NYPD Shooting Incident Data Report

Data Science Student

06/03/2023

List of every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year.

This is analysis of NYPD Shooting Incident Data. The objective of this exercise is to analyze the data and try to answer some questions

Step 0: Import Library

```
# install.packages("tidyverse")
library(tidyverse)
library(lubridate)
```

Step 1: Load Data

- `read_csv()` reads comma delimited files, read_csv2() reads semicolon separated files (common in countries where , is used as the decimal place), read_tsv() reads tab delimited files, and read_delim() reads in files with any delimiter.

```
df = read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")
```

```
## Rows: 25596 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.
```

```
head(df)
```

```
## # A tibble: 6 x 19
##   INCIDE~1 OCCUR~2 OCCUR~3 BORO  PRECI~4 JURIS~5 LOCAT~6 STATI~7 PERP_~8 PERP_~9
##      <dbl> <chr>   <time>  <chr>   <dbl>   <dbl> <chr>   <lgl>   <chr>   <chr>
## 1   2.36e8 11/11/~ 15:04   BROO~      79       0 <NA>    FALSE   <NA>    <NA>
## 2   2.31e8 07/16/~ 22:05   BROO~      72       0 <NA>    FALSE   45-64   M
## 3   2.31e8 07/11/~ 01:09   BROO~      79       0 <NA>    FALSE   <18     M
## 4   2.38e8 12/11/~ 13:42   BROO~      81       0 <NA>    FALSE   <NA>    <NA>
## 5   2.24e8 02/16/~ 20:00   QUEE~     113       0 <NA>    FALSE   <NA>    <NA>
## 6   2.28e8 05/15/~ 04:13   QUEE~     113       0 <NA>    TRUE    <NA>    <NA>
## # ... with 9 more variables: 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>, and abbreviated variable
## #   names 1: INCIDENT_KEY, 2: OCCUR_DATE, 3: OCCUR_TIME, 4: PRECINCT,
## #   5: JURISDICTION_CODE, 6: LOCATION_DESC, 7: STATISTICAL_MURDER_FLAG,
## #   8: PERP_AGE_GROUP, 9: PERP_SEX
```

Step 2: Tidy and Transform Data

Let's first eliminate the columns I do not need for this assignment, which are: **PRECINCT**,**JURISDICTION_CODE**,**LO** **X_COORD_CD**, **Y_COORD_CD**, and **Lon_Lat**.

```r
df_2 = df %>% select(INCIDENT_KEY,
                   OCCUR_DATE,
                   OCCUR_TIME,
                   BORO,
                   STATISTICAL_MURDER_FLAG,
                   PERP_AGE_GROUP,
                   PERP_SEX,
                   PERP_RACE,
                   VIC_AGE_GROUP,
                   VIC_SEX,
                   VIC_RACE,
                   Latitude,
                   Longitude)

# Return the column name along with the missing values
lapply(df_2, function(x) sum(is.na(x)))
```

```
## $INCIDENT_KEY
## [1] 0
##
## $OCCUR_DATE
## [1] 0
##
## $OCCUR_TIME
## [1] 0
##
## $BORO
## [1] 0
##
## $STATISTICAL_MURDER_FLAG
## [1] 0
##
## $PERP_AGE_GROUP
## [1] 9344
##
## $PERP_SEX
## [1] 9310
```

```
##
## $PERP_RACE
## [1] 9310
##
## $VIC_AGE_GROUP
## [1] 0
##
## $VIC_SEX
## [1] 0
##
## $VIC_RACE
## [1] 0
##
## $Latitude
## [1] 0
##
## $Longitude
## [1] 0
```

Understanding the reasons why data are missing is important for handling the remaining data correctly. There's a fair amount of unidentifiable data on perpetrators (age, race, or sex.) Those cases are possibly still active and ongoing investigation. In fear of missing meaningful information, I handle this group of missing data by calling them as another group of "Unknown".

