Final_Project_NYPD_Shootings

V_MSDS_LAB

2022-08-01

```r
# Make sure these are installed:

# install.packages("tidyverse")
# install.packages("lubridate")
# install.packages("geosphere")
# install.packages("caTools")
# install.packages("randomForest")
# install.packages("class")

library(tidyverse)
library(lubridate)
library(geosphere)
library(caTools)
library(randomForest)
library(class)
```

About the Data:

-

  Introduction:

  - Our our initial or primary data set contains attributes surrounding every shooting incident in New York City since the year 2006 and all the way through until the end of the prior calendar year. This data is reviewed and updated every year by the Office of Management Analysis and Planning. In general, the information includes demographics surrounding the suspect as well as the victim, location and time of incident occurance, , as well as the label - which is a binary representation for whether the incident proved to be fatal for the victim.
  - Our supporting data sets include additional information surround the police precincts such as the shape of the precinct in terms of area and length. Additionally, we will use a data set with supporting information about nearby hospitals. This data includes attributes such as facility type, borough, longitude and latitude. For our purposes, the attributes we are interested in are; [Precinct, Shape_Leng, Shape_Area, Facility Type, Borough, Latitude, Longitude]. We will be merging the data sets on Precinct and Borough respectively. We will be using hospital Latitude and Longitude ('H_Latitude' / 'H_Longitude') for distance calculation relative to incident location.

-

Data Attributes (Combined):

- Independent Variables / Features:
  - INCIDENT_KEY
  - OCCUR_DATE
  - OCCUR_TIME
  - BORO
  - PRECINCT
  - JURISDICTION_CODE
  - LOCATION_DESC
  - PERP_AGE_GROUP
  - PERP_SEX
  - PERP_RACE
  - VIC_AGE_GROUP
  - VIC_SEX
  - VIC_RACE
  - X_COORD_CD
  - Y_COORD_CD
  - Latitude
  - Longitude
  - Shape_Area
  - Shape_Length
  - H_Latitude
  - H_Longitude
- Dependent / Target Variable:
  - STATISTICAL_MURDER_FLAG
- Columns we are interested in:
  - raw_dat1: [INCIDENT_KEY,OCCUR_DATE,OCCUR_TIME,BORO,PRECINCT,LOCATION_DESC,VI
  - raw_dat2: [Precinct,Shape_Area,Shape_Length]
  - raw_dat3: [Facility Type,Borough,Latitude,Longitude]

Importing Data:

First, we are going to import all the required data. Our primary data set is "raw_dat1", and supporting data is in "raw_dat2" & "raw_dat1".

```
rd1 <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
raw_dat1 <- read_csv(rd1,col_select = c(INCIDENT_KEY,OCCUR_DATE,OCCUR_TIME,BORO
                                        ,PRECINCT,LOCATION_DESC,VIC_AGE_GROUP
                                        ,VIC_SEX,VIC_RACE,Latitude,Longitude
                                        ,STATISTICAL_MURDER_FLAG))

rd2 <- "https://data.cityofnewyork.us/api/views/kmub-vria/rows.csv?accessType=DOWNLOAD"
raw_dat2 <- read_csv(rd2,col_select = c(Precinct,Shape_Area,Shape_Leng))
```

```r
rd3 <- "https://data.cityofnewyork.us/api/views/ymhw-9cz9/rows.csv?accessType=DOWNLOAD"
raw_dat3 <- read_csv(rd3,col_select = c('Facility Type',Borough,Latitude,Longitude))
```

```r
# Lets take a quick look at our data sets.

head(raw_dat1,5)
```

```
## # A tibble: 5 x 12
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO   PRECINCT LOCATION_DESC VIC_AGE_GROUP
##          <dbl> <chr>      <time>     <chr>     <dbl> <chr>         <chr>
## 1     24050482 08/27/2006 05:35      BRONX        52 <NA>          25-44
## 2     77673979 03/11/2011 12:03      QUEENS      106 <NA>          65+
## 3    226950018 04/14/2021 21:08      BRONX        42 COMMERCIAL B~ 18-24
## 4    237710987 12/10/2021 19:30      BRONX        52 <NA>          25-44
## 5    224701998 02/22/2021 00:18      MANHA~       34 <NA>          25-44
## # ... with 5 more variables: VIC_SEX <chr>, VIC_RACE <chr>, Latitude <dbl>,
## #   Longitude <dbl>, STATISTICAL_MURDER_FLAG <lgl>
```

```r
head(raw_dat2,5)
```

```
## # A tibble: 5 x 3
##   Precinct Shape_Area Shape_Leng
##      <dbl>      <dbl>      <dbl>
## 1        1  47300568.     81118.
## 2        5  18094527.     18807.
## 3        6  22103327.     26413.
## 4        7  18365996.     17288.
## 5        9  21395839.     19773.
```

```r
head(raw_dat3,5)
```

```
## # A tibble: 5 x 4
##   `Facility Type`     Borough   Latitude Longitude
##   <chr>               <chr>        <dbl>     <dbl>
## 1 Child Health Center Manhattan     NA        NA
## 2 Acute Care Hospital Queens        NA        NA
## 3 Child Health Center Brooklyn      40.6     -74.0
## 4 Child Health Center Queens        40.7     -73.8
## 5 Child Health Center Bronx         NA        NA
```

Cleaning Data:

There are a few adjustments we need to make here:

[raw_dat1]

1. Adjust raw_dat1.OCCUR_DATE from character to a date.
2. Add columns hor hour and year derived form the OCCUR_DATE
3. Rename and convert STATISTICAL_MURDER_FLAG(TRUE/FALSE) to TARGET(1/0)
4. Convert TARGET to a factor type.

5. Create binary representation columns for categorical variables in VIC_AGE_GROUP,VIC_SEX, VIC_RACE. For LOCATION_DESC we will create columns for only the following: PVT_HOUSE, HOTEL_MOTEL, MULTI_PUB_HOU, MULTI_APT, BAR_CLUB.
6. Convert all categorical variables to factor type.

[raw_dat2]

1. Rename columns to: c(PRECINCT, AREA, LENGTH)
2. Merge with raw_dat1 on 'PRECINCT'

[raw_dat3]

