

Police Data on Shooting Incidents in New York City

RLReichel

2024-04-06

Libraries and File Set Up

In this analysis of historic NYPD shooting incident data, we will be exploring the data as well as looking at the ratio of victims by demographic versus the ratio of the population of the borough to which they belong in both deaths and non-fatal shootings.

We have selected the victim data as a point of focus due to the mostly complete nature of the data compared to the perpetrator and location descriptions which are incomplete. The demographic information we intend to focus on is race, age, and sex, but we will still explore other information and the perpetrator data.

We will use these libraries:

- tidyverse
- readxl
- writexl
- lubridate
- downloader
- XML

```
library(tidyverse) #library for dataframe tidying and organization, includes other packages such as tibbles, purrr, and dplyr

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.4    v readr     2.1.5
## vforcats   1.0.0    v stringr   1.5.1
## v ggplot2   3.5.1    v tibble    3.2.1
## v lubridate 1.9.3    v tidyverse  1.3.1
## v purrr    1.0.2

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate) #library for working with times and dates
library(readxl) #library for reading from Excel files
library(writexl) #library for writing to Excel files
library(downloader) #library to download Excel file needed
library(XML) #library needed to read table from Excel file when data is not clean
library(latexpdf) #helps write PDF files
if(!file.exists("Data")){
  dir.create("Data")
} #creates Data file folder in wd if not present
```

Accessing Data Sources

In order to compare shooting victim demographics versus overall population demographics, we need access to data sets that contain the information we will clean and analyze. We will retrieve this data from the city of New York's database.

The specific data sets we will use are:

- NYPD Historical Shooting Incident Data +<https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD>
- NYC American Community Survey data set with estimated populations by demographic per borough for 2012-2016 +https://www1.nyc.gov/assets/planning/download/office/data-maps/nyc-population/acs/demo_2016acs5yr_nyc.xlsx

```
NYPD_Shooting_data1 <- read.csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")
write.csv(NYPD_Shooting_data1,file = "Data/data1.csv") #writes a CSV file into the Data file in the working directory

#students_demographics_boro <- read.csv("https://data.cityofnewyork.us/api/views/uh2w-zjsn/rows.csv?accessType=DOWNLOAD")
#retrieves 2014-2019 demographic snapshot data of students 18 and under by boro for NYC

#write.csv(students_demographics_boro,file = "Data/data2.csv")
#writes a CSV file into the Data file in the working directory of the retrieved data for back up storage

#Commented out above dataset due to lack of time to delve into 18 and under age group

download.file("https://www1.nyc.gov/assets/planning/download/office/data-maps/nyc-population/acs/demo_2016acs5yr_nyc.xlsx")
```

Tidy Excel Sheet Census Data

First we will focus on tidying the excel sheet in such a way that RStudio can handle it as a dataframe, since it is optimized for human eyes and unreadable to RStudio in its current form.

Tidy Excel Sheet Headings

In this section, we will extract the headings from the Excel sheet in order to skip the empty cells above and avoid creating confusion in the data. We will delete the empty columns and combine the category names (Boroughs) with their subcategories in order to create a single line of headings from the double-lined headings of the original Excel spreadsheet.

```
col_names <- read_excel("Data/data3.xlsx",skip = 3, n_max = 1) #skips over empty rows to copy in headings

## New names:
## * `` -> `...2`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...14`
## * `` -> `...16`
## * `` -> `...17`
## * `` -> `...18`
## * `` -> `...20`
## * `` -> `...21`
```

```

## * `` -> `...22`  

## * `` -> `...24`  

## * `` -> `...25`  

## * `` -> `...26`  

boro_list <- names(col_names[1]) #establishes boro_list object to append all borough names as first header  

i <- 3 #sets i index variable to first available filled index past 1 in col_names (first borough name)  

while (i <= 23) {  

  boro_list <- append(boro_list, names(col_names[i]))  

  i <- i + 4} #appends borough names, listed 4 spaces apart due to Excel sheet, without appending unfilled cells  

heading_suffix <- unlist(col_names[3],use.names = FALSE) #establishes heading suffix list with first header  

for (i in 4:6){heading_suffix <- append(heading_suffix,unlist(col_names[i],use.names = FALSE))} #fills heading suffix list  

n <- length(boro_list) #stores length of col_names list in n  

m <- length(heading_suffix) #stores length of heading_suffix in m  

headings <- boro_list[1] #establishes headings vector and puts in first heading without heading_suffix  

for (i in 2:n){  

  for (j in 1:m){ x <- paste(boro_list[i],heading_suffix[j])  

    headings <- append(headings,x)}} #creates descriptive heading titles for dataframe  

headings <- gsub(" ", "_",headings) #removes spaces and replaces them with underscores for use in datafram  

rm(i)  

rm(j)  

rm(m)  

rm(n)  

rm(x)  

#remove temporary variables for storage space

```

Retrieve Full Data from Excel Sheet

Now that we have our headings tidied, we are going to pull the rest of the data from the Excel sheet and apply our headings.

```

full_data_dirty <- read_excel("Data/data3.xlsx",skip = 6,col_names = FALSE) #transfers all data from the excel sheet to R  

## New names:  

## * `` -> `...1`  

## * `` -> `...2`  

## * `` -> `...3`  

## * `` -> `...4`  

## * `` -> `...5`  

## * `` -> `...6`  

## * `` -> `...7`  

## * `` -> `...8`  

## * `` -> `...9`  

## * `` -> `...10`  

## * `` -> `...11`  

## * `` -> `...12`  


```

```

## * `` -> `...13`  

## * `` -> `...14`  

## * `` -> `...15`  

## * `` -> `...16`  

## * `` -> `...17`  

## * `` -> `...18`  

## * `` -> `...19`  

## * `` -> `...20`  

## * `` -> `...21`  

## * `` -> `...22`  

## * `` -> `...23`  

## * `` -> `...24`  

## * `` -> `...25`  

## * `` -> `...26`  

sub_data_census <- subset(full_data_dirty, select = -c(2)) #creates a subset of full_data_dirty not including the first column  

colnames(sub_data_census) <- c(headings) #changes column names of sub_data_census to the values listed in headings

```

Tidy Full Excel Data and Create Subsets

Next we are going to isolate some of the information we need and remove any duplicate information in the data by creating a subset of the demographic information relevant to the police categories used.

```

df_transpose <- as.data.frame(t(sub_data_census)) #Transposes columns and rows of sub_data_census into a long format
df_subset <- subset(df_transpose, select = c(V1,V2,V3,V5,V6,V7,V8,V9,V10,V11,V12,V13,V14,V15,V16,V17,V40))
df_transpose <- as.data.frame(t(df_subset)) #transposing back to original column/row configuration

k <- length(df_transpose$Subject) #finding last index number of Subject column, which is a duplicate to the first one
for (i in 30:35){
  df_transpose$Subject[i] <- paste("Hispanic",df_transpose$Subject[i],sep="_")
} #clarifying category names as subcategories of Hispanic/Latino

df_transpose$Subject <- gsub(" ", "_",df_transpose$Subject) #erasing spaces for ease of use in later visualizations

rownames(df_transpose) <- df_transpose$Subject #adding row names back in from first column

cleaner_sub_data_census <- subset(df_transpose, select = -c(Subject)) #creating cleaner subset without the subject column
rm(df_transpose)
rm(df_subset) #remove temporary dataframes for storage space

```

Tidy Police Data

To tidy the police data, we're going to isolate the shooting victim sex, race, age, borough, and whether they survived in a subset for future visualization side by side with the demographic information. We are focusing on the victim data primarily because it is far more complete than perpetrator and location description data, but we will still analyze the perpetrator data.

Following this, we'll create a summary subset of demographics per borough.

```

Shooting_Victims <- subset(NYPD_Shooting_data1, select = -c(PERP_SEX,PERP_RACE,X_COORD_CD,Y_COORD_CD,PERP_AGE))
Shooting_Perps <- subset(NYPD_Shooting_data1, select = -c(VIC_SEX,VIC_RACE,X_COORD_CD,Y_COORD_CD,VIC_AGE))

```

```

k <- length(Shooting_Victims$STATISTICAL_MURDER_FLAG) #sets variable to length of column Statistical M

for(i in 1:k){if(identical(NYPD_Shooting_data1$STATISTICAL_MURDER_FLAG[i], "false")){Shooting_Victims$ST

Shooting_Perps$STATISTICAL_MURDER_FLAG <- Shooting_Victims$STATISTICAL_MURDER_FLAG #This copies the new

race_list <- "BLACK" #initialize race list

for (i in 1:k){flag <- !(Shooting_Victims$VIC_RACE[i] %in% race_list)
if(flag){race_list <- append(race_list,Shooting_Victims$VIC_RACE[i])}} #Harvest any races listed in Sho

print(race_list) #check for any potential misentered data such as typos

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

age_list <- "65+" #initialize age list

for (i in 1:k){flag <- !(Shooting_Victims$VIC_AGE_GROUP[i] %in% age_list)
if(flag){age_list <- append(age_list,Shooting_Victims$VIC_AGE_GROUP[i])}} #harvest all unique age categ

print(age_list) #check for any potential misentered data such as typos

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

sex_list <- "F" #initialize sex list

for (i in 1:k){flag <- !(Shooting_Victims$VIC_SEX[i] %in% sex_list)
if(flag){sex_list <- append(sex_list,Shooting_Victims$VIC_SEX[i])}} 

print(sex_list) #check for typos and intersex folk or other categories

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

#discovered U for unknown sex; no action needed in this block; added to future code

perp_race_list <- "BLACK" #initialize race list

for (i in 1:k){flag <- !(Shooting_Perps$PERP_RACE[i] %in% perp_race_list)
if(flag){perp_race_list <- append(perp_race_list,Shooting_Perps$PERP_RACE[i])}} #Harvest any races list

print(perp_race_list) #check for any potential misentered data such as typos

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

perp_age_list <- "65+" #initialize age list

for (i in 1:k){flag <- !(Shooting_Perps$PERP_AGE_GROUP[i] %in% perp_age_list)
if(flag){perp_age_list <- append(perp_age_list,Shooting_Perps$PERP_AGE_GROUP[i])}} #harvest all unique

print(perp_age_list) #check for any potential misentered data such as typos

```

```

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

k <- length(Shooting_Victims$VIC_SEX) #reset k to length of Shooting_Victims VIC_SEX column as the most

perp_sex_list <- "F"

for (i in 1:k){flag <- !(Shooting_Perps$PERP_SEX[i] %in% perp_sex_list)
if(flag){perp_sex_list <- append(perp_sex_list,Shooting_Perps$PERP_SEX[i])}

print(perp_sex_list) #check for typos and intersex folk or other categories

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

#from perp lists, discovered: "(null)" and "" in race, age, and sex as well as 940, 224, and 1020 in a

```

Expand Columns in Shooting Data and Clean for Calculations

For the police data and based on our age, sex, and race lists from the previous section, we need to expand:

- race into seven new columns.
- age into six new columns.
- sex into three new columns.

In these columns, instead of listing the string “WHITE” or another race, the column named after each race category will have a 1 for TRUE or a 0 for FALSE depending on whether they are that race (true) or aren’t that race (false), with the same happening in the age and sex columns. This allows for easy calculations later and easier conversion into grouping by borough.

While we’re at it, we can do the same for perpetrators to look at the data for any trends although it is incomplete. This will be completed in a separate subset.

```

temp <- Shooting_Victims #creates a dataframe copy of Shooting_Victims

temp$Male <- temp$VIC_SEX
temp$Female <- temp$VIC_SEX
temp$Unknown_Sex <- temp$VIC_SEX
temp$Unlisted_Sex <- temp$VIC_SEX

#these create new columns of the same length as VIC_SEX without mutating them, since for loops expand t

for (i in 1:k) {
  if (identical(temp$VIC_SEX[i], "M")){
    temp$Male[i] <- 1
    temp$Female[i] <- 0
    temp$Unknown_Sex[i] <- 0
    temp$Unlisted_Sex[i] <- 0
  }
  else if (identical(temp$VIC_SEX[i], "F")) {
    temp$Male[i] <- 0
    temp$Female[i] <- 1
    temp$Unknown_Sex[i] <- 0
    temp$Unlisted_Sex[i] <- 0
  }
  else if (identical(temp$VIC_SEX[i], "U")){
    temp$Male[i] <- 0
  }
}

```

```

    temp$Female[i] <- 0
    temp$Unknown_Sex[i] <- 1
    temp$Unlisted_Sex[i] <- 0
}
else{
    temp$Male[i] <- 0
    temp$Female[i] <- 0
    temp$Unknown_Sex[i] <- 0
    temp$Unlisted_Sex[i] <- 1
}
} #This assigns binary 1/0 for yes/no on whether the victim was male, female, or unknown sex in the res...
```

Now that we've expanded the victim columns to separate out sex into separate columns for male, female, and unknown with binary 1/0 entries in each, we'll do the same for race.

```

temp$BLACK <- temp$VIC_RACE

temp$WHITE <- temp$VIC_RACE

temp$WHITE_HISPANIC <- temp$VIC_RACE

temp$BLACK_HISPANIC <- temp$VIC_RACE

temp$ASIAN_PACIFIC_ISLANDER <- temp$VIC_RACE

temp$AMERICAN_INDIAN_ALASKAN_NATIVE <- temp$VIC_RACE

temp$UNKNOWN_RACE <- temp$VIC_RACE

temp$UNLISTED_RACE <- temp$VIC_RACE

#This block above creates columns for every race noted in the original data; for loop used rather than if
```

```

for (i in 1:k){
  if (identical(temp$VIC_RACE[i], "BLACK")) {
    temp$BLACK[i] <- 1
    temp$WHITE[i] <- 0
    temp$WHITE_HISPANIC[i] <- 0
    temp$BLACK_HISPANIC[i] <- 0
    temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
    temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
    temp$UNKNOWN_RACE[i] <- 0
    temp$UNLISTED_RACE[i] <- 0
  } #updates Black column if victim was Black
  else if (identical(temp$VIC_RACE[i], "WHITE")) {
    temp$BLACK[i] <- 0
    temp$WHITE[i] <- 1
    temp$WHITE_HISPANIC[i] <- 0
    temp$BLACK_HISPANIC[i] <- 0
    temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
    temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
    temp$UNKNOWN_RACE[i] <- 0
    temp$UNLISTED_RACE[i] <- 0
  } #updates White column if victim was white
}
```

```

else if (identical(temp$VIC_RACE[i], "WHITE HISPANIC")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 1
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates White Hispanic column if victim was white Hispanic
else if (identical(temp$VIC_RACE[i], "BLACK HISPANIC")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 1
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates Black Hispanic column if victim was Black Hispanic
else if (identical(temp$VIC_RACE[i], "ASIAN / PACIFIC ISLANDER")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 1
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates Asian/Pacific Islander column if victim was Asian or a Pacific Islander
else if (identical(temp$VIC_RACE[i], "AMERICAN INDIAN/ALASKAN NATIVE")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 1
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates American Indian/Alaskan Native column if victim was American Indian/Alaskan Native
else if (identical(temp$VIC_RACE[i], "UNKNOWN")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 1
  temp$UNLISTED_RACE[i] <- 0
} #updates Unknown Race column if victim race was unknown
else {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
}

```

```

    temp$WHITE_HISPANIC[i] <- 0
    temp$BLACK_HISPANIC[i] <- 0
    temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
    temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
    temp$UNKNOWN_RACE[i] <- 0
    temp$UNLISTED_RACE[i] <- 1
}
}

#updates race columns with binary 1/0 depending on if the victim was that race (1) or was not that race

```

Now that we've expanded the victim columns to separate out race into separate columns for each listed race umbrella category provided by the police data with binary 1/0 entries in each, we'll do the same for age group.

```

"#65+"      "18-24"     "25-44"      "<18"        "45-64"      "UNKNOWN"

temp$SixtyFivePlus <- temp$VIC_AGE_GROUP
temp$FortyFive_SixtyFour <- temp$VIC_AGE_GROUP
temp$TwentyFive_FortyFour <- temp$VIC_AGE_GROUP
temp$Eighteen_TwentyFour <- temp$VIC_AGE_GROUP
temp$Below_Eighteen <- temp$VIC_AGE_GROUP
temp$Unknown_Age <- temp$VIC_AGE_GROUP
temp$Unlisted_Age <- temp$VIC_AGE_GROUP

for (i in 1:k) {
  if (identical(temp$VIC_AGE_GROUP[i], "65+")){
    temp$SixtyFivePlus[i] <- 1
    temp$FortyFive_SixtyFour[i] <- 0
    temp$TwentyFive_FortyFour[i] <- 0
    temp$Eighteen_TwentyFour[i] <- 0
    temp$Below_Eighteen[i] <- 0
    temp$Unknown_Age[i] <- 0
    temp$Unlisted_Age[i] <- 0
  }
  else if (identical(temp$VIC_AGE_GROUP[i], "45-64")){
    temp$SixtyFivePlus[i] <- 0
    temp$FortyFive_SixtyFour[i] <- 1
    temp$TwentyFive_FortyFour[i] <- 0
    temp$Eighteen_TwentyFour[i] <- 0
    temp$Below_Eighteen[i] <- 0
    temp$Unknown_Age[i] <- 0
    temp$Unlisted_Age[i] <- 0
  }
  else if (identical(temp$VIC_AGE_GROUP[i], "25-44")){
    temp$SixtyFivePlus[i] <- 0
    temp$FortyFive_SixtyFour[i] <- 0
    temp$TwentyFive_FortyFour[i] <- 1
    temp$Eighteen_TwentyFour[i] <- 0
    temp$Below_Eighteen[i] <- 0
    temp$Unknown_Age[i] <- 0
    temp$Unlisted_Age[i] <- 0
  }
  else if (identical(temp$VIC_AGE_GROUP[i], "18-24")){
    temp$SixtyFivePlus[i] <- 0
    temp$FortyFive_SixtyFour[i] <- 0
    temp$TwentyFive_FortyFour[i] <- 0
    temp$Eighteen_TwentyFour[i] <- 1
    temp$Below_Eighteen[i] <- 0
    temp$Unknown_Age[i] <- 0
    temp$Unlisted_Age[i] <- 0
  }
}

```

```

temp$FortyFive_SixtyFour[i] <- 0
temp$TwentyFive_FortyFour[i] <- 0
temp$Eighteen_TwentyFour[i] <- 1
temp$Below_Eighteen[i] <- 0
temp$Unknown_Age[i] <- 0
temp$Unlisted_Age[i] <- 0
}
else if (identical(temp$VIC AGE GROUP[i], "<18")) {
  temp$SixtyFivePlus[i] <- 0
  temp$FortyFive_SixtyFour[i] <- 0
  temp$TwentyFive_FortyFour[i] <- 0
  temp$Eighteen_TwentyFour[i] <- 0
  temp$Below_Eighteen[i] <- 1
  temp$Unknown_Age[i] <- 0
  temp$Unlisted_Age[i] <- 0
}
else if (identical(temp$VIC AGE GROUP[i], "UNKNOWN")) {
  temp$SixtyFivePlus[i] <- 0
  temp$FortyFive_SixtyFour[i] <- 0
  temp$TwentyFive_FortyFour[i] <- 0
  temp$Eighteen_TwentyFour[i] <- 0
  temp$Below_Eighteen[i] <- 0
  temp$Unknown_Age[i] <- 1
  temp$Unlisted_Age[i] <- 0
}
else {
  temp$SixtyFivePlus[i] <- 0
  temp$FortyFive_SixtyFour[i] <- 0
  temp$TwentyFive_FortyFour[i] <- 0
  temp$Eighteen_TwentyFour[i] <- 0
  temp$Below_Eighteen[i] <- 0
  temp$Unknown_Age[i] <- 0
  temp$Unlisted_Age[i] <- 1
}
}

Shooting_Victims <- temp %>% select(-c(VIC AGE GROUP, VIC SEX, VIC RACE))

```

Next, we'll follow these steps for the perpetrators, as well, separating out sex, race, and age group into their own columns with binary 1/0 entries in each to indicate whether there was a perpetrator belonging to that category.

We will start with sex.

```

temp <- Shooting_Perps #creates a dataframe copy of Shooting_Perps

temp$Male <- temp$PERP_SEX
temp$Female <- temp$PERP_SEX
temp$Unknown_Sex <- temp$PERP_SEX
temp$Unlisted_Sex <- temp$PERP_SEX

#these create new columns of the same length as VIC_SEX without mutating them, since for loops expand to

for (i in 1:k) {
  if (identical(temp$PERP_SEX[i], "M")){

```

```

temp$Male[i] <- 1
temp$Female[i] <- 0
temp$Unknown_Sex[i] <- 0
temp$Unlisted_Sex[i] <- 0
}
else if (identical(temp$PERP_SEX[i], "F")) {
  temp$Male[i] <- 0
  temp$Female[i] <- 1
  temp$Unknown_Sex[i] <- 0
  temp$Unlisted_Sex[i] <- 0
}
else if (identical(temp$PERP_SEX[i], "U")) {
  temp$Male[i] <- 0
  temp$Female[i] <- 0
  temp$Unknown_Sex[i] <- 1
  temp$Unlisted_Sex[i] <- 0
}
else{
  temp$Male[i] <- 0
  temp$Female[i] <- 0
  temp$Unknown_Sex[i] <- 0
  temp$Unlisted_Sex[i] <- 1
}
} #This assigns binary 1/0 for yes/no on whether the victim was male, female, or unknown sex in the res...
```

Following expanding the columns for sex, we will now expand the race columns.

```

temp$BLACK <- temp$PERP_RACE

temp$WHITE <- temp$PERP_RACE

temp$WHITE_HISPANIC <- temp$PERP_RACE

temp$BLACK_HISPANIC <- temp$PERP_RACE

temp$ASIAN_PACIFIC_ISLANDER <- temp$PERP_RACE

temp$AMERICAN_INDIAN_ALASKAN_NATIVE <- temp$PERP_RACE

temp$UNKNOWN_RACE <- temp$PERP_RACE

temp$UNLISTED_RACE <- temp$PERP_RACE

#This block above creates columns for every race noted in the original data; for loop used rather than if
```

```

for (i in 1:k){
  if (identical(temp$PERP_RACE[i], "BLACK")) {
    temp$BLACK[i] <- 1
    temp$WHITE[i] <- 0
    temp$WHITE_HISPANIC[i] <- 0
    temp$BLACK_HISPANIC[i] <- 0
    temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
    temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
    temp$UNKNOWN_RACE[i] <- 0
  }
}
```

```

    temp$UNLISTED_RACE[i] <- 0
} #updates Black column if victim was Black
else if (identical(temp$PERP_RACE[i], "WHITE")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 1
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates White column if victim was white
else if (identical(temp$PERP_RACE[i], "WHITE HISPANIC")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 1
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates White Hispanic column if victim was white Hispanic
else if (identical(temp$PERP_RACE[i], "BLACK HISPANIC")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 1
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates Black Hispanic column if victim was Black Hispanic
else if (identical(temp$PERP_RACE[i], "ASIAN / PACIFIC ISLANDER")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 1
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates Asian/Pacific Islander column if victim was Asian or a Pacific Islander
else if (identical(temp$PERP_RACE[i], "AMERICAN INDIAN/ALASKAN NATIVE")) {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 1
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 0
} #updates American Indian/Alaskan Native column if victim was American Indian/Alaskan Native
else if (identical(temp$PERP_RACE[i], "UNKNOWN")) {

```

```
temp$BLACK[i] <- 0
temp$WHITE[i] <- 0
temp$WHITE_HISPANIC[i] <- 0
temp$BLACK_HISPANIC[i] <- 0
temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
temp$UNKNOWN_RACE[i] <- 1
temp$UNLISTED_RACE[i] <- 0
} #updates Unknown Race column if victim race was unknown
else {
  temp$BLACK[i] <- 0
  temp$WHITE[i] <- 0
  temp$WHITE_HISPANIC[i] <- 0
  temp$BLACK_HISPANIC[i] <- 0
  temp$ASIAN_PACIFIC_ISLANDER[i] <- 0
  temp$AMERICAN_INDIAN_ALASKAN_NATIVE[i] <- 0
  temp$UNKNOWN_RACE[i] <- 0
  temp$UNLISTED_RACE[i] <- 1
}
} #updates race columns with binary 1/0 depending on if the victim was that race (1) or was not that race
```

Next we will expand the columns for the age groups for the perpetrators.

#"65+" "18-24" "25-44" "<18" "45-64" "UNKNOWN"

```
temp$SixtyFivePlus <- temp$PERP_AGE_GROUP  
temp$FortyFive_SixtyFour <- temp$PERP_AGE_GROUP  
temp$TwentyFive_FortyFour <- temp$PERP_AGE_GROUP  
temp$Eighteen_TwentyFour <- temp$PERP_AGE_GROUP  
temp$Below_Eighteen <- temp$PERP_AGE_GROUP  
temp$Unknown_Age <- temp$PERP_AGE_GROUP  
temp$Unlisted_Age <- temp$PERP_AGE_GROUP
```

```
for (i in 1:k) {  
  if (identical(temp$PERP_AGE_GROUP[i], "65+")){  
    temp$SixtyFivePlus[i] <- 1  
    temp$FortyFive_SixtyFour[i] <- 0  
    temp$TwentyFive_FortyFour[i] <- 0  
    temp$Eighteen_TwentyFour[i] <- 0  
    temp$Below_Eighteen[i] <- 0  
    temp$Unknown_Age[i] <- 0  
    temp$Unlisted_Age[i] <- 0  
  }  
  else if (identical(temp$PERP_AGE_GROUP[i], "45-64")){  
    temp$SixtyFivePlus[i] <- 0  
    temp$FortyFive_SixtyFour[i] <- 1  
    temp$TwentyFive_FortyFour[i] <- 0  
    temp$Eighteen_TwentyFour[i] <- 0  
    temp$Below_Eighteen[i] <- 0  
    temp$Unknown_Age[i] <- 0  
    temp$Unlisted_Age[i] <- 0  
  }  
  else if (identical(temp$PERP_AGE_GROUP[i], "25-44")){  
    temp$SixtyFivePlus[i] <- 0  
    temp$FortyFive_SixtyFour[i] <- 0  
    temp$TwentyFive_FortyFour[i] <- 0  
    temp$Eighteen_TwentyFour[i] <- 0  
    temp$Below_Eighteen[i] <- 1  
    temp$Unknown_Age[i] <- 0  
    temp$Unlisted_Age[i] <- 0  
  }  
}
```

```

temp$SixtyFivePlus[i] <- 0
temp$FortyFive_SixtyFour[i] <- 0
temp$TwentyFive_FortyFour[i] <- 1
temp$Eighteen_TwentyFour[i] <- 0
temp$Below_Eighteen[i] <- 0
temp$Unknown_Age[i] <- 0
temp$Unlisted_Age[i] <- 0
}
else if (identical(temp$PERP AGE GROUP[i], "18-24")){
  temp$SixtyFivePlus[i] <- 0
  temp$FortyFive_SixtyFour[i] <- 0
  temp$TwentyFive_FortyFour[i] <- 0
  temp$Eighteen_TwentyFour[i] <- 1
  temp$Below_Eighteen[i] <- 0
  temp$Unknown_Age[i] <- 0
  temp$Unlisted_Age[i] <- 0
}
else if (identical(temp$PERP AGE GROUP[i], "<18")){
  temp$SixtyFivePlus[i] <- 0
  temp$FortyFive_SixtyFour[i] <- 0
  temp$TwentyFive_FortyFour[i] <- 0
  temp$Eighteen_TwentyFour[i] <- 0
  temp$Below_Eighteen[i] <- 1
  temp$Unknown_Age[i] <- 0
  temp$Unlisted_Age[i] <- 0
}
else if (identical(temp$PERP AGE GROUP[i], "UNKNOWN")){
  temp$SixtyFivePlus[i] <- 0
  temp$FortyFive_SixtyFour[i] <- 0
  temp$TwentyFive_FortyFour[i] <- 0
  temp$Eighteen_TwentyFour[i] <- 0
  temp$Below_Eighteen[i] <- 0
  temp$Unknown_Age[i] <- 1
  temp$Unlisted_Age[i] <- 0
}
else {
  temp$SixtyFivePlus[i] <- 0
  temp$FortyFive_SixtyFour[i] <- 0
  temp$TwentyFive_FortyFour[i] <- 0
  temp$Eighteen_TwentyFour[i] <- 0
  temp$Below_Eighteen[i] <- 0
  temp$Unknown_Age[i] <- 0
  temp$Unlisted_Age[i] <- 1
}
}

Shooting_Perps <- temp %>% select(-c(PERP AGE GROUP, PERP SEX, PERP RACE))

```

Now, we'll make all the new columns (victim and perpetrator) numeric to allow for us to use them in calculations.

```
temp <- Shooting_Victims
```

```
temp <- transform(temp,
```

```

    BLACK = as.numeric(BLACK),
    BLACK_HISPANIC = as.numeric(BLACK_HISPANIC),
    WHITE = as.numeric(WHITE),
    WHITE_HISPANIC = as.numeric(WHITE_HISPANIC),
    ASIAN_PACIFIC_ISLANDER = as.numeric(ASIAN_PACIFIC_ISLANDER),
    AMERICAN_INDIAN_ALASKAN_NATIVE = as.numeric(AMERICAN_INDIAN_ALASKAN_NATIVE),
    UNKNOWN_RACE = as.numeric(UNKNOWN_RACE),
    UNLISTED_RACE = as.numeric(UNLISTED_RACE),
    STATISTICAL_MURDER_FLAG = as.numeric(STATISTICAL_MURDER_FLAG),
    Female = as.numeric(Female),
    Male = as.numeric(Male),
    Unlisted_Sex = as.numeric(Unlisted_Sex),
    Unknown_Sex = as.numeric(Unknown_Sex),
    Below_Eighteen = as.numeric(Below_Eighteen),
    Eighteen_TwentyFour = as.numeric(Eighteen_TwentyFour),
    TwentyFive_FortyFour = as.numeric(TwentyFive_FortyFour),
    FortyFive_SixtyFour = as.numeric(FortyFive_SixtyFour),
    SixtyFivePlus = as.numeric(SixtyFivePlus),
    Unlisted_Age = as.numeric(Unlisted_Age),
    Unknown_Age = as.numeric(Unknown_Age))
Shooting_Victims <- temp

temp <- Shooting_Perps

temp <- transform(temp,
    BLACK = as.numeric(BLACK),
    BLACK_HISPANIC = as.numeric(BLACK_HISPANIC),
    WHITE = as.numeric(WHITE),
    WHITE_HISPANIC = as.numeric(WHITE_HISPANIC),
    ASIAN_PACIFIC_ISLANDER = as.numeric(ASIAN_PACIFIC_ISLANDER),
    AMERICAN_INDIAN_ALASKAN_NATIVE = as.numeric(AMERICAN_INDIAN_ALASKAN_NATIVE),
    UNKNOWN_RACE = as.numeric(UNKNOWN_RACE),
    UNLISTED_RACE = as.numeric(UNLISTED_RACE),
    STATISTICAL_MURDER_FLAG = as.numeric(STATISTICAL_MURDER_FLAG),
    Female = as.numeric(Female),
    Male = as.numeric(Male),
    Unlisted_Sex = as.numeric(Unlisted_Sex),
    Unknown_Sex = as.numeric(Unknown_Sex),
    Below_Eighteen = as.numeric(Below_Eighteen),
    Eighteen_TwentyFour = as.numeric(Eighteen_TwentyFour),
    TwentyFive_FortyFour = as.numeric(TwentyFive_FortyFour),
    FortyFive_SixtyFour = as.numeric(FortyFive_SixtyFour),
    SixtyFivePlus = as.numeric(SixtyFivePlus),
    Unlisted_Age = as.numeric(Unlisted_Age),
    Unknown_Age = as.numeric(Unknown_Age))

Shooting_Perps <- temp
rm(temp)

```

Next we will create a subset for shooting victims per borough and shooting perpetrators per borough. This will group the information in totals by borough in each column, retaining date.

```

Vic_By_Boro <- Shooting_Victims %>%
  group_by(BORO,OCCUR_DATE, OCCUR_TIME, STATISTICAL_MURDER_FLAG) %>%

```

```

summarize(BLACK = sum(BLACK),
          BLACK_HISPANIC = sum(BLACK_HISPANIC),
          WHITE = sum(WHITE),
          WHITE_HISPANIC = sum(WHITE_HISPANIC),
          ASIAN_PACIFIC_ISLANDER = sum(ASIAN_PACIFIC_ISLANDER),
          AMERICAN_INDIAN_ALASKAN_NATIVE = sum(AMERICAN_INDIAN_ALASKAN_NATIVE),
          UNLISTED_RACE = sum(UNLISTED_RACE),
          UNKNOWN_RACE = sum(UNKNOWN_RACE),
          Female = sum(Female),
          Male = sum(Male),
          Unknown_Age = sum(Unknown_Age),
          Unlisted_Age = sum(Unlisted_Age),
          Unknown_Sex = sum(Unknown_Sex),
          Unlisted_Sex = sum(Unlisted_Sex),
          FortyFive_SixtyFour = sum(FortyFive_SixtyFour),
          Below_Eighteen = sum(Below_Eighteen),
          Eighteen_TwentyFour = sum(Eighteen_TwentyFour),
          TwentyFive_FortyFour = sum(TwentyFive_FortyFour),
          SixtyFivePlus = sum(SixtyFivePlus)) %>%
mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
select(OCCUR_DATE, OCCUR_TIME, BORO, BLACK, BLACK_HISPANIC, WHITE, WHITE_HISPANIC, AMERICAN_INDIAN_ALASKAN_NATIVE)
ungroup()

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

Perp_By_Boro <- Shooting_Perps %>%
  group_by(BORO, OCCUR_DATE, OCCUR_TIME, STATISTICAL_MURDER_FLAG) %>%
  summarize(BLACK = sum(BLACK),
            BLACK_HISPANIC = sum(BLACK_HISPANIC),
            WHITE = sum(WHITE),
            WHITE_HISPANIC = sum(WHITE_HISPANIC),
            ASIAN_PACIFIC_ISLANDER = sum(ASIAN_PACIFIC_ISLANDER),
            AMERICAN_INDIAN_ALASKAN_NATIVE = sum(AMERICAN_INDIAN_ALASKAN_NATIVE),
            UNLISTED_RACE = sum(UNLISTED_RACE),
            UNKNOWN_RACE = sum(UNKNOWN_RACE),
            Female = sum(Female),
            Male = sum(Male),
            Unknown_Age = sum(Unknown_Age),
            Unlisted_Age = sum(Unlisted_Age),
            Unknown_Sex = sum(Unknown_Sex),
            Unlisted_Sex = sum(Unlisted_Sex),
            FortyFive_SixtyFour = sum(FortyFive_SixtyFour),
            Below_Eighteen = sum(Below_Eighteen),
            Eighteen_TwentyFour = sum(Eighteen_TwentyFour),
            TwentyFive_FortyFour = sum(TwentyFive_FortyFour),
            SixtyFivePlus = sum(SixtyFivePlus)) %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
  select(OCCUR_DATE, OCCUR_TIME, BORO, BLACK, BLACK_HISPANIC, WHITE, WHITE_HISPANIC, AMERICAN_INDIAN_ALASKAN_NATIVE)
ungroup()

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

```

The prior chunk successfully created data subsets of the shooting victims and perpetrators grouped by

borough and date/time of the shooting.

Standardize Excel Census and Police Shooting Data Subsets

Next we will organize and standardize the cleaned census data subset to more closely match the NYPD demographic data categories without altering data integrity. We will change the column names and make all numbers listed into numeric data from character data, then add smaller age groups (for example, 25-34 and 35-44 into 25-44) in order to compare to the groups provided by police data without affecting integrity.

This will allow us to add in the population data as new columns for easy calculations and analysis.

```
census_join <- cleaner_sub_data_census %>% select(c(Bronx_Estimate,Brooklyn_Estimate, Manhattan_Estimate
#Isolates columns above from error data

colnames(census_join) <- c("BRONX","BROOKLYN","MANHATTAN","QUEENS","STATEN_ISLAND", "NEW_YORK_CITY") #replaces the original column names with the ones we want

census_join <- as.data.frame(t(census_join)) #transposes columns and rows

colnames(census_join) <- c("Total_Population", "Male_Pop", "Female_Pop", "Under_5_Pop", "Five_Nine_Pop"

temp <- census_join #stores census_join into temp as a dataframe in order to avoid remaking census_join

#Attempting to use an index number and loop to substitute all commas resulted in full lists forced into one column

temp$Total_Population <- gsub(",","",temp$Total_Population)

temp$Male_Pop <- gsub(",","",temp$Male_Pop)

temp$Female_Pop <- gsub(",","",temp$Female_Pop)

temp$Under_5_Pop <- gsub(",","",temp$Under_5_Pop)

temp$Five_Nine_Pop <- gsub(",","",temp$Five_Nine_Pop)

temp$Ten_Fourteen_Pop <- gsub(",","",temp$Ten_Fourteen_Pop)

temp$Fifteen_Nineteen_Pop <- gsub(",","",temp$Fifteen_Nineteen_Pop)

temp$Twenty_TwentyFour_Pop <- gsub(",","",temp$Twenty_TwentyFour_Pop)

temp$TwentyFive_ThirtyFour_Pop <- gsub(",","",temp$TwentyFive_ThirtyFour_Pop)

temp$ThirtyFive_FortyFour_Pop <- gsub(",","",temp$ThirtyFive_FortyFour_Pop)

temp$FortyFive_FiftyFour_Pop <- gsub(",","",temp$FortyFive_FiftyFour_Pop)

temp$FiftyFive_FiftyNine_Pop <- gsub(",","",temp$FiftyFive_FiftyNine_Pop)

temp$Sixty_SixtyFour_Pop <- gsub(",","",temp$Sixty_SixtyFour_Pop)

temp$SixtyFive_SeventyFour_Pop <- gsub(",","",temp$SixtyFive_SeventyFour_Pop)

temp$SeventyFive_EightyFour_Pop <- gsub(",","",temp$SeventyFive_EightyFour_Pop)

temp$EightyFive_Plus_Pop <- gsub(",","",temp$EightyFive_Plus_Pop)
```

```

temp$BLACK_POP <- gsub(",","",temp$BLACK_POP)

temp$WHITE_POP <- gsub(",","",temp$WHITE_POP)

temp$WHITE_HISPANIC_POP <- gsub(",","",temp$WHITE_HISPANIC_POP)

temp$BLACK_HISPANIC_POP <- gsub(",","",temp$BLACK_HISPANIC_POP)

temp$ASIAN_PACIFIC_ISLANDER_POP <- gsub(",","",temp$ASIAN_PACIFIC_ISLANDER_POP)

temp$AMERICAN_INDIAN_ALASKAN_NATIVE_POP <- gsub(",","",temp$AMERICAN_INDIAN_ALASKAN_NATIVE_POP)

temp$HISPANIC_LATINO_ANY_RACE_POP <- gsub(",","",temp$HISPANIC_LATINO_ANY_RACE_POP)

temp$TWO_PLUS_RACES_POP <- gsub(",","",temp$TWO_PLUS_RACES_POP)

temp$UNKNOWN_OTHER_RACE_POP <- gsub(",","",temp$UNKNOWN_OTHER_RACE_POP)
#These commands "replaced" all commas in the desired columns with no character, effectively deleting them

temp <- transform(temp,
                  Total_Population = as.numeric(Total_Population),
                  Male_Pop = as.numeric(Male_Pop),
                  Female_Pop = as.numeric(Female_Pop),
                  Under_5_Pop = as.numeric(Under_5_Pop),
                  Five_Nine_Pop = as.numeric(Five_Nine_Pop),
                  Ten_Fourteen_Pop = as.numeric(Ten_Fourteen_Pop),
                  Fifteen_Nineteen_Pop = as.numeric(Fifteen_Nineteen_Pop),
                  Twenty_TwentyFour_Pop = as.numeric(Twenty_TwentyFour_Pop),
                  TwentyFive_ThirtyFour_Pop = as.numeric(TwentyFive_ThirtyFour_Pop),
                  ThirtyFive_FortyFour_Pop = as.numeric(ThirtyFive_FortyFour_Pop),
                  FortyFive_FiftyFour_Pop = as.numeric(FortyFive_FiftyFour_Pop),
                  FiftyFive_FiftyNine_Pop = as.numeric(FiftyFive_FiftyNine_Pop),
                  Sixty_SixtyFour_Pop = as.numeric(Sixty_SixtyFour_Pop),
                  SixtyFive_SeventyFour_Pop = as.numeric(SixtyFive_SeventyFour_Pop),
                  SeventyFive_EightyFour_Pop = as.numeric(SeventyFive_EightyFour_Pop),
                  EightyFive_Plus_Pop = as.numeric(EightyFive_Plus_Pop),
                  WHITE_POP = as.numeric(WHITE_POP),
                  BLACK_POP = as.numeric(BLACK_POP),
                  AMERICAN_INDIAN_ALASKAN_NATIVE_POP = as.numeric(AMERICAN_INDIAN_ALASKAN_NATIVE_POP),
                  ASIAN_PACIFIC_ISLANDER_POP = as.numeric(ASIAN_PACIFIC_ISLANDER_POP),
                  UNKNOWN_OTHER_RACE_POP = as.numeric(UNKNOWN_OTHER_RACE_POP),
                  TWO_PLUS_RACES_POP = as.numeric(TWO_PLUS_RACES_POP),
                  HISPANIC_LATINO_ANY_RACE_POP = as.numeric(HISPANIC_LATINO_ANY_RACE_POP),
                  WHITE_HISPANIC_POP = as.numeric(WHITE_HISPANIC_POP),
                  BLACK_HISPANIC_POP = as.numeric(BLACK_HISPANIC_POP))

#changes columns to numeric

temp <- temp %>% mutate(Below_Nineteen_Pop = Under_5_Pop + Five_Nine_Pop, TwentyFive_FortyFour_Pop = TwentyFive_ThirtyFour_Pop + Twenty_Four_Pop)
#adds together age group totals for larger age groups, retaining other columns

census_join <- temp #returns new organization to census_join

rm(temp) #removes temp from storage

```

Next we will simplify down to date of the shooting in another new subset.

```
Vic_By_Boro_Daily <- Vic_By_Boro %>%
  group_by(BORO, OCCUR_DATE, STATISTICAL_MURDER_FLAG) %>%
  summarize(BLACK = sum(BLACK),
            BLACK_HISPANIC = sum(BLACK_HISPANIC),
            WHITE = sum(WHITE),
            WHITE_HISPANIC = sum(WHITE_HISPANIC),
            ASIAN_PACIFIC_ISLANDER = sum(ASIAN_PACIFIC_ISLANDER),
            AMERICAN_INDIAN_ALASKAN_NATIVE = sum(AMERICAN_INDIAN_ALASKAN_NATIVE),
            UNLISTED_RACE = sum(UNLISTED_RACE),
            UNKNOWN_RACE = sum(UNKNOWN_RACE),
            Female = sum(Female),
            Male = sum(Male),
            Unknown_Age = sum(Unknown_Age),
            Unlisted_Age = sum(Unlisted_Age),
            Unknown_Sex = sum(Unknown_Sex),
            Unlisted_Sex = sum(Unlisted_Sex),
            FortyFive_SixtyFour = sum(FortyFive_SixtyFour),
            Below_Eighteen = sum(Below_Eighteen),
            Eighteen_TwentyFour = sum(Eighteen_TwentyFour),
            TwentyFive_FortyFour = sum(TwentyFive_FortyFour),
            SixtyFivePlus = sum(SixtyFivePlus)) %>%
  select(BORO,OCCUR_DATE,BLACK,BLACK_HISPANIC,WHITE, WHITE_HISPANIC, ASIAN_PACIFIC_ISLANDER, AMERICAN_INDIAN_ALASKAN_NATIVE)
ungroup()

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

#groups victims by borough, date, and murder flag

Perp_By_Boro_Daily <- Perp_By_Boro %>%
  group_by(BORO, OCCUR_DATE, STATISTICAL_MURDER_FLAG) %>%
  summarize(BLACK = sum(BLACK),
            BLACK_HISPANIC = sum(BLACK_HISPANIC),
            WHITE = sum(WHITE),
            WHITE_HISPANIC = sum(WHITE_HISPANIC),
            ASIAN_PACIFIC_ISLANDER = sum(ASIAN_PACIFIC_ISLANDER),
            AMERICAN_INDIAN_ALASKAN_NATIVE = sum(AMERICAN_INDIAN_ALASKAN_NATIVE),
            UNLISTED_RACE = sum(UNLISTED_RACE),
            UNKNOWN_RACE = sum(UNKNOWN_RACE),
            Female = sum(Female),
            Male = sum(Male),
            Unknown_Age = sum(Unknown_Age),
            Unlisted_Age = sum(Unlisted_Age),
            Unknown_Sex = sum(Unknown_Sex),
            Unlisted_Sex = sum(Unlisted_Sex),
            FortyFive_SixtyFour = sum(FortyFive_SixtyFour),
            Below_Eighteen = sum(Below_Eighteen),
            Eighteen_TwentyFour = sum(Eighteen_TwentyFour),
            TwentyFive_FortyFour = sum(TwentyFive_FortyFour),
            SixtyFivePlus = sum(SixtyFivePlus)) %>%
  select(BORO,OCCUR_DATE,BLACK,BLACK_HISPANIC,WHITE, WHITE_HISPANIC, ASIAN_PACIFIC_ISLANDER, AMERICAN_INDIAN_ALASKAN_NATIVE)
ungroup()
```

```
## `summarise()` has grouped output by 'BORO', 'OCCUR_DATE'. You can override
## using the `groups` argument.
#groups perpetrators by boro, date, and murder flag
```

Now we will prepare to join the data sets.

```
census_join$BORO <- census_join$Male_Pop
census_join$BORO[1] <- "BRONX"
census_join$BORO[2] <- "BROOKLYN"
census_join$BORO[3] <- "MANHATTAN"
census_join$BORO[4] <- "QUEENS"
census_join$BORO[5] <- "STATEN ISLAND"
census_join$BORO[6] <- "NEW YORK CITY"
#This adds in a column that contains the borough names in the correct rows to allow for an easier group

Vic_By_Boro_Daily_with_Pop <- Vic_By_Boro_Daily %>%
  left_join(census_join, by = c("BORO")) #This joins the census data grouped by borough

Perp_By_Boro_Daily_with_Pop <- Perp_By_Boro_Daily %>%
  left_join(census_join, by = c("BORO")) #This joins the census data grouped by borough
```

Transform the Data

Now we can finally start looking at transformations of the datasets. First we will create some percentages of the demographics based on borough.

```
Perp_By_Boro_Totals_with_Pop <- Perp_By_Boro_Daily_with_Pop %>%
  group_by(BORO) %>%
  summarize(BLACK = sum(BLACK),
            BLACK_HISPANIC = sum(BLACK_HISPANIC),
            WHITE = sum(WHITE),
            WHITE_HISPANIC = sum(WHITE_HISPANIC),
            ASIAN_PACIFIC_ISLANDER = sum(ASIAN_PACIFIC_ISLANDER),
            AMERICAN_INDIAN_ALASKAN_NATIVE = sum(AMERICAN_INDIAN_ALASKAN_NATIVE),
            UNLISTED_RACE = sum(UNLISTED_RACE),
            UNKNOWN_RACE = sum(UNKNOWN_RACE),
            Female = sum(Female),
            Male = sum(Male),
            Unknown_Age = sum(Unknown_Age),
            Unlisted_Age = sum(Unlisted_Age),
            Unknown_Sex = sum(Unknown_Sex),
            Unlisted_Sex = sum(Unlisted_Sex),
            FortyFive_SixtyFour = sum(FortyFive_SixtyFour),
            Below_Eighteen = sum(Below_Eighteen),
            Eighteen_TwentyFour = sum(Eighteen_TwentyFour),
            TwentyFive_FortyFour = sum(TwentyFive_FortyFour),
            SixtyFivePlus = sum(SixtyFivePlus),
            Totals = sum(Female) + sum(Male) + sum(Unknown_Sex) + sum(Unlisted_Sex)) %>%
  left_join(census_join, by = c("BORO")) %>%
  ungroup()
#This creates a totals summary of the perpetrators by borough with the population census data joined
```

```
Vic_By_Boro_Totals_with_Pop <- Vic_By_Boro_Daily_with_Pop %>%
  group_by(BORO) %>%
```

```

summarize(BLACK = sum(BLACK),
          BLACK_HISPANIC = sum(BLACK_HISPANIC),
          WHITE = sum(WHITE),
          WHITE_HISPANIC = sum(WHITE_HISPANIC),
          ASIAN_PACIFIC_ISLANDER = sum(ASIAN_PACIFIC_ISLANDER),
          AMERICAN_INDIAN_ALASKAN_NATIVE = sum(AMERICAN_INDIAN_ALASKAN_NATIVE),
          UNLISTED_RACE = sum(UNLISTED_RACE),
          UNKNOWN_RACE = sum(UNKNOWN_RACE),
          Female = sum(Female),
          Male = sum(Male),
          Unknown_Age = sum(Unknown_Age),
          Unlisted_Age = sum(Unlisted_Age),
          Unknown_Sex = sum(Unknown_Sex),
          Unlisted_Sex = sum(Unlisted_Sex),
          FortyFive_SixtyFour = sum(FortyFive_SixtyFour),
          Below_Eighteen = sum(Below_Eighteen),
          Eighteen_TwentyFour = sum(Eighteen_TwentyFour),
          TwentyFive_FortyFour = sum(TwentyFive_FortyFour),
          SixtyFivePlus = sum(SixtyFivePlus),
          Totals = sum(Female) + sum(Male) + sum(Unknown_Sex) + sum(Unlisted_Sex)) %>%
left_join(census_join, by = c("BORO")) %>%
ungroup()
#creates dataframe with shooting victims totals per boro

```

Next we will isolate age group into its own subset for visualization.

```

Vic_Ages <- Vic_By_Boro_Totals_with_Pop %>%
  mutate(Below_TwentyFive = Below_Eighteen + Eighteen_TwentyFour,
         Below_TwentyFive_Pop = Below_Nineteen_Pop + Twenty_TwentyFour_Pop) %>%
  pivot_longer(cols = c(Below_TwentyFive,TwentyFive_FortyFour,FortyFive_SixtyFour,SixtyFivePlus),
               names_to = "Age_Group",
               values_to = "Incidents") %>%
  pivot_longer(cols = c(Below_TwentyFive_Pop, TwentyFive_FortyFour_Pop, FortyFive_SixtyFour_Pop, SixtyFivePlus_Pop),
               names_to = "Age_Group_Pop",
               values_to = "Population") %>%
  select(c(BORO, Age_Group, Incidents, Age_Group_Pop, Population))

#Pivots age groups and age group total populations into four columns for visualization

Perp_Ages <- Perp_By_Boro_Totals_with_Pop %>%
  mutate(Below_TwentyFive = Below_Eighteen + Eighteen_TwentyFour,
         Below_TwentyFive_Pop = Below_Nineteen_Pop + Twenty_TwentyFour_Pop) %>%
  pivot_longer(cols = c(Below_TwentyFive,TwentyFive_FortyFour,FortyFive_SixtyFour,SixtyFivePlus),
               names_to = "Age_Group",
               values_to = "Incidents") %>%
  pivot_longer(cols = c(Below_TwentyFive_Pop, TwentyFive_FortyFour_Pop, FortyFive_SixtyFour_Pop, SixtyFivePlus_Pop),
               names_to = "Age_Group_Pop",
               values_to = "Population") %>%
  select(c(BORO, Age_Group, Incidents, Age_Group_Pop, Population))
#Pivots age groups and age group total populations into four columns for visualization

Vic_Age_Percent <- Vic_By_Boro_Totals_with_Pop %>%
  mutate(Below_TwentyFive_Percent = (Below_Eighteen + Eighteen_TwentyFour)* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         TwentyFive_FortyFour_Percent = TwentyFive_FortyFour* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex))

```

```

FortyFive_SixtyFour_Percent = FortyFive_SixtyFour* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
SixtyFivePlus_Percent = SixtyFivePlus* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
#begin percents for population
Below_TwentyFive_Pop_Percent = (Below_Nineteen_Pop + Twenty_TwentyFour_Pop) * 100 / Total_Population,
TwentyFive_FortyFour_Pop_Percent = TwentyFive_FortyFour_Pop * 100 / Total_Population,
FortyFive_SixtyFour_Pop_Percent = FortyFive_SixtyFour_Pop * 100 / Total_Population,
SixtyFivePlus_Pop_Percent = SixtyFivePlus_Pop * 100 / Total_Population) %>%
pivot_longer(cols = c(Below_TwentyFive_Percent,TwentyFive_FortyFour_Percent,FortyFive_SixtyFour_Percent),
             names_to = "Age_Group_Percent",
             values_to = "Incidents_Percent") %>%
pivot_longer(cols = c(Below_TwentyFive_Pop_Percent, TwentyFive_FortyFour_Pop_Percent, FortyFive_SixtyFour_Pop_Percent),
             names_to = "Age_Group_Pop_Percent",
             values_to = "Population_Percent") %>%
select(c(BORO, Age_Group_Percent, Incidents_Percent, Age_Group_Pop_Percent, Population_Percent))
#Pivots age groups and age group total population percentages into four columns for visualization

Perp_Age_Percent <- Perp_By_Boro_Totals_with_Pop %>%
  mutate(Below_TwentyFive_Percent = (Below_Eighteen + Eighteen_TwentyFour) * 100 /(sum(Male) + sum(Female)),
         TwentyFive_FortyFour_Percent = TwentyFive_FortyFour* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         FortyFive_SixtyFour_Percent = FortyFive_SixtyFour * 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         SixtyFivePlus_Percent = SixtyFivePlus* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
#begin percents for population
         Below_TwentyFive_Pop_Percent = (Below_Nineteen_Pop + Twenty_TwentyFour_Pop) * 100 / Total_Population,
         TwentyFive_FortyFour_Pop_Percent = TwentyFive_FortyFour_Pop * 100 / Total_Population,
         FortyFive_SixtyFour_Pop_Percent = FortyFive_SixtyFour_Pop * 100 / Total_Population,
         SixtyFivePlus_Pop_Percent = SixtyFivePlus_Pop * 100 / Total_Population) %>%
pivot_longer(cols = c(Below_TwentyFive_Percent,TwentyFive_FortyFour_Percent,FortyFive_SixtyFour_Percent),
             names_to = "Age_Group_Percent",
             values_to = "Incidents_Percent") %>%
pivot_longer(cols = c(Below_TwentyFive_Pop_Percent, TwentyFive_FortyFour_Pop_Percent, FortyFive_SixtyFour_Pop_Percent),
             names_to = "Age_Group_Pop_Percent",
             values_to = "Population_Percent") %>%
select(c(BORO, Age_Group_Percent, Incidents_Percent, Age_Group_Pop_Percent, Population_Percent))
#Pivots age groups and age group total population percentages into four columns for visualization

```

Now that we have tidied and transformed our data, we can begin to visualize it.

Visualizing the Data

We've prepped our data to look at shooters and victims by demographic compared to population demographic per borough and also at shootings and murders per day and per time of day in each borough.

Visualizing Victims and Shooters by Demographic Groups

Our data is tidied to allow for visualizing the victims and shooters by age, race, and sex, grouped by borough.

First we will look at age of victims and shooters per borough as raw numbers.

```

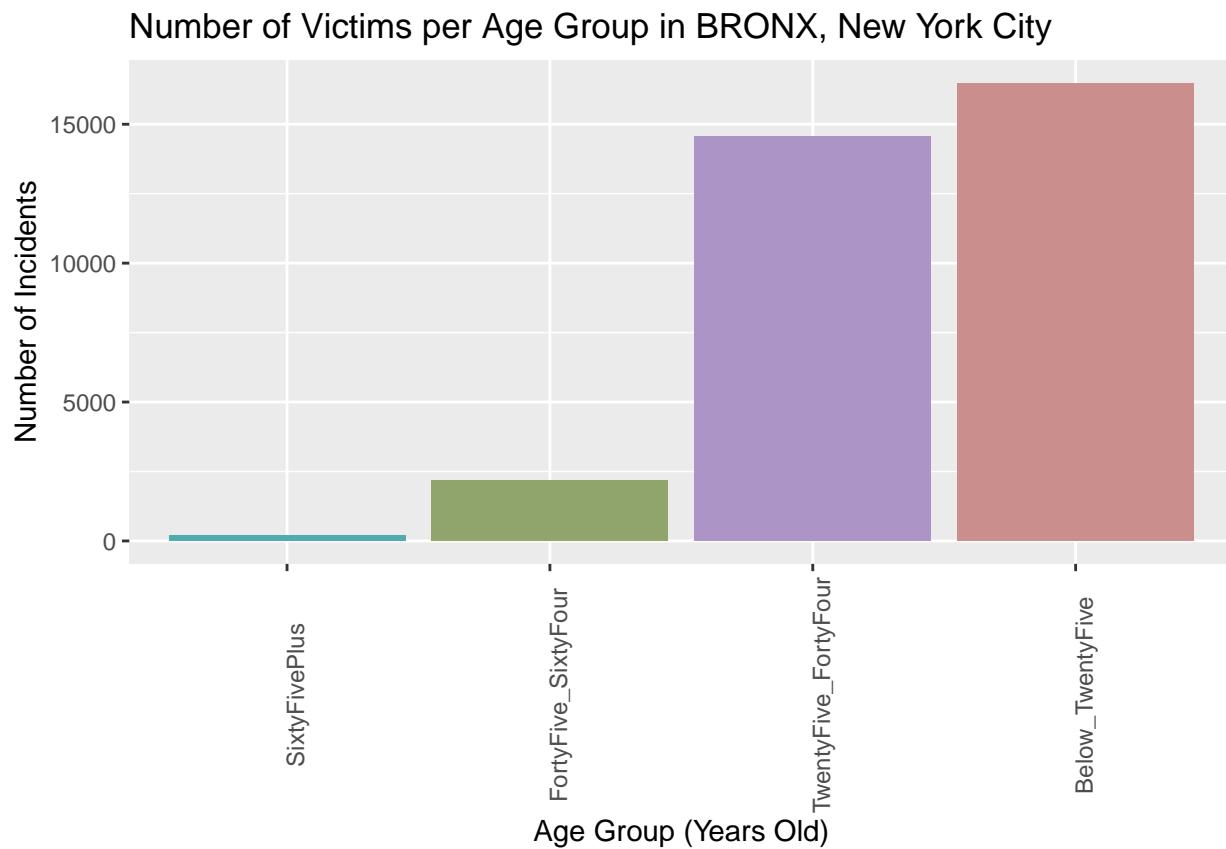
boro <- "BRONX"
Vic_Ages %>%
  filter(BORO == boro) %>%
ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +

```

```

ylab("Number of Incidents") +
xlab("Age Group (Years Old)") +
labs(title = str_c("Number of Victims per Age Group in ", boro, ", New York City"))

```



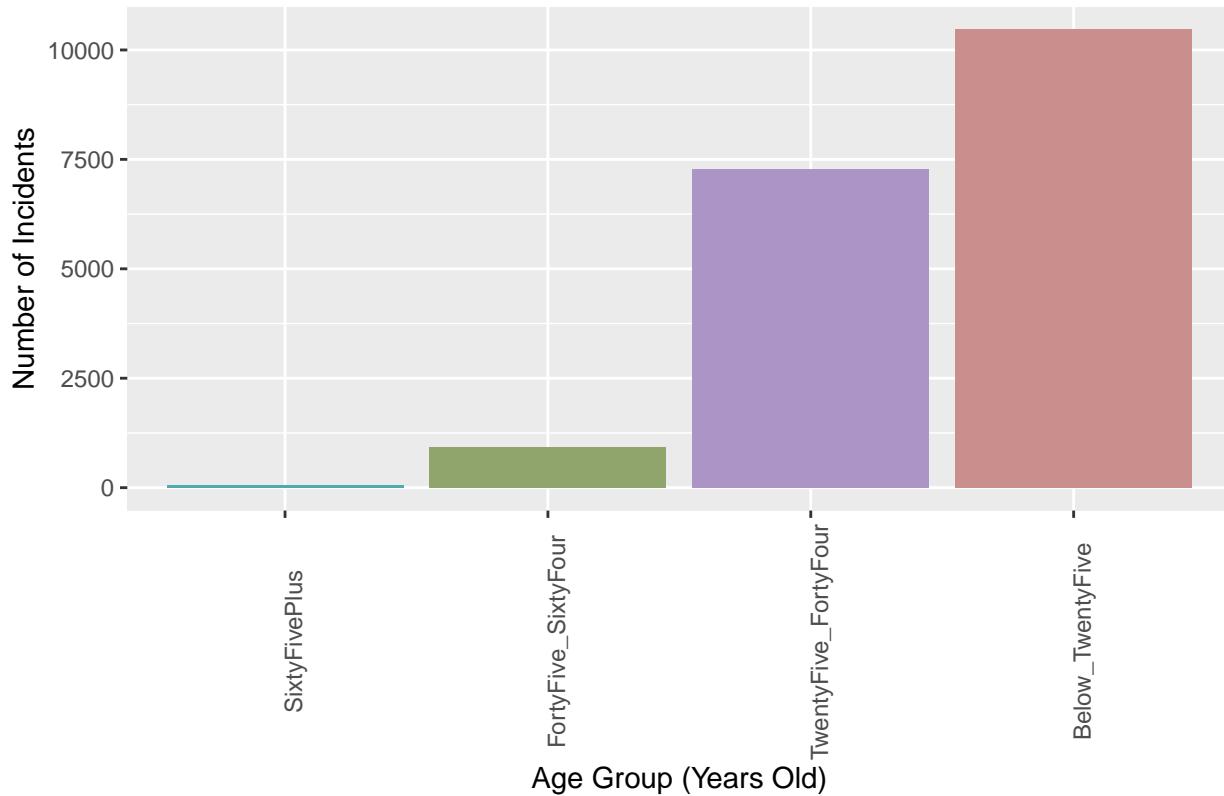
```
#creates a bar graph showing the number of victims per age group in the selected borough
```

```

Perp_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Shooters per Age Group in ", boro, ", New York City"))

```

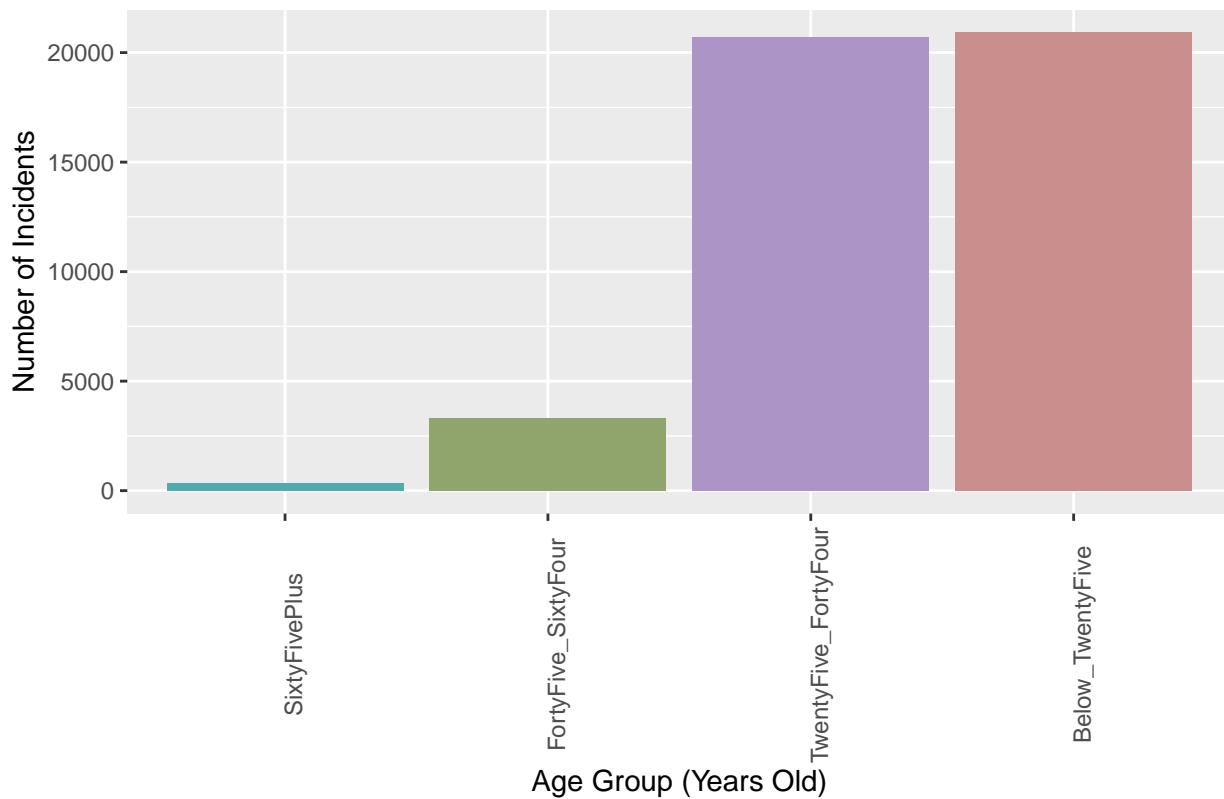
Number of Shooters per Age Group in BRONX, New York City



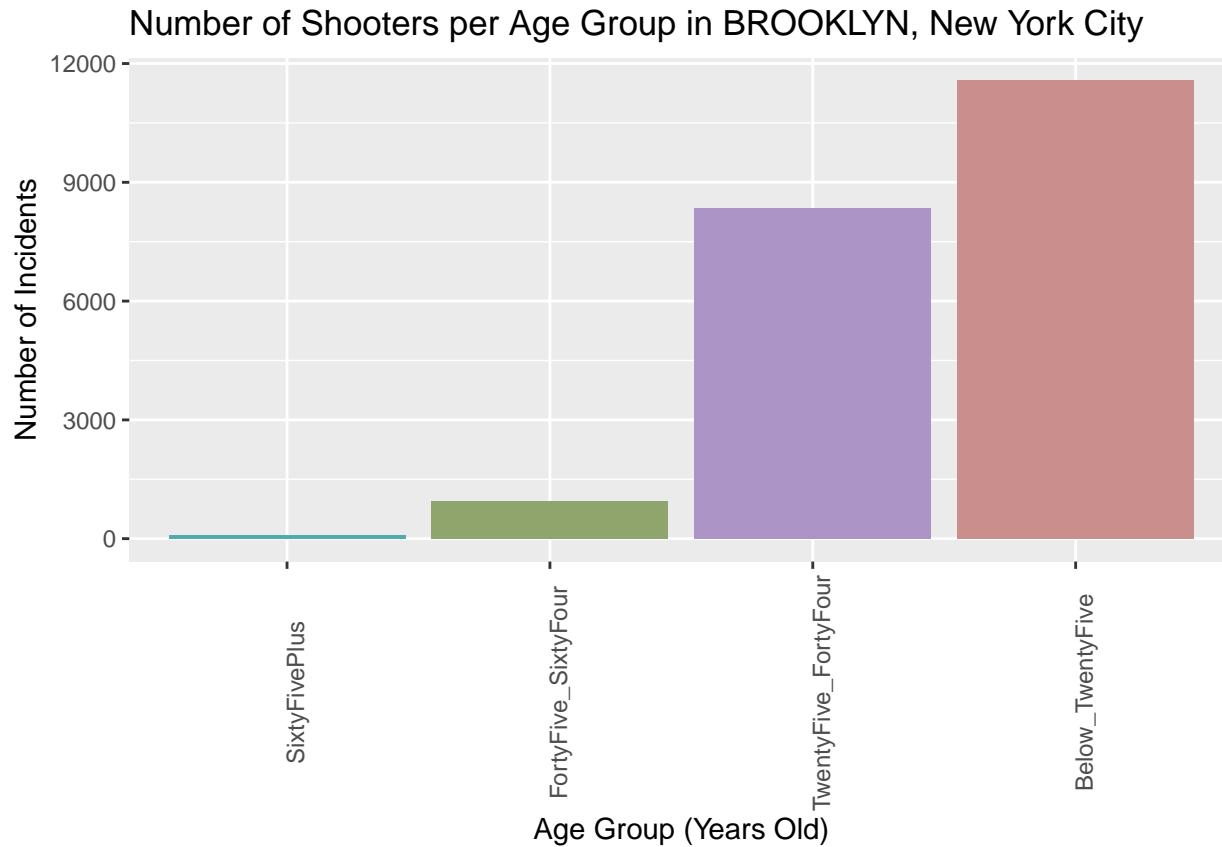
```
#creates a bar graph showing the number of shooters per age group in the selected borough
boro <- "BROOKLYN"

Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Victims per Age Group in ", boro, ", New York City"))
```

Number of Victims per Age Group in BROOKLYN, New York City



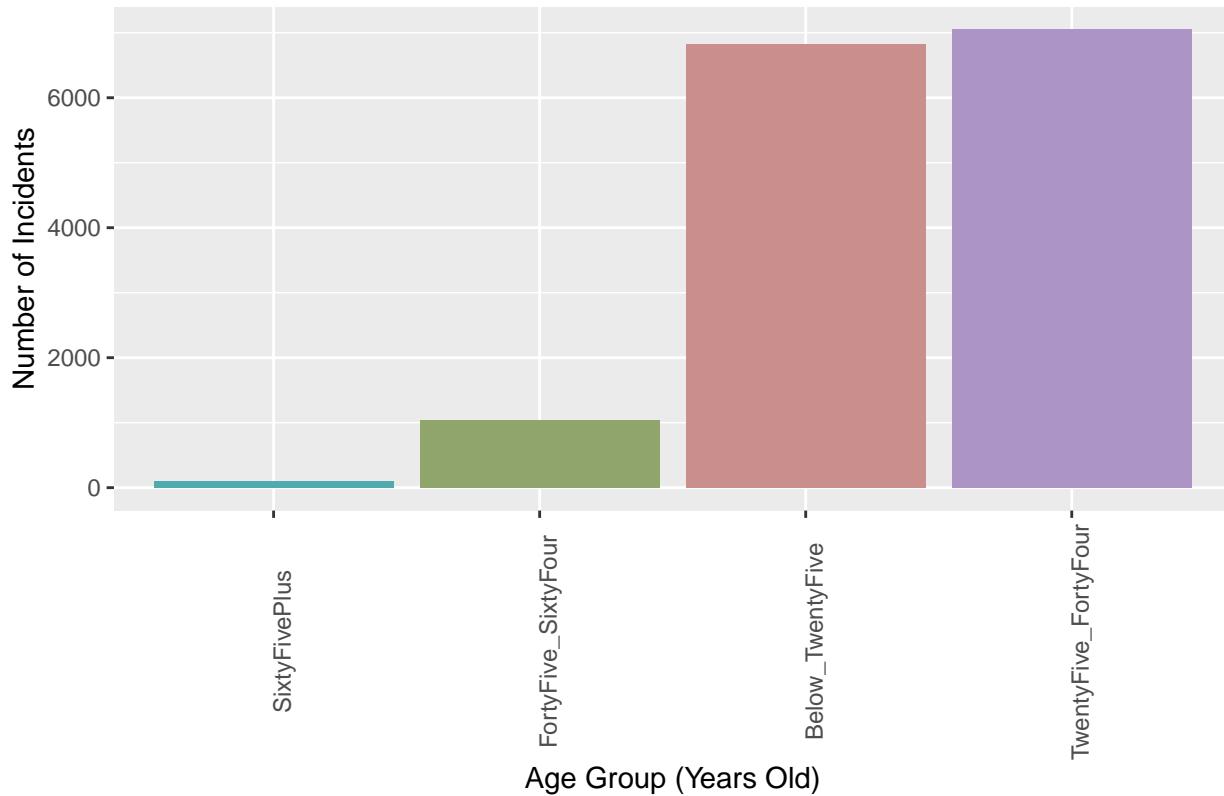
```
#creates a bar graph showing the number of victims per age group in the selected borough
Perp_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Shooters per Age Group in ", boro, ", New York City"))
```



```
#creates a bar graph showing the number of shooters per age group in the selected borough
boro <- "MANHATTAN"

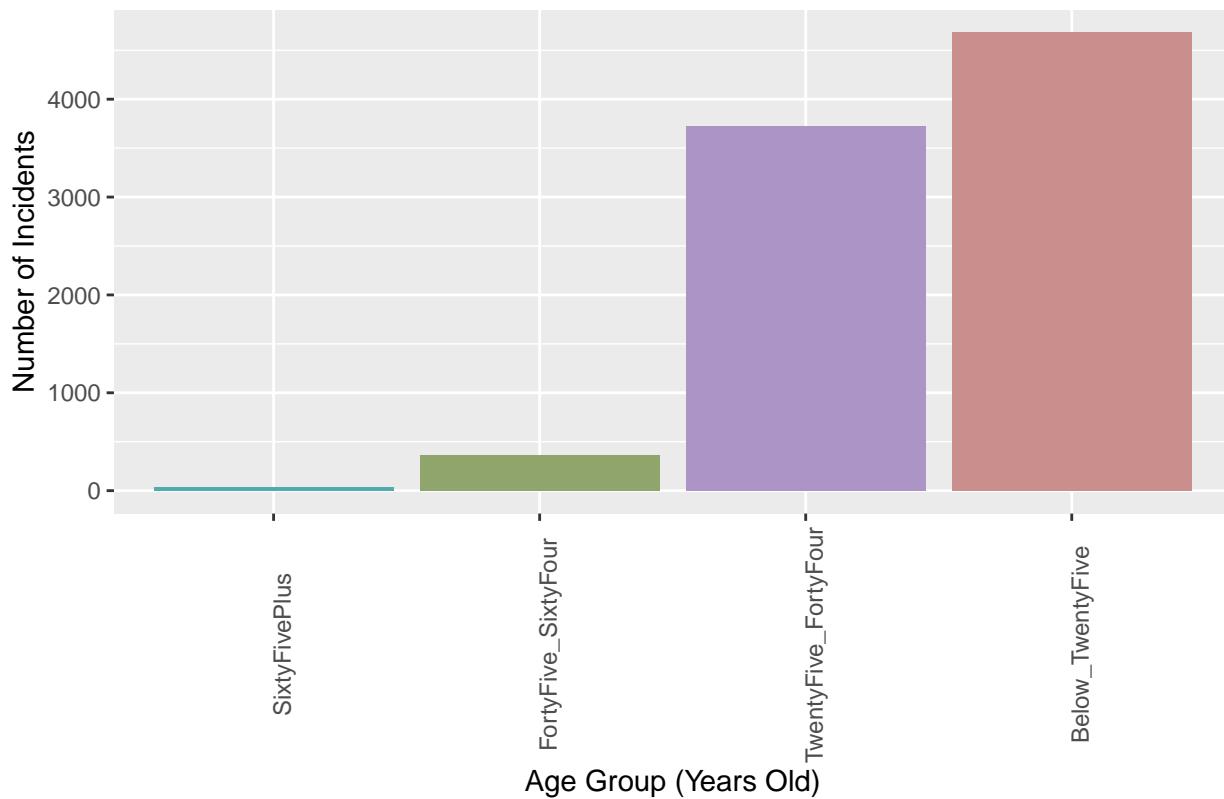
Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Victims per Age Group in ", boro, ", New York City"))
```

Number of Victims per Age Group in MANHATTAN, New York City



```
#creates a bar graph showing the number of victims per age group in the selected borough
Perp_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Shooters per Age Group in ", boro, ", New York City"))
```

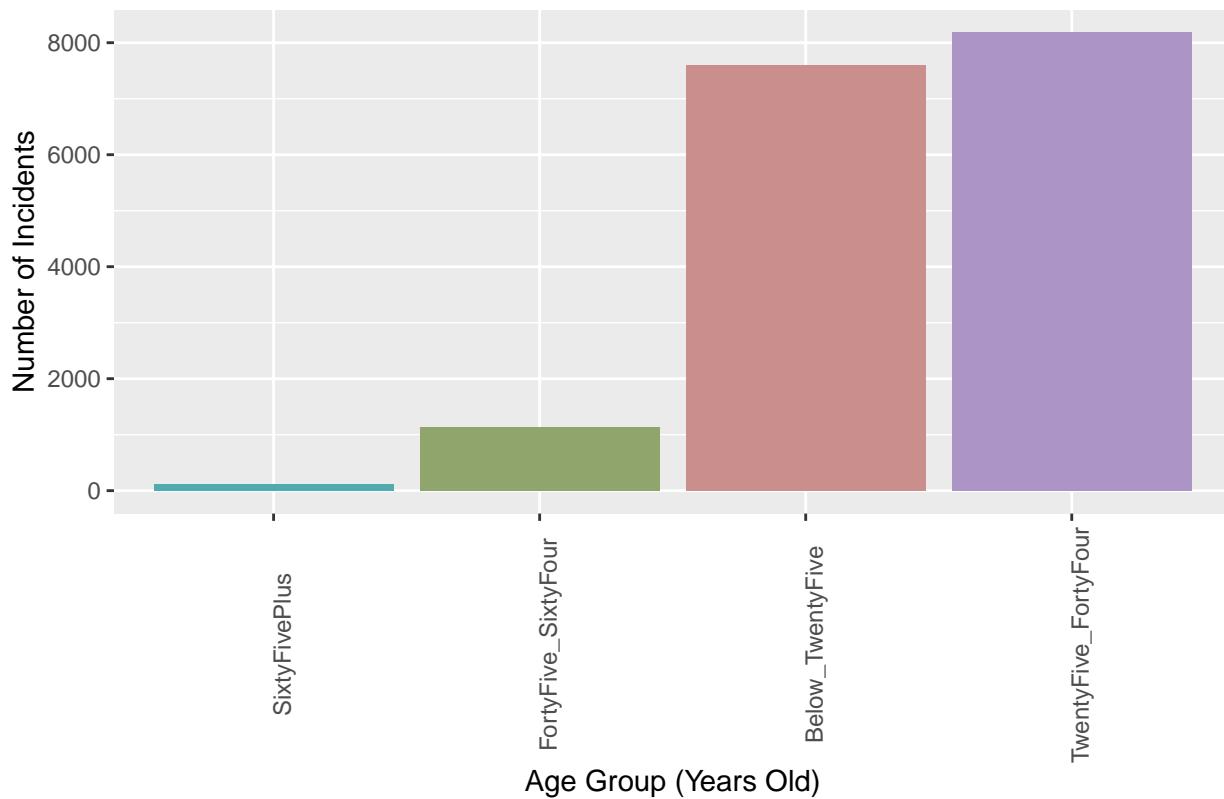
Number of Shooters per Age Group in MANHATTAN, New York City



```
#creates a bar graph showing the number of shooters per age group in the selected borough
boro <- "QUEENS"

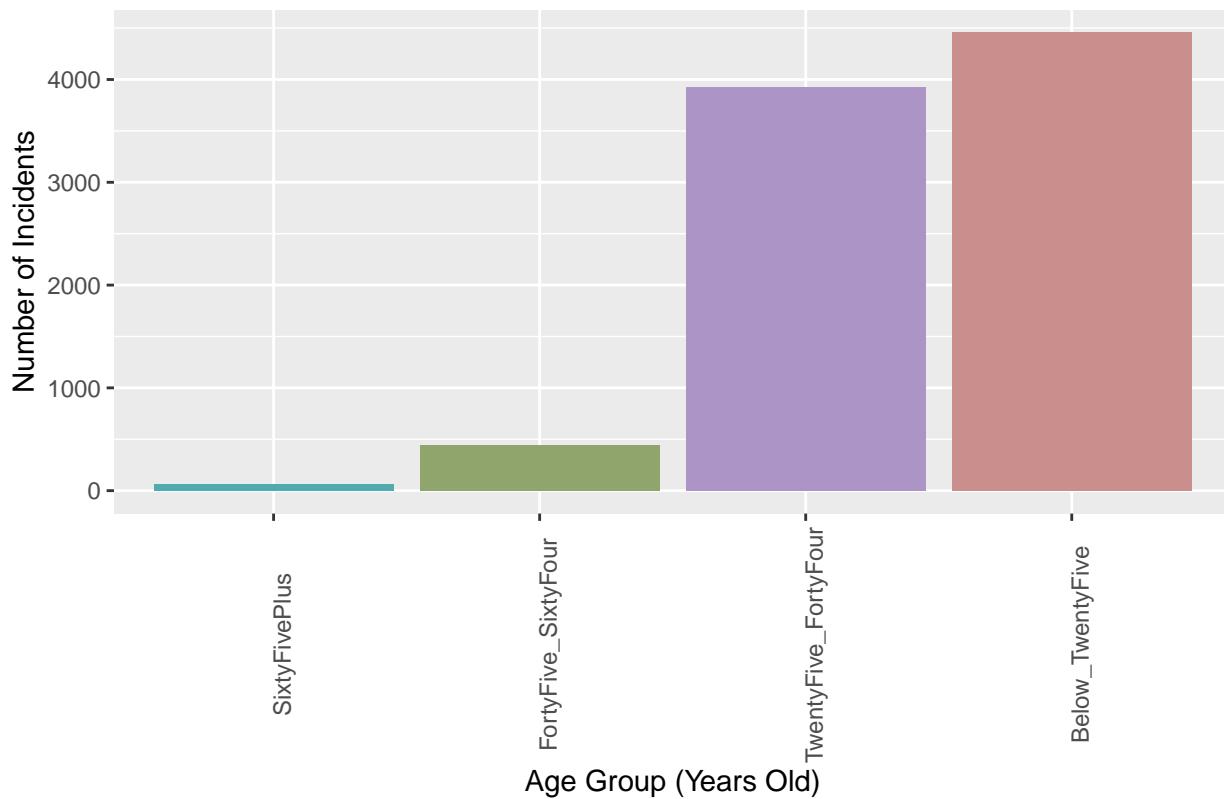
Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Victims per Age Group in ", boro, ", New York City"))
```

Number of Victims per Age Group in QUEENS, New York City



```
#creates a bar graph showing the number of victims per age group in the selected borough
Perp_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Shooters per Age Group in ", boro, ", New York City"))
```

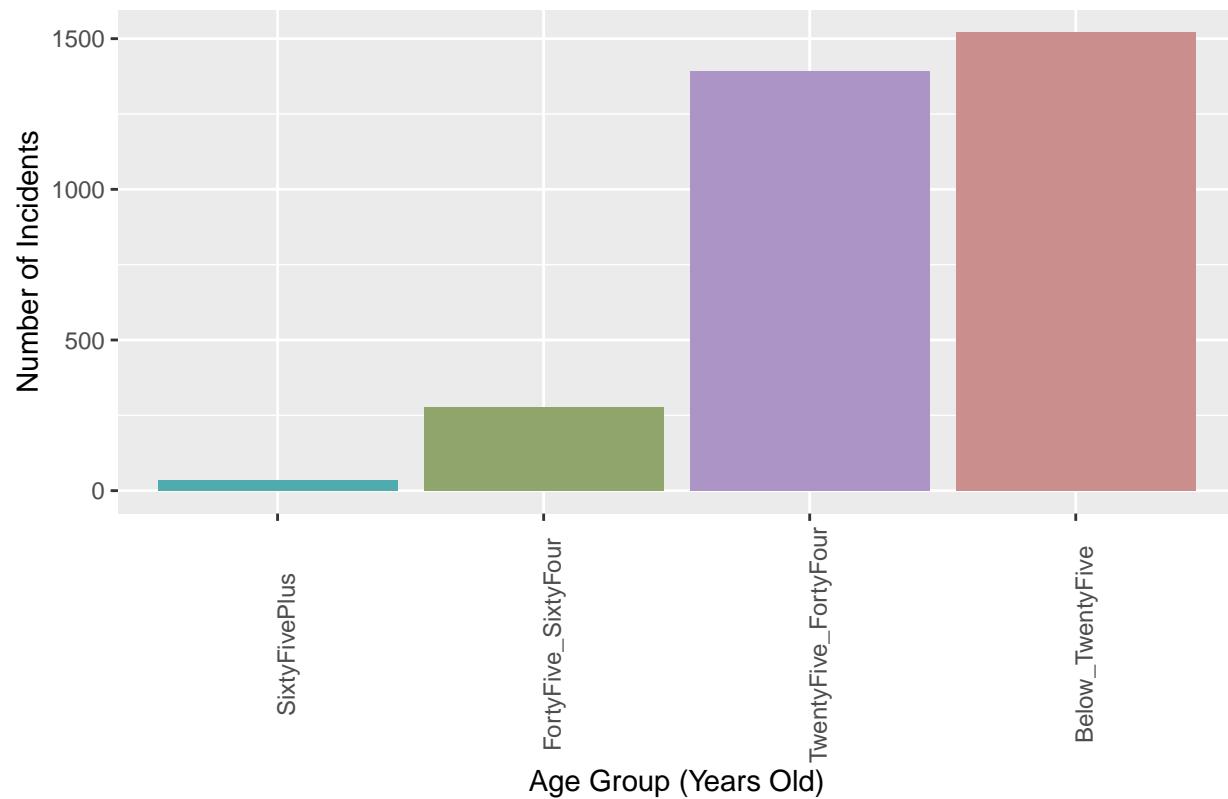
Number of Shooters per Age Group in QUEENS, New York City



```
#creates a bar graph showing the number of shooters per age group in the selected borough
boro <- "STATEN ISLAND"

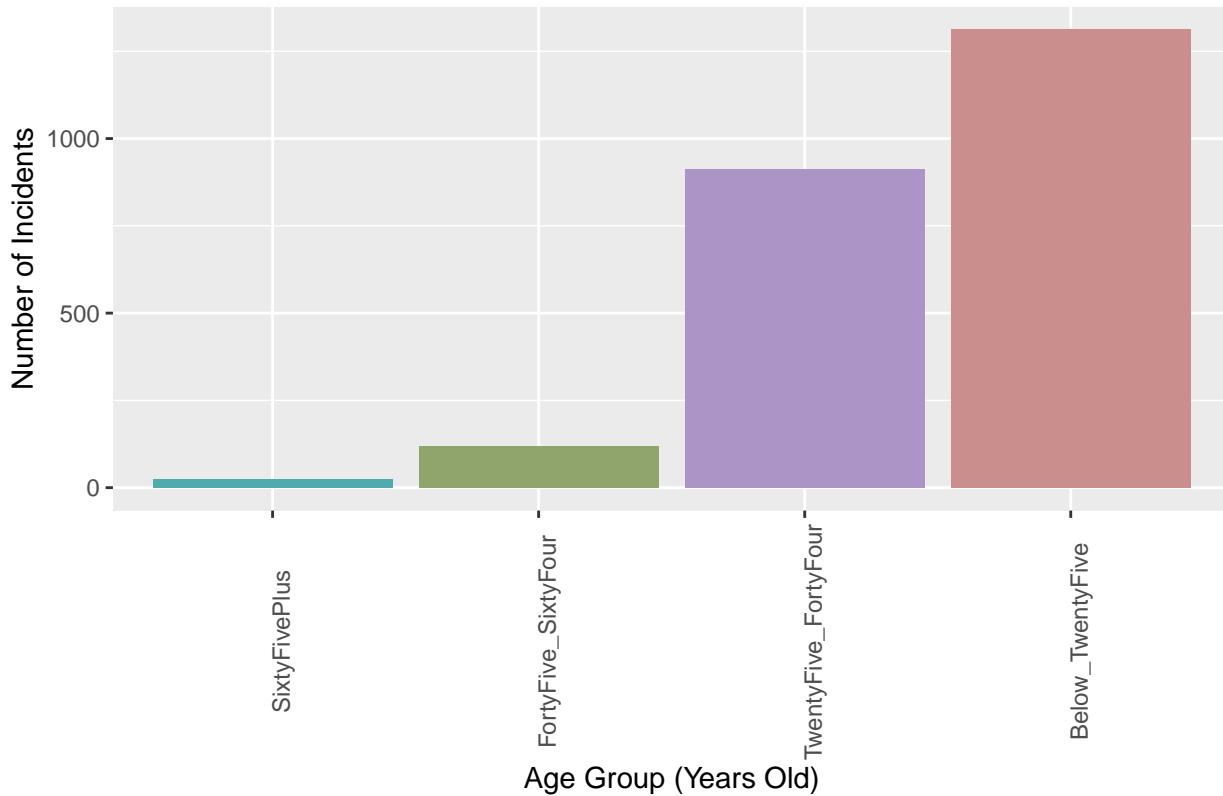
Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Victims per Age Group in ", boro, ", New York City"))
```

Number of Victims per Age Group in STATEN ISLAND, New York City



```
#creates a bar graph showing the number of victims per age group in the selected borough
Perp_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group, Incidents)), fill = as.factor(Age_Group), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Number of Shooters per Age Group in ", boro, ", New York City"))
```

Number of Shooters per Age Group in STATEN ISLAND, New York City



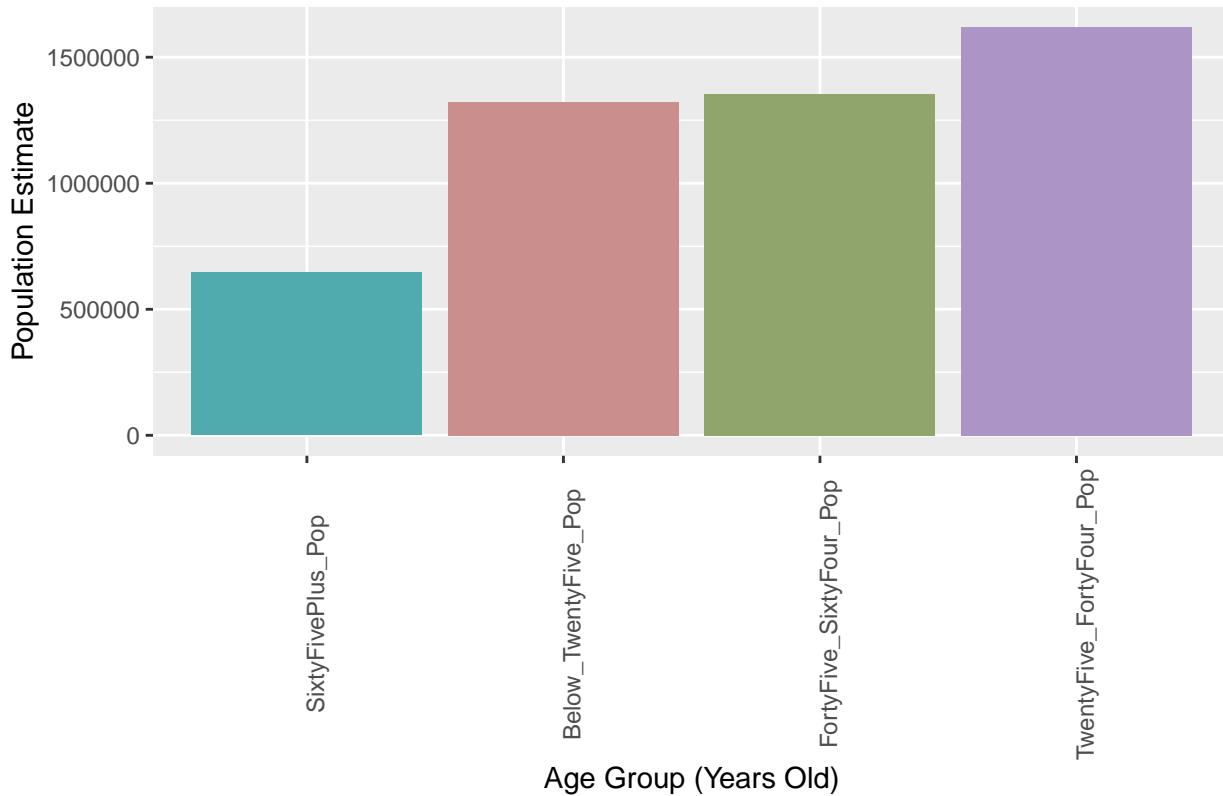
```
#creates a bar graph showing the number of shooters per age group in the selected borough
```

We combined the under 18 groups and 18-24 groups due to the census data showing a division at 19 between 0 and 24, which allows us a better comparison later since both groups will have “Below 25” instead of disparate subgroups.

