NYPD Shooting Data Analysis

Description of Data

This data set contains a list of NYPD shooting incidents that occurred between 2006 and 2020. Each record contains details on when and where the shooting occurred as well as details about the victim and prep.

Install Tasks

Ensure the following tasks are installed prior to running the code. 1. tinytex::install_tinytex(version = "latest") 2. install.packages("tidyverse") 3. install.packages("ggplot2")

Load Libraries

The following libraries will be required to successfully reproduce the data.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(ggplot2)
```

Step 1: Import Data

Goal: Start an Rmd document that describes and imports the shooting project data set in a reproducible manner.

1. Import Data into Rmd

```
url_in <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
shooting_data <- read_csv(url_in[1])
```

```
## Rows: 23585 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): OCCUR_DATE, BORO, LOCATION_DESC, PERP_AGE_GROUP, PERP_SEX, PERP_R...
## dbl   (7): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD...
## lgl   (1): STATISTICAL_MURDER_FLAG
## time  (1): OCCUR_TIME
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
shooting_data
```

```
## # A tibble: 23,585 x 19
##    INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     PRECINCT JURISDICTION_CODE
##           <dbl> <chr>      <time>     <chr>       <dbl>             <dbl>
##  1     24050482 08/27/2006 05:35      BRONX          52                 0
##  2     77673979 03/11/2011 12:03      QUEENS        106                 0
##  3    203350417 10/06/2019 01:09      BROOKLYN       77                 0
##  4     80584527 09/04/2011 03:35      BRONX          40                 0
##  5     90843766 05/27/2013 21:16      QUEENS        100                 0
##  6     92393427 09/01/2013 04:17      BROOKLYN       67                 0
##  7     73057167 06/05/2010 21:16      BROOKLYN       77                 0
##  8    211362213 03/20/2020 21:27      BROOKLYN       81                 0
##  9    137564752 07/04/2014 00:25      QUEENS        101                 0
## 10    147024011 10/18/2015 01:33      QUEENS        106                 0
## # ... with 23,575 more rows, and 13 more variables: LOCATION_DESC <chr>,
## #   STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>, PERP_SEX <chr>,
## #   PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>, VIC_RACE <chr>,
## #   X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>, Longitude <dbl>,
## #   Lon_Lat <chr>
```

Step 2: Tidy & Transform Data

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

1. After the data is added, we want to remove columns that we don't want to analyze.

- Removed 14 columns

```
shooting_data <- shooting_data %>%select(cols=-c('STATISTICAL_MURDER_FLAG',
                                                 'PERP_AGE_GROUP','PERP_SEX',
                                                 'PERP_RACE','VIC_RACE',
                                                 'X_COORD_CD','Y_COORD_CD',
                                                 'Latitude','Longitude',
                                                 'Lon_Lat'))
shooting_data <- shooting_data %>%select(cols=-c('INCIDENT_KEY','LOCATION_DESC',
                                                 'PRECINCT','JURISDICTION_CODE'))
shooting_data
```

```
## # A tibble: 23,585 x 5
##    OCCUR_DATE OCCUR_TIME BORO     VIC_AGE_GROUP VIC_SEX
##    <chr>      <time>     <chr>    <chr>         <chr>
##  1 08/27/2006 05:35      BRONX    25-44         F
##  2 03/11/2011 12:03      QUEENS   65+           M
##  3 10/06/2019 01:09      BROOKLYN 18-24         F
##  4 09/04/2011 03:35      BRONX    <18           M
##  5 05/27/2013 21:16      QUEENS   18-24         M
##  6 09/01/2013 04:17      BROOKLYN <18           M
##  7 06/05/2010 21:16      BROOKLYN <18           M
##  8 03/20/2020 21:27      BROOKLYN 25-44         M
##  9 07/04/2014 00:25      QUEENS   18-24         M
## 10 10/18/2015 01:33      QUEENS   18-24         M
## # ... with 23,575 more rows
```

2. From Step 2, we notice that the OCCURED_DATE is in char format. We will transform this to the date format.

- Note: To do this, the library(lubridate) must be successfully loaded from the Load R Packages section at the beginning of the document.

```
shooting_data <- shooting_data %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE))

shooting_data
```

```
## # A tibble: 23,585 x 5
##    OCCUR_DATE OCCUR_TIME BORO     VIC_AGE_GROUP VIC_SEX
##    <date>     <time>     <chr>    <chr>         <chr>
##  1 2006-08-27 05:35      BRONX    25-44         F
##  2 2011-03-11 12:03      QUEENS   65+           M
##  3 2019-10-06 01:09      BROOKLYN 18-24         F
##  4 2011-09-04 03:35      BRONX    <18           M
##  5 2013-05-27 21:16      QUEENS   18-24         M
##  6 2013-09-01 04:17      BROOKLYN <18           M
##  7 2010-06-05 21:16      BROOKLYN <18           M
##  8 2020-03-20 21:27      BROOKLYN 25-44         M
##  9 2014-07-04 00:25      QUEENS   18-24         M
## 10 2015-10-18 01:33      QUEENS   18-24         M
## # ... with 23,575 more rows
```

3. Create two new columns which will be used for analysis further below

- Introduce a year column based on the OCCUR_DATE column
- Introduce a time of day column based on the OCCUR_TIME column
- view what the data looks like