1. Drop all rows with missing Latitude.
2. Filter for Acute Care - Facility Type.
3. Rename columns to: 'fType','BORO','H_Latitude','H_Longitude'.
4. Replacing two Acute Care hospitals manually, due to missing Lat/Long on the original data file.
5. Convert boroughs to upper-case to match original raw_dat1 for merge.
6. Check for and remove any duplicate rows.
7. Merge raw_dat3 with the newly created file in prior merge.

```r
raw_dat1 <- raw_dat1 %>%
  mutate(OCCUR_DATE = lubridate::mdy(OCCUR_DATE),
         YEAR = year(OCCUR_DATE),
         HOUR = hour(OCCUR_TIME),
         TARGET = if_else(STATISTICAL_MURDER_FLAG==TRUE,1,0),
         VIC_SEX_M = if_else(VIC_SEX=='M',1,0),
         PVT_HOUSE = if_else(LOCATION_DESC=='PVT_HOUSE',1,0),
         HOTEL_MOTEL = if_else(LOCATION_DESC=='HOTEL/MOTEL',1,0),
         MULTI_PUB_HOU = if_else(LOCATION_DESC=='MULTI DWELL - PUBLIC HOUS',1,0),
         MULTI_APT = if_else(LOCATION_DESC=='MULTI DWELL - APT',1,0),
         BAR_CLUB = if_else(LOCATION_DESC=='BAR/NIGHT CLUB',1,0),
         VC_AGE_18 = if_else(VIC_AGE_GROUP=="<18",1,0),
         VC_AGE_18_24 = if_else(VIC_AGE_GROUP=="18-24",1,0),
         VC_AGE_25_44 = if_else(VIC_AGE_GROUP=='25_44',1,0),
         VC_AGE_65 = if_else(VIC_AGE_GROUP=='65+',1,0))

raw_dat1$MULTI_APT[is.na(raw_dat1$MULTI_APT)] <- 0
raw_dat1$PVT_HOUSE[is.na(raw_dat1$PVT_HOUSE)] <- 0
raw_dat1$HOTEL_MOTEL[is.na(raw_dat1$HOTEL_MOTEL)] <- 0
raw_dat1$MULTI_PUB_HOU[is.na(raw_dat1$MULTI_PUB_HOU)] <- 0
raw_dat1$BAR_CLUB[is.na(raw_dat1$BAR_CLUB)] <- 0

clean_dat1 <- raw_dat1 %>%
  select(TARGET,INCIDENT_KEY,PRECINCT,OCCUR_DATE,YEAR,HOUR,BORO,Latitude,Longitude
         ,VIC_SEX_M,PVT_HOUSE,HOTEL_MOTEL,MULTI_PUB_HOU,MULTI_APT,BAR_CLUB
         ,VC_AGE_18,VC_AGE_18_24,VC_AGE_25_44,VC_AGE_65)
```

```r
# Lets take a look at our cleaned primary data set.

head(clean_dat1,5)
```

```
## # A tibble: 5 x 19
##   TARGET INCIDENT_KEY PRECINCT OCCUR_DATE  YEAR  HOUR BORO    Latitude Longitude
##    <dbl>        <dbl>    <dbl> <date>     <dbl> <int> <chr>      <dbl>     <dbl>
## 1      1     24050482       52 2006-08-27  2006     5 BRONX       40.9     -73.9
## 2      0     77673979      106 2011-03-11  2011    12 QUEENS      40.7     -73.8
## 3      1    226950018       42 2021-04-14  2021    21 BRONX       40.8     -73.9
## 4      0    237710987       52 2021-12-10  2021    19 BRONX       40.9     -73.9
## 5      0    224701998       34 2021-02-22  2021     0 MANHAT~     40.9     -73.9
## # ... with 10 more variables: VIC_SEX_M <dbl>, PVT_HOUSE <dbl>,
## #   HOTEL_MOTEL <dbl>, MULTI_PUB_HOU <dbl>, MULTI_APT <dbl>, BAR_CLUB <dbl>,
## #   VC_AGE_18 <dbl>, VC_AGE_18_24 <dbl>, VC_AGE_25_44 <dbl>, VC_AGE_65 <dbl>
```

```
colnames(raw_dat2) = c('PRECINCT', 'AREA', 'LENGTH')

clean_dat2 <- left_join(clean_dat1,raw_dat2,by='PRECINCT')
```

```
head(clean_dat2,5)
```

```
## # A tibble: 5 x 21
##   TARGET INCIDENT_KEY PRECINCT OCCUR_DATE  YEAR  HOUR BORO    Latitude Longitude
##    <dbl>        <dbl>    <dbl> <date>     <dbl> <int> <chr>      <dbl>     <dbl>
## 1      1     24050482       52 2006-08-27  2006     5 BRONX       40.9     -73.9
## 2      0     77673979      106 2011-03-11  2011    12 QUEENS      40.7     -73.8
## 3      1    226950018       42 2021-04-14  2021    21 BRONX       40.8     -73.9
## 4      0    237710987       52 2021-12-10  2021    19 BRONX       40.9     -73.9
## 5      0    224701998       34 2021-02-22  2021     0 MANHAT~     40.9     -73.9
## # ... with 12 more variables: VIC_SEX_M <dbl>, PVT_HOUSE <dbl>,
## #   HOTEL_MOTEL <dbl>, MULTI_PUB_HOU <dbl>, MULTI_APT <dbl>, BAR_CLUB <dbl>,
## #   VC_AGE_18 <dbl>, VC_AGE_18_24 <dbl>, VC_AGE_25_44 <dbl>, VC_AGE_65 <dbl>,
## #   AREA <dbl>, LENGTH <dbl>
```

```
raw_dat3 <- raw_dat3  %>%
  select('Facility Type','Borough','Latitude','Longitude') %>%
  drop_na('Latitude')
head(raw_dat3,5)
```

```
## # A tibble: 5 x 4
##   `Facility Type`     Borough  Latitude Longitude
##   <chr>               <chr>       <dbl>     <dbl>
## 1 Child Health Center Brooklyn     40.6     -74.0
## 2 Child Health Center Queens       40.7     -73.8
## 3 Child Health Center Queens       40.7     -73.8
## 4 Child Health Center Queens       40.7     -73.8
## 5 Child Health Center Queens       40.8     -73.9
```

```
# Filtering for Acute Care - Facility Type
filt1 <- raw_dat3$`Facility Type`=='Acute Care Hospital'
raw_dat3 <- raw_dat3[filt1,]

# Renaming for ease of use.
colnames(raw_dat3) <- c('fType','BORO','H_Latitude','H_Longitude')

# Adding in two Acute Care Hospitals manually, due to Lat/Long issue.
```

```
raw_dat3 <- raw_dat3 %>% add_row(fType='Acute Care Hospital'
                                 ,BORO='Bronx'
                                 ,H_Latitude=40.817688484049
                                 ,H_Longitude=-73.924200271483)

raw_dat3  <- raw_dat3 %>% add_row(fType='Acute Care Hospital'
                                  ,BORO='Queens'
                                  ,H_Latitude=40.738710402563
                                  ,H_Longitude=--73.878351155182)
raw_dat3$BORO <- toupper(raw_dat3$BORO)
# Making sure that we did not create duplicates.
raw_dat3 <- distinct(raw_dat3)
```

```
# Merge into final data set.
clean_dat3 <- left_join(clean_dat2,raw_dat3,by='BORO')
```

```
head(clean_dat3,15)
```

```
## # A tibble: 15 x 24
##    TARGET INCIDENT_KEY PRECINCT OCCUR_DATE  YEAR  HOUR BORO   Latitude Longitude
##     <dbl>        <dbl>    <dbl> <date>     <dbl> <int> <chr>     <dbl>     <dbl>
##  1      1     24050482       52 2006-08-27  2006     5 BRONX      40.9     -73.9
##  2      1     24050482       52 2006-08-27  2006     5 BRONX      40.9     -73.9
##  3      1     24050482       52 2006-08-27  2006     5 BRONX      40.9     -73.9
##  4      0     77673979      106 2011-03-11  2011    12 QUEENS     40.7     -73.8
##  5      0     77673979      106 2011-03-11  2011    12 QUEENS     40.7     -73.8
##  6      1    226950018       42 2021-04-14  2021    21 BRONX      40.8     -73.9
##  7      1    226950018       42 2021-04-14  2021    21 BRONX      40.8     -73.9
##  8      1    226950018       42 2021-04-14  2021    21 BRONX      40.8     -73.9
##  9      0    237710987       52 2021-12-10  2021    19 BRONX      40.9     -73.9
## 10      0    237710987       52 2021-12-10  2021    19 BRONX      40.9     -73.9
## 11      0    237710987       52 2021-12-10  2021    19 BRONX      40.9     -73.9
## 12      0    224701998       34 2021-02-22  2021     0 MANHA~     40.9     -73.9
## 13      0    224701998       34 2021-02-22  2021     0 MANHA~     40.9     -73.9
## 14      0    224701998       34 2021-02-22  2021     0 MANHA~     40.9     -73.9
## 15      1    225295736       75 2021-03-07  2021     6 BROOK~     40.7     -73.9
## # ... with 15 more variables: VIC_SEX_M <dbl>, PVT_HOUSE <dbl>,
## #   HOTEL_MOTEL <dbl>, MULTI_PUB_HOU <dbl>, MULTI_APT <dbl>, BAR_CLUB <dbl>,
## #   VC_AGE_18 <dbl>, VC_AGE_18_24 <dbl>, VC_AGE_25_44 <dbl>, VC_AGE_65 <dbl>,
## #   AREA <dbl>, LENGTH <dbl>, fType <chr>, H_Latitude <dbl>, H_Longitude <dbl>
```

```
tail(clean_dat3,15)
```

```
## # A tibble: 15 x 24
##    TARGET INCIDENT_KEY PRECINCT OCCUR_DATE  YEAR  HOUR BORO   Latitude Longitude
##     <dbl>        <dbl>    <dbl> <date>     <dbl> <int> <chr>     <dbl>     <dbl>
##  1      0    206524906       52 2019-12-14  2019    21 BRONX      40.9     -73.9
##  2      0    186329304       84 2018-08-12  2018    19 BROOK~     40.7     -74.0
##  3      0    186329304       84 2018-08-12  2018    19 BROOK~     40.7     -74.0
##  4      0    186329304       84 2018-08-12  2018    19 BROOK~     40.7     -74.0
##  5      0     29277330       81 2007-05-26  2007     4 BROOK~     40.7     -73.9
##  6      0     29277330       81 2007-05-26  2007     4 BROOK~     40.7     -73.9
##  7      0     29277330       81 2007-05-26  2007     4 BROOK~     40.7     -73.9
##  8      0     77443443       81 2011-02-25  2011     1 BROOK~     40.7     -73.9
##  9      0     77443443       81 2011-02-25  2011     1 BROOK~     40.7     -73.9
## 10      0     77443443       81 2011-02-25  2011     1 BROOK~     40.7     -73.9
## 11      0    176027888       43 2018-03-17  2018     0 BRONX      40.8     -73.9
## 12      0    176027888       43 2018-03-17  2018     0 BRONX      40.8     -73.9
## 13      0    176027888       43 2018-03-17  2018     0 BRONX      40.8     -73.9
## 14      0    218777493      113 2020-10-05  2020    12 QUEENS     40.7     -73.8
## 15      0    218777493      113 2020-10-05  2020    12 QUEENS     40.7     -73.8
## # ... with 15 more variables: VIC_SEX_M <dbl>, PVT_HOUSE <dbl>,
## #   HOTEL_MOTEL <dbl>, MULTI_PUB_HOU <dbl>, MULTI_APT <dbl>, BAR_CLUB <dbl>,
## #   VC_AGE_18 <dbl>, VC_AGE_18_24 <dbl>, VC_AGE_25_44 <dbl>, VC_AGE_65 <dbl>,
## #   AREA <dbl>, LENGTH <dbl>, fType <chr>, H_Latitude <dbl>, H_Longitude <dbl>
```