From this first set of visualizations, it would seem that most shooters *and* victims come from the Below 25 and 25-44 age groups. However, we don’t know how much of the population is in those age groups, so we can’t tell if this is due to these age groups being more involved in violence, or if they are simply more numerous than the other age groups. Let’s look at their populations in each borough.

```
boro <- "BRONX"
Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group_Pop, Population)), fill = as.factor(Age_Group_Pop), y = Pop))
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population Estimate") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Population Estimate per Age Group in ", boro, ", New York City"))
```

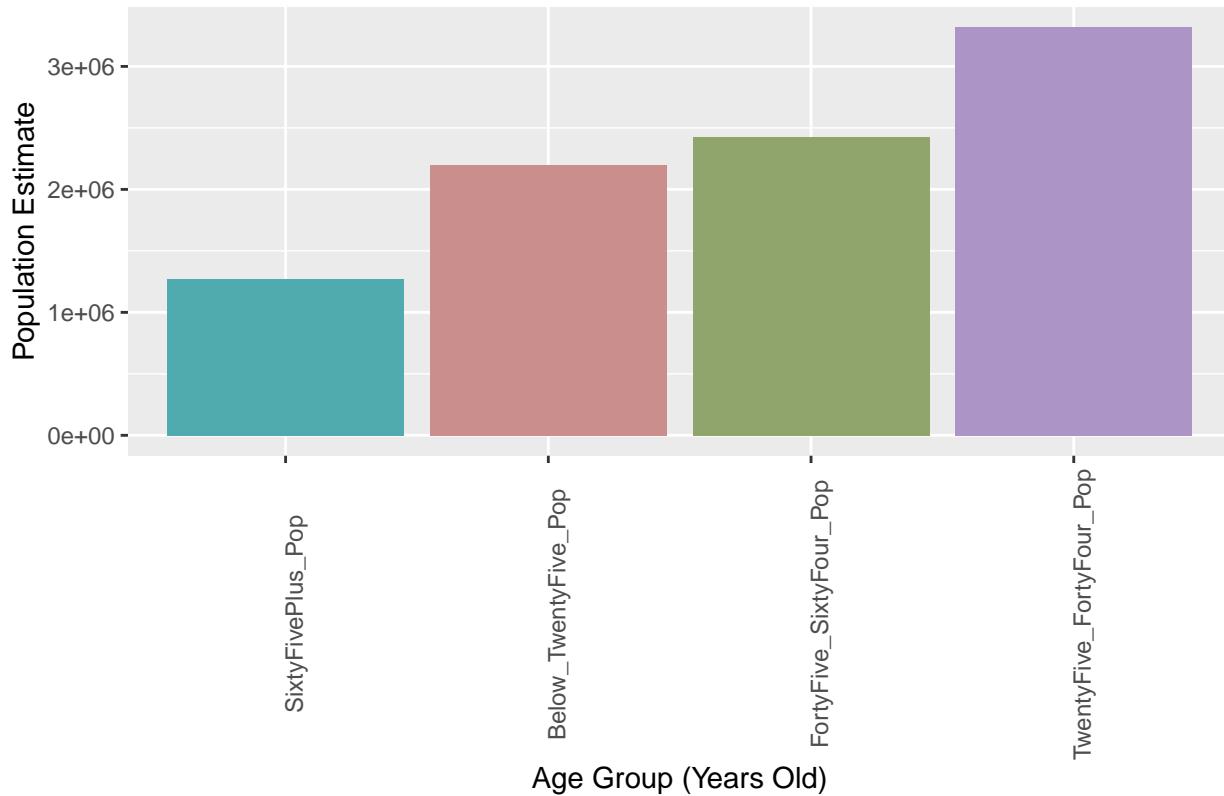
Population Estimate per Age Group in BRONX, New York City



```
#creates bar graph of population estimates per age group in selected borough

boro <- "BROOKLYN"
Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group_Pop, Population)), fill = as.factor(Age_Group_Pop), y = Population),
         theme(legend.position="none",
               axis.text.x = element_text(angle = 90)) +
  ylab("Population Estimate") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Population Estimate per Age Group in ", boro, ", New York City"))
```

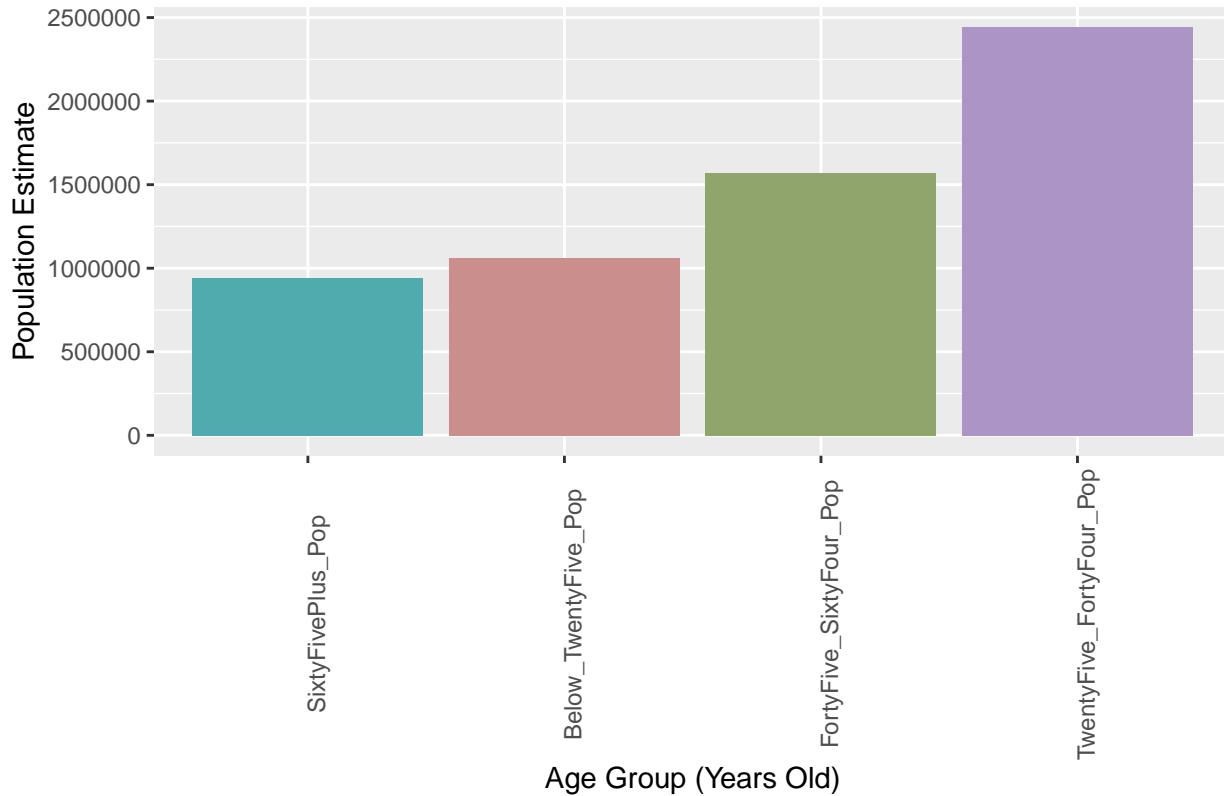
Population Estimate per Age Group in BROOKLYN, New York City



```
#creates bar graph of population estimates per age group in selected borough

boro <- "MANHATTAN"
Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group_Pop, Population)), fill = as.factor(Age_Group_Pop), y = Population),
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population Estimate") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Population Estimate per Age Group in ", boro, ", New York City"))
```

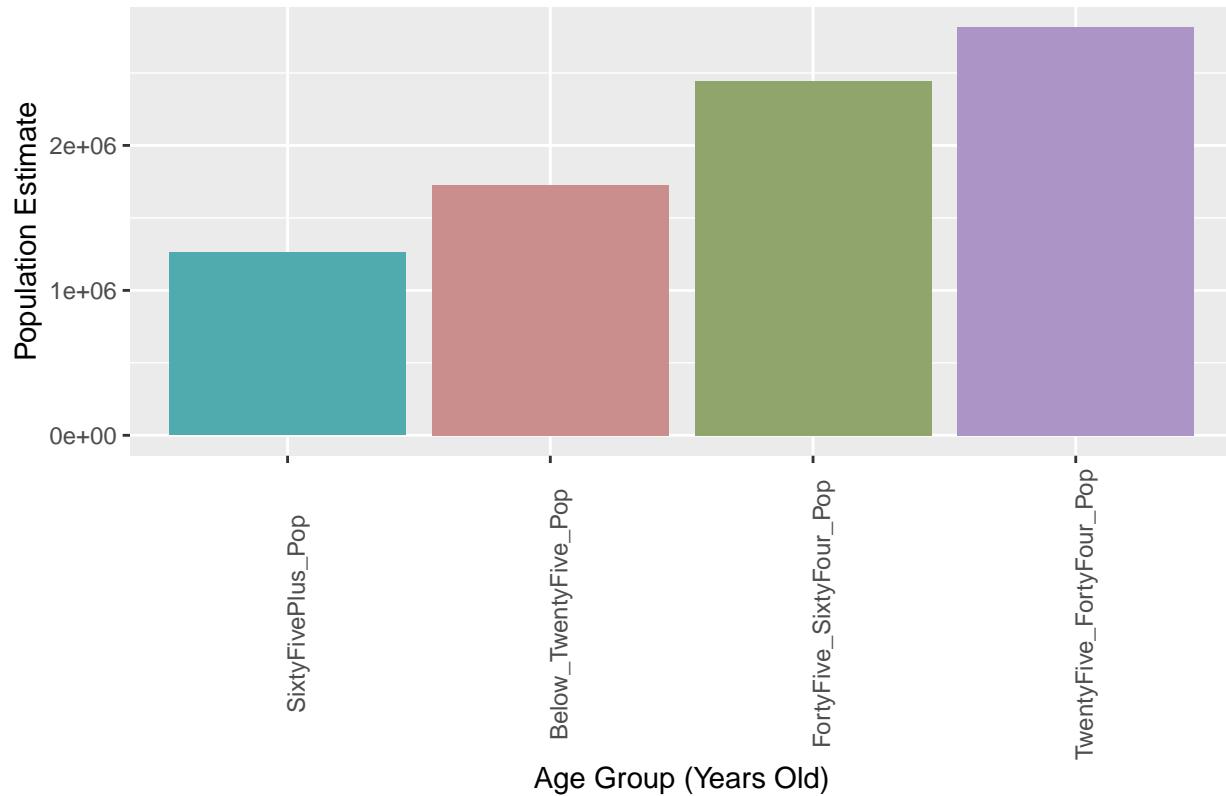
Population Estimate per Age Group in MANHATTAN, New York City



```
#creates bar graph of population estimates per age group in selected borough

boro <- "QUEENS"
Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group_Pop, Population)), fill = as.factor(Age_Group_Pop), y = Population),
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population Estimate") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Population Estimate per Age Group in ", boro, ", New York City"))
```

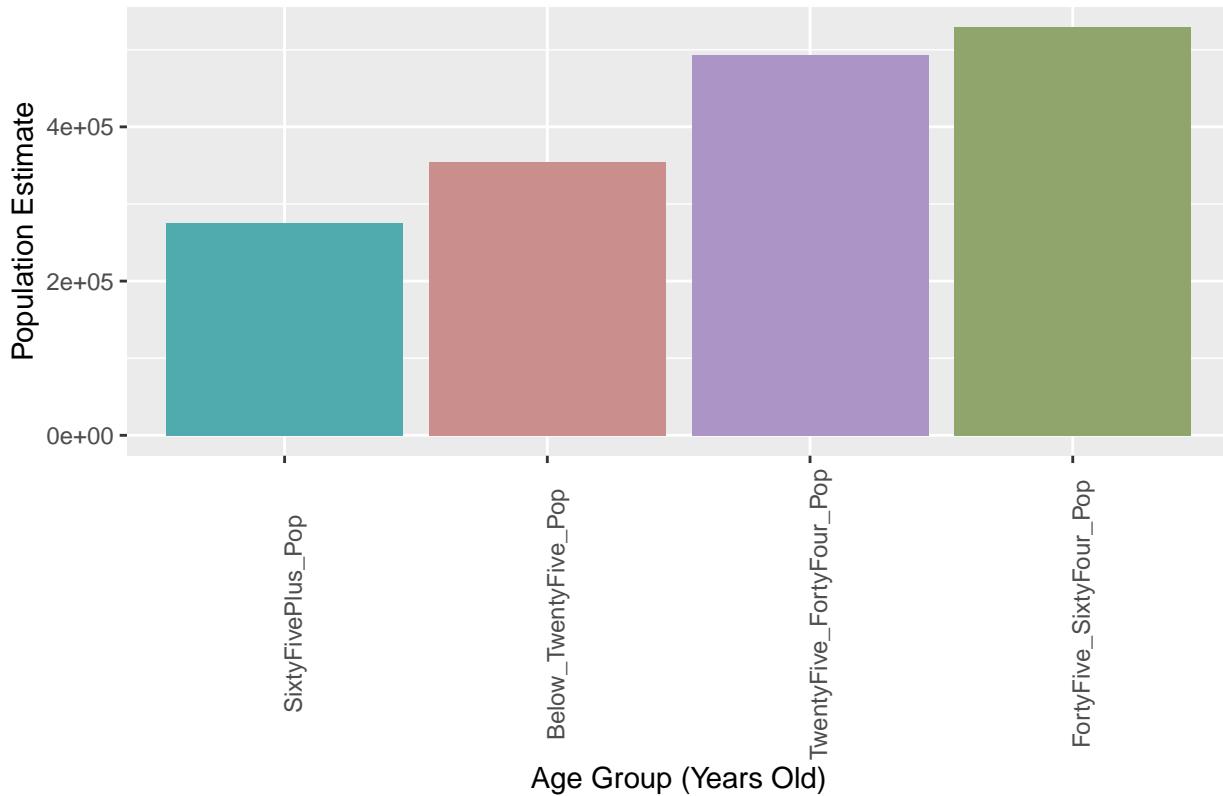
Population Estimate per Age Group in QUEENS, New York City



```
#creates bar graph of population estimates per age group in selected borough

boro <- "STATEN ISLAND"
Vic_Ages %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Age_Group_Pop, Population)), fill = as.factor(Age_Group_Pop), y = Population),
         theme(legend.position="none",
               axis.text.x = element_text(angle = 90)) +
  ylab("Population Estimate") +
  xlab("Age Group (Years Old)") +
  labs(title = str_c("Population Estimate per Age Group in ", boro, ", New York City"))
```

Population Estimate per Age Group in STATEN ISLAND, New York City



```
#creates bar graph of population estimates per age group in selected borough
```

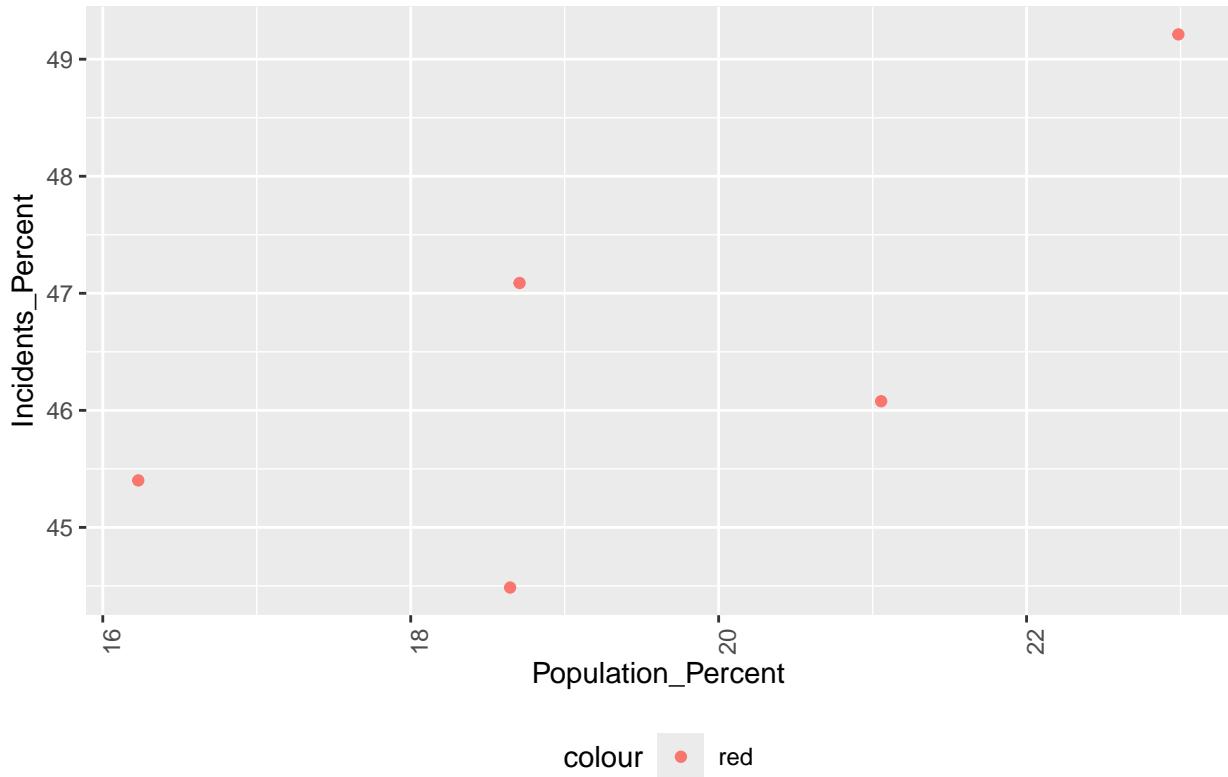
Just looking at these graphs seems to support the idea that the age groups most represented in gun violence might be affected by the population of those age groups present. However, we don't know for sure if these age groups are over or under represented in gun violence compared to population just by looking with the naked eye and the below 25 group is not the most prevalent in population.

Thus, we will graph the percentage of the population each age group makes up versus their percentages of shooters or victims of gun violence and model a function to see if population percentage is a good indicator of the percentage of that age group as a victim or perpetrator of a shooting.

```
temp <- Vic_Age_Percent %>% filter(Age_Group_Pop_Percent == "Below_TwentyFive_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "Below_TwentyFive_Percent")
#Changes selection of age data to match age groups between shooting victim percent and population percent

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "red")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooting Victims (Below 25 Years Old)")
```

Percent of Population versus Percent of Shooting Victims (Below 25 Years C)

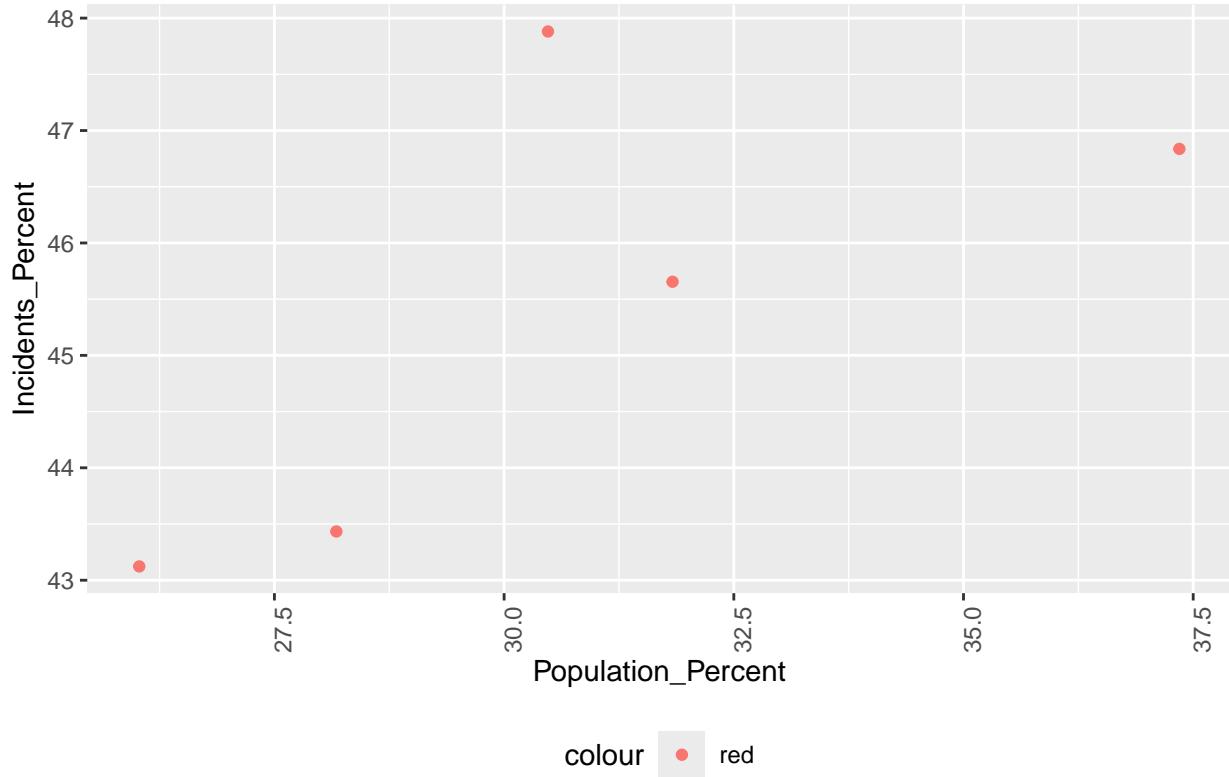


```
#Creates a graph comparing one age group over percent of population versus percent of shooting victims

temp <- Vic_Age_Percent %>% filter(Age_Group_Pop_Percent == "TwentyFive_FortyFour_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "TwentyFive_FortyFour_Percent")
#Changes selection of age data to match age groups between shooting victim percent and population percent

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "red")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooting Victims (25 - 44 Years Old)")
```

Percent of Population versus Percent of Shooting Victims (25 – 44 Years Old)

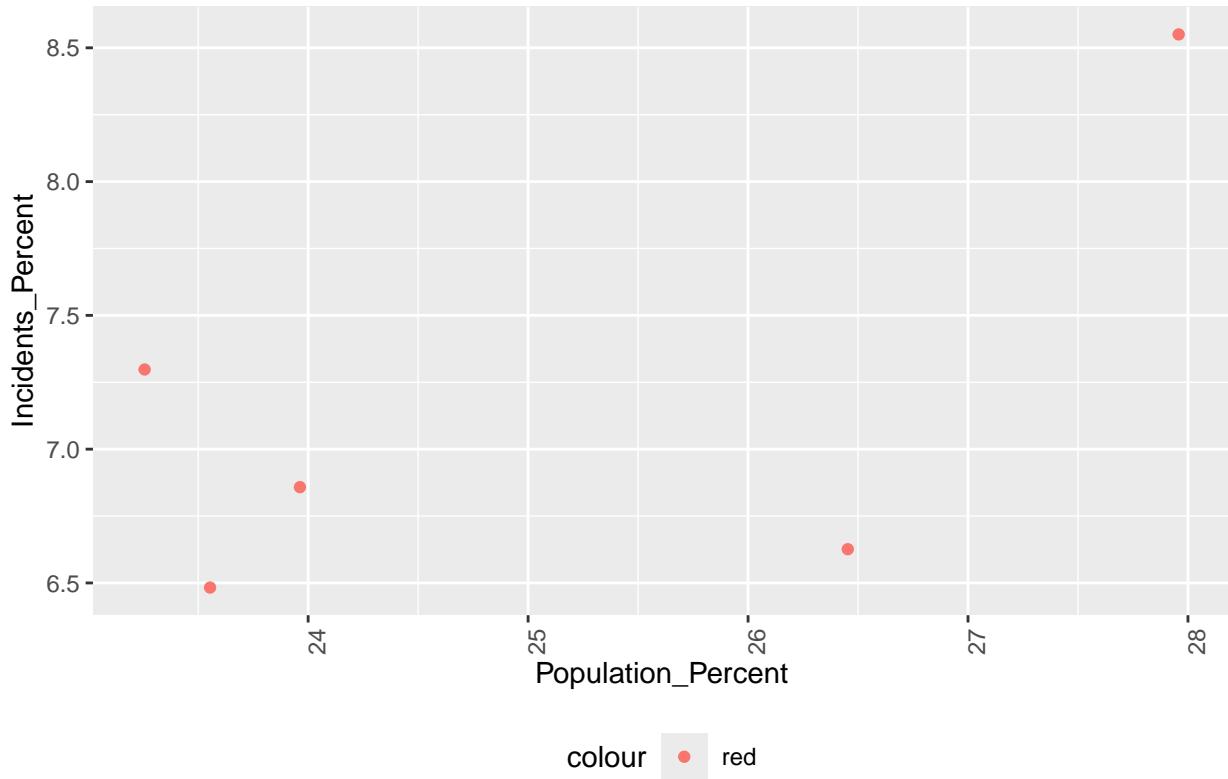


```
#Creates a graph comparing one age group over percent of population versus percent of shooting victims

temp <- Vic_Age_Percent %>% filter(Age_Group_Pop_Percent == "FortyFive_SixtyFour_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "FortyFive_SixtyFour_Percent")
#Changes selection of age data to match age groups between shooting victim percent and population percent

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "red")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooting Victims (45 - 64 Years Old)")
```

Percent of Population versus Percent of Shooting Victims (45 – 64 Years Old)

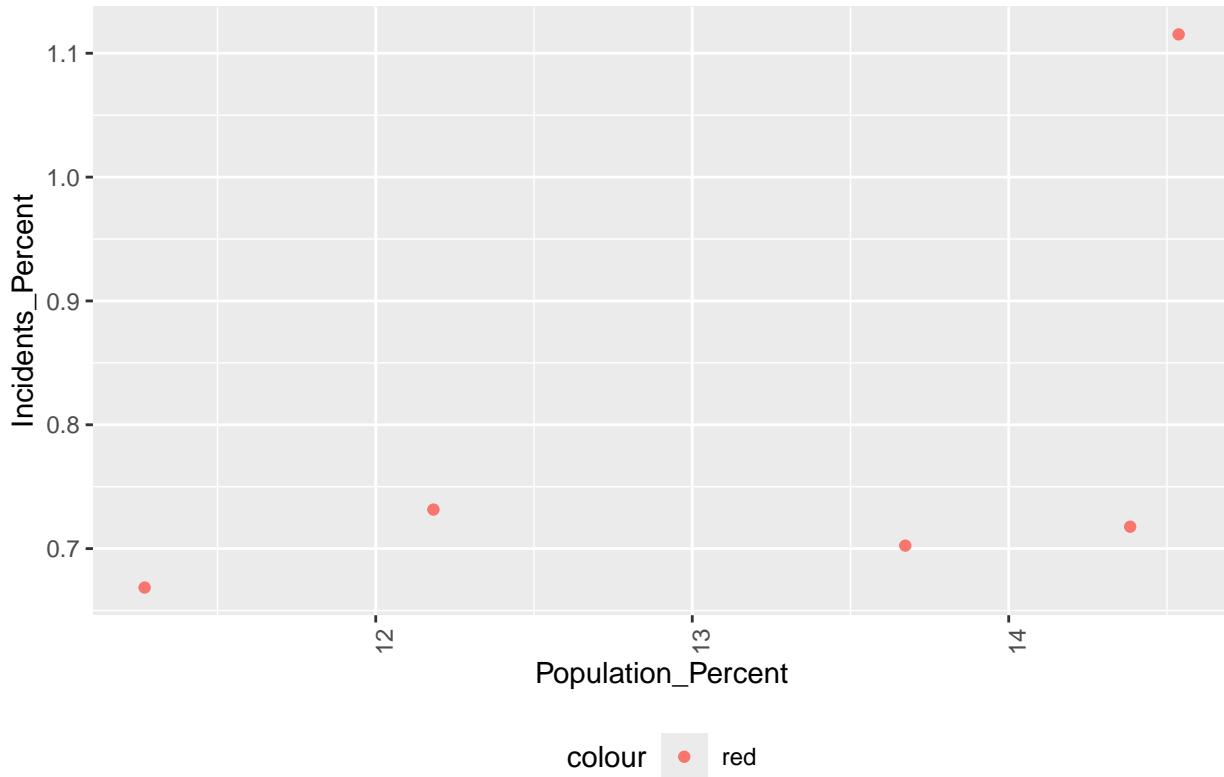


```
#Creates a graph comparing one age group over percent of population versus percent of shooting victims

temp <- Vic_Age_Percent %>% filter(Age_Group_Pop_Percent == "SixtyFivePlus_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "SixtyFivePlus_Percent")
#Changes selection of age data to match age groups between shooting victim percent and population percent

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "red")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooting Victims (65+ Years Old)")
```

Percent of Population versus Percent of Shooting Victims (65+ Years Old)



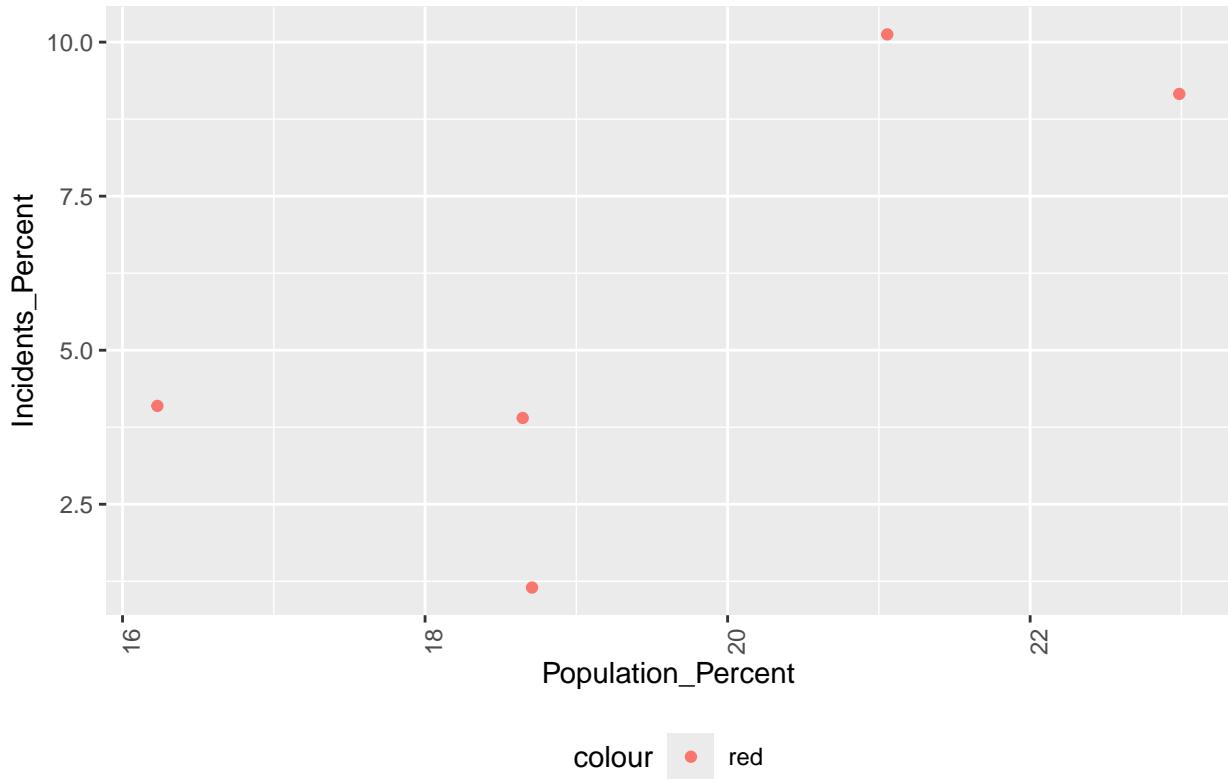
```
#Creates a graph comparing one age group over percent of population versus percent of shooting victims
```

Now we will do the same for perpetrators.

```
temp <- Perp_Age_Percent %>% filter(Age_Group_Pop_Percent == "Below_TwentyFive_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "Below_TwentyFive_Percent")
#Changes selection of age data to match age groups between shooting perps percent and population percent

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "red")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooters (Below 25 Years Old)")
```

Percent of Population versus Percent of Shooters (Below 25 Years Old)

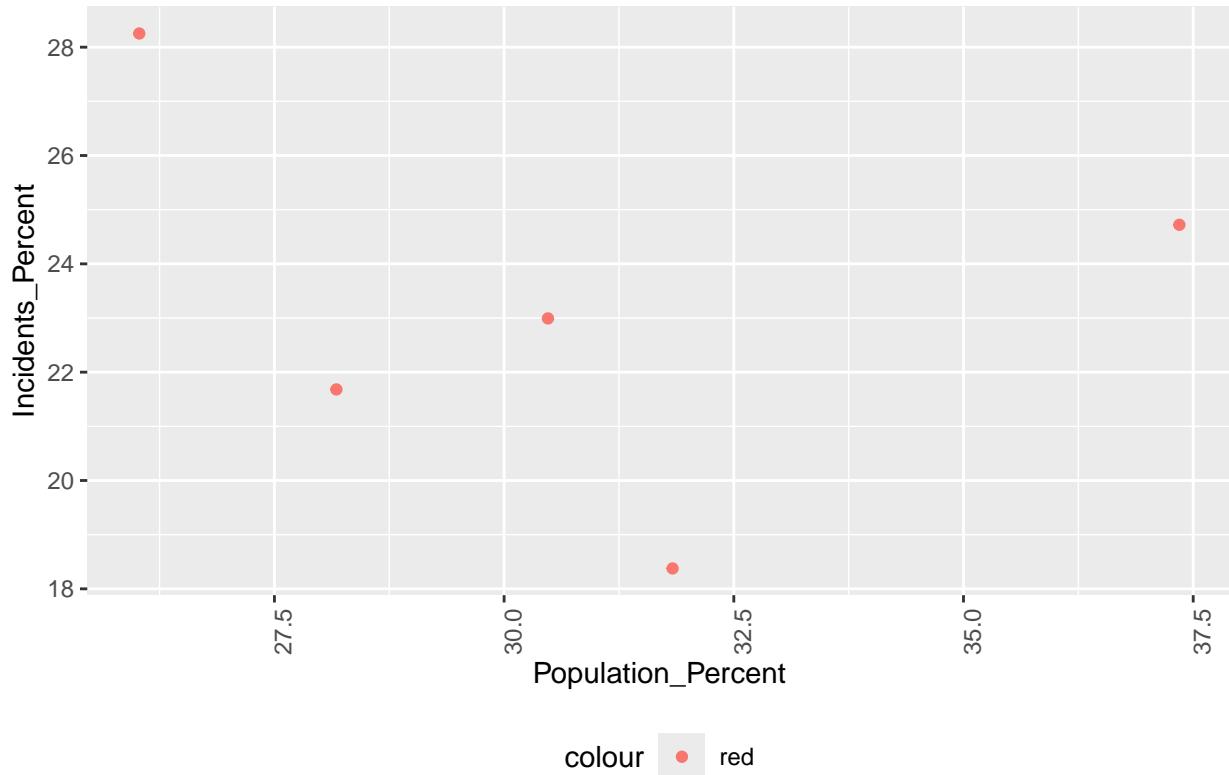


```
#Creates a graph comparing one age group over percent of population versus percent of shooting perps

temp <- Perp_Age_Percent %>% filter(Age_Group_Pop_Percent == "TwentyFive_FortyFour_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "TwentyFive_FortyFour_Percent")
#Changes selection of age data to match age groups between shooting perps percent and population percent

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "red")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooters (25 - 44 Years Old)")
```

Percent of Population versus Percent of Shooters (25 – 44 Years Old)

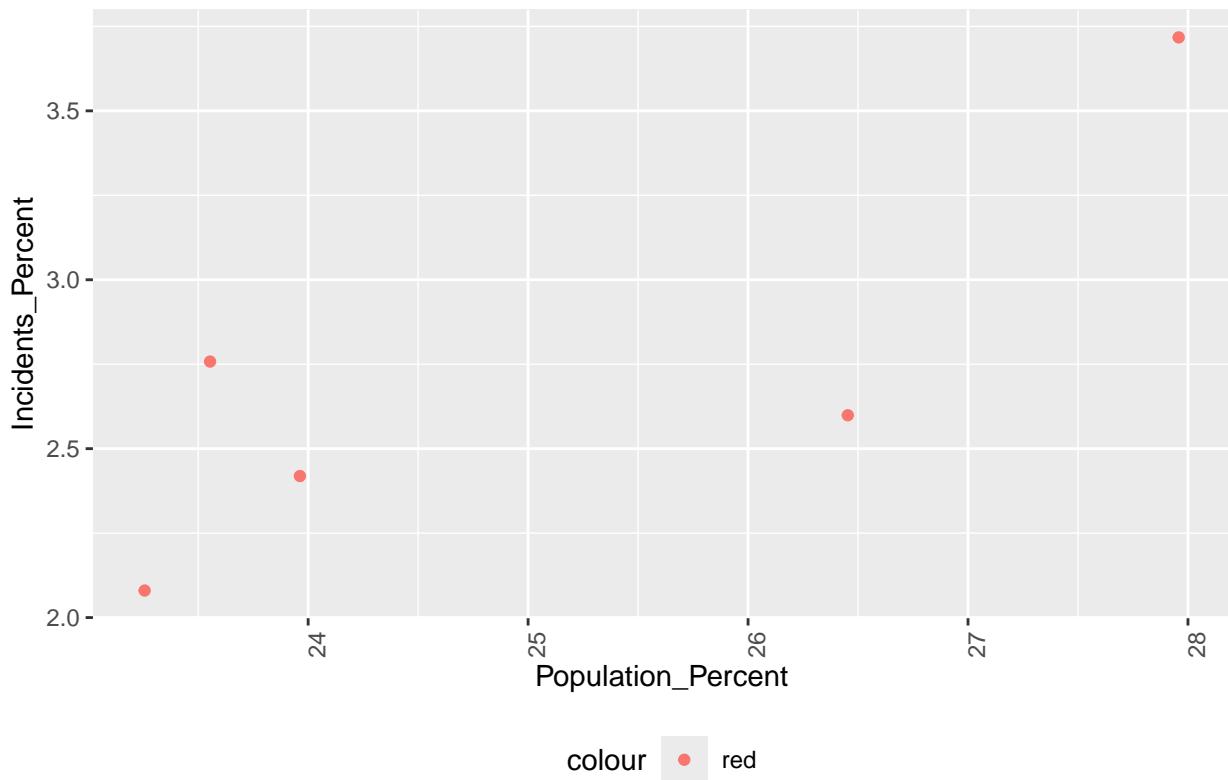


```
#Creates a graph comparing one age group over percent of population versus percent of shooting perps

temp <- Perp_Age_Percent %>% filter(Age_Group_Pop_Percent == "FortyFive_SixtyFour_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "FortyFive_SixtyFour_Percent")
#Changes selection of age data to match age groups between shooting perps percent and population percent

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "red")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooters (45 – 64 Years Old)")
```

Percent of Population versus Percent of Shooters (45 – 64 Years Old)

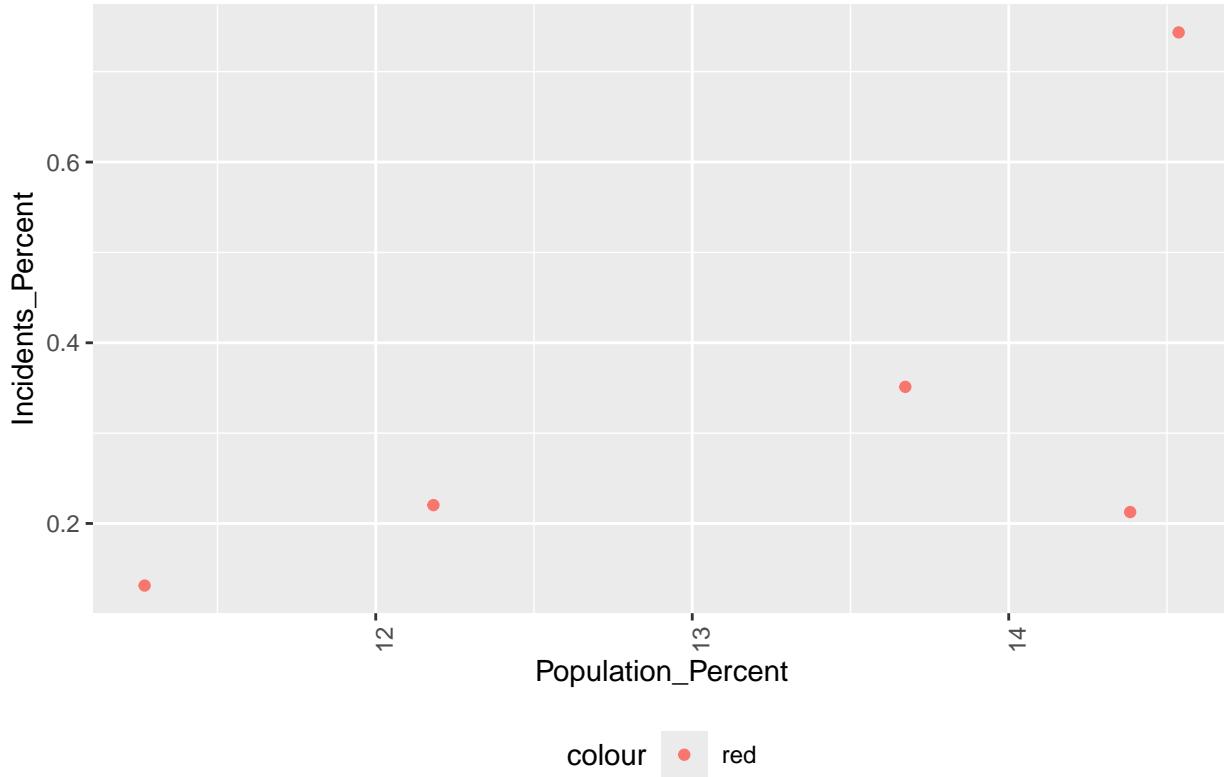


```
#Creates a graph comparing one age group over percent of population versus percent of shooting perps

temp <- Perp_Age_Percent %>% filter(Age_Group_Pop_Percent == "SixtyFivePlus_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "SixtyFivePlus_Percent")
#Changes selection of age data to match age groups between shooting perps percent and population percent

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "red")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooters (65+ Years Old)")
```

Percent of Population versus Percent of Shooters (65+ Years Old)



```
#Creates a graph comparing one age group over percent of population versus percent of shooting perps
```

Our graphs give us a sense that, except for the 25 - 44 year group which displays the opposite trend in perpetrators, most populations that have more of one age group also have more perpetrators and victims of gun violence. However, the 25-44 age group shows a lower number of perpetrators of gun violence in a borough with a higher number of people in that age group, meaning something other than age and random chance may be affecting gun violence.

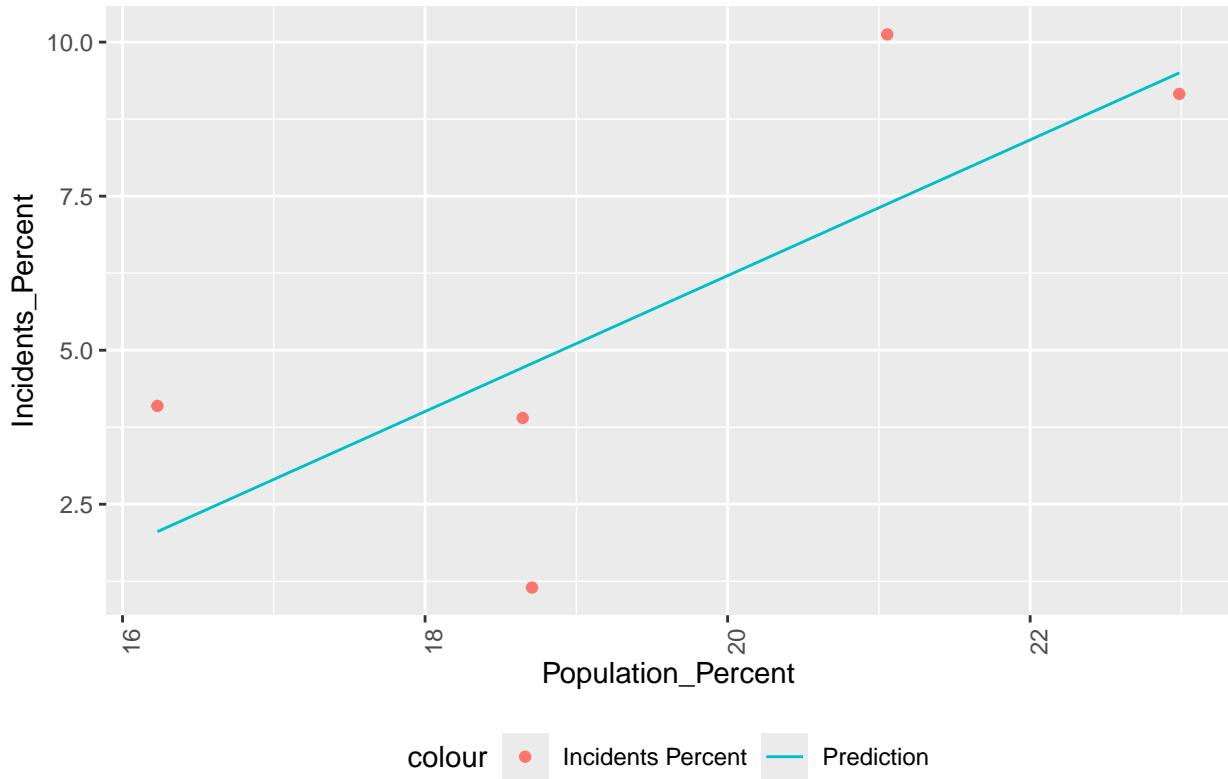
Modeling a Linear Regression on Age Groups Percent Population versus Percent in Gun Violence

We will model a linear regression to these graphs in order to determine if the correlations are significant. First we will start with the perpetrators.

```
temp <- Perp_Age_Percent %>% filter(Age_Group_Pop_Percent == "Below_TwentyFive_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "Below_TwentyFive_Percent")

mod <- lm(Incidents_Percent ~ poly(Population_Percent, degree = 1), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "Incidents_Percent")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooters (Below 25 Years Old)")
```

Percent of Population versus Percent of Shooters (Below 25 Years Old)

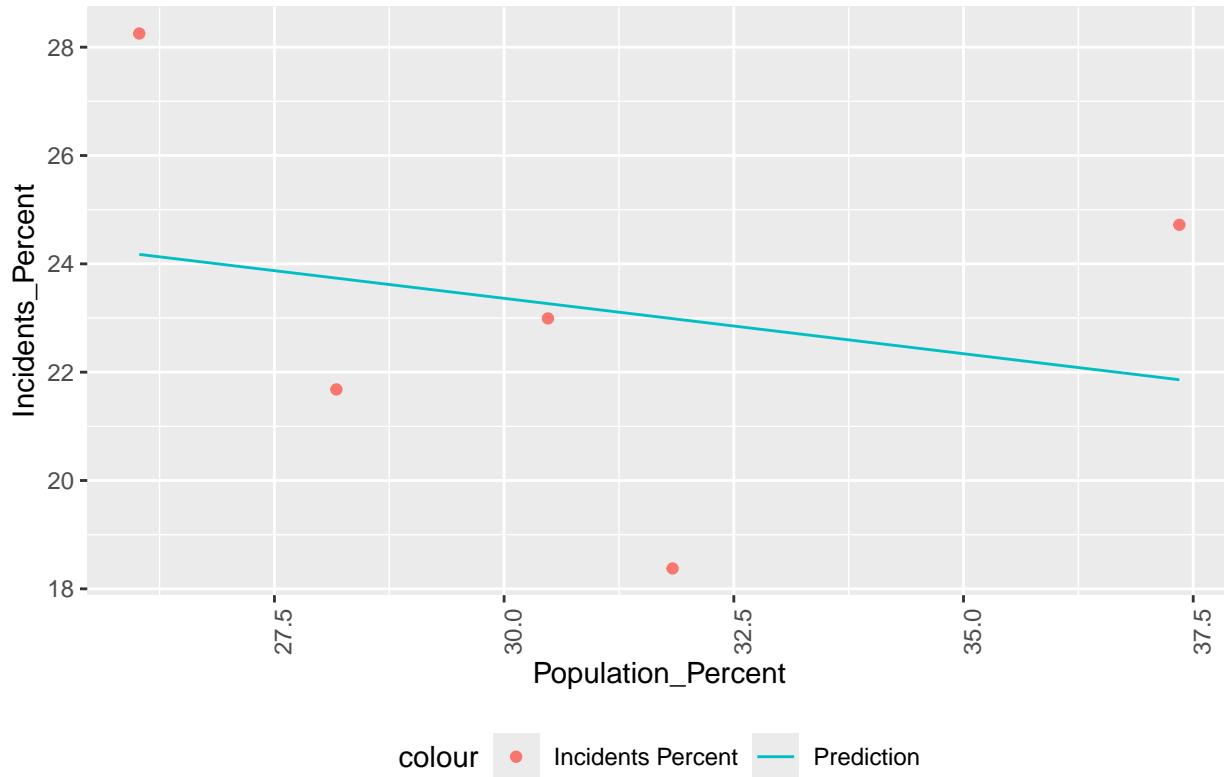


```
Below_TwentyFive_mod <- mod

temp <- Perp_Age_Percent %>% filter(Age_Group_Pop_Percent == "TwentyFive_FortyFour_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "TwentyFive_FortyFour_Percent")

mod <- lm(Incidents_Percent ~ poly(Population_Percent, degree = 1), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooters (25-44 Years Old)")
```

Percent of Population versus Percent of Shooters (25–44 Years Old)

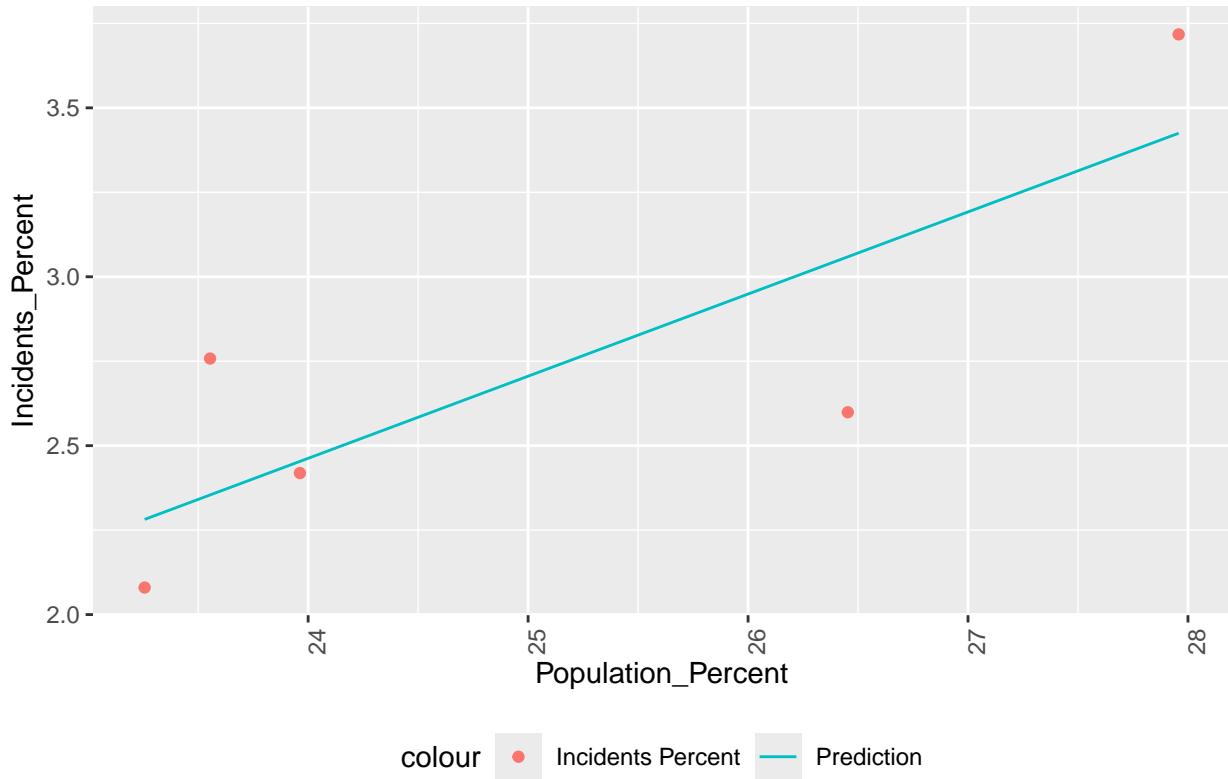


```
TwentyFive_FortyFour_mod <- mod

temp <- Perp_Age_Percent %>% filter(Age_Group_Pop_Percent == "FortyFive_SixtyFour_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "FortyFive_SixtyFour_Percent")

mod <- lm(Incidents_Percent ~ poly(Population_Percent, degree = 1), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooters (45-64 Years Old)")
```

Percent of Population versus Percent of Shooters (45–64 Years Old)

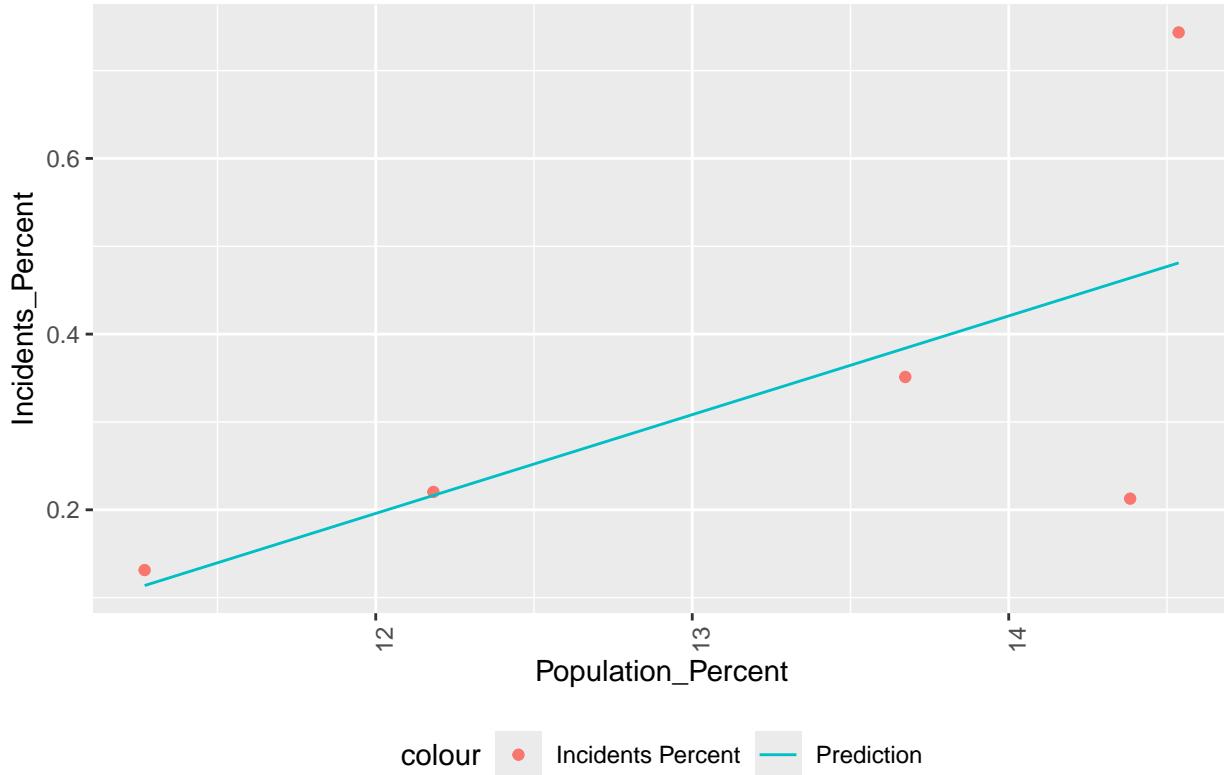


```
FortyFive_SixtyFour_Mod <- mod

temp <- Perp_Age_Percent %>% filter(Age_Group_Pop_Percent == "SixtyFivePlus_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "SixtyFivePlus_Percent")

mod <- lm(Incidents_Percent ~ Population_Percent, data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Shooters (65+ Years Old)")
```

Percent of Population versus Percent of Shooters (65+ Years Old)



```
SixtyFivePlus_Mod <- mod
```

With only four data points, we can't put too much weight on these predictions, especially since each point is from a different borough, but lacking population over time, we have to work with what we have.

R Value Results Here we will display the summaries of the linear regressions:

```
print("Below 25 Year Old Perpetrators: Coefficients and R-Values")
## [1] "Below 25 Year Old Perpetrators: Coefficients and R-Values"
summary(Below_TwentyFive_mod)

##
## Call:
## lm(formula = Incidents_Percent ~ poly(Population_Percent, degree = 1),
##      data = temp)
##
## Residuals:
##      1       2       3       4       5 
## -0.3411  2.7528  2.0411 -0.8166 -3.6362 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  5.686     1.310   4.340   0.0226 *  
## poly(Population_Percent, degree = 1) 5.686     2.930   1.941   0.1476  
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

## 
## Residual standard error: 2.93 on 3 degrees of freedom
## Multiple R-squared:  0.5566, Adjusted R-squared:  0.4088
## F-statistic: 3.766 on 1 and 3 DF,  p-value: 0.1476
print("25 - 44 Year Old Perpetrators: Coefficients and R-Values")

## [1] "25 - 44 Year Old Perpetrators: Coefficients and R-Values"
summary(TwentyFive_FortyFour_mod)

##
## Call:
## lm(formula = Incidents_Percent ~ poly(Population_Percent, degree = 1),
##      data = temp)
##
## Residuals:
##       1       2       3       4       5 
## -2.0550 -4.6113  2.8610 -0.2728  4.0781 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 23.205     1.833   12.662 0.00106 ***
## poly(Population_Percent, degree = 1) -1.756      4.098   -0.428  0.69724  
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 4.098 on 3 degrees of freedom
## Multiple R-squared:  0.05765, Adjusted R-squared:  -0.2565 
## F-statistic: 0.1835 on 1 and 3 DF,  p-value: 0.6972

print("45 - 64 Year Old Perpetrators: Coefficients and R-Values")

## [1] "45 - 64 Year Old Perpetrators: Coefficients and R-Values"
summary(FortyFive_SixtyFour_Mod)

##
## Call:
## lm(formula = Incidents_Percent ~ poly(Population_Percent, degree = 1),
##      data = temp)
##
## Residuals:
##       1       2       3       4       5 
## 0.40382 -0.20171 -0.03449 -0.46019  0.29256 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  2.7146     0.1830   14.835 0.000665 ***
## poly(Population_Percent, degree = 1) 1.0044     0.4092    2.455 0.091305 .  
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 0.4092 on 3 degrees of freedom
## Multiple R-squared:  0.6676, Adjusted R-squared:  0.5568 
## F-statistic: 6.025 on 1 and 3 DF,  p-value: 0.09131

```

```

print("65+ Year Old Perpetrators: Coefficients and R-Values")

## [1] "65+ Year Old Perpetrators: Coefficients and R-Values"
summary(SixtyFivePlus_Mod)

##
## Call:
## lm(formula = Incidents_Percent ~ Population_Percent, data = temp)
##
## Residuals:
##       1        2        3        4        5
## 0.017571  0.003992 -0.251202 -0.032769  0.262408
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.15332   0.97870 -1.178   0.324
## Population_Percent 0.11243   0.07375  1.525   0.225
##
## Residual standard error: 0.2108 on 3 degrees of freedom
## Multiple R-squared:  0.4365, Adjusted R-squared:  0.2487
## F-statistic: 2.324 on 1 and 3 DF,  p-value: 0.2248

```

As can be seen above, these hover just on the edge of acceptable R-values, so we would need more data to make these findings significant. This means we cannot prove a direct, positive relationship between the age group's percent of the population and the age group's percent of shooting perpetrators.

Now we will examine the shooting victims in the same manner.

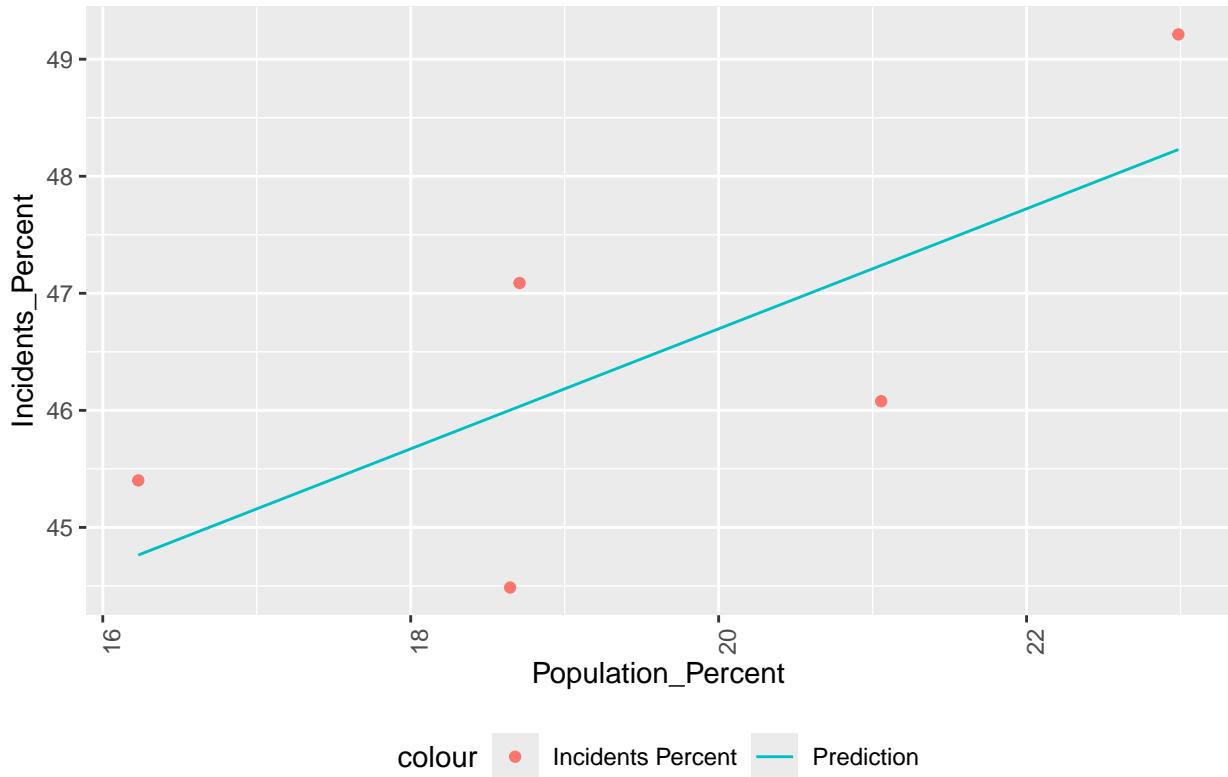
```

temp <- Vic_Age_Percent %>% filter(Age_Group_Pop_Percent == "Below_TwentyFive_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "Below_TwentyFive_Percent")

mod <- lm(Incidents_Percent ~ poly(Population_Percent, degree = 1), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Victims (Below 25 Years Old)")

```

Percent of Population versus Percent of Victims (Below 25 Years Old)

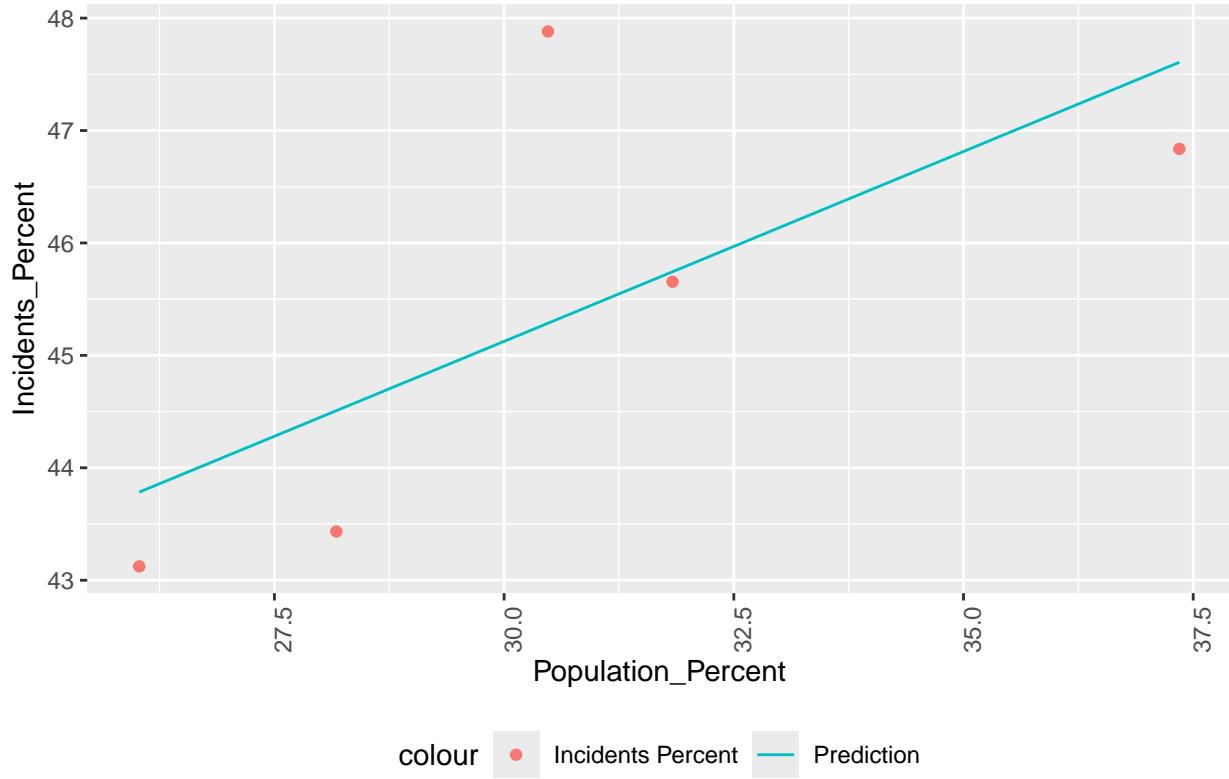


```
Below_TwentyFive_mod <- mod

temp <- Vic_Age_Percent %>% filter(Age_Group_Pop_Percent == "TwentyFive_FortyFour_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "TwentyFive_FortyFour_Percent")

mod <- lm(Incidents_Percent ~ poly(Population_Percent, degree = 1), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Victims (25-44 Years Old)")
```

Percent of Population versus Percent of Victims (25–44 Years Old)



```

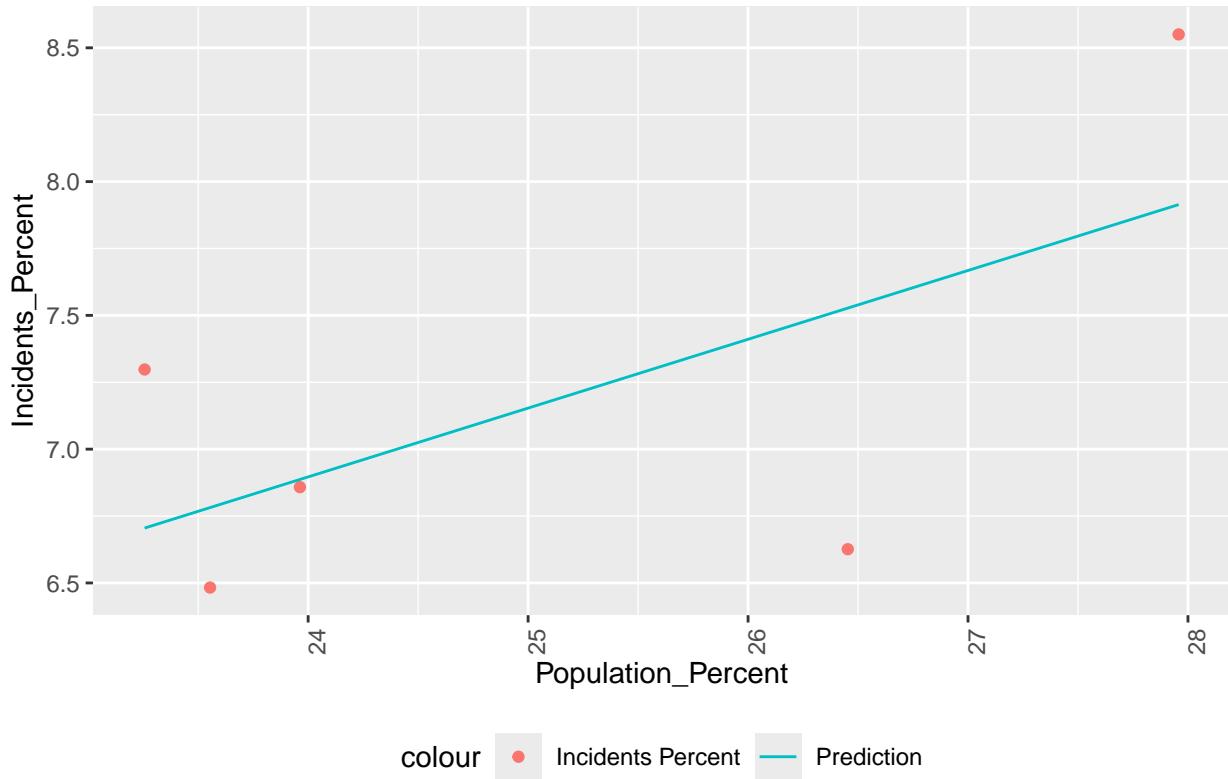
TwentyFive_FortyFour_mod <- mod

temp <- Vic_Age_Percent %>% filter(Age_Group_Pop_Percent == "FortyFive_SixtyFour_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "FortyFive_SixtyFour_Percent")

mod <- lm(Incidents_Percent ~ poly(Population_Percent, degree = 1), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Victims (45-64 Years Old)")

```

Percent of Population versus Percent of Victims (45–64 Years Old)



```

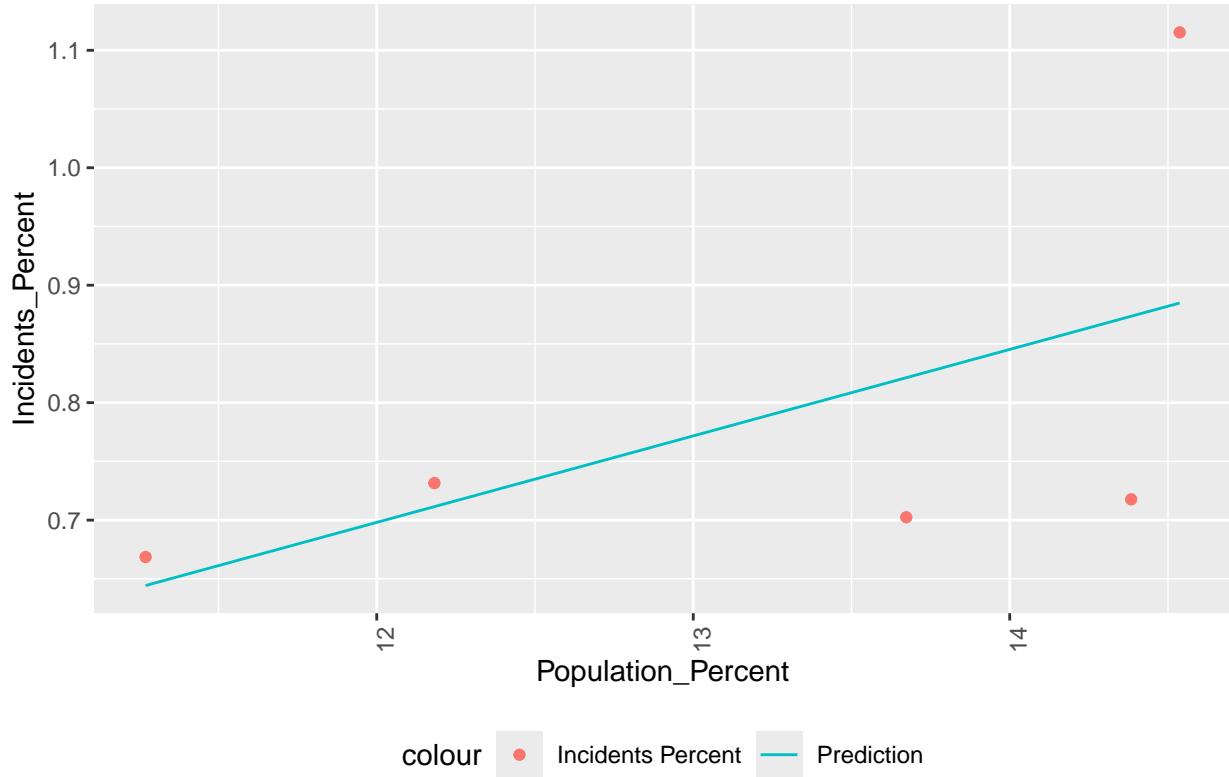
FortyFive_SixtyFour_Mod <- mod

temp <- Vic_Age_Percent %>% filter(Age_Group_Pop_Percent == "SixtyFivePlus_Pop_Percent")
temp <- temp %>% filter(Age_Group_Percent == "SixtyFivePlus_Percent")

mod <- lm(Incidents_Percent ~ Population_Percent, data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents_Percent)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Percent of Population versus Percent of Victims (65+ Years Old)")

```

Percent of Population versus Percent of Victims (65+ Years Old)



```
SixtyFivePlus_Mod <- mod
```

As you can see, these graphs all have positive slopes, implying a positive correlation. We will examine the R-values and coefficients to see if there is a significant relationship here.

```
print("Below 25 Year Old Victims: Coefficients and R-Values")
## [1] "Below 25 Year Old Victims: Coefficients and R-Values"
summary(Below_TwentyFive_mod)

##
## Call:
## lm(formula = Incidents_Percent ~ poly(Population_Percent, degree = 1),
##      data = temp)
##
## Residuals:
##      1       2       3       4       5 
## 0.9840 -1.1600  0.6378 -1.5161  1.0543 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                46.4531    0.6393  72.661 5.74e-06 ***
## poly(Population_Percent, degree = 1) 2.6458     1.4295   1.851   0.161  
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.43 on 3 degrees of freedom
```

```

## Multiple R-squared:  0.5331, Adjusted R-squared:  0.3775
## F-statistic: 3.425 on 1 and 3 DF,  p-value: 0.1613
print("25 - 44 Year Old Victims: Coefficients and R-Values")

## [1] "25 - 44 Year Old Victims: Coefficients and R-Values"
summary(TwentyFive_FortyFour_mod)

##
## Call:
## lm(formula = Incidents_Percent ~ poly(Population_Percent, degree = 1),
##      data = temp)
##
## Residuals:
##       1        2        3        4        5
## -1.07458 -0.08919 -0.77032  2.59504 -0.66096
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 45.3858    0.7715  58.831 1.08e-05 ***
## poly(Population_Percent, degree = 1)  2.8999    1.7250   1.681   0.191
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.725 on 3 degrees of freedom
## Multiple R-squared:  0.4851, Adjusted R-squared:  0.3134
## F-statistic: 2.826 on 1 and 3 DF,  p-value: 0.1913
print("45 - 64 Year Old Victims: Coefficients and R-Values")

## [1] "45 - 64 Year Old Victims: Coefficients and R-Values"
summary(FortyFive_SixtyFour_Mod)

##
## Call:
## lm(formula = Incidents_Percent ~ poly(Population_Percent, degree = 1),
##      data = temp)
##
## Residuals:
##       1        2        3        4        5
## -0.29881  0.59261 -0.02863 -0.90120  0.63603
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 7.1630    0.3325  21.545 0.000219 ***
## poly(Population_Percent, degree = 1)  1.0623    0.7434   1.429  0.248363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7434 on 3 degrees of freedom
## Multiple R-squared:  0.405, Adjusted R-squared:  0.2066
## F-statistic: 2.042 on 1 and 3 DF,  p-value: 0.2484
print("65+ Year Old Victims: Coefficients and R-Values")

## [1] "65+ Year Old Victims: Coefficients and R-Values"

```

```

summary(SixtyFivePlus_Mod)

##
## Call:
## lm(formula = Incidents_Percent ~ Population_Percent, data = temp)
##
## Residuals:
##      1       2       3       4       5 
## 0.02428  0.02005 -0.15587 -0.11885  0.23038
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.18550   0.81504  -0.228   0.835    
## Population_Percent 0.07363   0.06142   1.199   0.317    
## 
## Residual standard error: 0.1756 on 3 degrees of freedom
## Multiple R-squared:  0.3239, Adjusted R-squared:  0.09855 
## F-statistic: 1.437 on 1 and 3 DF,  p-value: 0.3166

```

On average, our R-values don't meet standard to predict population percent versus victim percent well. Either our sample size or our variables are lacking. However, we can see something close to a relationship. It's not a perfect prediction, but there is a trend between population size of an age group and whether they are a victim or perpetrator of gun violence, implying that age group is not a solid predictor. Let's look at sex next to see if there's a stronger relationship.

