# NYPD Shooting Incident Data Report

Manpreet Singh

2024-01-20

## NYPD Shooting Incident Data Report

## 1. Data set

As first step, lets fetch the NYPD shooting data and see how it looks like

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

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:27312       Length:27312       Length:27312
##  1st Qu.: 63860880   Class :character   Class :character   Class :character
##  Median : 90372218   Mode  :character   Mode  :character   Mode  :character
##  Mean   :120860536
##  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       Length:27312            Length:27312
##  Class :character   Class :character        Class :character
##  Mode  :character   Mode  :character        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.:1000029   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 data set contains 21 columns. Below is data definition

| Field Name | Description |
|---|---|
| INCIDENT_KEY | Randomly generated persistent ID for each incident |
| OCCUR_DATE | Exact date of the shooting incident |
| OCCUR_TIME | Exact time of the shooting incident |
| BORO | Borough where the shooting incident occurred |
| PRECINCT | Precinct where the shooting incident occurredPrecinct where the shooting incident occurred |
| JURISDICTION_CODE | Jurisdiction where the shooting incident occurred. Jurisdiction codes 0(Patrol), 1(Transit) and 2(Housing) represent NYPD whilst codes 3 and more represent non NYPD jurisdictions |
| LOCATION_DESC | Location of the shooting incident |
| STATISTICAL_MURDER_FLAG | Shooting resulted in the victim's death which would be counted as a murder |
| PERP_AGE_GROUP | Perpetrator's age within a category |
| PERP_SEX | Perpetrator's sex description |
| PERP_RACE | Perpetrator's race description |
| VIC_AGE_GROUP | Victim's age within a category |
| VIC_SEX | Victim's sex description |
| VIC_RACE | Victim's race description |
| X_COORD_CD | Midblock X-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet (FIPS 3104) |
| Y_COORD_CD | Midblock Y-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet (FIPS 3104) |
| Latitude, Longitude | Latitude, Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326) |

Reference: https://data.cityofnewyork.us/api/views/833y-fsy8/files/e4e3d86c-348f-4a16-a17f-19480c089429?download=true&filename=NYPD_Shootings_Incident_Level_Data_Footnotes.pdf

## 2. Data cleaning

1. From NYPD column definitions, the geo tagging related information is not accurate. E.g shootings in open areas are geo tagged to nearest street. Doing analysis on inaccurate geo tagged data may result in inaccurate conclusion, hence excluding geo tagging related columns from the data set

```
raw_data <- subset(raw_data, select = -c(X_COORD_CD, Y_COORD_CD, Latitude, Longitude, Lon_Lat))
```

2. Replacing missing data with -1 for column Jurisdiction.

```
raw_data$JURISDICTION_CODE = replace_na(raw_data$JURISDICTION_CODE, -1)
```

3. Replacing empty string with NA

```
raw_data$LOC_OF_OCCUR_DESC = dplyr::na_if(raw_data$LOC_OF_OCCUR_DESC, "")
raw_data$LOC_CLASSFCTN_DESC = dplyr::na_if(raw_data$LOC_CLASSFCTN_DESC, "")
raw_data$LOCATION_DESC = dplyr::na_if(raw_data$LOCATION_DESC, "")
raw_data$LOCATION_DESC = dplyr::na_if(raw_data$LOCATION_DESC, "(null)")

raw_data$PERP_AGE_GROUP = dplyr::na_if(raw_data$PERP_AGE_GROUP, "")
raw_data$PERP_SEX = dplyr::na_if(raw_data$PERP_SEX, "")
raw_data$PERP_RACE = dplyr::na_if(raw_data$PERP_RACE, "")

raw_data$VIC_AGE_GROUP = dplyr::na_if(raw_data$VIC_AGE_GROUP, "")
raw_data$VIC_SEX = dplyr::na_if(raw_data$VIC_SEX, "")
raw_data$VIC_RACE = dplyr::na_if(raw_data$VIC_RACE, "")
```

4. Replacing invalid values like “null”, age of 940, 1022 to UNKNOWN

```
#cleaning age, sex, race
raw_data$PERP_AGE_GROUP = dplyr::na_if(raw_data$PERP_AGE_GROUP, "(null)")
raw_data$PERP_SEX = dplyr::na_if(raw_data$PERP_SEX, "(null)")
raw_data$PERP_RACE = dplyr::na_if(raw_data$PERP_RACE, "(null)")
raw_data$VIC_AGE_GROUP = dplyr::na_if(raw_data$VIC_AGE_GROUP, "(null)")
raw_data$VIC_SEX = dplyr::na_if(raw_data$VIC_SEX, "(null)")
raw_data$VIC_RACE = dplyr::na_if(raw_data$VIC_RACE, "(null)")

col_repl = c("PERP_AGE_GROUP", "VIC_AGE_GROUP", "PERP_SEX")
val_repl = c("940", "1020", "224", "1022", "U")
raw_data[col_repl] <- sapply(raw_data[col_repl],
                             function(x) replace(x, x %in% val_repl, "UNKNOWN"))
```

5. Replacing all NAs to a common category called “UNKNOWN”

```
raw_data[is.na(raw_data)] <- "UNKNOWN"
```

6. Convert time to 24hr format, extract month and year explicitly from “OCCUR_TIME” column

```
raw_data$OCCUR_TIME = hms(raw_data$OCCUR_TIME)
raw_data$OCCUR_DATE = mdy(raw_data$OCCUR_DATE)
raw_data$OCCUR_MONTH = month(raw_data$OCCUR_DATE)
raw_data$OCCUR_YEAR = year(raw_data$OCCUR_DATE)
raw_data$Weekday = wday(raw_data$OCCUR_DATE, label=TRUE)
```

Converting time to time intervals of 1hour

```
get_time_category <- function(time) {
  switch(
    floor(time)+1,
      "0-1",
      "1-2",
      "2-3",
      "3-4",
      "4-5",
      "5-6",
      "6-7",
      "7-8",
      "8-9",
      "9-10",
      "10-11",
      "11-12",
      "12-13",
      "13-14",
      "14-15",
      "15-16",
      "16-17",
      "17-18",
      "18-19",
      "19-20",
      "20-21",
      "21-22",
      "22-23",
      "23-00"
  )
}
time_category_order = c("0-1", "1-2", "2-3", "3-4", "4-5","5-6","6-7",
                        "7-8","8-9", "9-10","10-11","11-12","12-13",
                        "13-14","14-15","15-16","16-17","17-18","18-19",
                        "19-20","20-21","21-22","22-23","23-00")
raw_data$time_category <- sapply(as.numeric(raw_data$OCCUR_TIME, "hours"), get_time_category)
raw_data$time_category <- factor(raw_data$time_category, levels=time_category_order)
```

5. Pre calculating death rate

```
shootings_by_year = raw_data |> group_by(OCCUR_YEAR) |> count(OCCUR_YEAR)
deaths_in_shooting = raw_data |> group_by(OCCUR_YEAR) |> count(STATISTICAL_MURDER_FLAG) |> filter(S
death_rate_df = left_join(shootings_by_year, deaths_in_shooting,
                          by = "OCCUR_YEAR")
death_rate_df$death_rate = death_rate_df$n.y*100/death_rate_df$n.x
```

6. Check if any value is left as NA. Zero would indicate no missing data left in data set

```
sum(is.na(raw_data))
```

```
## [1] 0
```