Transform Data

- Our final merge produced a file with duplicate incident keys. This is because some of the boroughs hold more than one Acute Care Hospital. In order to handle for this, we are going to apply the Haversine method to calculate the shortest distance between incident location and the surrounding hospitals using Latitude and Longitude of the incident and the hospitals. Finally, we retain the row in the data frame which contains the nearest hospital in relation to the incident location.
- In order to complete this, we are going to use the 'distHaversine' function within the 'geosphere' package.
- The new column containing the calculated shortest distance will be labeled as 'H_dist'
- NOTE: Per the database, Staten Island does not have an Acute Care facility. This will be handled by assigning H_latitude and H_Longitude values to the closest Acute Care facility, in the closest borough [Brooklyn, (40.58655, -73.96617)]

```
#clean_dat3
clean_dat4 <- clean_dat3 %>%
  mutate(H_Latitude = if_else(BORO=='STATEN ISLAND',40.58655,H_Latitude),
         H_Longitude = if_else(BORO=='STATEN ISLAND',-73.96617,H_Longitude),
         H_DIST = distHaversine(cbind(Longitude,Latitude),cbind(H_Longitude,H_Latitude)))
         #H_DIST = if_else(BORO=='STATEN ISLAND',mean(clean_dat4$H_DIST, na.rm = TRUE),H_DIST))
head(clean_dat4,5)
```

```
## # A tibble: 5 x 25
##   TARGET INCIDENT_KEY PRECINCT OCCUR_DATE  YEAR  HOUR BORO   Latitude Longitude
##    <dbl>        <dbl>    <dbl> <date>     <dbl> <int> <chr>     <dbl>     <dbl>
## 1      1     24050482       52 2006-08-27  2006     5 BRONX      40.9     -73.9
## 2      1     24050482       52 2006-08-27  2006     5 BRONX      40.9     -73.9
## 3      1     24050482       52 2006-08-27  2006     5 BRONX      40.9     -73.9
## 4      0     77673979      106 2011-03-11  2011    12 QUEENS     40.7     -73.8
## 5      0     77673979      106 2011-03-11  2011    12 QUEENS     40.7     -73.8
## # ... with 16 more variables: VIC_SEX_M <dbl>, PVT_HOUSE <dbl>,
## #   HOTEL_MOTEL <dbl>, MULTI_PUB_HOU <dbl>, MULTI_APT <dbl>, BAR_CLUB <dbl>,
## #   VC_AGE_18 <dbl>, VC_AGE_18_24 <dbl>, VC_AGE_25_44 <dbl>, VC_AGE_65 <dbl>,
## #   AREA <dbl>, LENGTH <dbl>, fType <chr>, H_Latitude <dbl>, H_Longitude <dbl>,
## #   H_DIST <dbl>
```

```r
retain_date <- as.data.frame(clean_dat3[,c('INCIDENT_KEY','OCCUR_DATE','YEAR')])
colnames(retain_date) <- c('INCIDENT_KEY','DATE1','YEAR1')
retain_date <- distinct(retain_date)

clean_dat5 <- clean_dat4 %>%
  select(TARGET,INCIDENT_KEY,PRECINCT,OCCUR_DATE,YEAR,HOUR,BORO,Latitude,Longitude,VIC_SEX_M
         ,PVT_HOUSE,HOTEL_MOTEL,MULTI_PUB_HOU,MULTI_APT,BAR_CLUB
         ,VC_AGE_18,VC_AGE_18_24,VC_AGE_25_44,VC_AGE_65,AREA,LENGTH
         , H_DIST)%>%
  group_by(INCIDENT_KEY) %>%
  slice(which.min(H_DIST))

clean_dat6 <- left_join(retain_date,clean_dat5,by='INCIDENT_KEY')
# It looks like
clean_dat6 <- clean_dat6 %>% select(TARGET,INCIDENT_KEY,PRECINCT,OCCUR_DATE,YEAR,HOUR,BORO,Latitude,Long
         ,PVT_HOUSE,HOTEL_MOTEL,MULTI_PUB_HOU,MULTI_APT,BAR_CLUB
         ,VC_AGE_18,VC_AGE_18_24,VC_AGE_25_44,VC_AGE_65,AREA,LENGTH
         , H_DIST)
```

```r
# Lets check the data one more time.
head(clean_dat6,5)
```

```
##   TARGET INCIDENT_KEY PRECINCT OCCUR_DATE YEAR HOUR      BORO Latitude
## 1      1     24050482       52 2006-08-27 2006    5     BRONX 40.86906
## 2      0     77673979      106 2011-03-11 2011   12    QUEENS 40.67737
## 3      1    226950018       42 2021-04-14 2021   21     BRONX 40.83376
## 4      0    237710987       52 2021-12-10 2021   19     BRONX 40.86941
## 5      0    224701998       34 2021-02-22 2021    0 MANHATTAN 40.86572
##   Longitude VIC_SEX_M PVT_HOUSE HOTEL_MOTEL MULTI_PUB_HOU MULTI_APT BAR_CLUB
## 1 -73.87963         0         0           0             0         0        0
## 2 -73.84392         1         0           0             0         0        0
## 3 -73.90880         1         0           0             0         0        0
## 4 -73.88000         1         0           0             0         0        0
## 5 -73.92344         1         0           0             0         0        0
##   VC_AGE_18 VC_AGE_18_24 VC_AGE_25_44 VC_AGE_65      AREA    LENGTH   H_DIST
## 1         0            0            0         0  80713476  65959.29 1280.672
## 2         0            0            0         1 173409706 107428.39 5599.590
## 3         0            1            0         0  44812434  33497.47 2210.465
## 4         0            0            0         0  80713476  65959.29 1238.317
## 5         0            0            0         0  52225254  44064.86 5898.895
```

```r
# Now that we confirmed our data looks good, lets only the necessary columns.

clean_dat7 <- clean_dat6 %>%
  select(TARGET,PRECINCT,YEAR,HOUR,BORO,Latitude,Longitude,VIC_SEX_M
         ,PVT_HOUSE,HOTEL_MOTEL,MULTI_PUB_HOU,MULTI_APT,BAR_CLUB
         ,VC_AGE_18,VC_AGE_18_24,VC_AGE_25_44,VC_AGE_65,AREA,LENGTH
         , H_DIST) %>%
  mutate(
    T1 = if_else(
      (HOUR>=0 & HOUR<3),1,0),
    T2 = if_else(
      (HOUR>=3 & HOUR<6),1,0),
    T3 = if_else(
      (HOUR>=6 & HOUR<9),1,0),
    T4 = if_else(
      (HOUR>=9 & HOUR<12),1,0),
    T5 = if_else(
      (HOUR>=12 & HOUR<15),1,0),
    T6 = if_else(
      (HOUR>=15 & HOUR<18),1,0),
    T7 = if_else(
      (HOUR>=18 & HOUR<21),1,0),
    T8 = if_else(
      (HOUR>=21 & HOUR<24),1,0),
    Tx = if_else(T1==1,'T1',
                 if_else(T2==1,'T2',
                         if_else(T3==1,'T3',
                                 if_else(T4==1,'T4',
                                         if_else(T5==1,'T5',
                                                 if_else(T6==1,'T6',
                                                         if_else(T7==1,'T7','T8'))))))))) %>%
  mutate_at(vars(T1,T2,T3,T4,T5,T6,T7,T8,Tx,
                 VC_AGE_18,VC_AGE_18_24,VC_AGE_25_44,VC_AGE_65
             ,PVT_HOUSE,HOTEL_MOTEL,MULTI_PUB_HOU,MULTI_APT,BAR_CLUB
             ,VIC_SEX_M,TARGET),factor)



head(clean_dat7,5)
```

```
##   TARGET PRECINCT YEAR HOUR      BORO Latitude Longitude VIC_SEX_M PVT_HOUSE
## 1      1       52 2006    5     BRONX 40.86906 -73.87963         0         0
## 2      0      106 2011   12    QUEENS 40.67737 -73.84392         1         0
## 3      1       42 2021   21     BRONX 40.83376 -73.90880         1         0
## 4      0       52 2021   19     BRONX 40.86941 -73.88000         1         0
## 5      0       34 2021    0 MANHATTAN 40.86572 -73.92344         1         0
##   HOTEL_MOTEL MULTI_PUB_HOU MULTI_APT BAR_CLUB VC_AGE_18 VC_AGE_18_24
## 1           0             0         0        0         0            0
## 2           0             0         0        0         0            0
## 3           0             0         0        0         0            1
## 4           0             0         0        0         0            0
## 5           0             0         0        0         0            0
##   VC_AGE_25_44 VC_AGE_65      AREA    LENGTH   H_DIST T1 T2 T3 T4 T5 T6 T7 T8
## 1            0         0  80713476  65959.29 1280.672  0  1  0  0  0  0  0  0
## 2            0         1 173409706 107428.39 5599.590  0  0  0  0  1  0  0  0
## 3            0         0  44812434  33497.47 2210.465  0  0  0  0  0  0  0  1
## 4            0         0  80713476  65959.29 1238.317  0  0  0  0  0  0  1  0
## 5            0         0  52225254  44064.86 5898.895  1  0  0  0  0  0  0  0
##   Tx
## 1 T2
## 2 T5
## 3 T8
## 4 T7
## 5 T1
```

```
# Transform all categoricals into factor type
cat_vars = c('T1','T2','T3','T4','T5','T6','T7','T8','Tx'
             ,'VC_AGE_18','VC_AGE_18_24','VC_AGE_25_44','VC_AGE_65'
             ,'PVT_HOUSE','HOTEL_MOTEL','MULTI_PUB_HOU','MULTI_APT','BAR_CLUB'
             ,'VIC_SEX_M','TARGET')


num_vars = c('Latitude','Longitude','AREA','LENGTH','H_DIST')

placeholders = c('PRECINCT','BORO','HOUR')
```

