NA, "25-4~
## $ PERP_SEX                <chr> NA, NA, NA, NA, "M", NA, NA, NA, NA, "M", NA, ~
## $ PERP_RACE               <chr> NA, NA, NA, NA, "BLACK", NA, NA, NA, NA, "BLAC~
## $ VIC_AGE_GROUP           <chr> "18-24", "18-24", "25-44", "<18", "45-64", "25~
## $ VIC_SEX                 <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
## $ VIC_RACE                <chr> "BLACK", "BLACK", "WHITE", "WHITE HISPANIC", "~
## $ X_COORD_CD              <dbl> 1058925.0, 1005028.0, 1007667.9, 1006537.4, 10~
## $ Y_COORD_CD              <dbl> 180924.0, 234516.0, 209836.5, 244511.1, 262189~
## $ Latitude                <dbl> 40.66296, 40.81035, 40.74261, 40.83778, 40.886~
## $ Longitude               <dbl> -73.73084, -73.92494, -73.91549, -73.91946, -7~
## $ Lon_Lat                 <chr> "POINT (-73.73083868899994 40.662964620000025)~
```

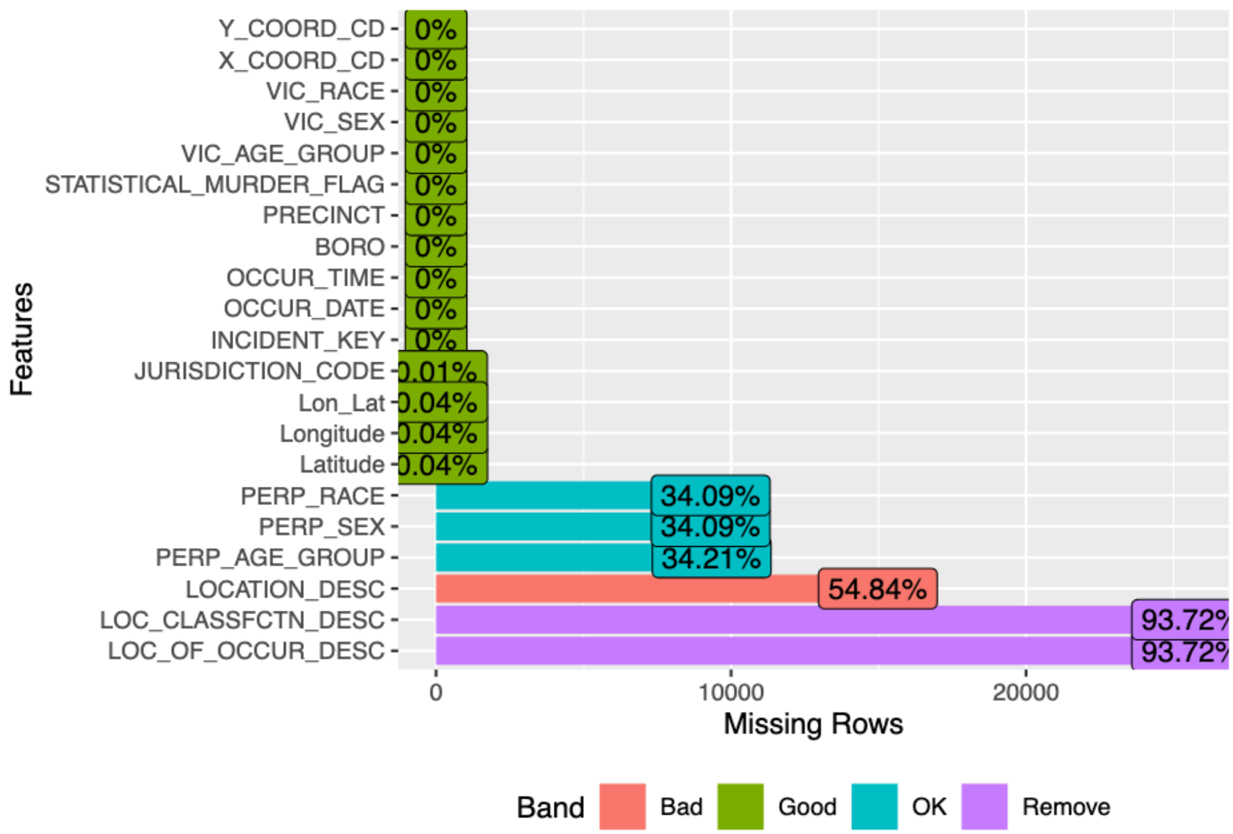
```r
summary(data)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:27312       Length:27312      Length:27312
##  1st Qu.: 63860880   Class :character   Class1:hms        Class :character
##  Median : 90372218   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :120860536                      Mode  :numeric
##  3rd Qu.:188810230
##  Max.   :261190187
##
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:27312       Min.   :  1.00   Min.   :0.0000    Length:27312
##  Class :character   1st Qu.: 44.00   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 68.00   Median :0.0000    Mode  :character
##                     Mean   : 65.64   Mean   :0.3269
##                     3rd Qu.: 81.00   3rd Qu.:0.0000
##                     Max.   :123.00   Max.   :2.0000
##                                      NA's   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:27312       Mode :logical           Length:27312
##  Class :character   FALSE:22046             Class :character
##  Mode  :character   TRUE :5266              Mode  :character
##
##
##
##
##    PERP_SEX          PERP_RACE         VIC_AGE_GROUP        VIC_SEX
##  Length:27312       Length:27312       Length:27312       Length:27312
##  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:27312       Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character   1st Qu.:1000028   1st Qu.:182834   1st Qu.:40.67
##  Mode  :character   Median :1007731   Median :194487   Median :40.70
##                     Mean   :1009449   Mean   :208127   Mean   :40.74
##                     3rd Qu.:1016838   3rd Qu.:239518   3rd Qu.:40.82
##                     Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                        NA's   :10
##    Longitude        Lon_Lat
##  Min.   :-74.25   Length:27312
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA's   :10
```

The use of colSums() function is to check the number of missing values(row) in the data. LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION_DESC columns have more than 50% missing values.