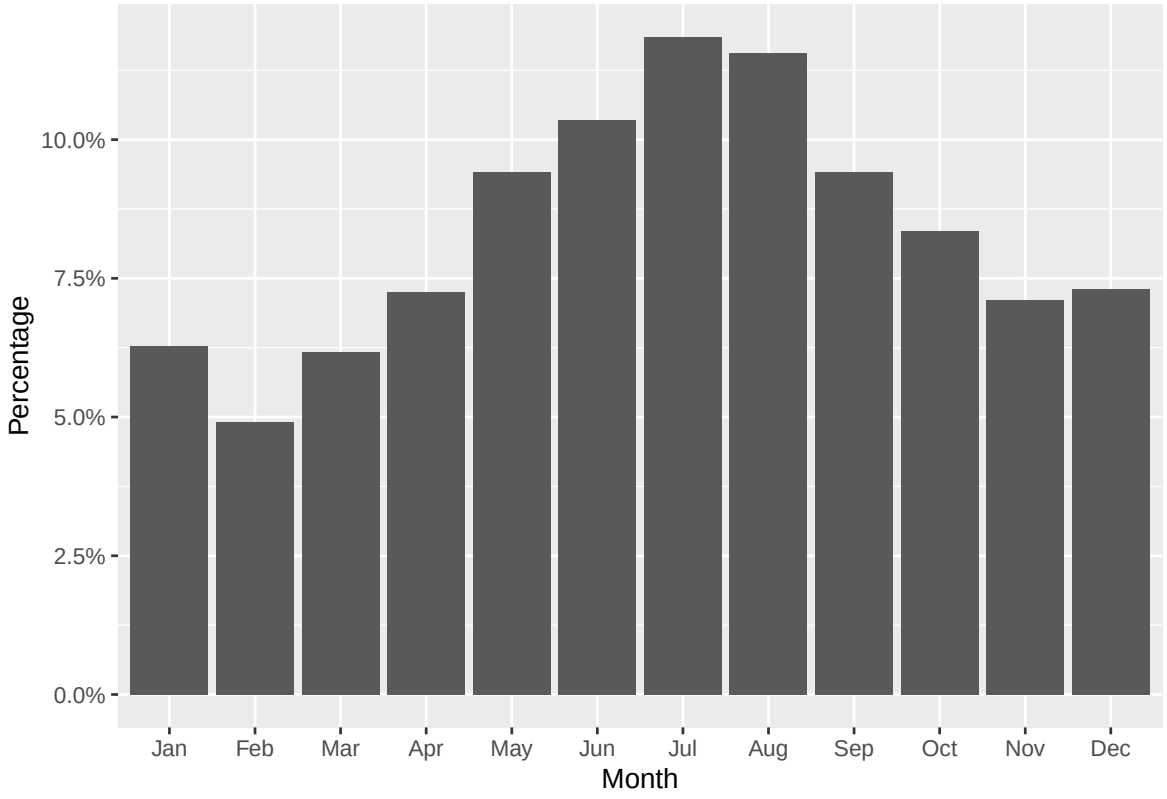
## 3. Analysis

Let's analyze the clean data set

1. Shootings by Month

   Observing shootings grouped by months suggests shooting incidents occur more frequently in the summer months of May to Oct and less frequently during winter months of Nov to Feb. Coincidentally, this correlates with average temperature of summer and winter seasons and available daylight. However since this requires additional data it is out of scope for this assignment.
   NOTE: In continuation of winter-summer correlation with shootings, I have not investigated whether the shootings increase from winter months (Jan-Feb) to summer months (May-Oct) and vice versa every year. I'm not treating this data set as a time series of shooting incidents.

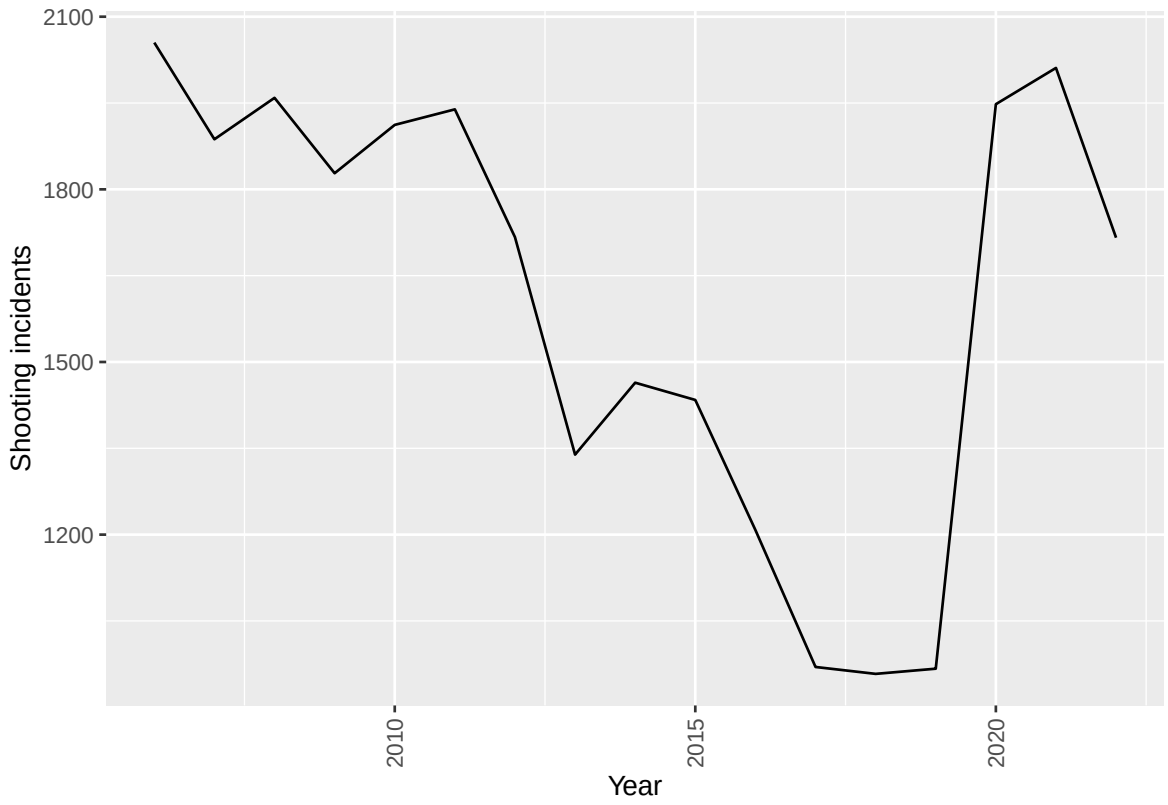
```
month_order = c('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                                    'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec')
ggplot(raw_data) + geom_bar(aes(x=factor(month_order[OCCUR_MONTH], levels=month_order), y = after_s
  scale_y_continuous(labels = scales::percent) +
  ylab("Percentage") + xlab("Month")
```




2. Yearly trend of shootings
   Yearly trend of shootings suggests, shootings incidents were declining over the years until 2020. This raises a question, does onset of global pandemic resulted in more shooting incidents? Does economic factors from pandemic led to increased shooting incidents?

```
shootings_by_year = raw_data |> group_by(OCCUR_YEAR) |> count(OCCUR_YEAR)
deaths_in_shooting = raw_data |> group_by(OCCUR_YEAR) |> count(STATISTICAL_MURDER_FLAG) |> filter(S
death_rate_df = left_join(shootings_by_year, deaths_in_shooting,
                          by = "OCCUR_YEAR")
death_rate_df$death_rate = death_rate_df$n.y*100/death_rate_df$n.x
```

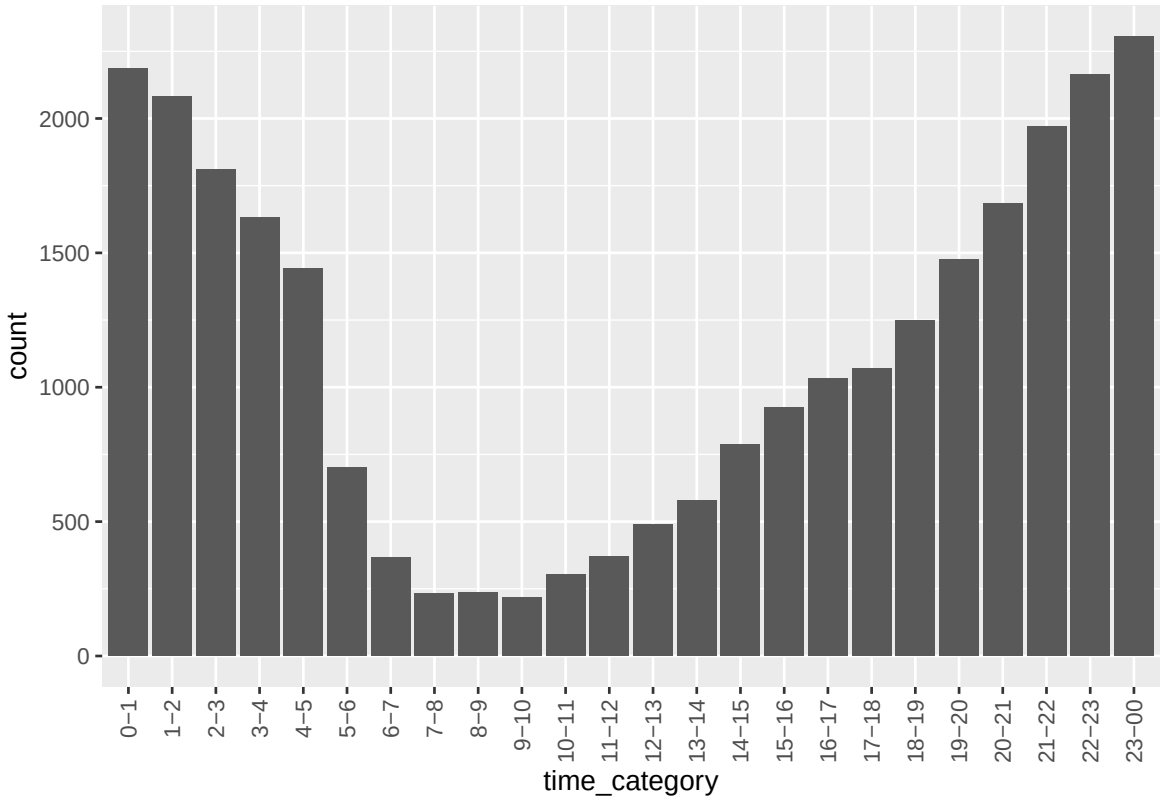
```
ggplot(shootings_by_year, aes(x=OCCUR_YEAR, y=n) ) + geom_line() +
  xlab("Year") + ylab("Shooting incidents") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




3. Time of the day (24hr format)

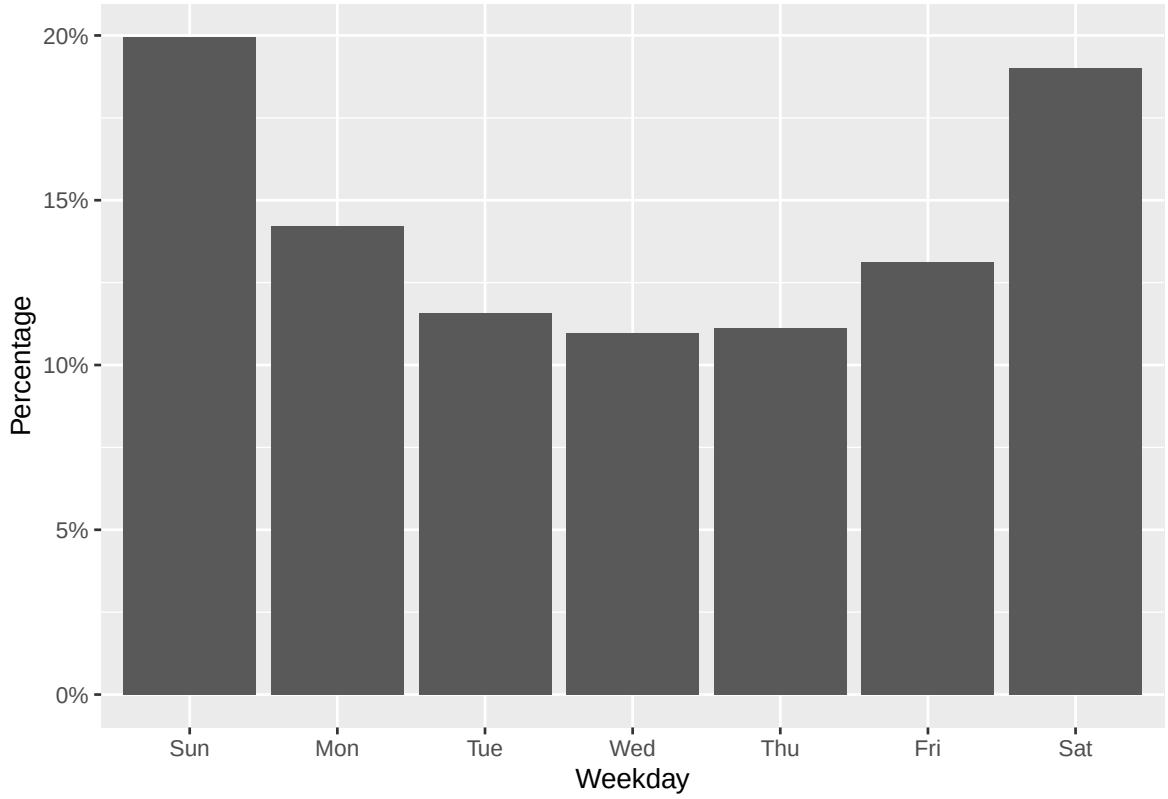
   Observing the incidents by time of the day suggests, these incidents are more frequent during late night hours than mornings.