Next, we are going to split up our data set into a training and testing set in order to avoid introducing potential bias, the split will be 80% training and 20% for testing.

```
train_set1=clean_dat7
# Secondary split - splitting data into training and validation
set.seed(123)
split_dat = sample.split(train_set1$TARGET, SplitRatio = 0.8)

train_set2 = subset(train_set1, split_dat == TRUE)
validation_set = subset(train_set1, split_dat == FALSE)

# Checking proportions

prop.table(table(train_set2$TARGET))
```

```
##
##         0         1
## 0.8249689 0.1750311
```

```
prop.table(table(validation_set$TARGET))
```

```
##
##         0         1
## 0.8248882 0.1751118
```

Visualize the Data

- Next, we are going to visualize our data within the training set as part of exploratory analysis.

```
vis_dat <- train_set2[c(placeholders,cat_vars,num_vars)]
# Summary
summary(vis_dat)
```

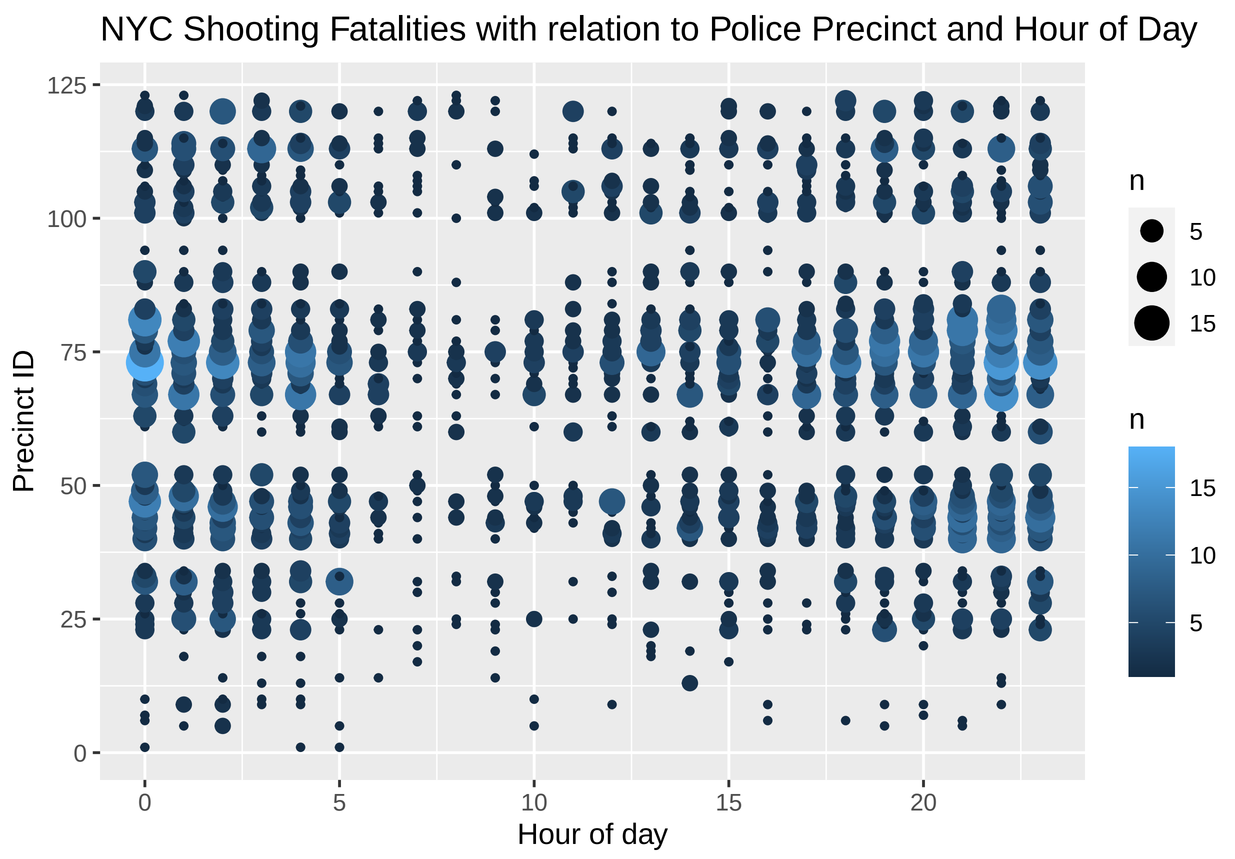
```
##     PRECINCT          BORO                  HOUR        T1         T2
##  Min.   :  1.00   Length:16100       Min.   : 0.00   0:12476   0:13868
##  1st Qu.: 44.00   Class :character   1st Qu.: 3.00   1: 3624   1: 2232
##  Median : 69.00   Mode  :character   Median :15.00
##  Mean   : 66.31                      Mean   :12.18
##  3rd Qu.: 81.00                      3rd Qu.:20.00
##  Max.   :123.00                      Max.   :23.00
##
##  T3        T4        T5        T6        T7        T8              Tx
##  0:15617   0:15551   0:15008   0:14318   0:13568   0:12294   T8     :3806
##  1:  483   1:  549   1: 1092   1: 1782   1: 2532   1: 3806   T1     :3624
##                                                              T7     :2532
##                                                              T2     :2232
##                                                              T6     :1782
##                                                              T5     :1092
##                                                              (Other):1032
##  VC_AGE_18 VC_AGE_18_24 VC_AGE_25_44 VC_AGE_65 PVT_HOUSE HOTEL_MOTEL
##  0:14486   0:10054      0:16100      0:15993   0:16100   0:16080
##  1: 1614   1: 6046                   1:  107             1:   20
##
##
##
##
##
##  MULTI_PUB_HOU MULTI_APT BAR_CLUB  VIC_SEX_M TARGET       Latitude
##  0:13135       0:16100   0:15782   0: 1244   0:13282   Min.   :40.51
##  1: 2965                 1:  318   1:14856   1: 2818   1st Qu.:40.67
##                                                        Median :40.70
##                                                        Mean   :40.73
##                                                        3rd Qu.:40.82
##                                                        Max.   :40.91
##
##    Longitude           AREA                LENGTH           H_DIST
##  Min.   :-74.25   Min.   : 15294022   Min.   : 17106   Min.   :    0.813
##  1st Qu.:-73.94   1st Qu.: 44812434   1st Qu.: 31464   1st Qu.: 1628.089
##  Median :-73.92   Median : 60400199   Median : 43256   Median : 2656.522
##  Mean   :-73.91   Mean   :101679394   Mean   : 63430   Mean   : 3511.272
##  3rd Qu.:-73.88   3rd Qu.:114119714   3rd Qu.: 80620   3rd Qu.: 4014.347
##  Max.   :-73.71   Max.   :475577638   Max.   :309678   Max.   :25361.387
##
```

```
# Structure
str(vis_dat)
```

```
## 'data.frame':    16100 obs. of  28 variables:
##  $ PRECINCT     : num  52 106 42 52 34 75 32 26 63 69 ...
##  $ BORO         : chr  "BRONX" "QUEENS" "BRONX" "BRONX" ...
##  $ HOUR         : int  5 12 21 19 0 6 0 20 22 22 ...
##  $ T1           : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 1 1 1 ...
##  $ T2           : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ T3           : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
##  $ T4           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ T5           : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
##  $ T6           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ T7           : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 2 1 1 ...
##  $ T8           : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 2 2 ...
##  $ Tx           : Factor w/ 8 levels "T1","T2","T3",..: 2 5 8 7 1 3 1 7 8 8 ...
##  $ VC_AGE_18    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ VC_AGE_18_24 : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 2 1 2 ...
##  $ VC_AGE_25_44 : Factor w/ 1 level "0": 1 1 1 1 1 1 1 1 1 1 ...
##  $ VC_AGE_65    : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
##  $ PVT_HOUSE    : Factor w/ 1 level "0": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HOTEL_MOTEL  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ MULTI_PUB_HOU: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ MULTI_APT    : Factor w/ 1 level "0": 1 1 1 1 1 1 1 1 1 1 ...
##  $ BAR_CLUB     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ VIC_SEX_M    : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
##  $ TARGET       : Factor w/ 2 levels "0","1": 2 1 2 1 1 2 1 1 1 1 ...
##  $ Latitude     : num  40.9 40.7 40.8 40.9 40.9 ...
##  $ Longitude    : num  -73.9 -73.8 -73.9 -73.9 -73.9 ...
##  $ AREA         : num  8.07e+07 1.73e+08 4.48e+07 8.07e+07 5.22e+07 ...
##  $ LENGTH       : num  65959 107428 33497 65959 44065 ...
##  $ H_DIST       : num  1281 5600 2210 1238 5899 ...
```