```
shooting_data$year <- year(shooting_data$OCCUR_DATE)

shooting_data$hour <- hour(shooting_data$OCCUR_TIME)

shooting_data
```

```
## # A tibble: 23,585 x 7
##    OCCUR_DATE OCCUR_TIME BORO     VIC_AGE_GROUP VIC_SEX  year  hour
##    <date>     <time>     <chr>    <chr>         <chr>   <dbl> <int>
##  1 2006-08-27 05:35      BRONX    25-44         F        2006     5
##  2 2011-03-11 12:03      QUEENS   65+           M        2011    12
##  3 2019-10-06 01:09      BROOKLYN 18-24         F        2019     1
##  4 2011-09-04 03:35      BRONX    <18           M        2011     3
##  5 2013-05-27 21:16      QUEENS   18-24         M        2013    21
##  6 2013-09-01 04:17      BROOKLYN <18           M        2013     4
##  7 2010-06-05 21:16      BROOKLYN <18           M        2010    21
##  8 2020-03-20 21:27      BROOKLYN 25-44         M        2020    21
##  9 2014-07-04 00:25      QUEENS   18-24         M        2014     0
## 10 2015-10-18 01:33      QUEENS   18-24         M        2015     1
## # ... with 23,575 more rows
```

4. Rename columns and look at summary

- The OCCUR_DATE, OCCUR_TIME, BORO, VIC_AGE_GROUP, VIC_SEX, year and hour column names were updated to easily read the data.
- Pulled summary of data ** we have not lost any chunks of data however additional analysis will be completed below to find null or unknown values ** the date was successfully converted from char to date format ** the year was successfully implemented because the min year and max year match the min and max year within the OCCUR_DATE column ** The only 0 values are in Hour_of_Day and this makes sense because the 0th hour is the time between 12am - 12:59am

```
names(shooting_data)[1] <- "Date"
names(shooting_data)[2] <- "Time"
names(shooting_data)[3] <- "Neighborhood"
names(shooting_data)[4] <- "Victim_Age_Group"
names(shooting_data)[5] <- "Victim_Sex"
names(shooting_data)[6] <- "Year"
names(shooting_data)[7] <- "Hour_of_Day"


summary(shooting_data)
```

```
##       Date                Time           Neighborhood       Victim_Age_Group  
##  Min.   :2006-01-01   Length:23585      Length:23585       Length:23585      
##  1st Qu.:2008-12-31   Class1:hms        Class :character   Class :character  
##  Median :2012-02-27   Class2:difftime   Mode  :character   Mode  :character  
##  Mean   :2012-10-05   Mode  :numeric
##  3rd Qu.:2016-03-02
##  Max.   :2020-12-31
##   Victim_Sex             Year       Hour_of_Day
##  Length:23585       Min.   :2006   Min.   : 0.00
##  Class :character   1st Qu.:2008   1st Qu.: 3.00
##  Mode  :character   Median :2012   Median :15.00
##                     Mean   :2012   Mean   :12.08
##                     3rd Qu.:2016   3rd Qu.:20.00
##                     Max.   :2020   Max.   :23.00
```

5. Group by neighborhood, victim age group, vistim sex, year and hour of the day to determine number of shootings in each unique category. This will be further broken down in the analysis section further down.

- Complete count by Neighborhood,Victim_Age_Group, Victim_Sex, Year, Hour_of_Day
- Assign column name "Shooting_Incident"
- assign this table to a new dataframe called gb_shooting_data
- view gb_shooting_data

```
gb_shooting_data <- shooting_data %>% count(Neighborhood, Victim_Age_Group,
                                            Victim_Sex, Year, Hour_of_Day,
                                            sort = TRUE)

names(gb_shooting_data)[6] <- "Shooting_Incident"


gb_shooting_data
```

```
## # A tibble: 5,753 x 6
##    Neighborhood Victim_Age_Group Victim_Sex  Year Hour_of_Day Shooting_Incident
##    <chr>        <chr>            <chr>      <dbl>       <int>             <int>
##  1 BROOKLYN     25-44            M           2020          22                43
##  2 BRONX        18-24            M           2011           1                41
##  3 BROOKLYN     25-44            M           2007          23                41
##  4 BROOKLYN     25-44            M           2020           1                40
##  5 BROOKLYN     18-24            M           2007           2                37
##  6 BROOKLYN     18-24            M           2008          22                34
##  7 BROOKLYN     25-44            M           2020          21                34
##  8 BROOKLYN     18-24            M           2010           1                33
##  9 BROOKLYN     18-24            M           2006          22                32
## 10 BROOKLYN     18-24            M           2007          23                32
## # ... with 5,743 more rows
```

6. Clean up unknown values because they can skew findings

- Check if there's any unknown values
- Filter out any data points with unknown values

```
gb_shooting_data_clean <- filter(gb_shooting_data, Neighborhood != "UNKNOWN"
                                 & Victim_Age_Group != "UNKNOWN"
                                 & Victim_Sex != "U"
                                 & Year != "UNKNOWN"
                                 & Hour_of_Day != "UNKNOWN")
```

7. confirm that unknowns are gone, we should see an empty list if there are no unknowns

```
filter(gb_shooting_data_clean, Victim_Age_Group =="UNKNOWN")
```

```
## # A tibble: 0 x 6
## # ... with 6 variables: Neighborhood <chr>, Victim_Age_Group <chr>,
## #   Victim_Sex <chr>, Year <dbl>, Hour_of_Day <int>, Shooting_Incident <int>
```

8. Review summary

- all unknown values are removed
- no values are missing
- Hour_of_Day is based on a 24 hour clock so the minimum of 0 means any time between 12:00am - 12:59am and maximum of 23 means any time between 11:00pm to 11:59pm.

```
summary(gb_shooting_data_clean)
```

```
##  Neighborhood       Victim_Age_Group    Victim_Sex              Year     
##  Length:5707        Length:5707        Length:5707        Min.   :2006  
##  Class :character   Class :character   Class :character   1st Qu.:2009  
##  Mode  :character   Mode  :character   Mode  :character   Median :2012  
##                                                            Mean   :2013  
##                                                            3rd Qu.:2016  
##                                                            Max.   :2020  
##   Hour_of_Day    Shooting_Incident
##  Min.   : 0.00   Min.   : 1.000   
##  1st Qu.: 4.00   1st Qu.: 1.000   
##  Median :13.00   Median : 2.000   
##  Mean   :12.04   Mean   : 4.121   
##  3rd Qu.:19.00   3rd Qu.: 5.000   
##  Max.   :23.00   Max.   :43.000   
```

Step 3: Add Visulaizations and Analysis

Research Questions

1. Which neighborhood in New York has the most shooting incidents? How do shooting incidents change over time?
2. How do shooting incidents vary by age for men and women?
3. Is hour of day related to shooting incidents?

Visualization for Research Question 1

Which neighborhood in New York has the most shooting incidents? How do shooting incidents change over time?

1. Create a data frame for number of shootings by neighborhood for each year.

- Group by year and neighborhood

- Sum shooting incidents
- Store in a new dataframe called df_vis1