Key observations on data type conversion are:

- **INCIDENT_KEY** should be treated as a string.
- **BORO** should be treated as a factor.
- **PERP_AGE_GROUP** should be treated as a factor.
- **PERP_SEX** should be treated as a factor.
- **PERP_RACE** should be treated as a factor.
- **VIC_AGE_GROUP** should be treated as a factor.
- **VIC_SEX** should be treated as a factor.
- **VIC_RACE** should be treated as a factor.

```r
# Tidy and transform data
df_2 = df_2 %>%
  replace_na(list(PERP_AGE_GROUP = "Unknown", PERP_SEX = "Unknown", PERP_RACE = "Unknown"))

# Remove extreme values in data
df_2 = subset(df_2, PERP_AGE_GROUP!="1020" & PERP_AGE_GROUP!="224" & PERP_AGE_GROUP!="940")

df_2$PERP_AGE_GROUP = recode(df_2$PERP_AGE_GROUP, UNKNOWN = "Unknown")
df_2$PERP_SEX = recode(df_2$PERP_SEX, U = "Unknown")
df_2$PERP_RACE = recode(df_2$PERP_RACE, UNKNOWN = "Unknown")
df_2$VIC_SEX   = recode(df_2$VIC_SEX, U = "Unknown")
df_2$VIC_RACE   = recode(df_2$VIC_RACE, UNKNOWN = "Unknown")
df_2$INCIDENT_KEY = as.character(df_2$INCIDENT_KEY)
df_2$BORO = as.factor(df_2$BORO)
df_2$PERP_AGE_GROUP = as.factor(df_2$PERP_AGE_GROUP)
df_2$PERP_SEX = as.factor(df_2$PERP_SEX)
df_2$PERP_RACE = as.factor(df_2$PERP_RACE)
df_2$VIC_AGE_GROUP = as.factor(df_2$VIC_AGE_GROUP)
df_2$VIC_SEX = as.factor(df_2$VIC_SEX)
```

```
df_2$VIC_RACE = as.factor(df_2$VIC_RACE)

# Return summary statistics
summary(df_2)
```

```
##  INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME                  BORO
##  Length:25593       Length:25593       Length:25593      BRONX        : 7400
##  Class :character   Class :character   Class1:hms        BROOKLYN     :10364
##  Mode  :character   Mode  :character   Class2:difftime   MANHATTAN    : 3265
##                                        Mode  :numeric    QUEENS       : 3828
##                                                          STATEN ISLAND:  736
##
##
##  STATISTICAL_MURDER_FLAG PERP_AGE_GROUP     PERP_SEX
##  Mode :logical           <18    : 1463   F      :  371
##  FALSE:20665             18-24  : 5844   M      :14413
##  TRUE :4928              25-44  : 5202   Unknown:10809
##                          45-64  :  535
##                          65+    :   57
##                          Unknown:12492
##
##                           PERP_RACE     VIC_AGE_GROUP      VIC_SEX
##  AMERICAN INDIAN/ALASKAN NATIVE:    2   <18    : 2681   F      : 2403
##  ASIAN / PACIFIC ISLANDER      :  141   18-24  : 9603   M      :23179
##  BLACK                         :10667   25-44  :11384   Unknown:   11
##  BLACK HISPANIC                : 1203   45-64  : 1698
##  Unknown                       :11146   65+    :  167
##  WHITE                         :  272   UNKNOWN:   60
##  WHITE HISPANIC                : 2162
##                            VIC_RACE        Latitude       Longitude
##  AMERICAN INDIAN/ALASKAN NATIVE:    9   Min.   :40.51   Min.   :-74.25
##  ASIAN / PACIFIC ISLANDER      :  354   1st Qu.:40.67   1st Qu.:-73.94
##  BLACK                         :18280   Median :40.70   Median :-73.92
##  BLACK HISPANIC                : 2485   Mean   :40.74   Mean   :-73.91
##  Unknown                       :   65   3rd Qu.:40.82   3rd Qu.:-73.88
##  WHITE                         :  660   Max.   :40.91   Max.   :-73.70
##  WHITE HISPANIC                : 3740
```