```
raw_data$time_category <- sapply(as.numeric(raw_data$OCCUR_TIME, "hours"), get_time_category)
raw_data$time_category <- factor(raw_data$time_category, levels=time_category_order)
ggplot(raw_data) + geom_bar(aes(x=time_category)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



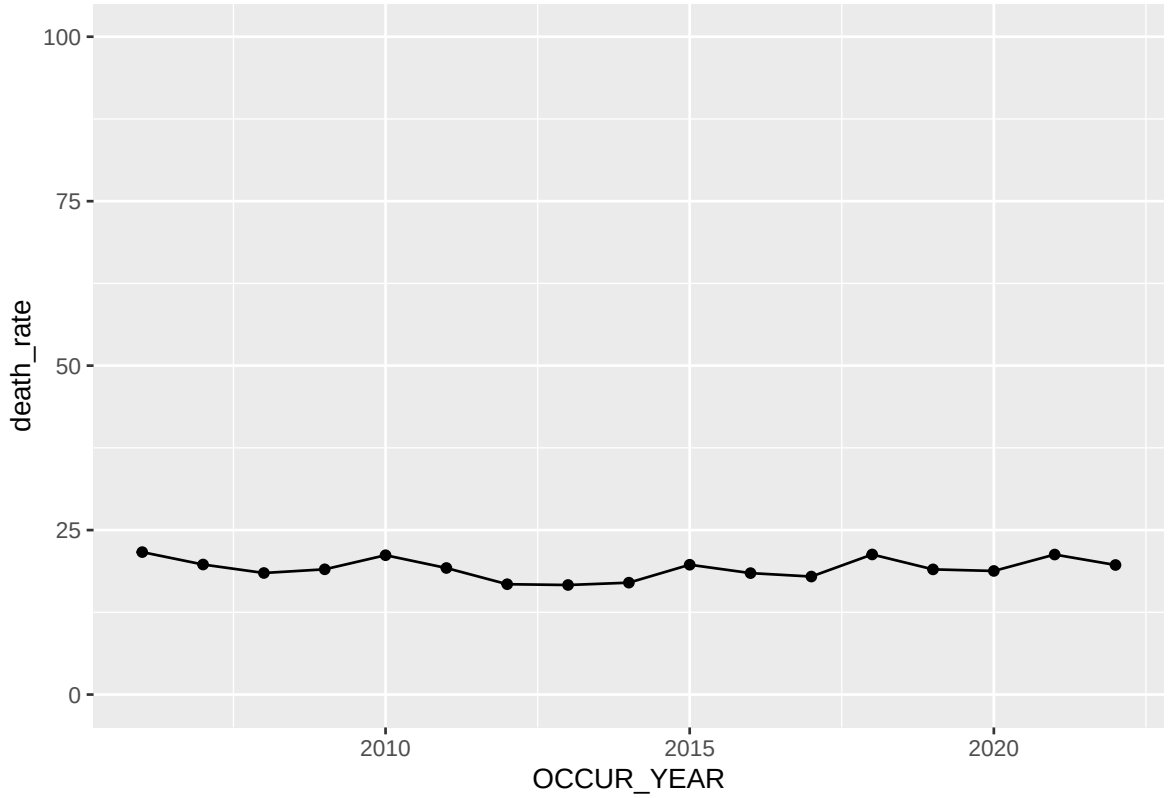

4. Weekday vs weekend
   Observing the incidents by day of the week suggests these incidents are more common during weekends than mid week days Tue, Wed, Thu. This could be linked to more social interaction during weekends than weekdays.

```
ggplot(raw_data) + geom_bar(aes(x=Weekday, y = after_stat(count/sum(count))))+
  scale_y_continuous(labels = scales::percent) +
  ylab("Percentage")
```




5. Death rate
   An interesting observations shows, even though shootings incidents were decreasing over the years, but the death rate in such incidents remained fairly consistent with mean 19.18%

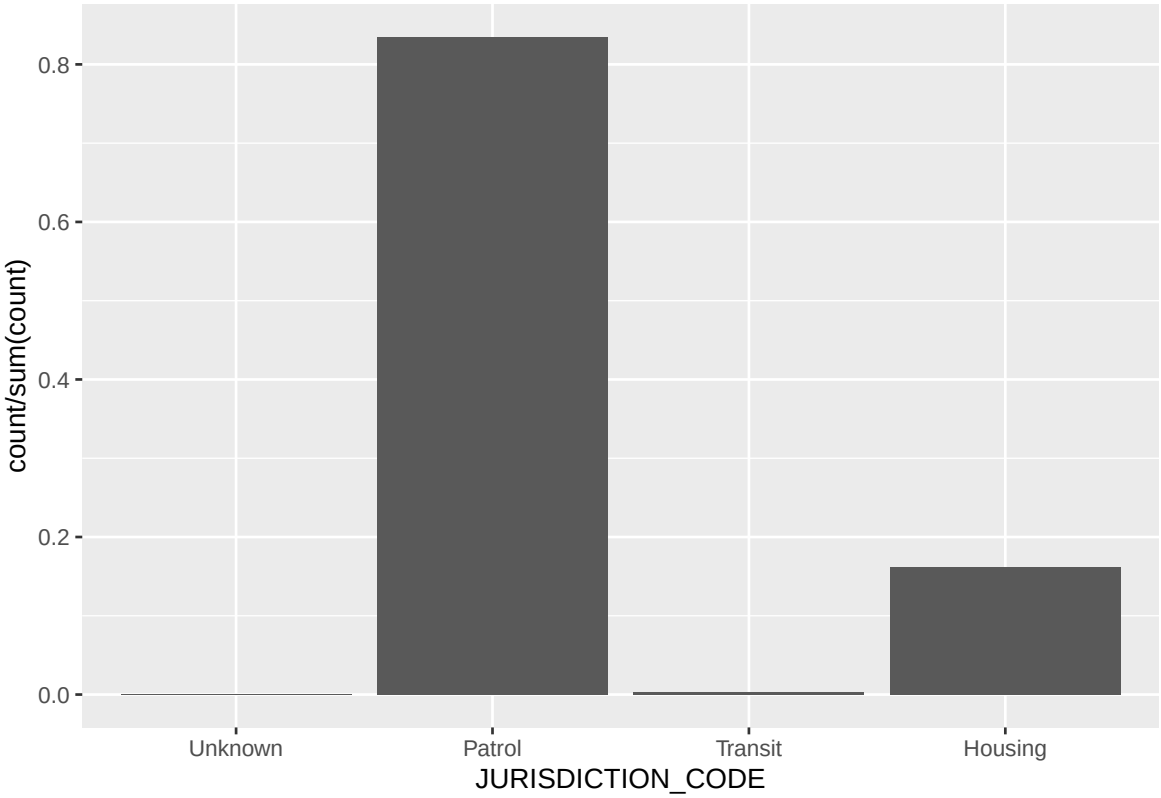
```
ggplot(death_rate_df, aes(x=OCCUR_YEAR, y=death_rate, group=1)) +
  geom_line() + geom_point() + ylim(0,100)
```