```
df_vis1 <- gb_shooting_data_clean %>% group_by(Year, Neighborhood) %>%
  summarise(Shooting_Incidents=sum(Shooting_Incident))
```

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

2. Rename the columns

- Renamed count (n) to Shooting_Incidents

```
summary(df_vis1)
```

```
##       Year      Neighborhood       Shooting_Incidents
##  Min.   :2006   Length:75          Min.   : 25.0     
##  1st Qu.:2009   Class :character   1st Qu.:143.0     
##  Median :2013   Mode  :character   Median :267.0     
##  Mean   :2013                      Mean   :313.6     
##  3rd Qu.:2017                      3rd Qu.:500.5     
##  Max.   :2020                      Max.   :848.0
```

3. Review maximum for Shooting_Incidents to determine if 848 makes sense.

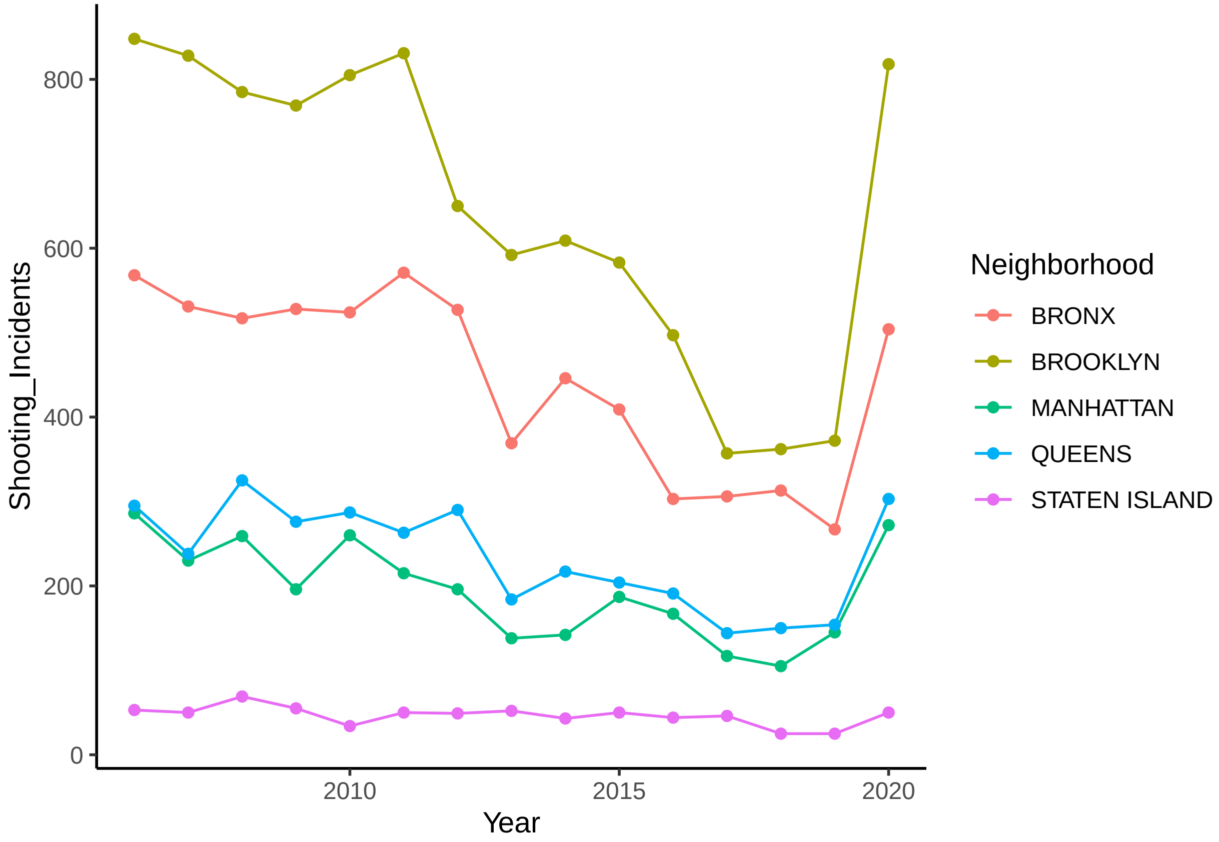
- There are many responses returned which indicates that this was not a typo

```
df_vis1 %>% filter(Shooting_Incidents > 750.00)
```

```
## # A tibble: 7 x 3
## # Groups:   Year [7]
##    Year Neighborhood Shooting_Incidents
##   <dbl> <chr>                     <int>
## 1  2006 BROOKLYN                    848
## 2  2007 BROOKLYN                    828
## 3  2008 BROOKLYN                    785
## 4  2009 BROOKLYN                    769
## 5  2010 BROOKLYN                    805
## 6  2011 BROOKLYN                    831
## 7  2020 BROOKLYN                    818
```

4. Create a Visualization *Create and store the graph in a variable ** Note: To do this, the library(ggplot2) must be successfully loaded from the Load R Packages section at the beginning of the document. *Call the graph to view it

```
ln_plot_vis1 <- ggplot(df_vis1, aes(x=Year, y=Shooting_Incidents,
                                    group=Neighborhood)) +
  geom_line(aes(color=Neighborhood)) + geom_point(aes(color=Neighborhood)) +
  theme_classic()

ln_plot_vis1
```



Analysis for Research Question 1

There appears to be a clear distinction in number of shootings by neighborhood throughout the years. At no point, do any of the lines cross each other which tells me that on average, Brooklyn sees the most shootings out of all of these neighborhoods. 2020 saw a significant increases in shootings which may be a skew however they may be due to the riots that took place in 2020.

Visualization for Research Question 2

How do shooting incidents vary by age for men and women?

1. Create a data frame for number of shootings by age and sex

- group by victim age and victim sex
- sum the shooting incidents
- store in a new data frame called df_vis2