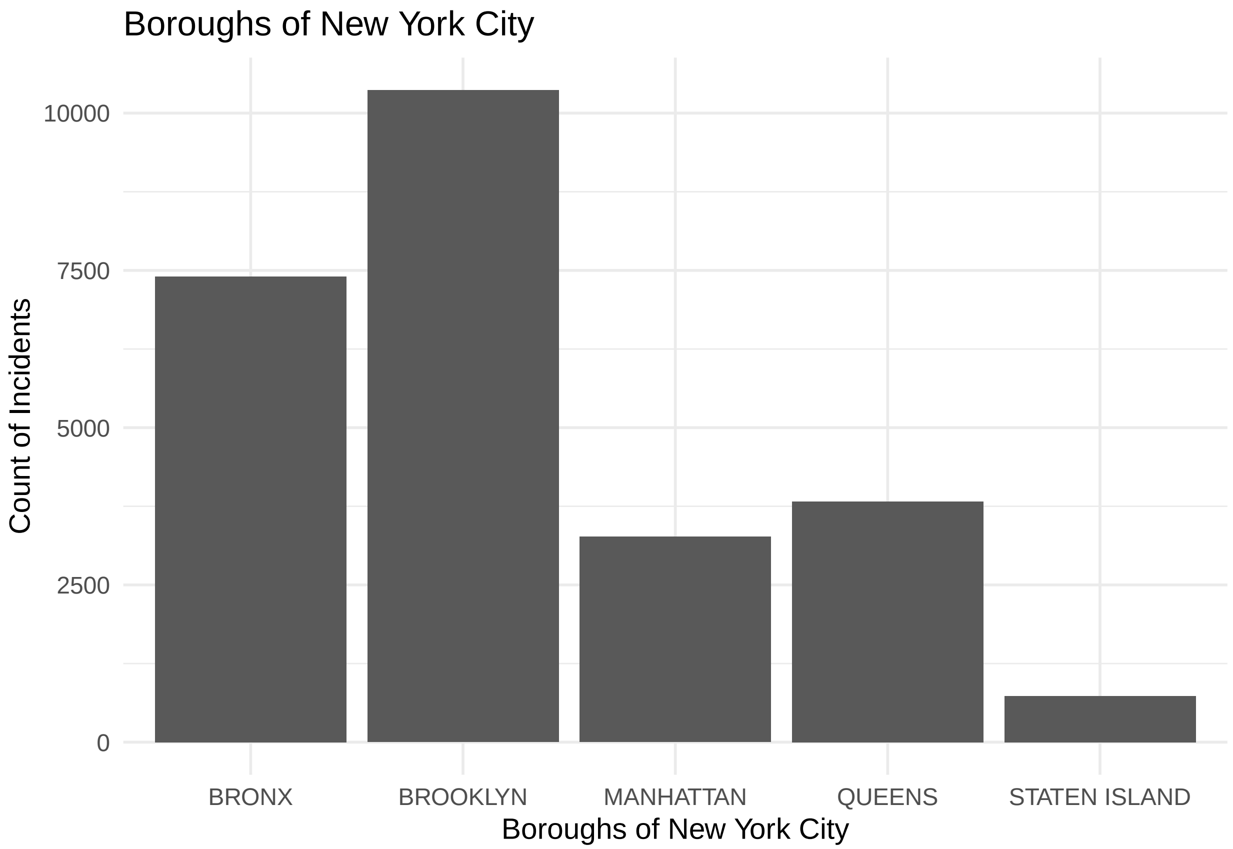
Step 3: Add Visualizations and Analysis

Question

1. Which part of New York has the most number of incidents? Of those incidents, how many are murder cases?

Brooklyn is the 1st in terms of the number of incidents, followed by Bronx and Queens respectively. Likewise, the number of murder cases follows the same pattern as that of incidents.

```
g <- ggplot(df_2, aes(x = BORO)) +
  geom_bar() +
  labs(title = "Boroughs of New York City",
       x = "Boroughs of New York City",
       y = "Count of Incidents") +
  theme_minimal()
g
```



```
table(df_2$BORO, df_2$STATISTICAL_MURDER_FLAG)
```

```
##
##                   FALSE TRUE
##   BRONX            5983 1417
##   BROOKLYN         8344 2020
##   MANHATTAN        2691  574
##   QUEENS           3066  762
##   STATEN ISLAND     581  155
```