6. Jurisdiction
   Observing incidents by jurisdiction reveals patrol dept has most number of jurisdiction followed by housing dept.

```
raw_data$JURISDICTION_CODE <- factor(raw_data$JURISDICTION_CODE, levels = c(-1,0,1,2),
                                     labels = c("Unknown","Patrol", "Transit", "Housing"))

ggplot(raw_data) + geom_bar(aes(x=JURISDICTION_CODE, y=after_stat(count/sum(count))))
```



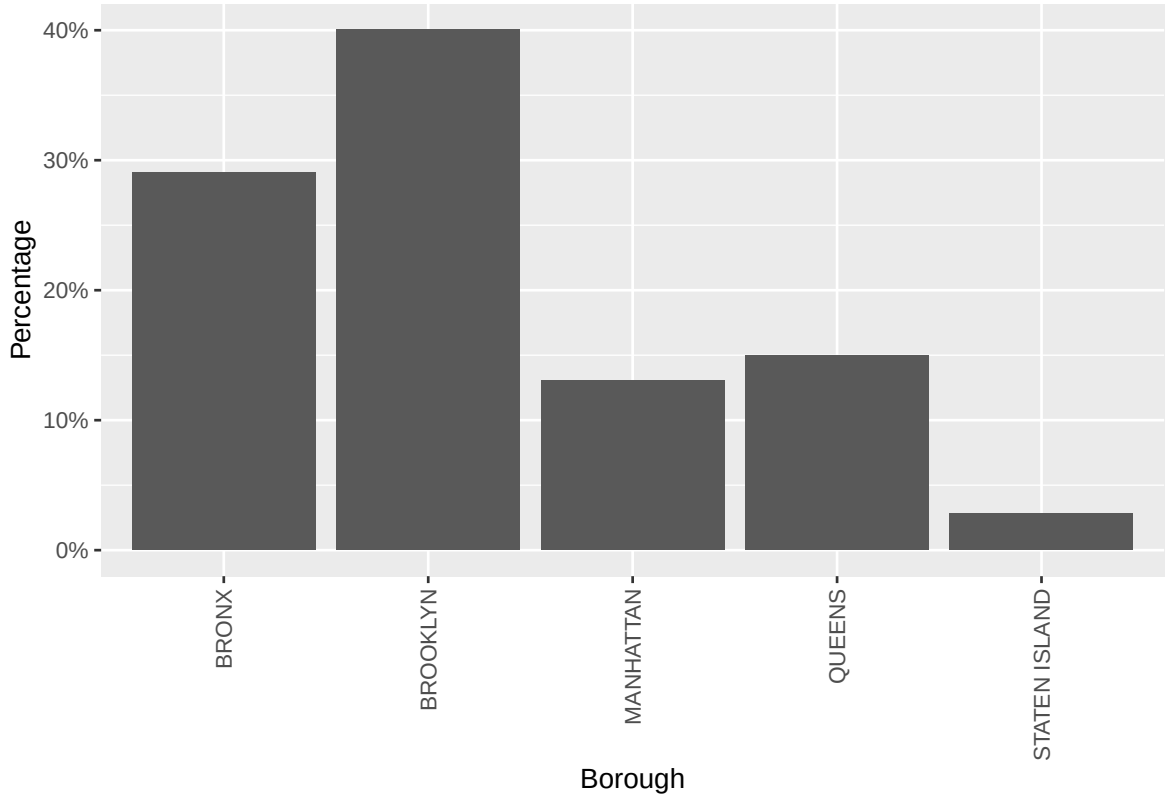

7. Locations of the shooting

   a. Distribution across borough

   Borough are administrative divisions of a city. This data set suggests Brooklyn reported highest shootings (~40%) while Staten Island reported the least ( ~2%)

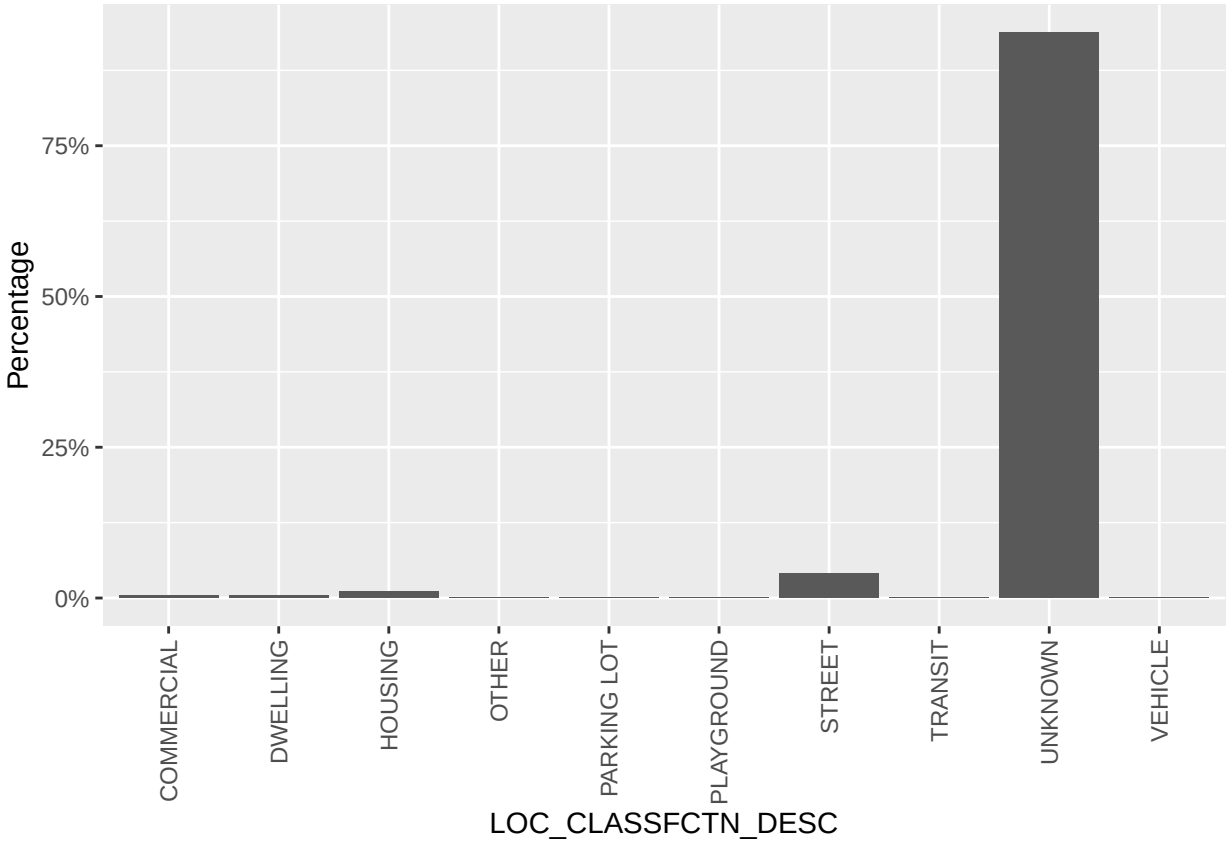
   **Comparing shooting incidents across borough as it is might be misleading, as it does not consider population of the borough. Ideally, shootings per 1000 or shootings per million should be looked at to compare boroughs.**

```
ggplot(raw_data) + geom_bar(aes(x=BORO, y=after_stat(count/sum(count)))) +
  scale_y_continuous(labels = scales::percent) +
  ylab("Percentage") + xlab("Borough") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



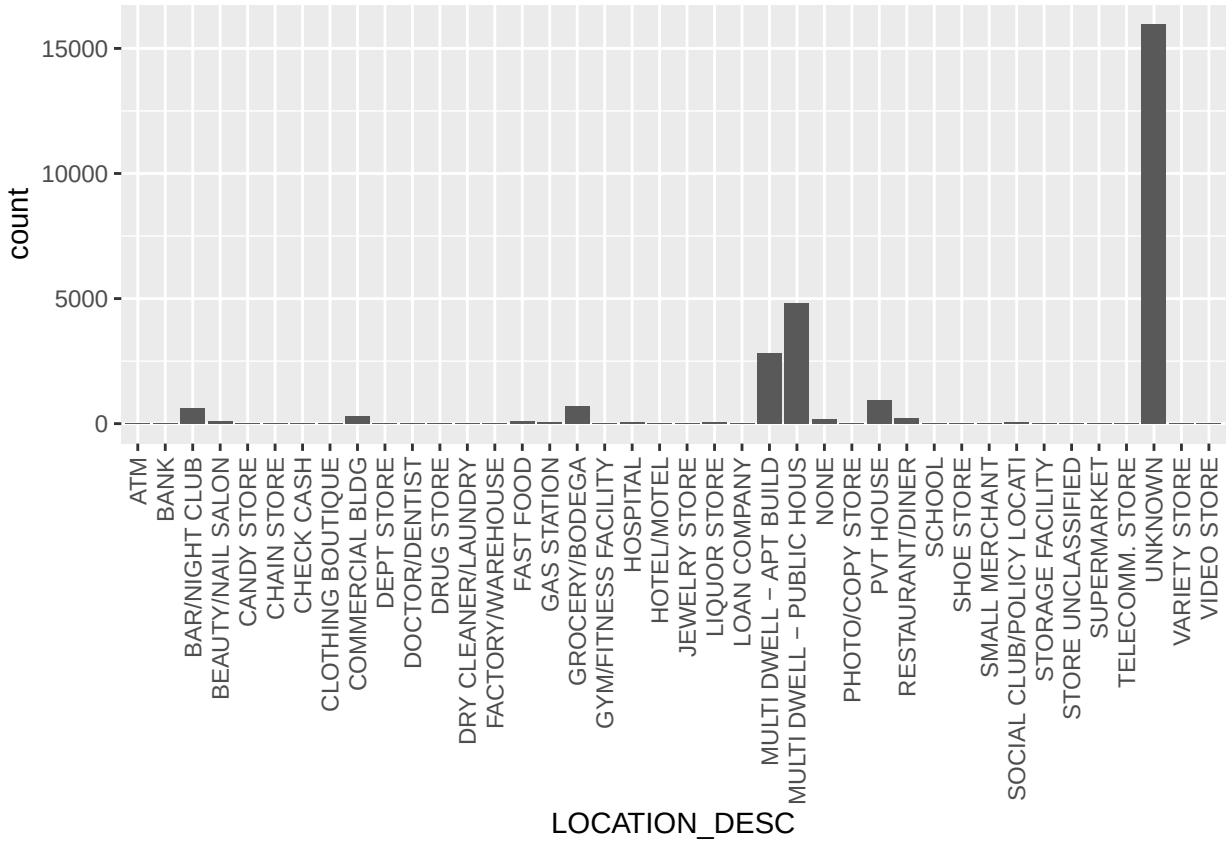

b. Location classification
More than 80% of shootings do not have location classification, thus not revealing much about the location of the shootings