```
df_vis2 <- gb_shooting_data_clean %>% group_by(Victim_Age_Group, Victim_Sex) %>%
  summarise(Shooting_Incidents=sum(Shooting_Incident))
```

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

2. Rename the column

- Rename count (n) to Shooting_Incidents

```
summary(df_vis2)
```

```
##  Victim_Age_Group    Victim_Sex        Shooting_Incidents
##  Length:10          Length:10          Min.   :  49.0    
##  Class :character   Class :character   1st Qu.: 316.2    
##  Mode  :character   Mode  :character   Median : 742.5    
##                                        Mean   :2352.0    
##                                        3rd Qu.:1930.2    
##                                        Max.   :9484.0    
```

3. Review maximum for Shooting_Incidents to determine if 9484 makes sense.

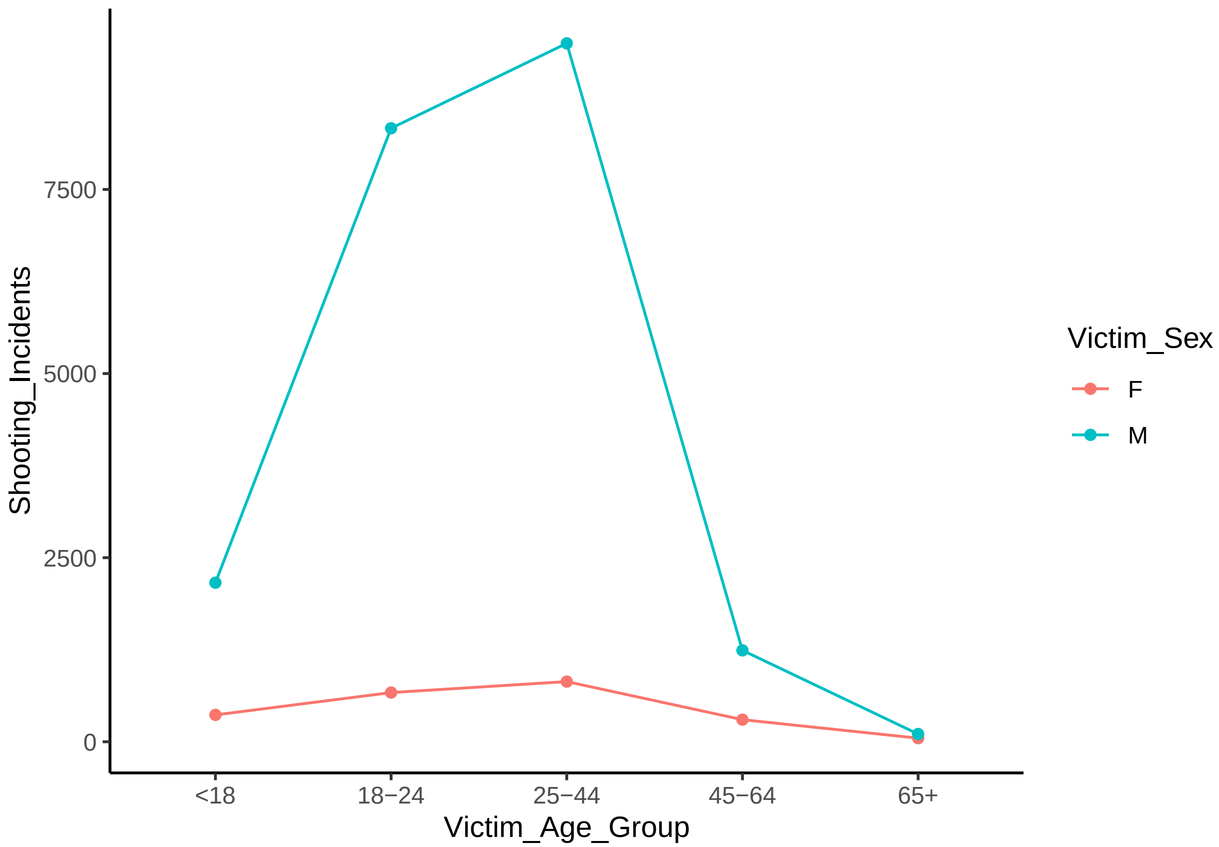
- There are a handful of responses returned which indicates that this was not a typo

```
df_vis2 %>% filter(Shooting_Incidents > 8000.00)
```

```
## # A tibble: 2 x 3
## # Groups:   Victim_Age_Group [2]
##   Victim_Age_Group Victim_Sex Shooting_Incidents
##   <chr>            <chr>                   <int>
## 1 18-24            M                        8331
## 2 25-44            M                        9484
```

4. Create a Visualization *Create and store the graph in a variable ** Note: To do this, the library(ggplot2) must be successfully loaded from the Load R Packages section at the beginning of the document. *Call the graph to view it

```
n_plot_vis2 <- ggplot(df_vis2, aes(x=Victim_Age_Group, y=Shooting_Incidents,
                                   group=Victim_Sex)) +
    geom_line(aes(color=Victim_Sex))+
    geom_point(aes(color=Victim_Sex))+
    theme_classic()
n_plot_vis2
```



Analysis for Research Question 2

There appears to be a stark difference in number of shooting incidents for men based on their age. The highest number of shootings appear to occur for men in the 25-44 age group. This makes sense because men in that age group are more likely to live in regions with higher shooting incidents. The number of shooting incidents where the victim is male drops significantly for men in the 45-64 age group because that age group tends to move towards the suburbs of the city where there are less shooting incidents (ex: Staten Island). On the other hand, women appear to be victims of shooting incidents at a consistent rate throughout their life span.

Model for Research Question 3

Is hour of day related to shooting incidents?

Build a Model & Visualize

1. Create a data frame for number of shootings by hour of the day

- Group by hour of the day
- Sum the shootings
- Assign this to the data frame df_vis3