Lets take a quick look at Precincts and the hour of day.

```
viz1 <- vis_dat %>%
  select(TARGET,PRECINCT, HOUR) %>%
  # We are going to filter for cases which resulted in a fatality.
  filter(TARGET==1) %>%
  count(PRECINCT,HOUR)

# Size and color of the points will vary based on the count of incidents within that specific precinct.
ggplot(data=viz1, aes(x=HOUR,y=PRECINCT, size=n,color=n))+
  geom_point()+
  labs(title='NYC Shooting Fatalities with relation to Police Precinct and Hour of Day', x='Hour of day
```



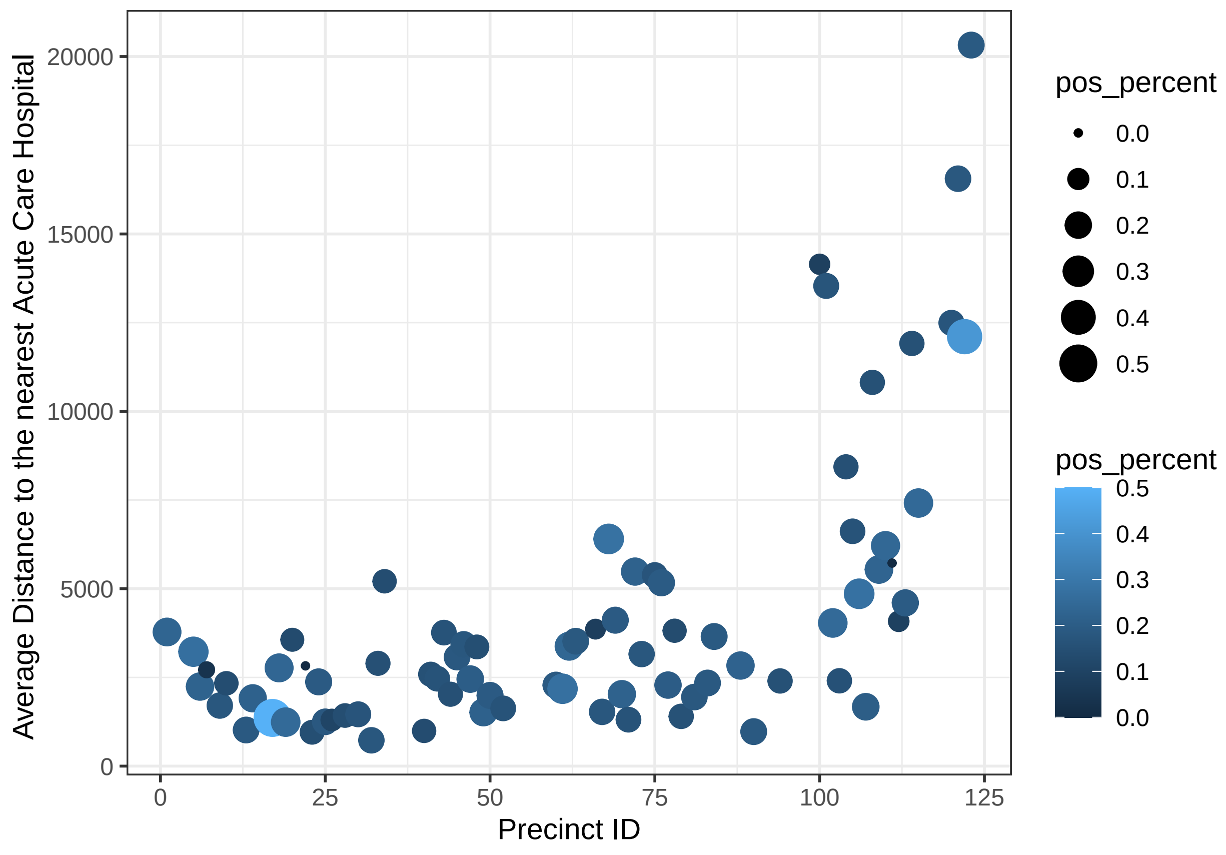
After accounting for the count of incidents by precinct and hour, we can determine that the highest concentration of fatal shootings actually takes place in precinct range of 25-85. An interesting observation here is in the difference of time ranges. Precincts in the range of 40-80 experience a relatively safe time frame of only a few hours, between 8am and 10am. Whereas precincts in the range of 25-40 experience very little fatal activity between the hours of 6am and 3pm, however, shooting fatalities pick right back up after 3pm and continue until 5am, which is when it starts to cool down. Another interesting observation here would be the increase in fatalities between the hours of approximately 4pm and 5am, this is especially true for precincts in the range of 100-125.

```r
viz2 <- vis_dat %>%
  select(TARGET,PRECINCT, H_DIST, AREA) %>%
  # We are going to filter for cases which resulted in a fatality.
  #filter(TARGET==0) %>%
  #count(PRECINCT,HOUR)
  mutate(t_pos = if_else(TARGET==1,1,0))%>%
  mutate(t_neg = if_else(TARGET==1,0,1))%>%
  group_by(PRECINCT)%>%
  summarize(
    H_DIST_avr = mean(H_DIST),
    AREA_avr = mean(AREA),
    pos_sum = sum(t_pos),
    neg_sum = sum(t_neg),
    tot_sum = pos_sum+neg_sum,
    pos_percent = pos_sum/tot_sum)

  #mutate(pos_percent_z = (pos_percent-mean(pos_percent))/sd(pos_percent))
head(viz2,5)
```

```
## # A tibble: 5 x 7
##   PRECINCT H_DIST_avr  AREA_avr pos_sum neg_sum tot_sum pos_percent
##      <dbl>      <dbl>     <dbl>   <dbl>   <dbl>   <dbl>       <dbl>
## 1        1      3780. 47300568.       3      10      13      0.231
## 2        5      3224. 18094527.       7      19      26      0.269
## 3        6      2243. 22103327.       4      14      18      0.222
## 4        7      2715. 18365996.       2      54      56      0.0357
## 5        9      1707. 21395839.      11      51      62      0.177
```