Transforming Sex for Visualization

We will need to transform the sex columns in the same way we transformed the age columns in order to easily visualize them per borough. First we will isolate the sex data.

```

Vic_Sexes <- Vic_By_Boro_Totals_with_Pop %>%
  pivot_longer(cols = c(Male, Female, Unknown_Sex, Unlisted_Sex),
               names_to = "Sex",
               values_to = "Incidents") %>%
  pivot_longer(cols = c(Male_Pop, Female_Pop),
               names_to = "Sex_Pop",
               values_to = "Population") %>%
  select(c(BORO, Sex, Incidents, Sex_Pop, Population))
#Pivots sexes and sexes total populations into four columns for visualization

Perp_Sexes <- Perp_By_Boro_Totals_with_Pop %>%
  pivot_longer(cols = c(Male, Female, Unknown_Sex, Unlisted_Sex),
               names_to = "Sex",
               values_to = "Incidents") %>%
  pivot_longer(cols = c(Male_Pop, Female_Pop),
               names_to = "Sex_Pop",
               values_to = "Population") %>%
  select(c(BORO, Sex, Incidents, Sex_Pop, Population))
#Pivots sexes and sexes total populations into four columns for visualization

Vic_Sex_Percent <- Vic_By_Boro_Totals_with_Pop %>%
  mutate(Male_Percent = Male * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         Female_Percent = Female * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         Unknown_Sex_Percent = Unknown_Sex * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         Unlisted_Sex_Percent = Unlisted_Sex * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),

```

```

#begin percents for population
Male_Pop_Percent = Male_Pop * 100 / Total_Population,
Female_Pop_Percent = Female_Pop * 100 / Total_Population,
Unknown_Sex_Pop_Percent = 100 - ((Male_Pop + Female_Pop) *100 / Total_Population)
) %>%
pivot_longer(cols = c(Male_Percent, Female_Percent, Unknown_Sex_Percent, Unlisted_Sex_Percent),
             names_to = "Sex_Percent",
             values_to = "Incidents_Percent") %>%
pivot_longer(cols = c(Male_Pop_Percent, Female_Pop_Percent, Unknown_Sex_Pop_Percent),
             names_to = "Sex_Pop_Percent",
             values_to = "Population_Percent") %>%
select(c(BORO, Sex_Percent, Incidents_Percent, Sex_Pop_Percent, Population_Percent))
#Pivots sexes percents and sexes total population percentages into four columns for visualization

Perp_Sex_Percent <- Perp_By_Boro_Totals_with_Pop %>%
  mutate(Male_Percent = Male * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         Female_Percent = Female * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         Unknown_Sex_Percent = Unknown_Sex * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         Unlisted_Sex_Percent = Unlisted_Sex * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
#begin percents for population
         Male_Pop_Percent = Male_Pop * 100 / Total_Population,
         Female_Pop_Percent = Female_Pop * 100 / Total_Population,
         Unknown_Sex_Pop_Percent = 100 - ((Male_Pop + Female_Pop) *100 / Total_Population)
) %>%
pivot_longer(cols = c(Male_Percent, Female_Percent, Unknown_Sex_Percent, Unlisted_Sex_Percent),
             names_to = "Sex_Percent",
             values_to = "Incidents_Percent") %>%
pivot_longer(cols = c(Male_Pop_Percent, Female_Pop_Percent, Unknown_Sex_Pop_Percent),
             names_to = "Sex_Pop_Percent",
             values_to = "Population_Percent") %>%
select(c(BORO, Sex_Percent, Incidents_Percent, Sex_Pop_Percent, Population_Percent))
#Pivots sexes percents and sexes total population percentages into four columns for visualization

```

This sets us up for visualization by isolating sex data for shooters and victims into four new dataframes, two of which focus on percent of shooters/victims which are each sex, and two of which focus on raw numbers.

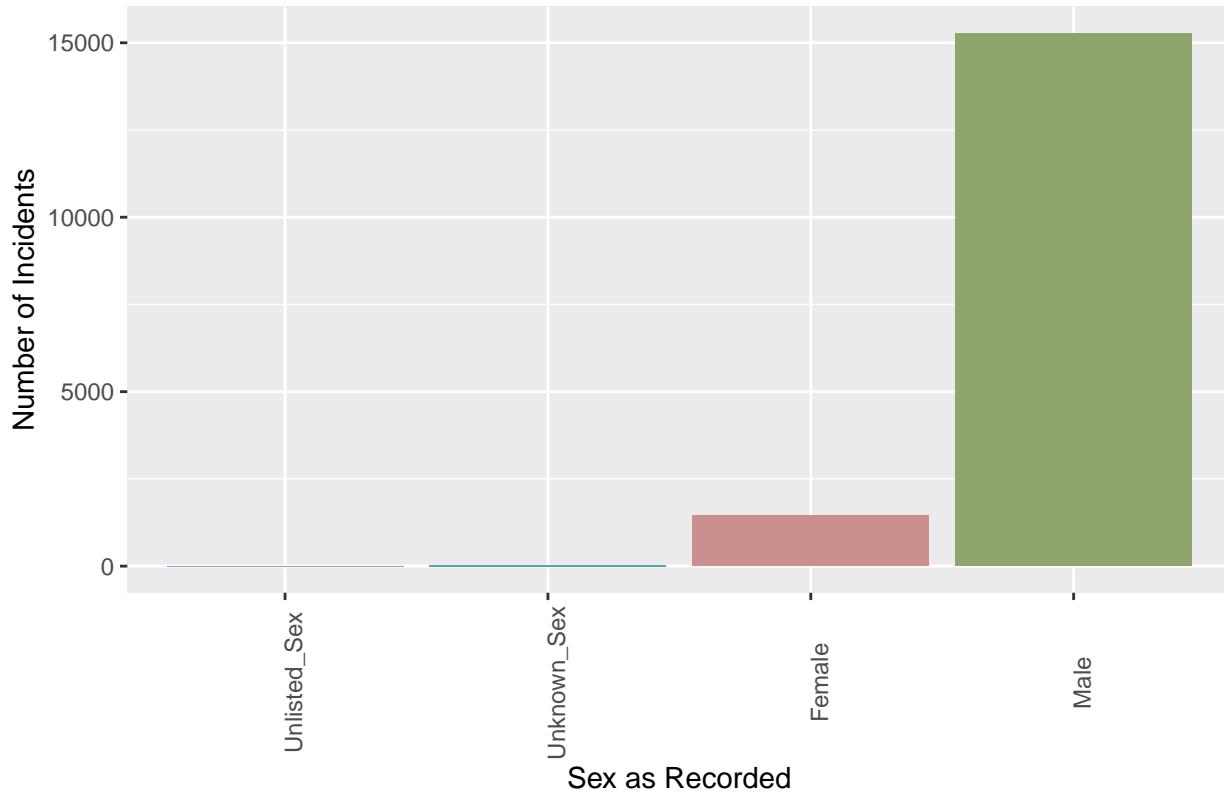
Visualizing Sexes of Perpetrators and Victims Next we will look at the distribution of perpetrators and victims over the boroughs of New York City. We will do this through bar graph visualizations.

```

boro <- "BRONX"
Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar(
    theme(legend.position="none",
      axis.text.x = element_text(angle = 90)) +
    ylab("Number of Incidents") +
    xlab("Sex as Recorded") +
    labs(title = str_c("Number of Victims per Sex in ", boro, ", New York City"))

```

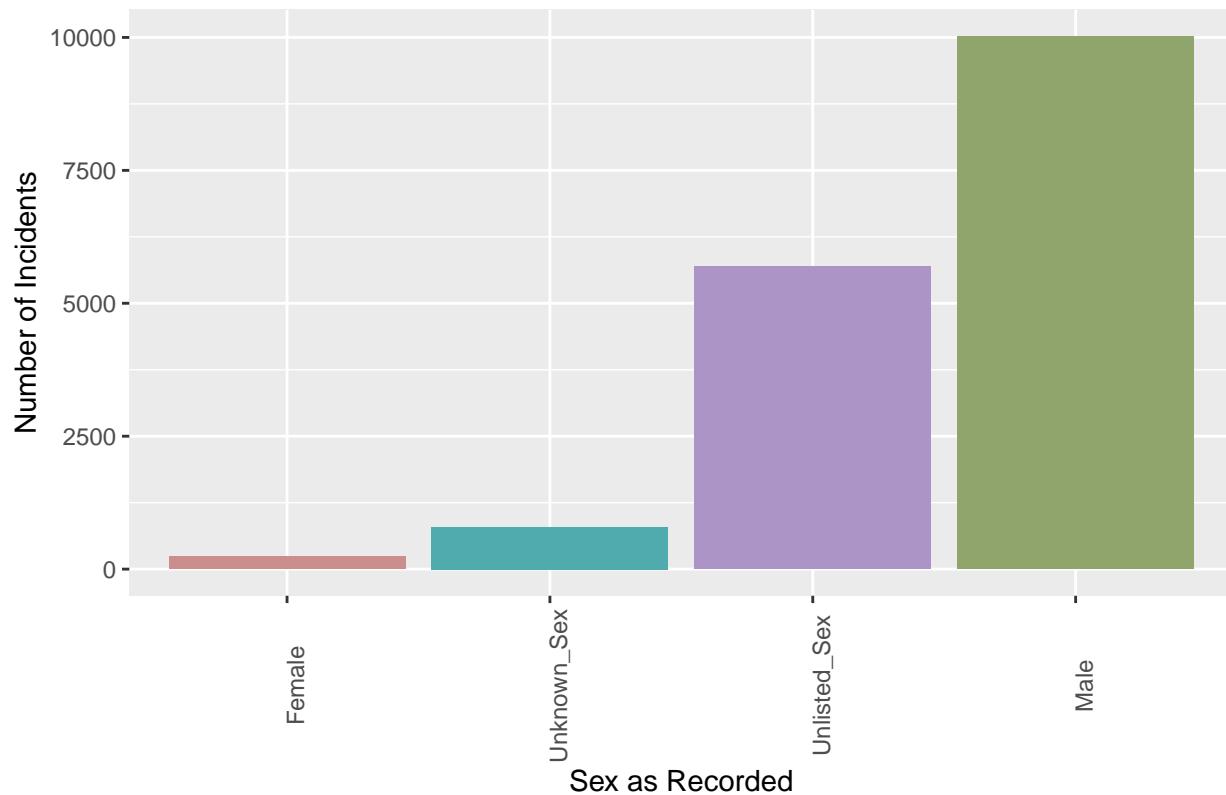
Number of Victims per Sex in BRONX, New York City



```
#creates a bar graph showing the number of victims per sex in the selected borough
```

```
Perp_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar(
    theme(legend.position="none",
      axis.text.x = element_text(angle = 90)) +
    ylab("Number of Incidents") +
    xlab("Sex as Recorded") +
    labs(title = str_c("Number of Shooters per Sex in ", boro, ", New York City"))
```

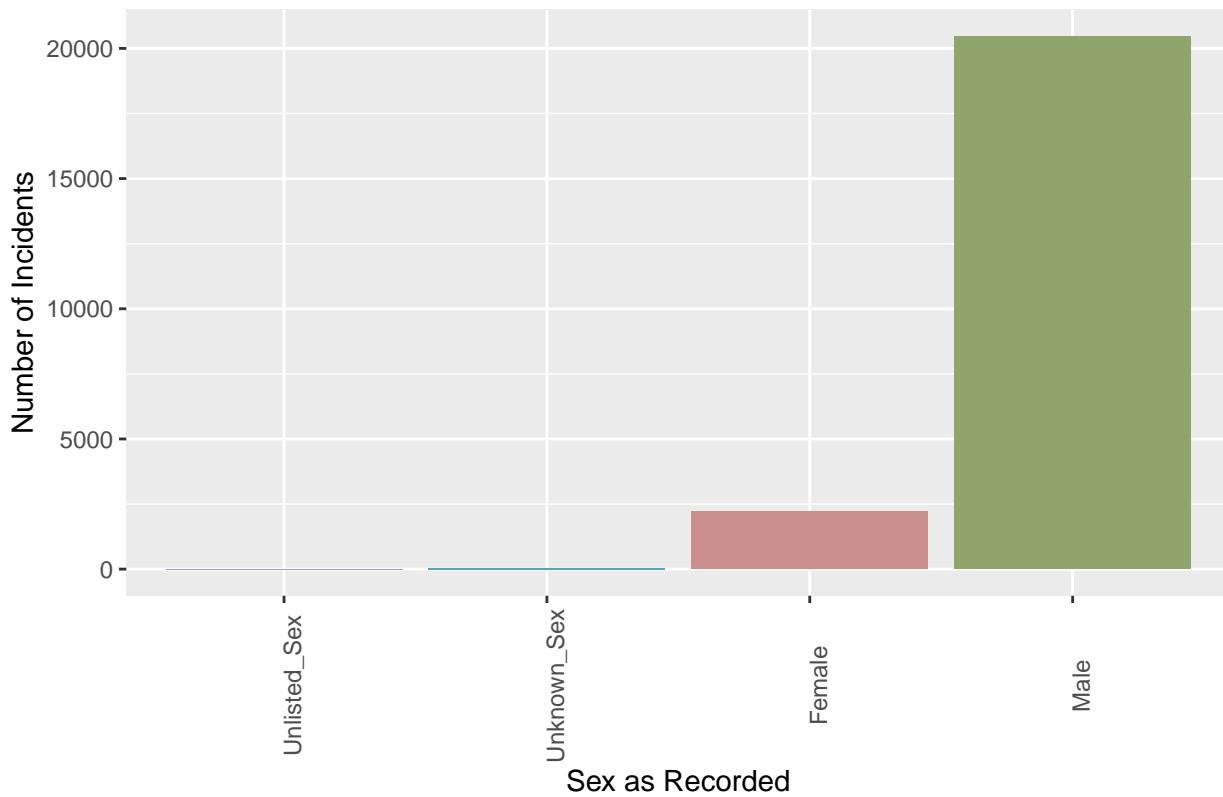
Number of Shooters per Sex in BRONX, New York City



```
#creates a bar graph showing the number of perpetrators per sex in the selected borough
boro <- "BROOKLYN"

Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar()
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Sex as Recorded") +
  labs(title = str_c("Number of Victims per Sex in ", boro, ", New York City"))
```

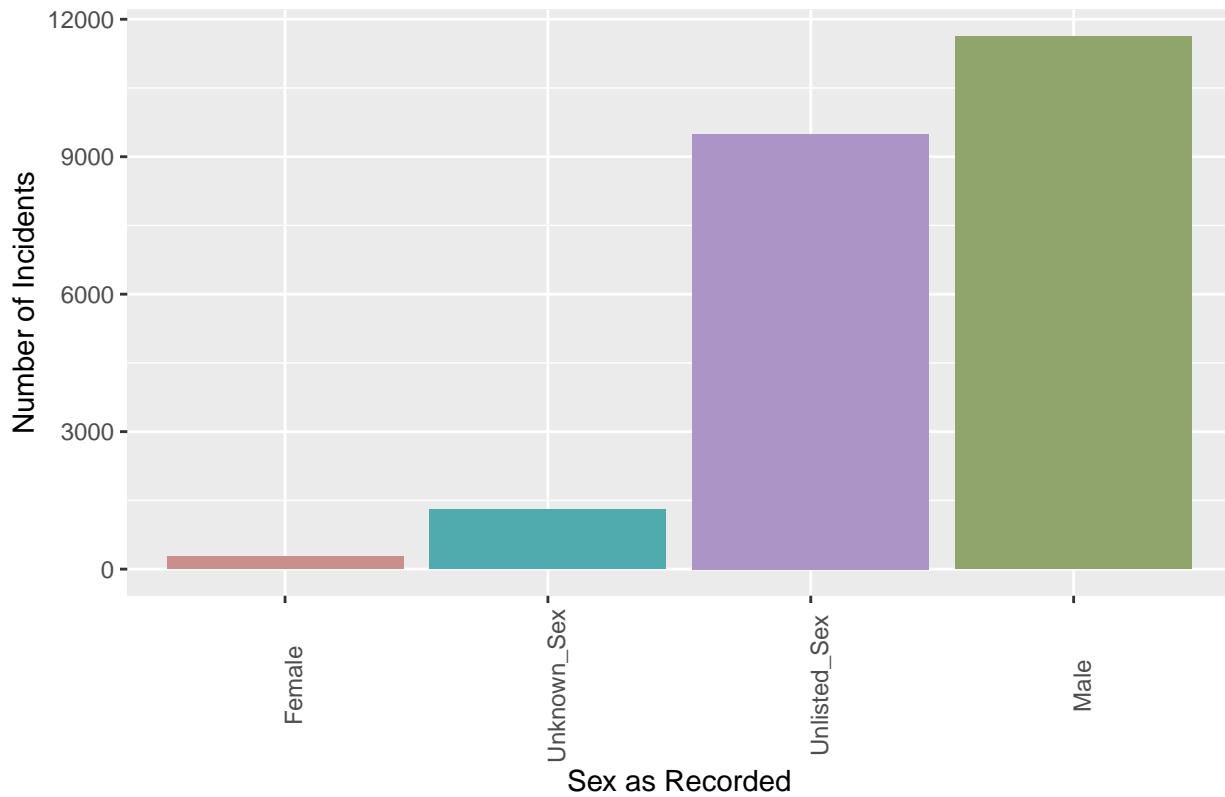
Number of Victims per Sex in BROOKLYN, New York City



```
#creates a bar graph showing the number of victims per sex in the selected borough
```

```
Perp_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar(
    theme(legend.position="none",
      axis.text.x = element_text(angle = 90)) +
    ylab("Number of Incidents") +
    xlab("Sex as Recorded") +
    labs(title = str_c("Number of Shooters per Sex in ", boro, ", New York City"))
```

Number of Shooters per Sex in BROOKLYN, New York City

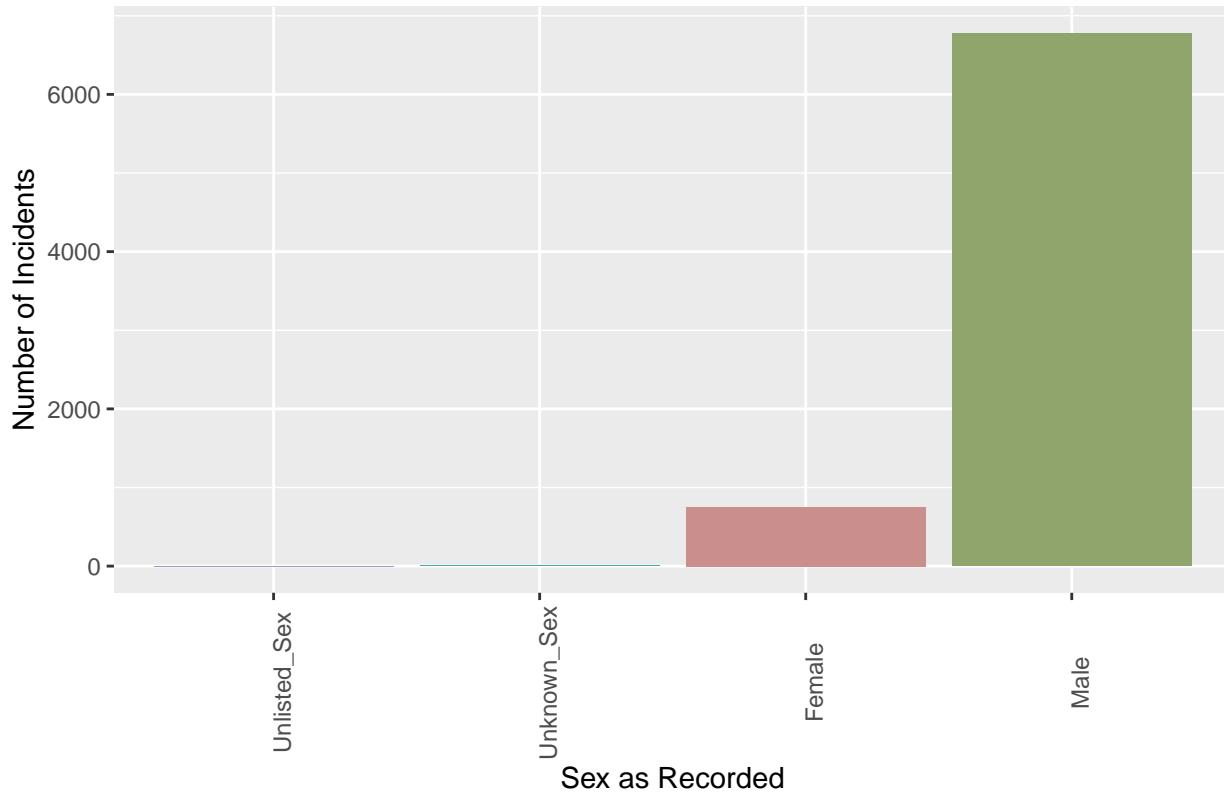


```
#creates a bar graph showing the number of perpetrators per sex in the selected borough

boro <- "MANHATTAN"

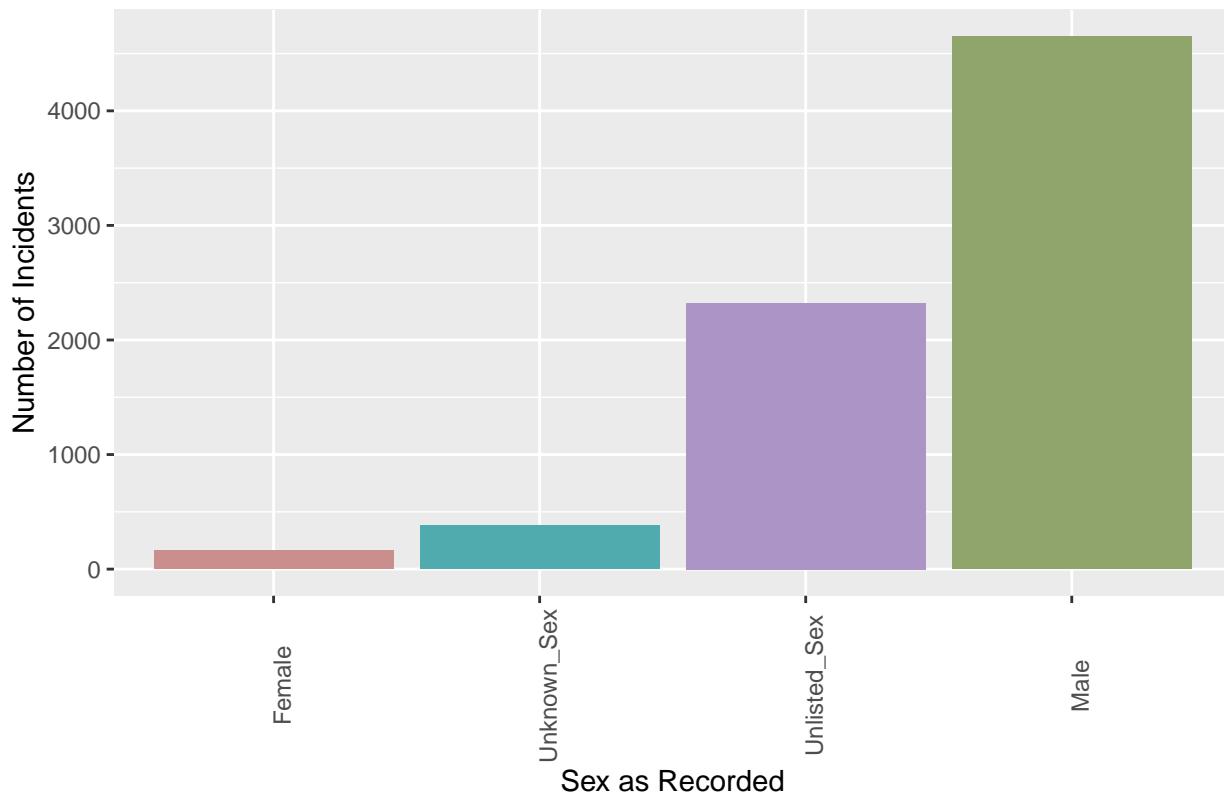
Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar()
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Sex as Recorded") +
  labs(title = str_c("Number of Victims per Sex in ", boro, ", New York City"))
```

Number of Victims per Sex in MANHATTAN, New York City



```
#creates a bar graph showing the number of victims per sex in the selected borough
Perp_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar(
    theme(legend.position="none",
      axis.text.x = element_text(angle = 90)) +
    ylab("Number of Incidents") +
    xlab("Sex as Recorded") +
    labs(title = str_c("Number of Shooters per Sex in ", boro, ", New York City"))
```

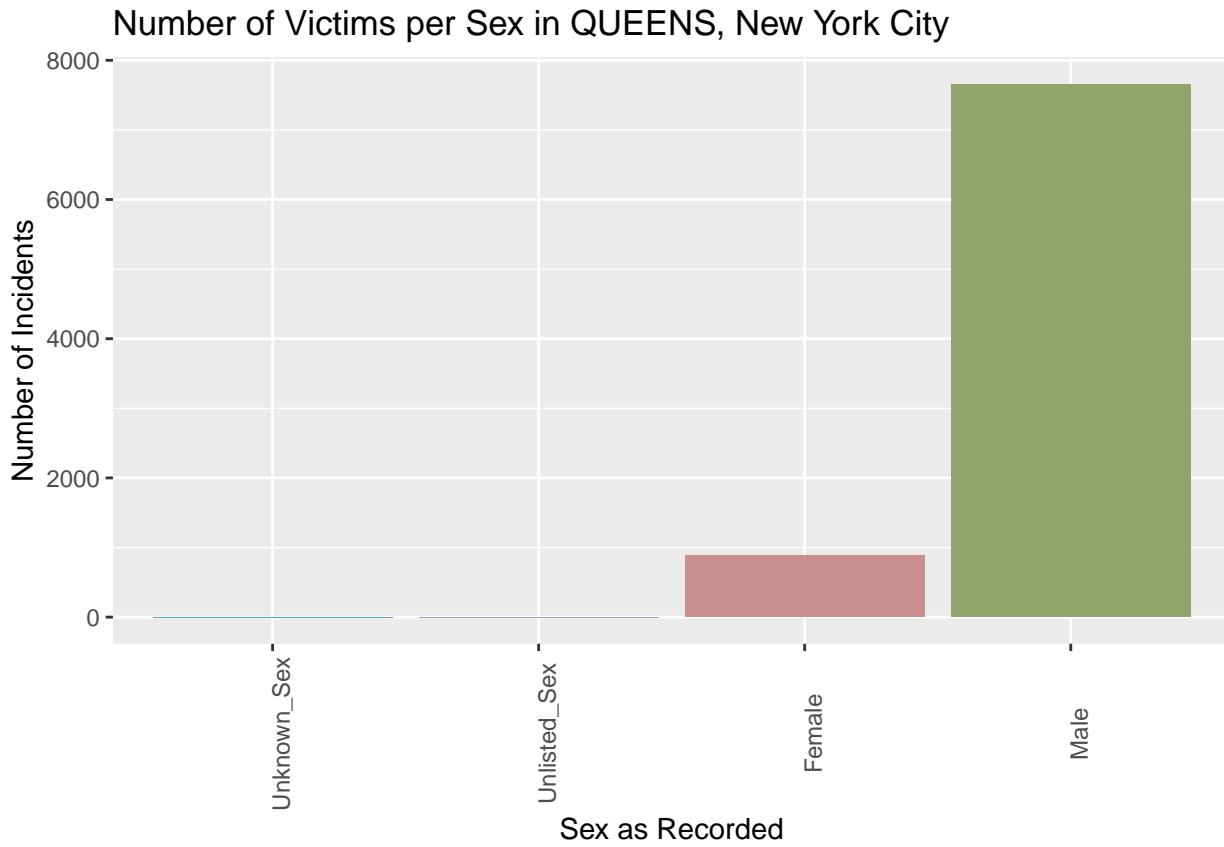
Number of Shooters per Sex in MANHATTAN, New York City



```
#creates a bar graph showing the number of perpetrators per sex in the selected borough

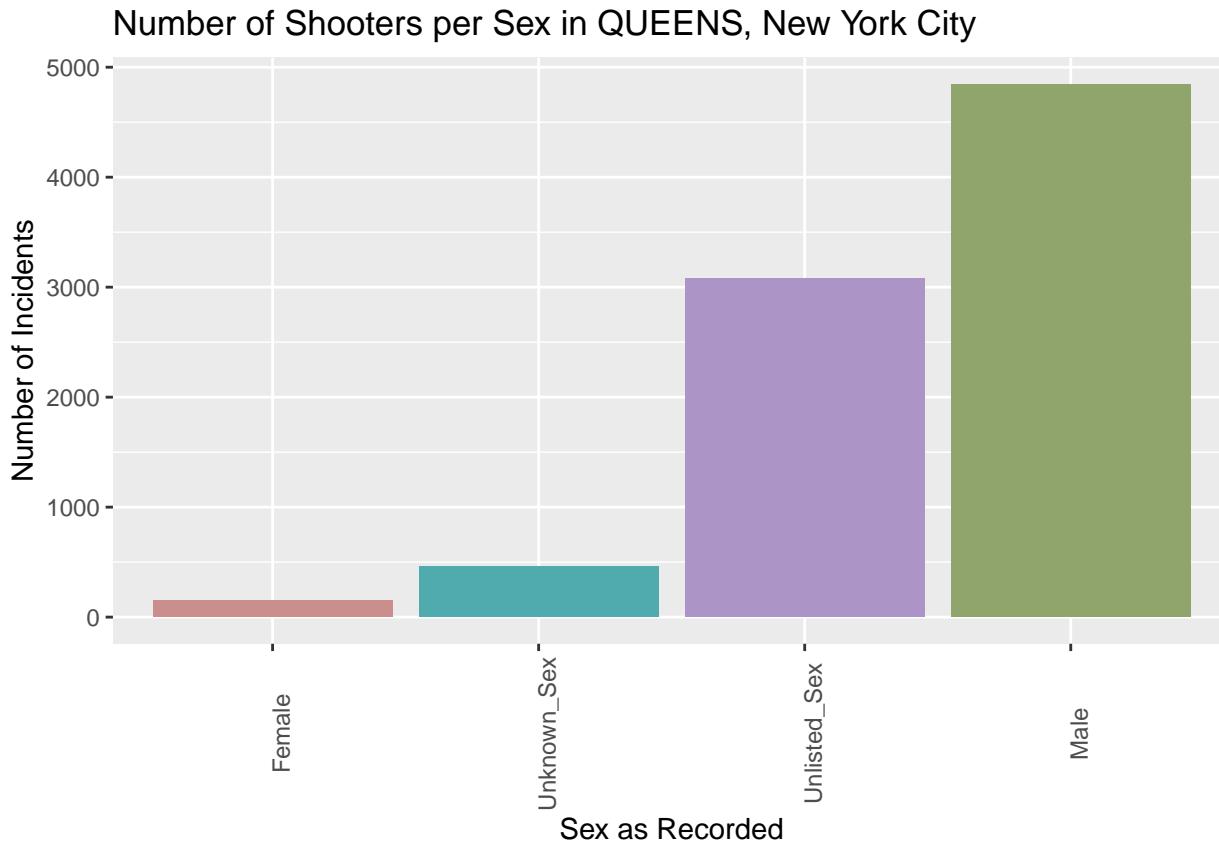
boro <- "QUEENS"

Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar()
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Sex as Recorded") +
  labs(title = str_c("Number of Victims per Sex in ", boro, ", New York City"))
```



```
#creates a bar graph showing the number of victims per sex in the selected borough
```

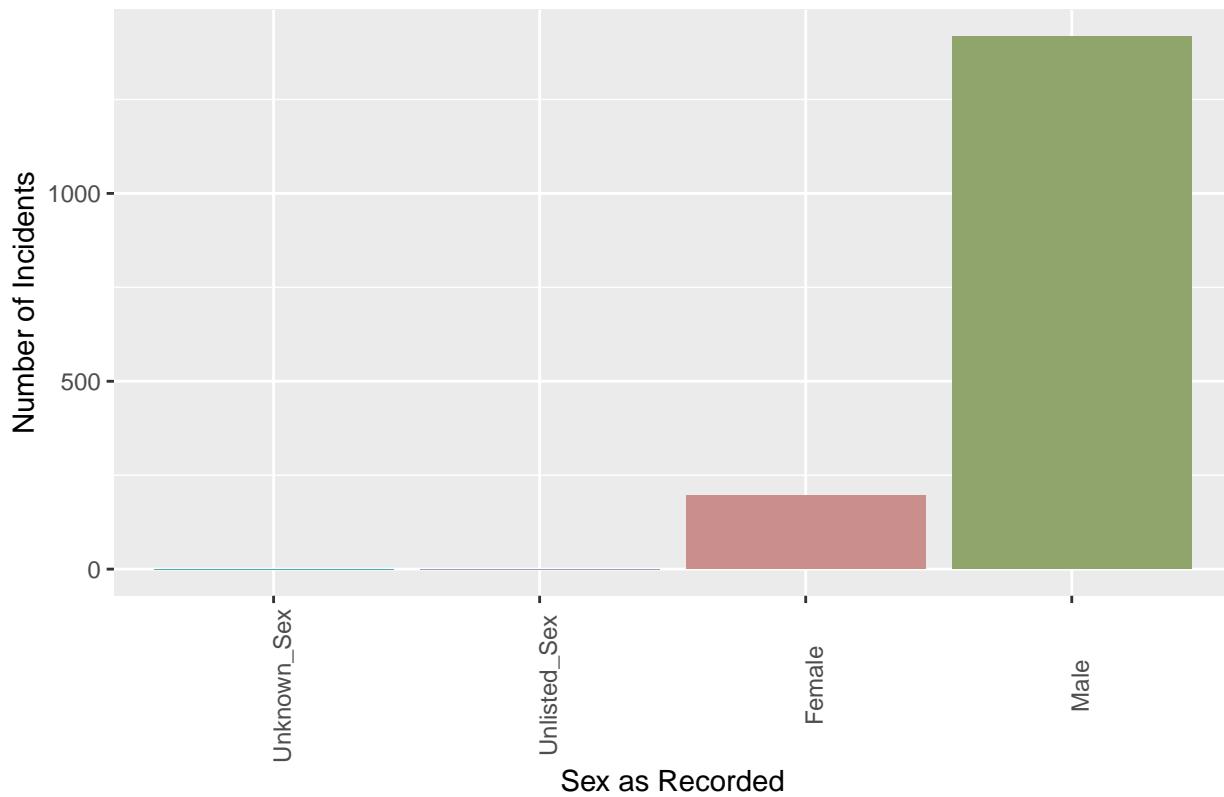
```
Perp_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar(
    theme(legend.position="none",
      axis.text.x = element_text(angle = 90)) +
    ylab("Number of Incidents") +
    xlab("Sex as Recorded") +
    labs(title = str_c("Number of Shooters per Sex in ", boro, ", New York City"))
```



```
#creates a bar graph showing the number of perpetrators per sex in the selected borough
boro <- "STATEN ISLAND"

Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar()
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Sex as Recorded") +
  labs(title = str_c("Number of Victims per Sex in ", boro, ", New York City"))
```

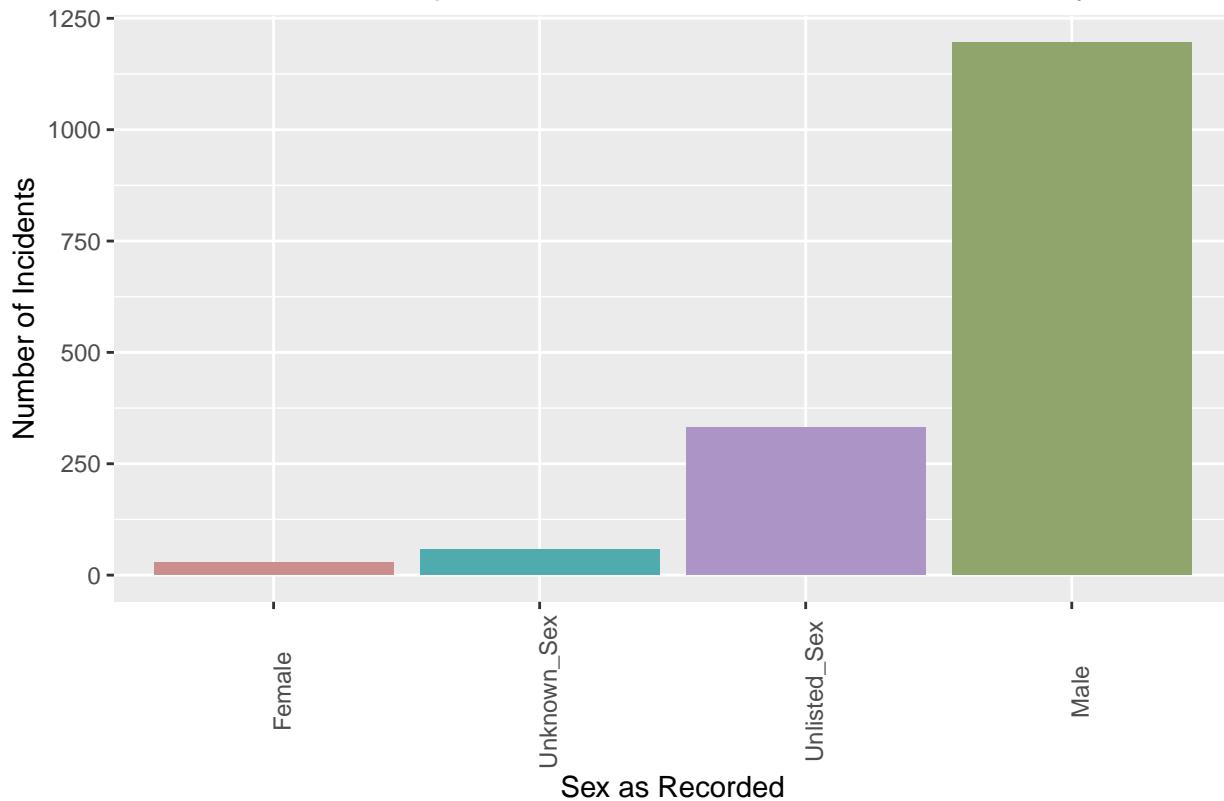
Number of Victims per Sex in STATEN ISLAND, New York City



```
#creates a bar graph showing the number of victims per sex in the selected borough
```

```
Perp_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex, Incidents)), fill = as.factor(Sex), y = Incidents)) + geom_bar(
    theme(legend.position="none",
      axis.text.x = element_text(angle = 90)) +
    ylab("Number of Incidents") +
    xlab("Sex as Recorded") +
    labs(title = str_c("Number of Shooters per Sex in ", boro, ", New York City"))
```

Number of Shooters per Sex in STATEN ISLAND, New York City

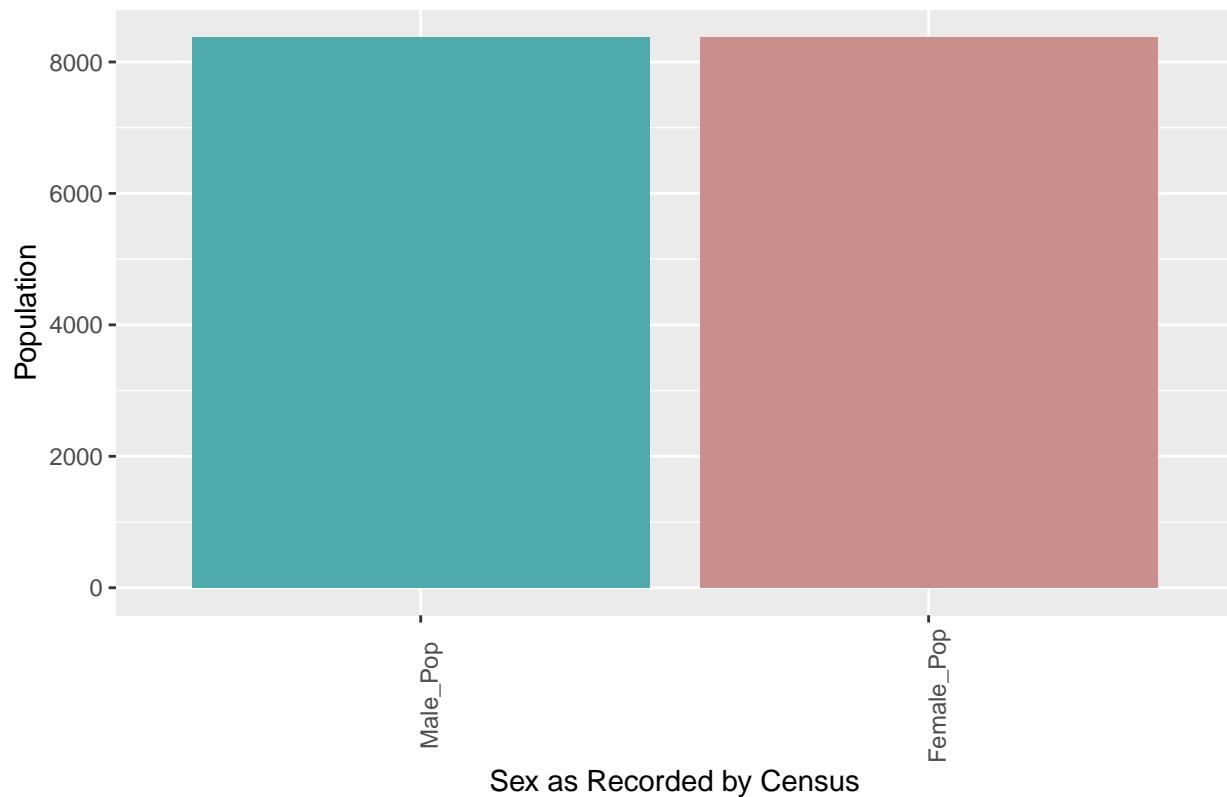


```
#creates a bar graph showing the number of perpetrators per sex in the selected borough
```

This seems to show that men are more often the perpetrators and victims of gun violence more than women across all boroughs. This does not account for intersex people nor can we clear up the unlisted and unknown entries from the data we have. However, it is always possible that this is only due to populations being heavily tilted towards men, so we will look at the populations of the sexes in these boroughs next.

```
boro <- "BRONX"
Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex_Pop, Population)), fill = as.factor(Sex_Pop), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population") +
  xlab("Sex as Recorded by Census") +
  labs(title = str_c("Populations per Sex in ", boro, ", New York City"))
```

Populations per Sex in BRONX, New York City

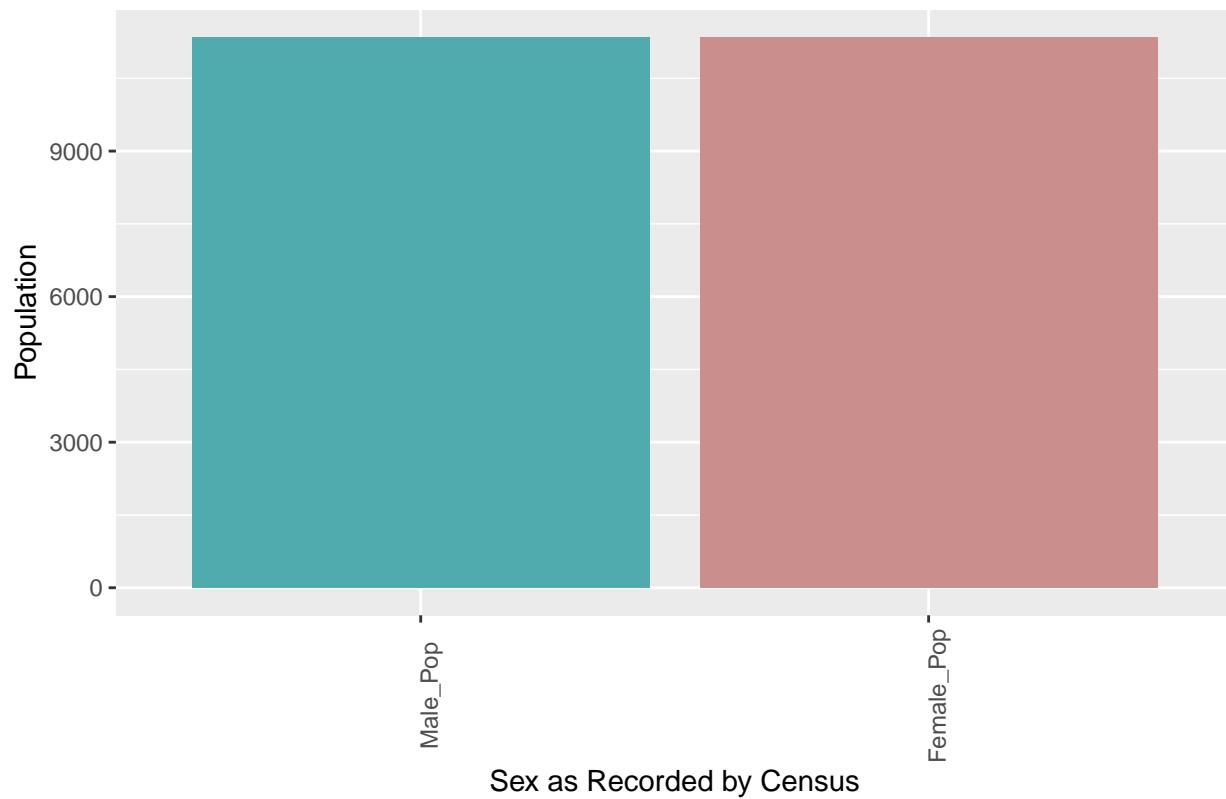


```
#creates a bar graph showing the population per sex in the selected borough
```

```
boro <- "BROOKLYN"
```

```
Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex_Pop, Population)), fill = as.factor(Sex_Pop), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population") +
  xlab("Sex as Recorded by Census") +
  labs(title = str_c("Populations per Sex in ", boro, ", New York City"))
```

Populations per Sex in BROOKLYN, New York City

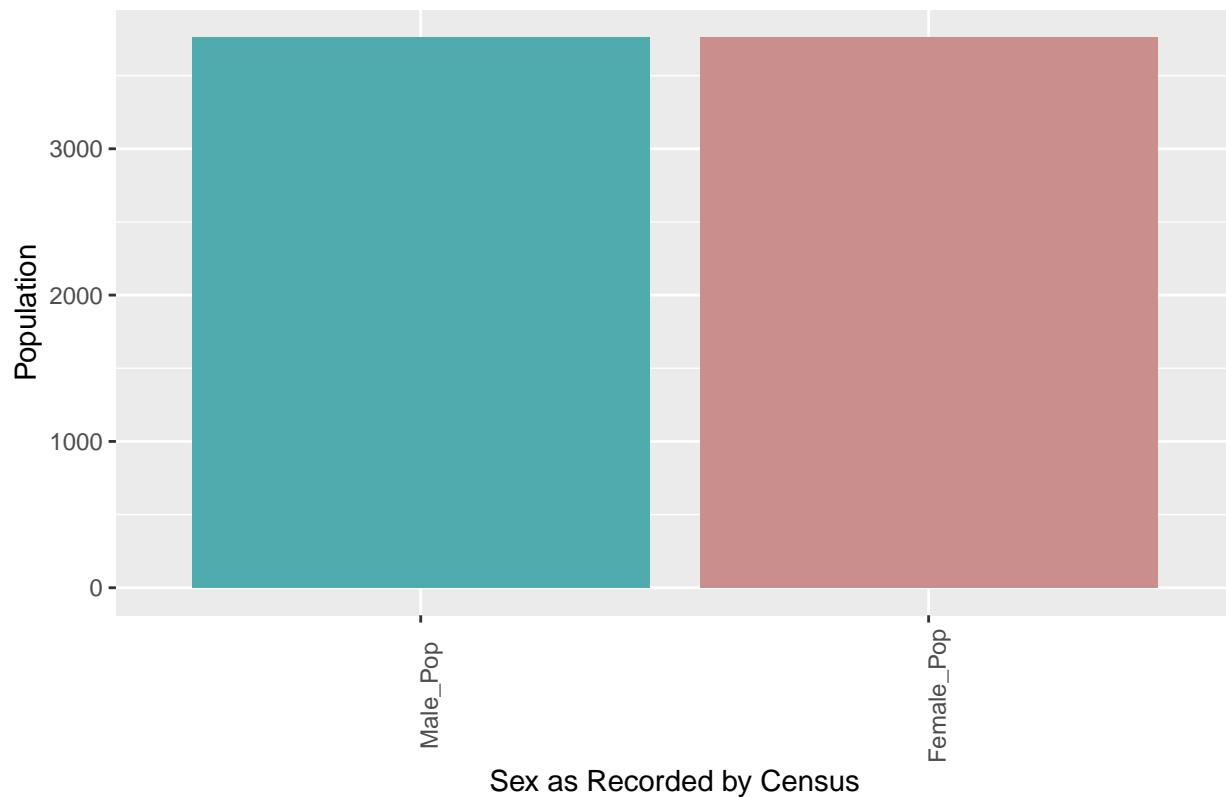


```
#creates a bar graph showing the population per sex in the selected borough
```

```
boro <- "MANHATTAN"
```

```
Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex_Pop, Population)), fill = as.factor(Sex_Pop), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population") +
  xlab("Sex as Recorded by Census") +
  labs(title = str_c("Populations per Sex in ", boro, ", New York City"))
```

Populations per Sex in MANHATTAN, New York City

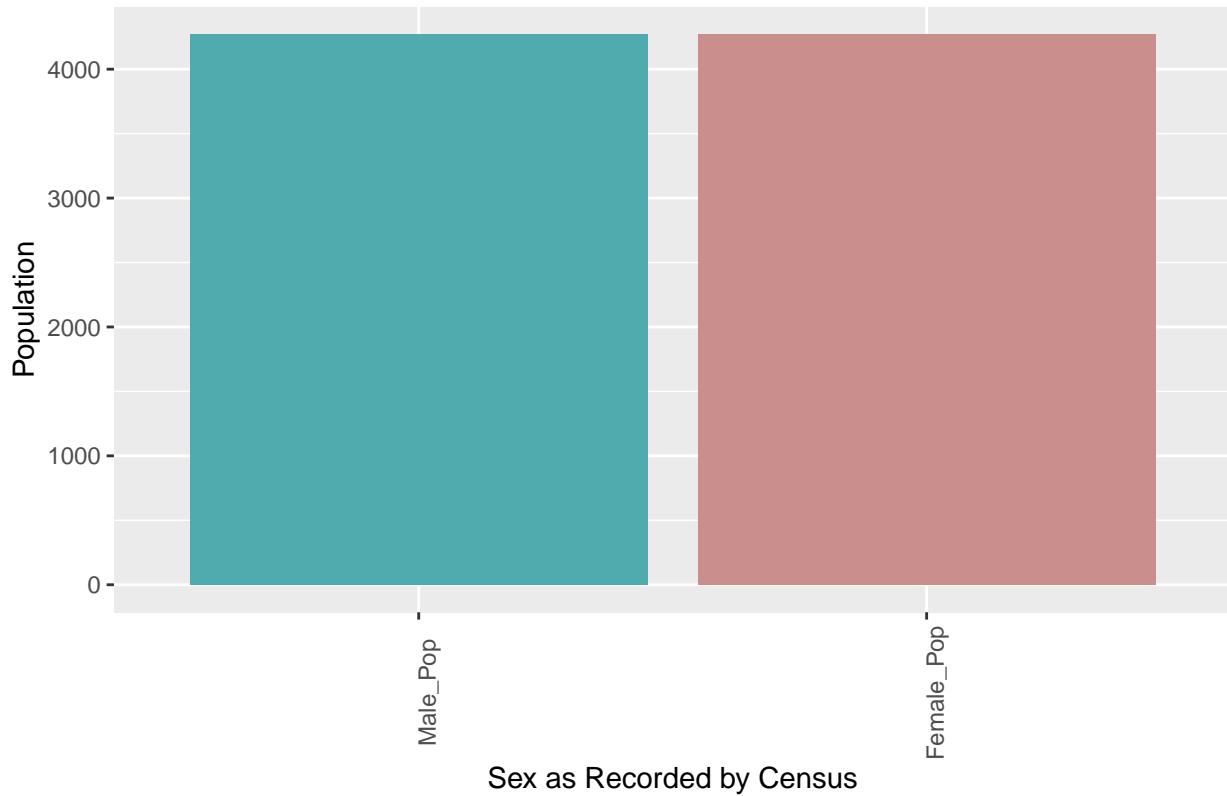


```
#creates a bar graph showing the population per sex in the selected borough
```

```
boro <- "QUEENS"

Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex_Pop, Population)), fill = as.factor(Sex_Pop), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population") +
  xlab("Sex as Recorded by Census") +
  labs(title = str_c("Populations per Sex in ", boro, ", New York City"))
```

Populations per Sex in QUEENS, New York City

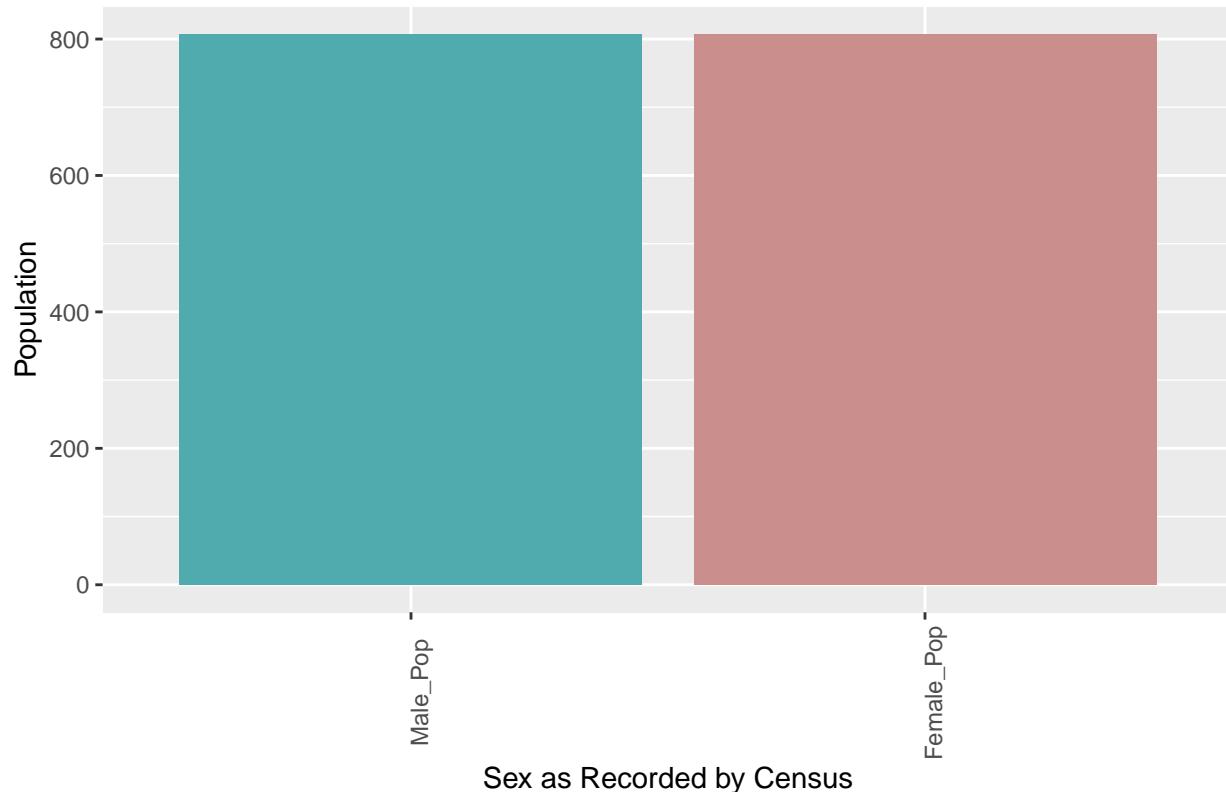


```
#creates a bar graph showing the population per sex in the selected borough
```

```
boro <- "STATEN ISLAND"
```

```
Vic_Sexes %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Sex_Pop, Population)), fill = as.factor(Sex_Pop), y = Incidents)) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population") +
  xlab("Sex as Recorded by Census") +
  labs(title = str_c("Populations per Sex in ", boro, ", New York City"))
```

Populations per Sex in STATEN ISLAND, New York City



```
#creates a bar graph showing the population per sex in the selected borough
```

This clearly shows that the population of men versus women is not related to the number of shooters or victims of gun violence in the boroughs, since the ratio is much closer to 1:1 compared to our gun violence data. Thus, it seems there is a relationship between being male and being a victim or perpetrator of a shooting incident. There is also a large chunk of shooters of unlisted sex which could swing the relationship closer to other sexes, however we don't have good data to analyze for that at this moment.

Now we will see if we can use male population to predict number of shooting incidents, since there seems to be a strong relationship between being male and perpetrating or experiencing gun violence.

Modeling Male Population of a Borough against Shooting Incident Totals This will involve creating a new subset and visualizing it as a graph, then producing a regression model to see if there is a strong relationship between a higher male population and a higher number of shooting incidents. We will also examine this between female population and number of incidents, in case it is just directly related to the population increasing overall. However, since male and female population in all four boroughs is very similar, it is possible this will not reach a solid conclusion.

```
temp <- Vic_Sexes %>%
  group_by(BORO, Sex_Pop, Population) %>%
  summarize(Incidents = sum(Incidents) / 2) %>% #divided by two due to duplicates from Male_Pop and Fem
  ungroup()
```

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

```
#This gives us the male and female populations of each borough as well as the total shooting incidents
```

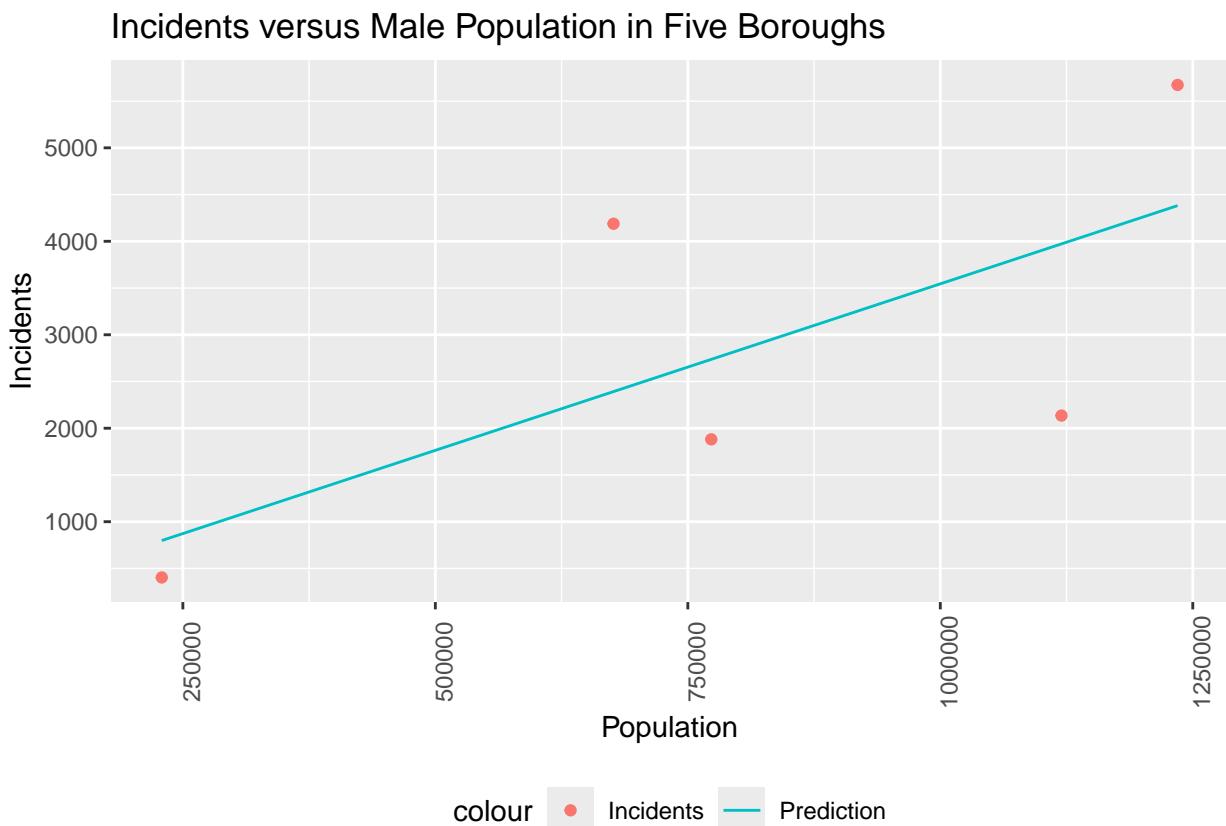
```

temp <- temp %>%
  filter(Sex_Pop == "Male_Pop")
#This filters out the female population entries

mod <- lm(Incidents ~ Population, temp)
#This creates a linear regression of Incidents based on Population from the filtered temp dataframe

temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus Male Population in Five Boroughs")

```



```

male_mod <- mod

temp <- Vic_Sexes %>%
  group_by(BORO, Sex_Pop, Population) %>%
  summarize(Incidents = sum(Incidents) / 2) %>% #divided by two due to duplicates from Male_Pop and Fem
ungroup()

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

```

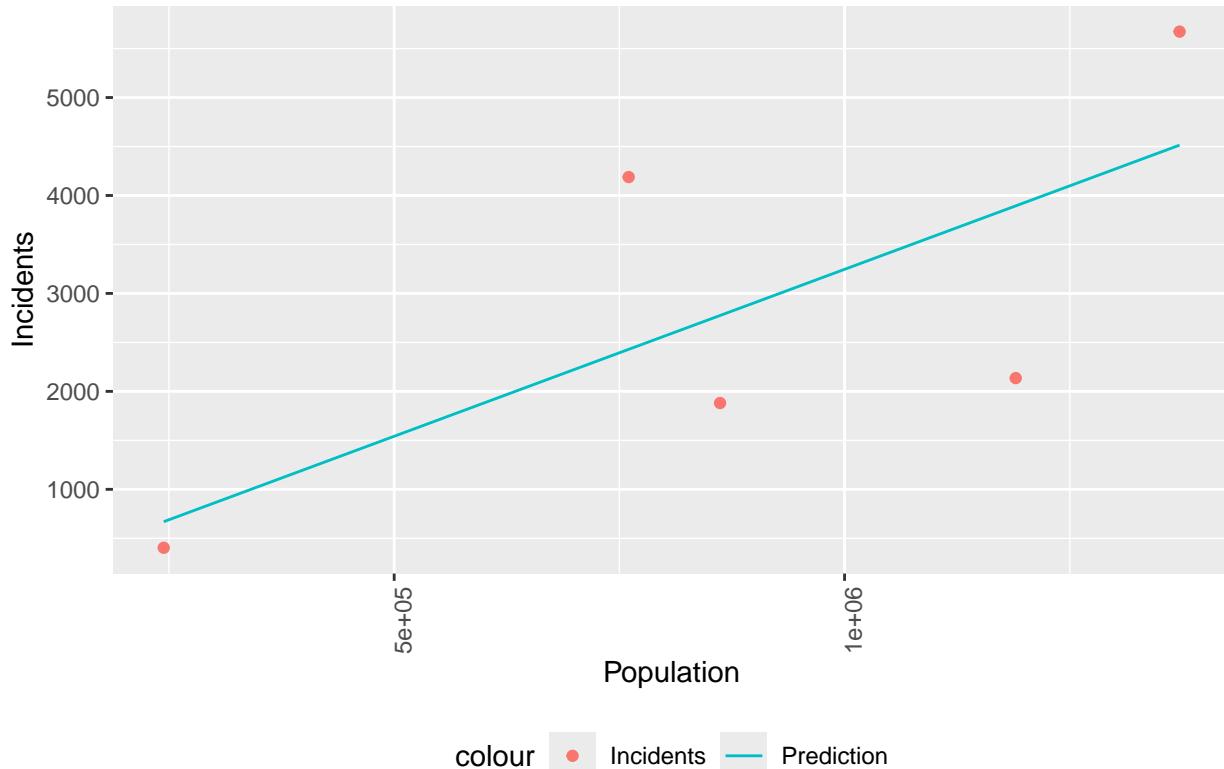
```
#This gives us the male and female populations of each borough as well as the total shooting incidents

temp <- temp %>%
  filter(Sex_Pop == "Female_Pop")
#This filters out the male population entries

mod <- lm(Incidents ~ Population, temp)
#This creates a linear regression of Incidents based on Population from the filtered temp dataframe

temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = Population, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus Female Population in Five Boroughs")
```

Incidents versus Female Population in Five Boroughs



```
print("For male population in raw numbers versus shooting incidents, our coefficients and R-values are:

## [1] "For male population in raw numbers versus shooting incidents, our coefficients and R-values are
summary(male_mod)

##
```

```

## 
## Residuals:
##      1       2       3       4       5 
## 1796.1 1291.1 -855.8 -1836.8 -394.5 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.821e+01 1.936e+03 -0.009   0.993    
## Population   3.563e-03 2.195e-03  1.623   0.203    
## 
## Residual standard error: 1747 on 3 degrees of freedom
## Multiple R-squared:  0.4676, Adjusted R-squared:  0.2901 
## F-statistic: 2.634 on 1 and 3 DF, p-value: 0.203 

print("For female population in raw numbers versus shooting incidents, our coefficients and R-values are")

```

```

## [1] "For female population in raw numbers versus shooting incidents, our coefficients and R-values are"
summary(mod)

```

```

## 
## Call:
## lm(formula = Incidents ~ Population, data = temp)
## 
## Residuals:
##      1       2       3       4       5 
## 1759.0 1159.1 -893.8 -1758.3 -266.0 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.631e+02 1.861e+03 -0.088   0.936    
## Population   3.409e-03 1.923e-03  1.773   0.174    
## 
## Residual standard error: 1673 on 3 degrees of freedom
## Multiple R-squared:  0.5115, Adjusted R-squared:  0.3487 
## F-statistic: 3.142 on 1 and 3 DF, p-value: 0.1744

```

It seems here, that shooting incidents go up with population size regardless of sex, with a linear regression at one degree. Next, we will look at percentages to take away the factor of having more people involved.

```

temp <- Vic_Sex_Percent %>%
  filter(Sex_Pop_Percent == "Male_Pop_Percent") %>%
  filter(Sex_Percent == "Male_Percent")

temp2 <- Vic_Sexes %>%
  filter(Sex_Pop == "Male_Pop") %>%
  group_by(BORO) %>%
  summarize(Incidents = sum(Incidents))

temp <- temp %>%
  left_join(temp2, by = c("BORO")) %>%
  select(-c(Incidents_Percent, Sex_Percent))

mod <- lm(Incidents ~ Population_Percent, temp)

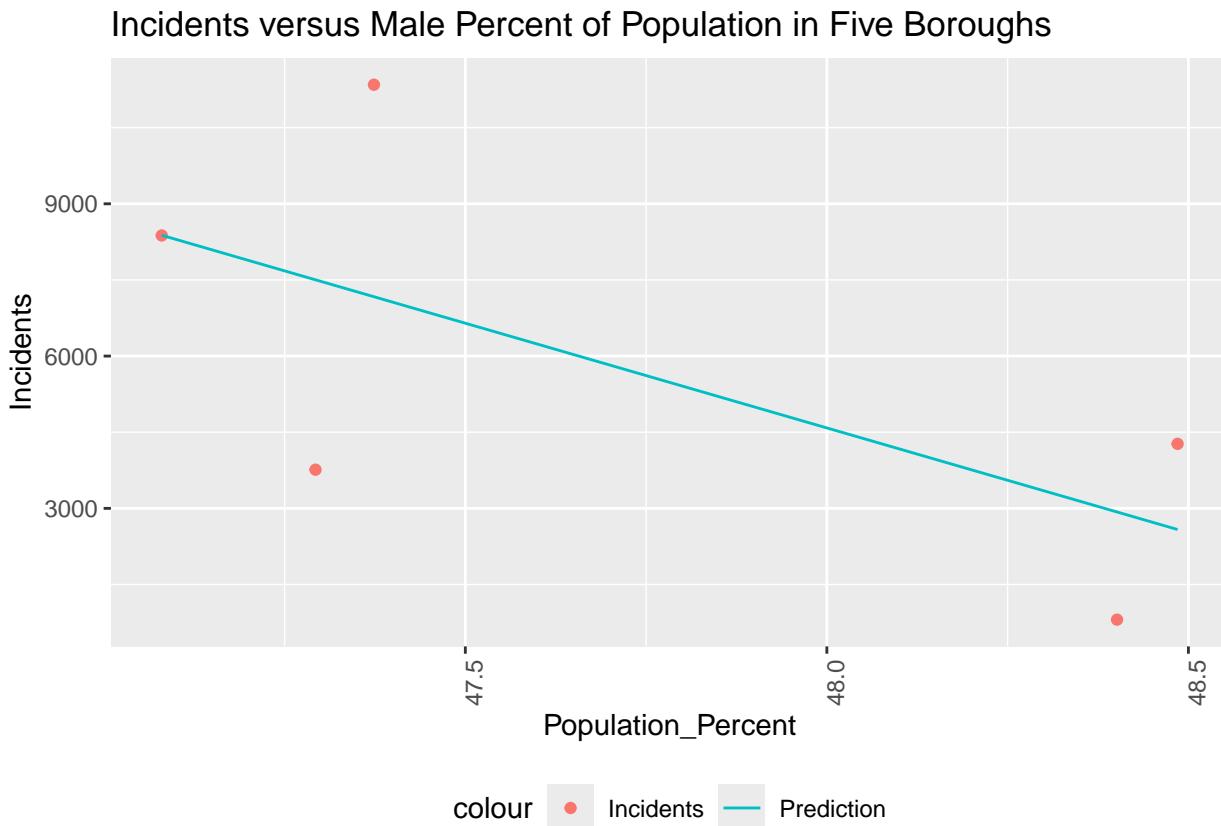
temp <- temp %>% mutate(pred = predict(mod))

```

```

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus Male Percent of Population in Five Boroughs")

```



Here, we see a relationship where total incidents actually go down with a higher male percent of the population. So although we can say men are more likely to be involved as a shooter or victim of shooting incidents, we cannot say that populations with higher percentages of men are more likely to have higher numbers of shooting incidents.

Transform Race Data for Visualization

Next we will transform our race demographics for visualization. The first step is isolating our race data from the rest of our demographics.

```

Vic_Races <- Vic_By_Boro_Totals_with_Pop %>%
  pivot_longer(cols = c(BLACK,BLACK_HISPANIC,WHITE,WHITE_HISPANIC, ASIAN_PACIFIC_ISLANDER, AMERICAN_INDIAN),
               names_to = "Race_Group",
               values_to = "Incidents") %>%
  pivot_longer(cols = c(BLACK_POP, BLACK_HISPANIC_POP, WHITE_POP, WHITE_HISPANIC_POP, ASIAN_PACIFIC_ISLANDER_POP),
               names_to = "Race_Group_Pop",
               values_to = "Population") %>%
  select(c(BORO, Race_Group, Incidents, Race_Group_Pop, Population))

```

```

#Pivots race groups and race group total populations into four columns for visualization

Perp_Races <- Perp_By_Boro_Totals_with_Pop %>%
  pivot_longer(cols = c(BLACK,BLACK_HISPANIC,WHITE,WHITE_HISPANIC, ASIAN_PACIFIC_ISLANDER, AMERICAN_INDIAN_ALASKAN_NATIVE),
               names_to = "Race_Group",
               values_to = "Incidents") %>%
  pivot_longer(cols = c(BLACK_POP, BLACK_HISPANIC_POP, WHITE_POP, WHITE_HISPANIC_POP, ASIAN_PACIFIC_ISLANDER_POP),
               names_to = "Race_Group_Pop",
               values_to = "Population") %>%
  select(c(BORO, Race_Group, Incidents, Race_Group_Pop, Population))
#Pivots race groups and race group total populations into four columns for visualization

Vic_Race_Percent <- Vic_By_Boro_Totals_with_Pop %>%
  mutate(BLACK_Percent = BLACK* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         BLACK_HISPANIC_Percent = BLACK_HISPANIC* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         WHITE_Percent = WHITE* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         WHITE_HISPANIC_Percent = WHITE* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         ASIAN_PACIFIC_ISLANDER_Percent = ASIAN_PACIFIC_ISLANDER * 100 / Total_Population,
         AMERICAN_INDIAN_ALASKAN_NATIVE_Percent = AMERICAN_INDIAN_ALASKAN_NATIVE * 100 / Total_Population,
         UNLISTED_RACE_Percent = UNLISTED_RACE * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         UNKNOWN_RACE_Percent = UNKNOWN_RACE * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         WHITE_POP_Percent = WHITE_POP * 100 / Total_Population,
         WHITE_HISPANIC_POP_Percent = WHITE_HISPANIC * 100 / Total_Population,
         BLACK_POP_Percent = BLACK_POP * 100 / Total_Population,
         BLACK_HISPANIC_POP_Percent = BLACK_HISPANIC_POP * 100 / Total_Population,
         ASIAN_PACIFIC_ISLANDER_POP_Percent = ASIAN_PACIFIC_ISLANDER_POP * 100 / Total_Population,
         AMERICAN_INDIAN_ALASKAN_NATIVE_POP_Percent = AMERICAN_INDIAN_ALASKAN_NATIVE_POP * 100 / Total_Population,
         UNKNOWN_OTHER_RACE_POP_Percent = UNKNOWN_OTHER_RACE_POP * 100 / Total_Population) %>%
  pivot_longer(cols = c(BLACK_Percent, BLACK_HISPANIC_Percent, WHITE_Percent, WHITE_HISPANIC_Percent,
                       names_to = "Race_Group_Percent",
                       values_to = "Incidents_Percent") %>%
  pivot_longer(cols = c(BLACK_POP_Percent, BLACK_HISPANIC_POP_Percent, WHITE_POP_Percent, WHITE_HISPANIC_POP_Percent,
                       names_to = "Race_Group_Pop_Percent",
                       values_to = "Population_Percent") %>%
  select(c(BORO, Race_Group_Percent, Incidents_Percent, Race_Group_Pop_Percent, Population_Percent))
#creates race percents dataframe for victims

Perp_Race_Percent <- Perp_By_Boro_Totals_with_Pop %>%
  mutate(BLACK_Percent = BLACK* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         BLACK_HISPANIC_Percent = BLACK_HISPANIC* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         WHITE_Percent = WHITE* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         WHITE_HISPANIC_Percent = WHITE* 100 /(Male + Female + Unknown_Sex + Unlisted_Sex),
         ASIAN_PACIFIC_ISLANDER_Percent = ASIAN_PACIFIC_ISLANDER * 100 / Total_Population,
         AMERICAN_INDIAN_ALASKAN_NATIVE_Percent = AMERICAN_INDIAN_ALASKAN_NATIVE * 100 / Total_Population,
         UNLISTED_RACE_Percent = UNLISTED_RACE * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         UNKNOWN_RACE_Percent = UNKNOWN_RACE * 100 / (Male + Female + Unknown_Sex + Unlisted_Sex),
         WHITE_POP_Percent = WHITE_POP * 100 / Total_Population,
         WHITE_HISPANIC_POP_Percent = WHITE_HISPANIC * 100 / Total_Population,
         BLACK_POP_Percent = BLACK_POP * 100 / Total_Population,
         BLACK_HISPANIC_POP_Percent = BLACK_HISPANIC_POP * 100 / Total_Population,
         ASIAN_PACIFIC_ISLANDER_POP_Percent = ASIAN_PACIFIC_ISLANDER_POP * 100 / Total_Population,
         AMERICAN_INDIAN_ALASKAN_NATIVE_POP_Percent = AMERICAN_INDIAN_ALASKAN_NATIVE_POP * 100 / Total_Population,
         UNKNOWN_OTHER_RACE_POP_Percent = UNKNOWN_OTHER_RACE_POP * 100 / Total_Population) %>%

```

```

pivot_longer(cols = c(BLACK_Percent, BLACK_HISPANIC_Percent, WHITE_Percent, WHITE_HISPANIC_Percent,
                     names_to = "Race_Group_Percent",
                     values_to = "Incidents_Percent") %>%
pivot_longer(cols = c(BLACK_POP_Percent, BLACK_HISPANIC_POP_Percent, WHITE_POP_Percent, WHITE_HISPANIC_POP_Percent,
                     names_to = "Race_Group_Pop_Percent",
                     values_to = "Population_Percent") %>%
select(c(BORO, Race_Group_Percent, Incidents_Percent, Race_Group_Pop_Percent, Population_Percent))
#creates race percents dataframe for perpetrators

```

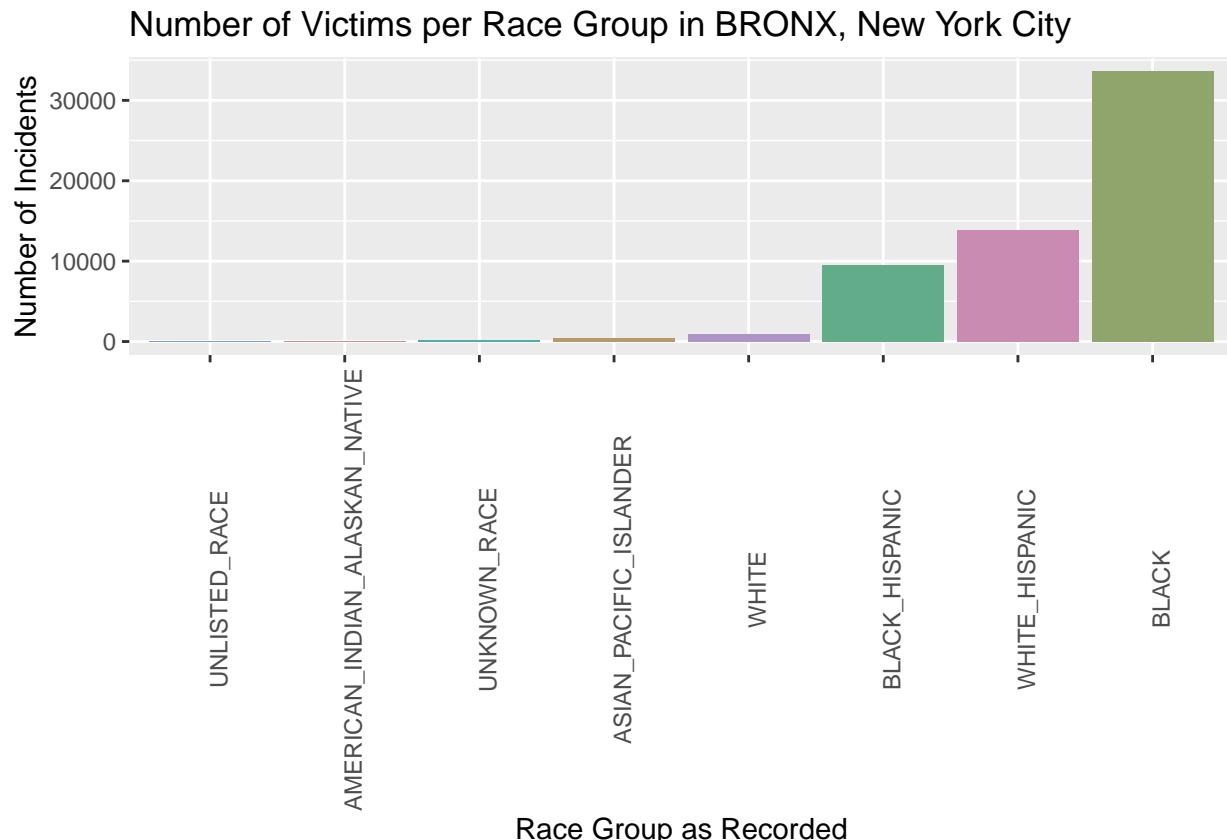
Now that the race group data has been isolated and transformed, we will visualize the data.