```r
df_vis3 <- gb_shooting_data_clean %>% group_by(Hour_of_Day) %>%
  summarise(Shooting_Incidents=sum(Shooting_Incident))
df_vis3
```

```
## # A tibble: 24 x 2
##    Hour_of_Day Shooting_Incidents
##          <int>              <int>
##  1           0               1902
##  2           1               1864
##  3           2               1618
##  4           3               1462
##  5           4               1292
##  6           5                635
##  7           6                300
##  8           7                198
##  9           8                188
## 10           9                177
## # ... with 14 more rows
```

2. Create the model

```r
mod <- lm(Shooting_Incidents ~ Hour_of_Day, data = df_vis3)
```

3. summarize the model

```r
summary(mod)
```

```
##
## Call:
## lm(formula = Shooting_Incidents ~ Hour_of_Day, data = df_vis3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -773.5 -584.2 -149.1  591.3 1057.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   844.09     253.10   3.335    0.003 **
## Hour_of_Day    11.82      18.86   0.627    0.537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 639.4 on 22 degrees of freedom
## Multiple R-squared:  0.01754,    Adjusted R-squared:  -0.02711
## F-statistic: 0.3928 on 1 and 22 DF,  p-value: 0.5373
```

4. interpret in this scenario, my shooting incidents are 844 + 11 times the time of day

5. add Predictions

```
df_vis3 %>% mutate(Predictions = predict(mod))
```

```
## # A tibble: 24 x 3
##    Hour_of_Day Shooting_Incidents Predictions
##          <int>              <int>       <dbl>
##  1           0               1902        844.
##  2           1               1864        856.
##  3           2               1618        868.
##  4           3               1462        880.
##  5           4               1292        891.
##  6           5                635        903.
##  7           6                300        915.
##  8           7                198        927.
##  9           8                188        939.
## 10           9                177        950.
## # ... with 14 more rows
```

6. create a new data set to see the predictions

- New data frame is called df_vis_w_pred

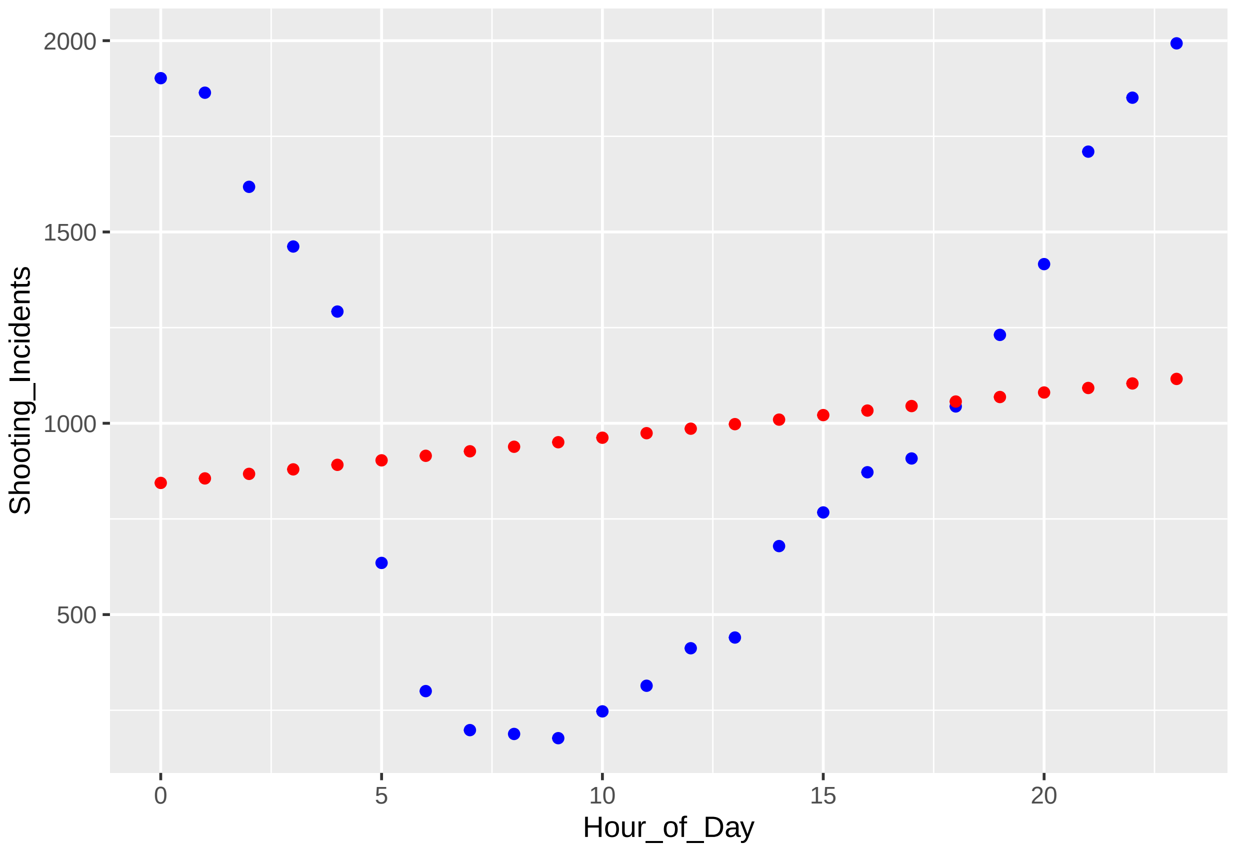
```
df_vis3_w_pred <- df_vis3 %>% mutate(Predictions = predict(mod))


df_vis3_w_pred
```

```
## # A tibble: 24 x 3
##    Hour_of_Day Shooting_Incidents Predictions
##          <int>              <int>       <dbl>
##  1           0               1902        844.
##  2           1               1864        856.
##  3           2               1618        868.
##  4           3               1462        880.
##  5           4               1292        891.
##  6           5                635        903.
##  7           6                300        915.
##  8           7                198        927.
##  9           8                188        939.
## 10           9                177        950.
## # ... with 14 more rows
```

7. plot the data to see how we're doing

- Note: To do this, the library(ggplot2) must be successfully loaded from the Load R Packages section at the beginning of the document.