```
##            INCIDENT_KEY              OCCUR_DATE              OCCUR_TIME
##                       0                       0                       0
##                    BORO       LOC_OF_OCCUR_DESC                PRECINCT
##                       0                   25596                       0
##       JURISDICTION_CODE      LOC_CLASSFCTN_DESC           LOCATION_DESC
##                       2                   25596                   14977
## STATISTICAL_MURDER_FLAG          PERP_AGE_GROUP                PERP_SEX
##                       0                    9344                    9310
##               PERP_RACE           VIC_AGE_GROUP                 VIC_SEX
##                    9310                       0                       0
##                VIC_RACE              X_COORD_CD              Y_COORD_CD
##                       0                       0                       0
##                Latitude               Longitude                 Lon_Lat
##                      10                      10                      10
```



2. Tidying and Transforming the data:

Since the OCCUR_DATE column is in character data type, converting it into DATE format and also adding Month and Year columns to the data set to make it easier for further analysis and for visualization. I am changing the OCCUR_TIME column format from 24hrs(hms) to 12hrs(ims) for better graphs and analysis like whether a murder incident happened during the daytime/nighttime.

```
data$OCCUR_DATE <- as.Date(data$OCCUR_DATE, '%m/%d/%Y')
data$Month <- month(data$OCCUR_DATE)
data$Year <- year(data$OCCUR_DATE)
data$OCCUR_TIME <- format(strptime(data$OCCUR_TIME, format = '%H:%M:%S'), '%I:%M:%S %p')
```

Some columns have null values, blank values, and just some random numbers which may affect the analysis and graph. So will rename it and fill the missing values by UNKNOWN.

```
data$PERP_AGE_GROUP[data$PERP_AGE_GROUP %in% c('(null)', NA, '224', '1020', '940', '1022')] <- NA
data$PERP_RACE[data$PERP_RACE %in% c('(null)', NA)] <- 'UNKNOWN'
data$PERP_SEX[data$PERP_SEX %in% c('(null)', NA, 'U')] <- 'UNKNOWN'
data$LOCATION_DESC[data$LOCATION_DESC %in% c('(null)', NA)] <- 'UNKNOWN'
data$VIC_SEX[data$VIC_SEX == 'U'] <- 'UNKNOWN'
data$VIC_AGE_GROUP[data$VIC_AGE_GROUP == '1022'] <- NA
data$STATISTICAL_MURDER_FLAG[data$STATISTICAL_MURDER_FLAG == TRUE] <- 1
data$STATISTICAL_MURDER_FLAG[data$STATISTICAL_MURDER_FLAG == FALSE] <- 0
```

Columns like LOC_OF_OCCUR_DESC and LOC_CLASSFCTN_DESC have more than 50% missing values and X_COORD_CD, Y_COORD_CD, and Lon_Lat columns are not helping for the analysis so I will drop a few columns which is not required for analysis and also dropping NA values and duplicated values.