```
ggplot(raw_data) +
  geom_bar(aes(x=LOC_CLASSFCTN_DESC, y = after_stat(count/sum(count))))+
  scale_y_continuous(labels = scales::percent) +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  ylab("Percentage")
```



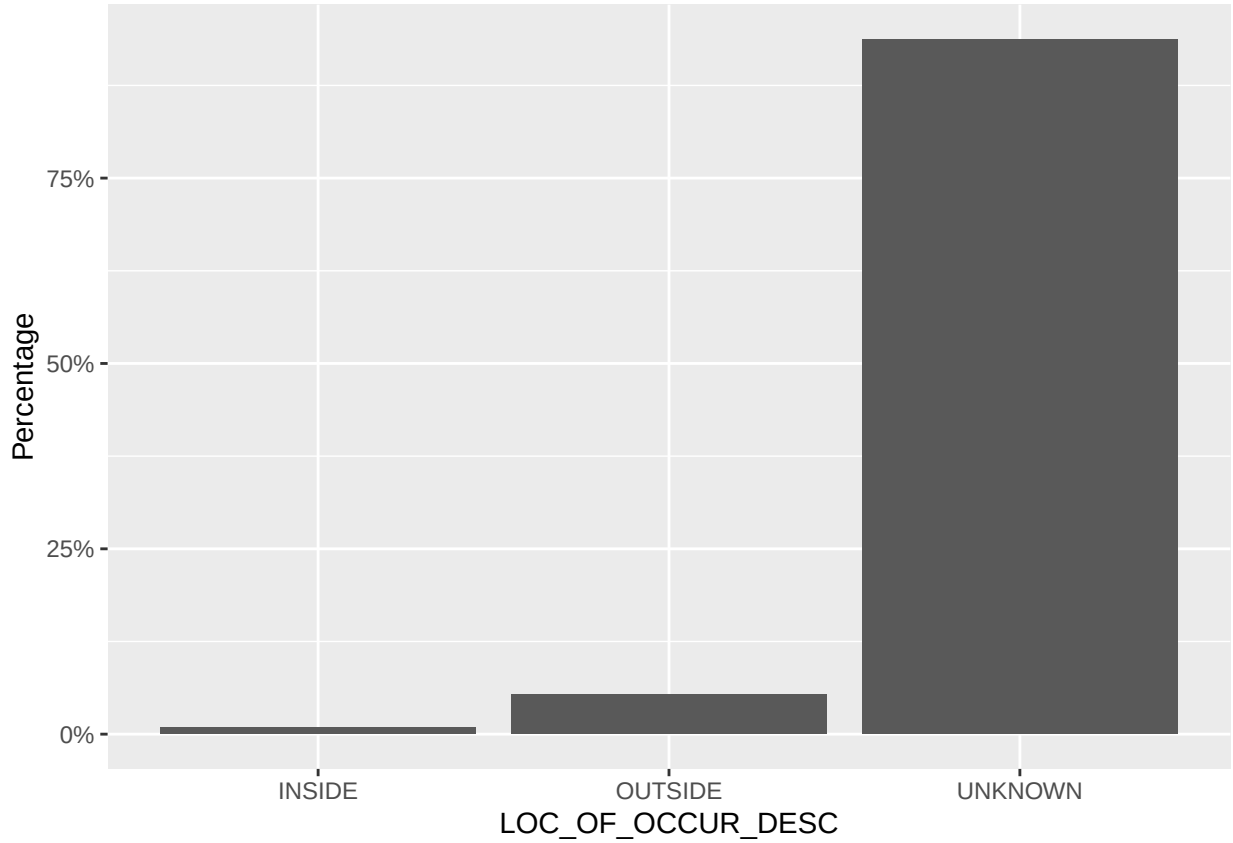

Similarly location description also has majority of Unknown.
However, within the known locations, shooting is common among multi dwell (apartment and public house). Knowing where most of the shootings occur may help law enforces to create better strategies like where most of the patrolling should happen

```
ggplot(raw_data) + geom_bar(aes(x=LOCATION_DESC)) + theme(axis.text.x = element_text(angle = 90, vjust =
```




c. Location (Inside or outside)
Another location classification (inside, outside) also have majority of unknown