2. Which day and time should people in New York be cautious of falling into victims of crime?

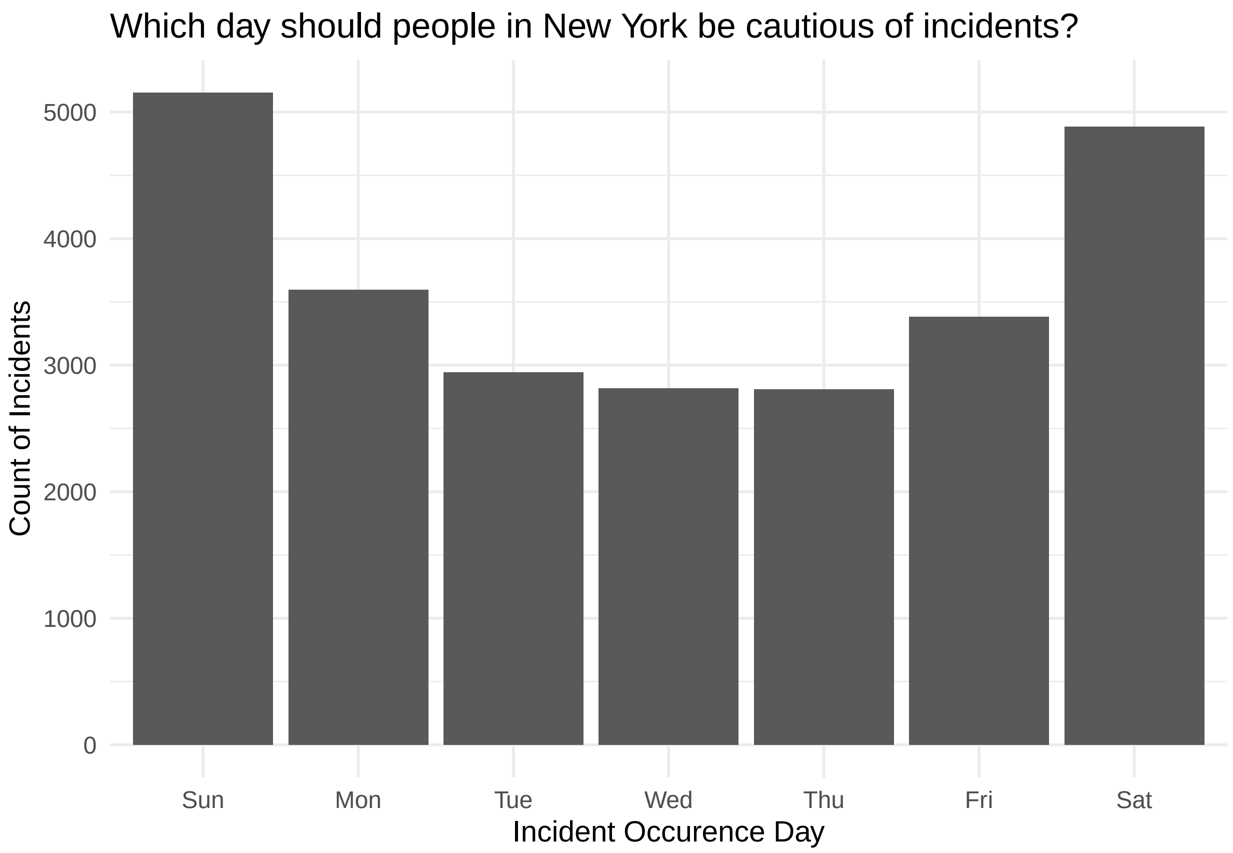
- Weekends in NYC have the most chances of incidents. Be cautious!
- Incidents historically happen in the evening and night time. If there's nothing urgent, recommend people staying at home!

```
df_2$OCCUR_DAY = mdy(df_2$OCCUR_DATE)
df_2$OCCUR_DAY = wday(df_2$OCCUR_DAY, label = TRUE)
df_2$OCCUR_HOUR = hour(hms(as.character(df_2$OCCUR_TIME)))
```