```
data <- subset(data, select = -c(LOC_OF_OCCUR_DESC, X_COORD_CD, Y_COORD_CD, Lon_Lat, LOC_CLASSFCTN_DESC)

data <- na.omit(data)

data <- data[!duplicated(data$INCIDENT_KEY),]
```

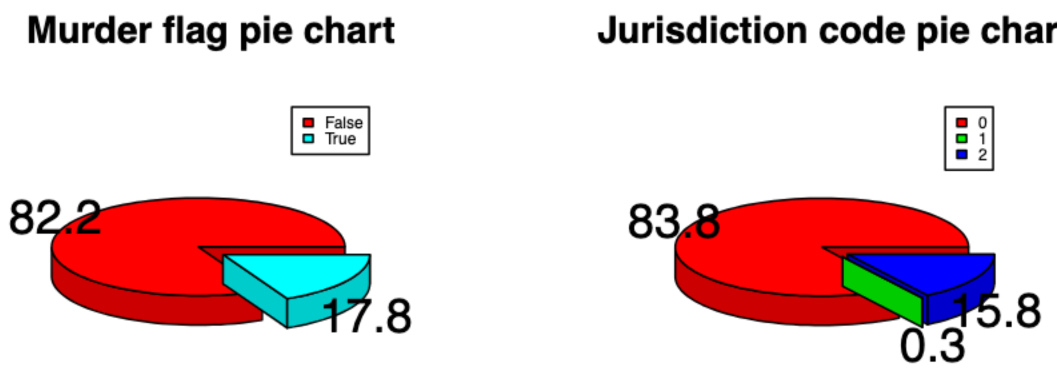
3.Exploratory Data Analysis / Visualizing Data:

```
par(mfrow = c(1, 2))

table <- table(data$STATISTICAL_MURDER_FLAG)
lab <- round(100*table/sum(table), 1)
pie3D(table, labels = lab, explode = 0.1, main = 'Murder flag pie chart',  col = rainbow(length(table))
legend("topright", c('False', 'True'), cex = 0.5, fill = rainbow(length(table)))

table <- table(data$JURISDICTION_CODE)
lab <- round(100*table/sum(table), 1)
pie3D(table, labels = lab, explode = 0.1, main = 'Jurisdiction code pie chart',  col = rainbow(length(ta
legend("topright", c('0', '1', '2'), cex = 0.5, fill = rainbow(length(table)))
```

1. **Finding STATISTICAL_MURDER_FLAGs and JURISDICTION_CODE total percent-**



age using Pie chart.

The pie chart for STATISTICAL_MURDER_FLAG shows that 17.4% of cases were murder flags out of total cases and the highest number of cases were based on JURISDICTION_CODE code 0 with 83% followed by code 2 with 16.7%.