```
ggplot(raw_data) +
  geom_bar(aes(x=LOC_OF_OCCUR_DESC, y = after_stat(count/sum(count))))+
  scale_y_continuous(labels = scales::percent) +
  ylab("Percentage")
```

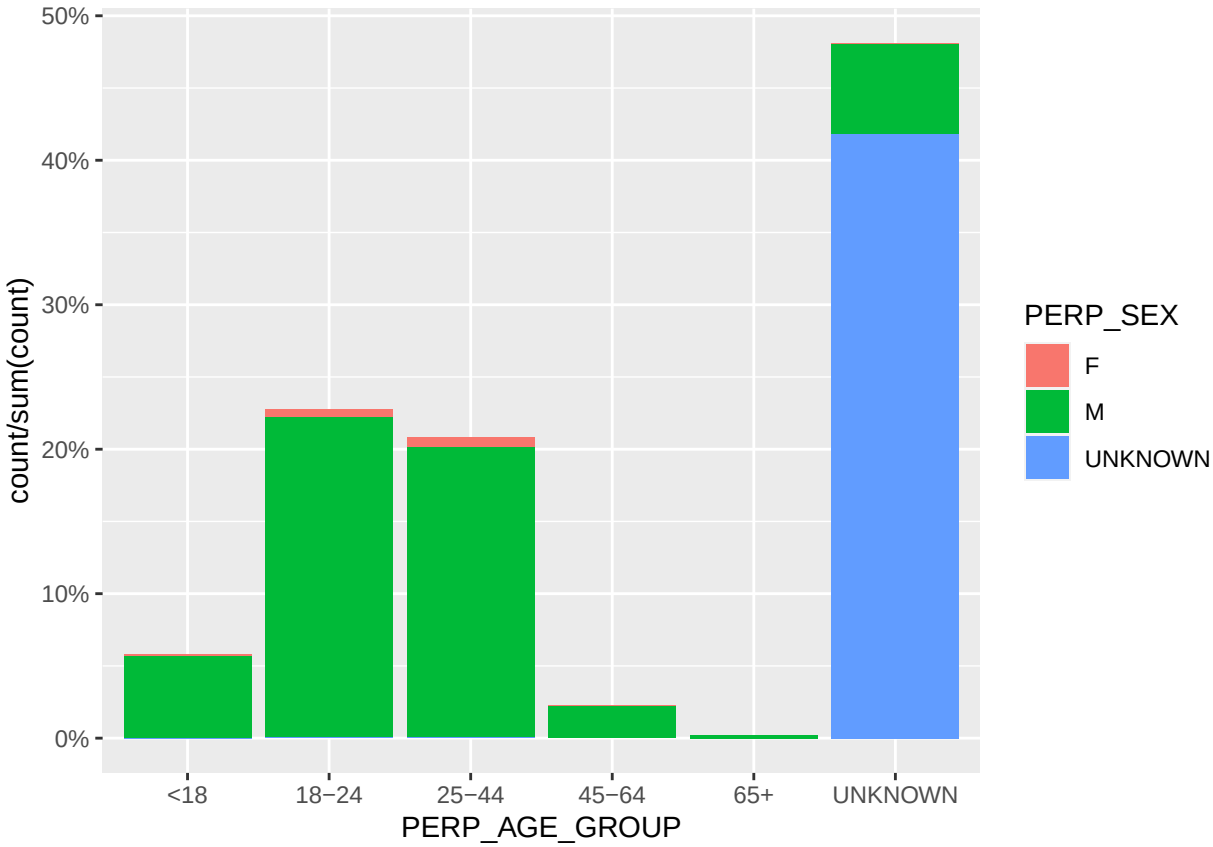



Since for location related columns have high percentage of Unknown, I will be excluding them

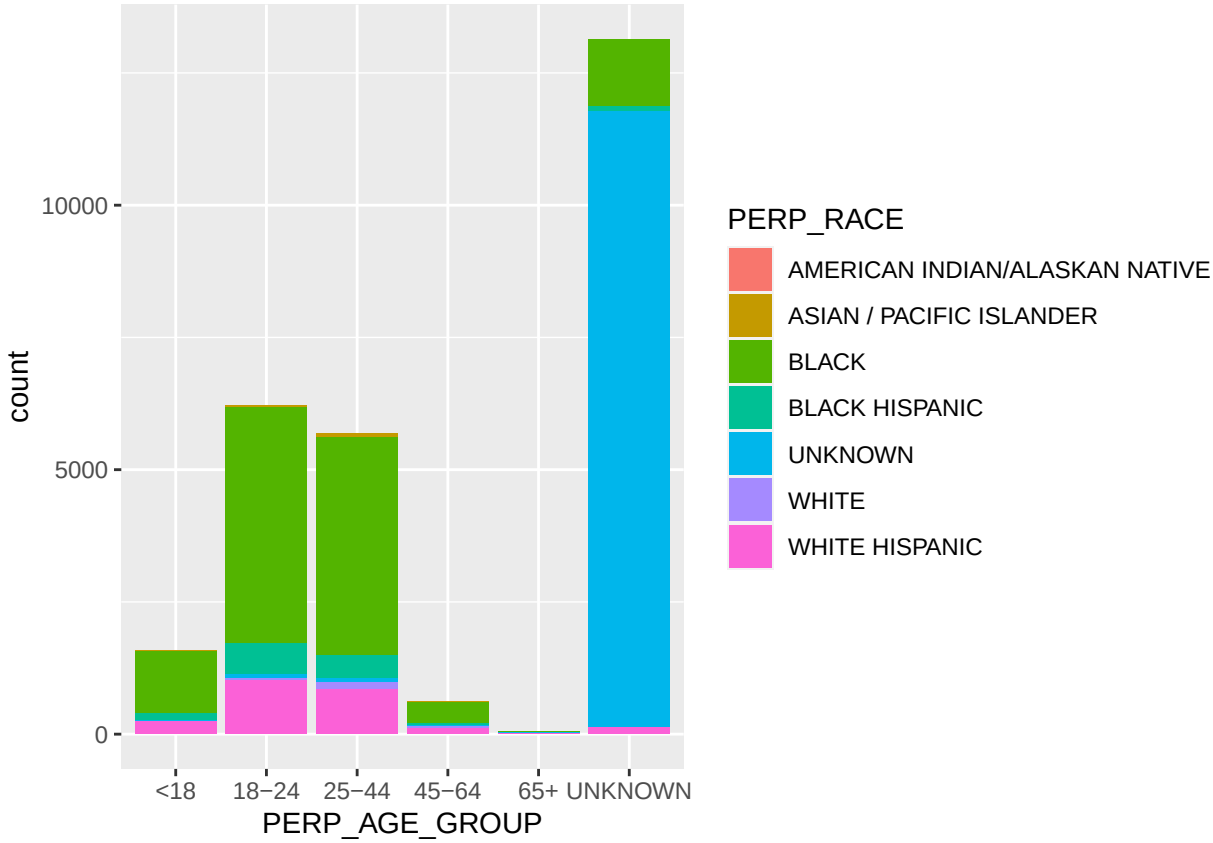
```
raw_data <- subset(raw_data, select = -c(LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION_DESC))
```

7. Perpetrator (Perp) profiling
   Profiling perpetrator by age, sex and race reveals
   - Majority of the known perps are from age group 18-24 and 25-44 (40% of known perps)
   - Gender of majority of the known perps is males across all age groups
   - Race of majority of the known perps is "Black" across all age groups

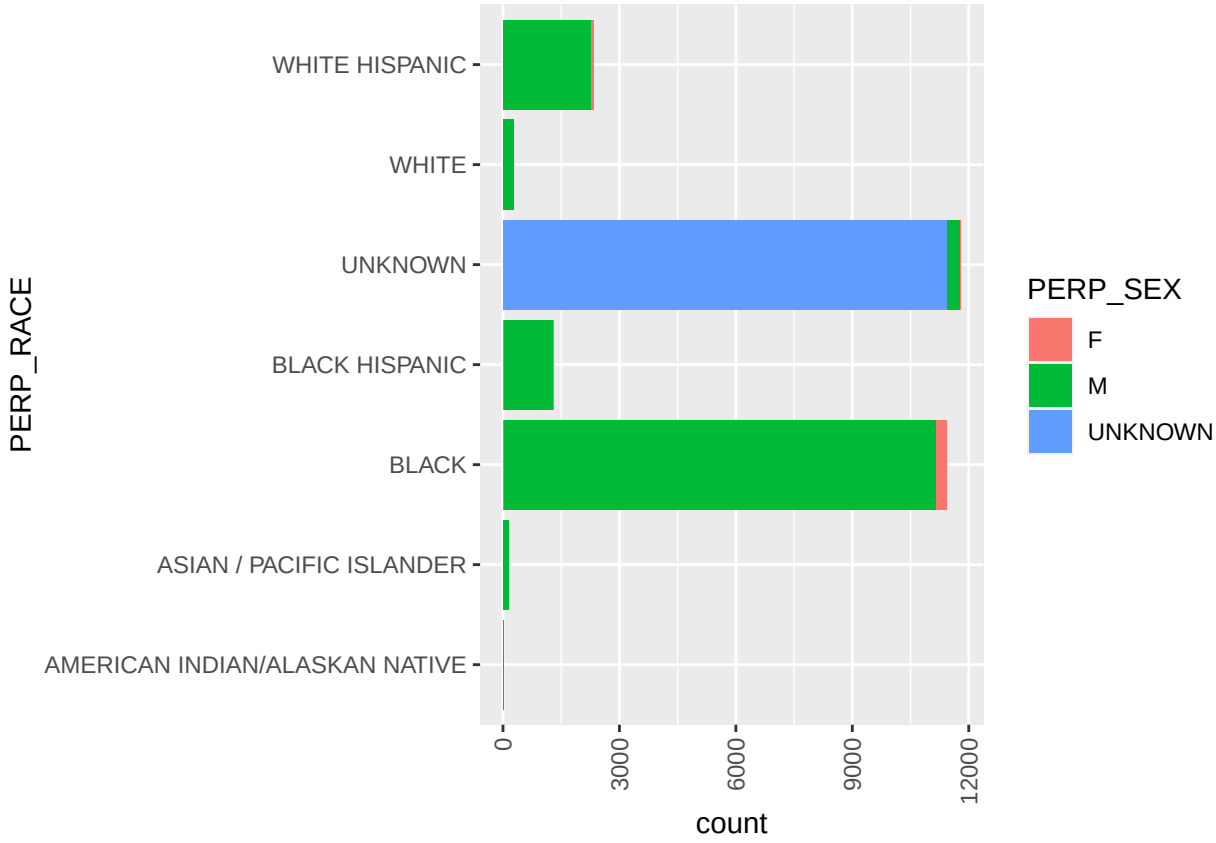
```
ggplot(raw_data) + geom_bar(aes(x=PERP_AGE_GROUP, y=after_stat(count/sum(count)), fill = PERP_SEX))+
  scale_y_continuous(labels = scales::percent)
```




```
ggplot(raw_data) + geom_bar(aes(x=PERP_AGE_GROUP, fill = PERP_RACE))
```



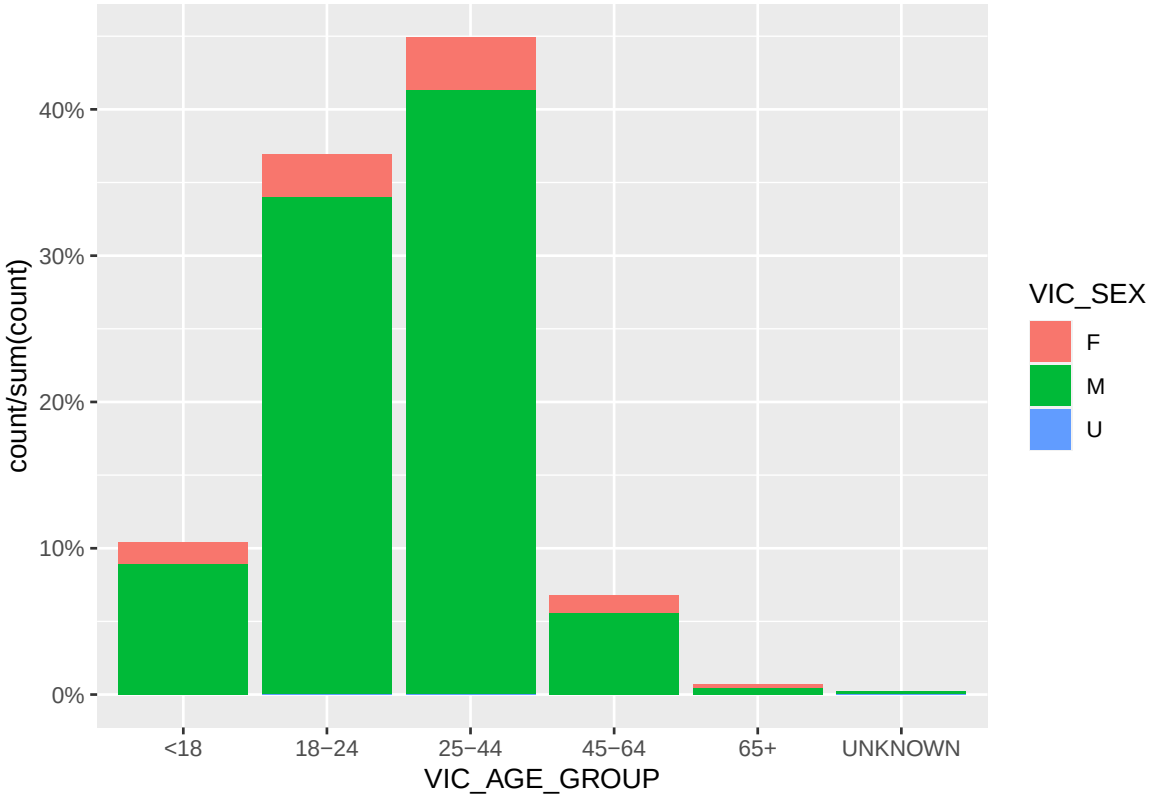