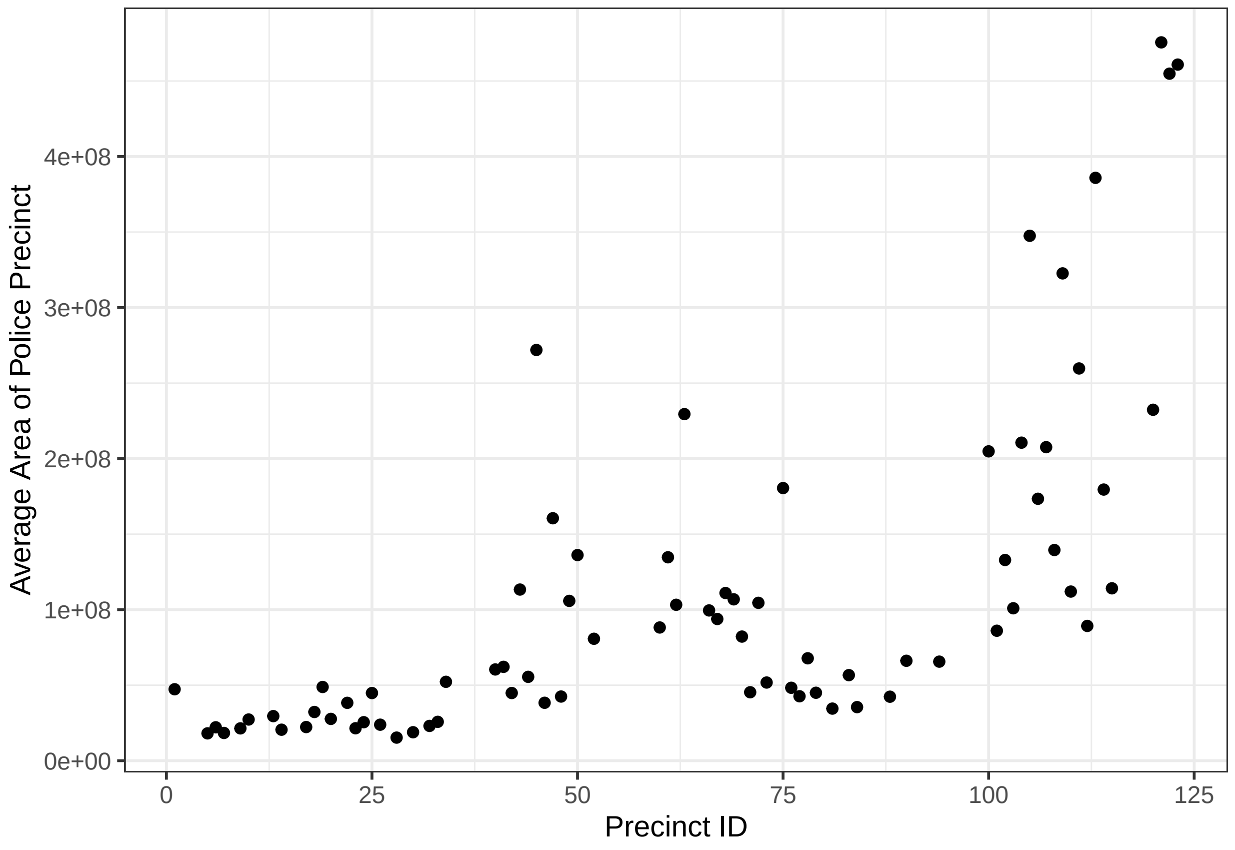
```r
# Size and color of the points will vary based on the count of incidents within that specific precinct.
p1 = ggplot(data=viz2,aes(y=H_DIST_avr,x=PRECINCT, size=pos_percent,color=pos_percent))+
  geom_point()+
  labs(x='Precinct ID',y='Average Distance to the nearest Acute Care Hospital ')+theme_bw()

p2 = ggplot(data=viz2,aes(x=PRECINCT,y=AREA_avr))+
  geom_point()+
  labs(y='Average Area of Police Precinct',x='Precinct ID')+
    theme_bw()


p3 = ggplot(data=viz2,aes(x=PRECINCT,y=pos_percent))+
  geom_point()+
  stat_smooth(method="lm", se=TRUE, formula=y~poly(x,6,raw=TRUE),color='red')+
  labs(y='Fatalities / Total Shootings',x='Precinct ID')+theme_bw()
```

```
p1
```

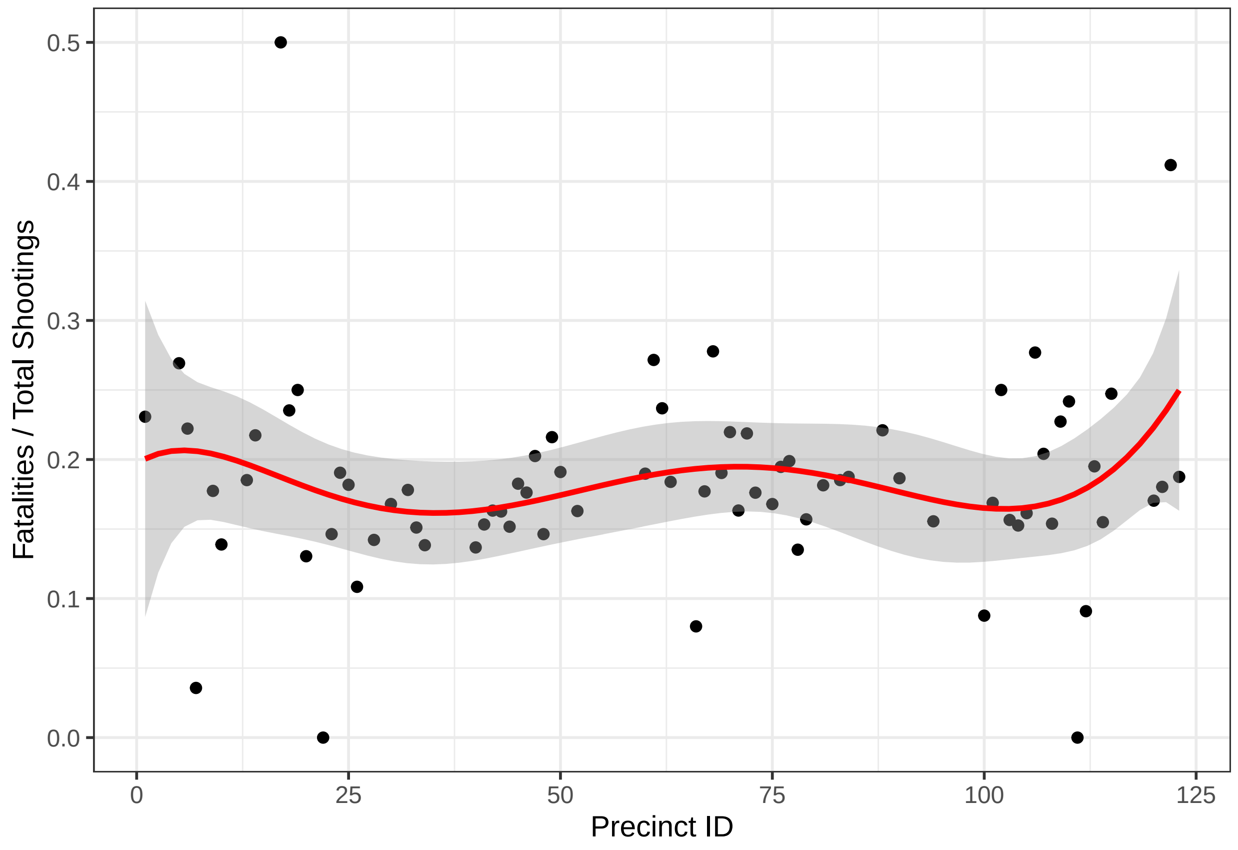


Average Distance to the nearest Acute Care Hospital
20000
15000
10000
5000
0
0
25
50
75
100
125
Precinct ID
pos_percent
0.0
0.1
0.2
0.3
0.4
0.5
pos_percent
0.5
0.4
0.3
0.2
0.1
0.0

```
p2
```