```
table(data$BORO)
```

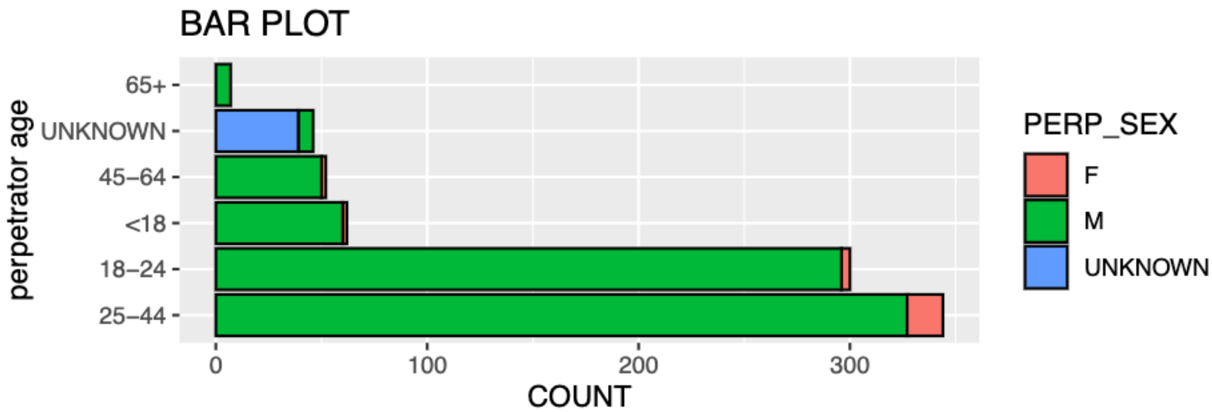
2. **Since most of the incidents took place in Brooklyn, will consider the perpetrator's age only in Brooklyn city with the number of murder cases.**

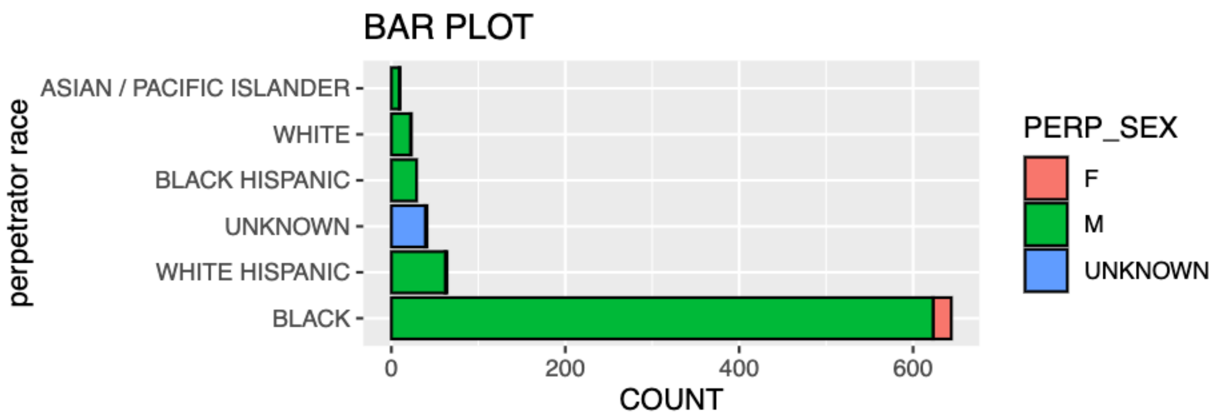
```
##
##         BRONX      BROOKLYN     MANHATTAN        QUEENS STATEN ISLAND
##          3679          4891          1769          1959           468
```

```
p_age <- ggplot(data = brooklyn, mapping = aes(x = fct_infreq(PERP_AGE_GROUP), fill = PERP_SEX)) +
     geom_bar(data = brooklyn, color = 'black') +
     labs(title = 'BAR PLOT') + xlab('perpetrator age') + ylab('COUNT') + coord_flip()

p_race <- ggplot(data = brooklyn, mapping = aes(x = fct_infreq(PERP_RACE), fill = PERP_SEX)) +
     geom_bar(data = brooklyn, color = 'black') +
     labs(title = 'BAR PLOT') + xlab('perpetrator race') + ylab('COUNT') + coord_flip()

grid.arrange(p_age, p_race)
```





From the graph, we can say that most murder incidents happened in Brooklyn. 344 shooting cases with perpetrators aged 25-44 years followed by 18-24 years with 300 cases in Brooklyn. Men are the perpetrators in the vast majority of those shooting incidents in Brooklyn. In Brooklyn, 644 reports say that the Black Race was the majority of perpetrators who were responsible for the incident followed by White Hispanics with 64 cases.

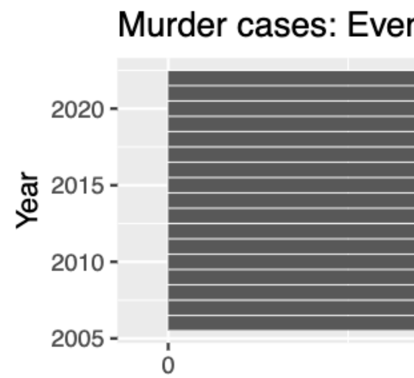
```
 murder_time <- function(var){ data |>
  select(OCCUR_TIME, Year, STATISTICAL_MURDER_FLAG) |>
    filter(str_detect(data$OCCUR_TIME, var)) |>
     filter(STATISTICAL_MURDER_FLAG == 1)
}

am <- murder_time('AM')
pm <- murder_time('PM')

morning <- ggplot(data = am, mapping = aes(x = Year))+
  geom_bar() +
     labs(title = 'Murder cases: Morning') + xlab('Year') + ylab('count') + coord_flip()