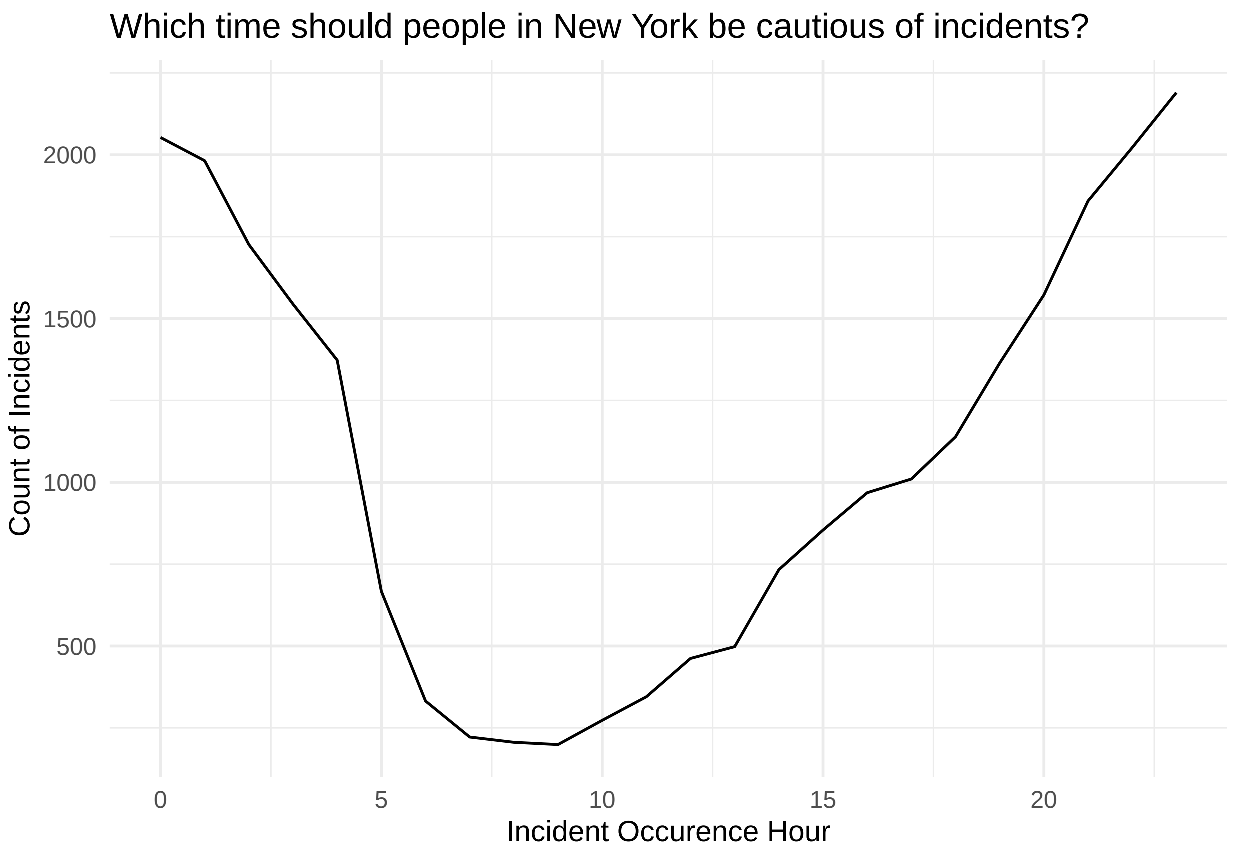
```
df_3 = df_2 %>%
  group_by(OCCUR_DAY) %>%
  count()

df_4 = df_2 %>%
  group_by(OCCUR_HOUR) %>%
  count()
```

```
g <- ggplot(df_3, aes(x = OCCUR_DAY, y = n)) +
  geom_col() +
  labs(title = "Which day should people in New York be cautious of incidents?",
       x = "Incident Occurence Day",
       y = "Count of Incidents") +
  theme_minimal()
g
```



```
g <- ggplot(df_4, aes(x = OCCUR_HOUR, y = n)) +
  geom_line() +
  labs(title = "Which time should people in New York be cautious of incidents?",
       x = "Incident Occurence Hour",
       y = "Count of Incidents") +
  theme_minimal()
g
```