```
df_vis3_w_pred %>% ggplot() +
  geom_point(aes(x = Hour_of_Day, y = Shooting_Incidents), color = "blue") +
  geom_point(aes(x = Hour_of_Day, y = Predictions), color = "red")
```



Analysis for Model

Shooting incidents appear to peak overnight and dip in the morning. This parabola makes it seem as though hour of the day doesn't impact shooting incidents however that's not the case. Lets break this out to look at the first half of the day and the second half of the day.

From the below graphs, we can see that in the first half of the day, hour of day is predictive of shooting incidents. The shooting incidents decrease towards noon because: * In the early morning, some folks may be outdoors or socializing and venues are still open. * As we approach 4-5am there's a large drop because most folks have gone home to sleep.

For the second half of the day, hour of day is predictive of shooting incidents as well however the tred is in the opposite direction. The shooting incidents increase towards midnight because: * In the afternoon, folks who slept late or slept in are waking up * Public venues are now open * As we approach the end of the working day (5pm), folks are going out to socialize after work

First Half of the Day

1. Create a data frame for number of shootings during the first half of the day

- Filter for first half of the day hours
- Group by hour of the day
- Sum the shootings
- Assign this to the data frame df_vis3

```r
gb_shooting_data_clean_morning <- gb_shooting_data_clean %>%
  filter(Hour_of_Day < 12)

df_vis4 <- gb_shooting_data_clean_morning %>% group_by(Hour_of_Day) %>%
  summarise(Shooting_Incidents=sum(Shooting_Incident))
df_vis4
```

```
## # A tibble: 12 x 2
##    Hour_of_Day Shooting_Incidents
##          <int>              <int>
##  1           0               1902
##  2           1               1864
##  3           2               1618
##  4           3               1462
##  5           4               1292
##  6           5                635
##  7           6                300
##  8           7                198
##  9           8                188
## 10           9                177
## 11          10                247
## 12          11                314
```

2. Create the model

```r
mod1 <- lm(Shooting_Incidents ~ Hour_of_Day, data = df_vis4)
```

3. summarize the model

```r
summary(mod1)
```

```
##
## Call:
## lm(formula = Shooting_Incidents ~ Hour_of_Day, data = df_vis4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -458.67 -231.24   90.57  175.41  466.08
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1851.58     159.80  11.587 4.06e-07 ***
## Hour_of_Day  -182.15      24.61  -7.402 2.31e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 294.3 on 10 degrees of freedom
## Multiple R-squared:  0.8457, Adjusted R-squared:  0.8302
## F-statistic: 54.79 on 1 and 10 DF,  p-value: 2.31e-05
```

4. add Predictions

```
df_vis4 %>% mutate(Predictions = predict(mod1))
```

```
## # A tibble: 12 x 3
##    Hour_of_Day Shooting_Incidents Predictions
##          <int>              <int>       <dbl>
##  1           0               1902      1852.
##  2           1               1864      1669.
##  3           2               1618      1487.
##  4           3               1462      1305.
##  5           4               1292      1123.
##  6           5                635       941.
##  7           6                300       759.
##  8           7                198       577.
##  9           8                188       394.
## 10           9                177       212.
## 11          10                247        30.1
## 12          11                314      -152.
```

5. create a new data set to see the predictions

- New data frame is called df_vis_w_pred

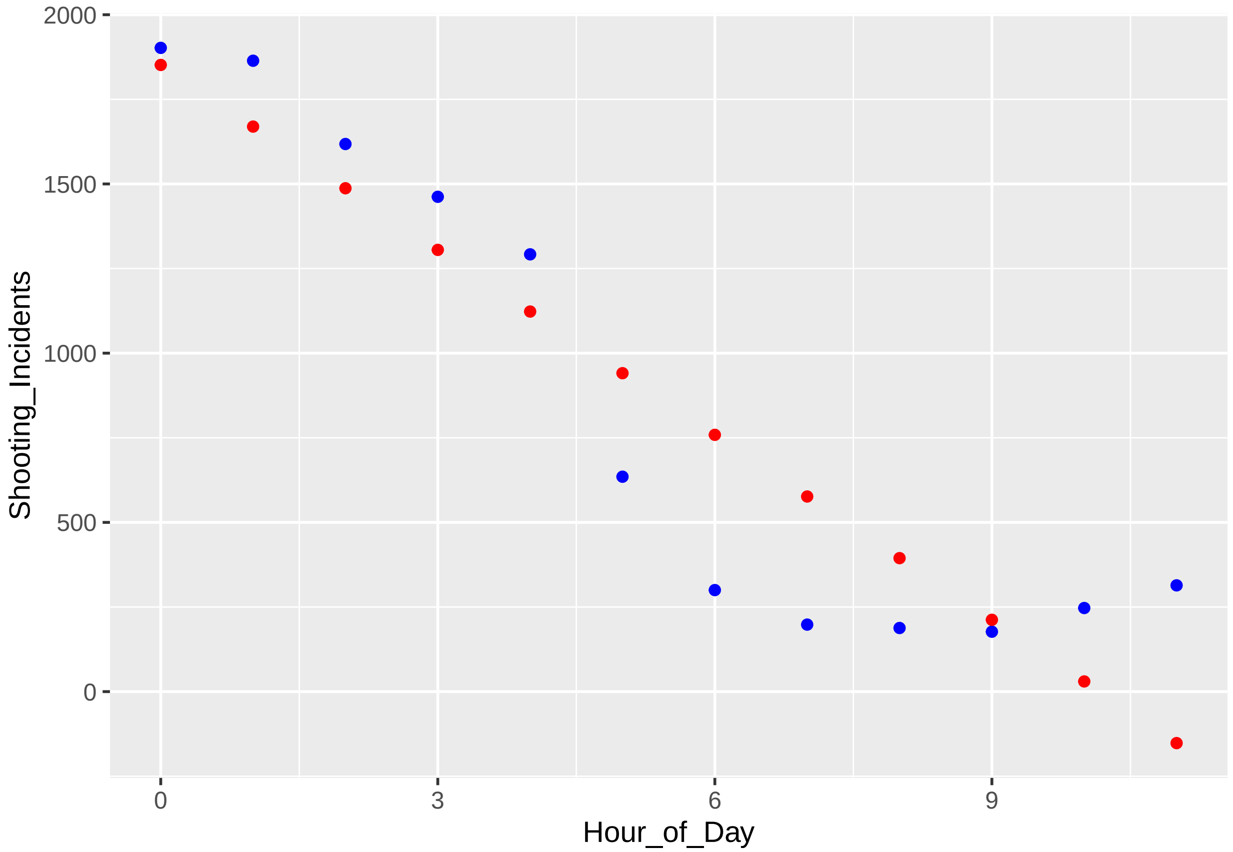
```
df_vis4_w_pred <- df_vis4 %>% mutate(Predictions = predict(mod1))