evening <- ggplot(data = pm, mapping = aes(x = Year))+
  geom_bar() +
     labs(title = 'Murder cases: Evening') + xlab('Year') + ylab('count') + coord_flip()

grid.arrange(morning, evening)
```



3. Finding at what time most of the shooting incidents took place every year:

So based on the graph, in the year 2006 the highest number of murder cases were reported at morning(AM) followed by the year 2010 and year 2021 but very less cases were reported in the year 2017 to the year 2019. Similarly in the years 2010 and 2021, the highest number of cases were reported in the evening(PM) time slot but very less cases were reported in the year 2017 to the year 2019 as well.

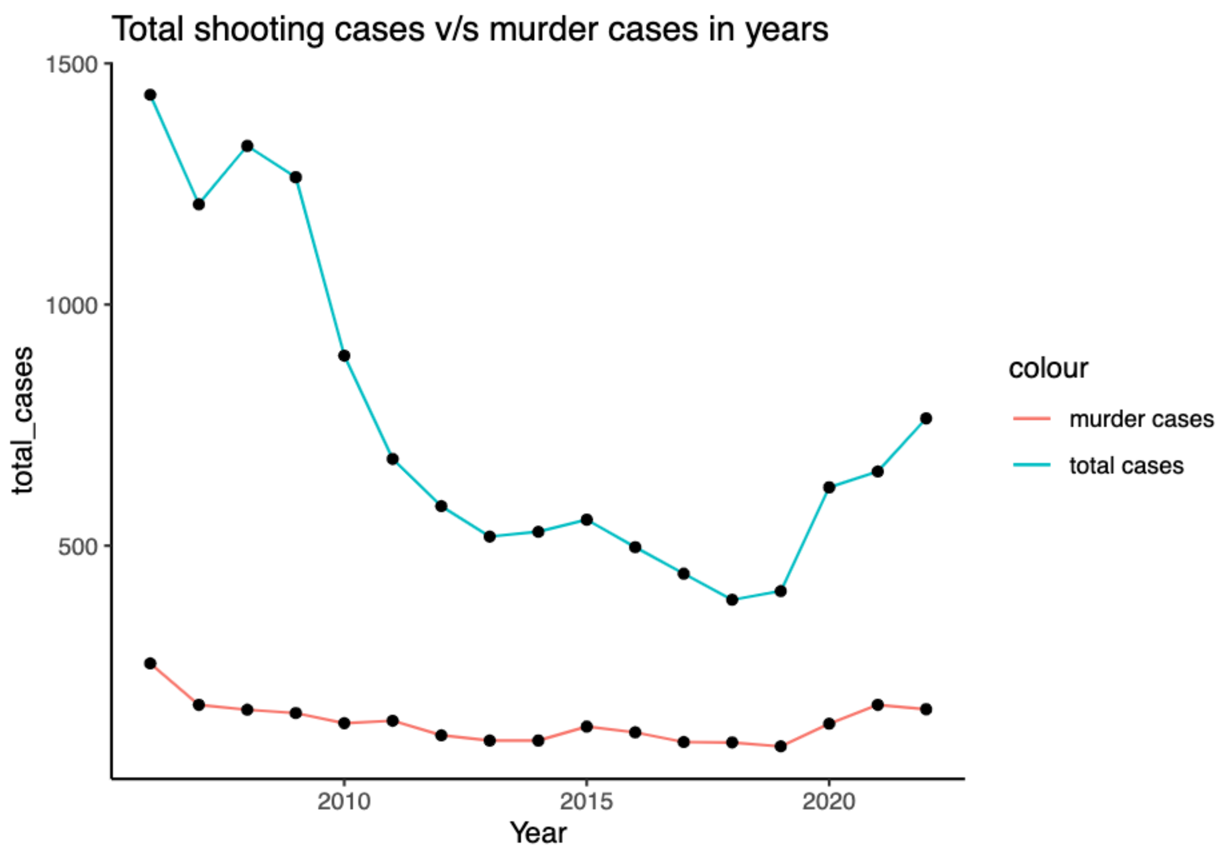
4. Data Analyzing:

Let us check the total number of cases and number of murder cases in each year and compare the results. While looking at the plot of total shooting cases reported from the year 2006 to the year 2022 and death(murder) cases, the total number of cases was reported less in the year from 2016 to the year 2019, but murder cases were pretty consistent.