```
ggplot(raw_data) + geom_bar(aes(x=PERP_RACE, fill = PERP_SEX)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  coord_flip()
```



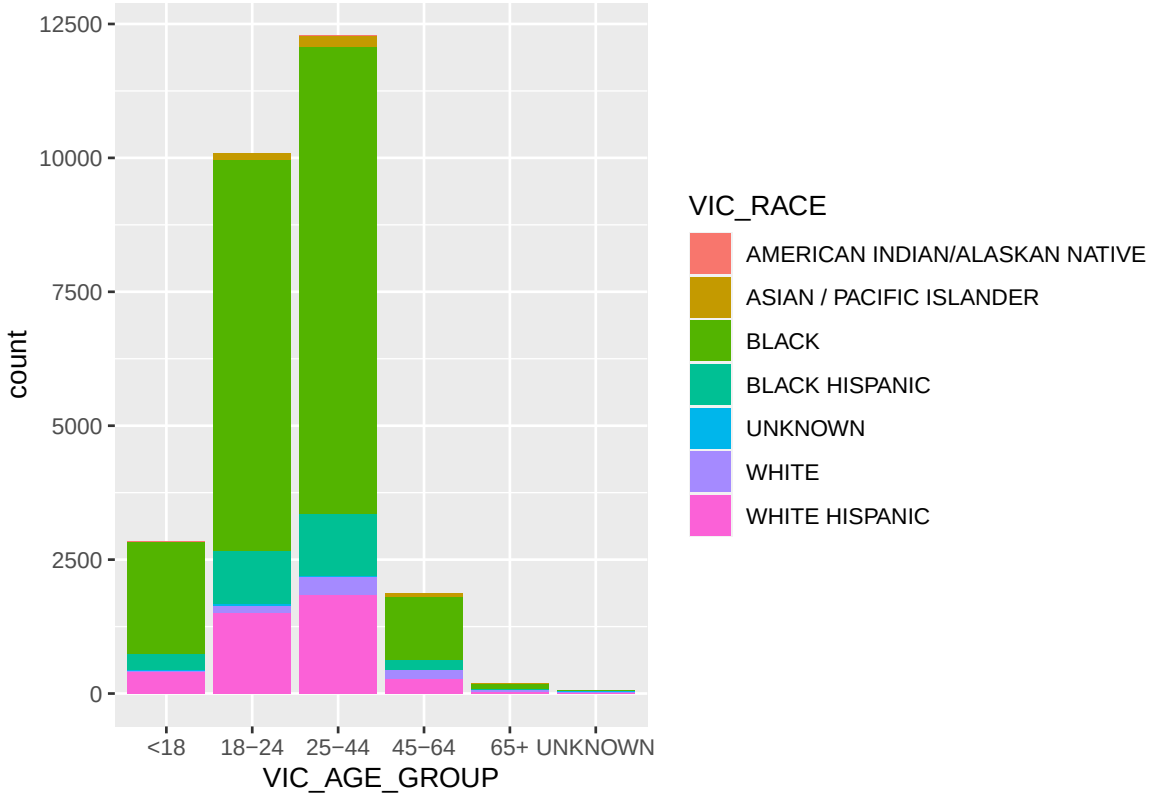

8. Victim profiling
   Profiling victims by age, sex and race reveals

   - There are more females as victims than perps across all age groups
   - Majority of victims lie in age group 25-44 (45%) followed by 18-24 (~37%)
   - There are significant ( ~10%) of minor victims
   - Race of majority of the victims is "Black" across all age groups

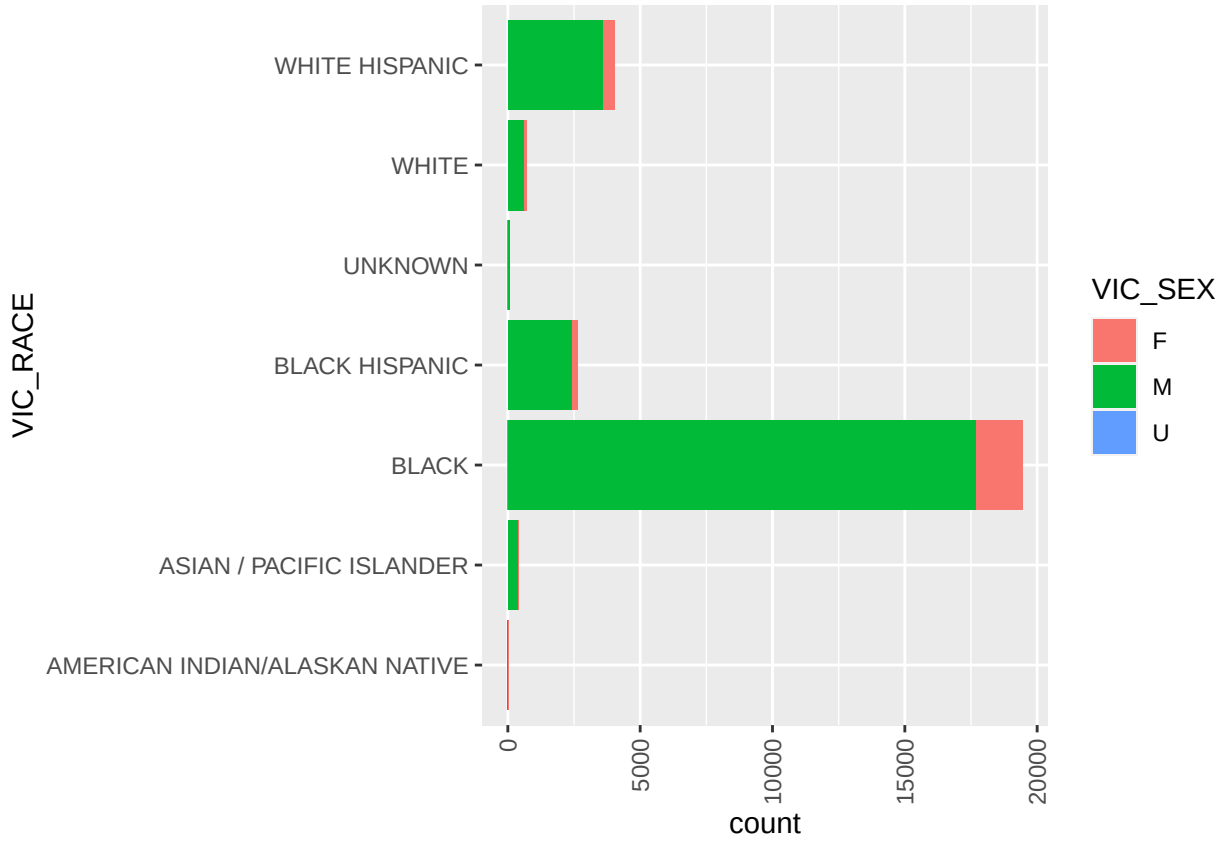
```
ggplot(raw_data) + geom_bar(aes(x=VIC_AGE_GROUP, y = after_stat(count/sum(count)),fill = VIC_SEX))+
    scale_y_continuous(labels = scales::percent)
```