Visualizing Race Data We will start with just a comparison of victims by race group per borough.

```
boro <- "BRONX"
```

```
Vic_Races %>%
```

```
filter(BORO == boro) %>%
ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
       theme(legend.position="none",
             axis.text.x = element_text(angle = 90)) +
ylab("Number of Incidents") +
xlab("Race Group as Recorded") +
labs(title = str_c("Number of Victims per Race Group in ", boro, ", New York City"))
```



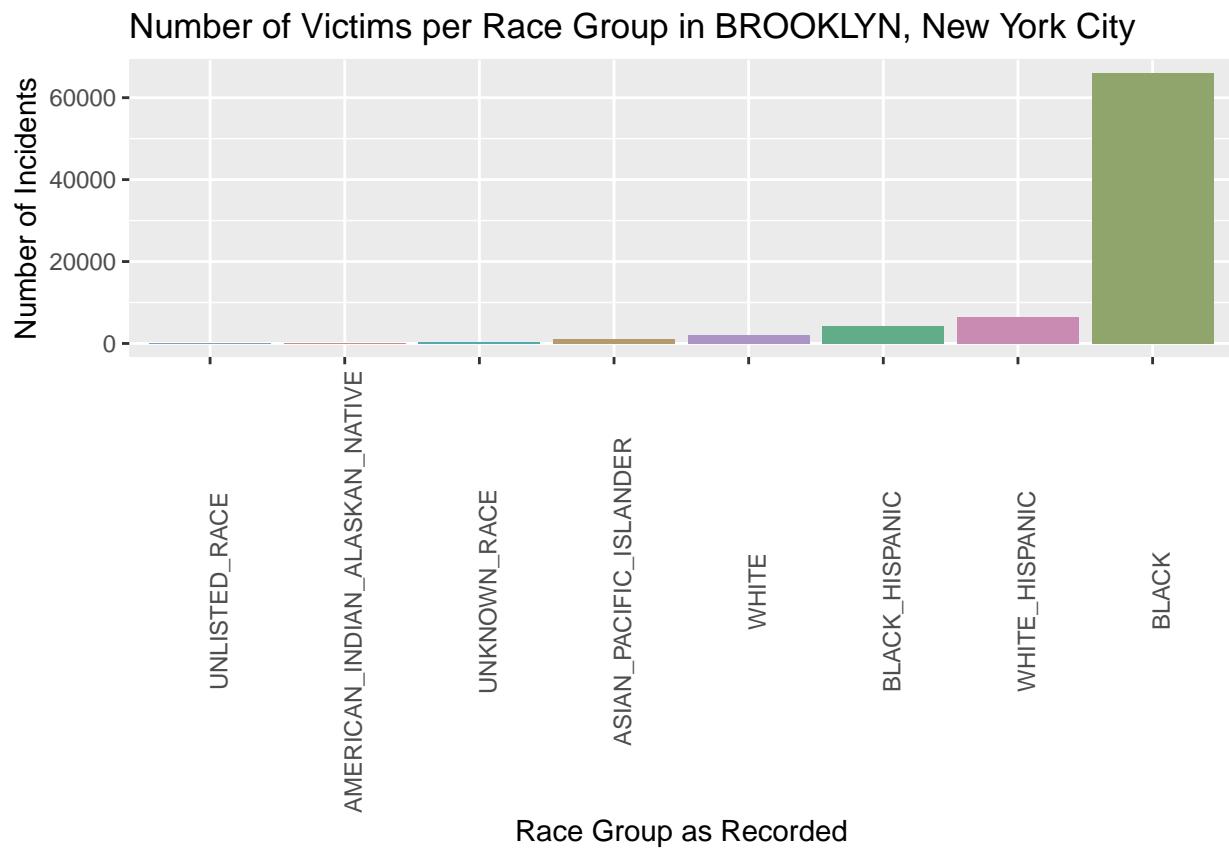
```
#creates bar plot of incidents per race group for the borough listed
```

```
boro <- "BROOKLYN"
```

```

Vic_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
    axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Victims per Race Group in ", boro, ", New York City"))

```



```

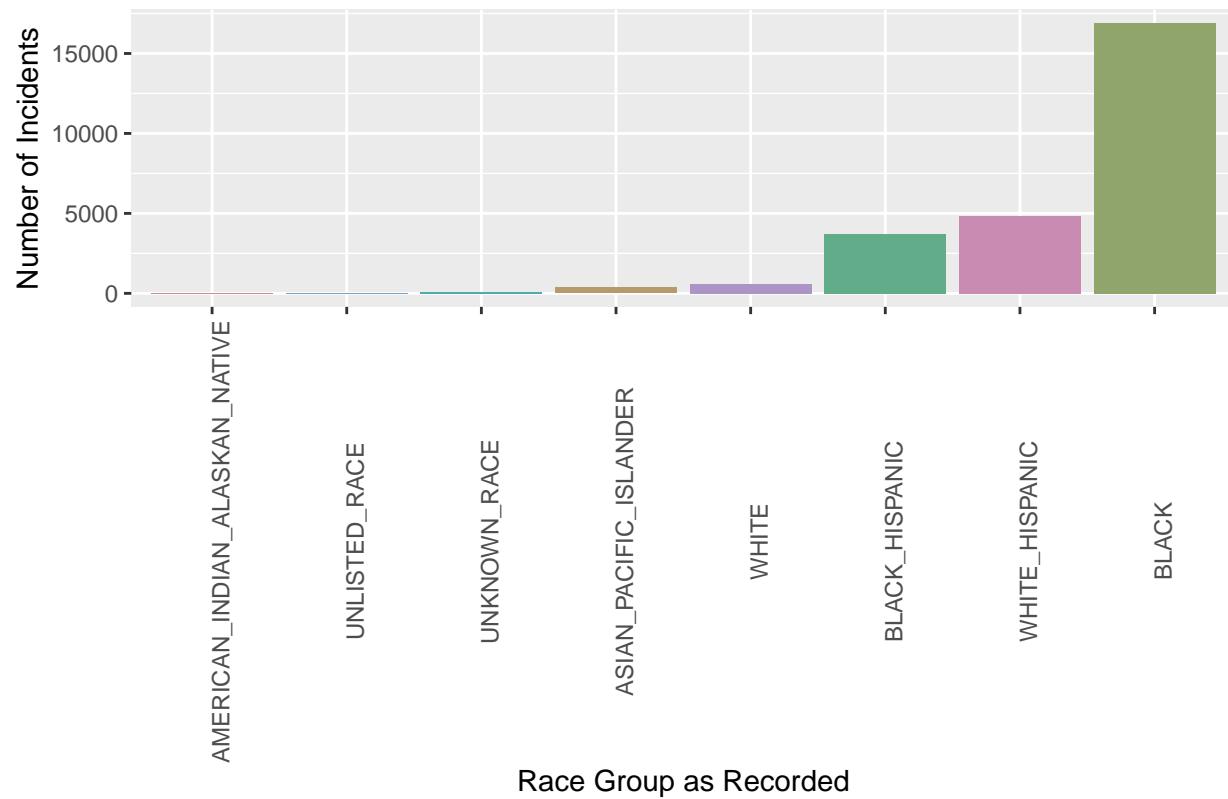
#creates bar plot of incidents per race group for the borough listed

boro <- "MANHATTAN"

Vic_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
    axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Victims per Race Group in ", boro, ", New York City"))

```

Number of Victims per Race Group in MANHATTAN, New York City

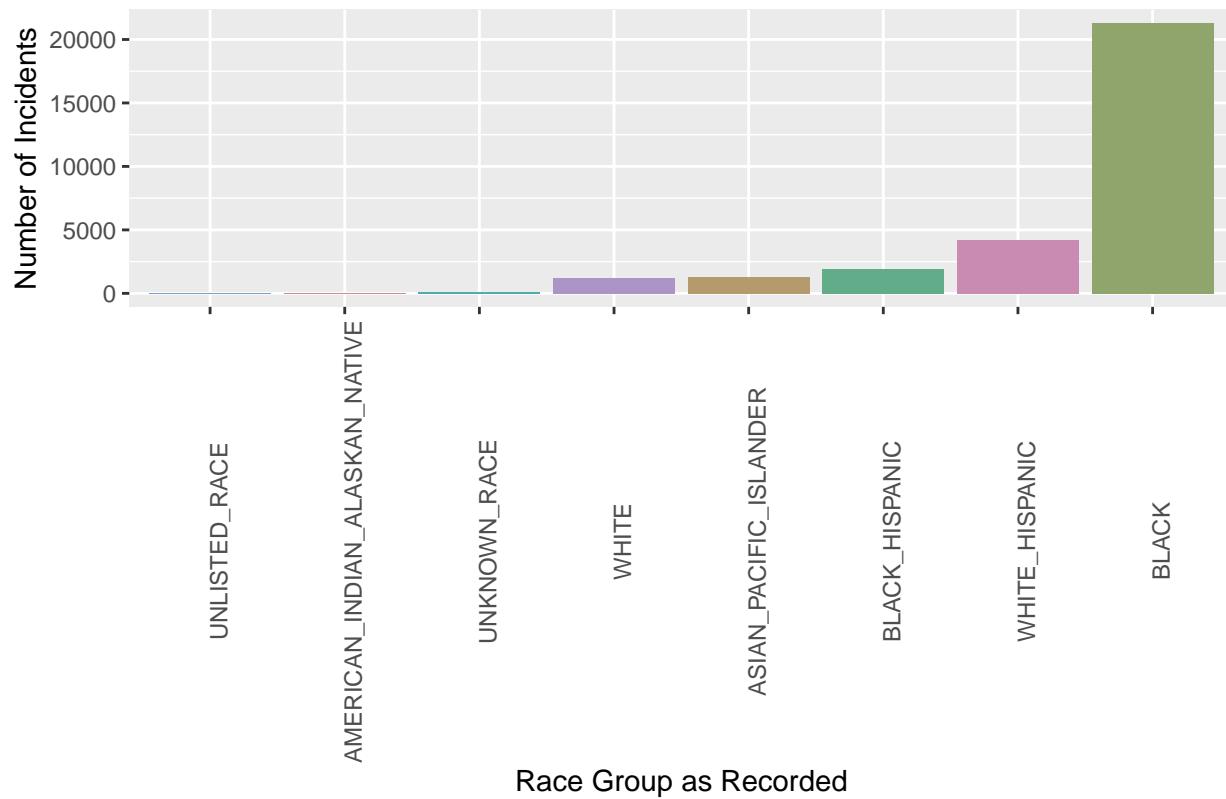


```
#creates bar plot of incidents per race group for the borough listed

boro <- "QUEENS"

Vic_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Victims per Race Group in ", boro, ", New York City"))
```

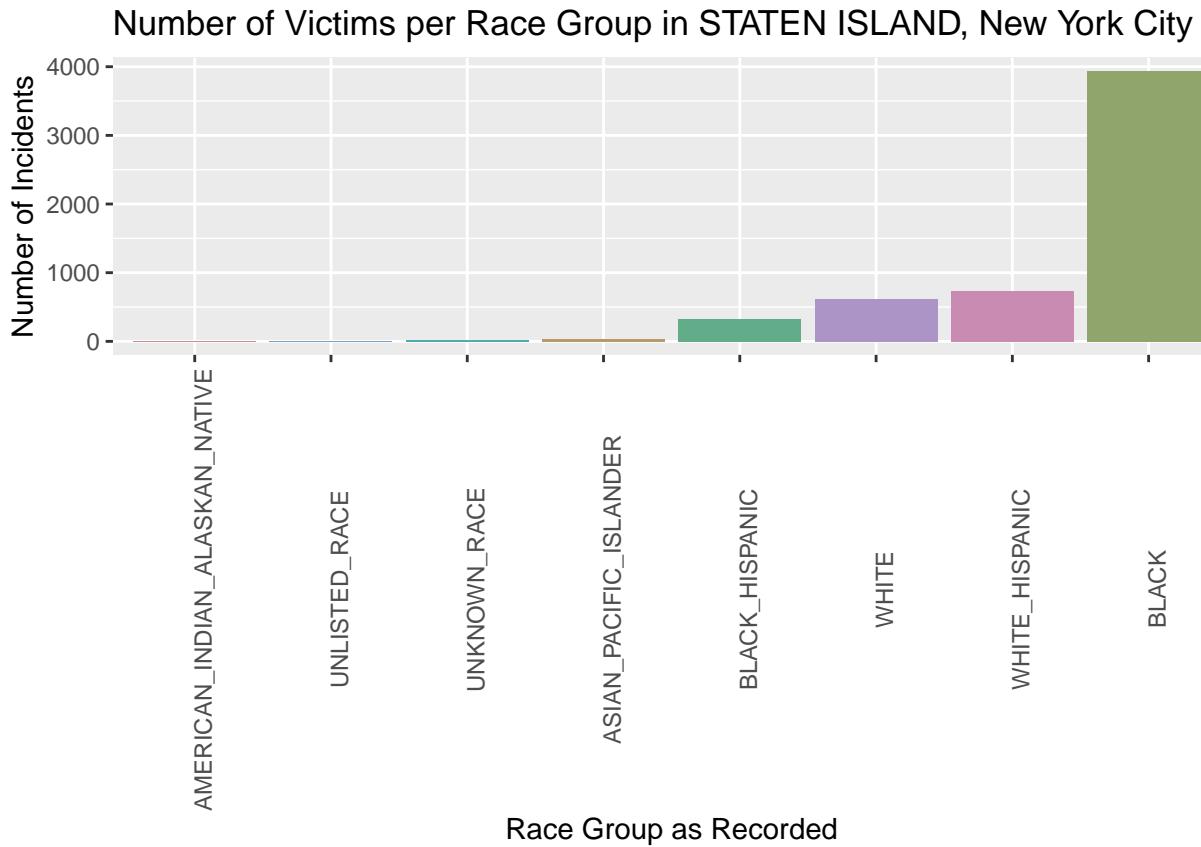
Number of Victims per Race Group in QUEENS, New York City



```
#creates bar plot of incidents per race group for the borough listed

boro <- "STATEN ISLAND"

Vic_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
  axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Victims per Race Group in ", boro, ", New York City"))
```



```
#creates bar plot of incidents per race group for the borough listed
```

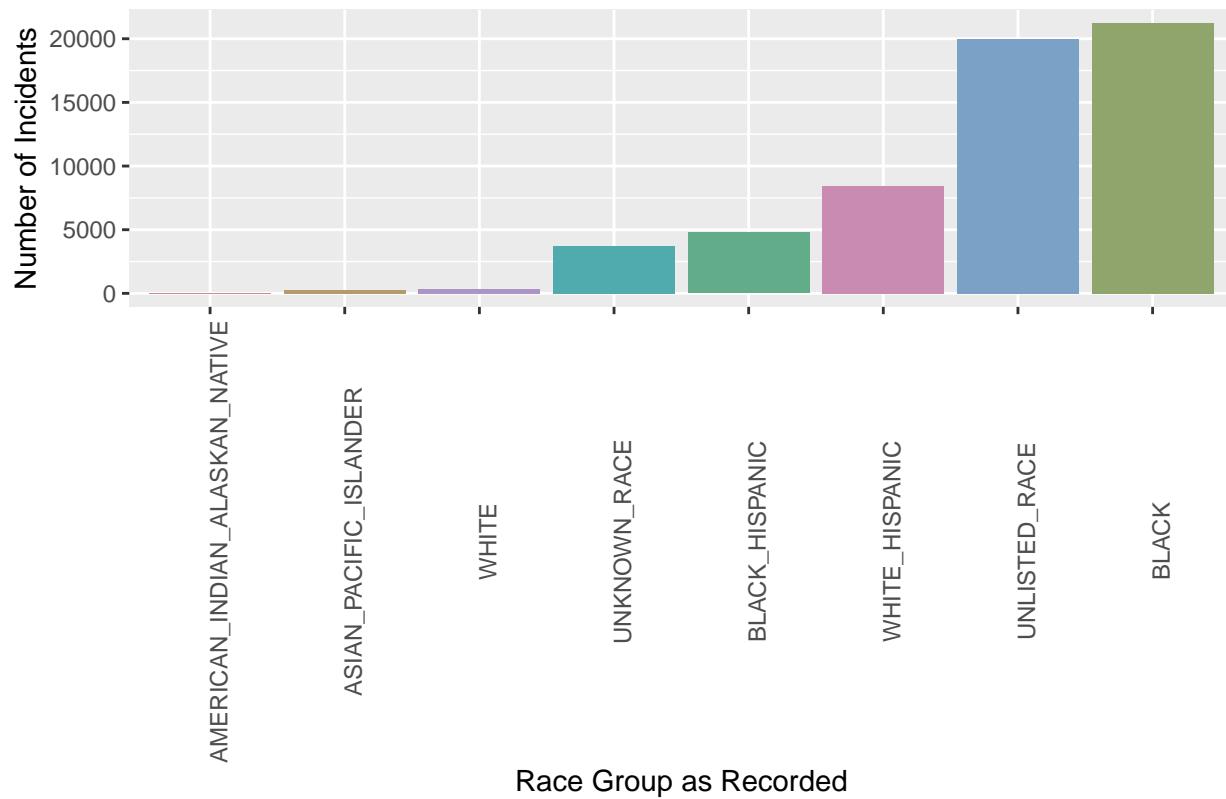
We can see here that Black and Hispanic (white and Black, specifically) people are the majority of victims of gun violence in all five boroughs. We don't know if the Hispanic population referenced here are native or Spanish since the term is often used for both, but we do know that the US has more native Hispanic people than Spanish Hispanic people.

Next, we will look at the perpetrators by race group.

```
boro <- "BRONX"
```

```
Perp_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
  axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Perpetrators per Race Group in ", boro, ", New York City"))
```

Number of Perpetrators per Race Group in BRONX, New York City

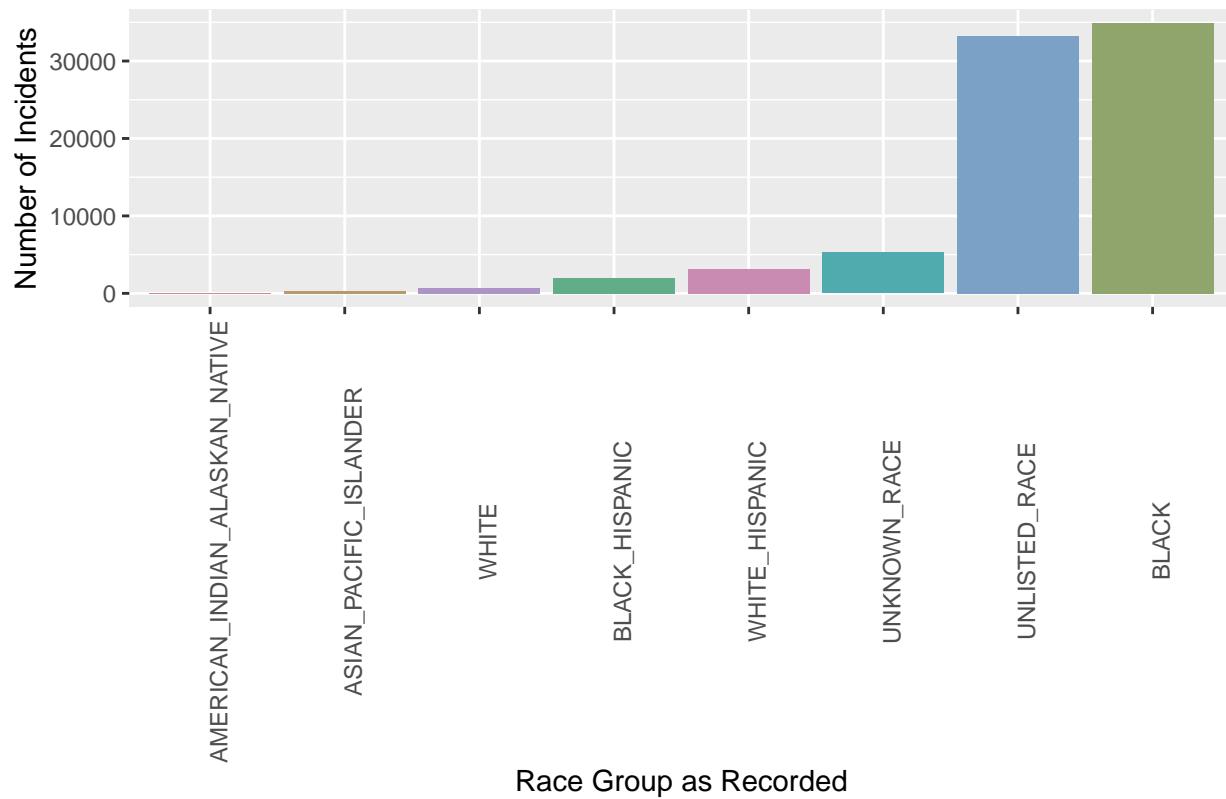


```
#creates bar plot of incidents per race group for the borough listed

boro <- "BROOKLYN"

Perp_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Perpetrators per Race Group in ", boro, ", New York City"))
```

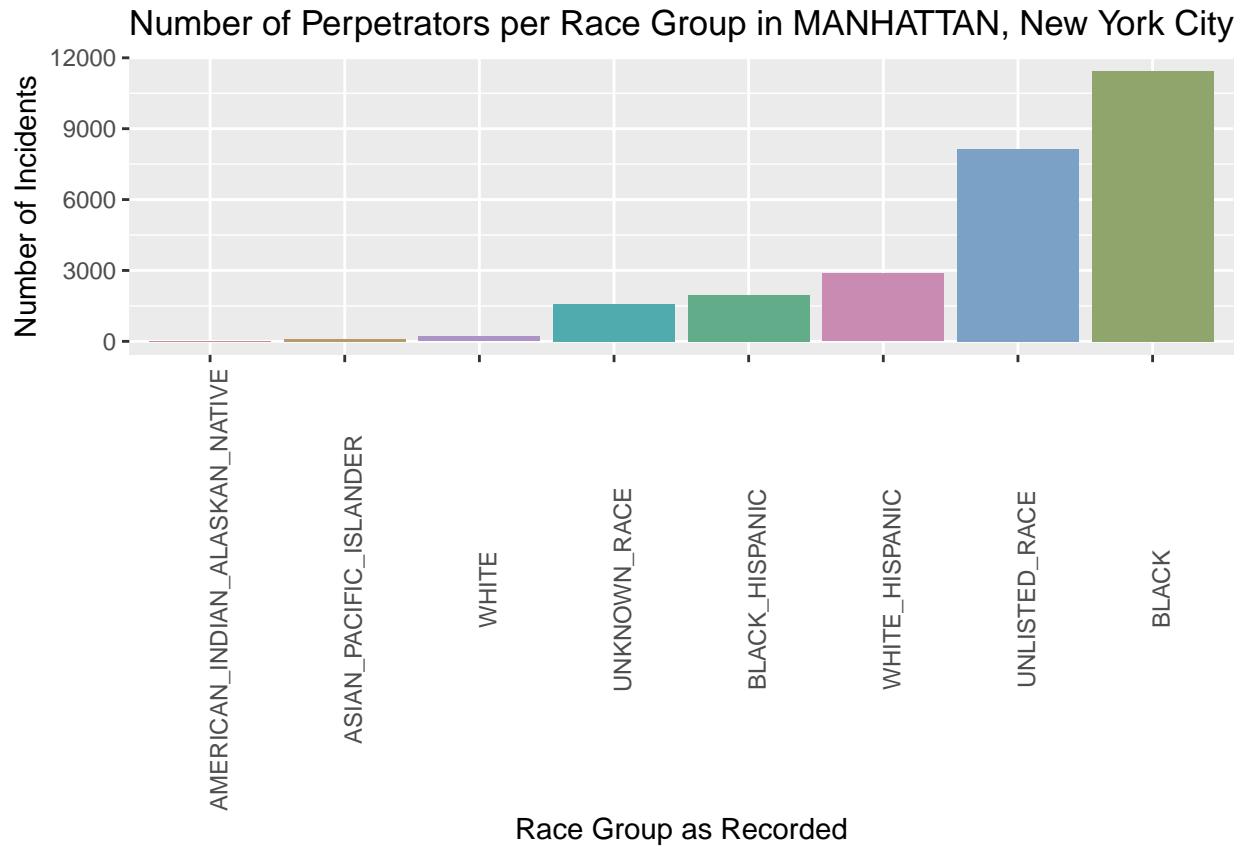
Number of Perpetrators per Race Group in BROOKLYN, New York City



```
#creates bar plot of incidents per race group for the borough listed

boro <- "MANHATTAN"

Perp_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Perpetrators per Race Group in ", boro, ", New York City"))
```

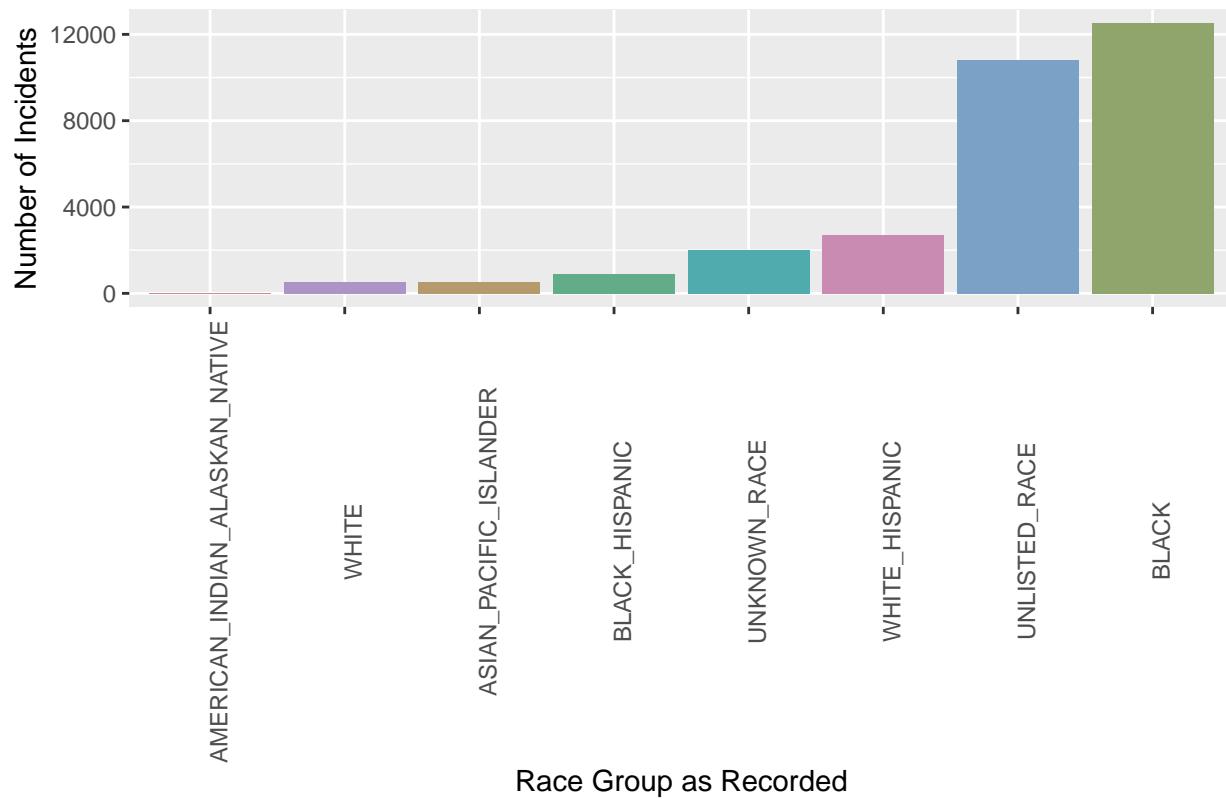


```
#creates bar plot of incidents per race group for the borough listed

boro <- "QUEENS"

Perp_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Perpetrators per Race Group in ", boro, ", New York City"))
```

Number of Perpetrators per Race Group in QUEENS, New York City

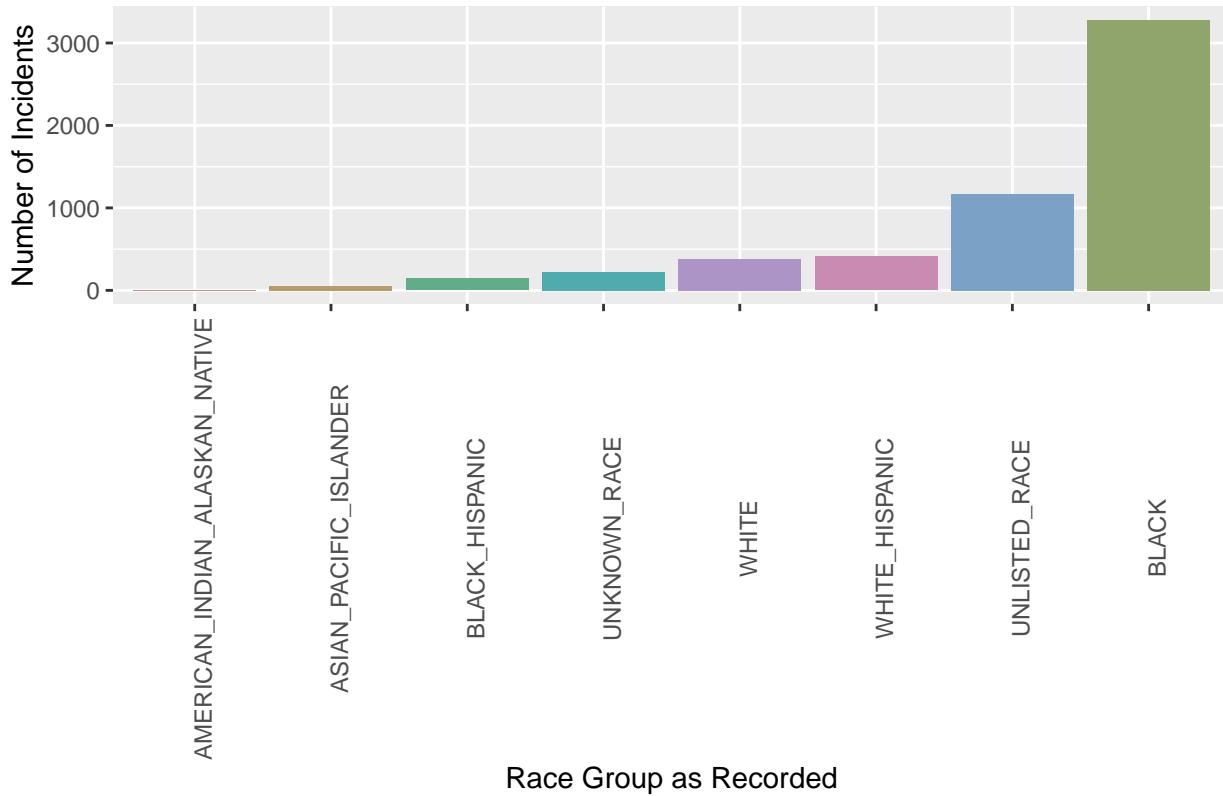


```
#creates bar plot of incidents per race group for the borough listed

boro <- "STATEN ISLAND"

Perp_Races %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group, Incidents)), fill = as.factor(Race_Group), y = Incidents),
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Number of Incidents") +
  xlab("Race Group as Recorded") +
  labs(title = str_c("Number of Perpetrators per Race Group in ", boro, ", New York City"))
```

Number of Perpetrators per Race Group in STATEN ISLAND, New York Ci

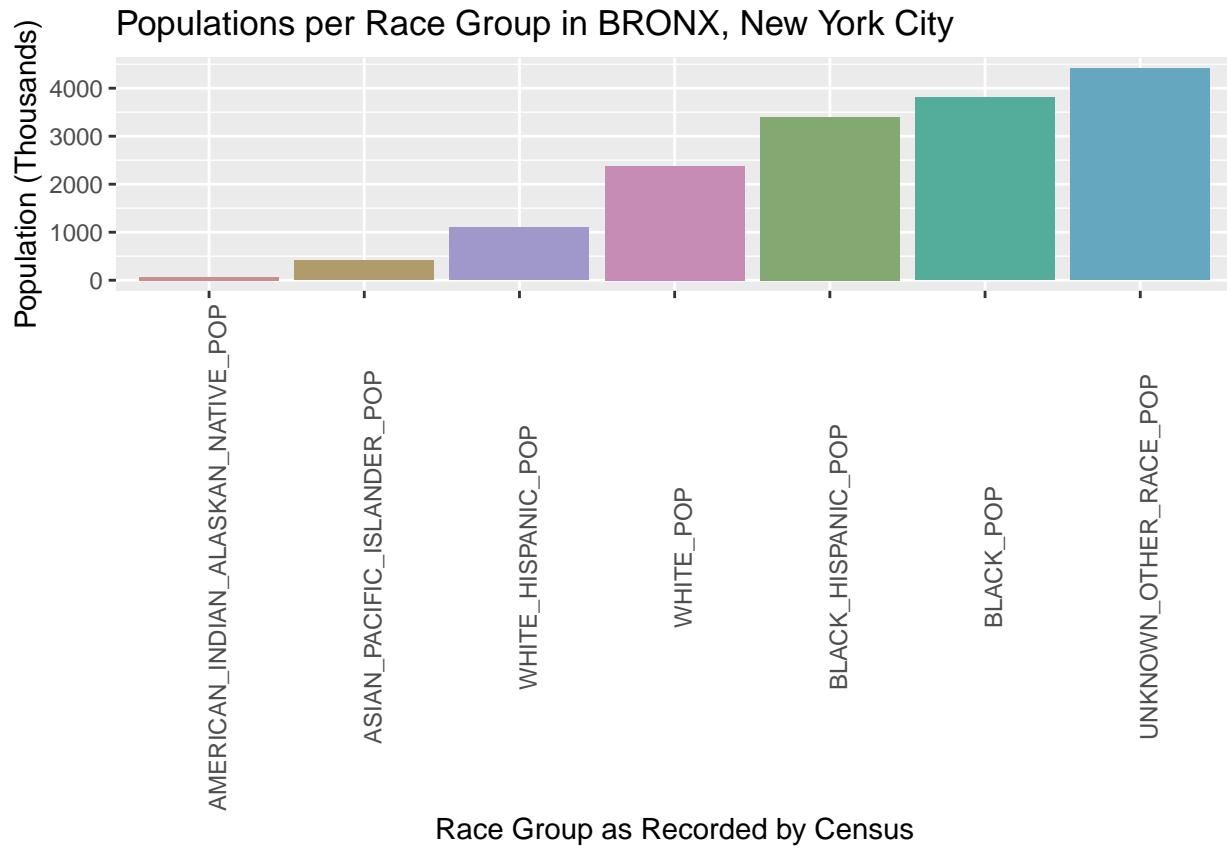


```
#creates bar plot of incidents per race group for the borough listed
```

Across all five boroughs, the groups with the highest number of perpetrators are Black and unlisted race. Next we will look at the boroughs' populations grouped by race in order to see if this is predicted by the population composition and later examine whether percent of the population accurately predicts percent of victims/shooters.

```
temp <- Vic_Races %>%
  mutate(Population = Population / 1000)

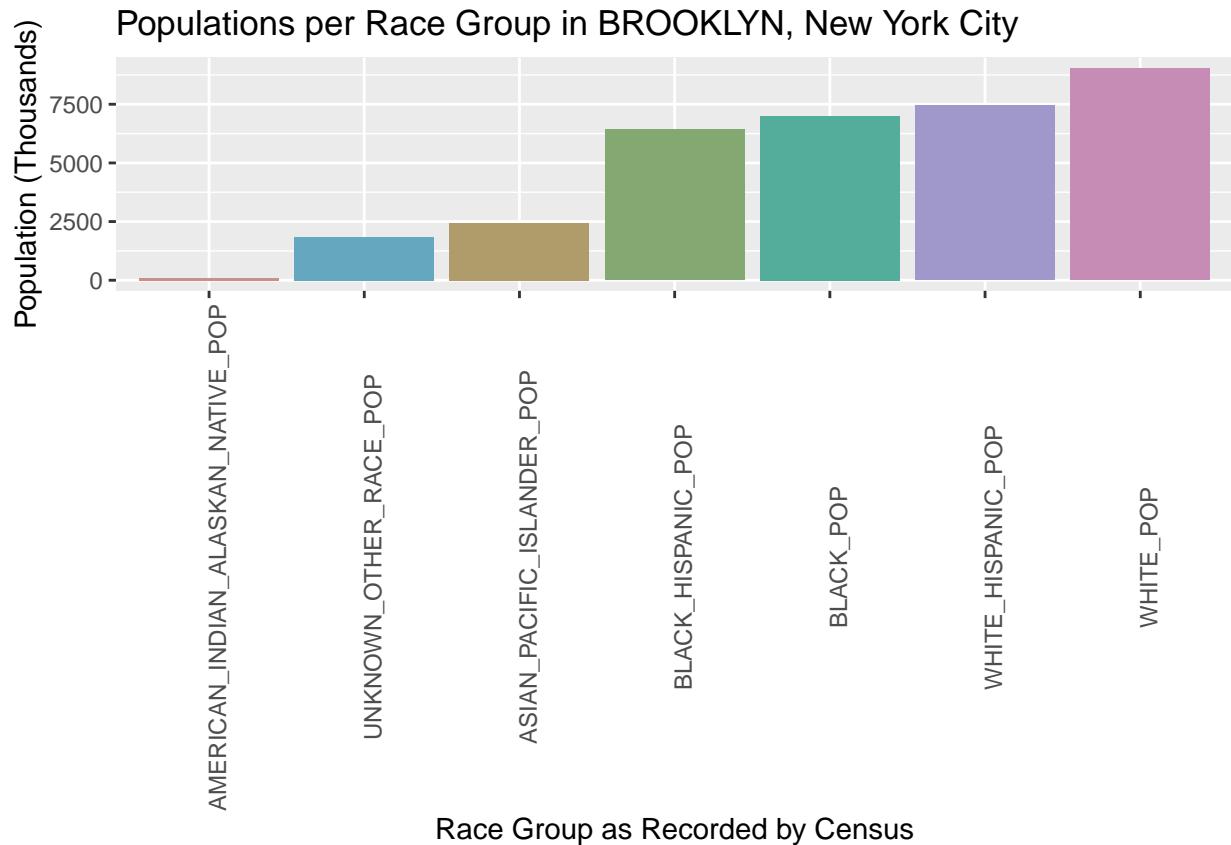
boro <- "BRONX"
temp %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group_Pop, Population)), fill = as.factor(Race_Group_Pop), y = 1),
  theme(legend.position="none",
  axis.text.x = element_text(angle = 90)) +
  ylab("Population (Thousands)") +
  xlab("Race Group as Recorded by Census") +
  labs(title = str_c("Populations per Race Group in ", boro, ", New York City"))
```



```

boro <- "BROOKLYN"

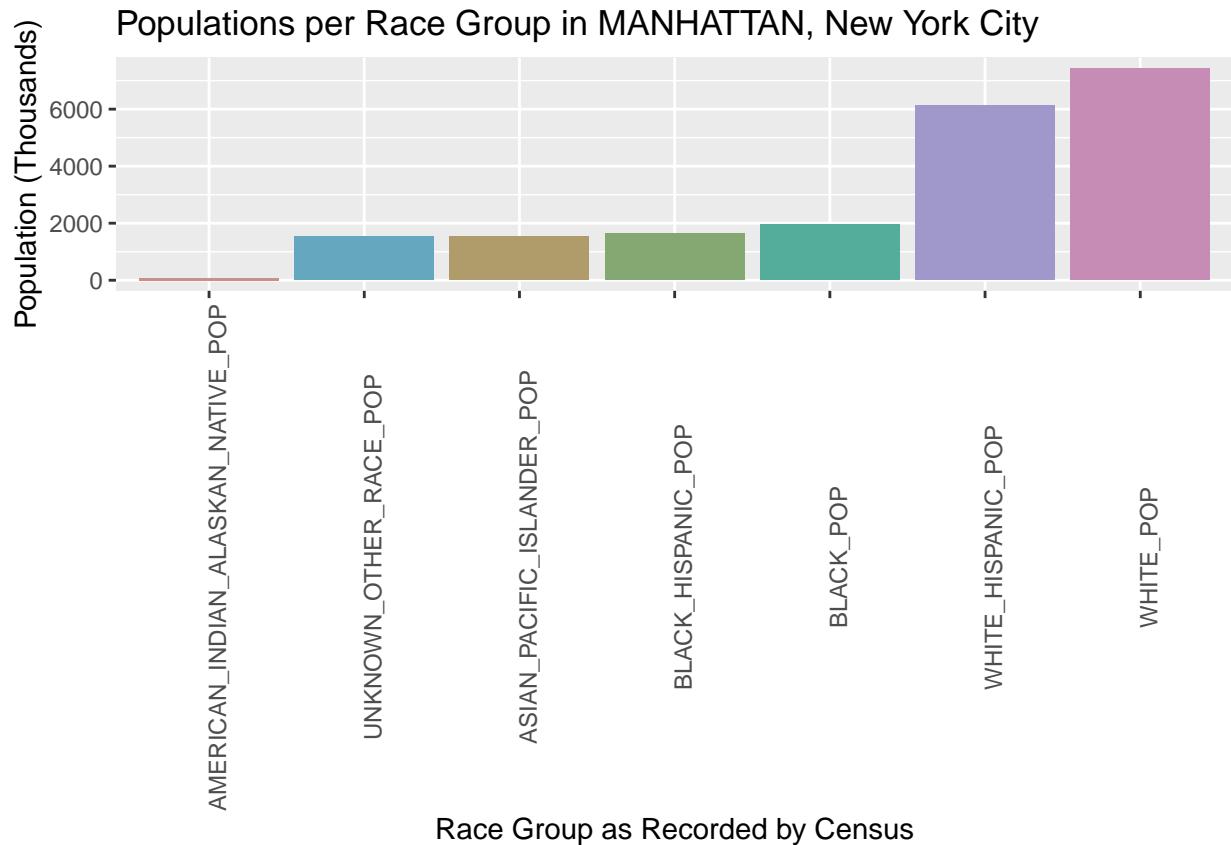
temp %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group_Pop, Population)), fill = as.factor(Race_Group_Pop), y =
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population (Thousands)") +
  xlab("Race Group as Recorded by Census") +
  labs(title = str_c("Populations per Race Group in ", boro, ", New York City"))
  
```



```

boro <- "MANHATTAN"

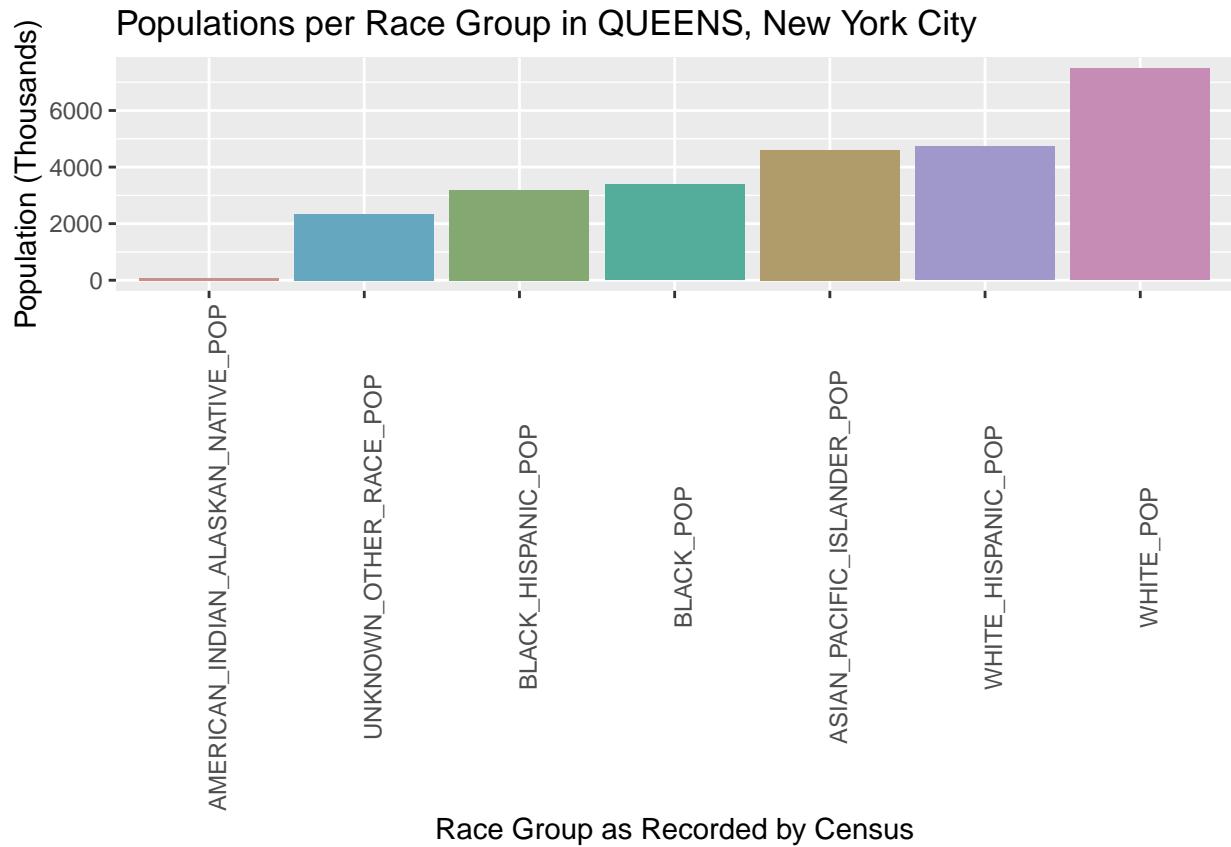
temp %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group_Pop, Population)), fill = as.factor(Race_Group_Pop), y =
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population (Thousands)") +
  xlab("Race Group as Recorded by Census") +
  labs(title = str_c("Populations per Race Group in ", boro, ", New York City"))
  
```



```

boro <- "QUEENS"

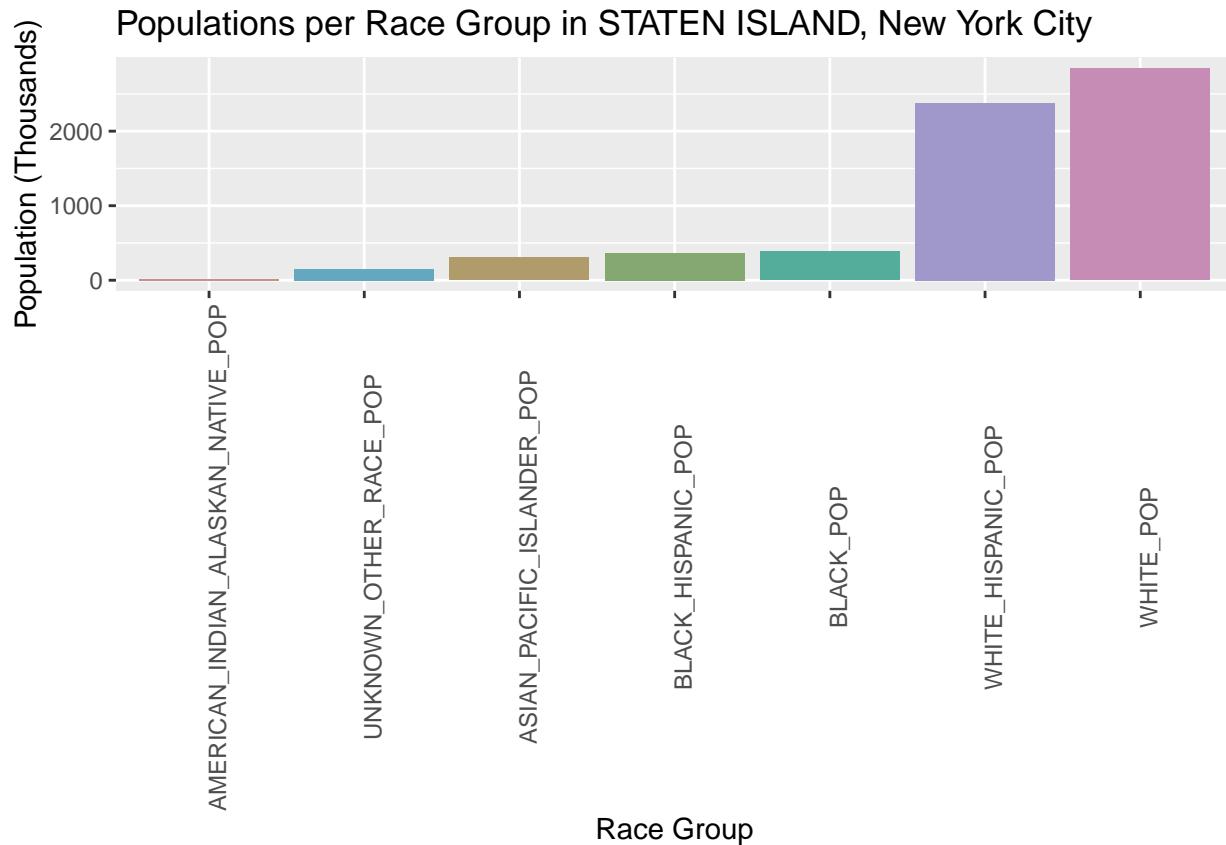
temp %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group_Pop, Population)), fill = as.factor(Race_Group_Pop), y =
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population (Thousands)") +
  xlab("Race Group as Recorded by Census") +
  labs(title = str_c("Populations per Race Group in ", boro, ", New York City"))
  
```



```

boro <- "STATEN ISLAND"

temp %>%
  filter(BORO == boro) %>%
  ggplot(aes(x = as.factor(reorder(Race_Group_Pop, Population)), fill = as.factor(Race_Group_Pop), y =
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Population (Thousands)") +
  xlab("Race Group") +
  labs(title = str_c("Populations per Race Group in ", boro, ", New York City"))
  
```



Black and Hispanic Black & white groups are not the highest in each borough, despite being the highest number of victims and perpetrators, so we will instead look at shooting incident number as we did with race instead of diving deeper into population. We will examine a linear regression to see if there is a linear relationship between percent of racial groups and shooting incidents per borough.

```
#Isolate Black data
filter1 <- "BLACK"

temp <- Vic_Race_Percent %>%
  filter(Race_Group_Pop_Percent == str_c(filter1, "_POP_Percent")) %>%
  filter(Race_Group_Percent == str_c(filter1, "_Percent"))

temp2 <- Vic_Races %>%
  filter(Race_Group_Pop == str_c(filter1, "_POP")) %>%
  group_by(BORO) %>%
  summarize(Incidents = sum(Incidents))

temp <- temp %>%
  left_join(temp2, by = c("BORO")) %>%
  select(-c(Incidents_Percent, Race_Group_Percent))

mod <- lm(Incidents ~ Population_Percent, temp)

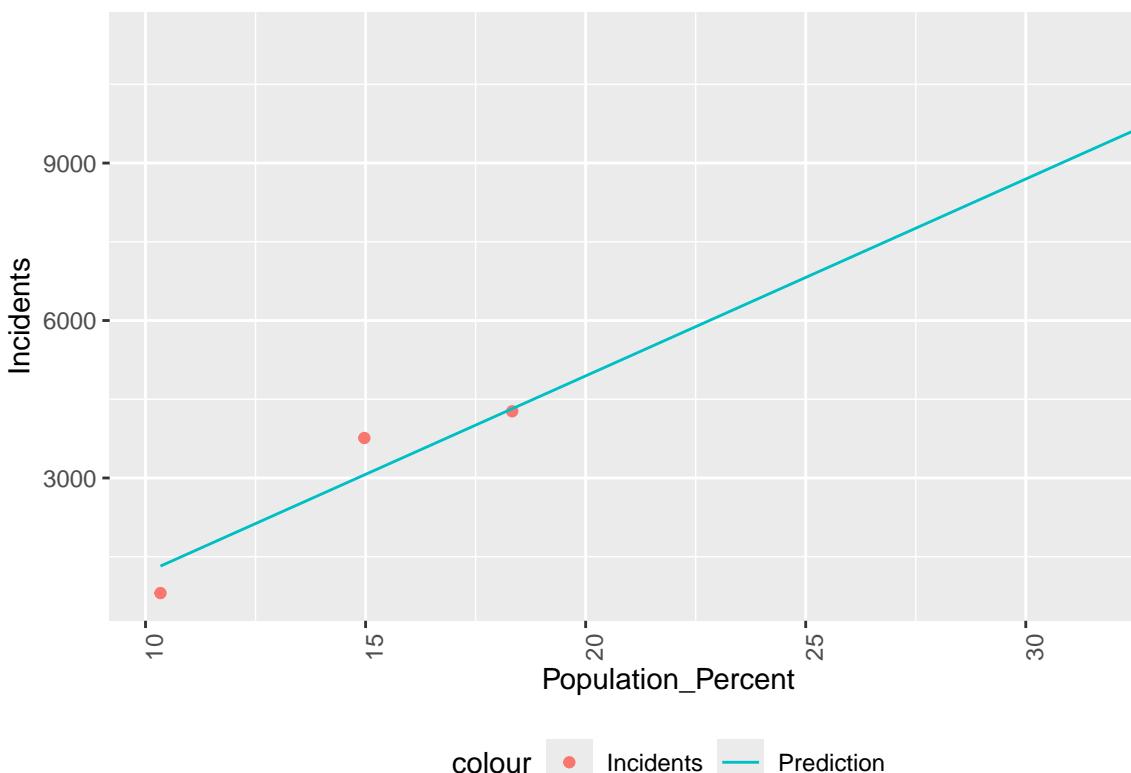
temp <- temp %>% mutate(pred = predict(mod))
```

```

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus Black Percent of Population in Five Boroughs")

```

Incidents versus Black Percent of Population in Five Boroughs



Modeling Race Data

```

BLACK_mod <- mod

filter1 <- "BLACK_HISPANIC"

temp <- Vic_Race_Percent %>%
  filter(Race_Group_Pop_Percent == str_c(filter1, "_POP_Percent")) %>%
  filter(Race_Group_Percent == str_c(filter1, "_Percent"))

temp2 <- Vic_Races %>%
  filter(Race_Group_Pop == str_c(filter1, "_POP")) %>%
  group_by(BORO) %>%
  summarize(Incidents = sum(Incidents))

temp <- temp %>%
  left_join(temp2, by = c("BORO")) %>%
  select(-c(Incidents_Percent, Race_Group_Percent))

mod <- lm(Incidents ~ Population_Percent, temp)

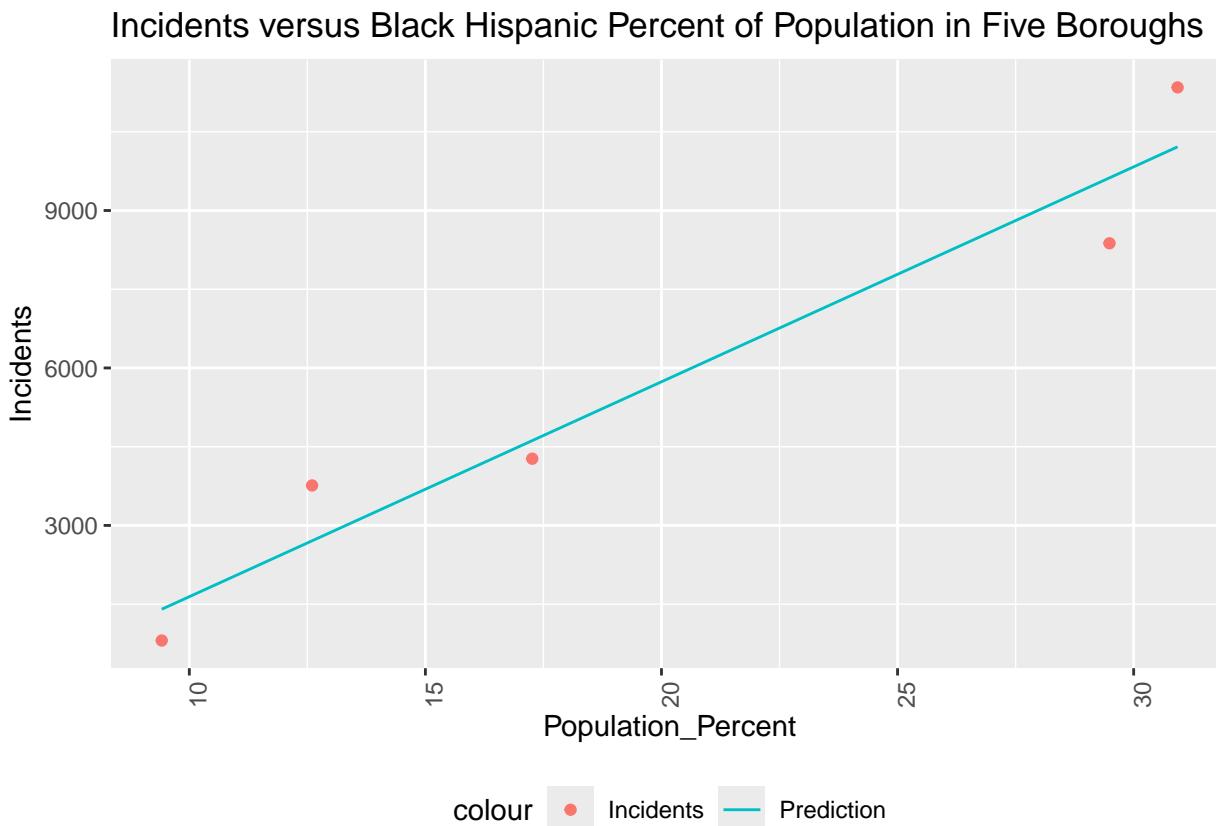
```

```

temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus Black Hispanic Percent of Population in Five Boroughs")

```



```

BLACK_HISPANIC_mod <- mod

filter1 <- "WHITE"

temp <- Vic_Race_Percent %>%
  filter(Race_Group_Pop_Percent == str_c(filter1, "_POP_Percent")) %>%
  filter(Race_Group_Percent == str_c(filter1, "_Percent"))

temp2 <- Vic_Races %>%
  filter(Race_Group_Pop == str_c(filter1, "_POP")) %>%
  group_by(BORO) %>%
  summarize(Incidents = sum(Incidents))

temp <- temp %>%
  left_join(temp2, by = c("BORO")) %>%
  select(-c(Incidents_Percent, Race_Group_Percent))

```

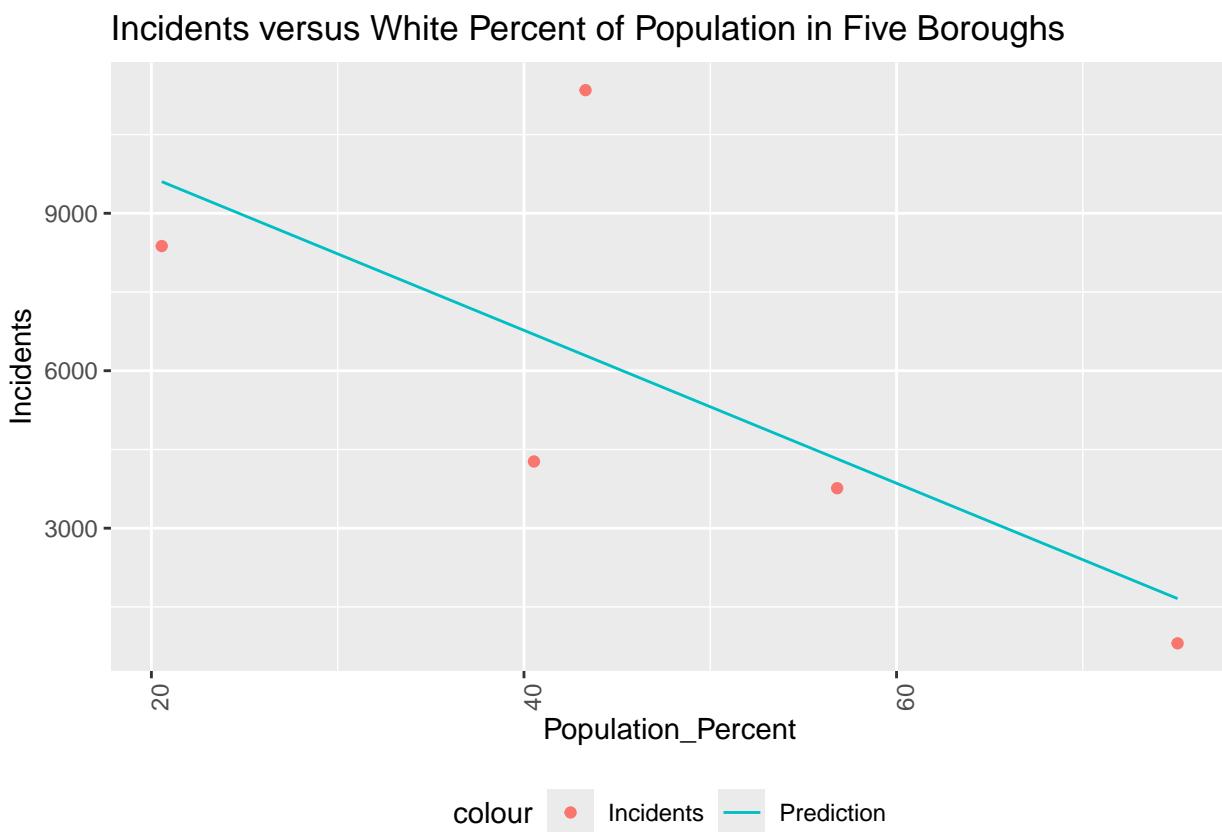
```

mod <- lm(Incidents ~ Population_Percent, temp)

temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus White Percent of Population in Five Boroughs")

```



```

WHITE_mod <- mod

filter1 <- "WHITE_HISPANIC"

temp <- Vic_Race_Percent %>%
  filter(Race_Group_Pop_Percent == str_c(filter1, "_POP_Percent")) %>%
  filter(Race_Group_Percent == str_c(filter1, "_Percent"))

temp2 <- Vic_Races %>%
  filter(Race_Group_Pop == str_c(filter1, "_POP")) %>%
  group_by(BORO) %>%
  summarize(Incidents = sum(Incidents))

temp <- temp %>%

```

```

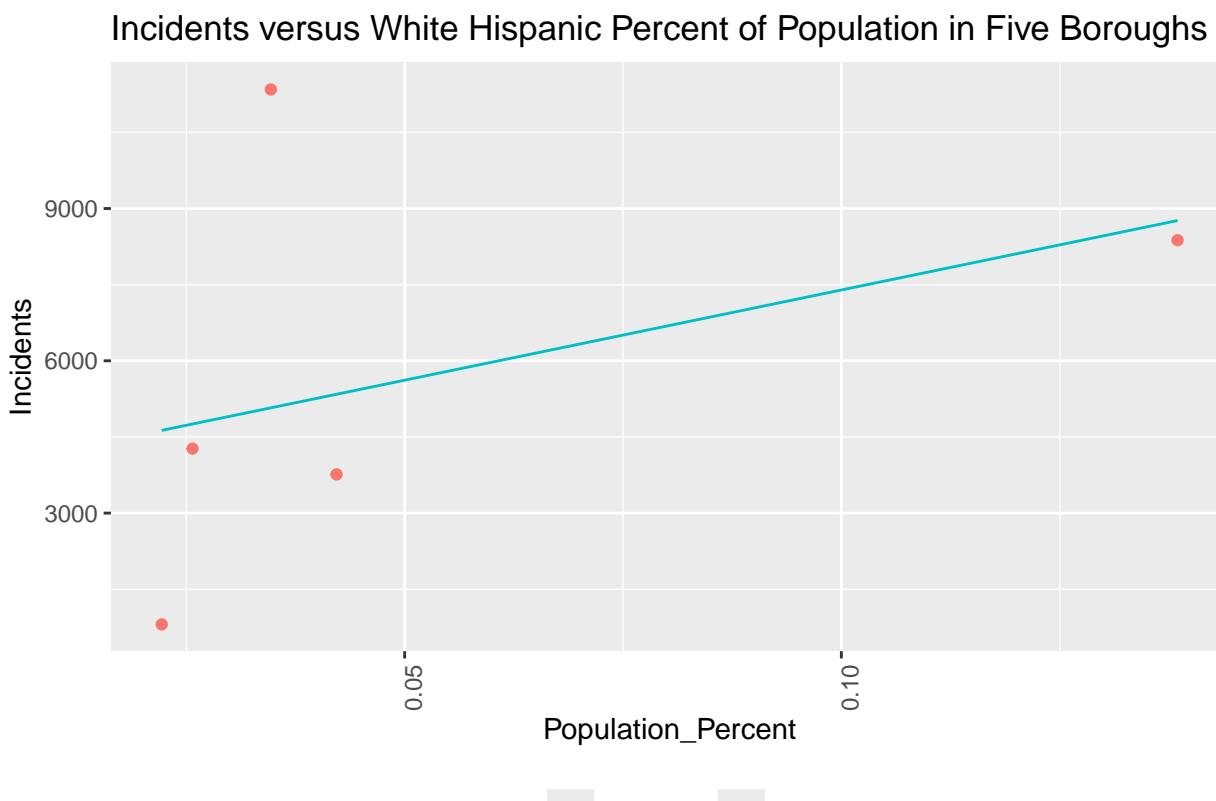
left_join(temp2, by = c("BORO")) %>%
  select(-c(Incidents_Percent, Race_Group_Percent))

mod <- lm(Incidents ~ Population_Percent, temp)

temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus White Hispanic Percent of Population in Five Boroughs")

```



```

WHITE_HISPANIC_mod <- mod

filter1 <- "AMERICAN_INDIAN_ALASKAN_NATIVE"

temp <- Vic_Race_Percent %>%
  filter(Race_Group_Pop_Percent == str_c(filter1, "_POP_Percent")) %>%
  filter(Race_Group_Percent == str_c(filter1, "_Percent"))

temp2 <- Vic_Races %>%
  filter(Race_Group_Pop == str_c(filter1, "_POP")) %>%
  group_by(BORO) %>%

```

```

summarize(Incidents = sum(Incidents))

temp <- temp %>%
  left_join(temp2, by = c("BORO")) %>%
  select(-c(Incidents_Percent, Race_Group_Percent))

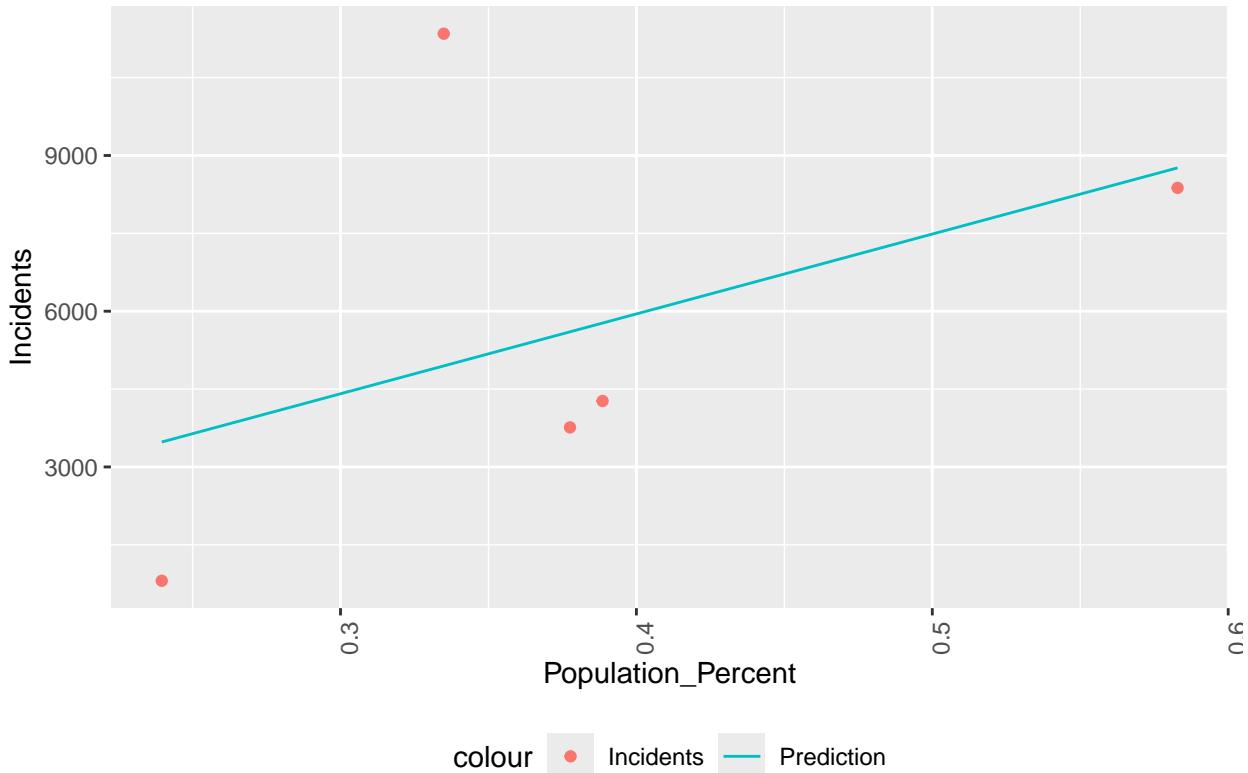
mod <- lm(Incidents ~ Population_Percent, temp)

temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus Native American or Alaskan Native Percent of Population in Five Boroughs")

```

Incidents versus Native American or Alaskan Native Percent of Population



```

AMERICAN_INDIAN_ALASKAN_NATIVE_mod <- mod

filter1 <- "ASIAN_PACIFIC_ISLANDER"

temp <- Vic_Race_Percent %>%
  filter(Race_Group_Pop_Percent == str_c(filter1, "_POP_Percent")) %>%
  filter(Race_Group_Percent == str_c(filter1, "_Percent"))

```

```

temp2 <- Vic_Races %>%
  filter(Race_Group_Pop == str_c(filter1, "_POP")) %>%
  group_by(BORO) %>%
  summarize(Incidents = sum(Incidents))

temp <- temp %>%
  left_join(temp2, by = c("BORO")) %>%
  select(-c(Incidents_Percent, Race_Group_Percent))

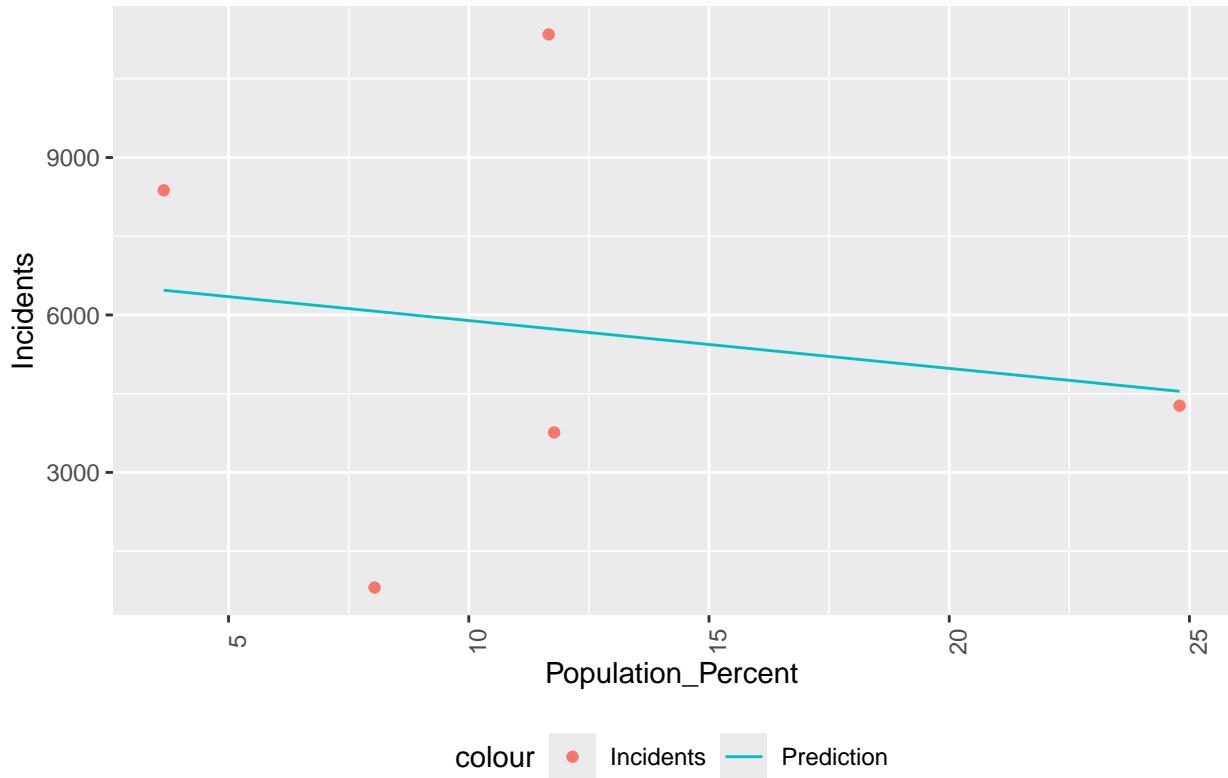
mod <- lm(Incidents ~ Population_Percent, temp)

temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Population_Percent, y = Incidents)) +
  geom_point(aes(color = "Incidents")) +
  geom_line(aes(x = Population_Percent, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Incidents versus Asian or Pacific Islander Percent of Population in Five Boroughs")

```

Incidents versus Asian or Pacific Islander Percent of Population in Five Bo



```
ASIAN_PACIFIC_ISLANDER_mod <- mod
```

These graphs show a trend towards recorded shootings increasing for most races except white and Asian/Pacific Islander. Now we will look at the coefficients and R-values to see how well this regression fits the data.

```

print("For Black population percent composition versus shooting incidents, our coefficients and R-value")

## [1] "For Black population percent composition versus shooting incidents, our coefficients and R-value
summary(BLACK_mod)

##
## Call:
## lm(formula = Incidents ~ Population_Percent, data = temp)
##
## Residuals:
##      1       2       3       4       5 
## -1507.86 1366.69  703.26 -47.91 -514.18 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2557.6    1441.1  -1.775  0.17403    
## Population_Percent 375.1      60.0   6.252  0.00826 ** 
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 1278 on 3 degrees of freedom
## Multiple R-squared:  0.9287, Adjusted R-squared:  0.905 
## F-statistic: 39.08 on 1 and 3 DF,  p-value: 0.008257 

print("For Black Hispanic population percent composition versus shooting incidents, our coefficients and R-value")

## [1] "For Black Hispanic population percent composition versus shooting incidents, our coefficients and R-value
summary(BLACK_HISPANIC_mod)

##
## Call:
## lm(formula = Incidents ~ Population_Percent, data = temp)
##
## Residuals:
##      1       2       3       4       5 
## -1246.6 1132.8 1055.2 -345.4 -596.1 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2452.9    1350.3  -1.817  0.16689    
## Population_Percent 409.5      62.0   6.605  0.00707 ** 
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 1215 on 3 degrees of freedom
## Multiple R-squared:  0.9357, Adjusted R-squared:  0.9142 
## F-statistic: 43.62 on 1 and 3 DF,  p-value: 0.007066 

print("For white Hispanic population percent composition versus shooting incidents, our coefficients and R-value")

## [1] "For white Hispanic population percent composition versus shooting incidents, our coefficients and R-value
summary(WHITE_HISPANIC_mod)

##

```

```

## Call:
## lm(formula = Incidents ~ Population_Percent, data = temp)
##
## Residuals:
##      1       2       3       4       5
## -387.3  6272.5 -1578.9 -483.9 -3822.4
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3841        3057    1.257   0.298
## Population_Percent    35539        44758    0.794   0.485
##
## Residual standard error: 4352 on 3 degrees of freedom
## Multiple R-squared:  0.1737, Adjusted R-squared:  -0.1018
## F-statistic: 0.6305 on 1 and 3 DF, p-value: 0.4852
print("For white population percent composition versus shooting incidents, our coefficients and R-value")

## [1] "For white population percent composition versus shooting incidents, our coefficients and R-value"
summary(WHITE_mod)

##
## Call:
## lm(formula = Incidents ~ Population_Percent, data = temp)
##
## Residuals:
##      1       2       3       4       5
## -1226.6  5057.6 -558.7 -2420.8 -851.5
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           12597.27    4206.67   2.995   0.0579 .
## Population_Percent -145.68      83.12   -1.753   0.1779
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3366 on 3 degrees of freedom
## Multiple R-squared:  0.5059, Adjusted R-squared:  0.3412
## F-statistic: 3.072 on 1 and 3 DF, p-value: 0.1779
print("For Asian or Pacific Islander population percent composition versus shooting incidents, our coefficients and R-value")

## [1] "For Asian or Pacific Islander population percent composition versus shooting incidents, our coefficients and R-value"
summary(ASIAN_PACIFIC_ISLANDER_mod)

##
## Call:
## lm(formula = Incidents ~ Population_Percent, data = temp)
##
## Residuals:
##      1       2       3       4       5
## 1904.0  5604.2 -1969.3 -273.9 -5265.0
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
```

```

## (Intercept)      6804.75    4154.73   1.638    0.20
## Population_Percent -91.14     298.68   -0.305    0.78
##
## Residual standard error: 4715 on 3 degrees of freedom
## Multiple R-squared:  0.0301, Adjusted R-squared:  -0.2932
## F-statistic: 0.09311 on 1 and 3 DF,  p-value: 0.7802
print("For Native American or Alaskan Native population percent composition versus shooting incidents, ")
## [1] "For Native American or Alaskan Native population percent composition versus shooting incidents,
summary(AMERICAN_INDIAN_ALASKAN_NATIVE_mod)

##
## Call:
## lm(formula = Incidents ~ Population_Percent, data = temp)
##
## Residuals:
##       1       2       3       4       5
## -385.6  6399.9 -1840.5 -1501.0 -2672.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -205.9     6772.5   -0.03   0.978
## Population_Percent 15384.4    16901.1    0.91   0.430
##
## Residual standard error: 4238 on 3 degrees of freedom
## Multiple R-squared:  0.2164, Adjusted R-squared:  -0.04478
## F-statistic: 0.8286 on 1 and 3 DF,  p-value: 0.4298

```

In this data, we can see a statistically significant R-Value for Black and Black Hispanic populations' linear regressions, but not for any of the other race groups, with white groups being just on the edge of matching a linear regression.

Now that we've looked at all of our demographic data, we will look at our time and date data.

Transform Date and Time Data

Next, we will look at times and dates that shooting occurred and create a counter column to count incidences before grouping by date and time, respectively.

```

Shooting_Times <- Shooting_Victims %>%
  select(c(OCCUR_DATE, OCCUR_TIME, BORO, STATISTICAL_MURDER_FLAG))
#Isolates the shooting dates times, and murders per borough from demographic information in a new datafarm

Shooting_Times$COUNTER <- 1
#Creates a column with 1 as each entry to count each entry as one shooting incident, as listed by the person

Shooting_By_Time <- Shooting_Times %>%
  group_by(OCCUR_TIME, BORO) %>%
  summarize(STATISTICAL_MURDER_FLAG = sum(STATISTICAL_MURDER_FLAG),
            COUNTER = sum(COUNTER)) %>%
  ungroup()

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

```

```

#Groups shootings by time of the incident throughout a day

Murder_By_Time <- Shooting_By_Time %>%
  select(-c(COUNTER)) %>%
  pivot_wider(
    names_from = BORO,
    values_from = STATISTICAL_MURDER_FLAG
  )
#Sorts gun murders by borough per time of the incident

Shooting_By_Time_Totals <- Shooting_By_Time %>%
  select(-c(STATISTICAL_MURDER_FLAG)) %>%
  pivot_wider(
    names_from = BORO,
    values_from = COUNTER
  )
#Sorts shootings by borough per time of the incident

colnames(Shooting_By_Time_Totals) <- c("OCCUR_TIME", "BRONX", "MANHATTAN", "QUEENS", "BROOKLYN", "STATEN_ISLAND")
#Fixes Staten Island to have an underscore instead of a space

colnames(Murder_By_Time) <- c("OCCUR_TIME", "BRONX", "MANHATTAN", "QUEENS", "BROOKLYN", "STATEN_ISLAND")
#Fixes Staten Island to have an underscore instead of a space

Shooting_By_Time_Totals[is.na(Shooting_By_Time_Totals)] <- 0
Murder_By_Time[is.na(Murder_By_Time)] <- 0
#Replaces n/a values with 0, since if there is no entry for this value, no shootings or murders were recorded

Shooting_By_Time_Totals <- Shooting_By_Time_Totals %>%
  transform(OCCUR_TIME = hms(OCCUR_TIME))

Murder_By_Time <- Murder_By_Time %>%
  transform(OCCUR_TIME = hms(OCCUR_TIME))
#puts time column into hms format for R

Shooting_By_Date <- Shooting_Times %>%
  group_by(OCCUR_DATE, BORO) %>%
  summarize(STATISTICAL_MURDER_FLAG = sum(STATISTICAL_MURDER_FLAG),
            COUNTER = sum(COUNTER))

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

Shooting_By_Date_Totals <- Shooting_By_Date %>%
  select(-c(STATISTICAL_MURDER_FLAG)) %>%
  pivot_wider(
    names_from = BORO,
    values_from = COUNTER
  )
#Sorts shooting incidents by date per borough

Murder_By_Date <- Shooting_By_Date %>%
  select(-c(COUNTER)) %>%
  pivot_wider(

```

```

    names_from = BORO,
    values_from = STATISTICAL_MURDER_FLAG
)
#Sorts murder incidents by date per borough

colnames(Shooting_By_Date_Totals) <- c("OCCUR_DATE", "BRONX", "BROOKLYN", "MANHATTAN", "QUEENS", "STATEN_ISLAND")
colnames(Murder_By_Date) <- c("OCCUR_DATE", "BRONX", "BROOKLYN", "MANHATTAN", "QUEENS", "STATEN_ISLAND")
#renames Staten Island to replace space with underscore for R ease of access

Shooting_By_Date_Totals[is.na(Shooting_By_Date_Totals)] <- 0
Murder_By_Date[is.na(Murder_By_Date)] <- 0
#Replaces na entries with 0 due to lack of recorded entry for that date meaning no shooting incidents were recorded

Shooting_By_Date_Totals <- Shooting_By_Date_Totals %>%
  transform(OCCUR_DATE = mdy(OCCUR_DATE))

Murder_By_Date <- Murder_By_Date %>%
  transform(OCCUR_DATE = mdy(OCCUR_DATE))
#Makes sure the date column is in date format for R

```

Visualize and Model Time and Date Data First we will visualize and model our data over time of day. After a preliminary visualization check, we determined that a polynomial degree of 2 would be best for the linear regression model.