```
year_df <- data |>
     group_by(Year) %>%
    summarise(death = sum(STATISTICAL_MURDER_FLAG), total_cases = n())

#year_df

ggplot(year_df)+
geom_line(aes(x = Year, y = total_cases, color = 'total cases'))+
geom_point(aes(x = Year, y = total_cases))+
geom_line(aes(x = Year, y = death, color = 'murder cases'))+
geom_point(aes(x = Year, y = death))+
labs(title = 'Total shooting cases v/s murder cases in years')+ theme_classic()
```



5. Data Modeling:

For modeling, we will classify based on the age groups of the perpetrators and whether they have been convicted of crimes.

Will split the data set into 80% training data and 20% testing for the analysis. We will consider training data for the modeling and will use testing data for the prediction.

```
set.seed(123)

split <- sample.split(data$STATISTICAL_MURDER_FLAG, SplitRatio = 0.8)

training <- subset(data, split == TRUE)
testing <- subset(data, split == FALSE)
```

Here I am considering the STATISTICAL_MURDER_FLAG column as the target variable and the PERP_AGE_GROUP columns as response variables.

```
model <- glm(STATISTICAL_MURDER_FLAG ~ PERP_AGE_GROUP, data = training, family = binomial(link = 'logit
summary(model)
```

Here I will fit the logistic regression model with a binomial as a parameter.

```
##
## Call:
## glm(formula = STATISTICAL_MURDER_FLAG ~ PERP_AGE_GROUP, family = binomial(link = "logit"),
##     data = training)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.62896    0.09178 -17.749  < 2e-16 ***
## PERP_AGE_GROUP18-24      0.15649    0.10156   1.541 0.123351
## PERP_AGE_GROUP25-44      0.53891    0.10025   5.375 7.64e-08 ***
## PERP_AGE_GROUP45-64      1.00207    0.14675   6.828 8.60e-12 ***
## PERP_AGE_GROUP65+        1.26606    0.33823   3.743 0.000182 ***
## PERP_AGE_GROUPUNKNOWN -1.72677    0.14907 -11.584  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9575.9  on 10212  degrees of freedom
## Residual deviance: 8950.2  on 10207  degrees of freedom
## AIC: 8962.2
##
## Number of Fisher Scoring iterations: 6
```

```r
exp(coef(model)['PERP_AGE_GROUP18-24'])
```

```
## PERP_AGE_GROUP18-24
##            1.169402
```

```r
exp(coef(model)['PERP_AGE_GROUP25-44'])
```

```
## PERP_AGE_GROUP25-44
##            1.714146
```

From the summary of the model, the coef value for the age group <18 is less than 0, and the probability of murdering someone decreases with that age. But all other age groups were greater than 0, which means the probability of involving in the murder was higher.

Also, using the expo() function, age group 18-24, the probability of involving in the murder was around 16%, and for the 25-44 age group, it was around 71%

```r
pred = predict(model, testing, type = 'response')

pred <- ifelse(pred  > 0.5, 1, 0)

testing$prediction <- pred

table(testing$STATISTICAL_MURDER_FLAG)
```

Now, we will predict the model using our testing data set. And adding a prediction column to the testing data to check the predicted values with the actual murder flag values in the STATISTICAL_MURDER_FLAG column.

```
##
##    0    1
## 2098  455
```

```r
table(testing$prediction)
```

```
##
##    0
## 2553
```

```r
matrix <- confusionMatrix(as.factor(testing$prediction), as.factor(testing$STATISTICAL_MURDER_FLAG))
```

```
## Warning in confusionMatrix.default(as.factor(testing$prediction),
## as.factor(testing$STATISTICAL_MURDER_FLAG)): Levels are not in the same order
## for reference and data. Refactoring data to match.
```

```r
matrix$overall['Accuracy']
```

```
##  Accuracy
## 0.8217783
```

```r
new = predict(model, newdata = data.frame(PERP_AGE_GROUP = '45-64'), type = 'response')
new
```

```
##         1
## 0.3482143
```

6. Conclusion:

As expected, most of the data points for the STATISTICAL_MURDER_FLAG fall under value 0(FALSE), so the model predicts the results based on the dominant class. That's the reason I got all 2553 predicted results under 0 value. I think the data set is highly imbalanced, with noisy features and missing values. Further analysis, new techniques and over sampling of the model are needed.

7.Biases:

Also, handling the missing values is needed and fitting poor model would also result in model performance. In my case, I have used PERP_AGE_GROUP as response variable, which has many levels.