Average Area of Police Precinct
0e+00
1e+08
2e+08
3e+08
4e+08
0
25
50
75
100
125
Precinct ID

```
p3
```



Here we can see the percentage of shooting victims who did not survive increases in precincts within a range of 100-125. One potential explanatory variables in determining whether a victim lives or dies could be tied to the hospitals located within or around the precinct in question. This can be observed in precinct ID's within the range of 100-125, where the average distance to the nearest Acute Care Hospital is 2-3x longer relative to precincts in lower ranges.

```
vis_dat1 <- vis_dat %>%
  mutate(TAR_POS = if_else(TARGET==1,1,0))%>%
  mutate(TAR_NEG = if_else(TARGET==0,1,0))%>%
  group_by(BORO,Tx)%>%
  summarize(
    tar_pos = sum(TAR_POS),
    tar_neg = sum(TAR_NEG),
    tot_count = (tar_pos+tar_neg),
    pos_rat = tar_pos/tot_count)
```

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

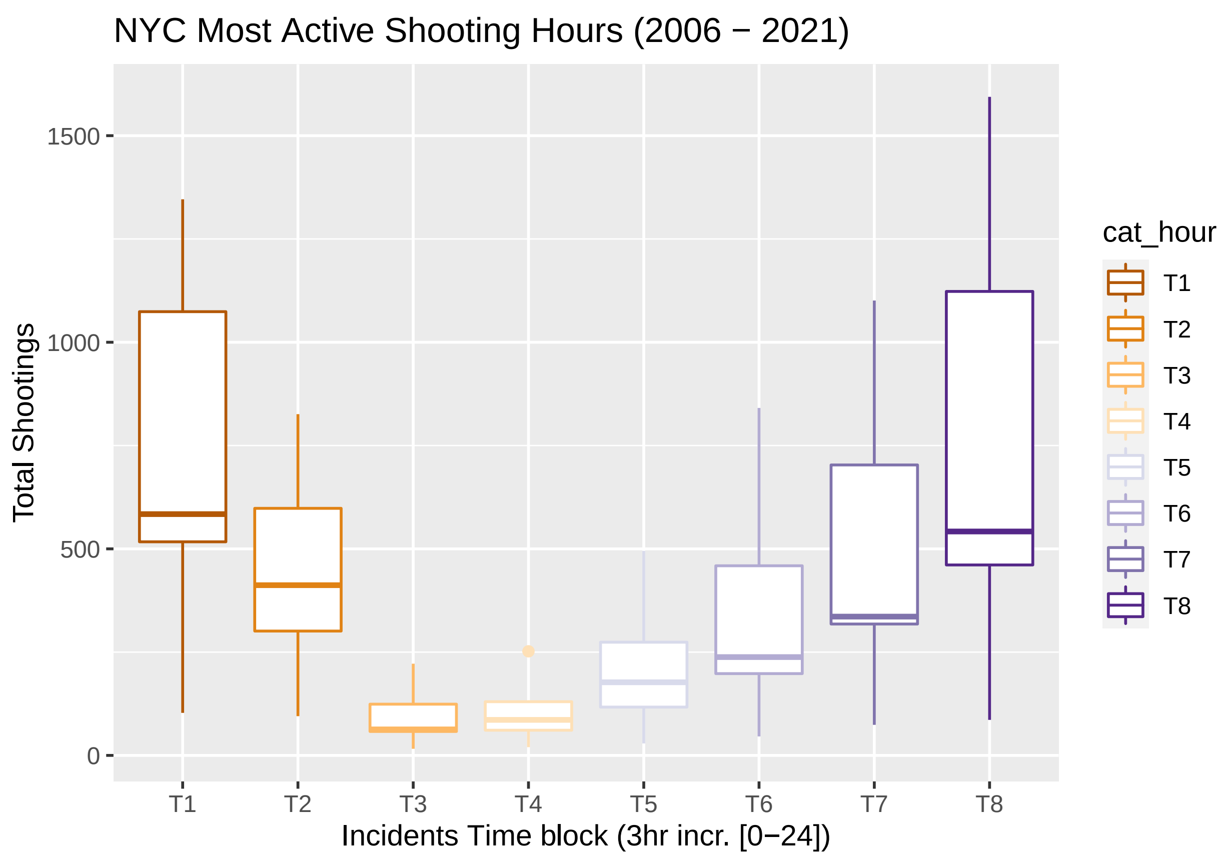
```
head(vis_dat1,5)
```

```
## # A tibble: 5 x 6
## # Groups:   BORO [1]
##   BORO  Tx    tar_pos tar_neg tot_count pos_rat
##   <chr> <fct>   <dbl>   <dbl>     <dbl>   <dbl>
## 1 BRONX T1        165     909      1074   0.154
## 2 BRONX T2        100     498       598   0.167
## 3 BRONX T3         20     104       124   0.161
## 4 BRONX T4         32      98       130   0.246
## 5 BRONX T5         57     217       274   0.208
```

```
#vis_datx <- train_set2[c(placeholders,cat_vars,num_vars)]
vis_datx1 <- vis_dat1 %>% select(BORO,Tx,tot_count)
head(vis_datx1,5)
```

```
## # A tibble: 5 x 3
## # Groups:   BORO [1]
##   BORO  Tx    tot_count
##   <chr> <fct>     <dbl>
## 1 BRONX T1         1074
## 2 BRONX T2          598
## 3 BRONX T3          124
## 4 BRONX T4          130
## 5 BRONX T5          274
```

```
cat_hour = as.factor(vis_datx1$Tx)

ggplot(data=vis_datx1, aes(x=cat_hour,y=tot_count, color=cat_hour)) +
  geom_boxplot()+scale_color_brewer(palette = "PuOr")+
  #geom_jitter(shape=1,position=(position_jitter(0.0)))+
  labs(title = 'NYC Most Active Shooting Hours (2006 - 2021)'
       ,y='Total Shootings',x='Incidents Time block (3hr incr. [0-24])')
```