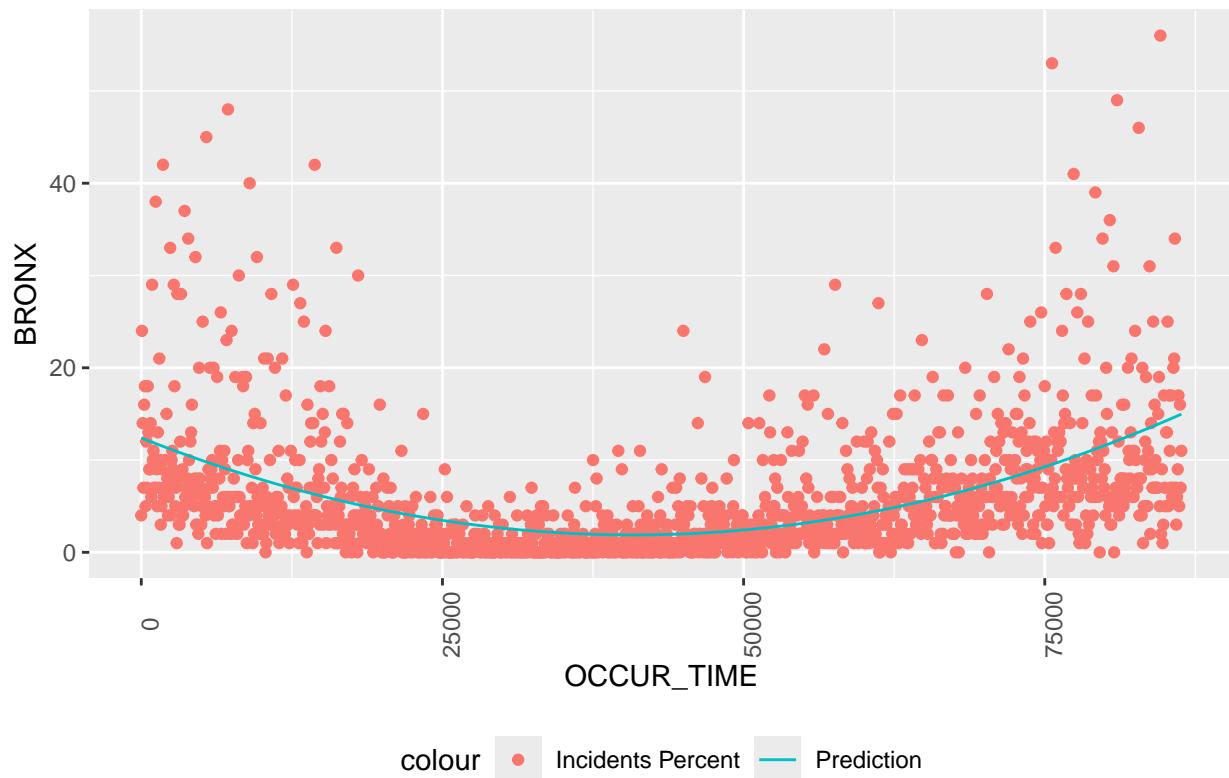
```

temp <- Shooting_By_Time_Totals

mod <- lm(BRONX ~ poly(OCCUR_TIME, degree = 2), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_TIME, y = BRONX)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_TIME, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Time of Day in Bronx, New York City")

```

Shootings by Time of Day in Bronx, New York City

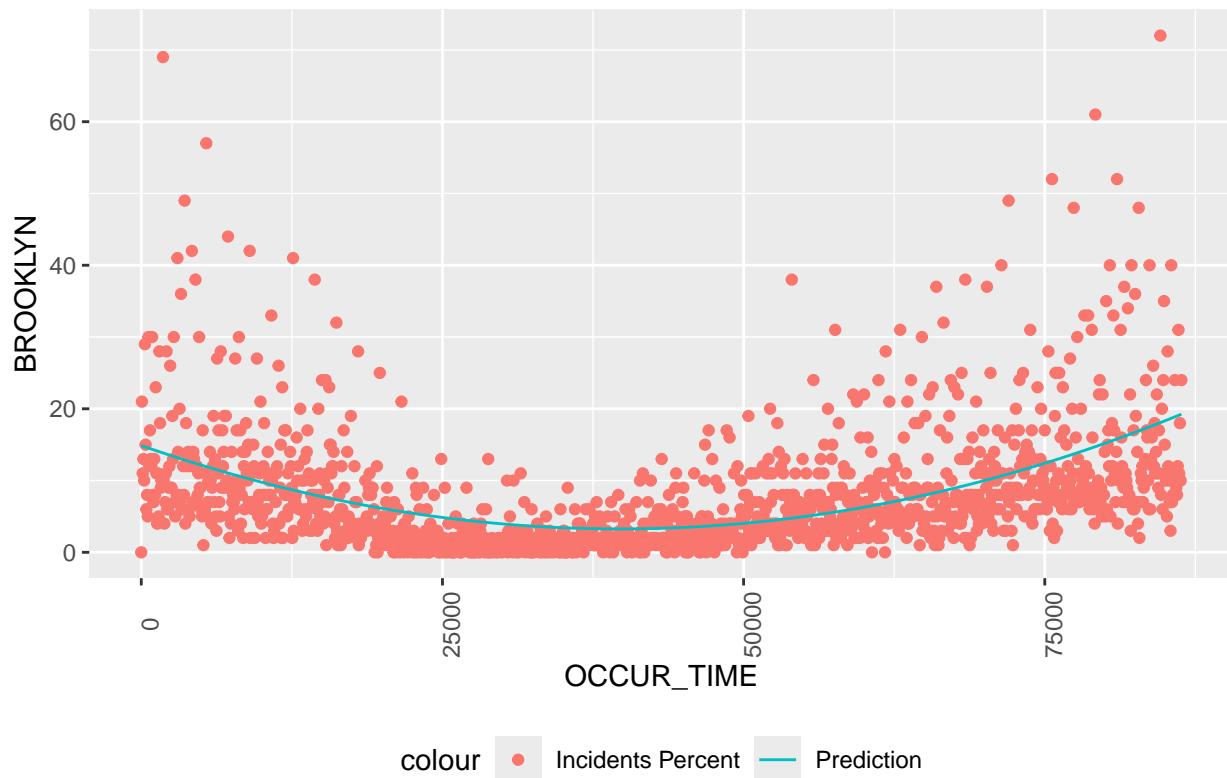


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Time_Totals

mod <- lm(BROOKLYN ~ poly(OCCUR_TIME, degree = 2), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_TIME, y = BROOKLYN)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_TIME, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Time of Day in Brooklyn, New York City")
```

Shootings by Time of Day in Brooklyn, New York City

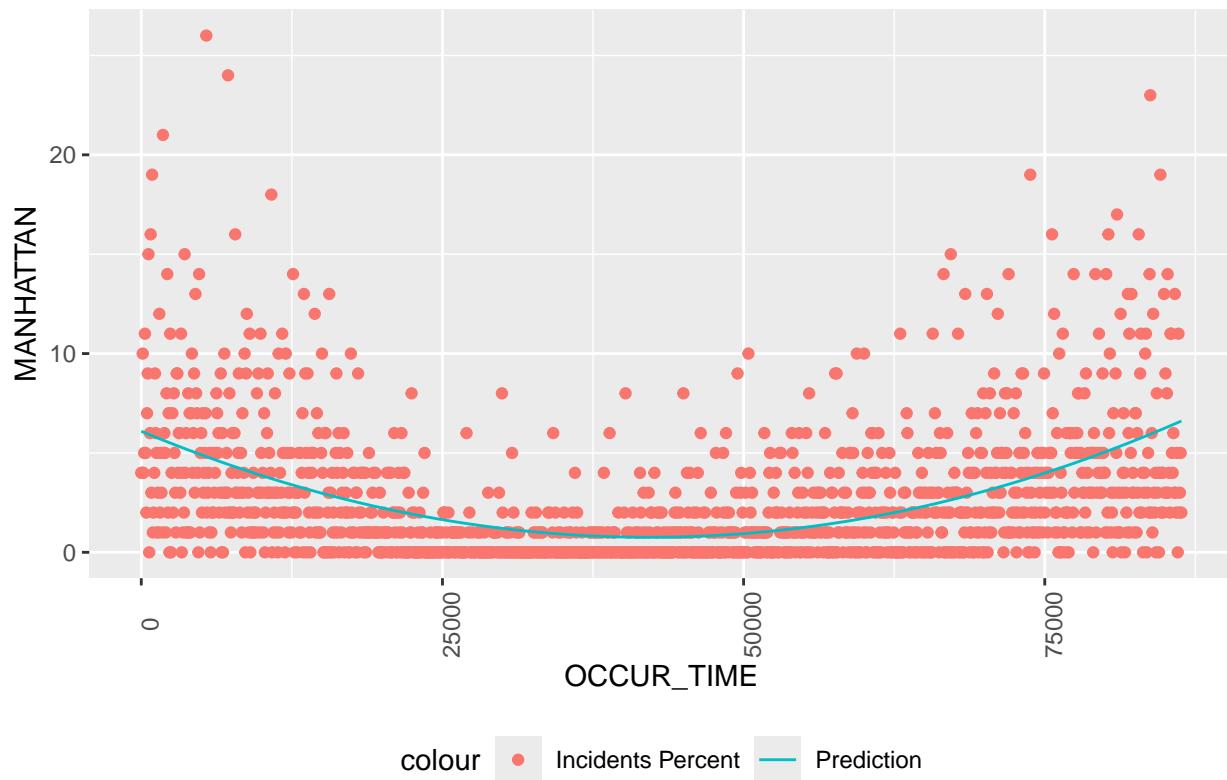


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Time_Totals

mod <- lm(MANHATTAN ~ poly(OCCUR_TIME, degree = 2), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_TIME, y = MANHATTAN)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_TIME, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Time of Day in Manhattan, New York City")
```

Shootings by Time of Day in Manhattan, New York City

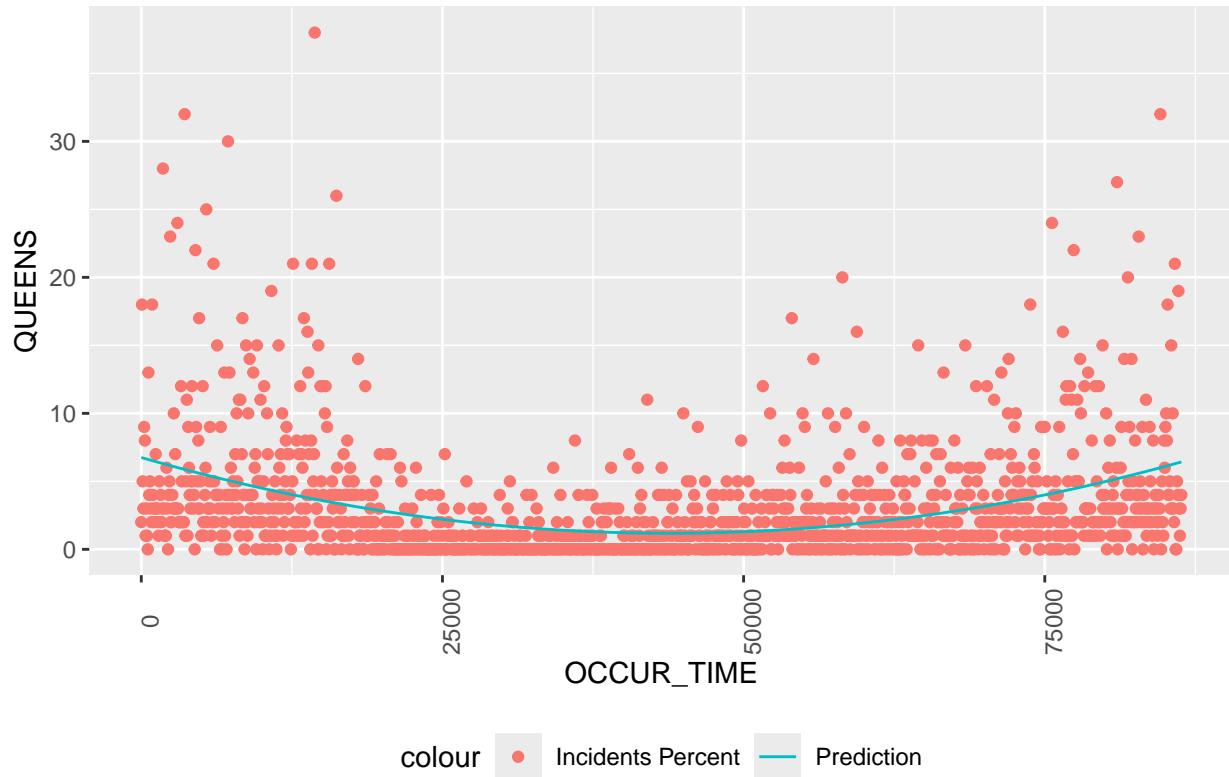


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Time_Totals

mod <- lm(QUEENS ~ poly(OCCUR_TIME, degree = 2), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_TIME, y = QUEENS)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_TIME, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Time of Day in Queens, New York City")
```

Shootings by Time of Day in Queens, New York City

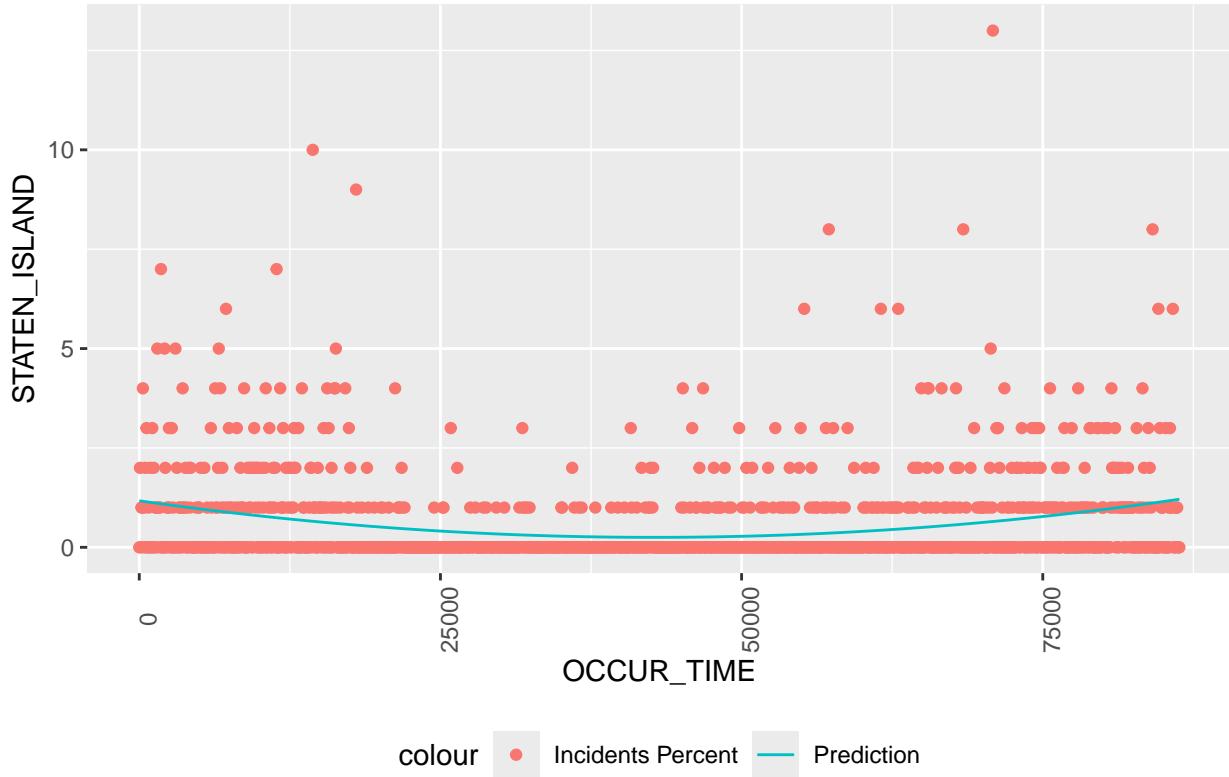


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Time_Totals

mod <- lm(STATEN_ISLAND ~ poly(OCCUR_TIME, degree = 2), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_TIME, y = STATEN_ISLAND)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_TIME, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Time of Day in Staten Island, New York City")
```

Shootings by Time of Day in Staten Island, New York City



```
# Creates a scatterplot graph of shootings over time of day in the borough selected
```

It seems like shootings might increase at night based on our graphs, but the R-values are not showing a good prediction based on our polynomial degree of 2 linear regression. So, let's look with our own eyes at day versus night, basing "day" as 6AM to 6PM and "night" as 6PM to 6AM.

```
tempDay <- Shooting_By_Time_Totals %>%
  filter(OCCUR_TIME > (6*60*60-1)) %>%
  filter(OCCUR_TIME < (18*60*60)) %>%
  summarize(OCCUR_TIME = "Day",
            BRONX = sum(BRONX),
            BROOKLYN = sum(BROOKLYN),
            MANHATTAN = sum(MANHATTAN),
            QUEENS = sum(QUEENS),
            STATEN_ISLAND = sum(STATEN_ISLAND))
#Separates and sums day shootings using (desired hour in military time * 60 * 60 -1) (and desired hour

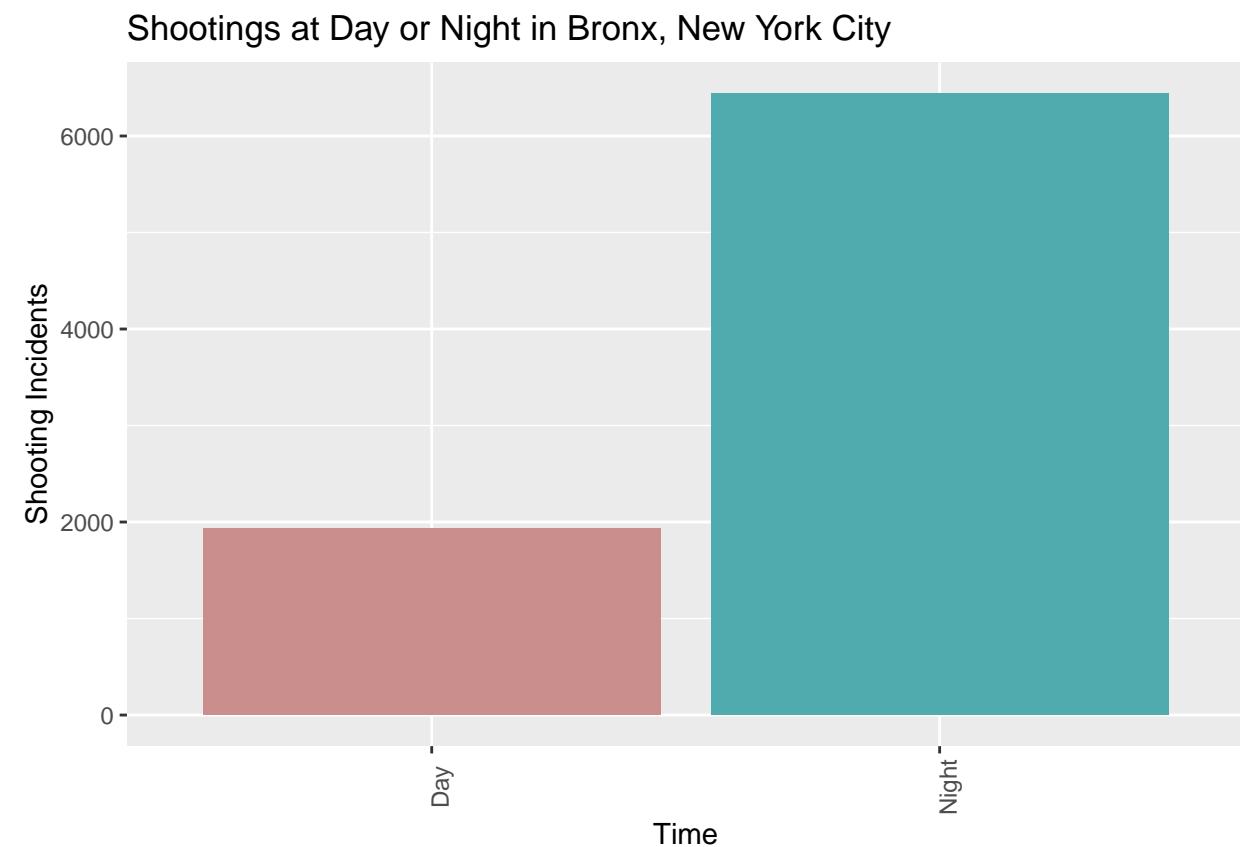
tempNight <- Shooting_By_Time_Totals %>%
  filter((OCCUR_TIME < (6*60*60)) | (OCCUR_TIME > (18*60*60-1))) %>%
  summarize(OCCUR_TIME = "Night",
            BRONX = sum(BRONX),
            BROOKLYN = sum(BROOKLYN),
            MANHATTAN = sum(MANHATTAN),
            QUEENS = sum(QUEENS),
            STATEN_ISLAND = sum(STATEN_ISLAND))
#Separates and sums night shootings, replace 6 and 18 with desired hours in military time if necessary
```

```
temp <- rbind(tempDay, tempNight)
#Adds day and night values into one dataframe with the same column names (vertical join)
```

Now that we've separated our shooting incident data by "day" and "night," we can visualize and easily compare.

```
boro <- "Bronx"

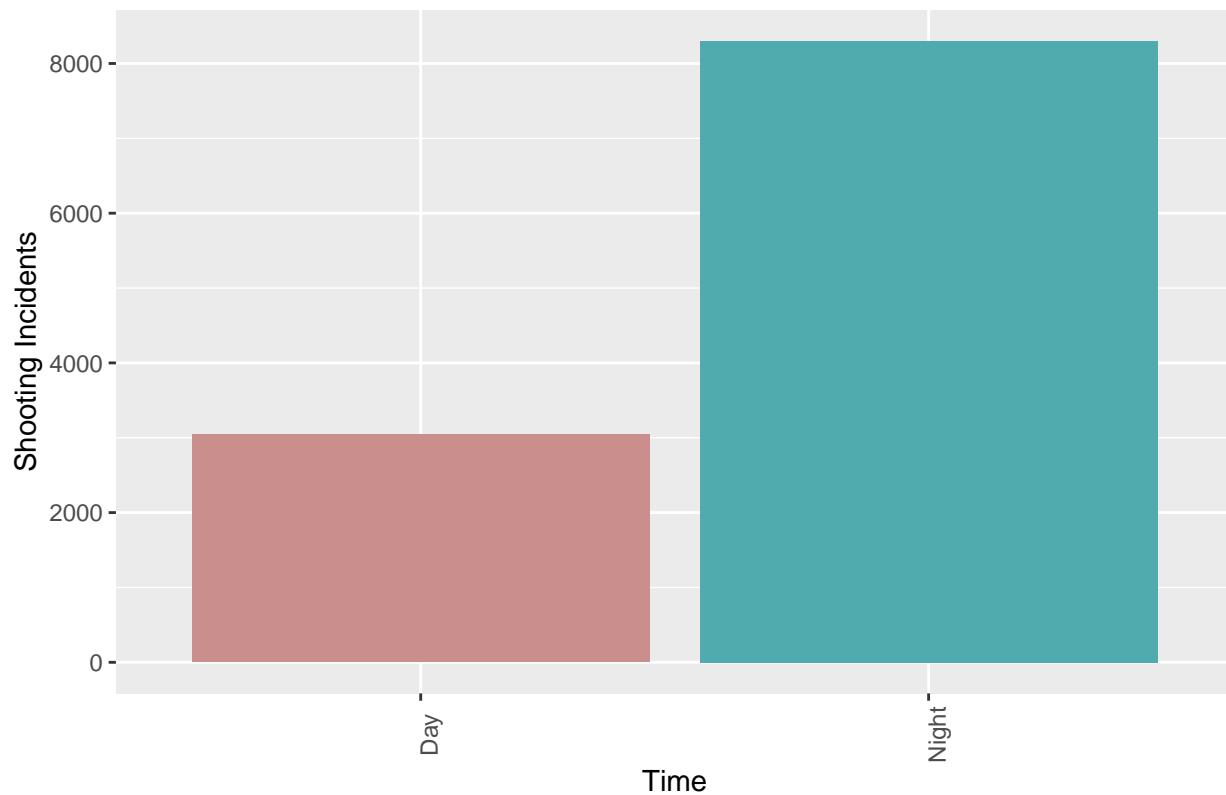
temp %>%
  ggplot(aes(x = OCCUR_TIME, fill = as.factor(OCCUR_TIME), y = BRONX)) +
  geom_bar(stat = "identity") + scale_fill_hue(c = 40) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Shooting Incidents") +
  xlab("Time") +
  labs(title = str_c("Shootings at Day or Night in ", boro, ", New York City"))
```



```
boro <- "Brooklyn"

temp %>%
  ggplot(aes(x = OCCUR_TIME, fill = as.factor(OCCUR_TIME), y = BROOKLYN)) +
  geom_bar(stat = "identity") + scale_fill_hue(c = 40) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Shooting Incidents") +
  xlab("Time") +
  labs(title = str_c("Shootings at Day or Night in ", boro, ", New York City"))
```

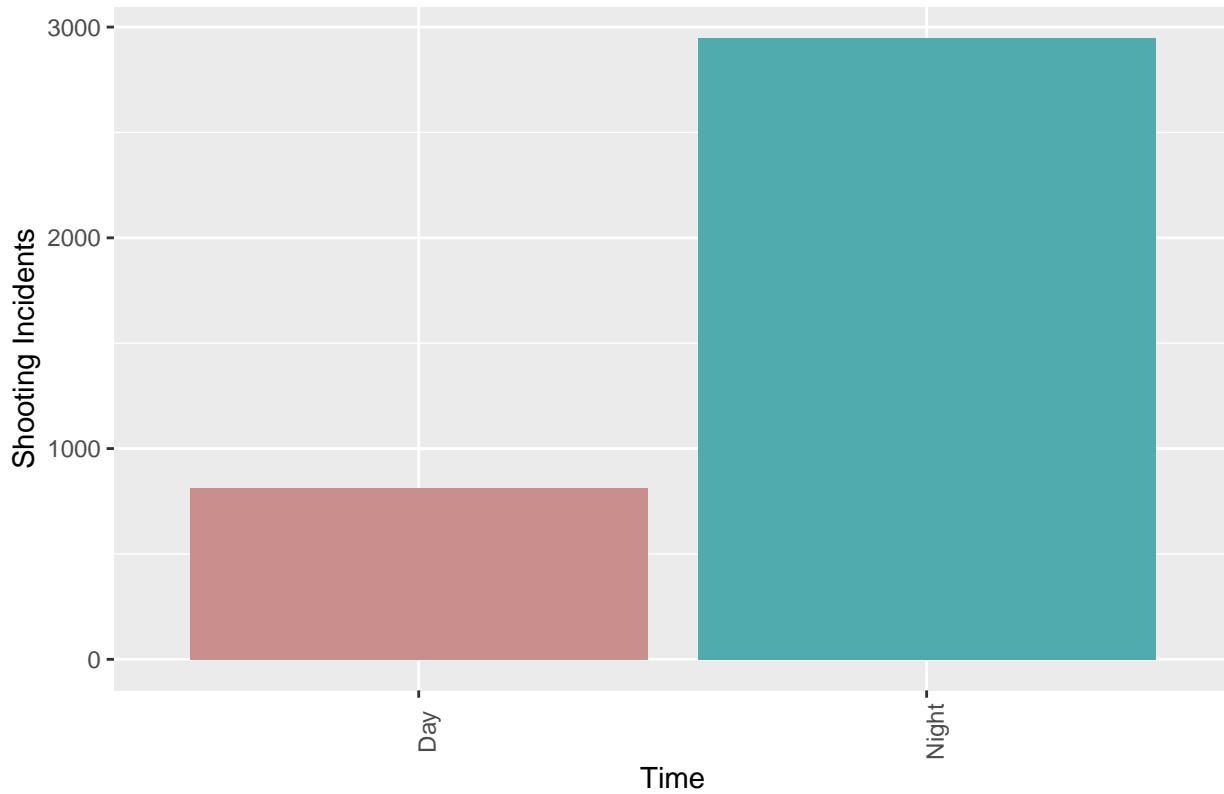
Shootings at Day or Night in Brooklyn, New York City



```
boro <- "Manhattan"

temp %>%
  ggplot(aes(x = OCCUR_TIME, fill = as.factor(OCCUR_TIME), y = MANHATTAN)) +
  geom_bar(stat = "identity") + scale_fill_hue(c = 40) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Shooting Incidents") +
  xlab("Time") +
  labs(title = str_c("Shootings at Day or Night in ", boro, ", New York City"))
```

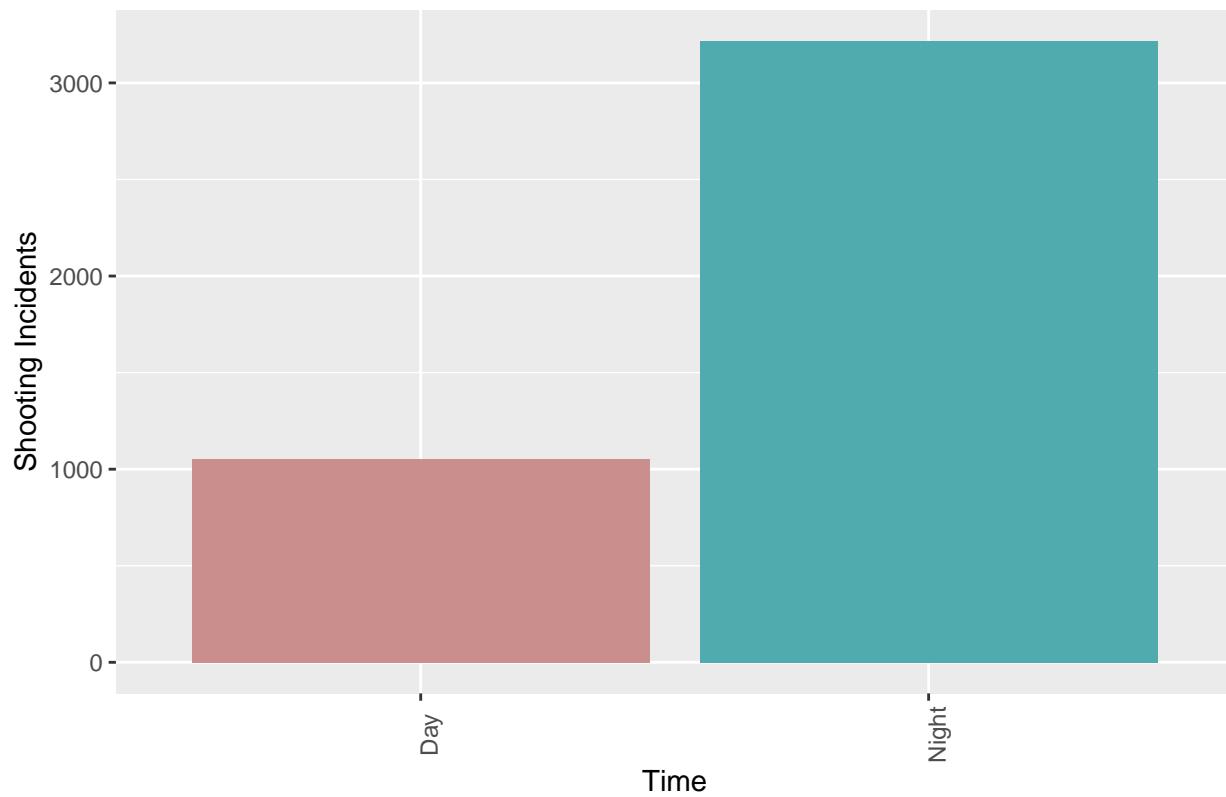
Shootings at Day or Night in Manhattan, New York City



```
boro <- "Queens"

temp %>%
  ggplot(aes(x = OCCUR_TIME, fill = as.factor(OCCUR_TIME), y = QUEENS)) +
  geom_bar(stat = "identity") + scale_fill_hue(c = 40) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Shooting Incidents") +
  xlab("Time") +
  labs(title = str_c("Shootings at Day or Night in ", boro, ", New York City"))
```

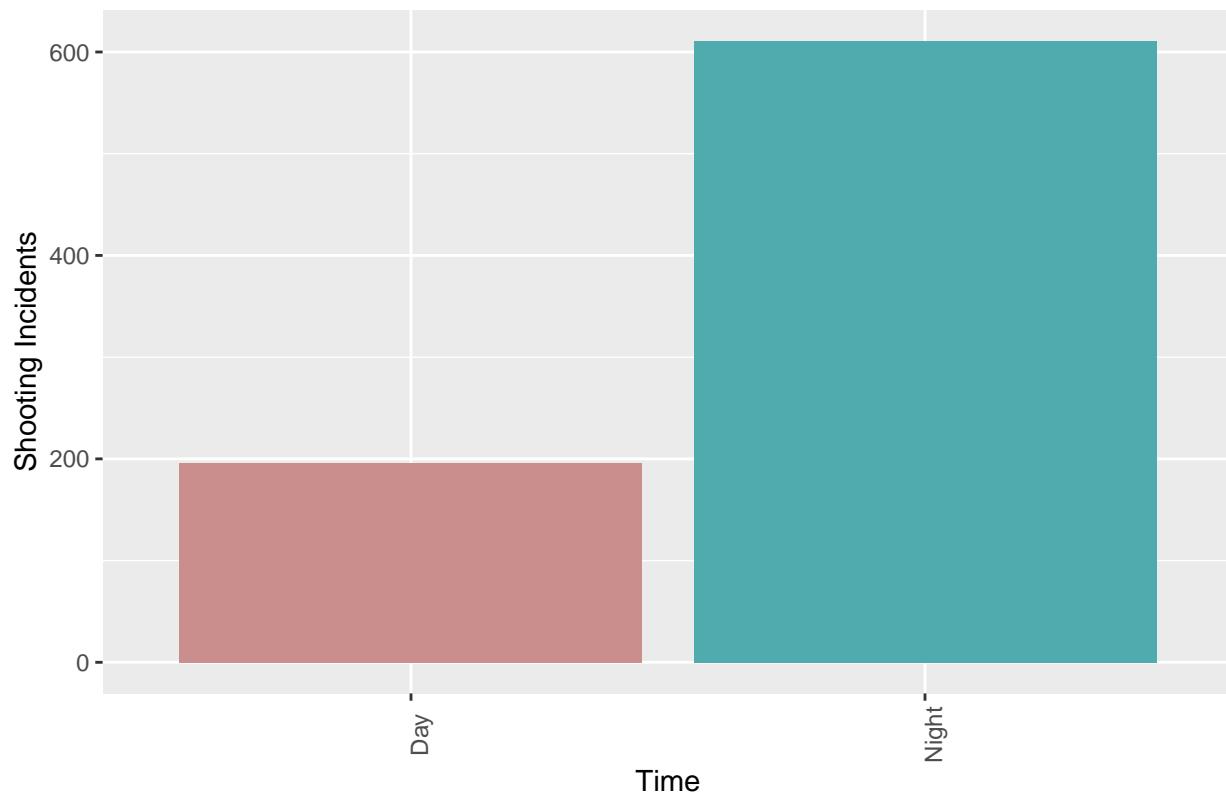
Shootings at Day or Night in Queens, New York City



```
boro <- "Staten Island"

temp %>%
  ggplot(aes(x = OCCUR_TIME, fill = as.factor(OCCUR_TIME), y = STATEN_ISLAND)) +
  geom_bar(stat = "identity") + scale_fill_hue(c = 40) +
  theme(legend.position="none",
        axis.text.x = element_text(angle = 90)) +
  ylab("Shooting Incidents") +
  xlab("Time") +
  labs(title = str_c("Shootings at Day or Night in ", boro, ", New York City"))
```

Shootings at Day or Night in Staten Island, New York City



From this idea of “Day” and “Night” bordered by 6AM and 6PM, there are far more shootings happening at night than during the day across all five boroughs when looking at the totals rather than an average day.

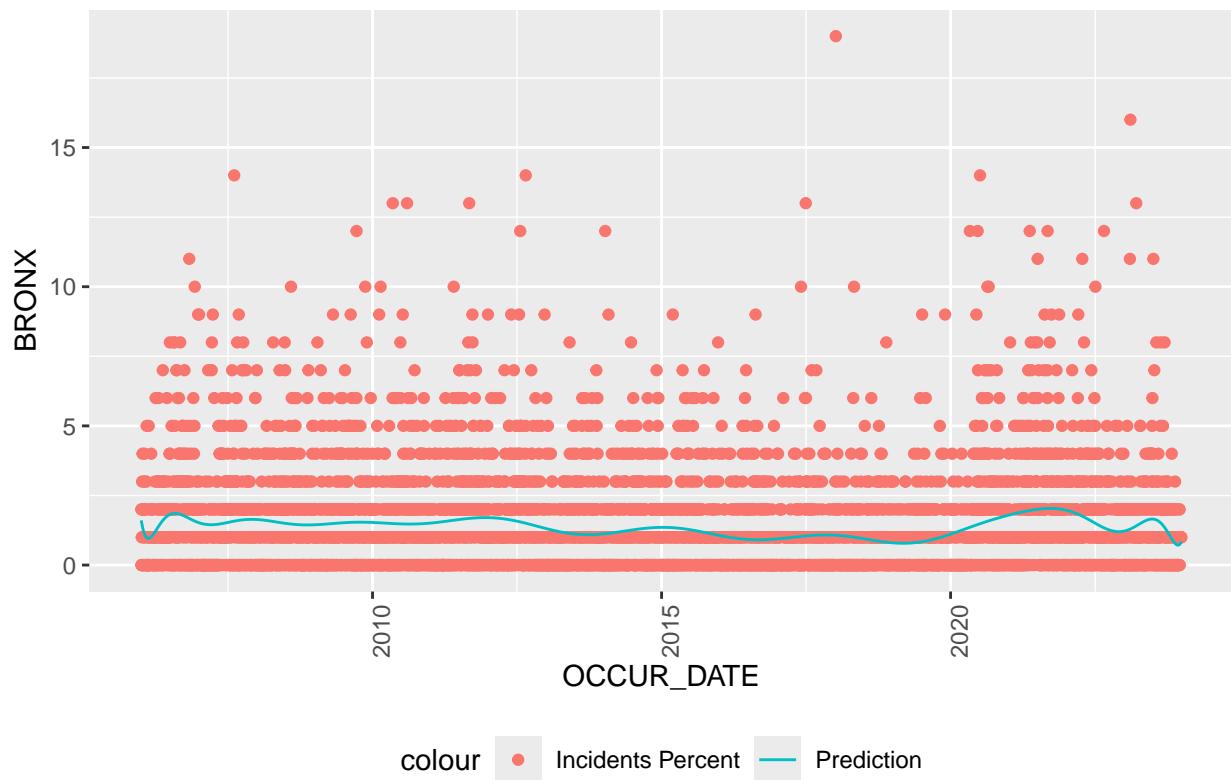
Now that we’ve analyzed shooting incidents over time of day, we will visualize shooting incidents by date.

```
n <- 20

temp <- Shooting_By_Date_Totals

mod <- lm(BRONX ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = BRONX)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Bronx, New York City")
```

Shootings by Date in Bronx, New York City

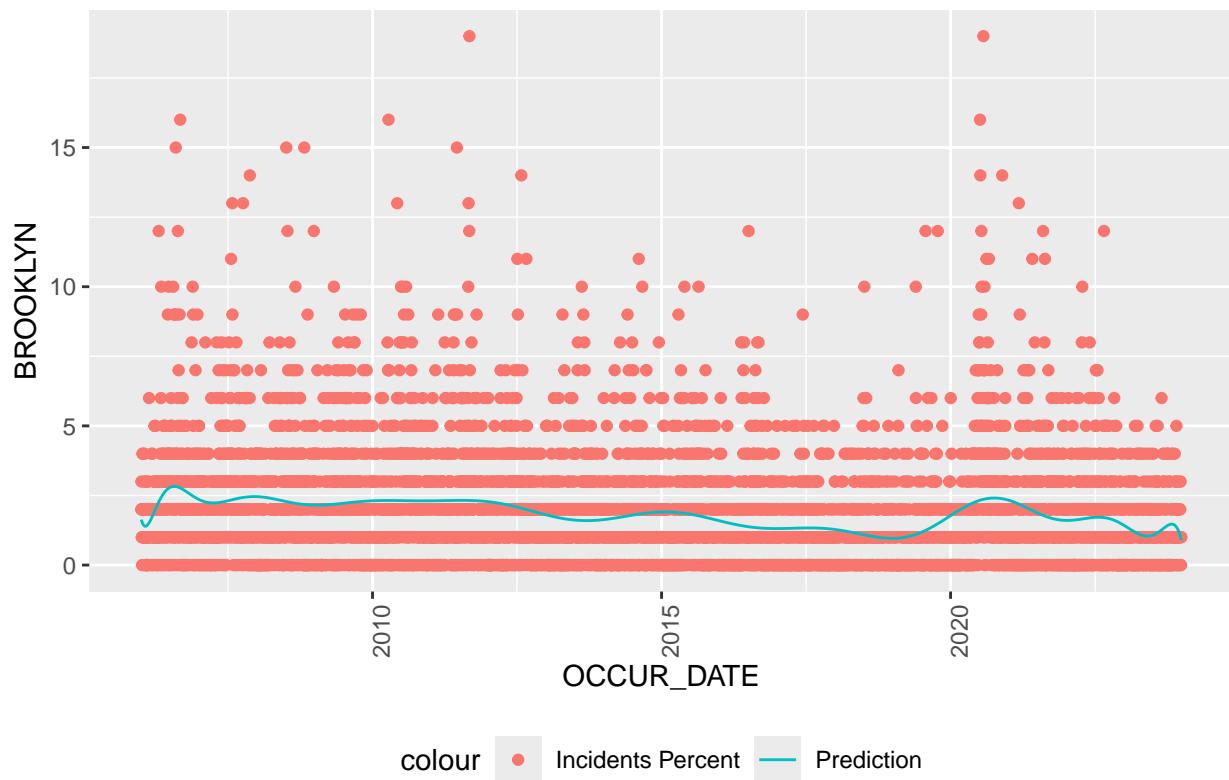


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Date_Totals

mod <- lm(BROOKLYN ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = BROOKLYN)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Brooklyn, New York City")
```

Shootings by Date in Brooklyn, New York City

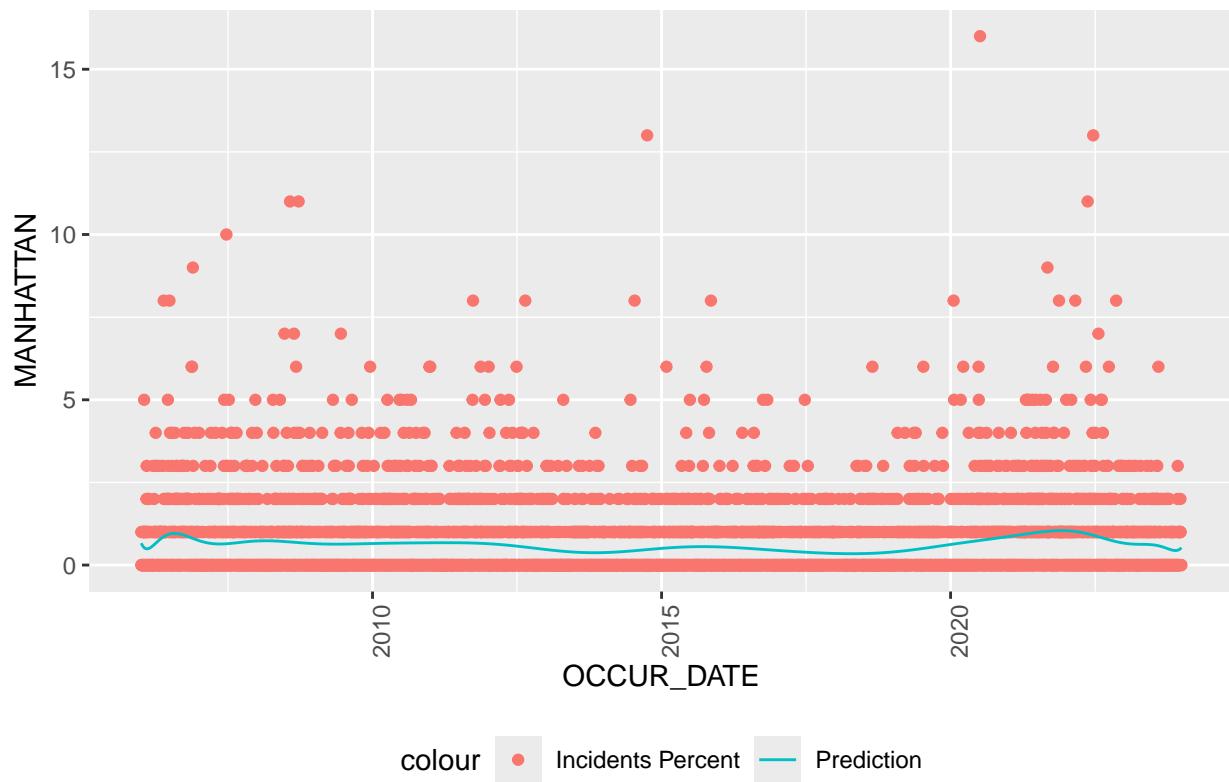


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Date_Totals

mod <- lm(MANHATTAN ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = MANHATTAN)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Manhattan, New York City")
```

Shootings by Date in Manhattan, New York City

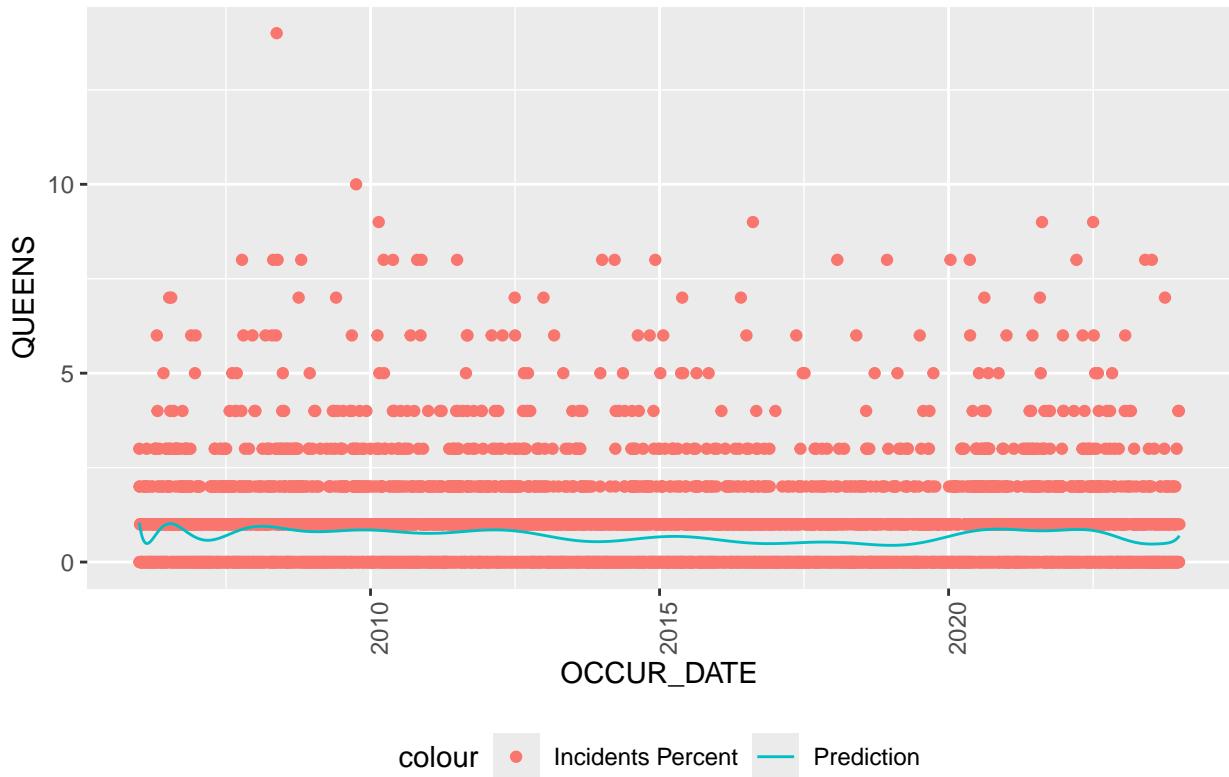


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Date_Totals

mod <- lm(QUEENS ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = QUEENS)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Queens, New York City")
```

Shootings by Date in Queens, New York City

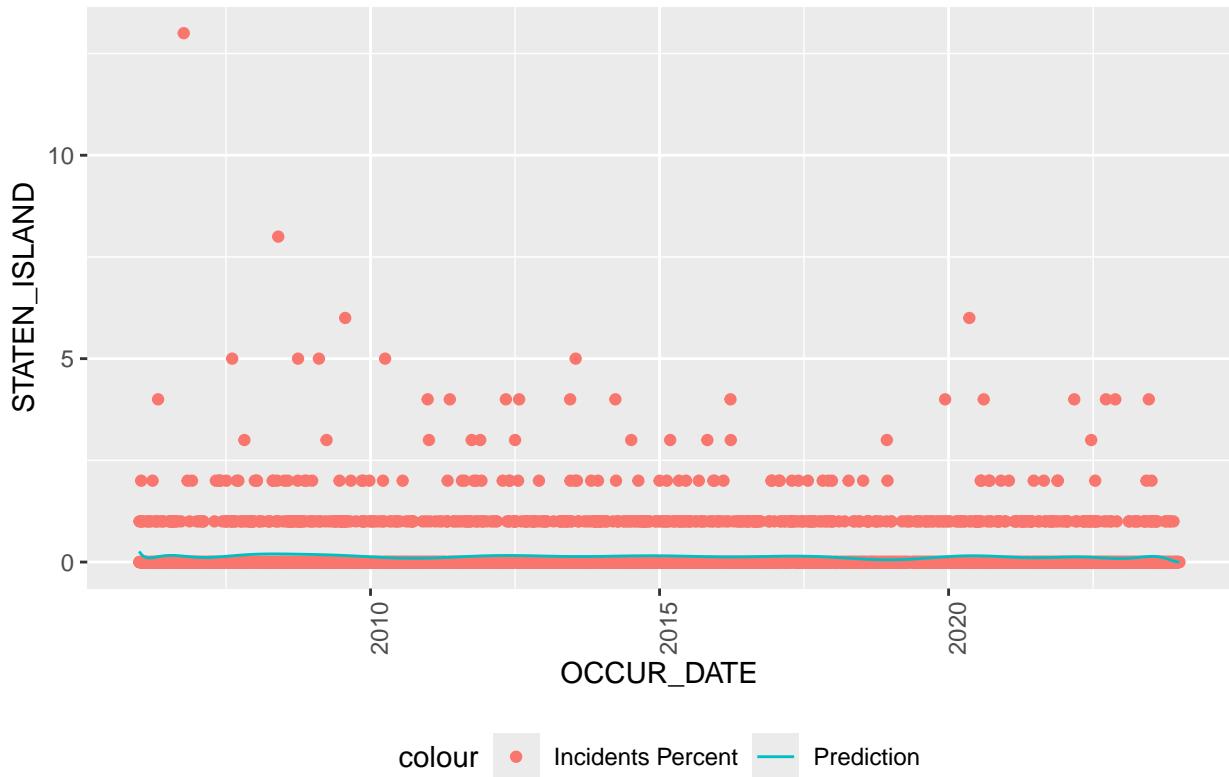


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Date_Totals

mod <- lm(STATEN_ISLAND ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = STATEN_ISLAND)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Staten Island, New York City")
```

Shootings by Date in Staten Island, New York City



```
# Creates a scatterplot graph of shootings over time of day in the borough selected
```

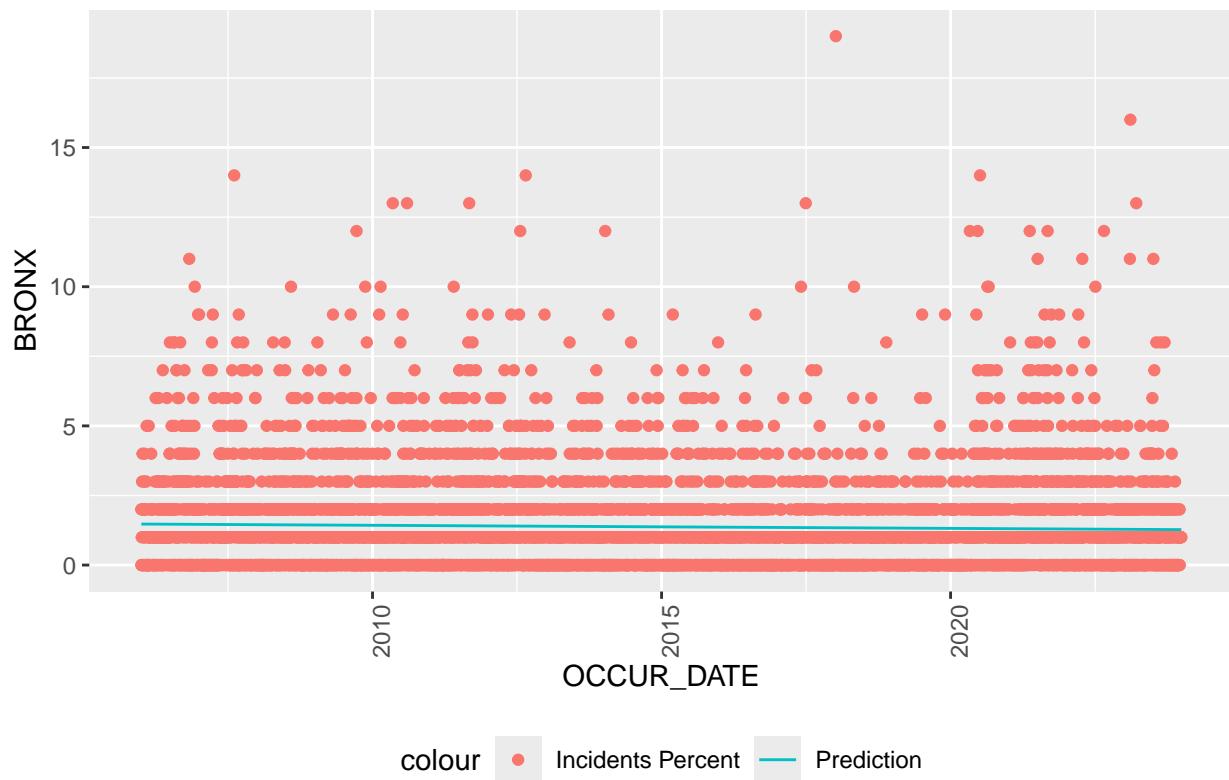
This doesn't seem to give us a good idea of trend, so we're going to do a linear regression without any polynomial degrees above 1 to see if there is an overall slope.

```
n <- 1

temp <- Shooting_By_Date_Totals

mod <- lm(BRONX ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = BRONX)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Bronx, New York City")
```

Shootings by Date in Bronx, New York City

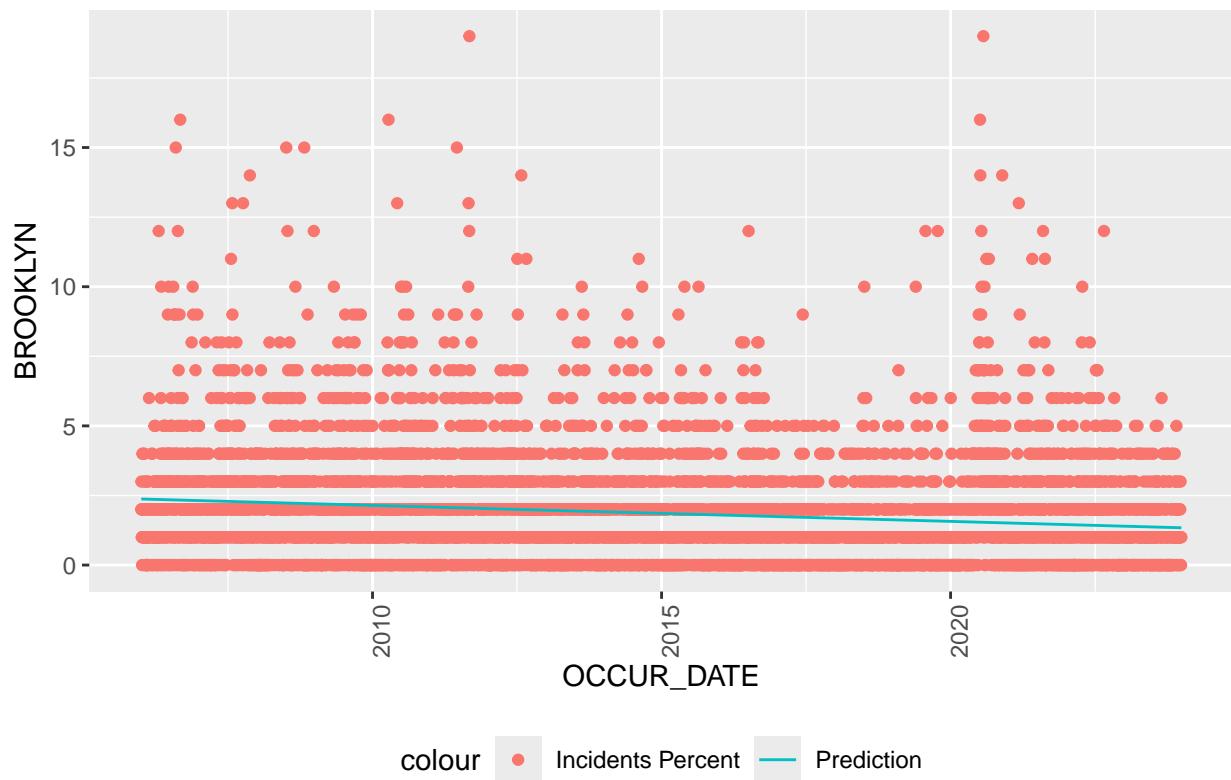


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Date_Totals

mod <- lm(BROOKLYN ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = BROOKLYN)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Brooklyn, New York City")
```

Shootings by Date in Brooklyn, New York City

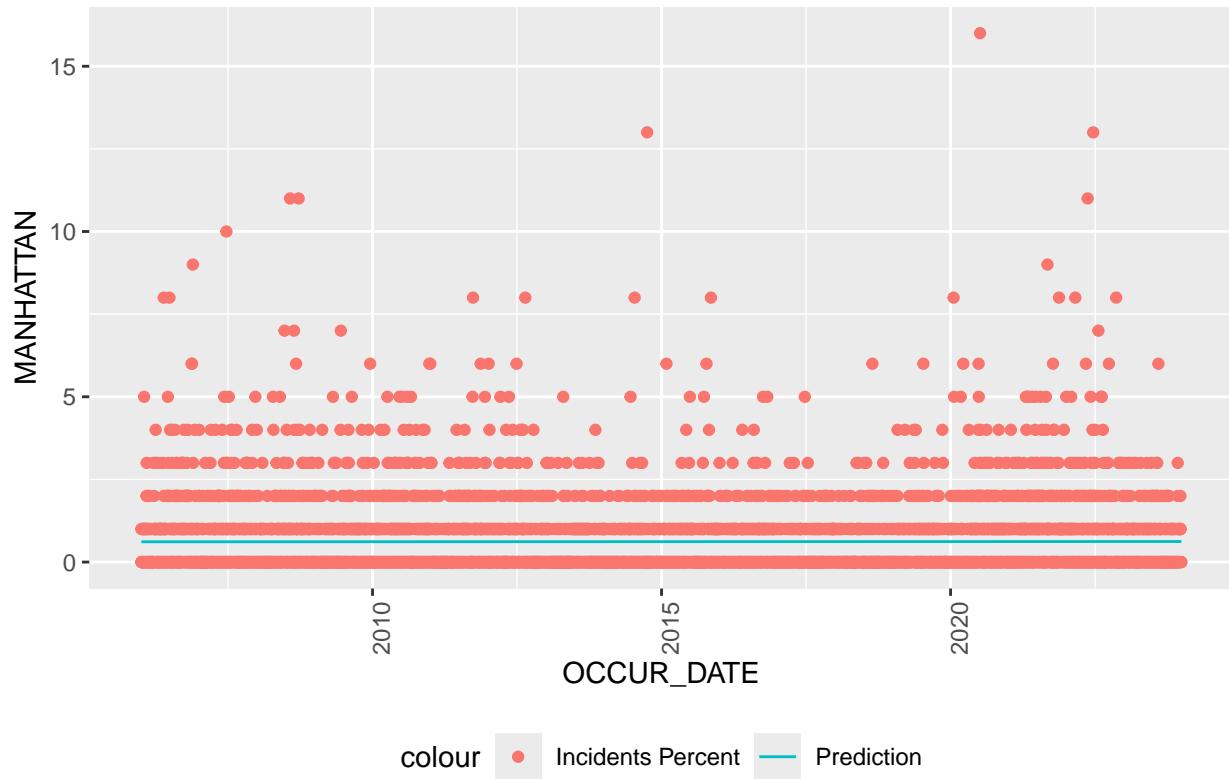


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Date_Totals

mod <- lm(MANHATTAN ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = MANHATTAN)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Manhattan, New York City")
```

Shootings by Date in Manhattan, New York City

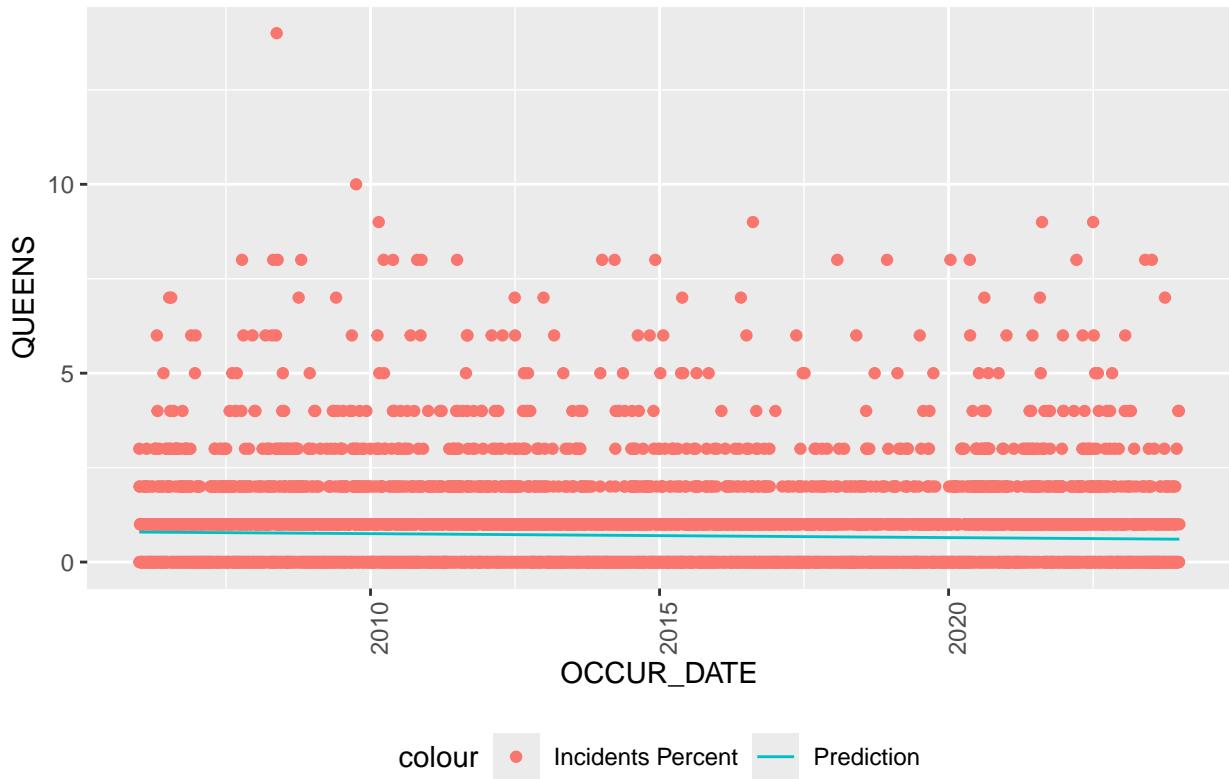


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Date_Totals

mod <- lm(QUEENS ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = QUEENS)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Queens, New York City")
```

Shootings by Date in Queens, New York City

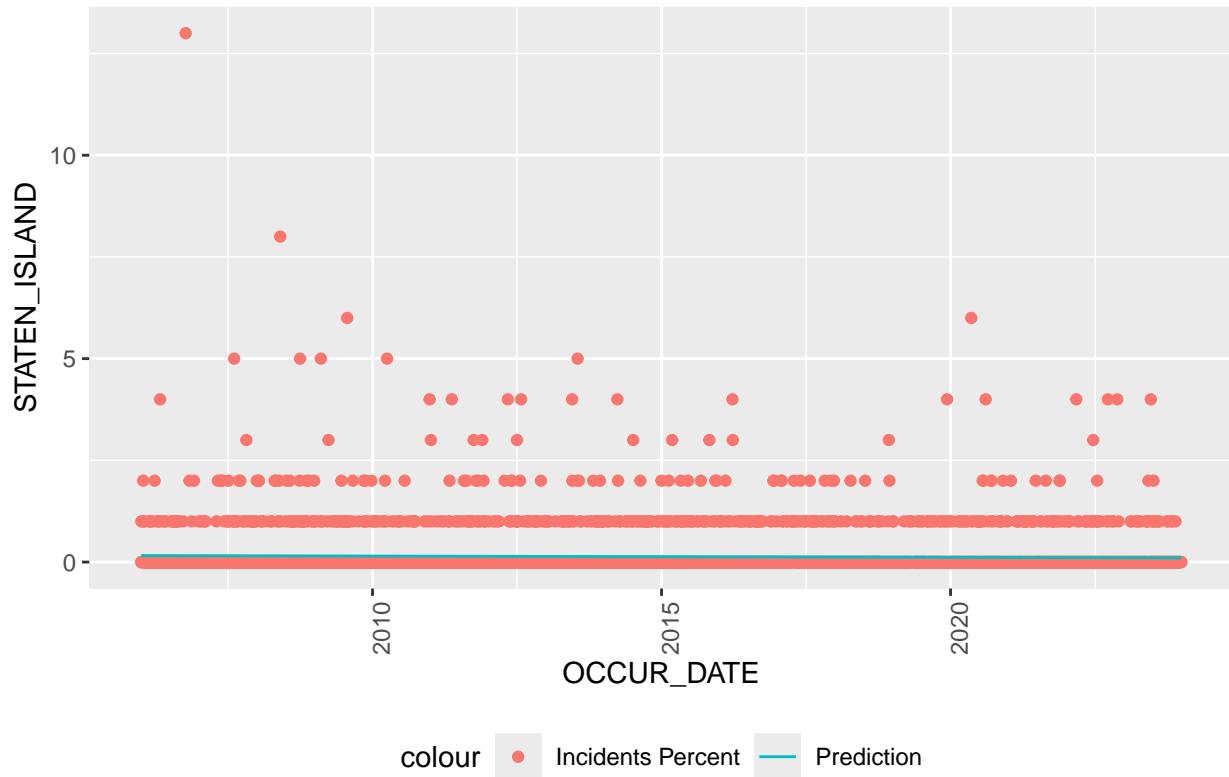


```
# Creates a scatterplot graph of shootings over time of day in the borough selected

temp <- Shooting_By_Date_Totals

mod <- lm(STATEN_ISLAND ~ poly(OCCUR_DATE, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))
temp %>%
  ggplot(aes(x = OCCUR_DATE, y = STATEN_ISLAND)) +
  geom_point(aes(color = "Incidents Percent")) +
  geom_line(aes(x = OCCUR_DATE, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shootings by Date in Staten Island, New York City")
```

Shootings by Date in Staten Island, New York City



```
# Creates a scatterplot graph of shootings over time of day in the borough selected
```

While there isn't much change, there does seem to be a downward trend over time.

Monthly Total Trends Next we will compress the dates into month and day in order to see if there is a seasonal trend over shooting totals.

```
temp <- Shooting_By_Date_Totals %>%
  mutate(OCCUR_MONTH = month(OCCUR_DATE),
        OCCUR_DAY = day(OCCUR_DATE))
# stores month and day separately in a new temp dataframe of the shooting by date totals

temp <- temp %>%
  group_by(OCCUR_MONTH, OCCUR_DAY) %>%
  summarise(BRONX = sum(BRONX),
            BROOKLYN = sum(BROOKLYN),
            MANHATTAN = sum(MANHATTAN),
            QUEENS = sum(QUEENS),
            STATEN_ISLAND = sum(STATEN_ISLAND))

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

#groups by month and day, ignoring year and summing up the totals

temp <- temp %>%
  mutate(Month_Day = 1) %>%
  select(c(OCCUR_MONTH, OCCUR_DAY, Month_Day, BRONX,BROOKLYN,MANHATTAN,QUEENS,STATEN_ISLAND))
```

```

#creates a new column Month_Day, initialized with 1

temp2 <- temp %>%
  filter(OCCUR_DAY < 10) %>%
  transform(Month_Day = str_c(OCCUR_MONTH, ".0", OCCUR_DAY))%>%
  select(c(OCCUR_MONTH, OCCUR_DAY, Month_Day, BRONX,BROOKLYN,MANHATTAN,QUEENS,STATEN_ISLAND))
#creates Month_Day as 1.01 for January 1st, 10.02 for October 2nd, etc.

temp3 <- temp %>%
  filter(OCCUR_DAY > 9) %>%
  transform(Month_Day = str_c(OCCUR_MONTH, ".", OCCUR_DAY))%>%
  select(c(OCCUR_MONTH, OCCUR_DAY, Month_Day, BRONX,BROOKLYN,MANHATTAN,QUEENS,STATEN_ISLAND))
#creates Month_Day for days => 10

temp <- rbind(temp2, temp3)
#combines days below and above 10

temp <- temp %>%
  transform(Month_Day = as.numeric(Month_Day))
#makes Month_Day numeric

rm(temp2)
rm(temp3)
#removes partial transformation dataframes

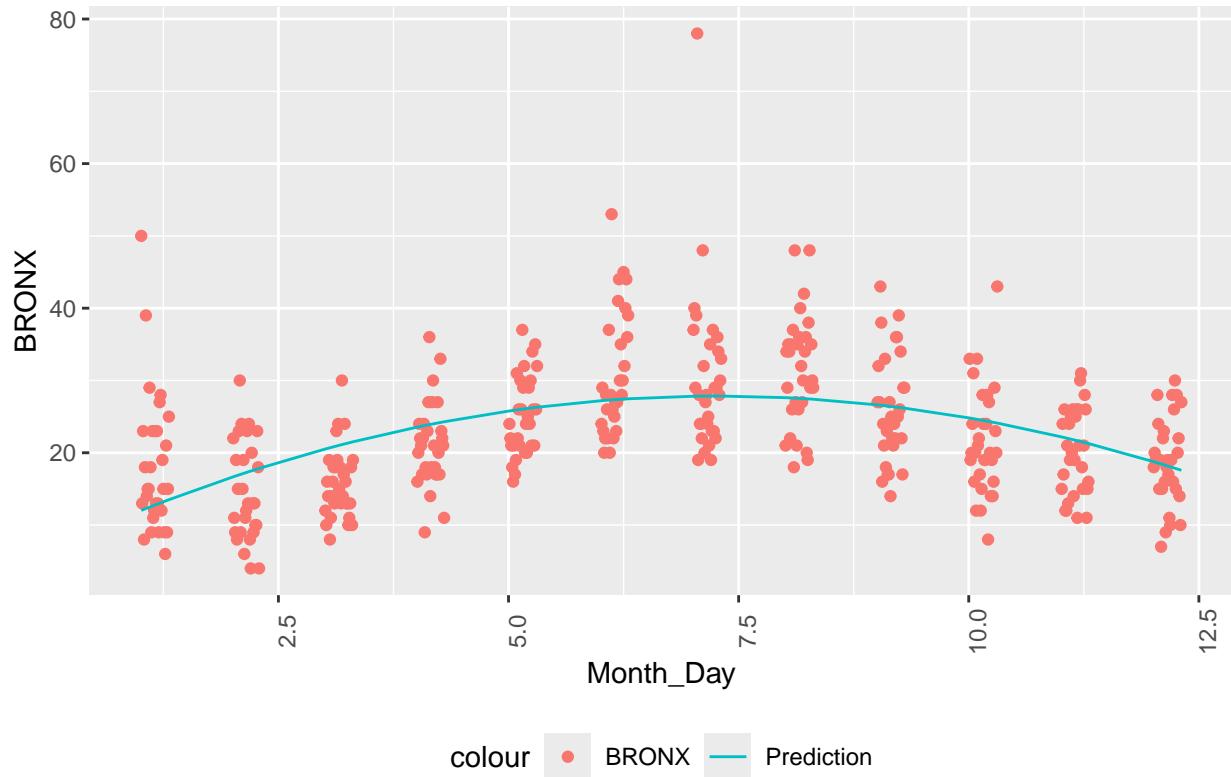
n <- 2
#sets degree for modeling in below codes

mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),) +
  labs(title = "Shootings by Date in Bronx, New York City")

```

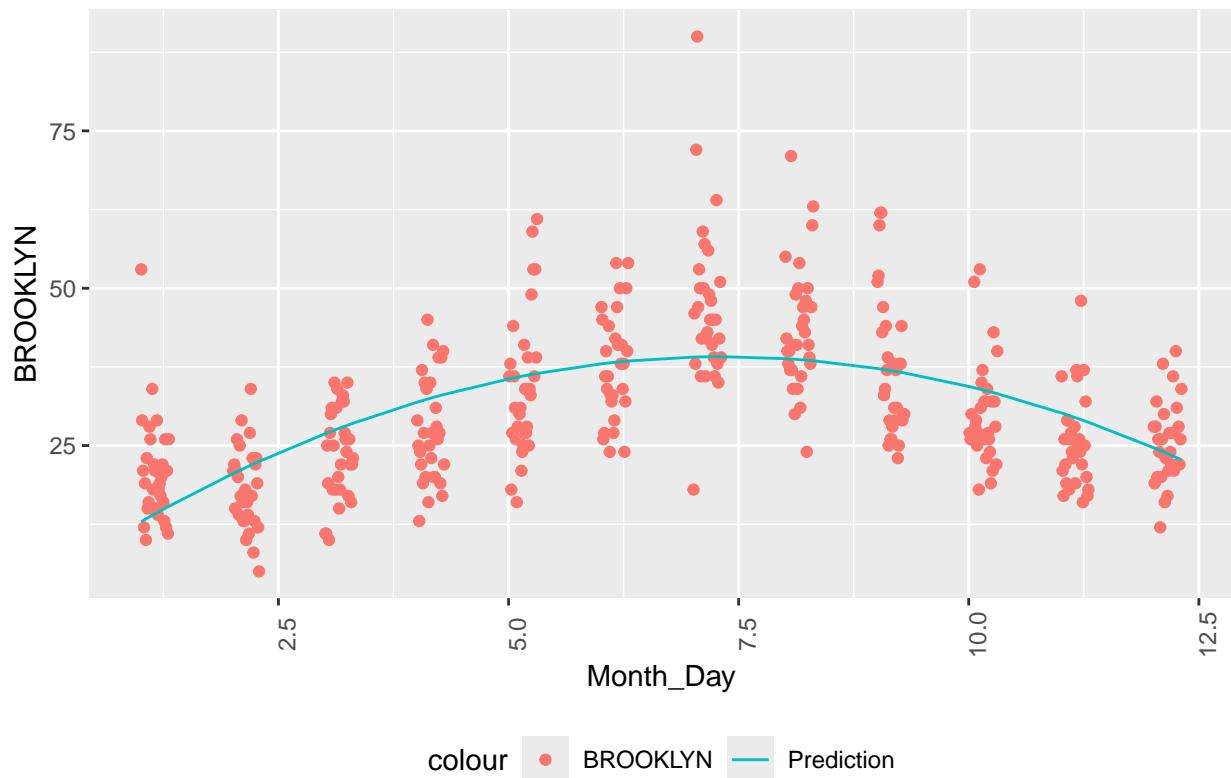
Shootings by Date in Bronx, New York City



```
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "Shootings by Date in Brooklyn, New York City")
```

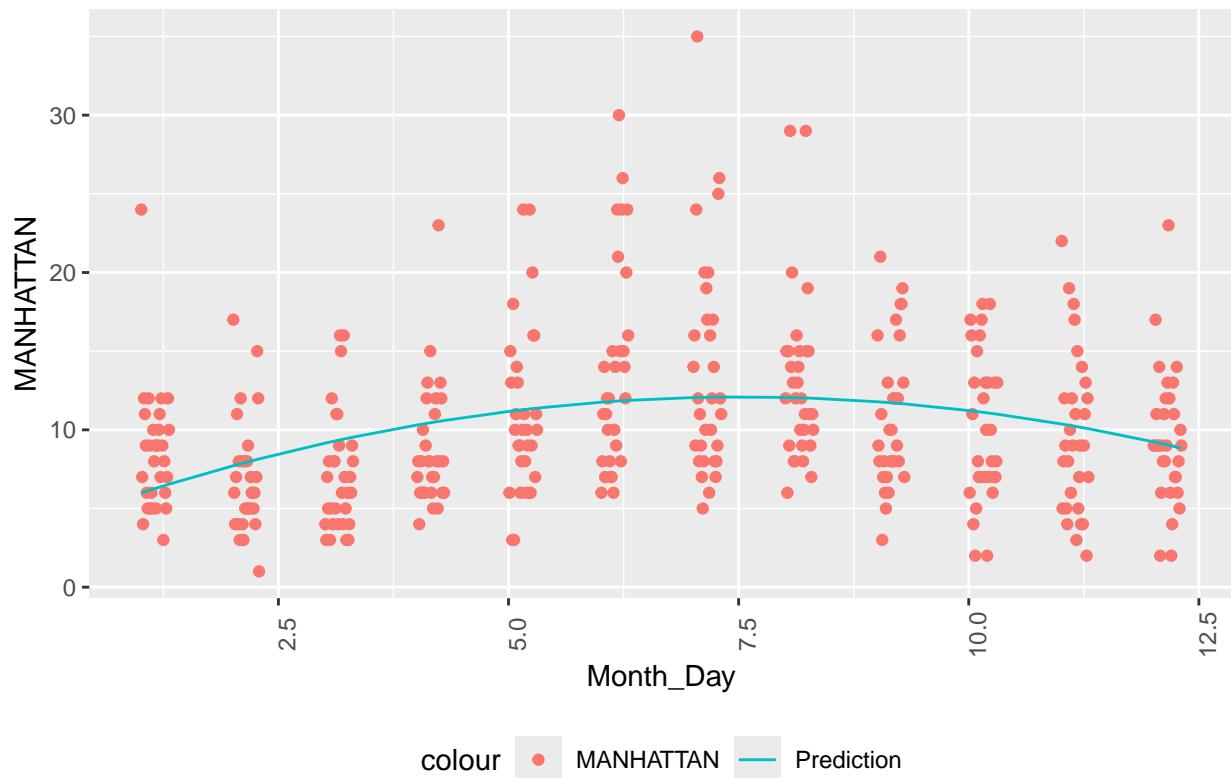
Shootings by Date in Brooklyn, New York City



```
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "Shootings by Date in Manhattan, New York City")
```

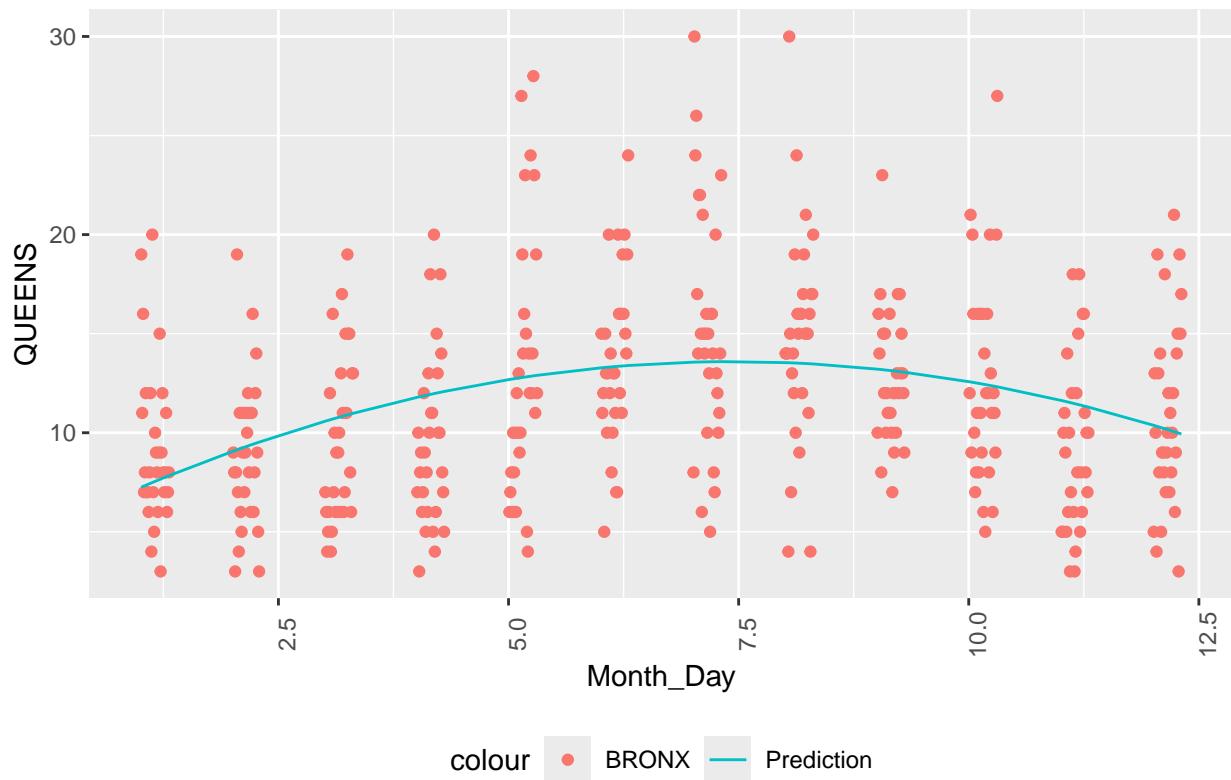
Shootings by Date in Manhattan, New York City



```
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "Shootings by Date in Queens, New York City")
```

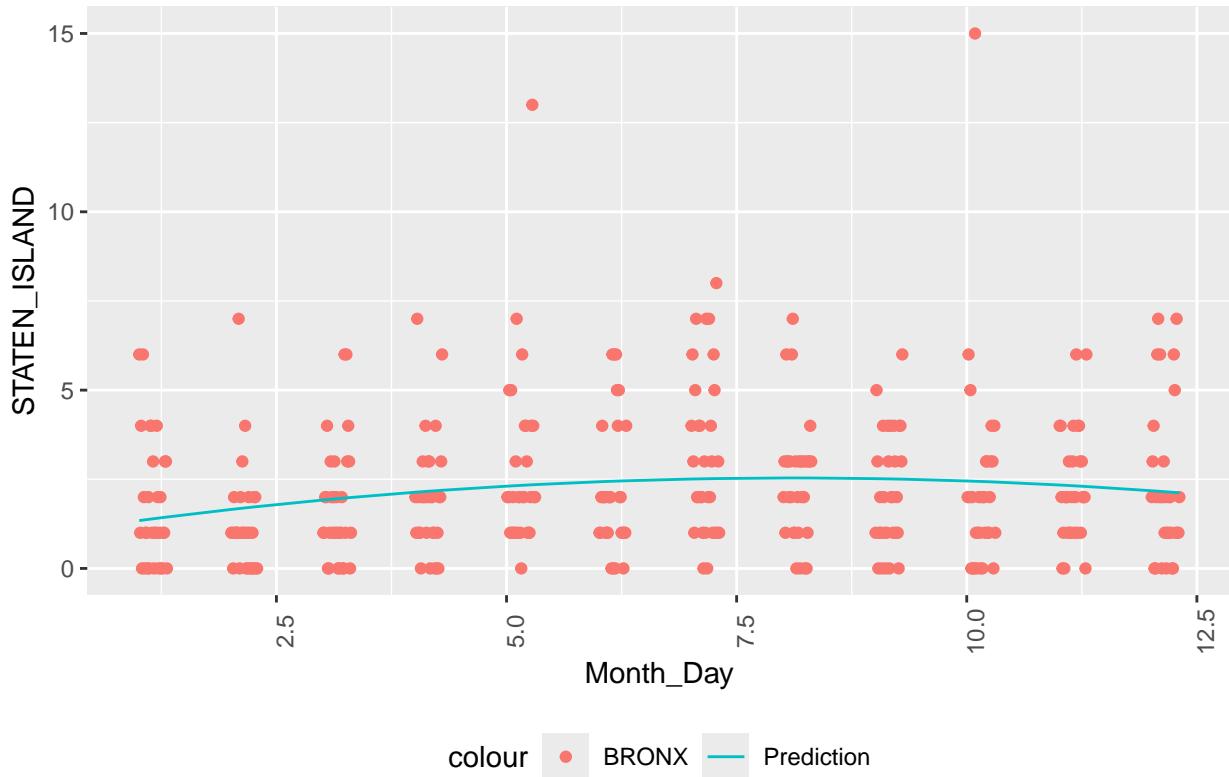
Shootings by Date in Queens, New York City



```
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "Shootings by Date in Staten Island, New York City")
```

Shootings by Date in Staten Island, New York City

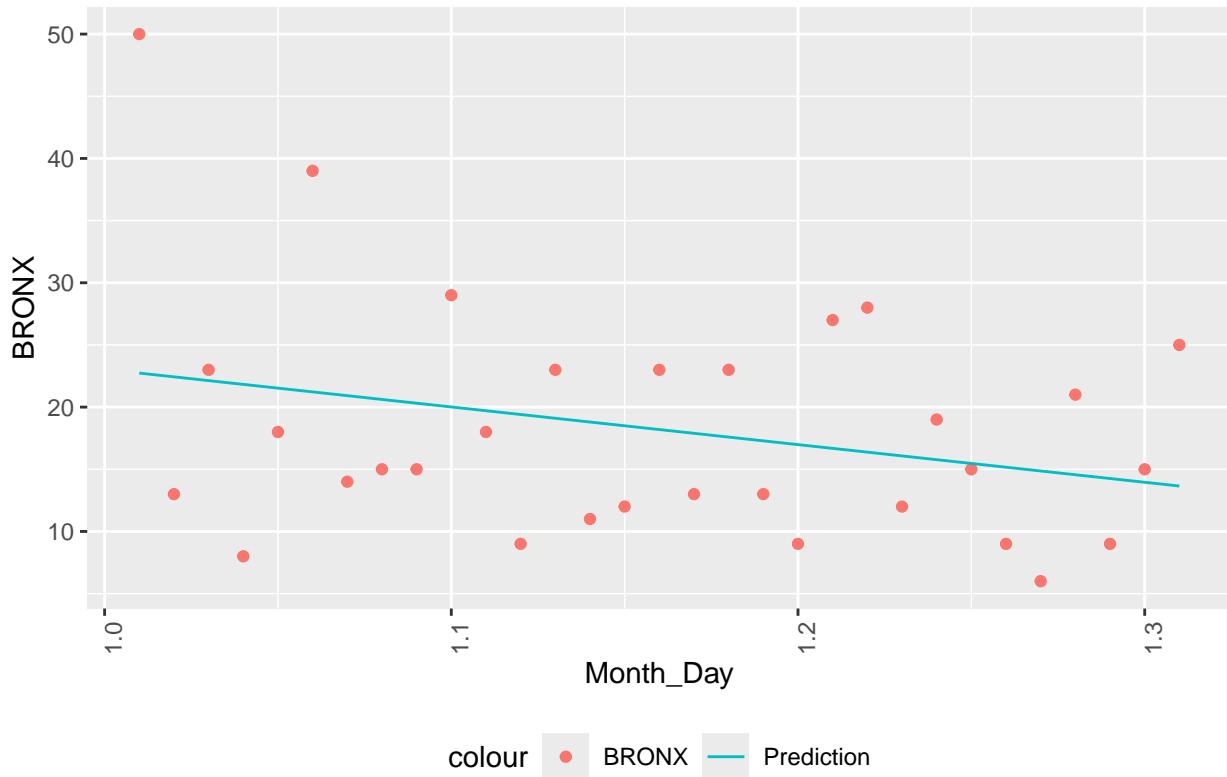


There does seem to be an increase in shootings midyear through the five boroughs. Let's look at this per month.