df_vis4_w_pred
```

```
## # A tibble: 12 x 3
##    Hour_of_Day Shooting_Incidents Predictions
##          <int>              <int>       <dbl>
##  1           0               1902      1852.
##  2           1               1864      1669.
##  3           2               1618      1487.
##  4           3               1462      1305.
##  5           4               1292      1123.
##  6           5                635       941.
##  7           6                300       759.
##  8           7                198       577.
##  9           8                188       394.
## 10           9                177       212.
## 11          10                247        30.1
## 12          11                314      -152.
```

6. plot the data to see how we're doing

- Note: To do this, the library(ggplot2) must be successfully loaded from the Load R Packages section at the beginning of the document.

```
df_vis4_w_pred %>% ggplot() +
  geom_point(aes(x = Hour_of_Day, y = Shooting_Incidents), color = "blue") +
  geom_point(aes(x = Hour_of_Day, y = Predictions), color = "red")
```



Second Half of the Day

1. Create a data frame for number of shootings for second half of the day

- Filter for second half of the day hours
- Group by hour of the day
- Sum the shootings
- Assign this to the data frame df_vis3

```
gb_shooting_data_clean_night <- gb_shooting_data_clean %>%
  filter(Hour_of_Day >= 12)

df_vis5 <- gb_shooting_data_clean_night %>% group_by(Hour_of_Day) %>%
  summarise(Shooting_Incidents=sum(Shooting_Incident))
df_vis5
```

```
## # A tibble: 12 x 2
##    Hour_of_Day Shooting_Incidents
##          <int>              <int>
##  1          12                412
##  2          13                440
##  3          14                679
##  4          15                767
##  5          16                872
##  6          17                908
##  7          18               1044
##  8          19               1231
##  9          20               1416
## 10          21               1710
## 11          22               1851
## 12          23               1993
```

2. Create the model

```
mod2 <- lm(Shooting_Incidents ~ Hour_of_Day, data = df_vis5)
```

3. summarize the model

```
summary(mod2)
```

```
## 
## Call:
## lm(formula = Shooting_Incidents ~ Hour_of_Day, data = df_vis5)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -139.27  -69.07    4.36   80.80  104.92 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1445.301    140.610  -10.28 1.23e-06 ***
## Hour_of_Day   146.031      7.883   18.52 4.53e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 94.27 on 10 degrees of freedom
## Multiple R-squared:  0.9717, Adjusted R-squared:  0.9689 
## F-statistic: 343.2 on 1 and 10 DF,  p-value: 4.533e-09
```

4. Add Predictions