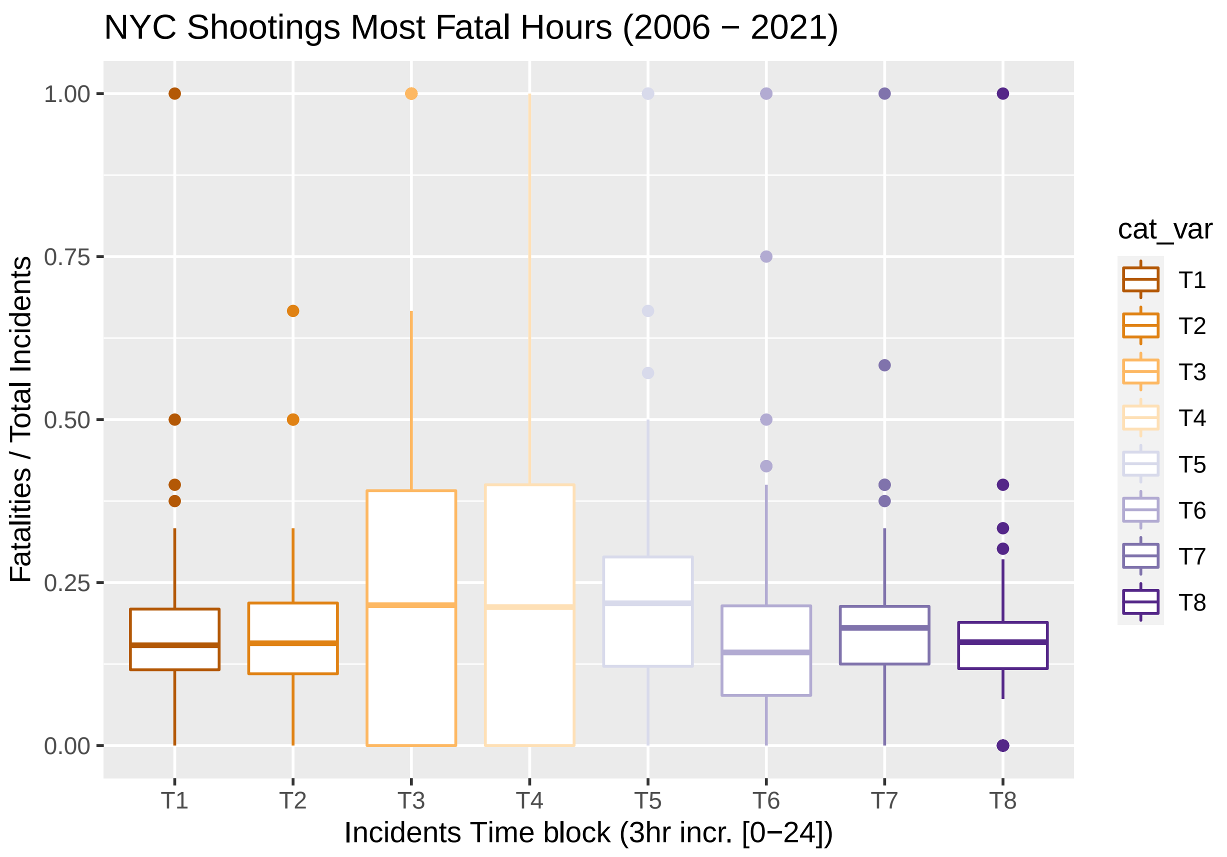


```
vis_datx3 <- vis_dat %>%
  select(PRECINCT,BORO,TARGET,AREA,H_DIST,LENGTH,Tx) %>%
  mutate(tar = if_else(TARGET==1,'YES','NO')) %>%
  mutate(YY = if_else(TARGET==1,1,0)) %>%
  mutate(NN = if_else(TARGET==1,0,1)) %>%
  #filter(BORO == 'MANHATTAN'
  group_by(Tx,PRECINCT) %>%
  summarize(#anss = if_else(TARGET==1,"YY","NN"),
            sum_yes = sum(YY),
            sum_no = sum(NN),
            tots = sum_yes+sum_no,
            yes_perc = sum_yes/tots,
            H_DIST_avr = mean(H_DIST),
            AREA_avr = mean(AREA))
```

```
## ‘summarise()‘ has grouped output by ’Tx’. You can override using the ‘.groups‘
## argument.
```

```
#vis_datx3

t1 <- vis_datx3[,c("Tx","AREA_avr","H_DIST_avr","yes_perc")]


colnames(t1) <- c("Tx", "AREA_avr","H_DIST_avr","yes_perc")
cat_var = as.factor(t1$Tx)
```

```
value_var = t1$yes_perc

plot1<- ggplot(data=t1, aes(x=cat_var,y=value_var, color=cat_var)) +
  geom_boxplot()+
  scale_color_brewer(palette = 'PuOr')+
  labs(title = 'NYC Shootings Most Fatal Hours (2006 - 2021)'
       ,y='Fatalities / Total Incidents'
       ,x='Incidents Time block (3hr incr. [0-24])')
plot1
```



One of the factors we could further explore is the time of day the shootings took place. Here we can see that the least active time period for shooters in New York City is between the hours of 6am and 12pm as indicated by time blocks 'T3' and 'T4' in the model above. These time blocks also carry a much larger interquartile range when it comes to shooting victims who did not survive as seen in the second graph.

- Note: Each 'T' in this model is a 3 hour increment starting from 0 and through the 24th hour.

Modeling Data

- Now we will see if we can build a prediction model given the data we have so far.
- We are going to be using the random forest algorithm as our approach for this model to take advantage of the "Wisdom of the crowds" concept where the collective opinion of many decision trees should yield a relatively better result than relying on a single tree. Furthermore, a random forest approach allows us to take advantage of feature importance in determining which of the variables are actually relevant to the survival of a shooting victim.

  - In taking this approach we are also reducing over-fit bias by averaging the result of many decision trees.

```
set.seed(123)
selected_vars = c('TARGET','T1','T2','T3','T4','T5','T6','T7','T8','Tx'
             ,'VC_AGE_18','VC_AGE_18_24','VC_AGE_25_44','VC_AGE_65'
             ,'PVT_HOUSE','HOTEL_MOTEL','MULTI_PUB_HOU','MULTI_APT','BAR_CLUB'
             ,'VIC_SEX_M','Latitude','Longitude','AREA','LENGTH','H_DIST')

trainingset <- train_set2%>%select(selected_vars)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(selected_vars)` instead of `selected_vars` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
rf <- randomForest(TARGET~.,data = trainingset)

importance(rf)
```

```
##               MeanDecreaseGini
## T1                   12.605704
## T2                   11.897424
## T3                    6.194163
## T4                    6.735941
## T5                    8.983127
## T6                   10.004809
## T7                   12.458966
## T8                   13.325587
## Tx                   69.275580
## VC_AGE_18            25.136609
## VC_AGE_18_24         32.765173
## VC_AGE_25_44          0.000000
## VC_AGE_65             8.441616
## PVT_HOUSE             0.000000
## HOTEL_MOTEL           7.209364
## MULTI_PUB_HOU        26.725337
## MULTI_APT             0.000000
## BAR_CLUB              8.911208
## VIC_SEX_M            26.725689
## Latitude            282.360381
## Longitude           281.124138
## AREA                 93.091025
## LENGTH               94.238636
## H_DIST              287.417275
```

```
print(rf)
```

```
##
## Call:
##  randomForest(formula = TARGET ~ ., data = trainingset)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.51%
## Confusion matrix:
##       0 1  class.error
## 0 13276 6 0.0004517392
## 1  2813 5 0.9982256920
```

```
validset<-validation_set%>%select(selected_vars)
pred = predict(rf,newdata=validset[-1])
table(validset[,1],pred)
```

```
##    pred
##        0    1
##   0 3320    1
##   1  703    2
```

```
accuracy=mean(validset[,1]==pred)
accuracy
```

```
## [1] 0.8251366
```

So it looks like our top 6 most important variables here would be:

1. (H_DIST): Distance to nearest Acute Care Hospital
2. (Latitude): Latitude of the incident location.
3. (Longitude): Longitude of the incident location.
4. (LENGTH): Length of police precinct
5. (AREA): Area of police precinct
6. (Tx): Time of day increments

Using all of the variables, our random forest model achieved accuracy of approximately 82.49% during training and 82.51% when tested against the testing set.

Now we are going to repeat the process one more time but this time only using the top 6 features.

Conclusion

There is an evident bias stemming from the imbalance in the positive cases and a lot more work is needed in terms of transformations and model tuning. We also cannot rule out the value of other features within the feature set and will need to dive deeper into the iterative process of modeling, transforming, and visualizing in order to improve our model performance.

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## 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
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] class_7.3-20         randomForest_4.7-1.1 caTools_1.18.2
##  [4] geosphere_1.5-14     lubridate_1.8.0      forcats_0.5.1
##  [7] stringr_1.4.0        dplyr_1.0.9          purrr_0.3.4
## [10] readr_2.1.2          tidyr_1.2.0          tibble_3.1.7
## [13] ggplot2_3.3.6        tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.3         bit64_4.0.5        vroom_1.5.7        jsonlite_1.8.0
##  [5] splines_4.2.1      modelr_0.1.8       assertthat_0.2.1   sp_1.5-0
##  [9] highr_0.9          cellranger_1.1.0   yaml_2.3.5         pillar_1.7.0
## [13] backports_1.4.1    lattice_0.20-45    glue_1.6.2         digest_0.6.29
## [17] RColorBrewer_1.1-3 rvest_1.0.2        colorspace_2.0-3   htmltools_0.5.2
## [21] Matrix_1.4-1       pkgconfig_2.0.3    broom_1.0.0        haven_2.5.0
## [25] scales_1.2.0       tzdb_0.3.0         mgcv_1.8-40        generics_0.1.3
## [29] farver_2.1.1       ellipsis_0.3.2     withr_2.5.0        cli_3.3.0
## [33] magrittr_2.0.3     crayon_1.5.1       readxl_1.4.0       evaluate_0.15
## [37] fs_1.5.2           fansi_1.0.3        nlme_3.1-157       xml2_1.3.3
## [41] tools_4.2.1        hms_1.1.1          lifecycle_1.0.1    munsell_0.5.0
## [45] reprex_2.0.1       compiler_4.2.1     rlang_1.0.3        grid_4.2.1
## [49] rstudioapi_0.13    bitops_1.0-7       labeling_0.4.2     rmarkdown_2.14
## [53] gtable_0.3.0       DBI_1.1.3          curl_4.3.2         R6_2.5.1
## [57] knitr_1.39         fastmap_1.1.0      bit_4.0.4          utf8_1.2.2
## [61] stringi_1.7.6      parallel_4.2.1     vctrs_0.4.1        dbplyr_2.2.1
## [65] tidyselect_1.1.2   xfun_0.31
```