```
n <- 1
temp1 <- temp %>%
  filter(OCCUR_MONTH==1)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 1) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "January Shooting Totals by Date in Bronx, New York City")
```

January Shooting Totals by Date in Bronx, New York City

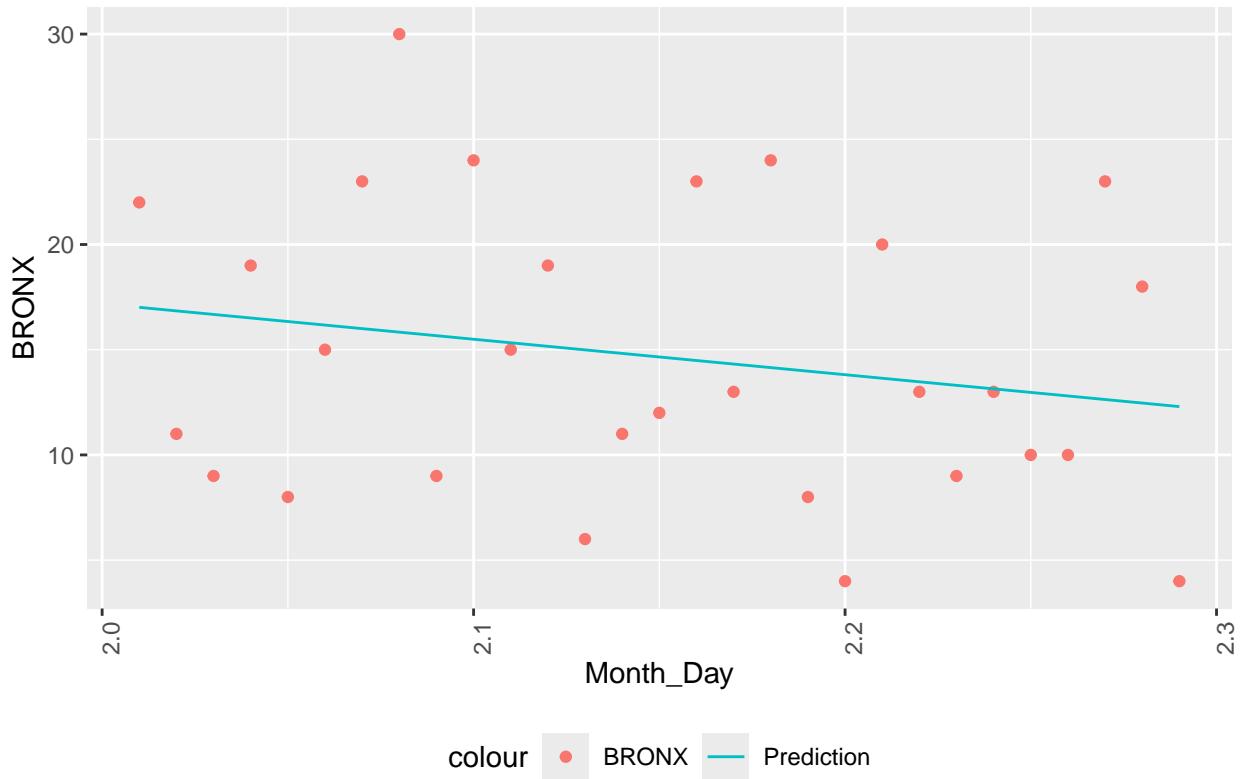


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==2)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 2) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "February Shooting Totals by Date in Bronx, New York City")
```

February Shooting Totals by Date in Bronx, New York City

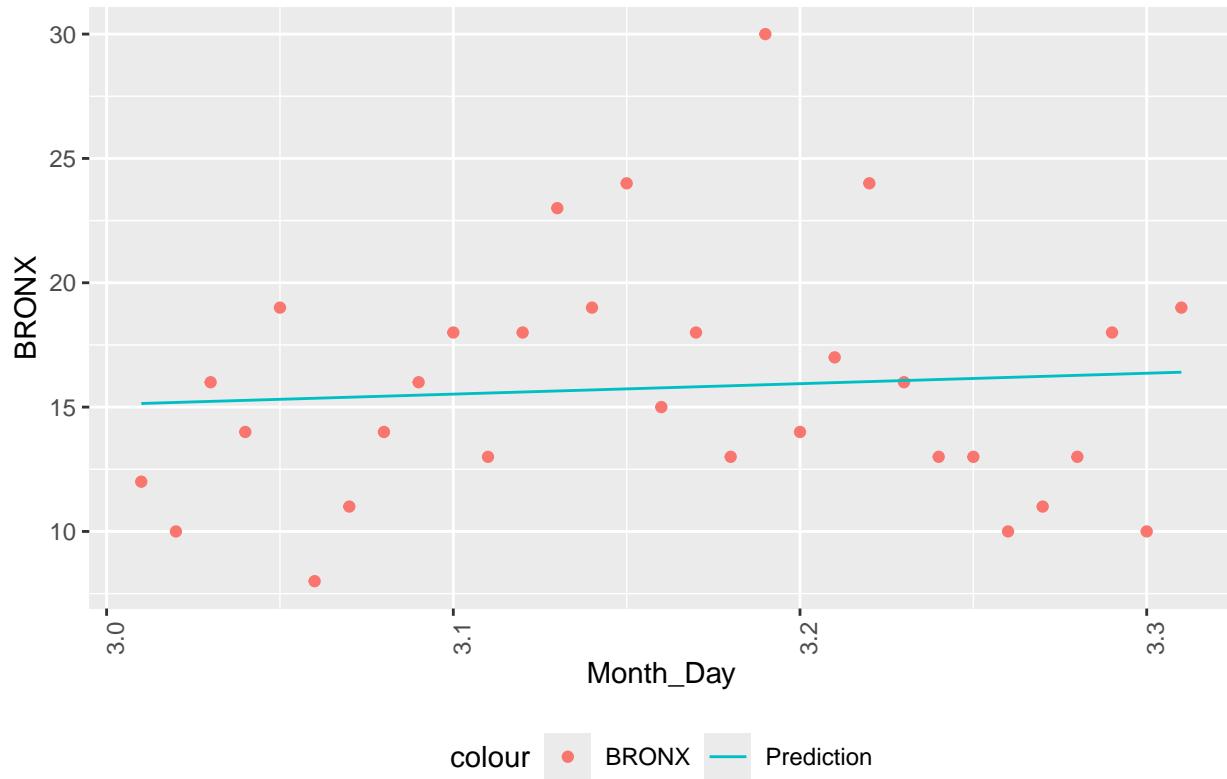


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==3)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 3) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "March Shooting Totals by Date in Bronx, New York City")
```

March Shooting Totals by Date in Bronx, New York City

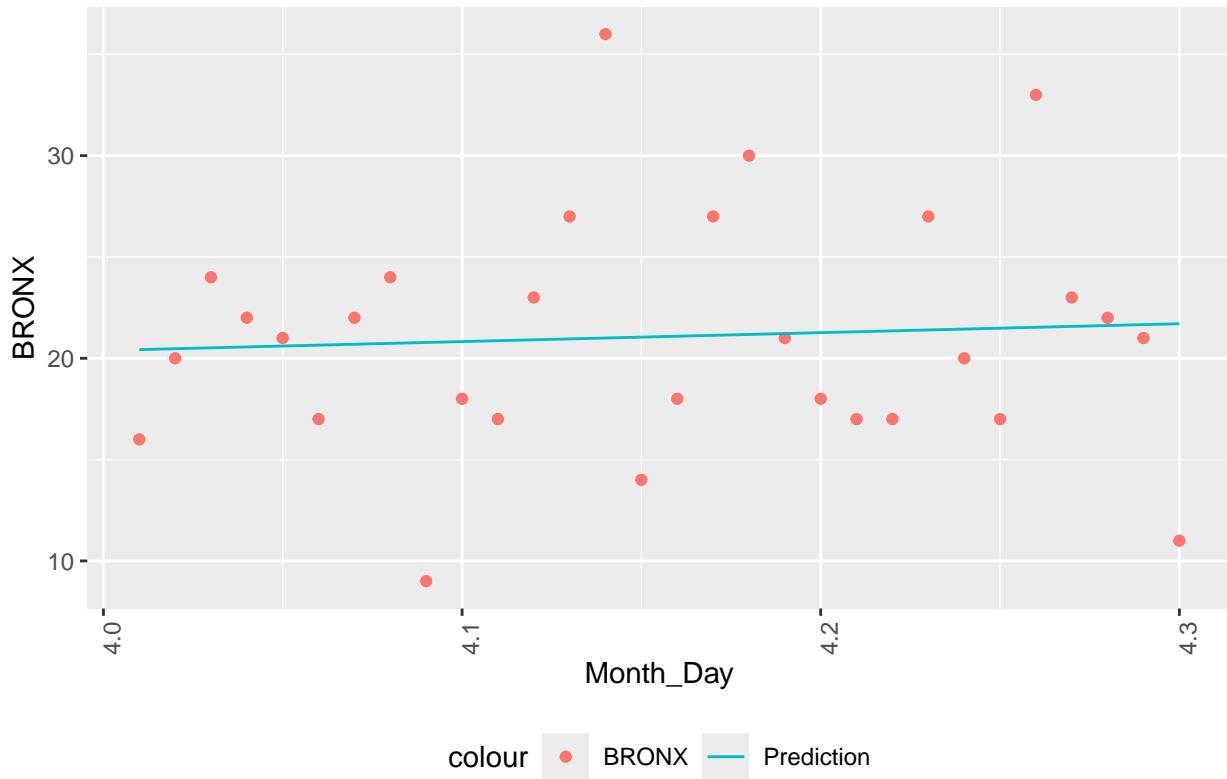


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==4)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 4) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "April Shooting Totals by Date in Bronx, New York City")
```

April Shooting Totals by Date in Bronx, New York City

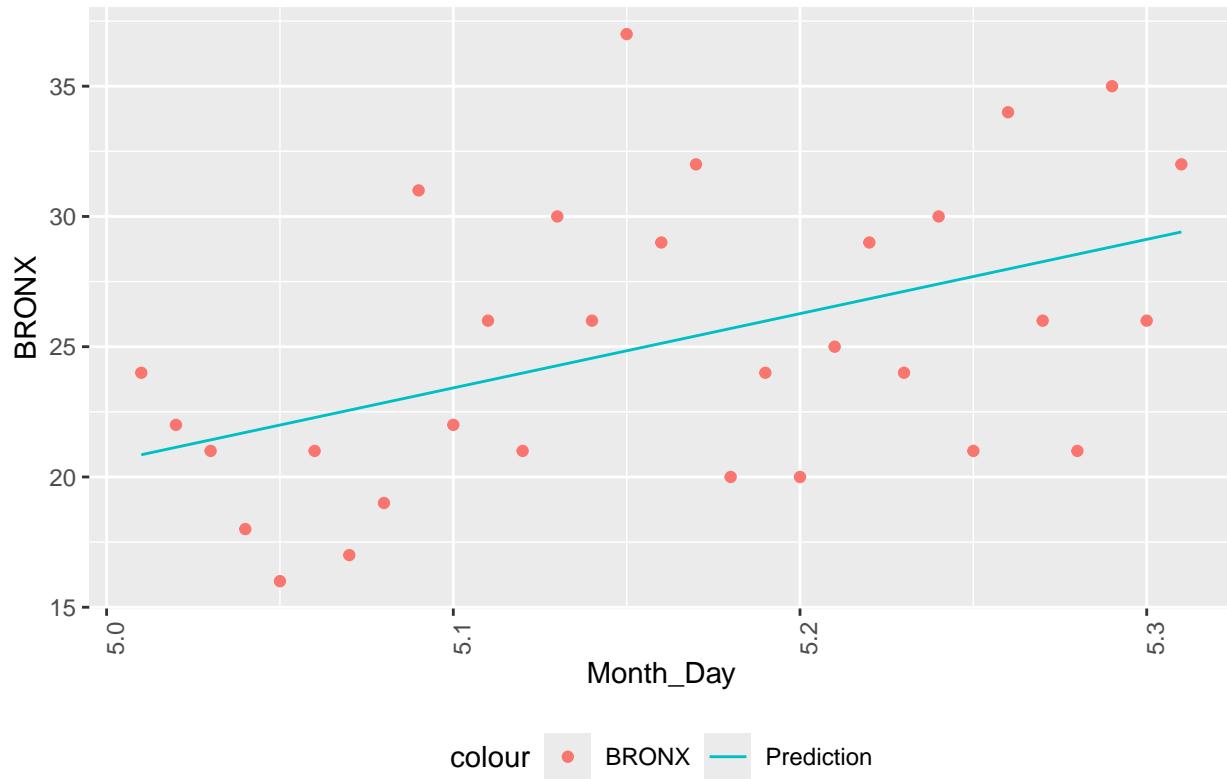


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==5)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 5) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "May Shooting Totals by Date in Bronx, New York City")
```

May Shooting Totals by Date in Bronx, New York City

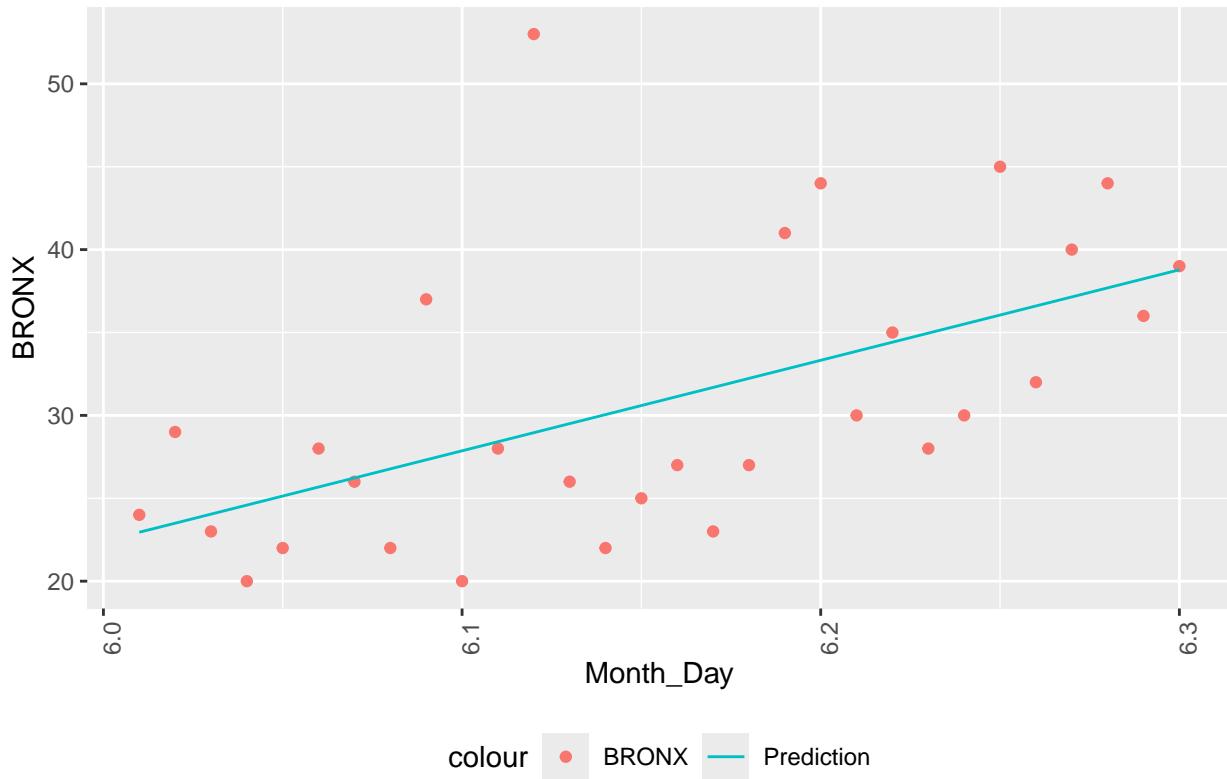


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==6)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 6) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "June Shooting Totals by Date in Bronx, New York City")
```

June Shooting Totals by Date in Bronx, New York City

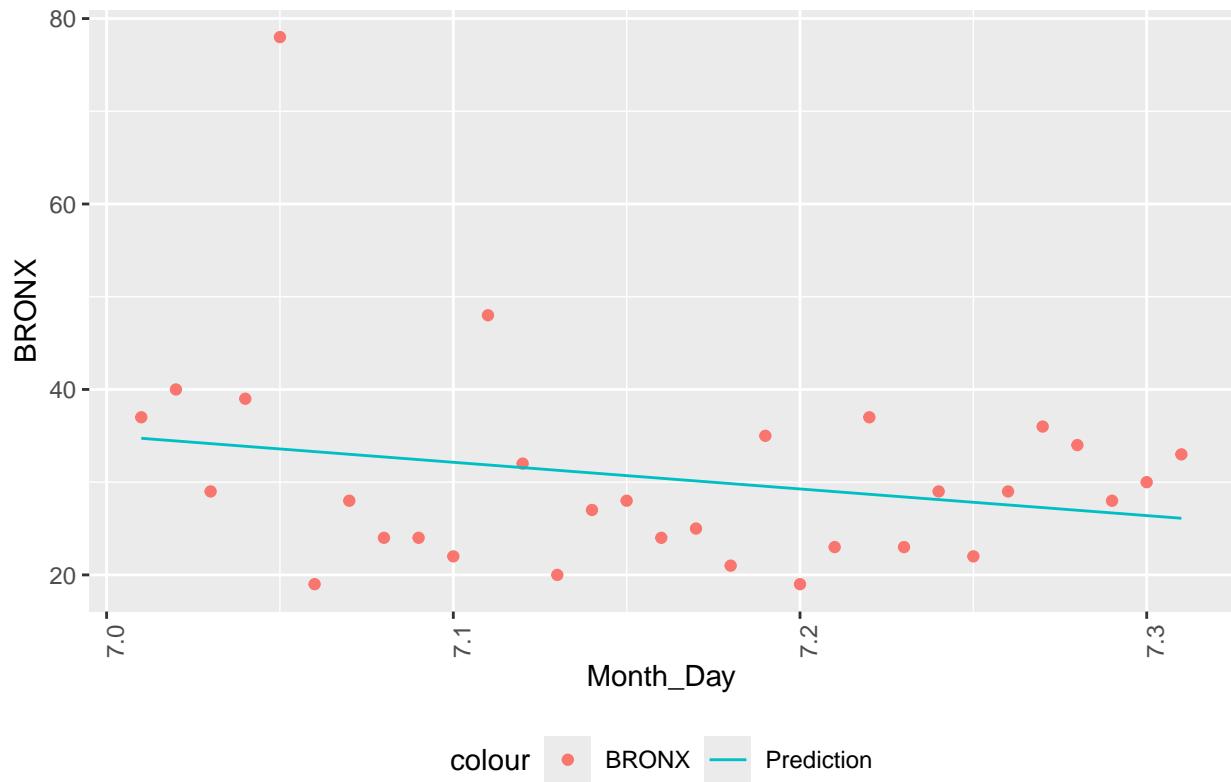


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==7)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 7) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "July Shooting Totals by Date in Bronx, New York City")
```

July Shooting Totals by Date in Bronx, New York City

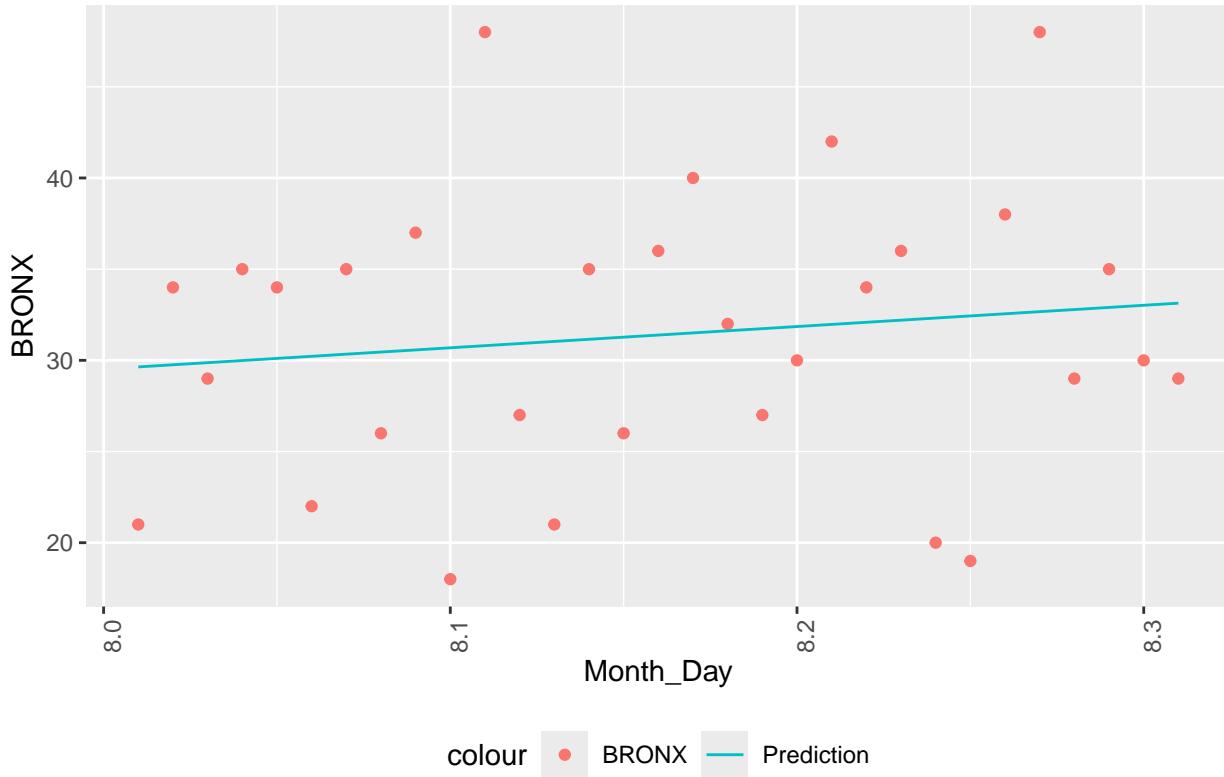


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==8)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 8) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "August Shooting Totals by Date in Bronx, New York City")
```

August Shooting Totals by Date in Bronx, New York City

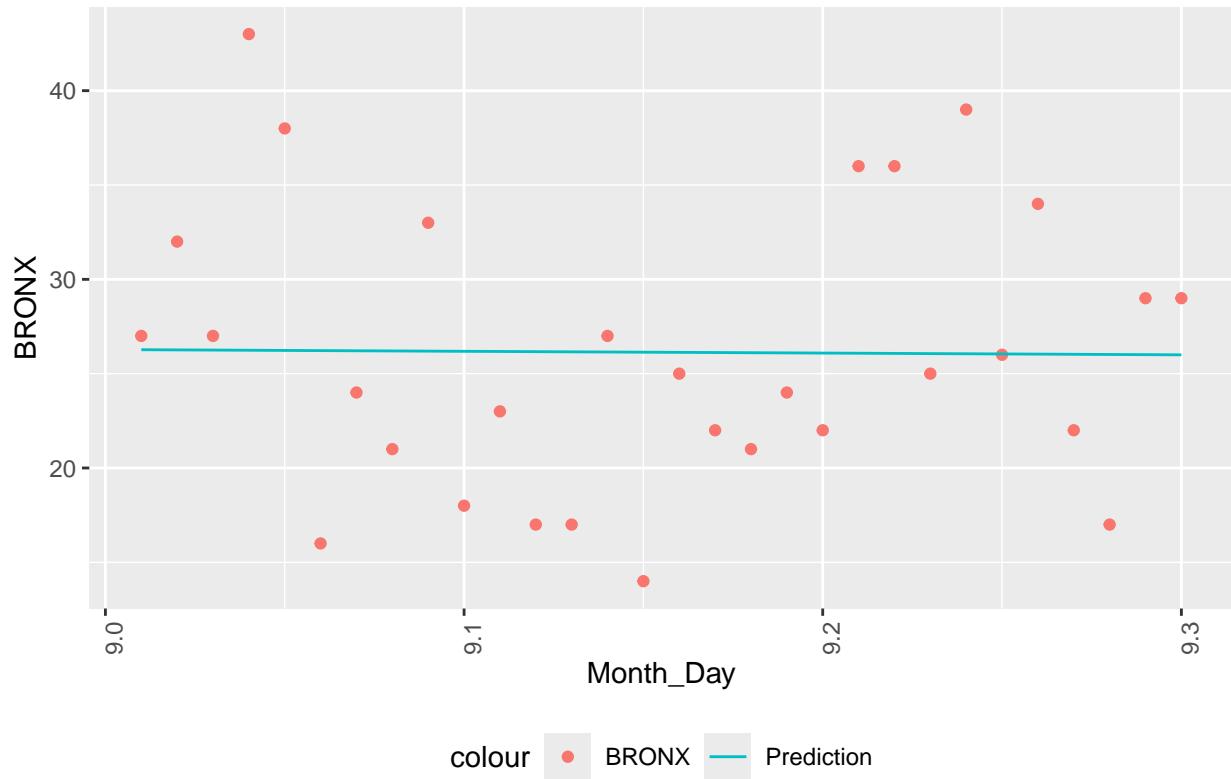


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==9)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 9) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "September Shooting Totals by Date in Bronx, New York City")
```

September Shooting Totals by Date in Bronx, New York City

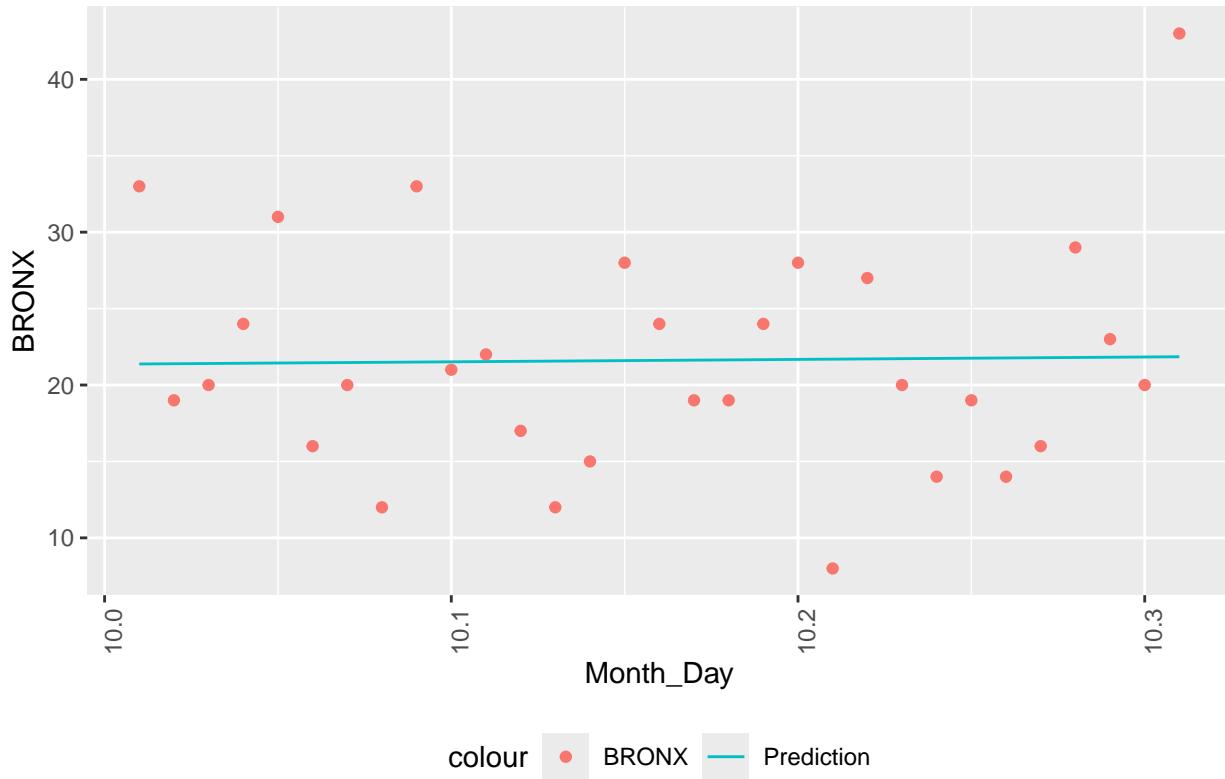


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==10)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 10) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "October Shooting Totals by Date in Bronx, New York City")
```

October Shooting Totals by Date in Bronx, New York City

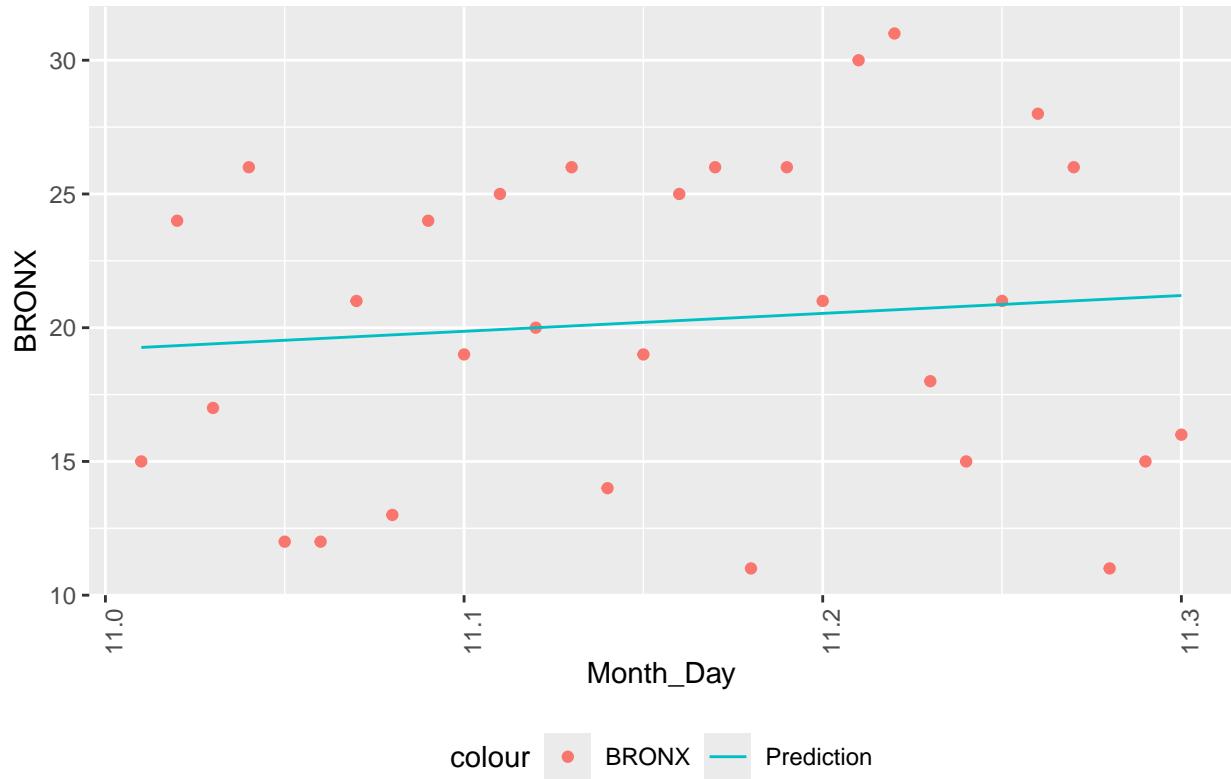


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==11)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 11) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "November Shooting Totals by Date in Bronx, New York City")
```

November Shooting Totals by Date in Bronx, New York City

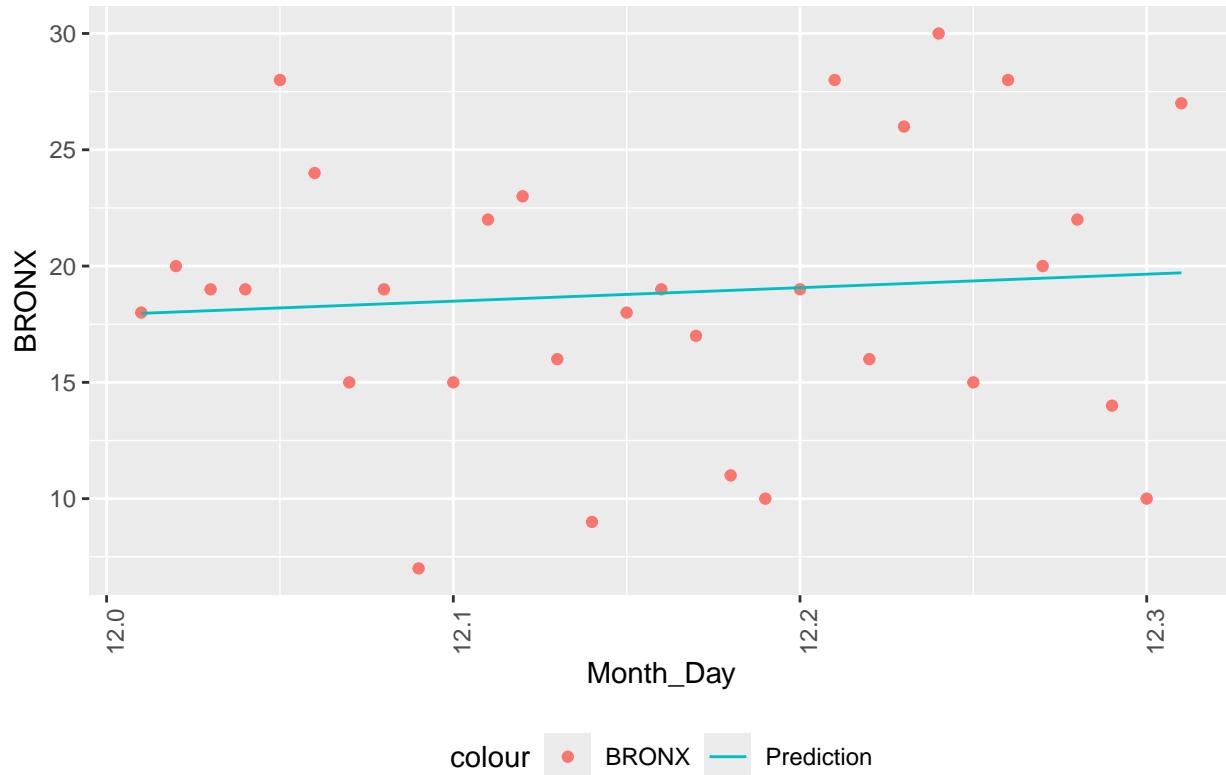


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==12)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 12) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "December Shooting Totals by Date in Bronx, New York City")
```

December Shooting Totals by Date in Bronx, New York City



```
#creates a scatterplot of the shootings over a month, daily, in the selected borough
```

Results:

- January: downward trend
- February: downward trend
- March: slight upward trend
- April: slight upward trend
- May: strong upward trend
- June: strong upward trend
- July: downward trend
- August: upward trend
- September: slight downward trend
- October: slight upward trend
- November: slight upward trend
- December: slight upward trend

These results in the Bronx seem to support the idea of a spike at midyear, around May and June. However, are total trends accurate?

Monthly Average Trends Let's look at the averages over the years 2006 through 2023 for the monthly and yearly trends next.

```
temp <- Shooting_By_Date_Totals %>%
  mutate(OCCUR_MONTH = month(OCCUR_DATE),
        OCCUR_DAY = day(OCCUR_DATE))
# stores month and day separately in a new temp dataframe of the shooting by date totals (years 2006 - 2023)
```

```

temp <- temp %>%
  group_by(OCCUR_MONTH, OCCUR_DAY) %>%
  summarize(BRONX = sum(BRONX)/(2023-2006),
            BROOKLYN = sum(BROOKLYN)/(2023-2006),
            MANHATTAN = sum(MANHATTAN)/(2023-2006),
            QUEENS = sum(QUEENS)/(2023-2006),
            STATEN_ISLAND = sum(STATEN_ISLAND)/(2023-2006))

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

#groups by month and day, ignoring year and creating averages of the totals per year instead
temp <- temp %>%
  mutate(Month_Day = 1) %>%
  select(c(OCCUR_MONTH, OCCUR_DAY, Month_Day, BRONX,BROOKLYN,MANHATTAN,QUEENS,STATEN_ISLAND))
#creates a new column Month_Day, initialized with 1

temp2 <- temp %>%
  filter(OCCUR_DAY < 10) %>%
  transform(Month_Day = str_c(OCCUR_MONTH, ".0", OCCUR_DAY))%>%
  select(c(OCCUR_MONTH, OCCUR_DAY, Month_Day, BRONX,BROOKLYN,MANHATTAN,QUEENS,STATEN_ISLAND))
#creates Month_Day as 1.01 for January 1st, 10.02 for October 2nd, etc.

temp3 <- temp %>%
  filter(OCCUR_DAY > 9) %>%
  transform(Month_Day = str_c(OCCUR_MONTH, ".", OCCUR_DAY))%>%
  select(c(OCCUR_MONTH, OCCUR_DAY, Month_Day, BRONX,BROOKLYN,MANHATTAN,QUEENS,STATEN_ISLAND))
#creates Month_Day for days => 10

temp <- rbind(temp2, temp3)
#combines days below and above 10

temp <- temp %>%
  transform(Month_Day = as.numeric(Month_Day))
#makes Month_Day numeric

rm(temp2)
rm(temp3)
#removes partial transformation dataframes

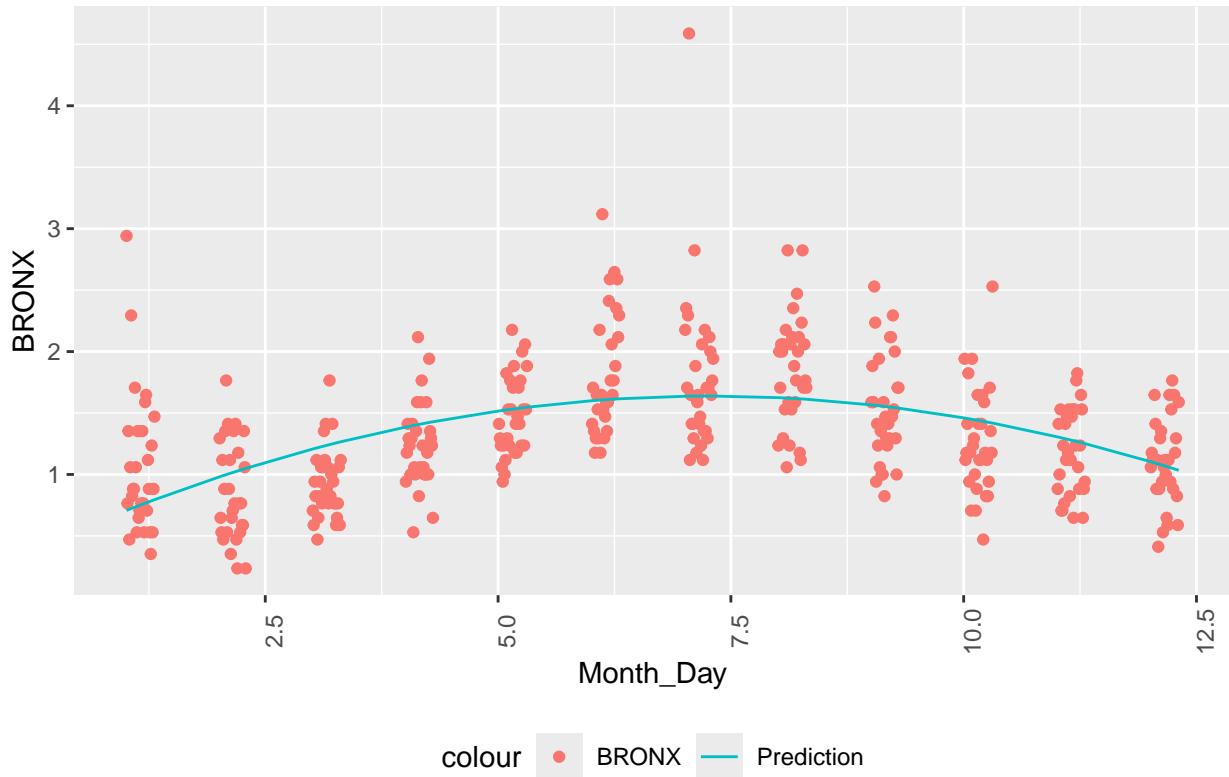
n <- 2
#sets degree for modeling in below codes

mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),) +
  labs(title = "Shootings by Date in Bronx, New York City")

```

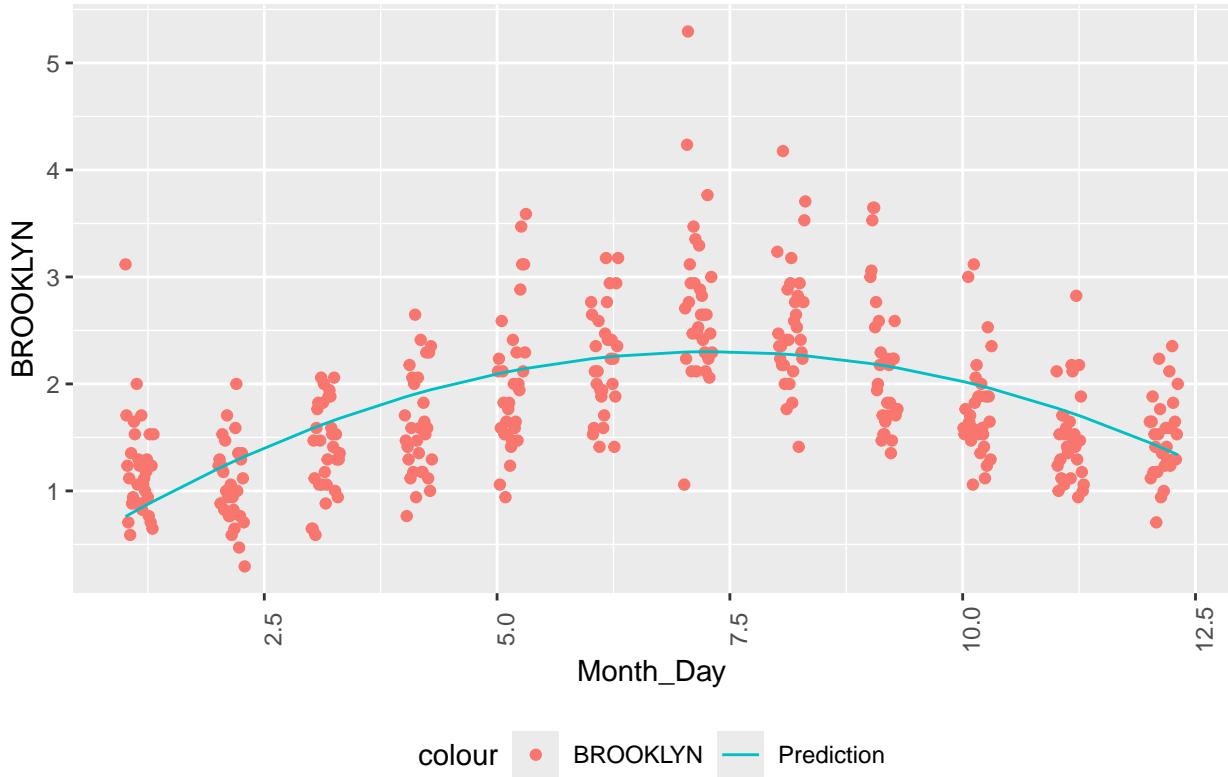
Shootings by Date in Bronx, New York City



```
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "Shootings by Date in Brooklyn, New York City")
```

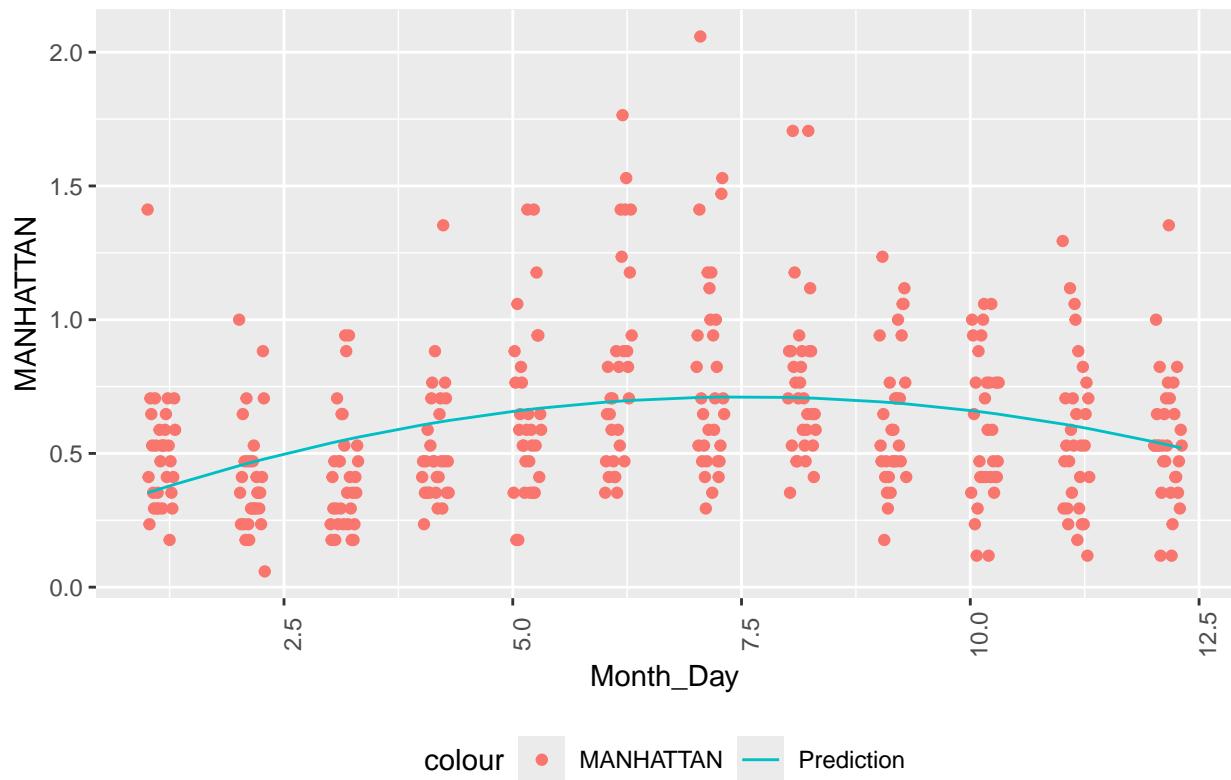
Shootings by Date in Brooklyn, New York City



```
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "Shootings by Date in Manhattan, New York City")
```

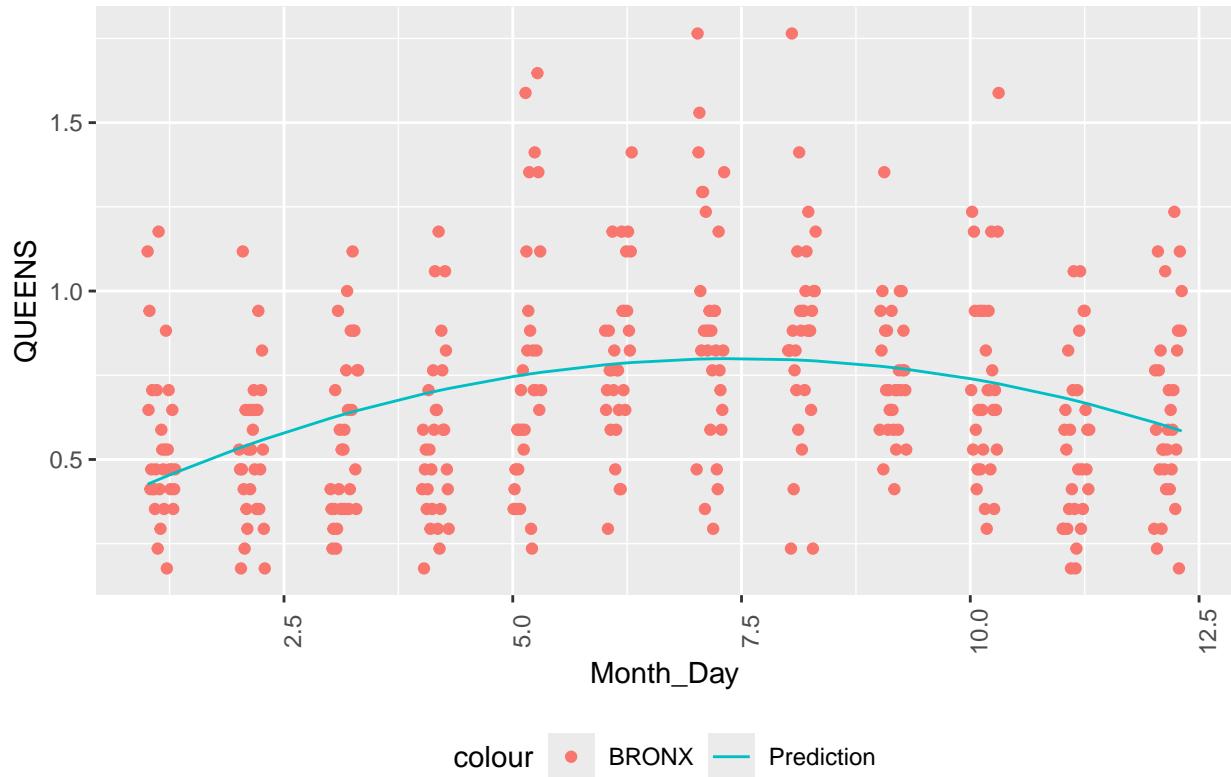
Shootings by Date in Manhattan, New York City



```
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "Shootings by Date in Queens, New York City")
```

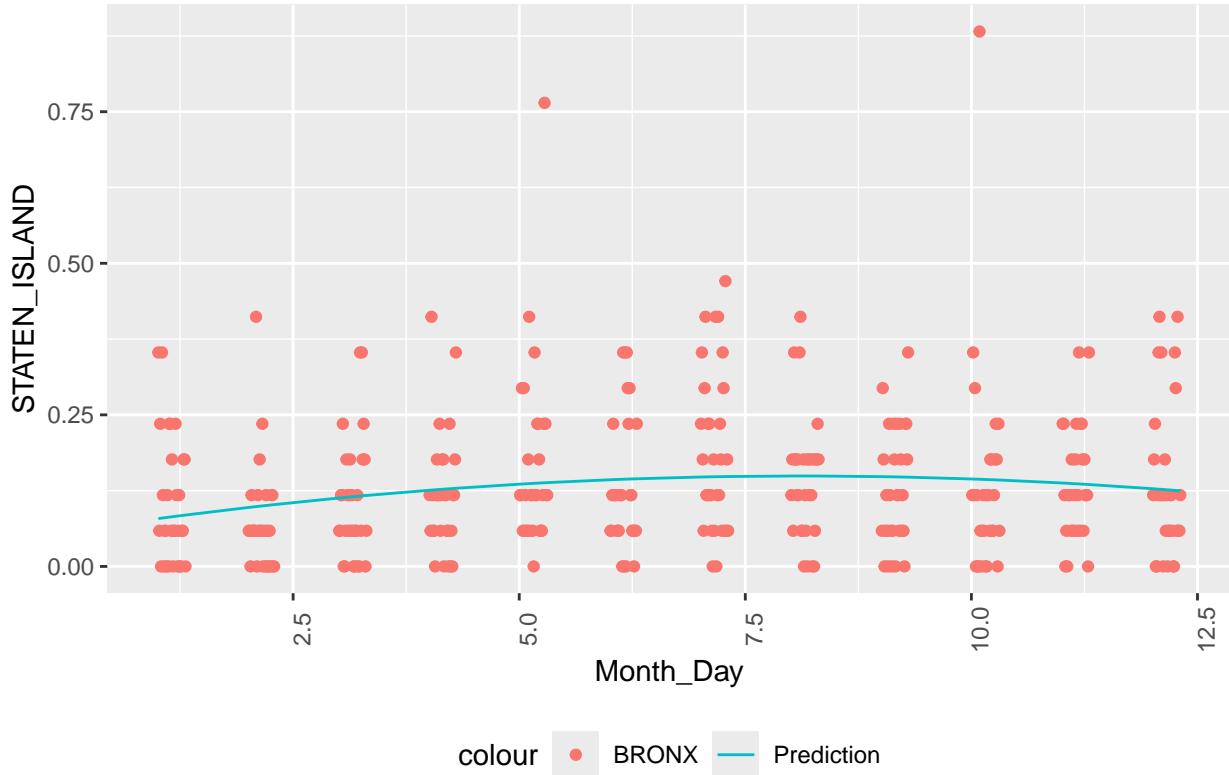
Shootings by Date in Queens, New York City



```
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp)
temp <- temp %>% mutate(pred = predict(mod))

temp %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "Shootings by Date in Staten Island, New York City")
```

Shootings by Date in Staten Island, New York City

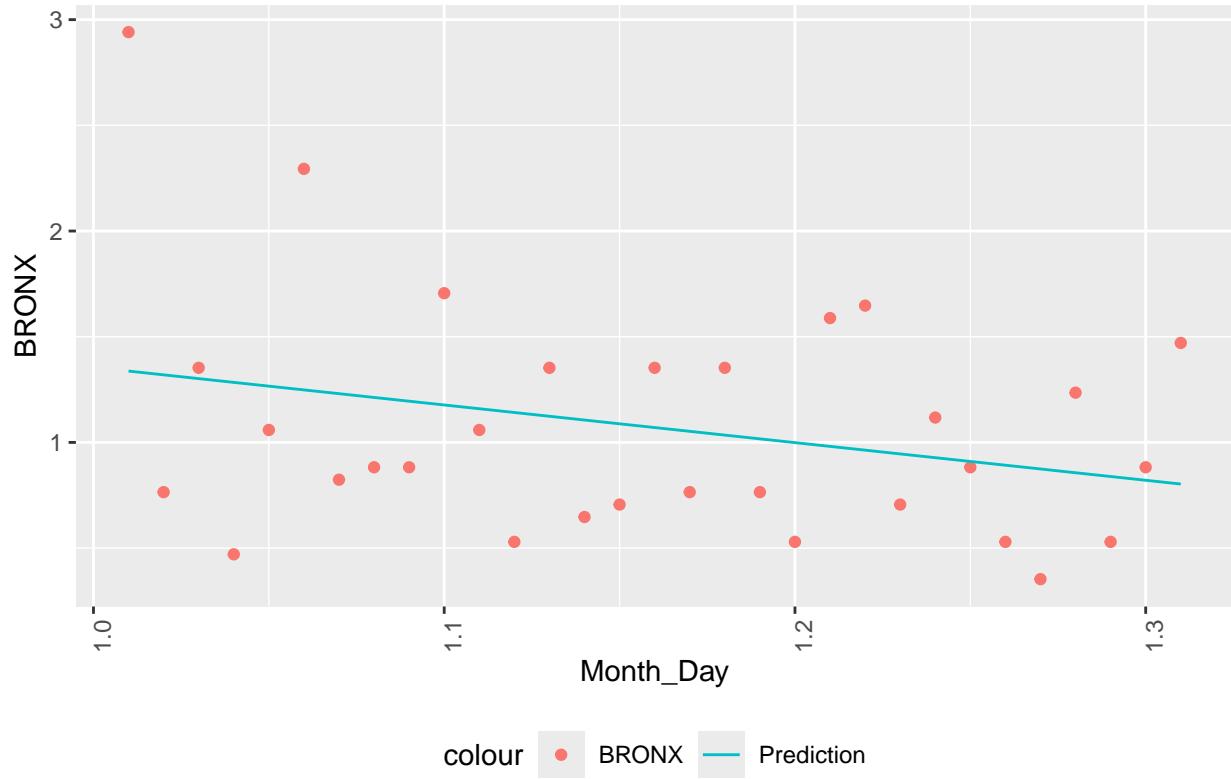


Taking the average per year hasn't made a large difference in the trend of spiking at midyear. Let's look at the Bronx per month again, using the averages, instead.

```
n <- 1
temp1 <- temp %>%
  filter(OCCUR_MONTH==1)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 1) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "January Shooting Averages by Date in Bronx, New York City")
```

January Shooting Averages by Date in Bronx, New York City

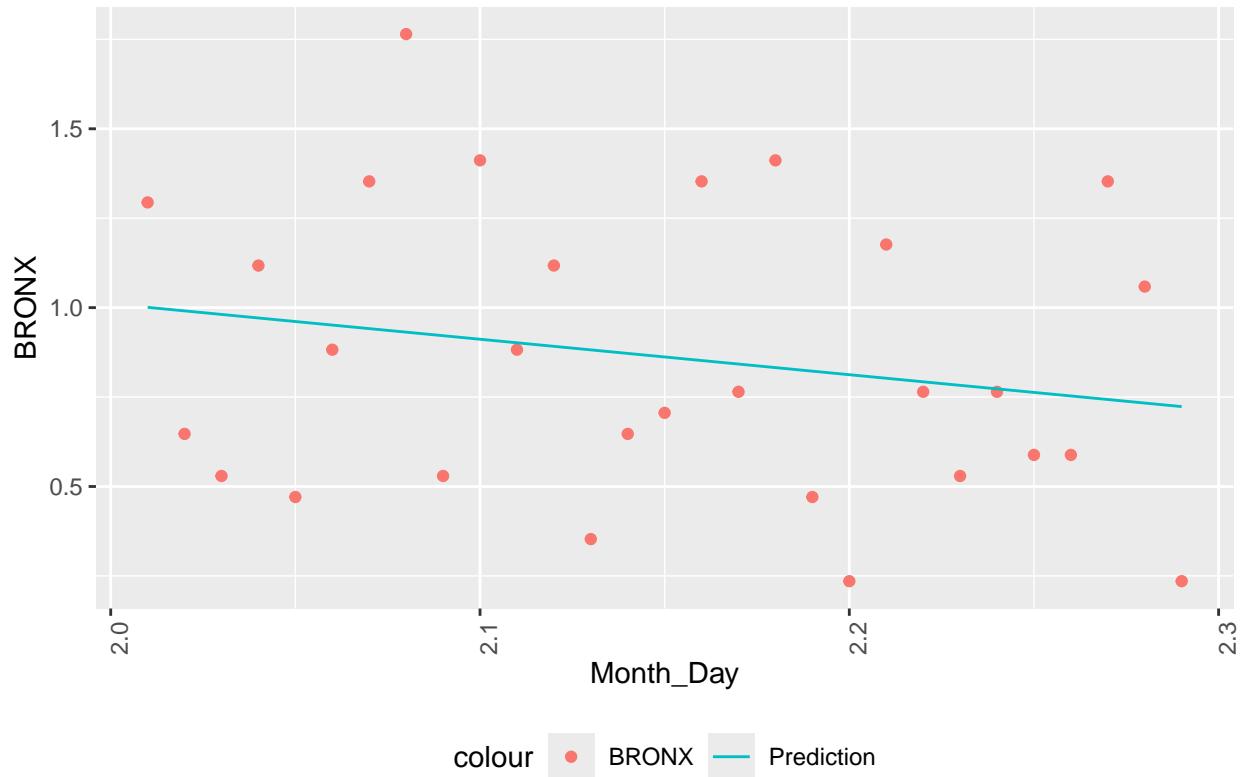


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==2)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 2) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "February Shooting Averages by Date in Bronx, New York City")
```

February Shooting Averages by Date in Bronx, New York City

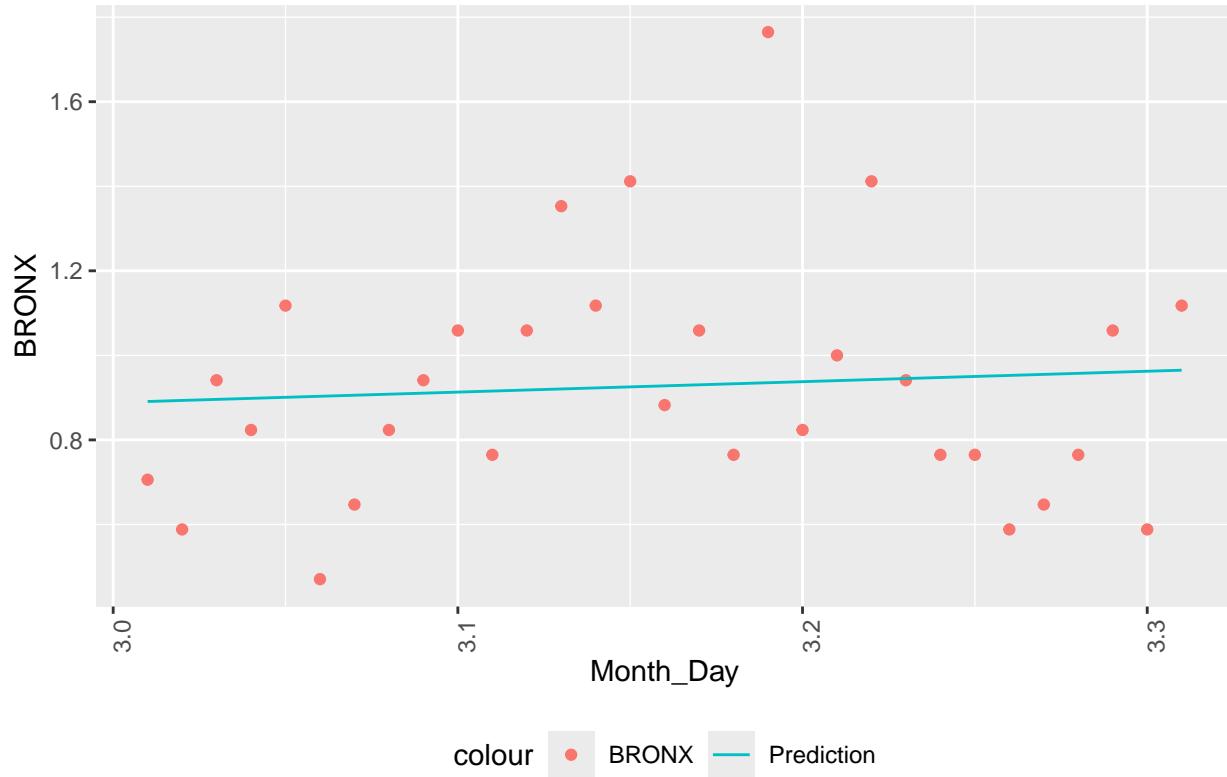


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==3)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 3) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "March Shooting Averages by Date in Bronx, New York City")
```

March Shooting Averages by Date in Bronx, New York City

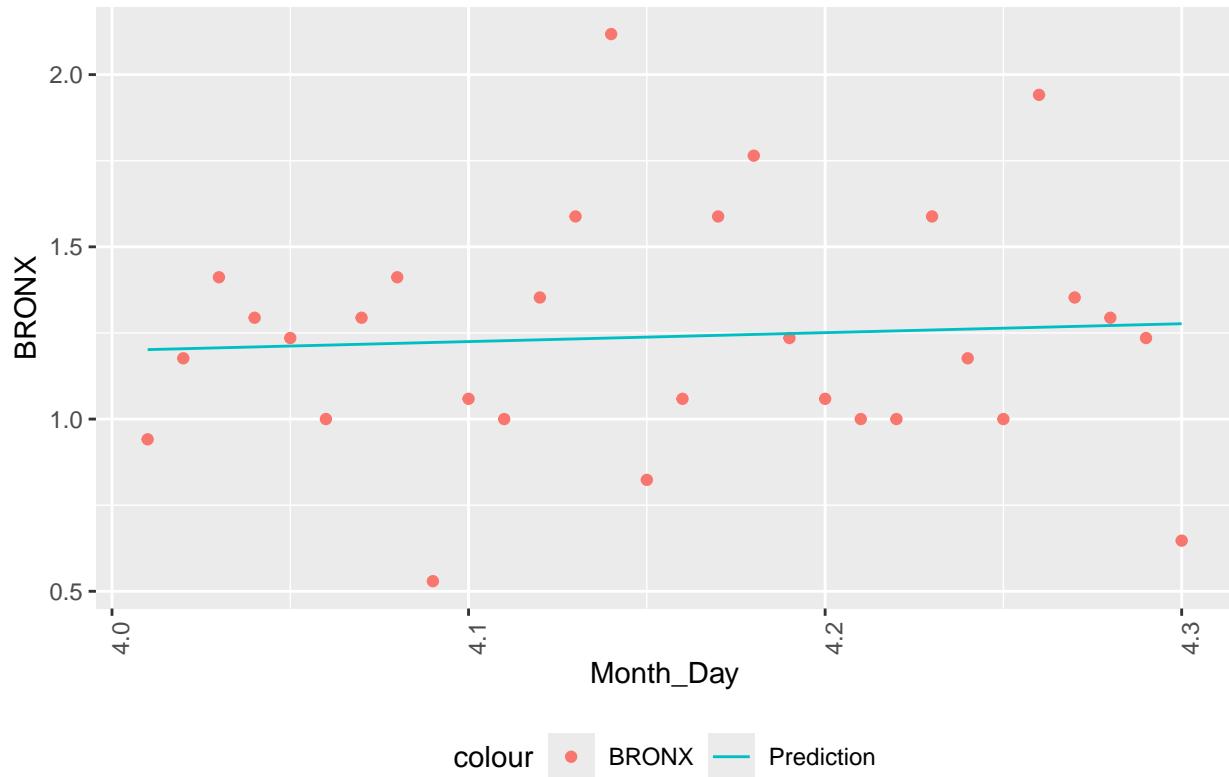


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==4)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 4) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "April Shooting Averages by Date in Bronx, New York City")
```

April Shooting Averages by Date in Bronx, New York City

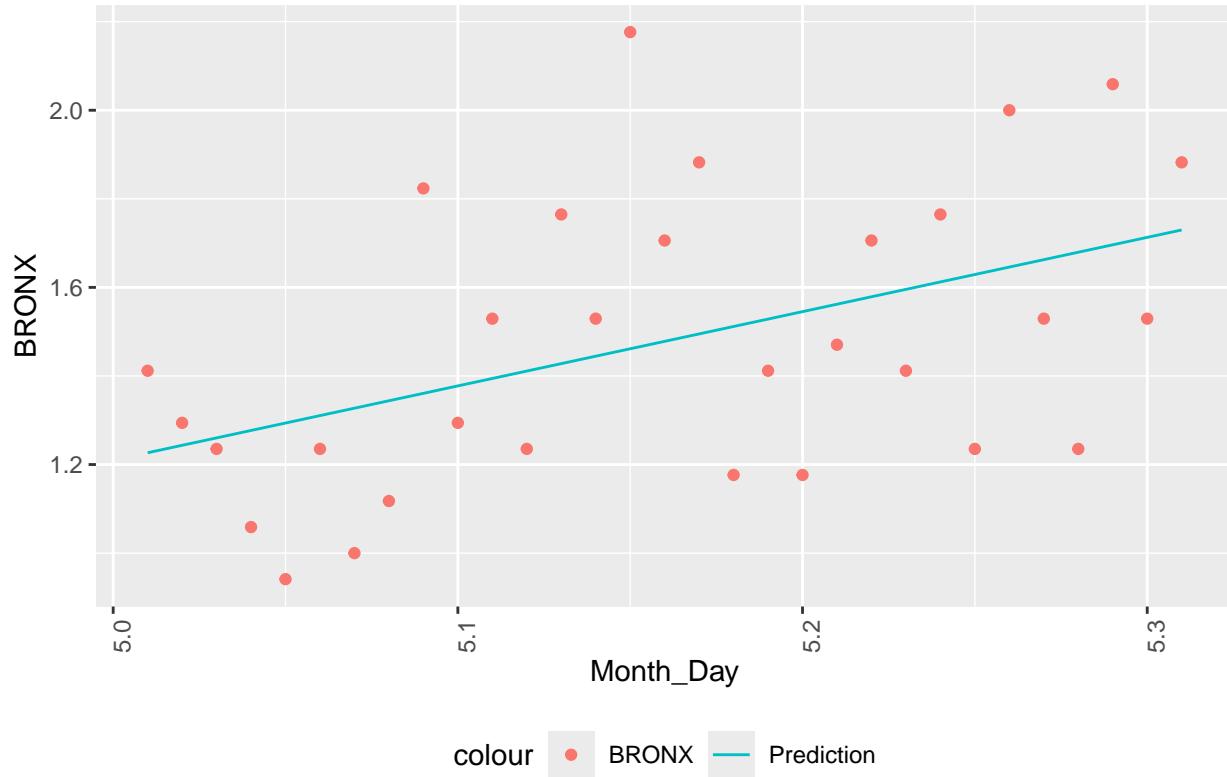


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==5)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 5) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "May Shooting Averages by Date in Bronx, New York City")
```

May Shooting Averages by Date in Bronx, New York City

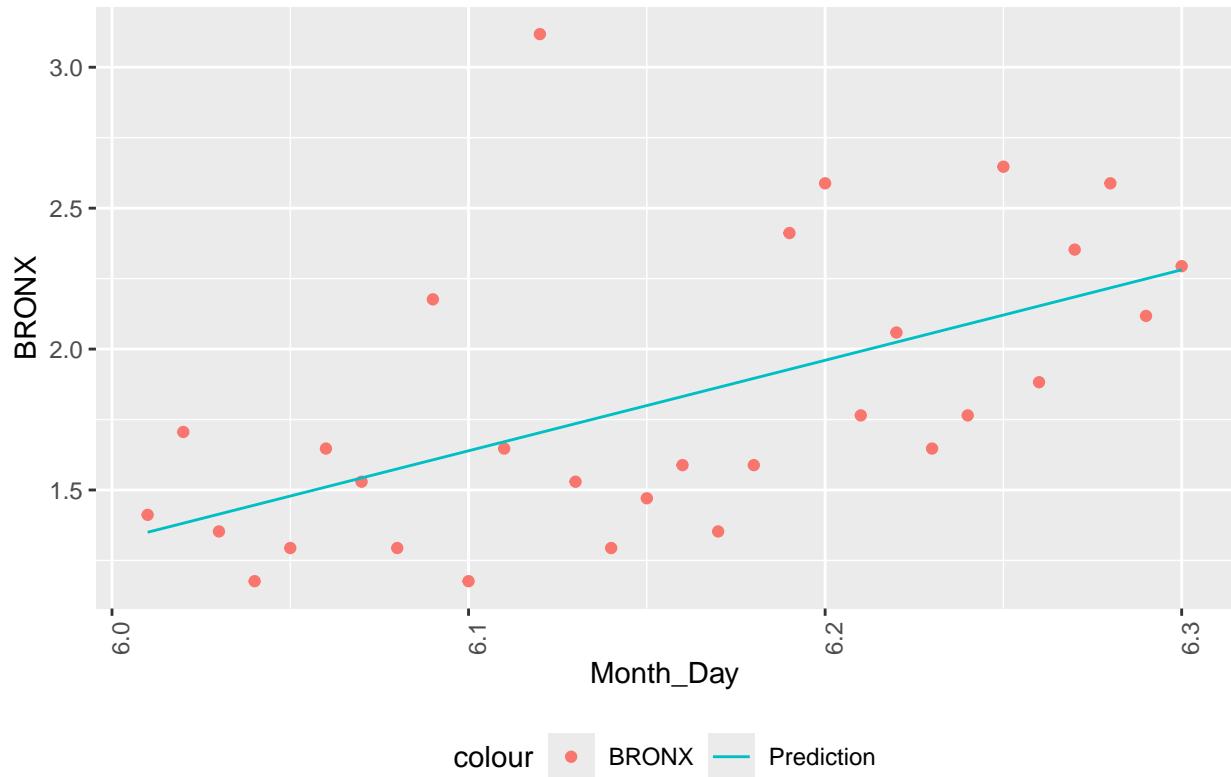


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==6)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 6) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "June Shooting Averages by Date in Bronx, New York City")
```

June Shooting Averages by Date in Bronx, New York City

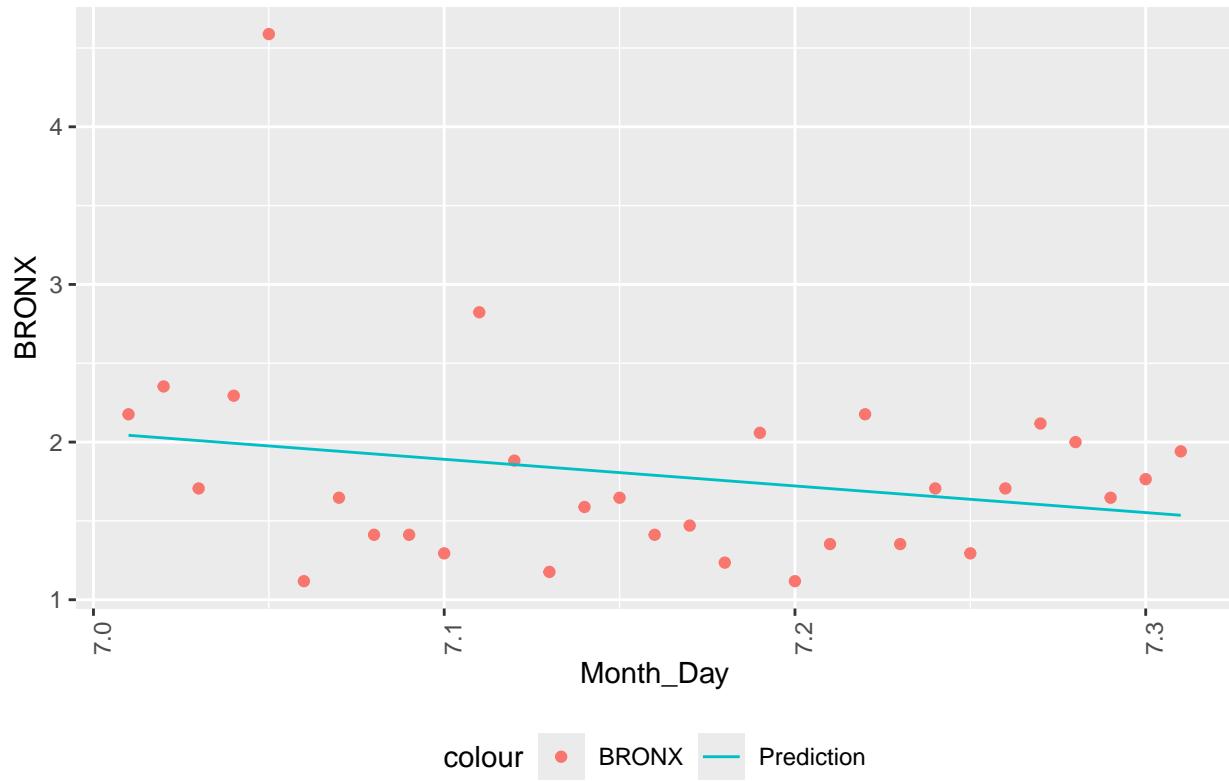


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==7)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 7) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "July Shooting Averages by Date in Bronx, New York City")
```

July Shooting Averages by Date in Bronx, New York City

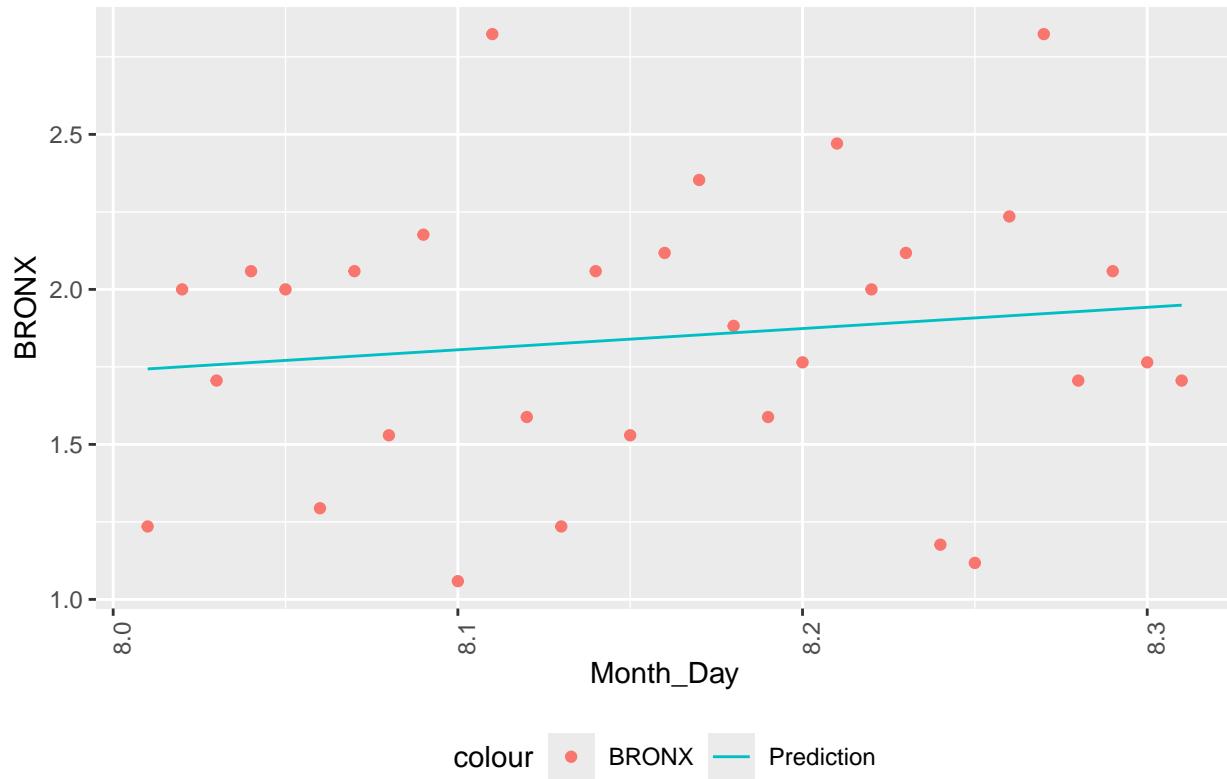


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==8)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 8) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "August Shooting Averages by Date in Bronx, New York City")
```

August Shooting Averages by Date in Bronx, New York City

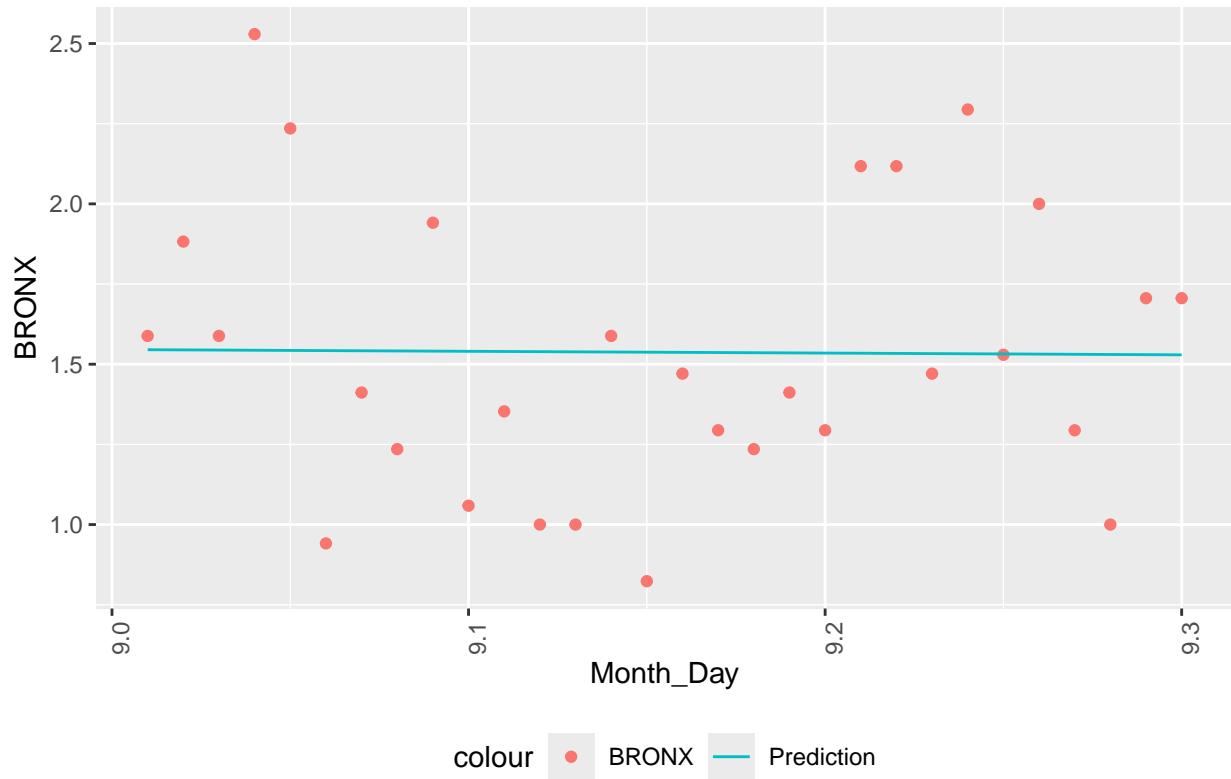


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==9)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 9) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "September Shooting Averages by Date in Bronx, New York City")
```