```
df_vis5 %>% mutate(Predictions = predict(mod2))
```

```
## # A tibble: 12 x 3
##    Hour_of_Day Shooting_Incidents Predictions
##          <int>              <int>       <dbl>
##  1          12                412        307.
##  2          13                440        453.
##  3          14                679        599.
##  4          15                767        745.
##  5          16                872        891.
##  6          17                908       1037.
##  7          18               1044       1183.
##  8          19               1231       1329.
##  9          20               1416       1475.
## 10          21                 1710       1621.
## 11          22                 1851       1767.
## 12          23                 1993       1913.
```

5. create a new data set to see the predictions

- New data frame is called df_vis_w_pred

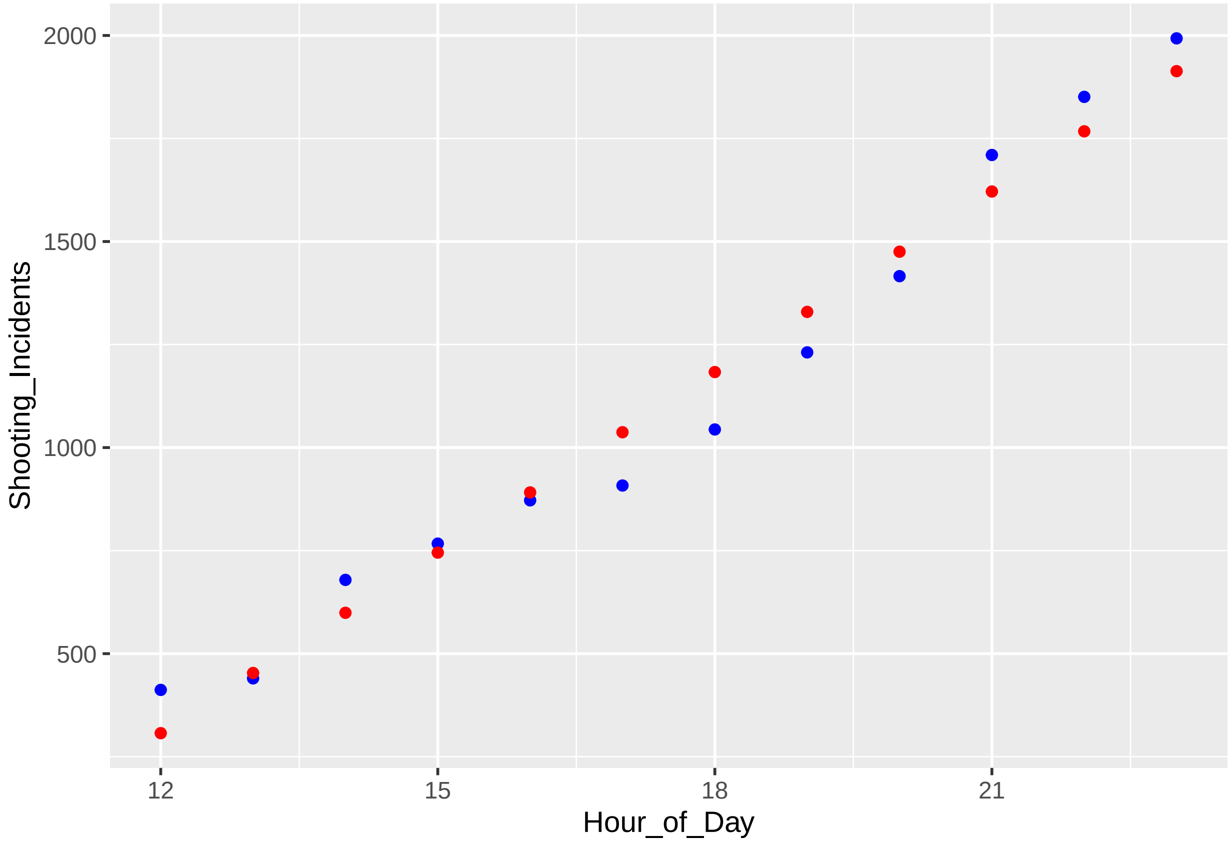
```
df_vis5_w_pred <- df_vis5 %>% mutate(Predictions = predict(mod2))


df_vis5_w_pred
```

```
## # A tibble: 12 x 3
##    Hour_of_Day Shooting_Incidents Predictions
##          <int>              <int>       <dbl>
##  1          12                412        307.
##  2          13                440        453.
##  3          14                679        599.
##  4          15                767        745.
##  5          16                872        891.
##  6          17                908       1037.
##  7          18               1044       1183.
##  8          19               1231       1329.
##  9          20               1416       1475.
## 10          21               1710       1621.
## 11          22               1851       1767.
## 12          23               1993       1913.
```

6. plot the data to see how we're doing

- Note: To do this, the library(ggplot2) must be successfully loaded from the Load R Packages section at the beginning of the document.

```
df_vis5_w_pred %>% ggplot() +
  geom_point(aes(x = Hour_of_Day, y = Shooting_Incidents), color = "blue") +
  geom_point(aes(x = Hour_of_Day, y = Predictions), color = "red")
```



Conclusion

- There appears to be a clear distinction in number of shootings by neighborhood throughout the years.
- Men are more likely to be victims of shooting incident than women
- In the first half of the day, as you approach noon, the chance of a shooting incident decreases
- In the second half of the day, as you approach midnight, the change of a shooting incident increases

Step 4: Identifying Bias

Some possible sources of bias are: 1. Selection bias 2. Confirmation bias

Selection bias occurs when the data under represents certain people or groups. In our case, the shooting data is based on government data on NYPD shootings. This doesn't take into account non citizen shootings because those people are not likely to inform or file a police report.

Confirmation bias occurs when during the analysis of data, the investigator looks for patterns of data that confirm their ideas. For me, this is an example of personal bias because I believed that Bronx would have the highest shooting incidents because I thought it was not safe through the news stories and TV shows. I mitigated this bias by checking maximums and comparing the Bronx data to other neighborhoods to ensure that the interpretation of shooting incidents is as accurate as possible.

Resources

- NYPD Shooting Data (Historic)

Appendix

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_CA.UTF-8/en_CA.UTF-8/en_CA.UTF-8/C/en_CA.UTF-8/en_CA.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0   dplyr_1.0.8
##  [5] purrr_0.3.4     readr_2.1.2     tidyr_1.2.0     tibble_3.1.6
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8       assertthat_0.2.1 digest_0.6.29    utf8_1.2.2
##  [5] R6_2.5.1         cellranger_1.1.0 backports_1.4.1  reprex_2.0.1
##  [9] evaluate_0.15    httr_1.4.2       highr_0.9        pillar_1.7.0
## [13] rlang_1.0.1      curl_4.3.2       readxl_1.3.1     rstudioapi_0.13
## [17] rmarkdown_2.11   labeling_0.4.2   bit_4.0.4        munsell_0.5.0
## [21] broom_0.7.12     compiler_4.1.2   modelr_0.1.8     xfun_0.29
## [25] pkgconfig_2.0.3  htmltools_0.5.2  tidyselect_1.1.2 fansi_1.0.2
## [29] crayon_1.5.0     tzdb_0.2.0       dbplyr_2.1.1     withr_2.4.3
## [33] grid_4.1.2       jsonlite_1.7.3   gtable_0.3.0     lifecycle_1.0.1
## [37] DBI_1.1.2        magrittr_2.0.2   scales_1.1.1     cli_3.2.0
## [41] stringi_1.7.6    vroom_1.5.7      farver_2.1.0     fs_1.5.2
## [45] xml2_1.3.3       ellipsis_0.3.2   generics_0.1.2   vctrs_0.3.8
## [49] tools_4.1.2      bit64_4.0.5      glue_1.6.1       hms_1.1.1
## [53] parallel_4.1.2   fastmap_1.1.0    yaml_2.3.5       colorspace_2.0-3
## [57] rvest_1.0.2      knitr_1.37       haven_2.4.3
```