4.Modeling It will be interesting to find out if a specific BORO has more importance on the number of incidences. In order to identify this significance, a linear regression model is created to find the cooeficients of BORO values on incidences. In order to do this, new dataframe is created with number of incidences

```
# Linear Model
nypd_trim_data <- df %>% group_by(OCCUR_DATE,BORO,PERP_AGE_GROUP, PERP_SEX, PERP_RACE,VIC_AGE_GROUP)
nypd_model <-lm(INCIDENT_KEY ~ BORO, data = nypd_trim_data)
summary(nypd_model)
```

```
##
## Call:
## lm(formula = INCIDENT_KEY ~ BORO, data = nypd_trim_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -106879954  -51495573  -25431599   54036083  129356330
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        114492061     788113 145.274  < 2e-16 ***
## BOROBROOKLYN        -5360598    1031836  -5.195 2.06e-07 ***
## BOROMANHATTAN        2341145    1424518   1.643    0.100
## BOROQUEENS           -981845    1349871  -0.727    0.467
## BOROSTATEN ISLAND   -3145739    2620646  -1.200    0.230
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 67810000 on 25591 degrees of freedom
## Multiple R-squared:  0.001806,   Adjusted R-squared:  0.00165
## F-statistic: 11.57 on 4 and 25591 DF,  p-value: 2.184e-09
```

Analysis: Since BORO is a factor, the first BORO that is BRONX is considered as Intercept. Note that this was something I had to dig and find out because I was confused why BRONX was not showing up. It is clear from p-values that BRONX, BROOKLYN and MANHATTAN maybe having similar impact on number of incidences. That is a person being in these BOROs could make a difference to the number of incidences.

Step 4: Identify Bias

When I saw this subject, I wanted to avoid any inference based on Race to avoid any internal biases that I might have. Also, I avoided using any data that could have missing information or Unknown data since more bias could be introduced owing to the same. During some internal analysis, I did observe not defining Perpetrator Sex (Unknown) actually could lead to misleading information. Also, the linear regression model's interpretation based on such simple data may not show the entire picture. More complex features and data need to be added to remove disturbing noises from interpretations. For instance adding proper perpertrator and victim information could give more insightful information. However, we need more clear data for the same.

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.2 forcats_1.0.0   stringr_1.5.0   dplyr_1.1.0
##  [5] purrr_1.0.1     readr_2.1.4     tidyr_1.3.0     tibble_3.1.8
##  [9] ggplot2_3.4.1   tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.2.1    digest_0.6.30       utf8_1.2.3
##  [4] R6_2.5.1            cellranger_1.1.0    backports_1.4.1
##  [7] reprex_2.0.2        evaluate_0.17       highr_0.9
## [10] httr_1.4.4          pillar_1.8.1        rlang_1.0.6
## [13] googlesheets4_1.0.1 curl_5.0.0          readxl_1.4.2
## [16] rstudioapi_0.14     rmarkdown_2.17      labeling_0.4.2
## [19] googledrive_2.0.0   bit_4.0.5           munsell_0.5.0
## [22] broom_1.0.3         compiler_4.2.1      modelr_0.1.10
## [25] xfun_0.34           pkgconfig_2.0.3     htmltools_0.5.3
## [28] tidyselect_1.2.0    fansi_1.0.4         crayon_1.5.2
## [31] tzdb_0.3.0         dbplyr_2.3.0       withr_2.5.0
## [34] grid_4.2.1         jsonlite_1.8.3     gtable_0.3.1
## [37] lifecycle_1.0.3    DBI_1.1.3          magrittr_2.0.3
## [40] scales_1.2.1       cli_3.6.0          stringi_1.7.8
## [43] vroom_1.6.1        farver_2.1.1       fs_1.5.2
## [46] xml2_1.3.3         ellipsis_0.3.2     generics_0.1.3
## [49] vctrs_0.5.2        tools_4.2.1        bit64_4.0.5
## [52] glue_1.6.2         hms_1.1.2          parallel_4.2.1
## [55] fastmap_1.1.0      yaml_2.3.6         timechange_0.2.0
## [58] colorspace_2.1-0   gargle_1.3.0       rvest_1.0.3
## [61] knitr_1.40         haven_2.5.1
```