September Shooting Averages by Date in Bronx, New York City

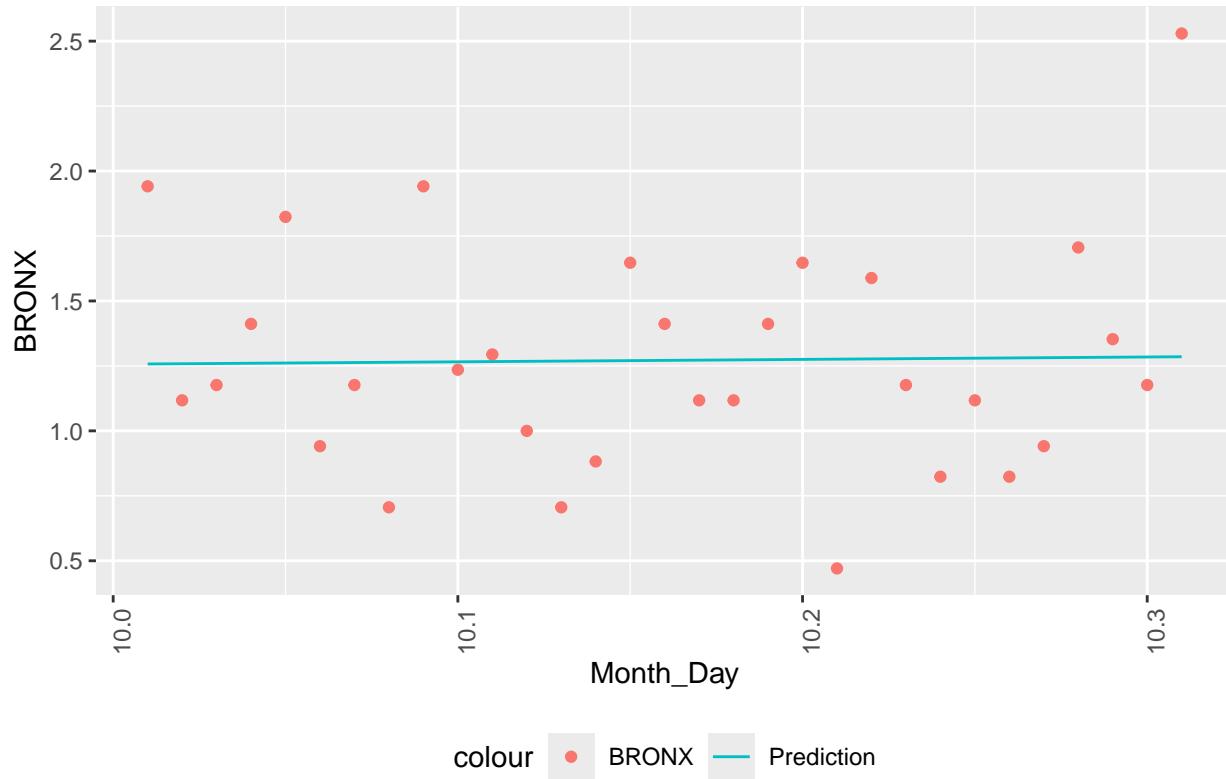


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==10)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 10) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "October Shooting Averages by Date in Bronx, New York City")
```

October Shooting Averages by Date in Bronx, New York City

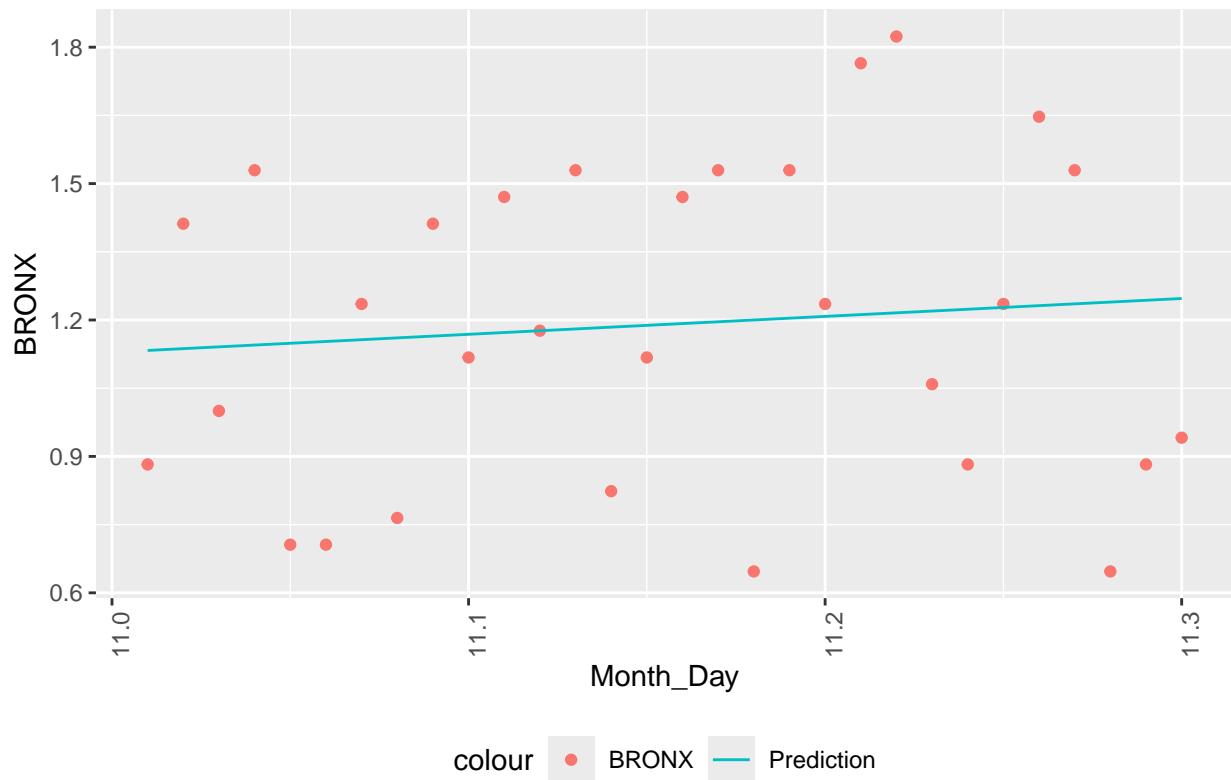


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==11)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 11) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "November Shooting Averages by Date in Bronx, New York City")
```

November Shooting Averages by Date in Bronx, New York City

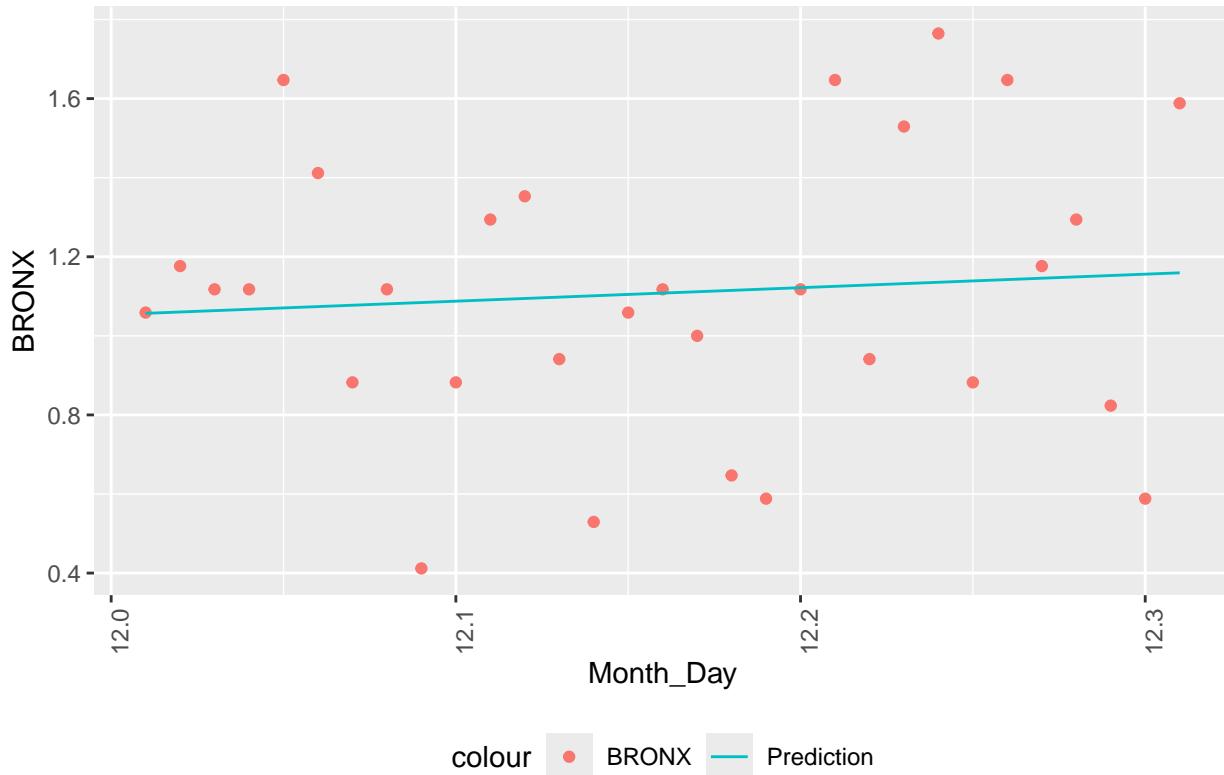


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==12)
mod <- lm(BRONX ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 12) %>%
  ggplot(aes(x = Month_Day, y = BRONX)) +
  geom_point(aes(color = "BRONX")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "December Shooting Averages by Date in Bronx, New York City")
```

December Shooting Averages by Date in Bronx, New York City



```
#creates a scatterplot of the shootings over a month, daily, in the selected borough
```

Results:

- January: downward trend
- February: downward trend
- March: slight upward trend
- April: slight upward trend
- May: strong upward trend
- June: strong upward trend
- July: downward trend
- August: upward trend
- September: slight downward trend
- October: slight upward trend
- November: slight upward trend
- December: slight upward trend

The results are nearly identical using totals or averages. Let's look at the other boroughs using averages.

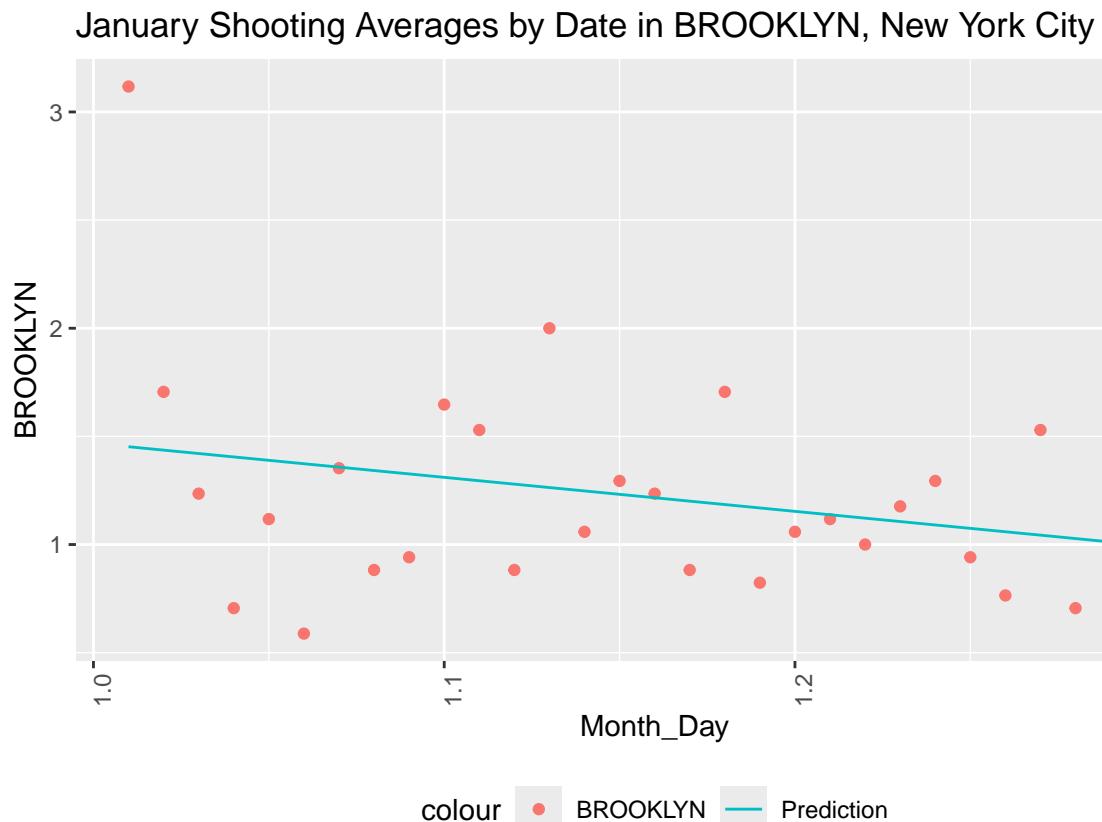
We'll start with Brooklyn.

```
n <- 1
temp1 <- temp %>%
  filter(OCCUR_MONTH==1)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month
```

```

temp1 %>%
  filter(OCCUR_MONTH == 1) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "January Shooting Averages by Date in BROOKLYN, New York City")

```



```

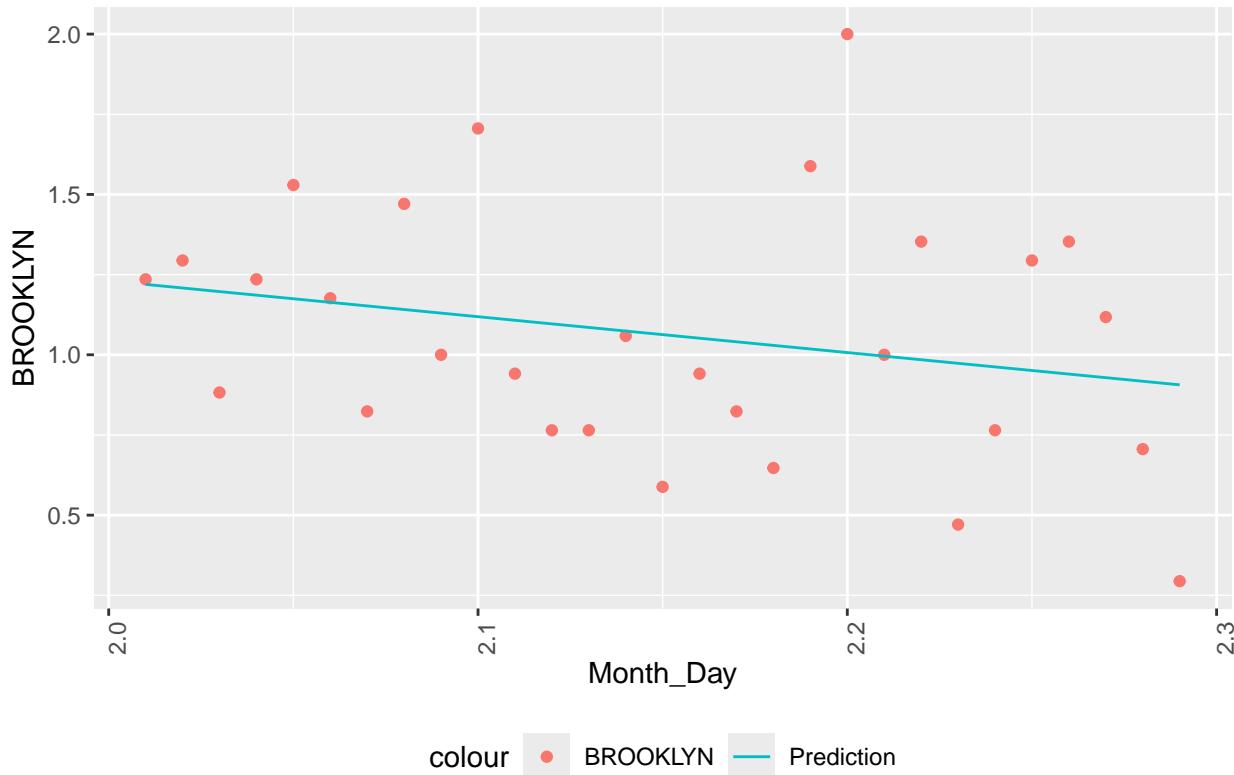
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==2)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 2) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "February Shooting Averages by Date in BROOKLYN, New York City")

```

February Shooting Averages by Date in BROOKLYN, New York City

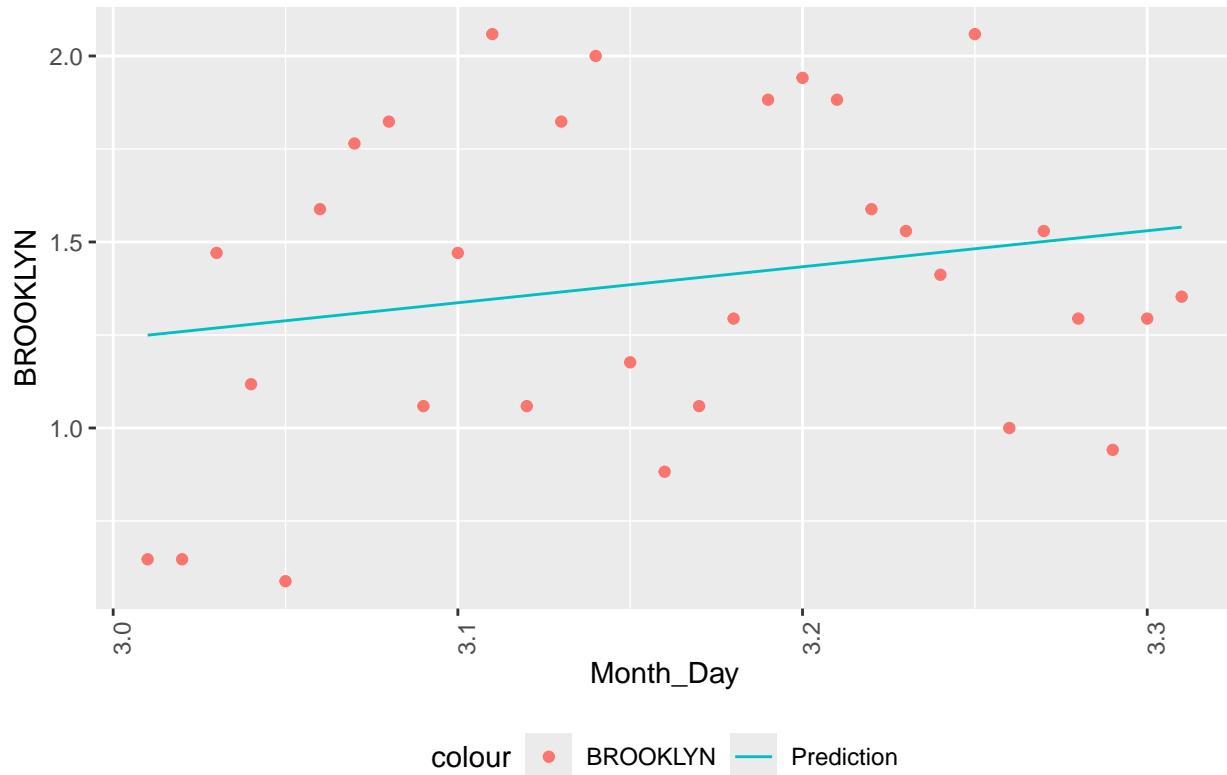


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==3)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 3) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "March Shooting Averages by Date in BROOKLYN, New York City")
```

March Shooting Averages by Date in BROOKLYN, New York City

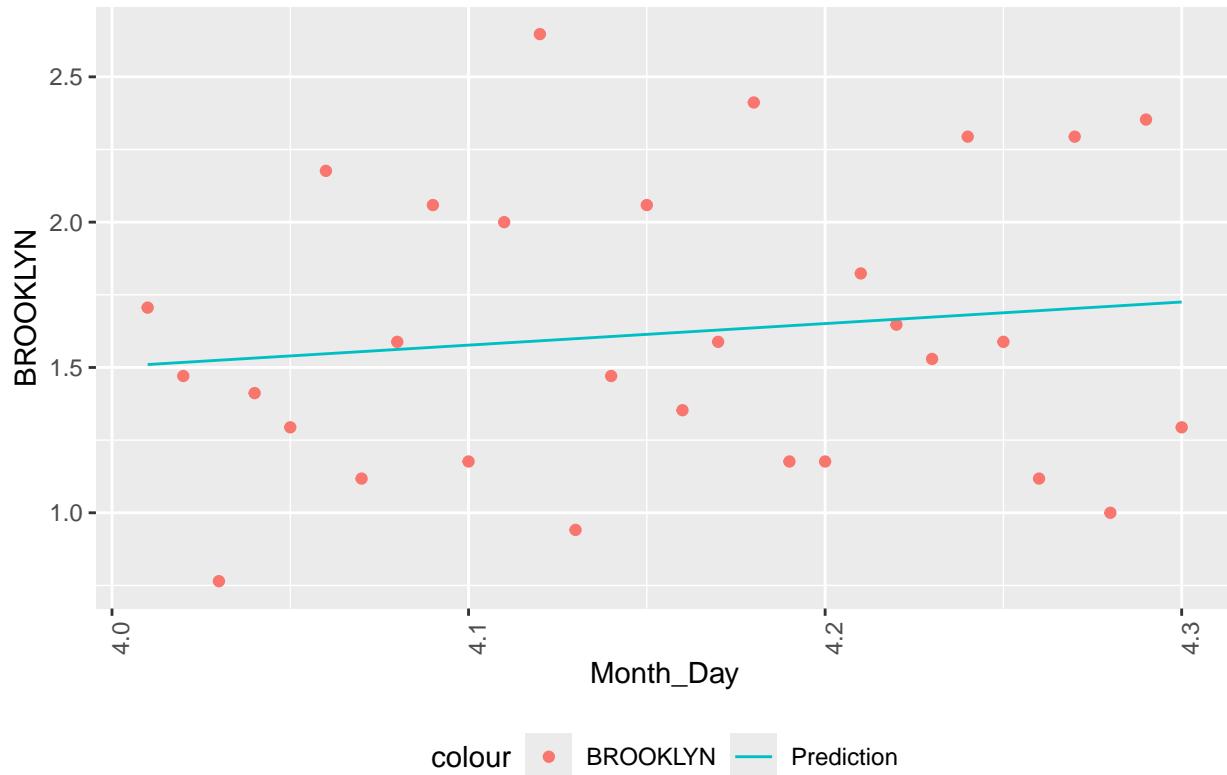


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==4)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 4) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "April Shooting Averages by Date in BROOKLYN, New York City")
```

April Shooting Averages by Date in BROOKLYN, New York City

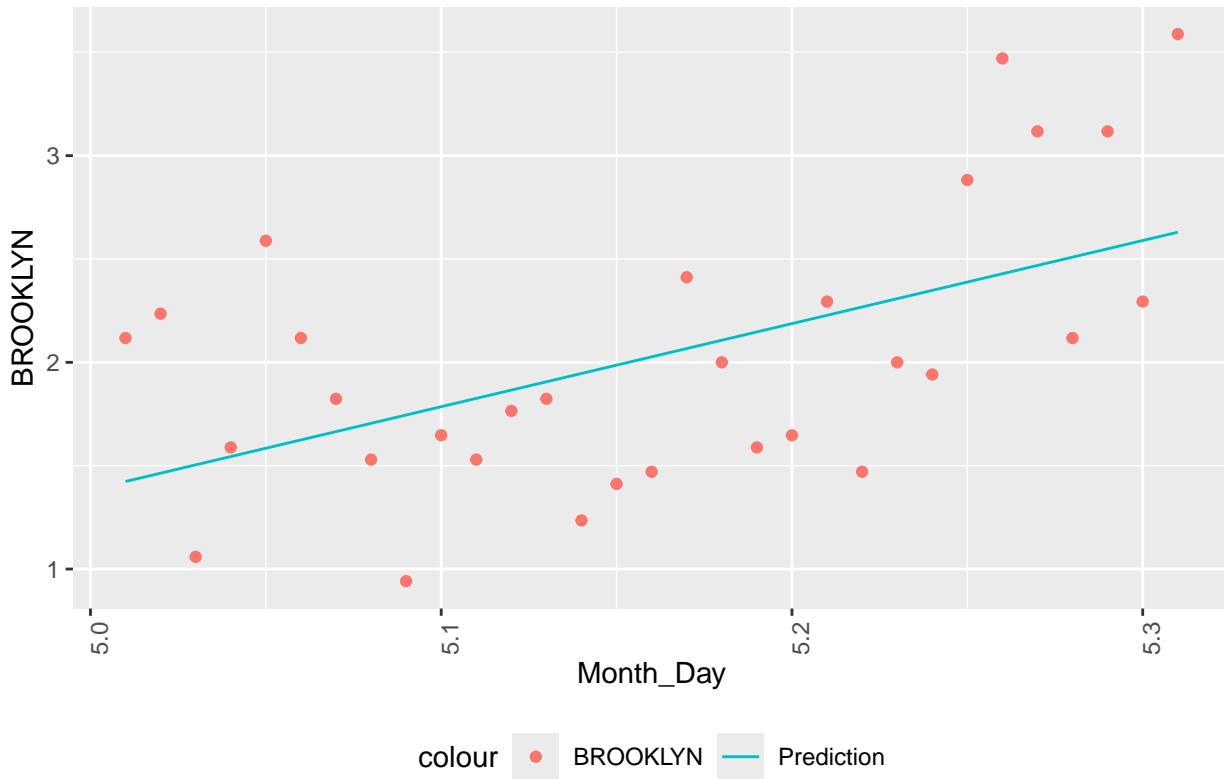


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==5)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 5) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "May Shooting Averages by Date in BROOKLYN, New York City")
```

May Shooting Averages by Date in BROOKLYN, New York City

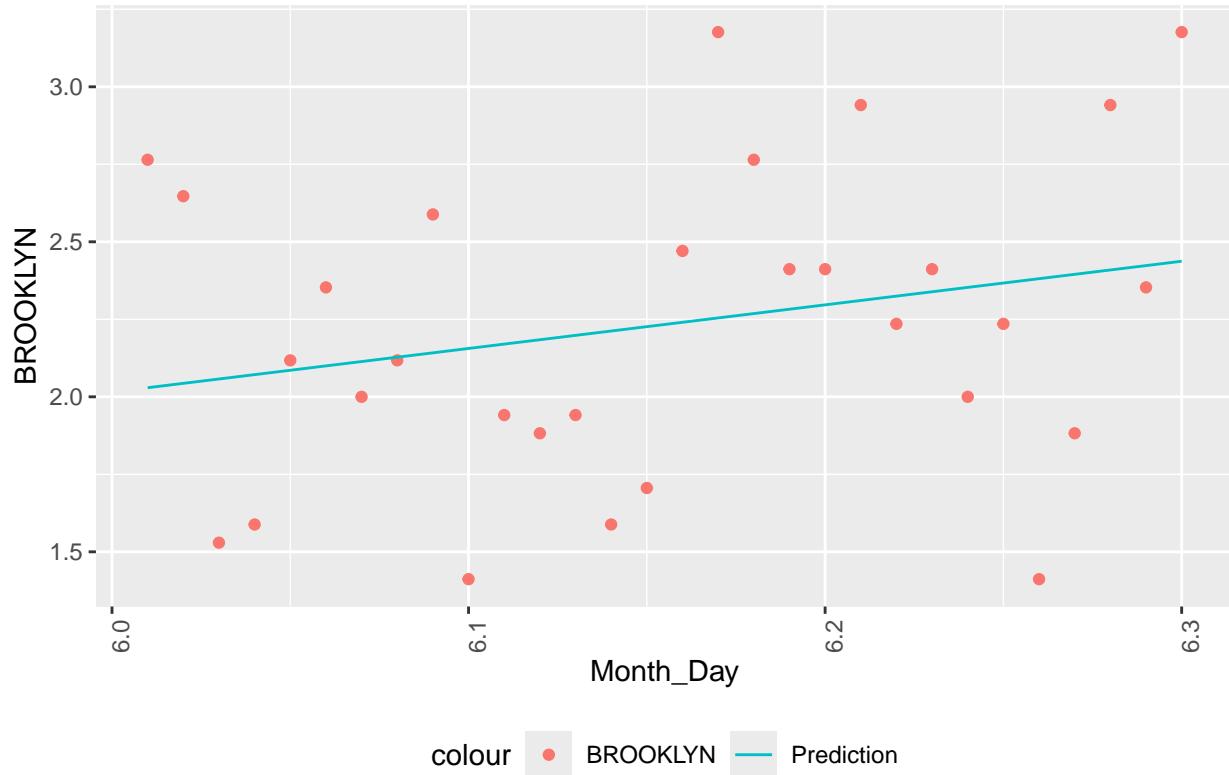


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==6)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 6) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "June Shooting Averages by Date in BROOKLYN, New York City")
```

June Shooting Averages by Date in BROOKLYN, New York City

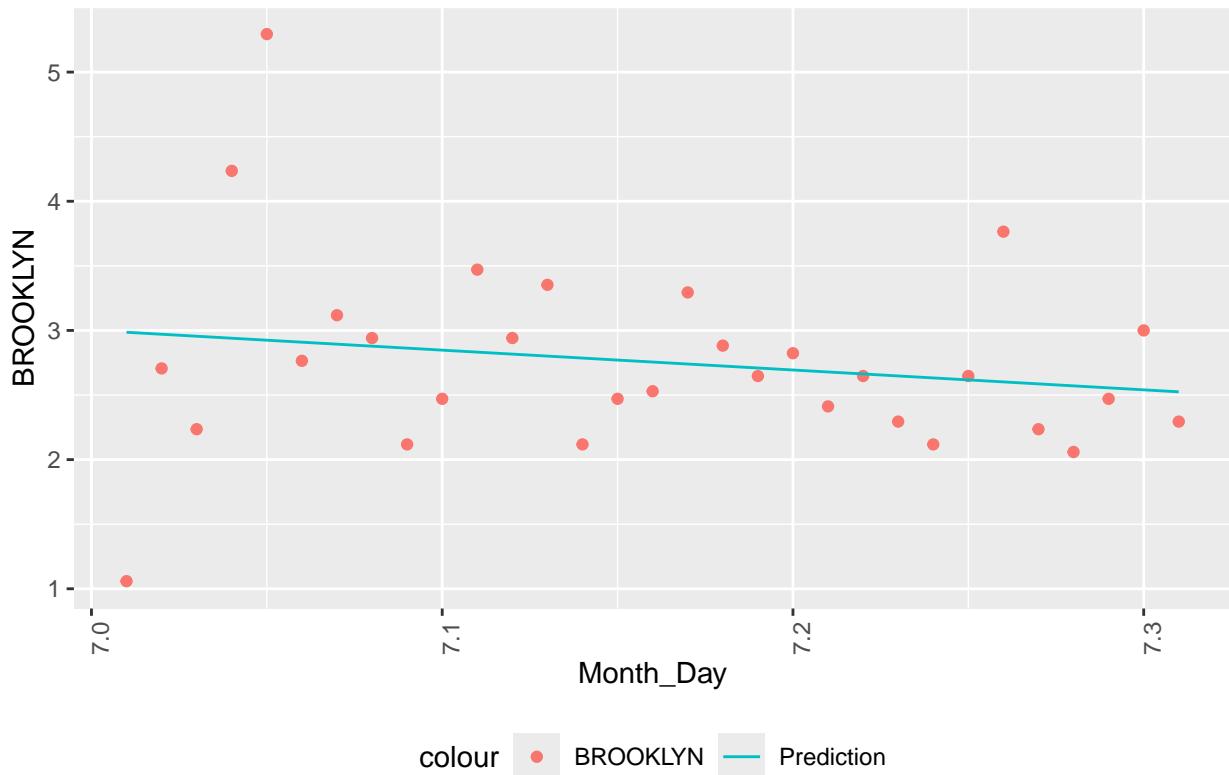


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==7)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 7) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "July Shooting Averages by Date in BROOKLYN, New York City")
```

July Shooting Averages by Date in BROOKLYN, New York City

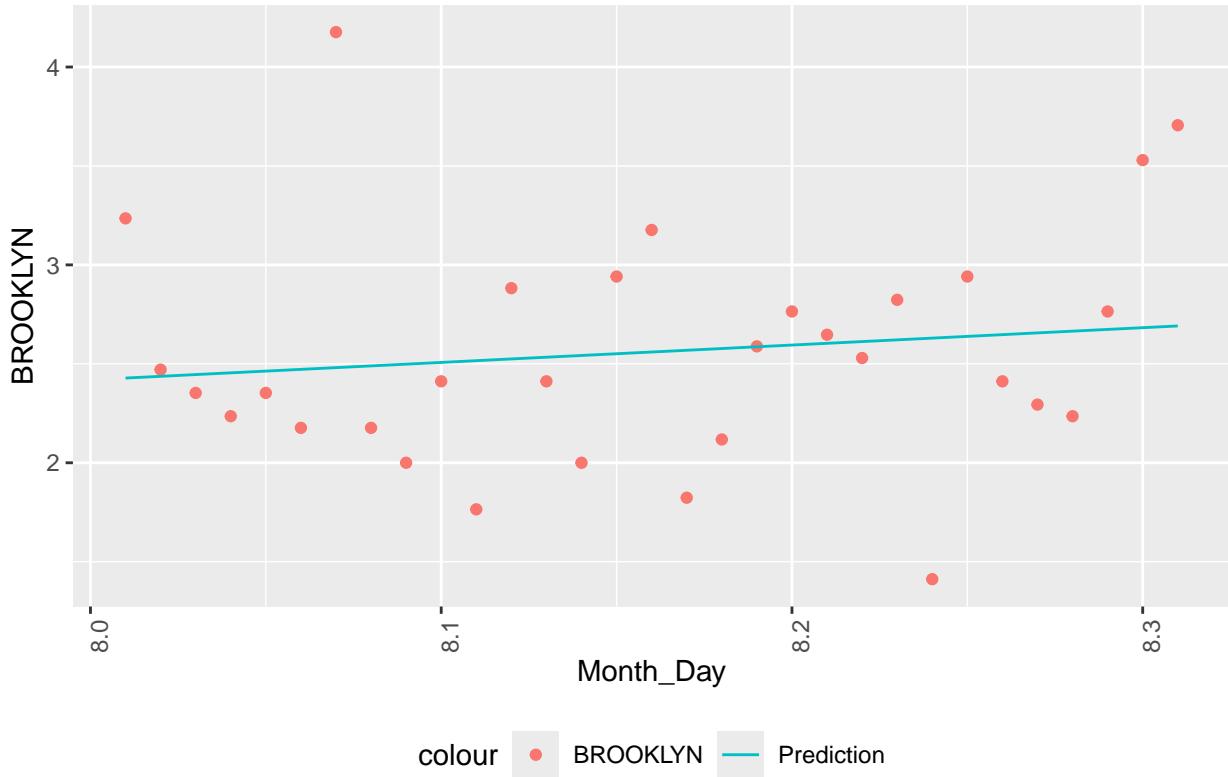


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==8)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 8) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "August Shooting Averages by Date in BROOKLYN, New York City")
```

August Shooting Averages by Date in BROOKLYN, New York City

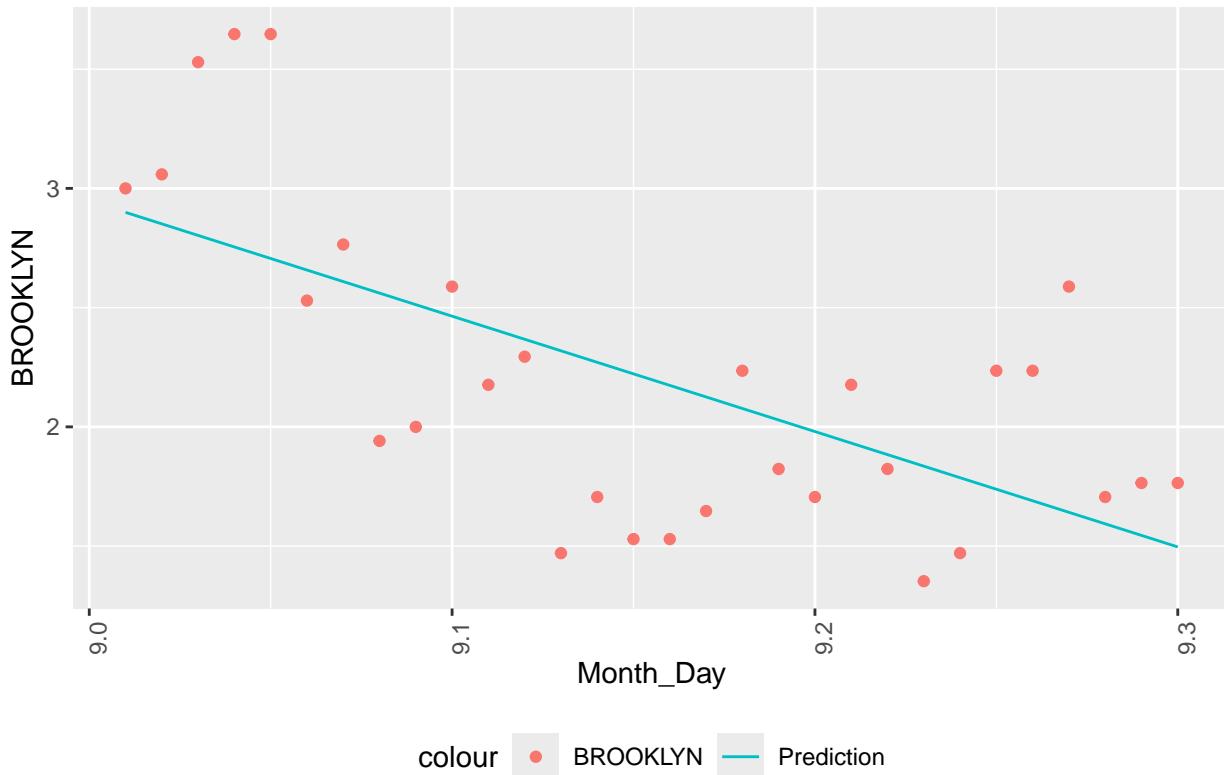


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==9)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 9) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "September Shooting Averages by Date in BROOKLYN, New York City")
```

September Shooting Averages by Date in BROOKLYN, New York City

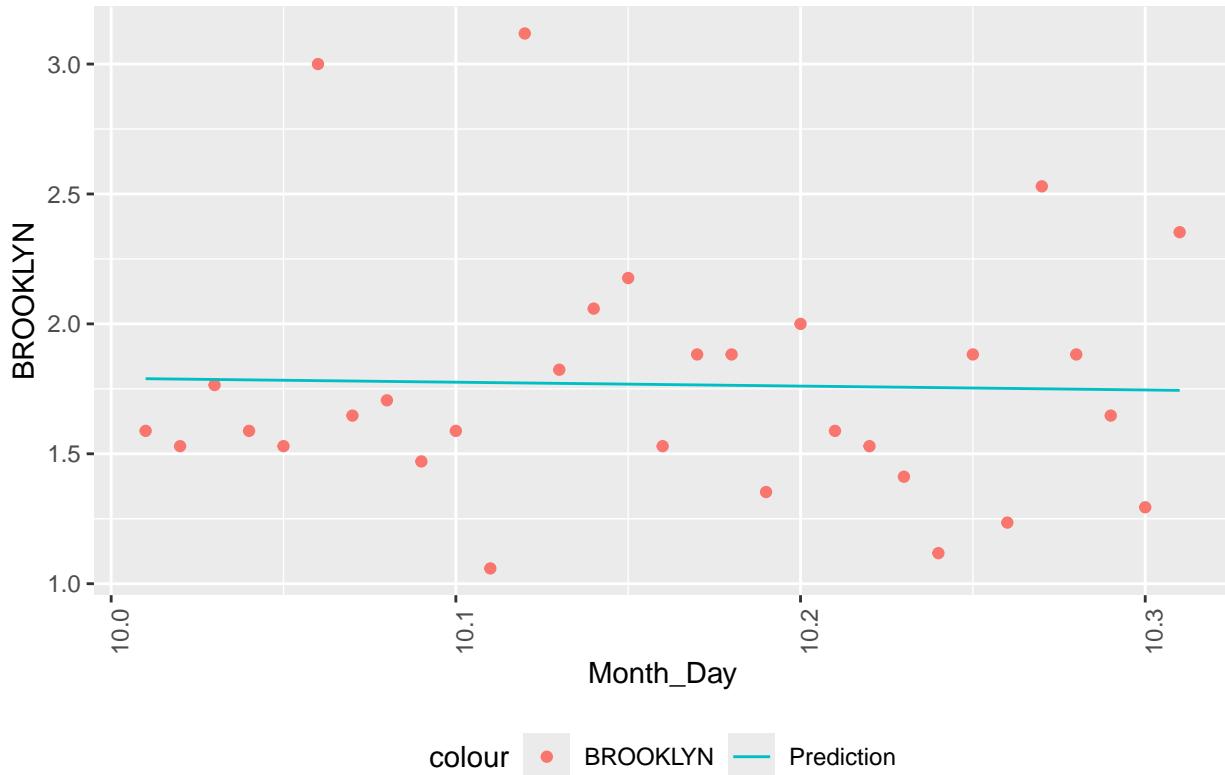


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==10)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 10) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "October Shooting Averages by Date in BROOKLYN, New York City")
```

October Shooting Averages by Date in BROOKLYN, New York City

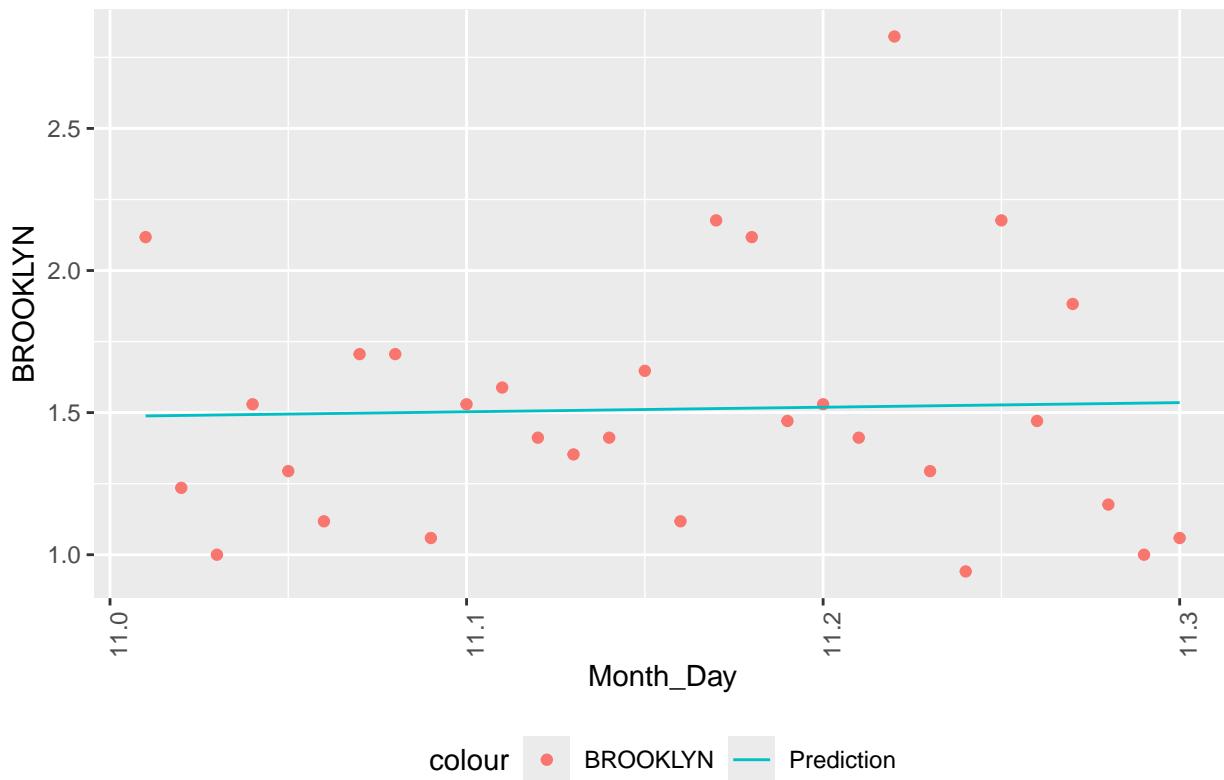


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==11)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 11) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "November Shooting Averages by Date in BROOKLYN, New York City")
```

November Shooting Averages by Date in BROOKLYN, New York City

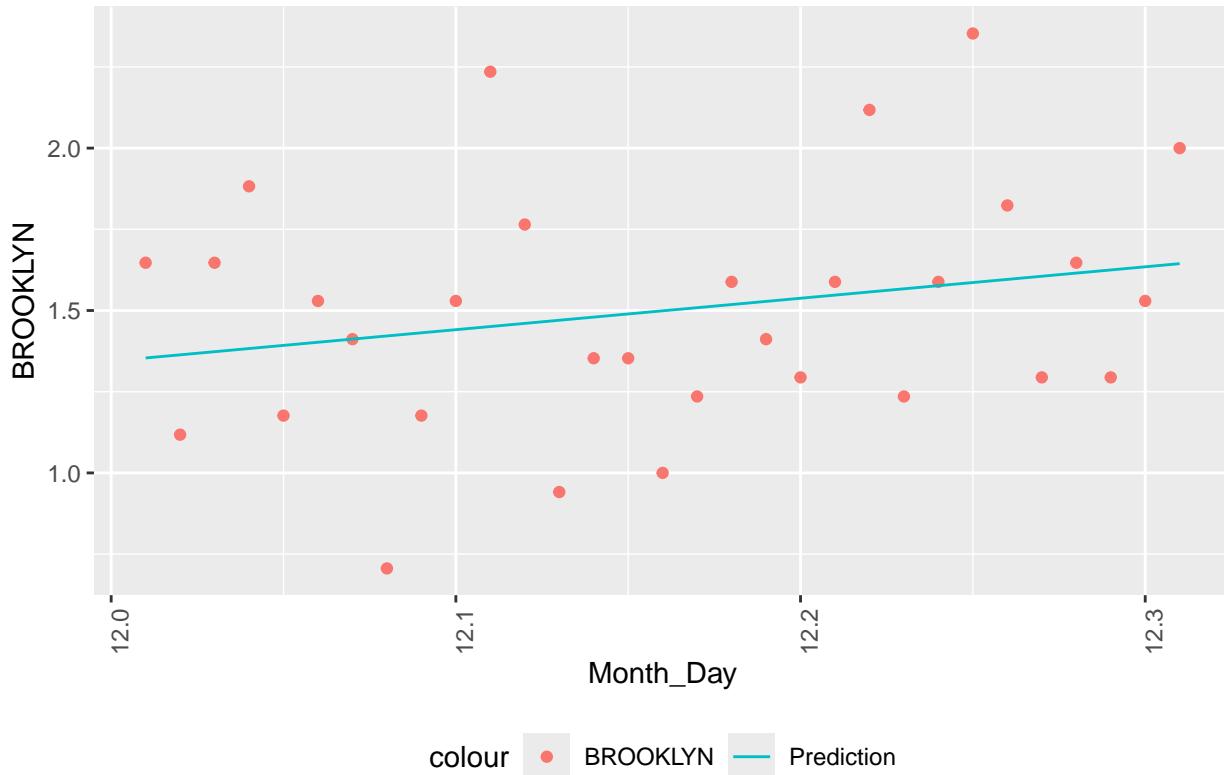


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==12)
mod <- lm(BROOKLYN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 12) %>%
  ggplot(aes(x = Month_Day, y = BROOKLYN)) +
  geom_point(aes(color = "BROOKLYN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "December Shooting Averages by Date in BROOKLYN, New York City")
```

December Shooting Averages by Date in BROOKLYN, New York City



```
#creates a scatterplot of the shootings over a month, daily, in the selected borough
```

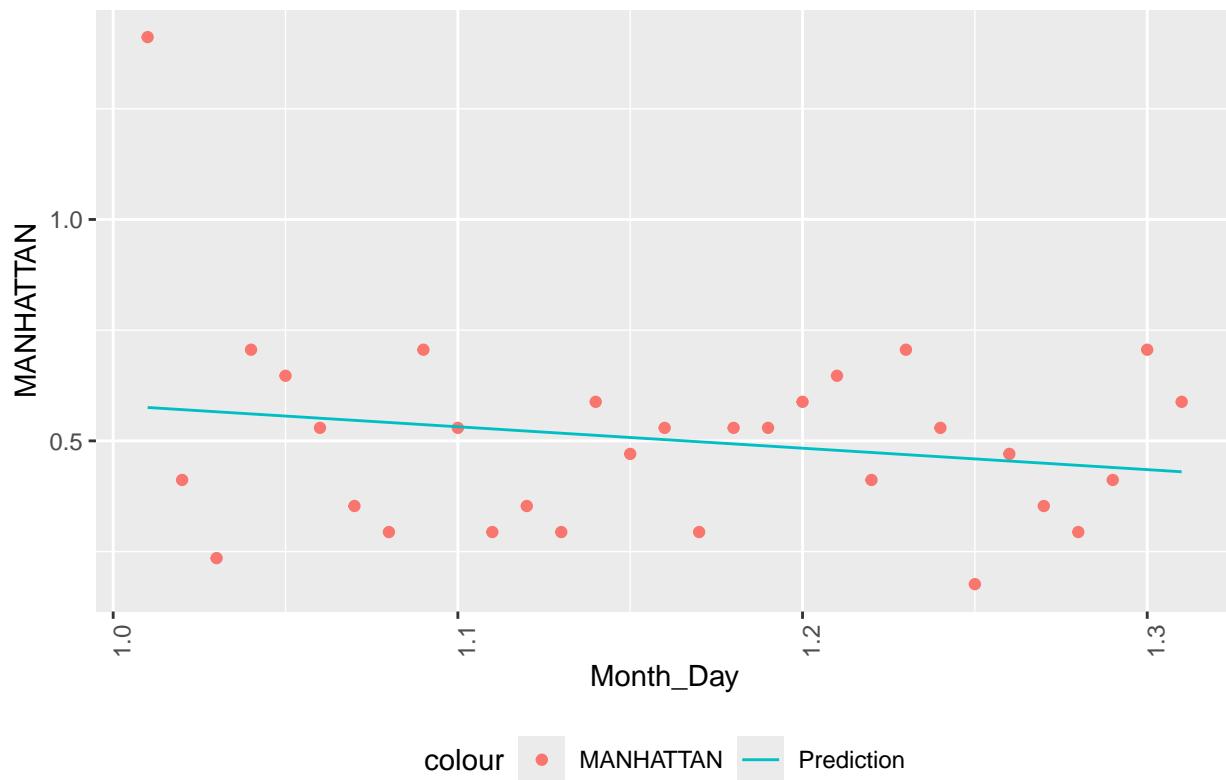
In Brooklyn, we see a weaker increase in May and June but the increasing number of shootings on average increases from March to June and eventually hits a much steeper decline in September. This still supports a midyear peak.

Next we'll look at Manhattan.

```
n <- 1
temp1 <- temp %>%
  filter(OCCUR_MONTH==1)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 1) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "January Shooting Averages by Date in MANHATTAN, New York City")
```

January Shooting Averages by Date in MANHATTAN, New York City

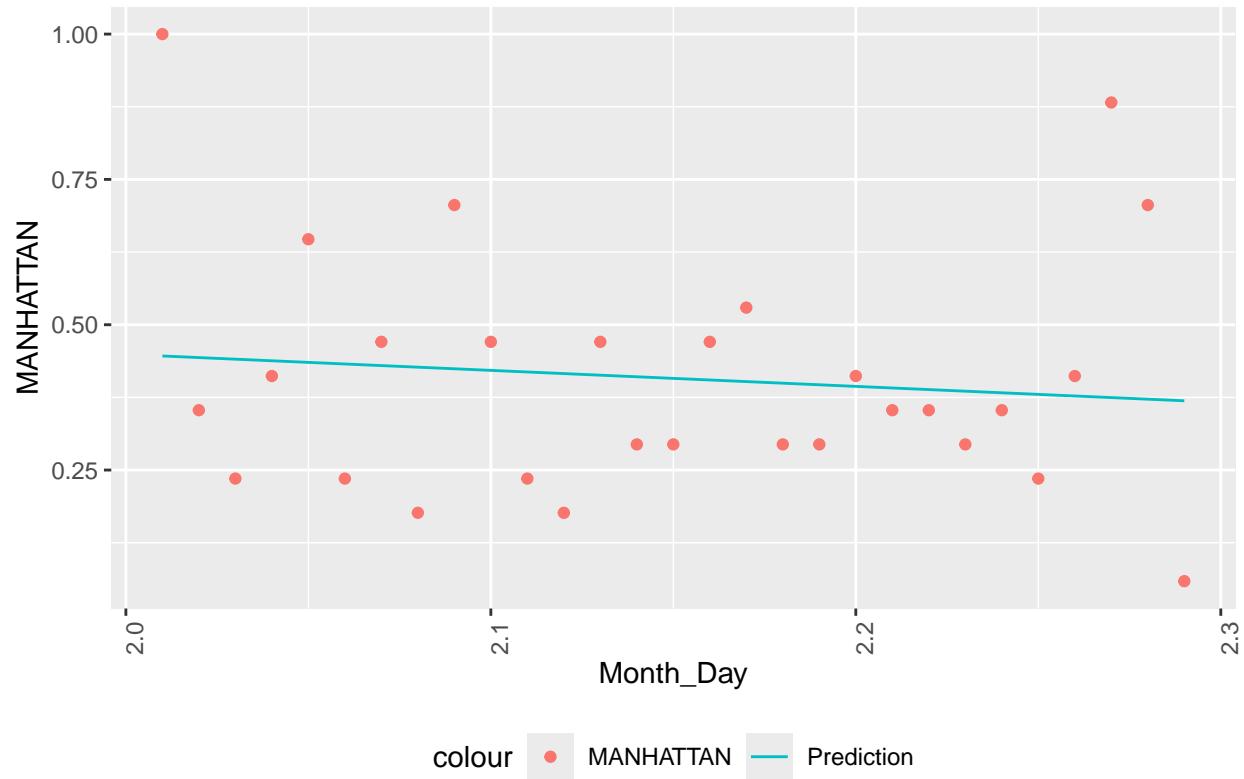


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==2)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 2) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "February Shooting Averages by Date in MANHATTAN, New York City")
```

February Shooting Averages by Date in MANHATTAN, New York City

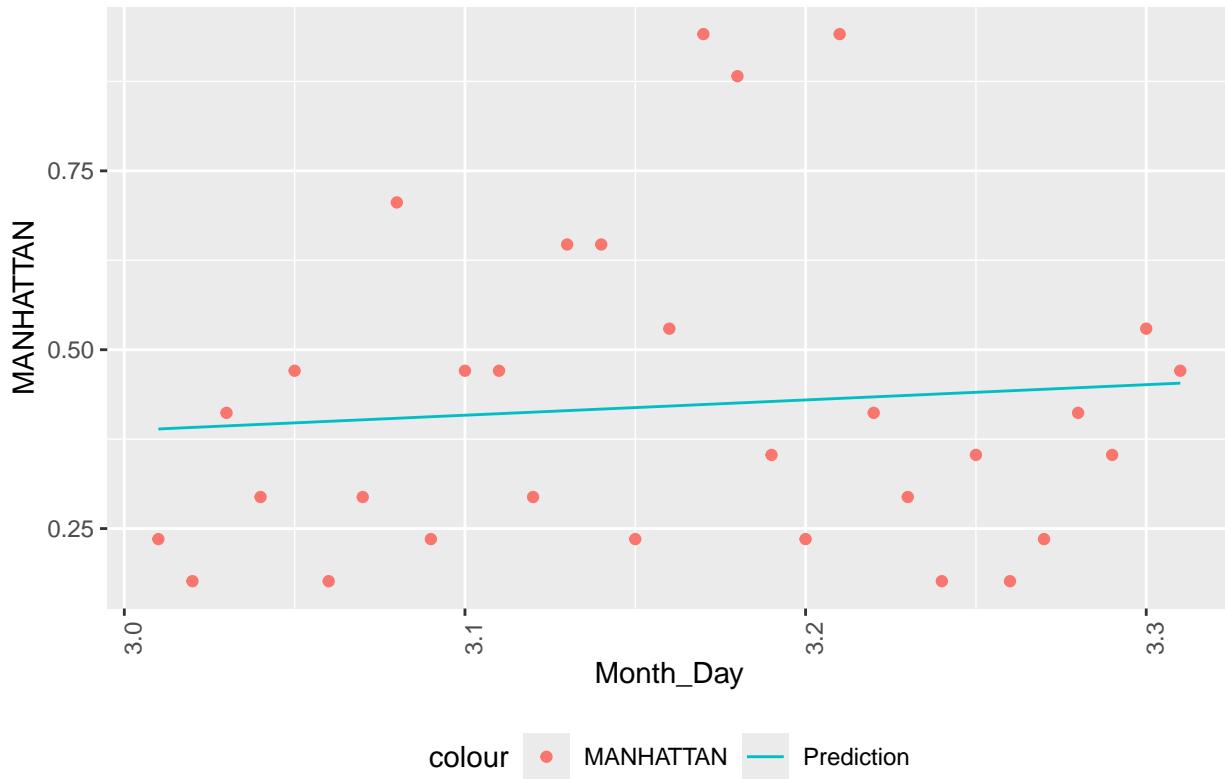


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==3)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 3) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "March Shooting Averages by Date in MANHATTAN, New York City")
```

March Shooting Averages by Date in MANHATTAN, New York City

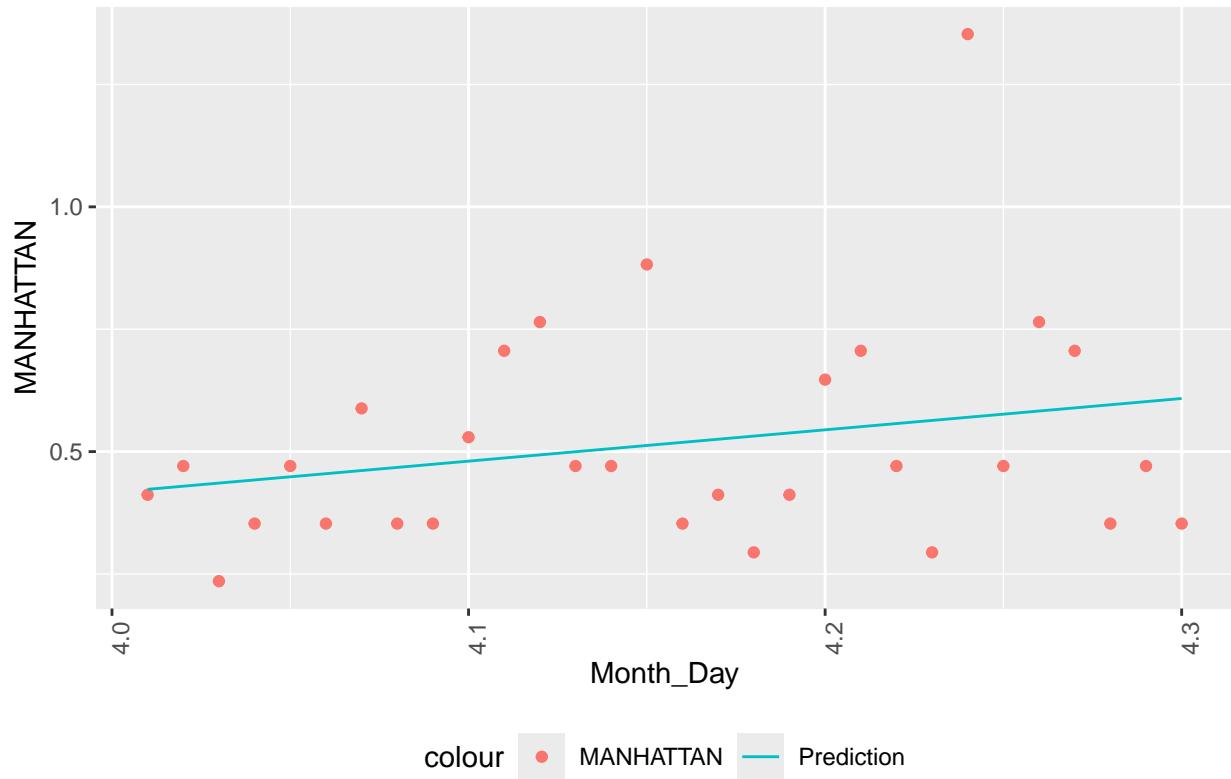


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==4)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 4) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "April Shooting Averages by Date in MANHATTAN, New York City")
```

April Shooting Averages by Date in MANHATTAN, New York City

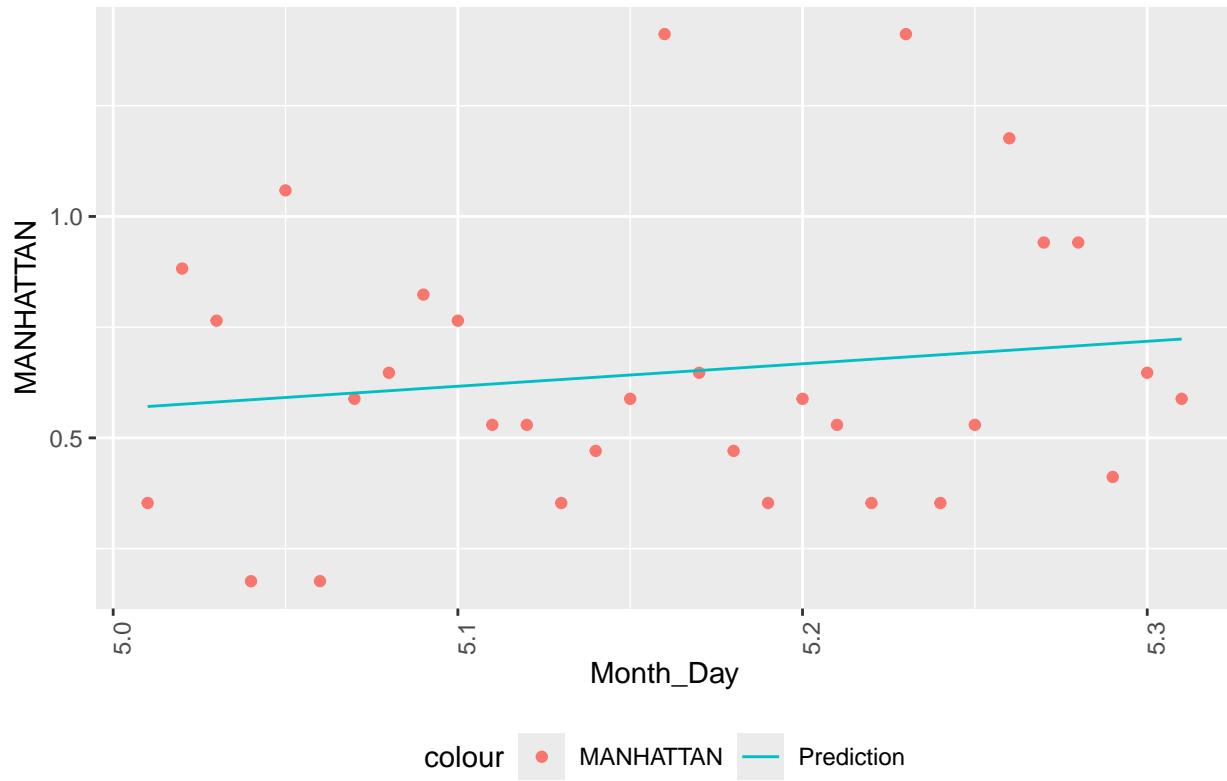


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==5)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 5) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "May Shooting Averages by Date in MANHATTAN, New York City")
```

May Shooting Averages by Date in MANHATTAN, New York City

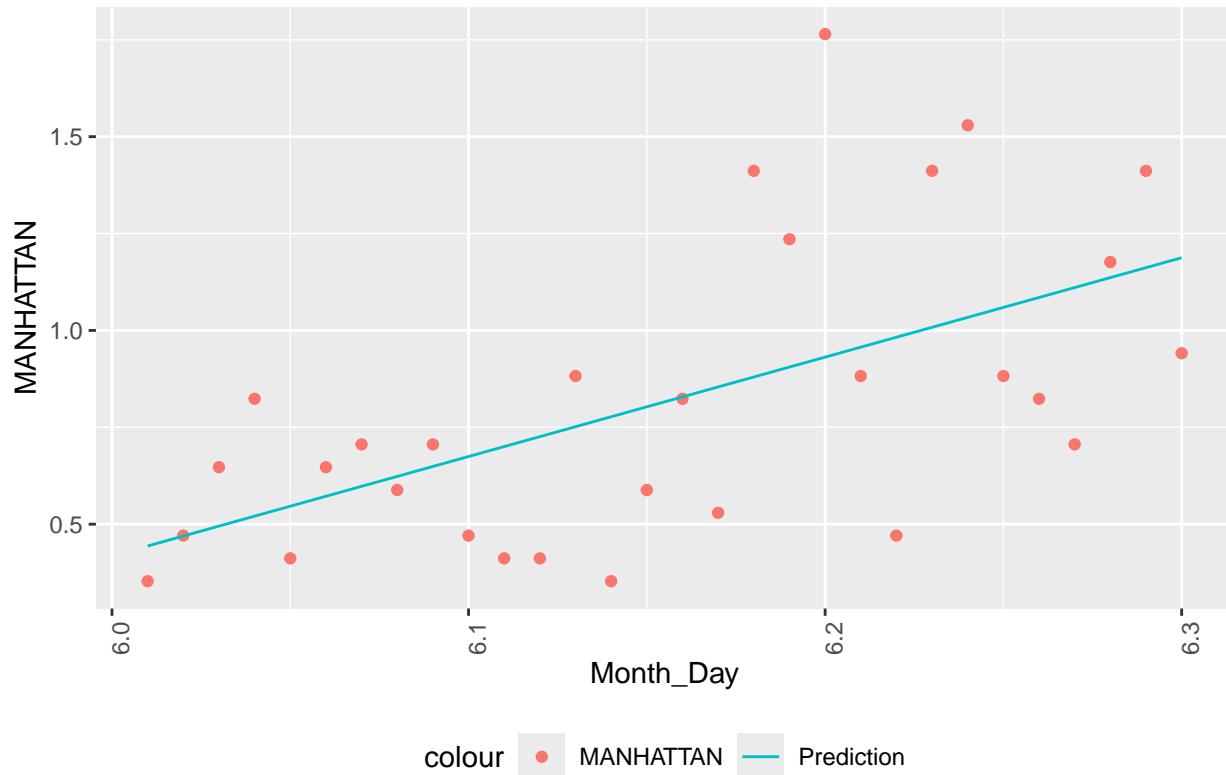


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==6)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 6) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "June Shooting Averages by Date in MANHATTAN, New York City")
```

June Shooting Averages by Date in MANHATTAN, New York City

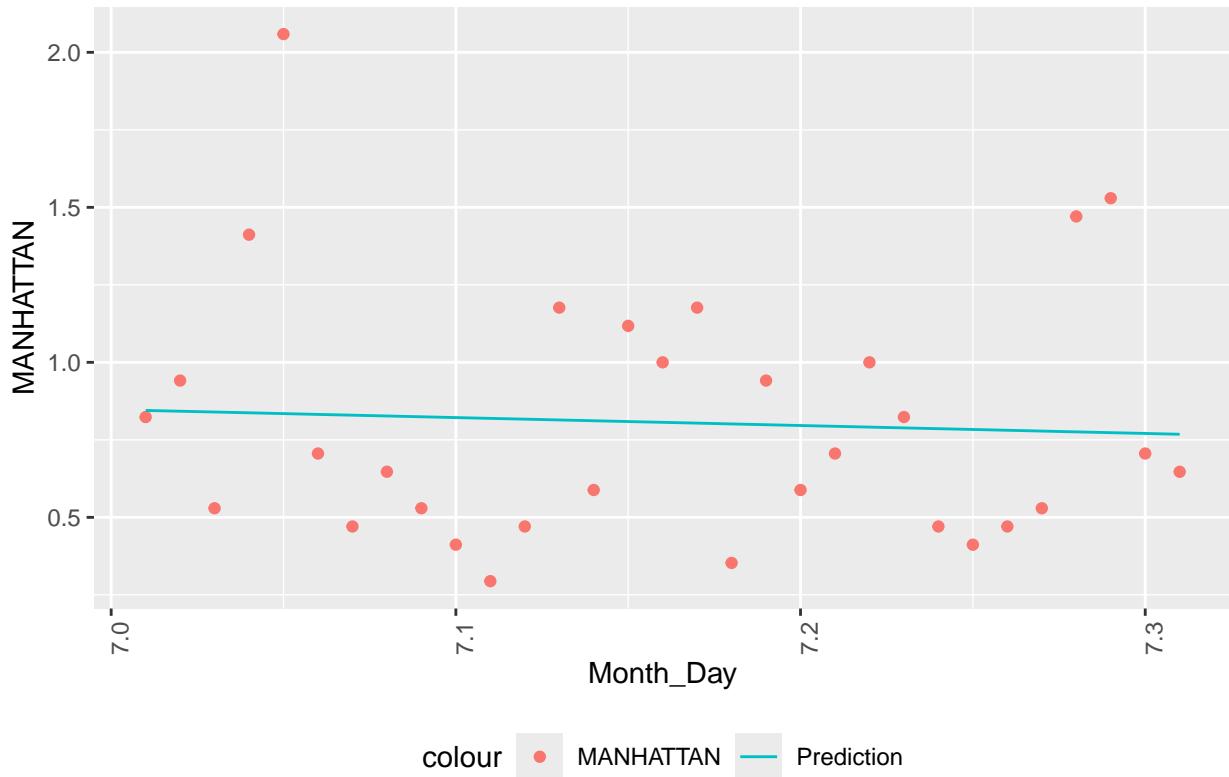


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==7)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 7) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "July Shooting Averages by Date in MANHATTAN, New York City")
```

July Shooting Averages by Date in MANHATTAN, New York City

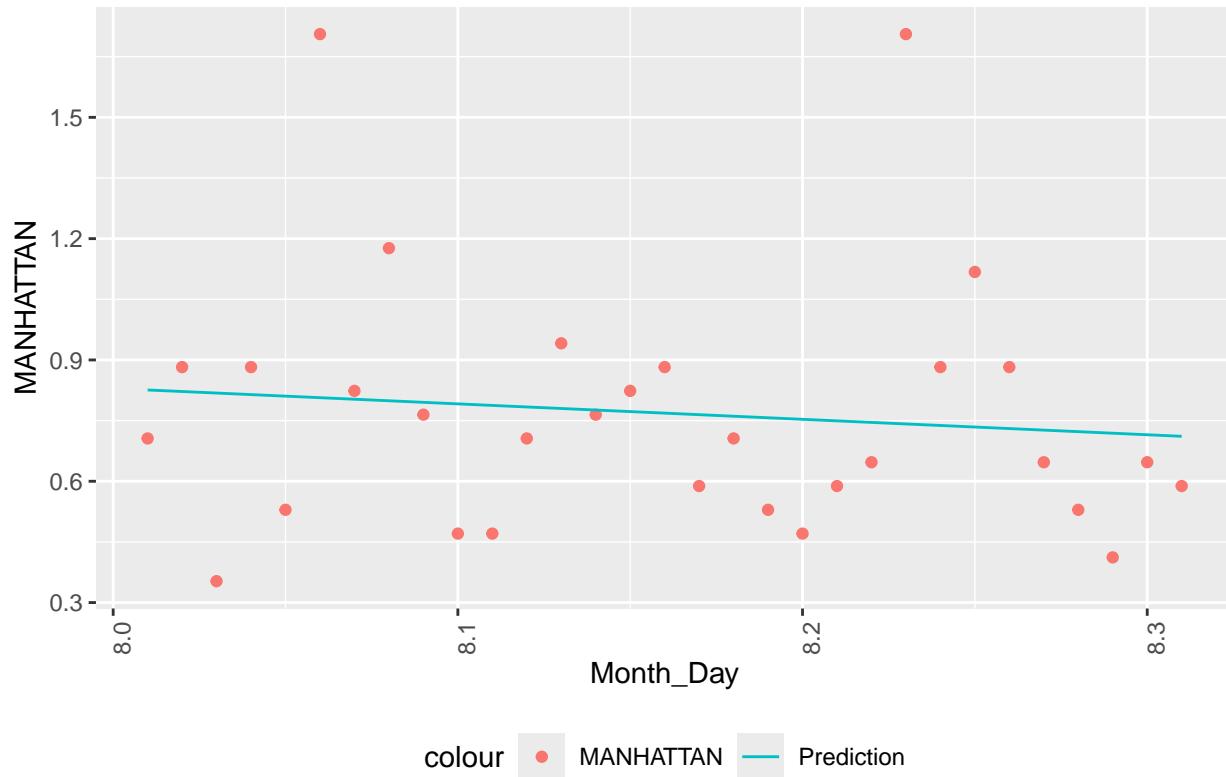


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==8)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 8) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "August Shooting Averages by Date in MANHATTAN, New York City")
```

August Shooting Averages by Date in MANHATTAN, New York City

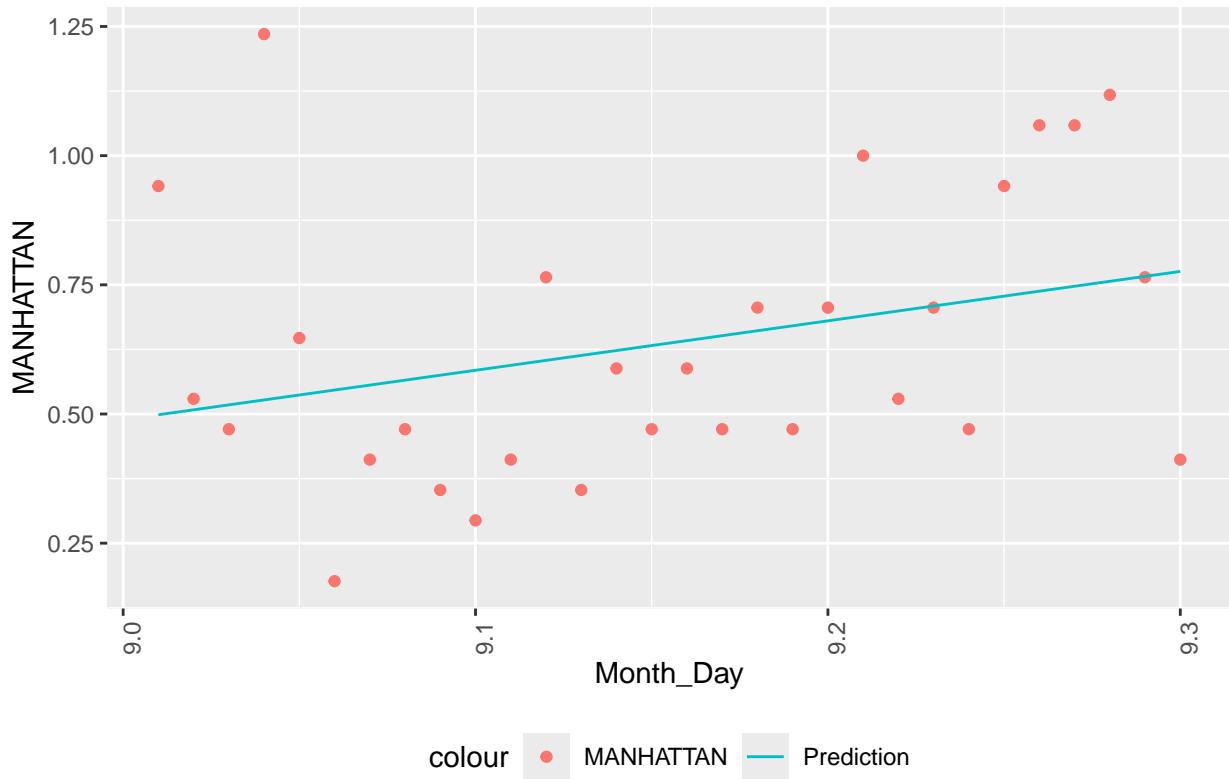


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==9)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 9) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "September Shooting Averages by Date in MANHATTAN, New York City")
```

September Shooting Averages by Date in MANHATTAN, New York City

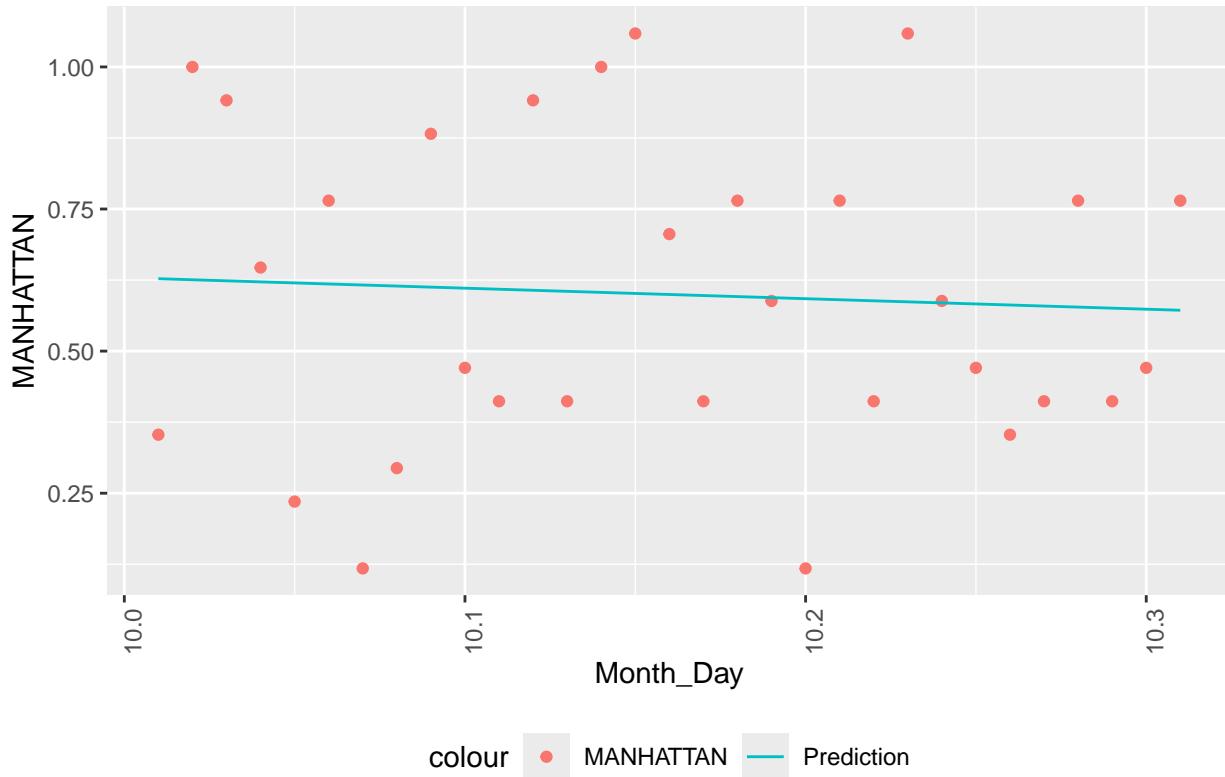


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==10)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 10) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "October Shooting Averages by Date in MANHATTAN, New York City")
```

October Shooting Averages by Date in MANHATTAN, New York City

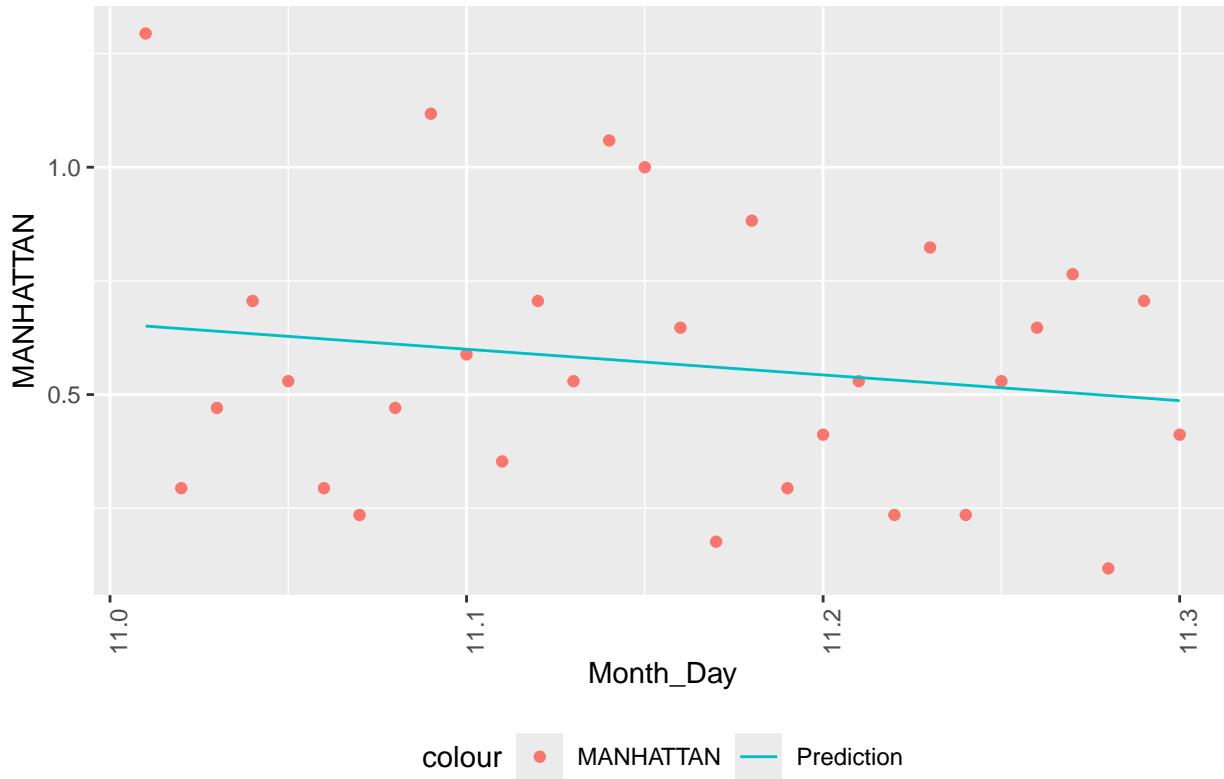


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==11)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 11) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "November Shooting Averages by Date in MANHATTAN, New York City")
```

November Shooting Averages by Date in MANHATTAN, New York City

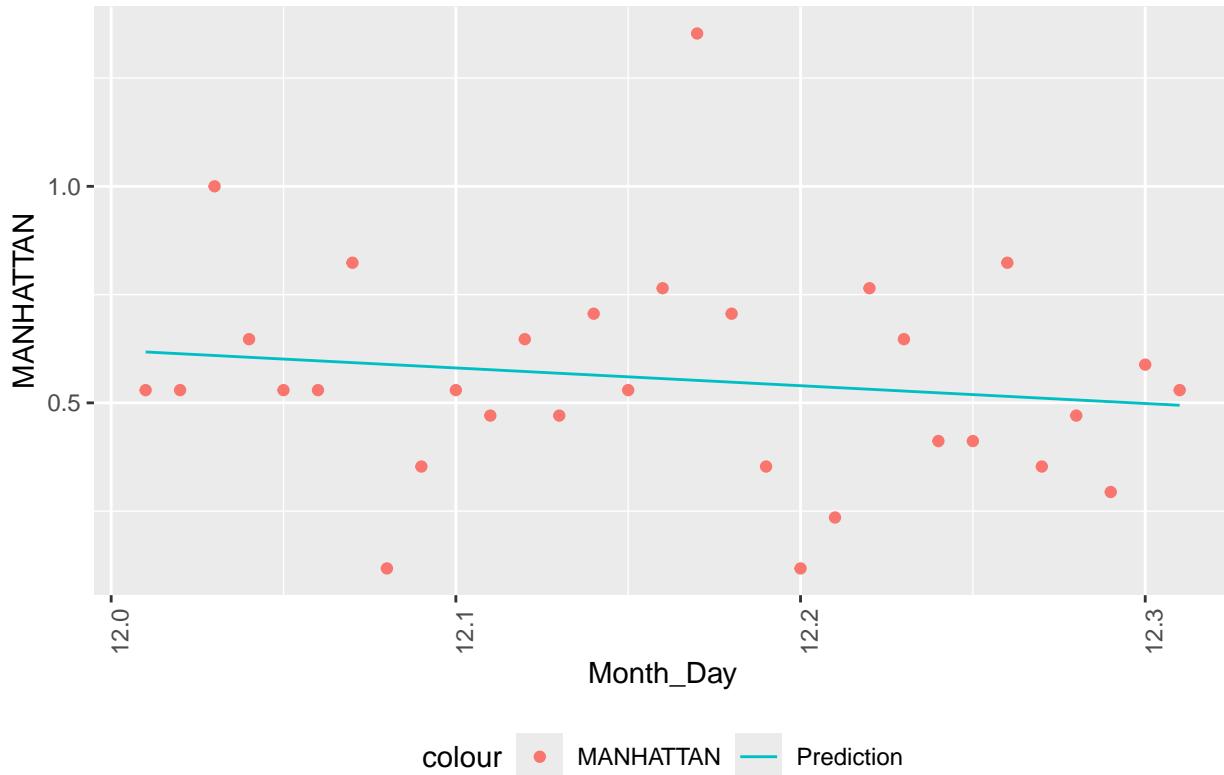


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==12)
mod <- lm(MANHATTAN ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 12) %>%
  ggplot(aes(x = Month_Day, y = MANHATTAN)) +
  geom_point(aes(color = "MANHATTAN")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "December Shooting Averages by Date in MANHATTAN, New York City")
```

December Shooting Averages by Date in MANHATTAN, New York City



```
#creates a scatterplot of the shootings over a month, daily, in the selected borough
```

In Manhattan, the strongest spike in shootings on average from 2006 to 2023 seems to be in June, but we see increases starting in March and becoming stronger until June.