```
ggplot(raw_data) + geom_bar(aes(x=VIC_AGE_GROUP, fill = VIC_RACE))
```



```
ggplot(raw_data) + geom_bar(aes(x=VIC_RACE, fill = VIC_SEX)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  coord_flip()
```




## 4. Modelling for shooting incidents

Seeing a pattern in time, month and location, i decided to predict shooting incidents given time, month and borough(location)

However, simply using predicting number of incidents might be of little use to NYPD.

Looking at incidents per month, I observed that 75% of data is below 3 ie 75% of time single month had 3 or less incidents. More than 3 shooting incidence in a month is unusual.

I decided to additionally provide probability of unusual shooting incident ie what are the chances that there will be more than 3 shooting incident.

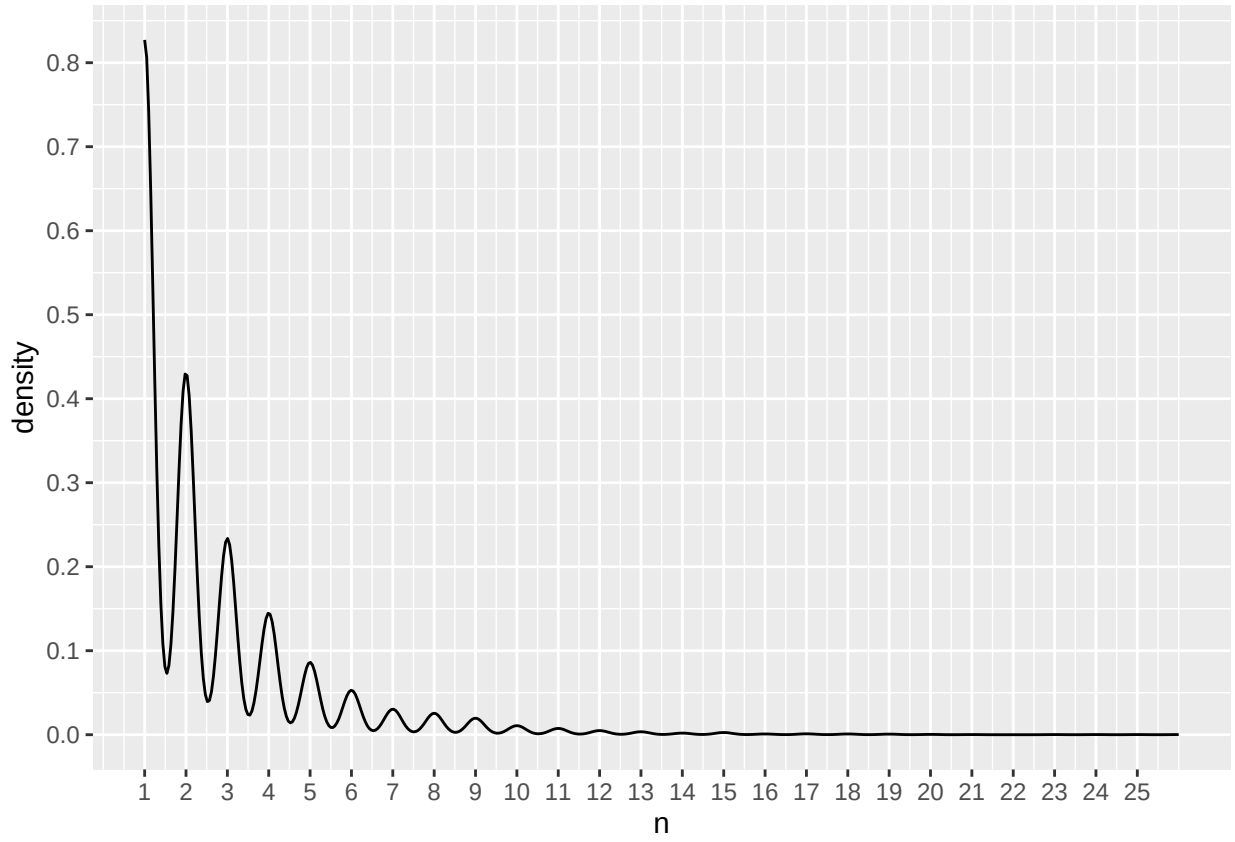
```
chance_greater_than_3 <- function(array){
  sum(array > 3)*100/800
  }

grouped_shootings_by_year = raw_data %>% group_by(BORO, OCCUR_MONTH, OCCUR_YEAR, time_category) %>% cou
grouped_shootings = subset(grouped_shootings_by_year, select = -c(OCCUR_YEAR))
```

```r
summary(grouped_shootings_by_year$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.527   3.000  26.000
```

```r
ggplot(grouped_shootings_by_year, aes(x=n)) +
  geom_density() + scale_x_continuous(breaks = seq(1, 25, by = 1)) + scale_y_continuous(breaks = seq(0,
```




Creating a random forest model. I chose tree based model as st

```r
sample <- sample(c(TRUE, FALSE), nrow(grouped_shootings), replace=TRUE, prob=c(0.65,0.35))
train  <- grouped_shootings[sample, ]
test   <- grouped_shootings[!sample, ]

##modelling shooting per month
set.seed(542)

rf_model <- randomForest(n ~ ., data=train, ntree=800,
                         keep.forest=TRUE, importance=TRUE, xtest= subset(test, select = -c(n)), ytest =
```

Finding an optimal threshold for when to consider unusual incident.
My expectation is to choose a threshold which minimizes false positive while have good true positives

Hence threshold of 65 is chosen, which gives ~80% accuracy, with ~70% of true positives and ~11% of false positive.

Generating confusion matrix for complete data set

```
# predictions
all_prediction_info = predict(rf_model, grouped_shootings, predict.all=TRUE)
grouped_shootings$predicted = all_prediction_info$aggregate
chance_list = list()
for(i in 1:10806){
  chance_list = append(chance_list, chance_greater_than_3(all_prediction_info[2]$individual[i,]))
}
grouped_shootings$chance_unusual = chance_list
```

```
# find optimal threshold
threshold_values = seq(50,95,5)
true_positive = c()
true_negative = c()
false_positive = c()
false_negative = c()
accuracy_list = c()

for (var in threshold_values){
  cm = confusionMatrix(as.factor(grouped_shootings$chance_unusual > var),
                 as.factor(grouped_shootings$n > 3))
  total_incidents = sum(cm$table)
  true_positive = append(true_positive, cm$table[1,1]/total_incidents)
  true_negative = append(true_negative, cm$table[2,2]/total_incidents)
  false_positive = append(false_positive, cm$table[1,2]/total_incidents)
  false_negative = append(false_negative, cm$table[2,1]/total_incidents)
  accuracy_list = append(accuracy_list, cm$overall[1])
}

data.frame(threshold_values,true_positive,true_negative,false_positive,false_negative,accuracy_list
)
```

```
##    threshold_values true_positive true_negative false_positive false_negative
## 1                50     0.6761059    0.10697761      0.1031834     0.11373311
## 2                55     0.6858227    0.10225800      0.1079030     0.10401629
## 3                60     0.6989635    0.09716824      0.1129928     0.09087544
## 4                65     0.7028503    0.09522488      0.1149361     0.08698871
## 5                70     0.7125671    0.08939478      0.1207662     0.07727189
## 6                75     0.7258930    0.07967796      0.1304831     0.06394596
## 7                80     0.7369054    0.07264483      0.1375162     0.05293356
## 8                85     0.7470850    0.06542661      0.1447344     0.04275403
## 9                90     0.7590228    0.05450676      0.1556543     0.03081621
## 10               95     0.7773459    0.03044605      0.1797150     0.01249306
##    accuracy_list
## 1      0.7830835
## 2      0.7880807
## 3      0.7961318
## 4      0.7980751
## 5      0.8019619
## 6      0.8055710
## 7      0.8095502
## 8      0.8125116
## 9      0.8135295
```

```
## 10     0.8077920
```

```r
#with optimal threshold
optimal_threshold = 65
confusionMatrix(as.factor(grouped_shootings$chance_unusual > optimal_threshold),
                as.factor(grouped_shootings$n > 3), positive = "TRUE")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  7595 1242
##      TRUE    940 1029
##
##                Accuracy : 0.7981
##                  95% CI : (0.7904, 0.8056)
##     No Information Rate : 0.7898
##     P-Value [Acc > NIR] : 0.01798
##
##                   Kappa : 0.3606
##
##  Mcnemar's Test P-Value : 1.166e-10
##
##             Sensitivity : 0.45310
##             Specificity : 0.88987
##          Pos Pred Value : 0.52260
##          Neg Pred Value : 0.85945
##              Prevalence : 0.21016
##          Detection Rate : 0.09522
##    Detection Prevalence : 0.18221
##       Balanced Accuracy : 0.67148
##
##        'Positive' Class : TRUE
##
```

## 5. Bias

Because of stereotype, it is assumed black males are involved more in criminal and violence activities. This could become my person bias as it could potentially turn into a confirmation bias, where I will look for supporting evidence for this prior belief.
In order to handle it, I should set aside my prejudice, and try to look for contradictory evidence. Additionally, I could mask the race information, encode it and reveal it after completing the analysis. However since I didn't use race information for modelling, this is not a concern.

Algorithmic bias
Another form of bias that could arise from this data set is algorithmic bias. Current model uses borough, time of day and month as input. Since Bronx and Brooklyn see more number of incidents, the resulting model will be biased toward these two boroughs
In order to mitigate this, re-sampling of data, specifically under sampling, could be used to reduce the imbalance of data. An under sampled data set can be created with population represented from all races equally.

## 5. Conclusion

This data report looks at location, timing of shooting in terms of month, year, time of the day, day of the week, jurisdiction and profile of vicitms and perps (age, sex and race).

This report could be a first step on understanding the nature of the shootings and eventually leading to measures to reduce such incidents. As part of the understanding, few questions got raised like

- Is there any correlation between daylight or temperature with shootings?
- Why did the shootings increased during 2020? Can it be linked to economic downturn?
- What could be nature of these shootings? Could it be personal revenge, inter racial conflict, domestic violence etc?

A probability of unusual shooting incidents (ie $> 3$) can be provided to NYPD which can help them be more proactive and alert.