There is also an increasing trend over September on average.

Now we'll check Queens.

```
n <- 1
temp1 <- temp %>%
  filter(OCCUR_MONTH==1)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 1) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "January Shooting Averages by Date in QUEENS, New York City")
```

January Shooting Averages by Date in QUEENS, New York City

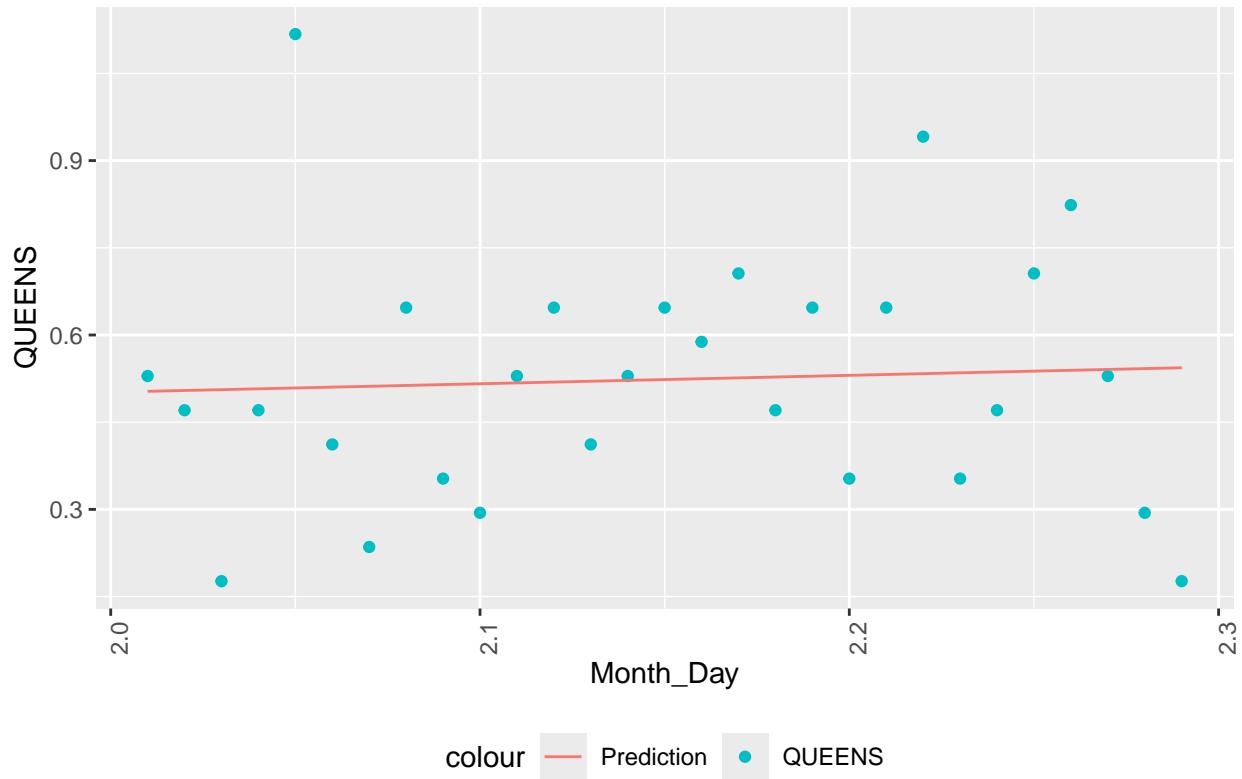


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==2)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 2) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "February Shooting Averages by Date in QUEENS, New York City")
```

February Shooting Averages by Date in QUEENS, New York City

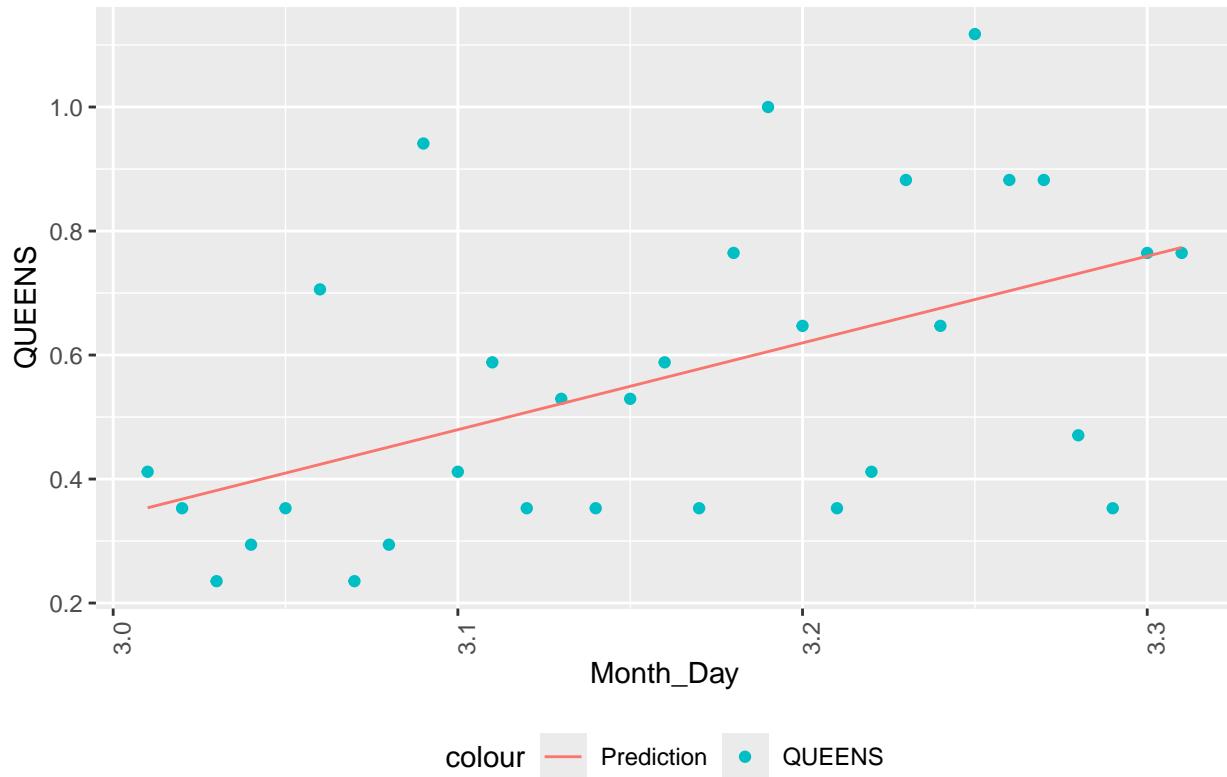


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==3)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 3) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "March Shooting Averages by Date in QUEENS, New York City")
```

March Shooting Averages by Date in QUEENS, New York City

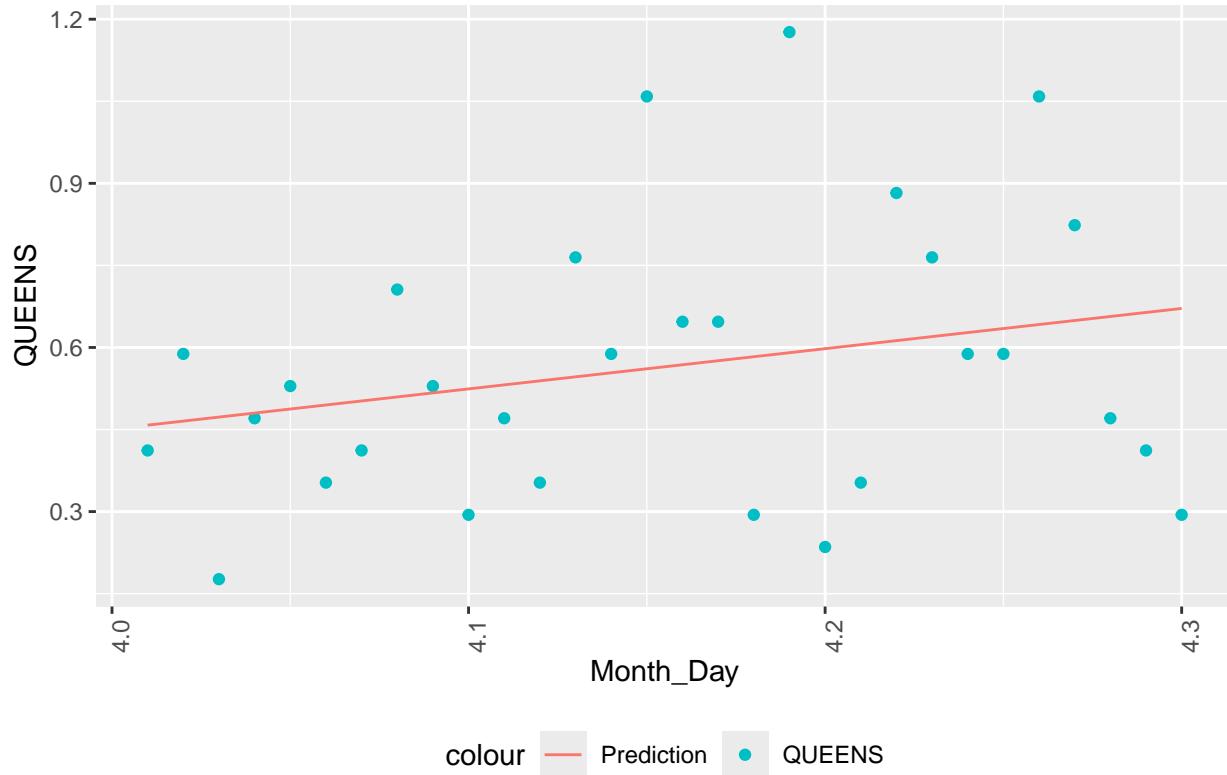


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==4)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 4) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "April Shooting Averages by Date in QUEENS, New York City")
```

April Shooting Averages by Date in QUEENS, New York City

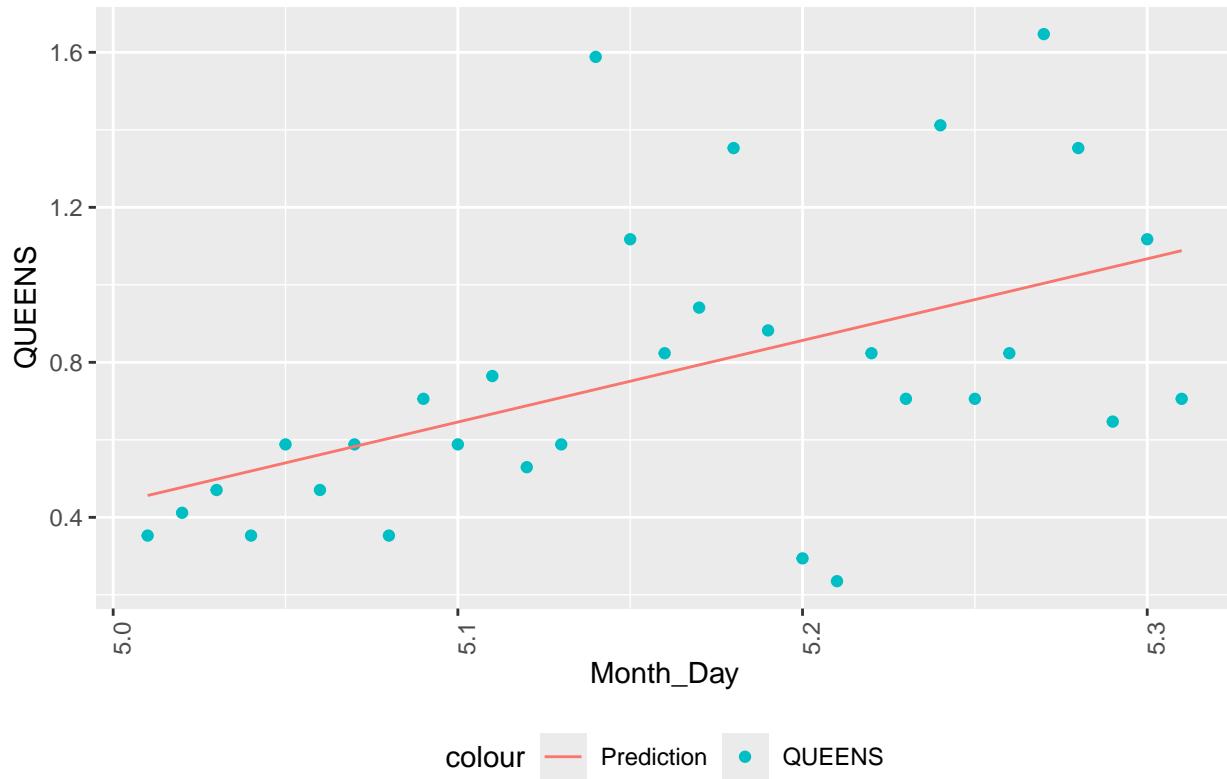


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==5)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 5) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "May Shooting Averages by Date in QUEENS, New York City")
```

May Shooting Averages by Date in QUEENS, New York City

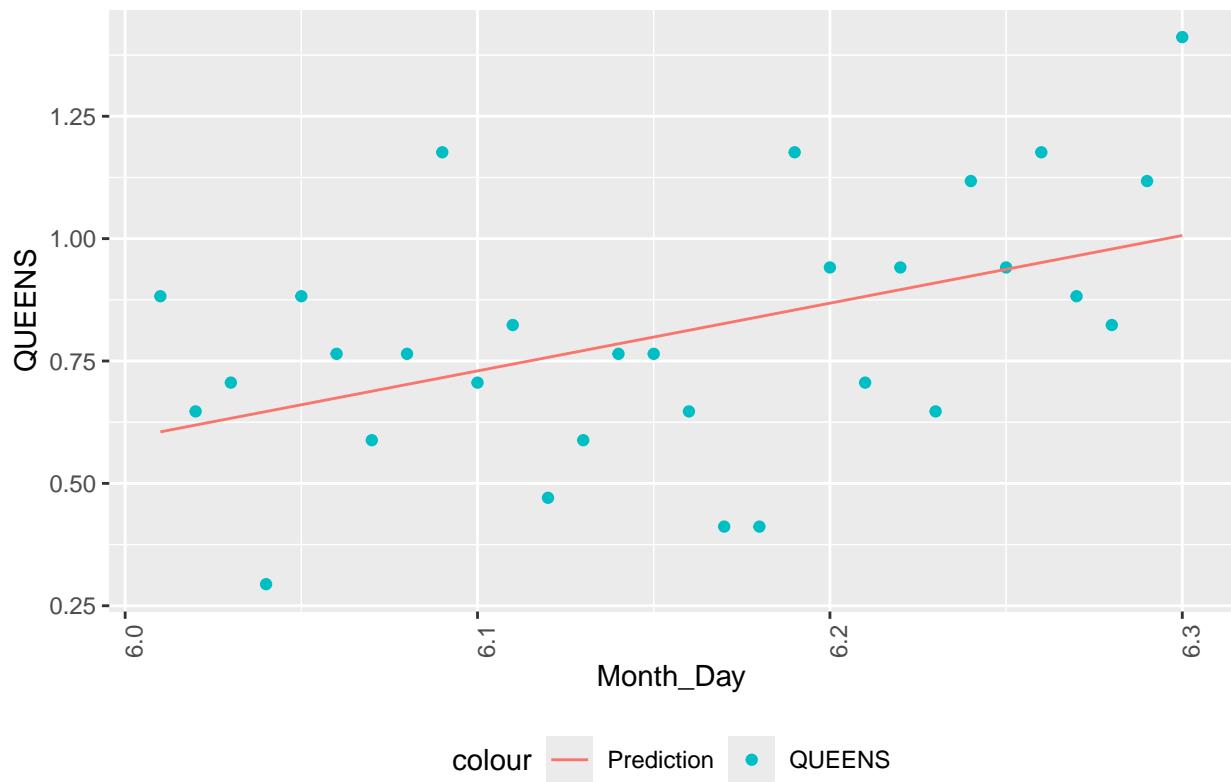


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==6)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 6) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "June Shooting Averages by Date in QUEENS, New York City")
```

June Shooting Averages by Date in QUEENS, New York City

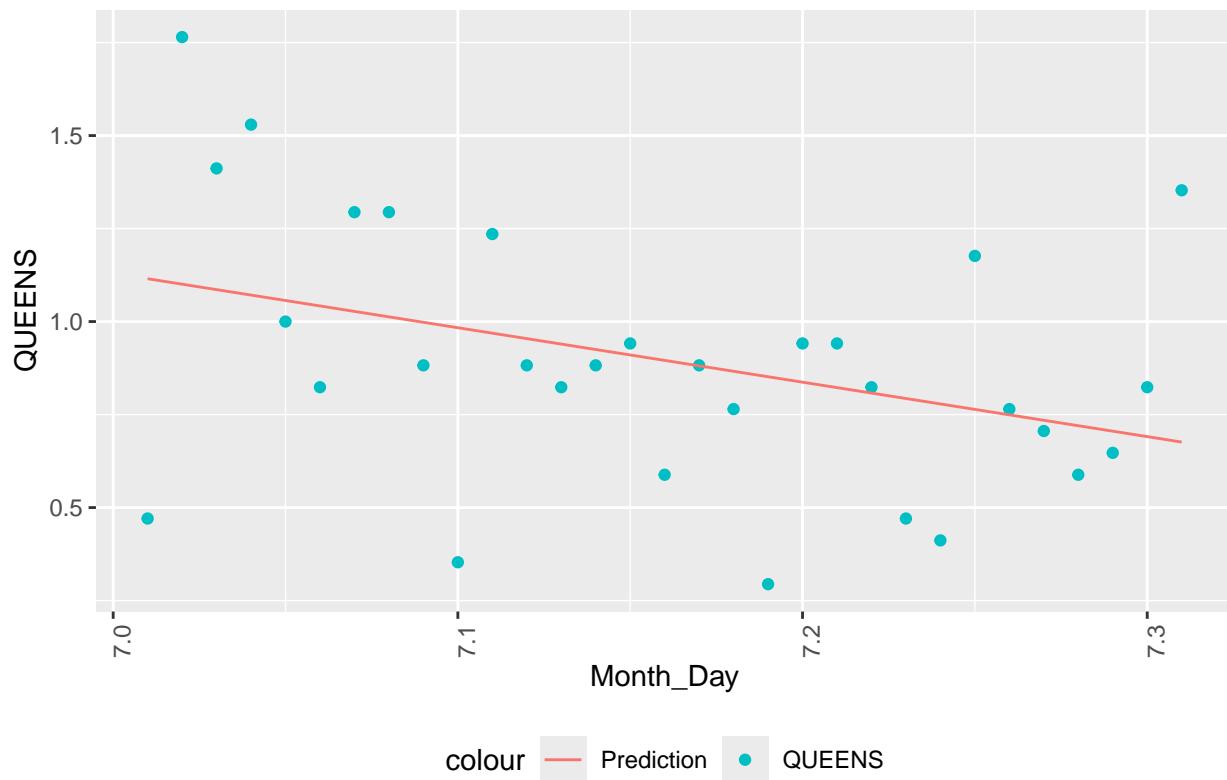


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==7)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 7) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "July Shooting Averages by Date in QUEENS, New York City")
```

July Shooting Averages by Date in QUEENS, New York City

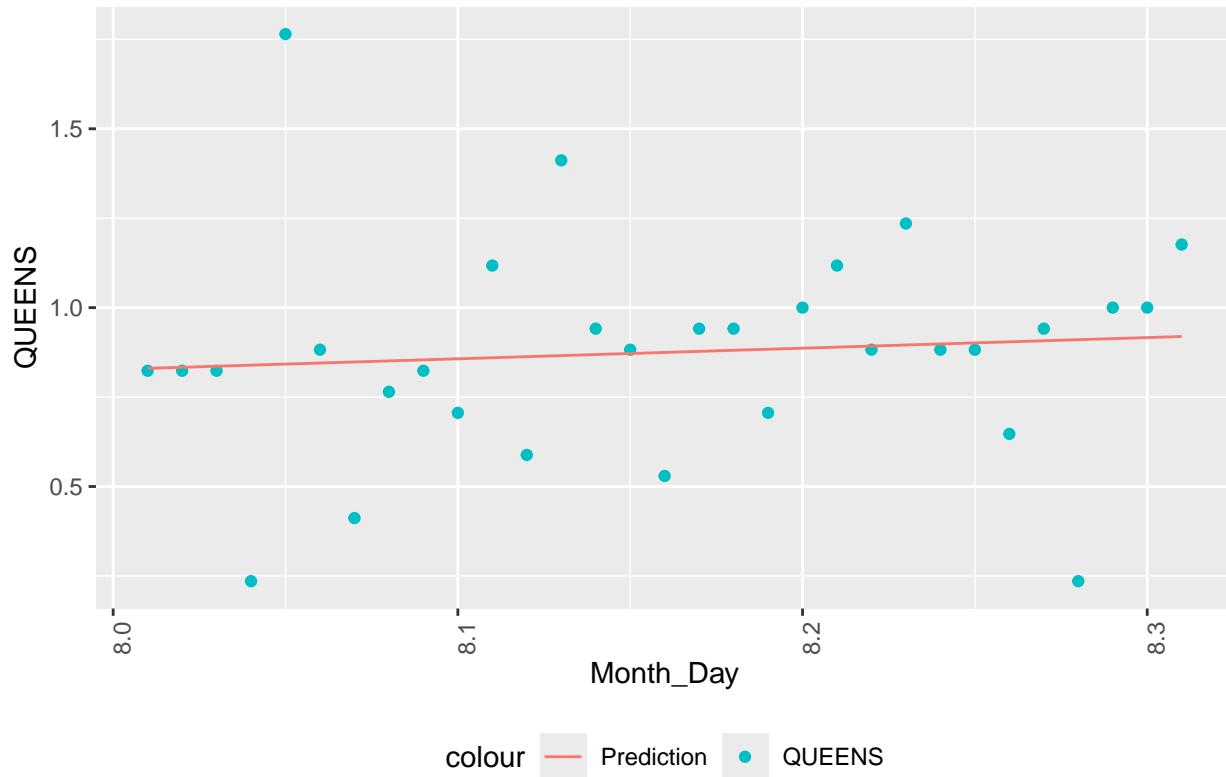


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==8)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 8) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "August Shooting Averages by Date in QUEENS, New York City")
```

August Shooting Averages by Date in QUEENS, New York City

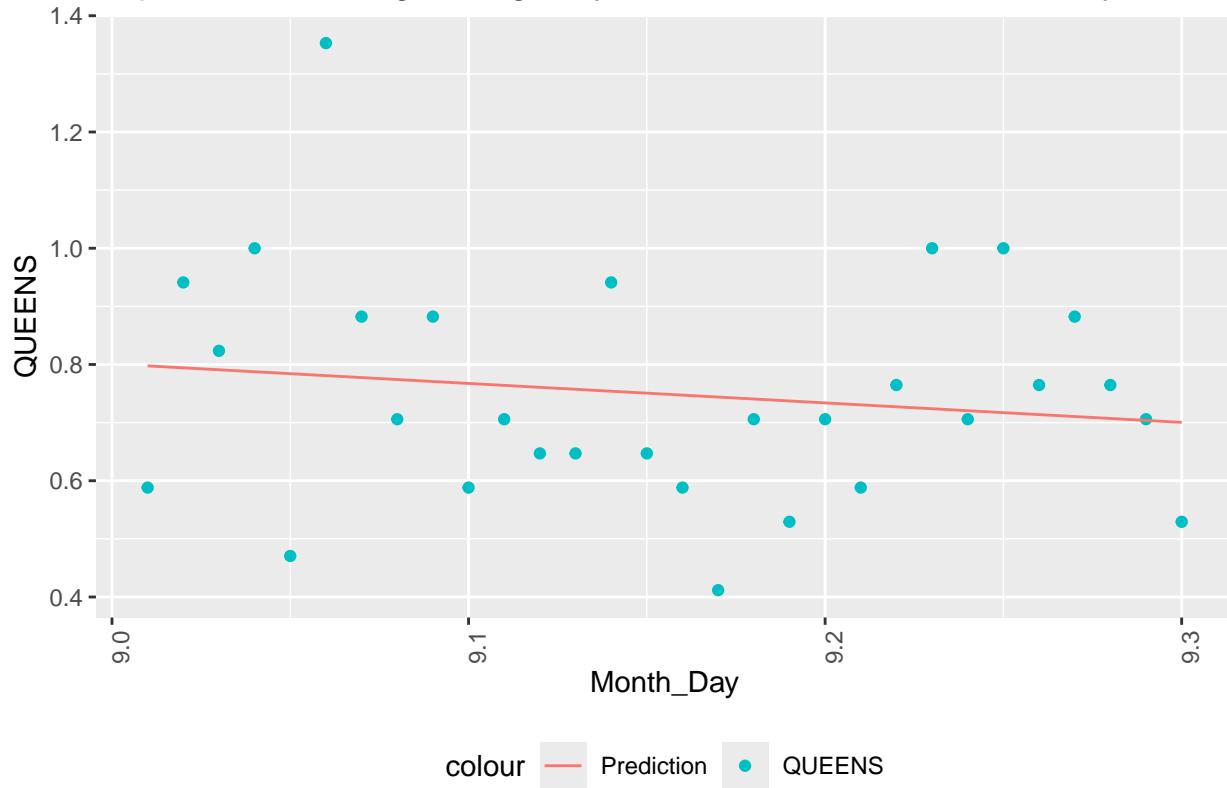


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==9)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 9) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "September Shooting Averages by Date in QUEENS, New York City")
```

September Shooting Averages by Date in QUEENS, New York City

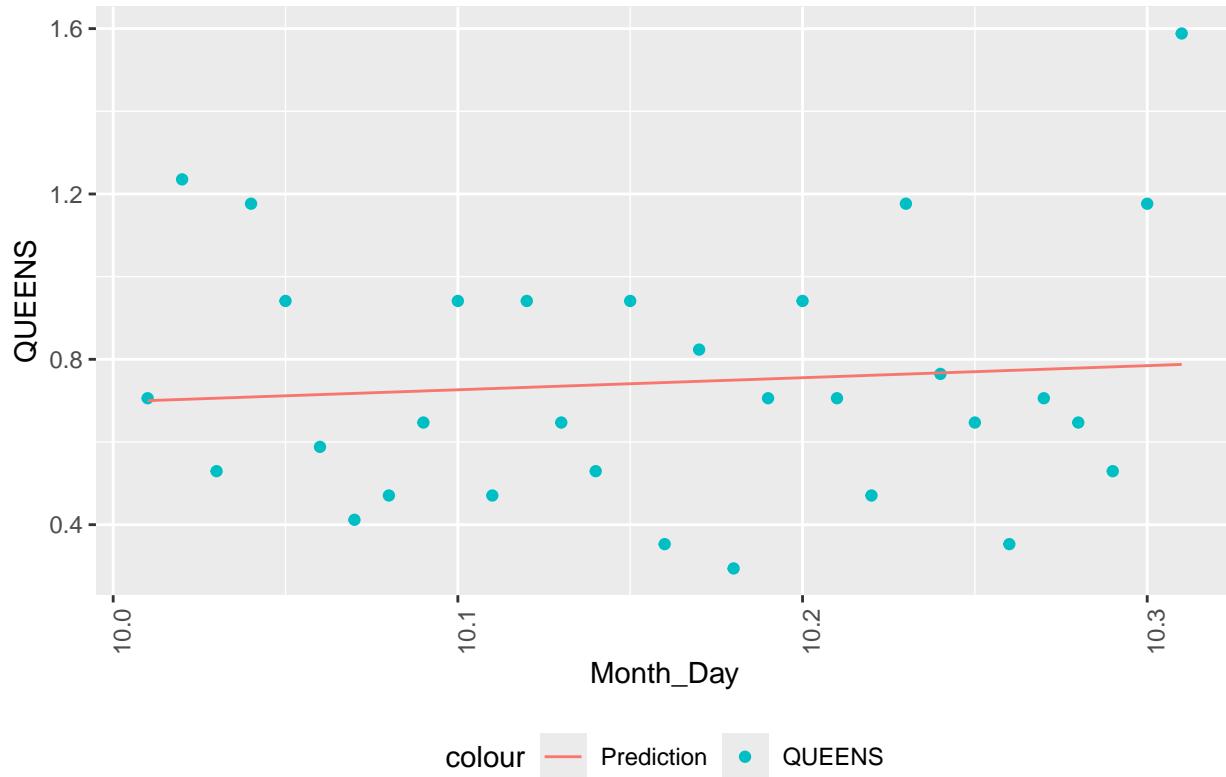


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==10)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 10) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "October Shooting Averages by Date in QUEENS, New York City")
```

October Shooting Averages by Date in QUEENS, New York City

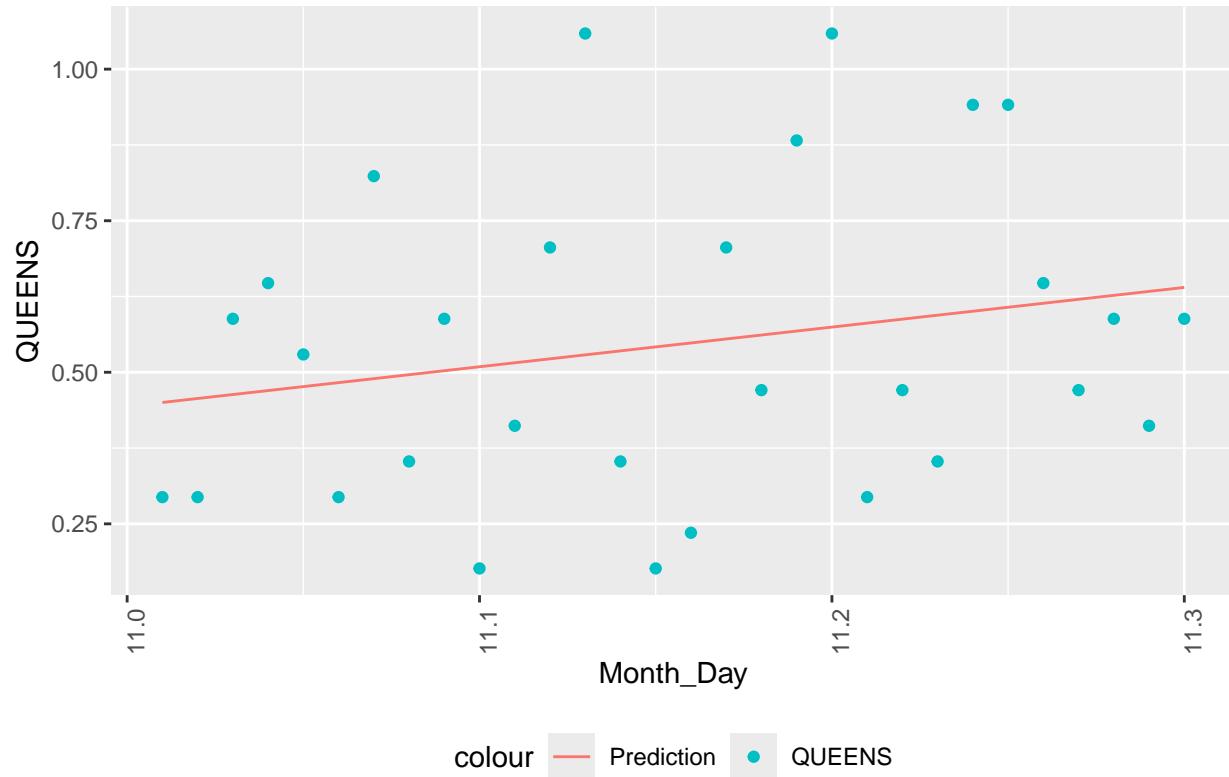


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==11)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 11) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "November Shooting Averages by Date in QUEENS, New York City")
```

November Shooting Averages by Date in QUEENS, New York City

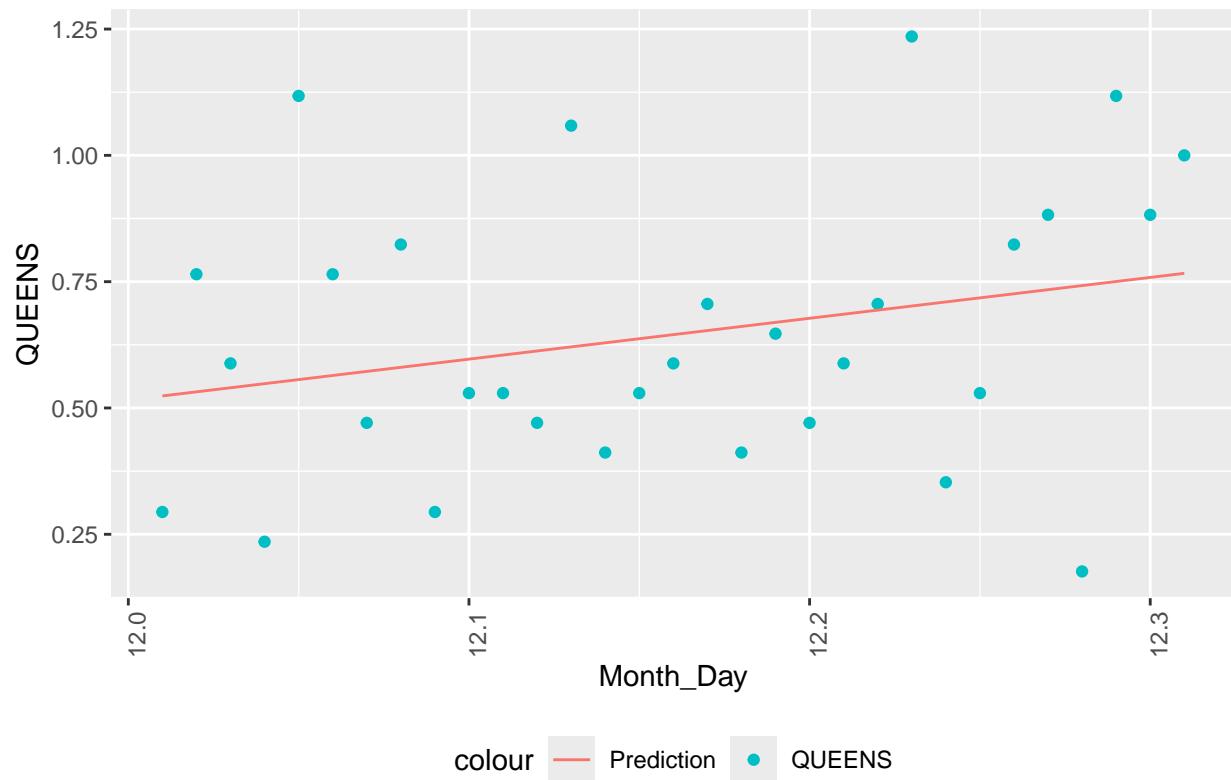


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==12)
mod <- lm(QUEENS ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 12) %>%
  ggplot(aes(x = Month_Day, y = QUEENS)) +
  geom_point(aes(color = "QUEENS")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "December Shooting Averages by Date in QUEENS, New York City")
```

December Shooting Averages by Date in QUEENS, New York City



```
#creates a scatterplot of the shootings over a month, daily, in the selected borough
```

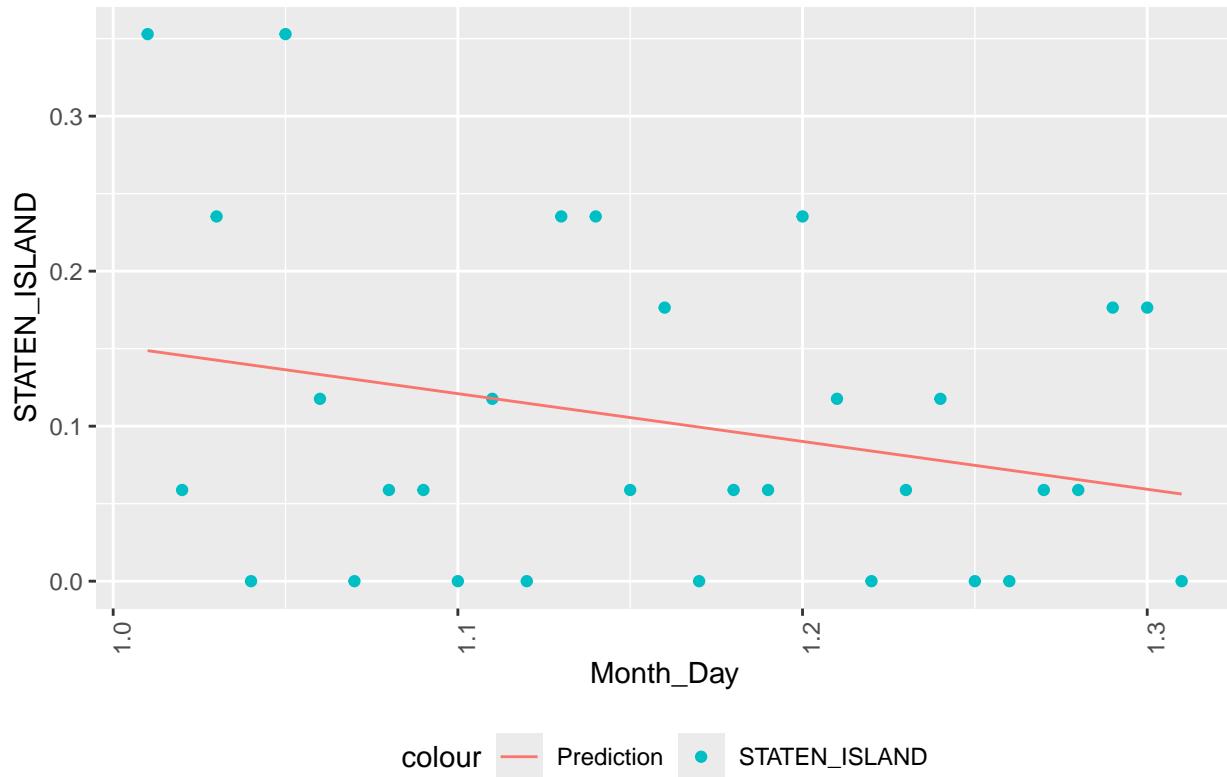
We still see midyear upward trends in Queens, as well. There is also an upward trend towards the end of the year, however.

Next, we will look at Staten Island.

```
n <- 1
temp1 <- temp %>%
  filter(OCCUR_MONTH==1)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 1) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "January Shooting Averages by Date in STATEN_ISLAND, New York City")
```

January Shooting Averages by Date in STATEN_ISLAND, New York City

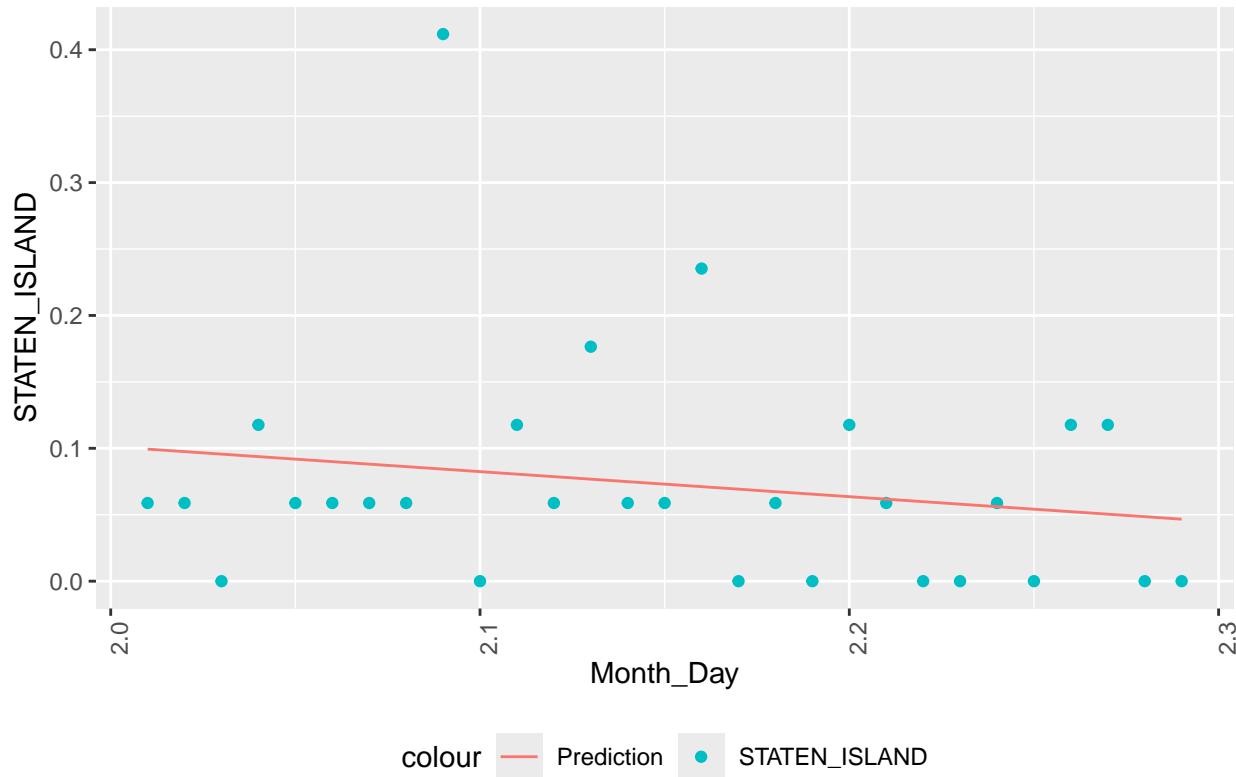


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==2)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 2) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "February Shooting Averages by Date in STATEN_ISLAND, New York City")
```

February Shooting Averages by Date in STATEN_ISLAND, New York City

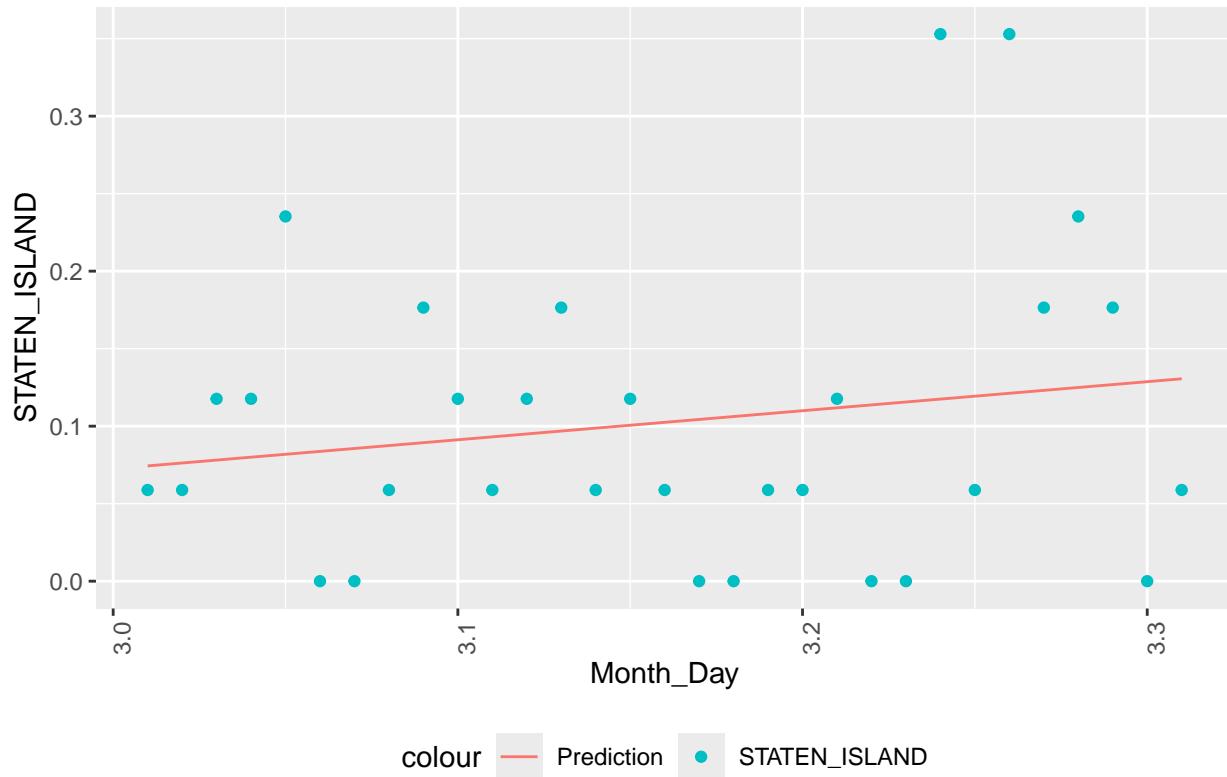


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

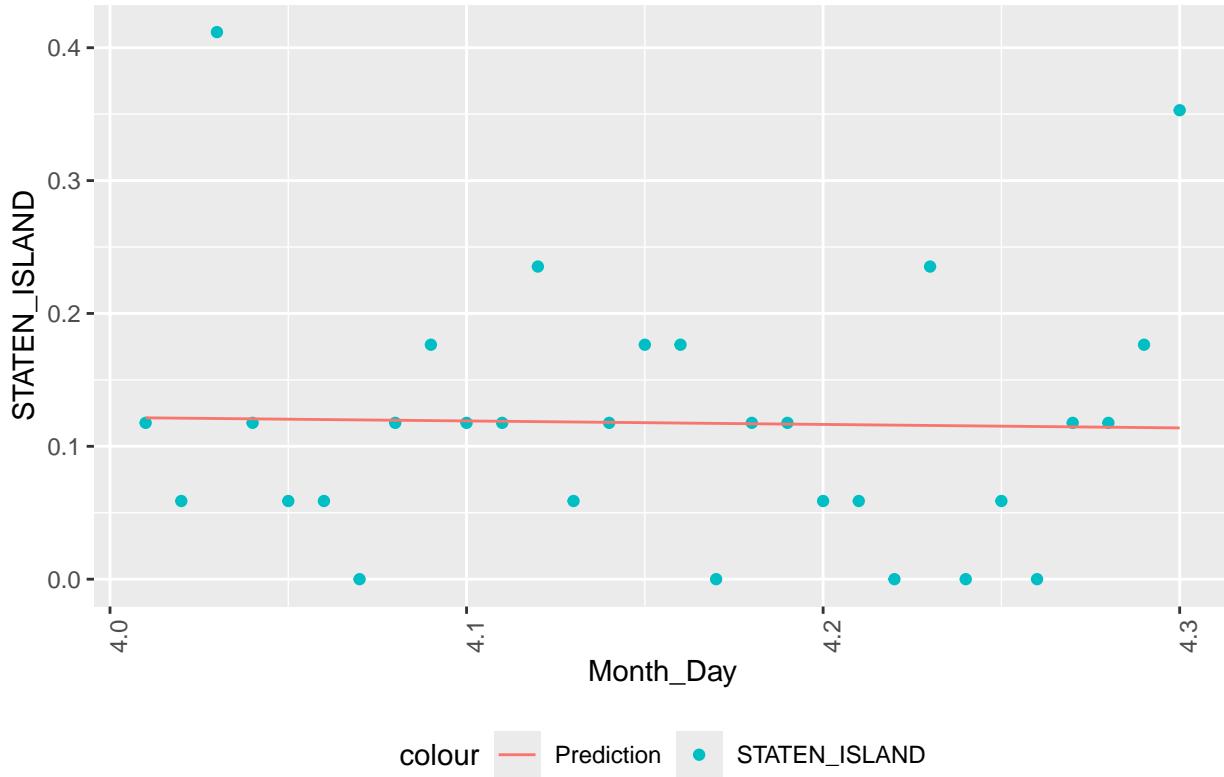
temp1 <- temp %>%
  filter(OCCUR_MONTH==3)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 3) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "March Shooting Averages by Date in STATEN_ISLAND, New York City")
```

March Shooting Averages by Date in STATEN_ISLAND, New York City



April Shooting Averages by Date in STATEN_ISLAND, New York City

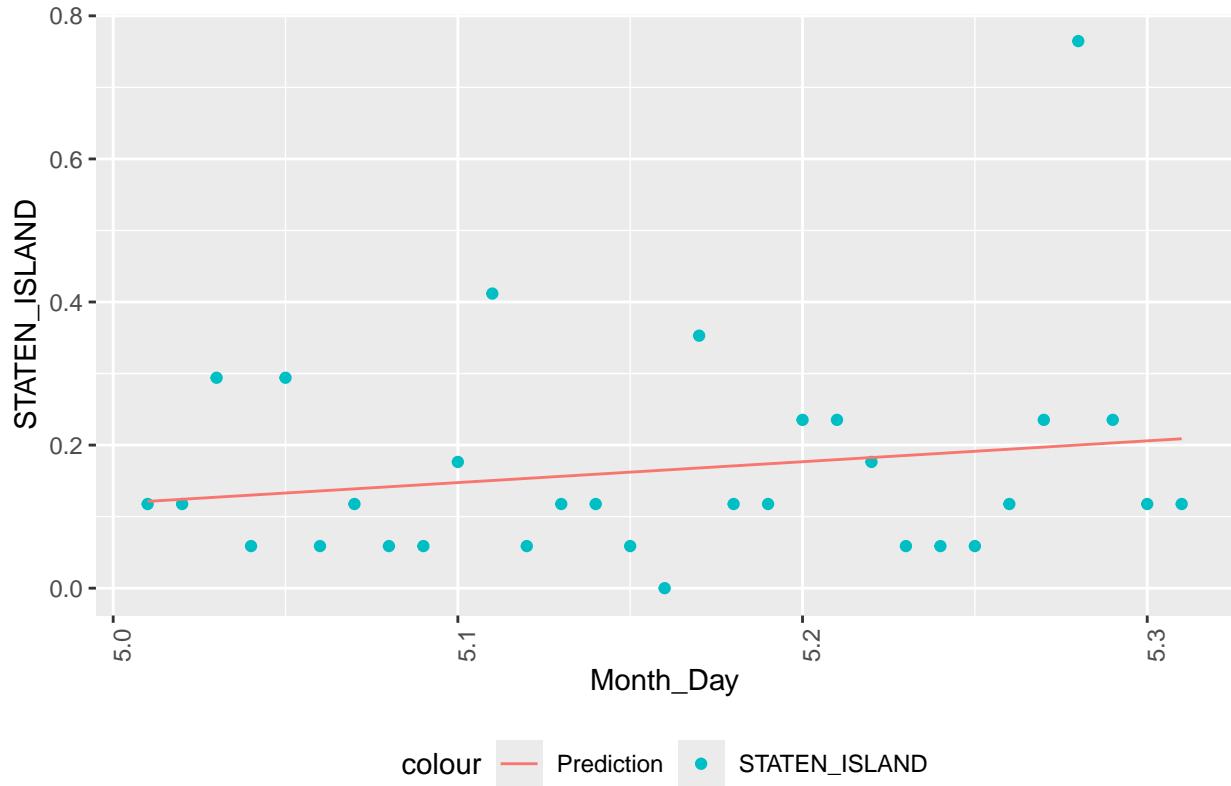


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==5)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 5) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "May Shooting Averages by Date in STATEN_ISLAND, New York City")
```

May Shooting Averages by Date in STATEN_ISLAND, New York City

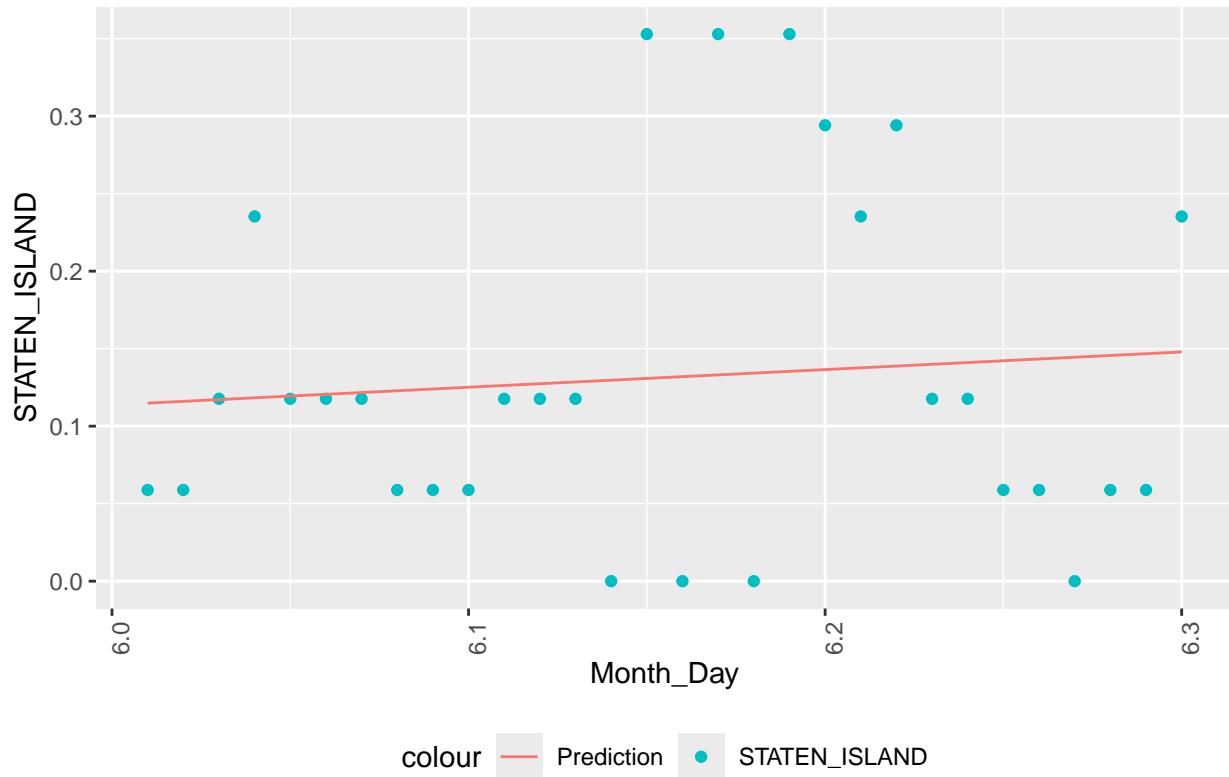


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==6)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 6) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "June Shooting Averages by Date in STATEN_ISLAND, New York City")
```

June Shooting Averages by Date in STATEN_ISLAND, New York City

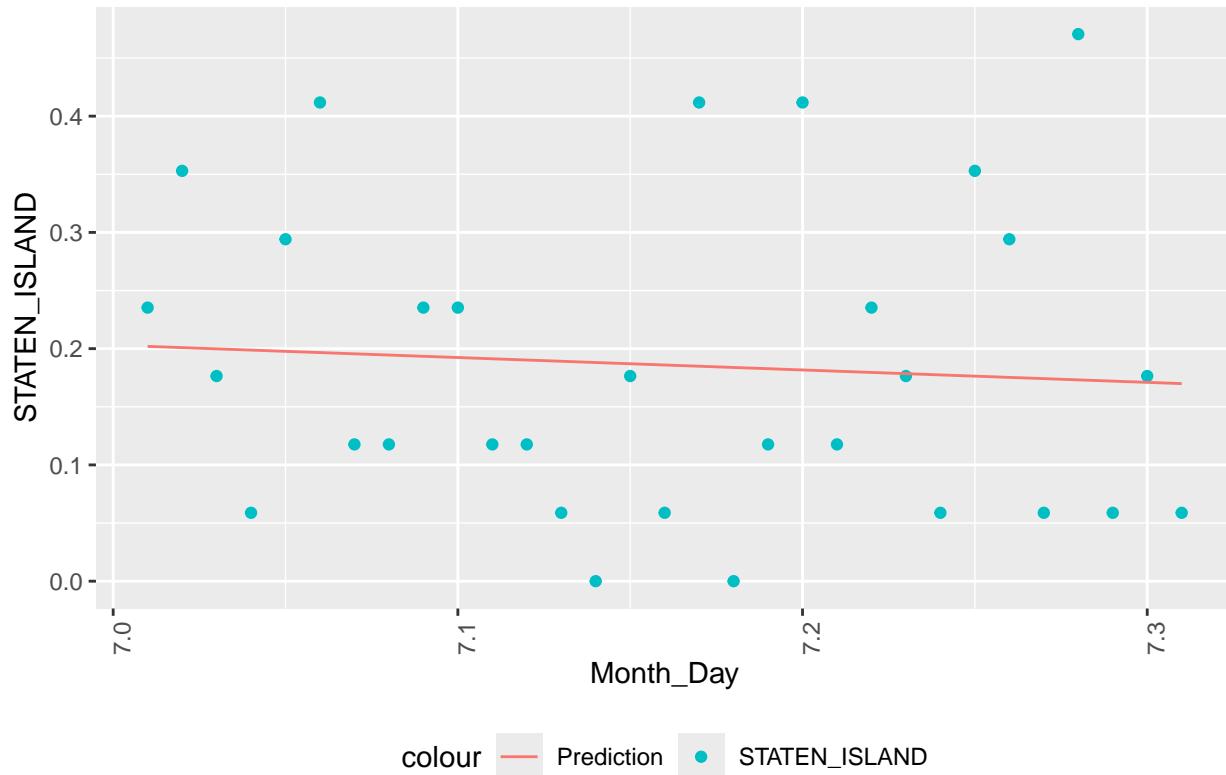


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==7)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 7) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "July Shooting Averages by Date in STATEN_ISLAND, New York City")
```

July Shooting Averages by Date in STATEN_ISLAND, New York City

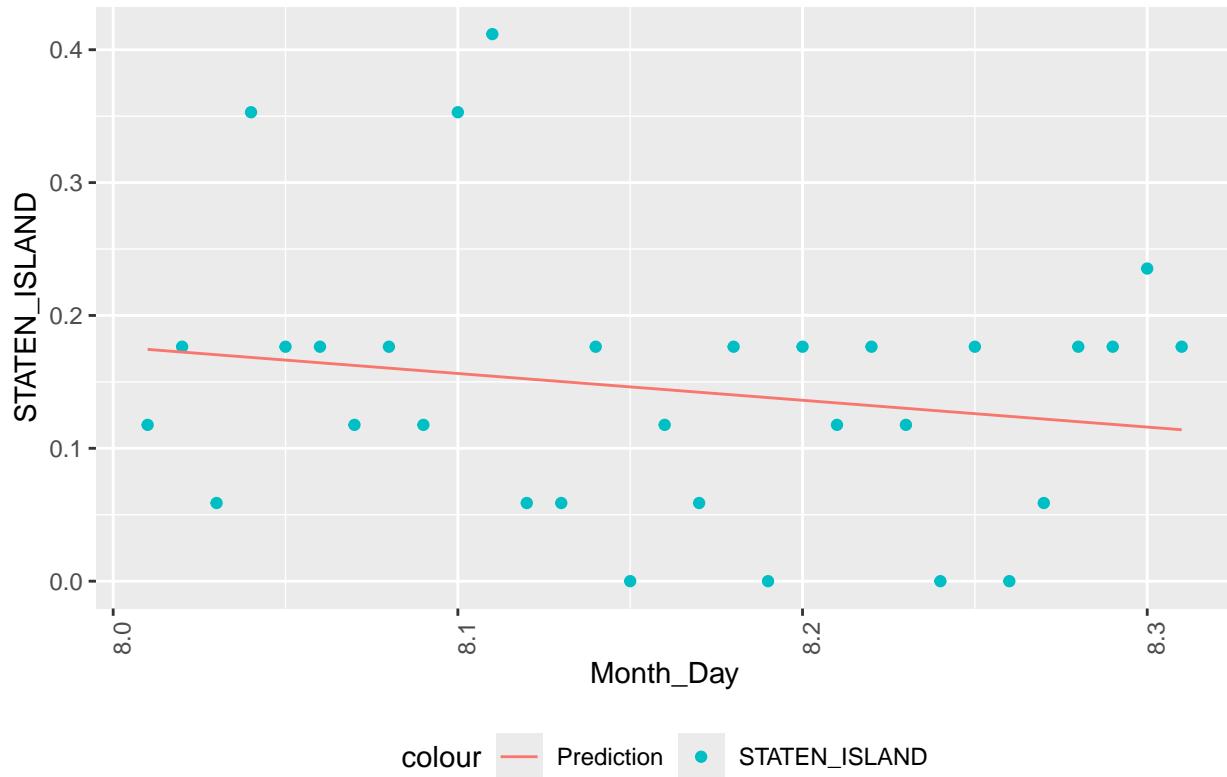


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==8)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 8) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "August Shooting Averages by Date in STATEN_ISLAND, New York City")
```

August Shooting Averages by Date in STATEN_ISLAND, New York City

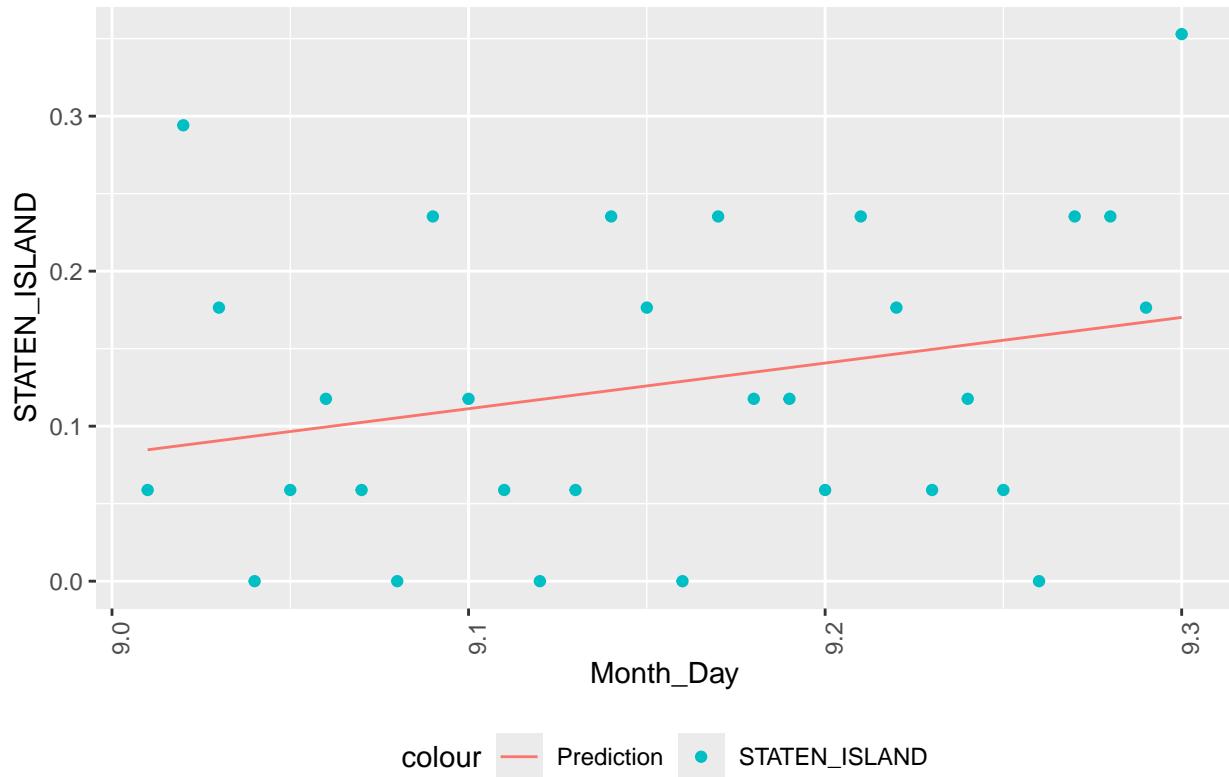


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==9)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 9) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "September Shooting Averages by Date in STATEN_ISLAND, New York City")
```

September Shooting Averages by Date in STATEN_ISLAND, New York City

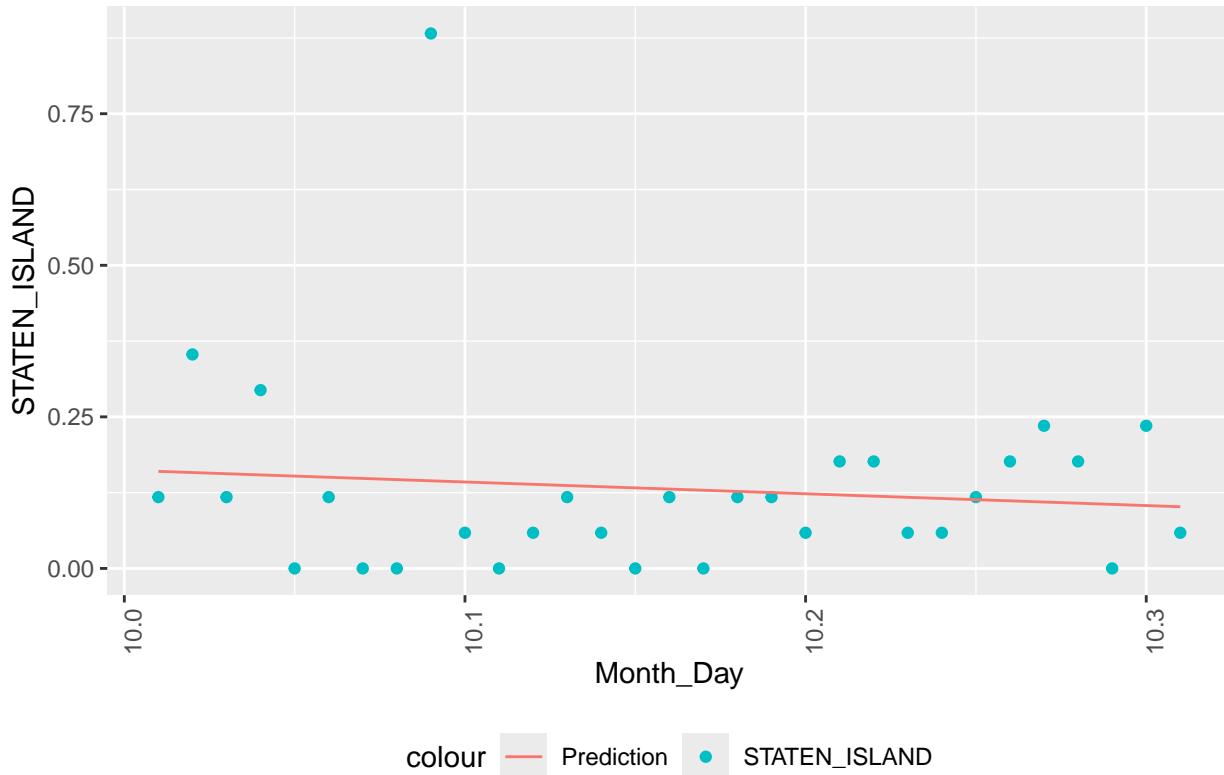


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==10)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 10) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "October Shooting Averages by Date in STATEN_ISLAND, New York City")
```

October Shooting Averages by Date in STATEN_ISLAND, New York City

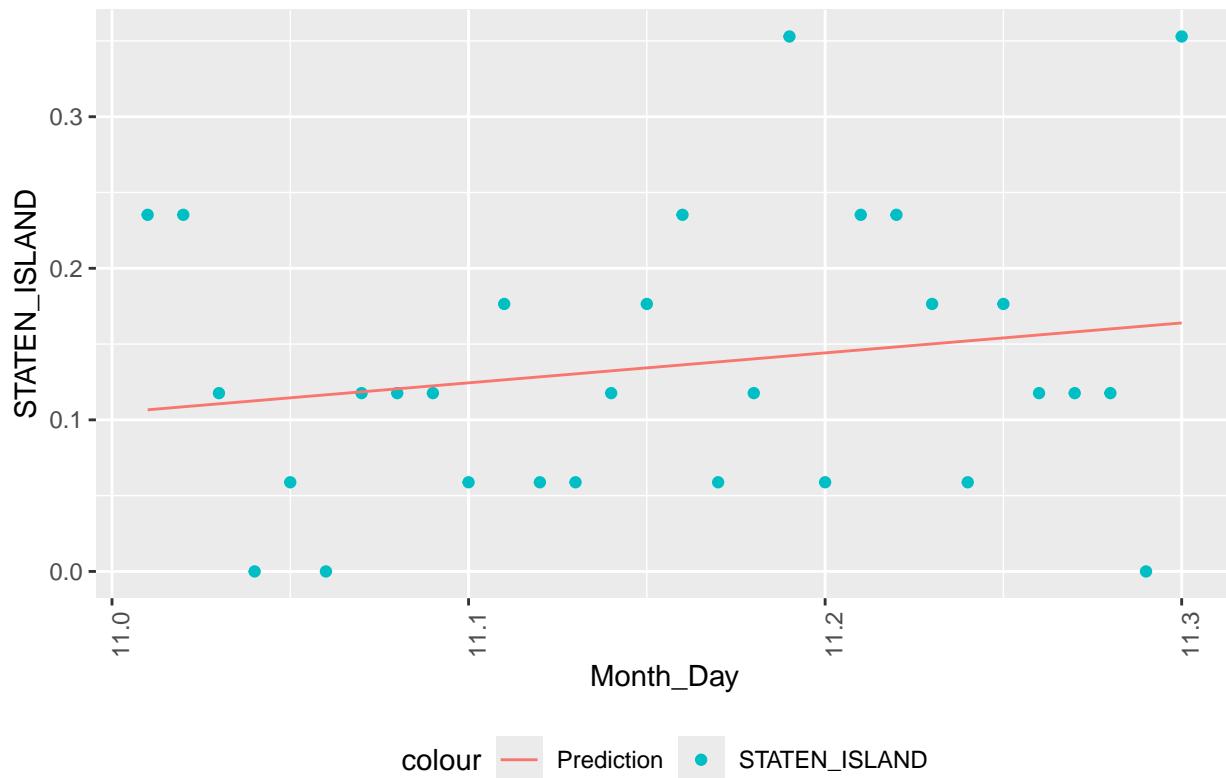


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==11)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 11) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "November Shooting Averages by Date in STATEN_ISLAND, New York City")
```

November Shooting Averages by Date in STATEN_ISLAND, New York City

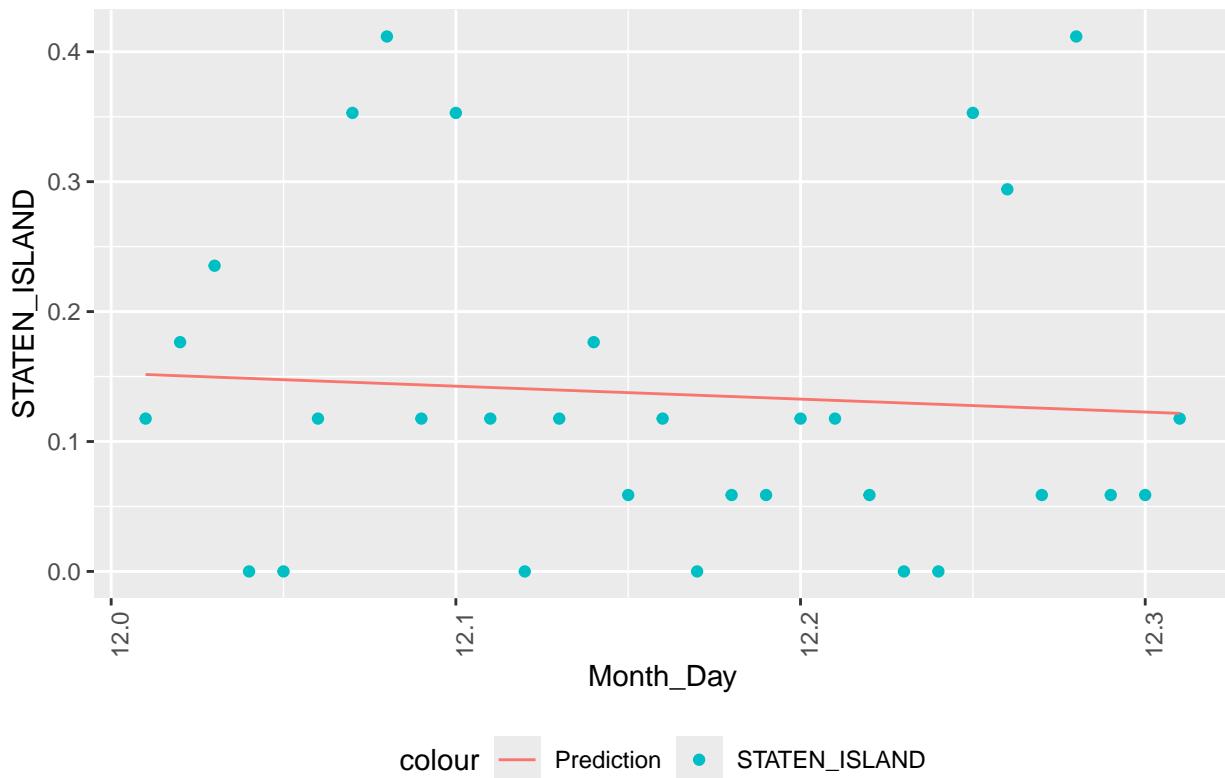


```
#creates a scatterplot of the shootings over a month, daily, in the selected borough

temp1 <- temp %>%
  filter(OCCUR_MONTH==12)
mod <- lm(STATEN_ISLAND ~ poly(Month_Day, degree = n), data = temp1)
temp1 <- temp1 %>% mutate(pred = predict(mod))
#creates model of just the selected month

temp1 %>%
  filter(OCCUR_MONTH == 12) %>%
  ggplot(aes(x = Month_Day, y = STATEN_ISLAND)) +
  geom_point(aes(color = "STATEN_ISLAND")) +
  geom_line(aes(x = Month_Day, y = pred, color = "Prediction")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90),)
  labs(title = "December Shooting Averages by Date in STATEN_ISLAND, New York City")
```

December Shooting Averages by Date in STATEN_ISLAND, New York City



```
#creates a scatterplot of the shootings over a month, daily, in the selected borough
```

Staten Island does not have a strong or sustained increase in shootings midyear according to these trends, but there is a slight increase midyear.

Conclusion

Based on demographic analysis, we found that the majority of victims of shooting incidents recorded by the NYPD across the five boroughs are:

- Black or Hispanic (Black or White)
- Between 0 and 44 years old
- Male

We also found that the majority of perpetrators across all five boroughs (ignoring unlisted demographics) are:

- Black or Hispanic (Black or white)
- Between 0 and 44 years old
- Male

Comparing population to demographics found that Black, Hispanic (Black and white), and male people are over-represented as both victims and perpetrators, along with the 0 to 25 year old age group, which consistently was the first or second highest group of shooting victims and perpetrators despite not breaching first or second highest group of the population in any borough for the census data used.

The 25-44 year old age group was often first or second highest age group in the total population of the borough, but we couldn't conclusively confirm that this was the reason they were higher as both victim and perpetrator.

We also found a yearly trend of a spike around May and June in all boroughs except Staten Island, which had less variation overall, and a higher number of shootings from 6PM to 6AM in all boroughs.

Bias Analysis

We separated victims and perpetrators from one another while also separating demographic information. This prevents a good deal of intersectional analysis that could have been done, had we gone for analyzing the demographics against one another instead of separately. This means analysis on who is shooting who, how those people present in terms of gender, and comparisons between, say white women versus Black women or old men versus young men. Here, some nuance may have been lost.

We also must rely on the race provided by the NYPD, which lacks differentiation. Hispanic groups include native people and sometimes Asian people and, furthermore, people of mixed race are not represented in the NYPD dataset.

We don't know the methodology for obtaining age, race, and sex of the perpetrators if they are not captured, nor do we know if the correct perpetrator was captured.

Additionally, the perpetrator data frequently has unlisted groups crop up in the higher side of things due to, most likely, not knowing, not recording the information of, or not seeing the perpetrator in some shootings. This being such a large portion of the shooters, all conclusions on perpetrator data should be taken with a grain of salt.

My own bias as a white woman in the 25-44 age group may have affected the tests I chose or the analysis I did, for example, the separation of demographics into isolated categories and the lack of perpetrator-victim connection in my analysis. However, I do believe that the analysis I chose is still viable, even though it could be continued and deepened with the above methods.

I mitigated my bias against men by applying the same tests to women and delving deeper into the data when I found out that more perpetrators were men than women (in addition to victims). In this way, I did find out that a higher percentage of men does not increase the number of shooting incidents.

I also was uncomfortable with the idea that Black and Hispanic people were in the majority of perpetrators in all boroughs, but still continued with the same analysis I gave sex groups, since it was also shown to be a higher number and percent than would be explained by the population totals. My bias is toward believing this would be due to socioeconomic pressures, but I refrained from including this in the conclusion above due to lack of data in this analysis to support the hypothesis.

Overall, I believe my own bias was fairly mitigated, but believe that the NYPD bias in collection could affect the outcomes, since there isn't outside data to compare to and we cannot find the perpetrators the police could not.

Session Information:

```
sessionInfo()

## R version 4.3.3 (2024-02-29 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## 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/Chicago
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets   methods    base
##
## other attached packages:
## [1] latexpdf_0.1.8 XML_3.99-0.16.1 downloader_0.4  writexl_1.5.0
## [5] readxl_1.4.3  lubridate_1.9.3 forcats_1.0.0  stringr_1.5.1
## [9] dplyr_1.1.4   purrr_1.0.2   readr_2.1.5   tidyr_1.3.1
## [13] tibble_3.2.1  ggplot2_3.5.1 tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
## [1] gtable_0.3.5     highr_0.10       compiler_4.3.3   tidyselect_1.2.1
## [5] scales_1.3.0     yaml_2.3.8       fastmap_1.1.1   R6_2.5.1
## [9] labeling_0.4.3   generics_0.1.3  knitr_1.46      munsell_0.5.1
## [13] pillar_1.9.0    tzdb_0.4.0      rlang_1.1.3    utf8_1.2.4
## [17] stringi_1.8.3   xfun_0.43      timechange_0.3.0 cli_3.6.1
## [21] withr_3.0.0     magrittr_2.0.3  digest_0.6.35   grid_4.3.3
## [25] rstudioapi_0.16.0 hms_1.1.3     lifecycle_1.0.4 vctrs_0.6.5
## [29] evaluate_0.23   glue_1.7.0      farver_2.1.1   cellranger_1.1.0
## [33] fansi_1.0.6     colorspace_2.1-0 rmarkdown_2.26  tools_4.3.3
## [37] pkgconfig_2.0.3  htmltools_0.5.8
```