NYPD Shooting Incident Data Report

2022-12-04

Step 1 - Identify and import the data

We start by reading the "tidyverse" library and read data from the csv file.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
url <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
```

Let's read in the data and see what we have.

```
NYPD <- read_csv(url)
NYPD
```

```
## # A tibble: 25,596 x 19
##    INCID~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>
##  7  2.27e8 04/14/~ 21:08   BRONX      42       0 COMMER~ TRUE    <NA>    <NA>
##  8  2.38e8 12/10/~ 19:30   BRONX      52       0 <NA>    FALSE   <NA>    <NA>
##  9  2.25e8 02/22/~ 00:18   MANH~      34       0 <NA>    FALSE   <NA>    <NA>
## 10  2.25e8 03/07/~ 06:15   BROO~      75       0 <NA>    TRUE    25-44   M
## # ... with 25,586 more rows, 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 Your Data

After looking at the shooting data, we would like to tidy this data set. We only need OCCUR_DATE, BORO, PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP, VIC_SEX, VIC_RACE for the analysis I am planning.

```
NYPD <- NYPD %>%
  select(c(OCCUR_DATE, BORO, PERP_AGE_GROUP, PERP_SEX, PERP_RACE, VIC_AGE_GROUP, VIC_SEX, VIC_RACE))
NYPD
```

```
## # A tibble: 25,596 x 8
##    OCCUR_DATE BORO      PERP_AGE_GROUP PERP_SEX PERP_R~1 VIC_A~2 VIC_SEX VIC_R~3
##    <chr>      <chr>     <chr>          <chr>    <chr>    <chr>   <chr>   <chr>
##  1 11/11/2021 BROOKLYN  <NA>           <NA>     <NA>     18-24   M       BLACK
##  2 07/16/2021 BROOKLYN  45-64          M        ASIAN /~ 25-44   M       ASIAN ~
##  3 07/11/2021 BROOKLYN  <18            M        BLACK    25-44   M       BLACK
##  4 12/11/2021 BROOKLYN  <NA>           <NA>     <NA>     25-44   M       BLACK
##  5 02/16/2021 QUEENS    <NA>           <NA>     <NA>     25-44   M       BLACK
##  6 05/15/2021 QUEENS    <NA>           <NA>     <NA>     25-44   M       BLACK
##  7 04/14/2021 BRONX     <NA>           <NA>     <NA>     18-24   M       BLACK
##  8 12/10/2021 BRONX     <NA>           <NA>     <NA>     25-44   M       BLACK
##  9 02/22/2021 MANHATTAN <NA>           <NA>     <NA>     25-44   M       BLACK ~
## 10 03/07/2021 BROOKLYN  25-44          M        BLACK H~ 25-44   M       WHITE ~
## # ... with 25,586 more rows, and abbreviated variable names 1: PERP_RACE,
## #   2: VIC_AGE_GROUP, 3: VIC_RACE
```

We will also read the "lubridate" library and change OCCUR_Date to date format.

```
library(lubridate)
```

```
## Loading required package: timechange
```

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

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

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

```
## # A tibble: 25,596 x 8
##    OCCUR_DATE BORO      PERP_AGE_GROUP PERP_SEX PERP_R~1 VIC_A~2 VIC_SEX VIC_R~3
##    <date>     <chr>     <chr>          <chr>    <chr>    <chr>   <chr>   <chr>
##  1 2021-11-11 BROOKLYN  <NA>           <NA>     <NA>     18-24   M       BLACK
##  2 2021-07-16 BROOKLYN  45-64          M        ASIAN /~ 25-44   M       ASIAN ~
##  3 2021-07-11 BROOKLYN  <18            M        BLACK    25-44   M       BLACK
##  4 2021-12-11 BROOKLYN  <NA>           <NA>     <NA>     25-44   M       BLACK
##  5 2021-02-16 QUEENS    <NA>           <NA>     <NA>     25-44   M       BLACK
```

```
##  6 2021-05-15 QUEENS    <NA>           <NA>     <NA>     25-44   M       BLACK
##  7 2021-04-14 BRONX     <NA>           <NA>     <NA>     18-24   M       BLACK
##  8 2021-12-10 BRONX     <NA>           <NA>     <NA>     25-44   M       BLACK
##  9 2021-02-22 MANHATTAN <NA>           <NA>     <NA>     25-44   M       BLACK ~
## 10 2021-03-07 BROOKLYN  25-44          M        BLACK H~ 25-44   M       WHITE ~
## # ... with 25,586 more rows, and abbreviated variable names 1: PERP_RACE,
## #   2: VIC_AGE_GROUP, 3: VIC_RACE
```

Here, we are running summary command to view a summary of our columns.

```
summary(NYPD)
```

```
##    OCCUR_DATE             BORO           PERP_AGE_GROUP       PERP_SEX
##  Min.   :2006-01-01   Length:25596       Length:25596       Length:25596
##  1st Qu.:2009-05-10   Class :character   Class :character   Class :character
##  Median :2012-08-26   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2013-06-13
##  3rd Qu.:2017-07-01
##  Max.   :2021-12-31
##   PERP_RACE         VIC_AGE_GROUP        VIC_SEX            VIC_RACE
##  Length:25596       Length:25596       Length:25596       Length:25596
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

Step 3 - Add Visualizations and Analysis

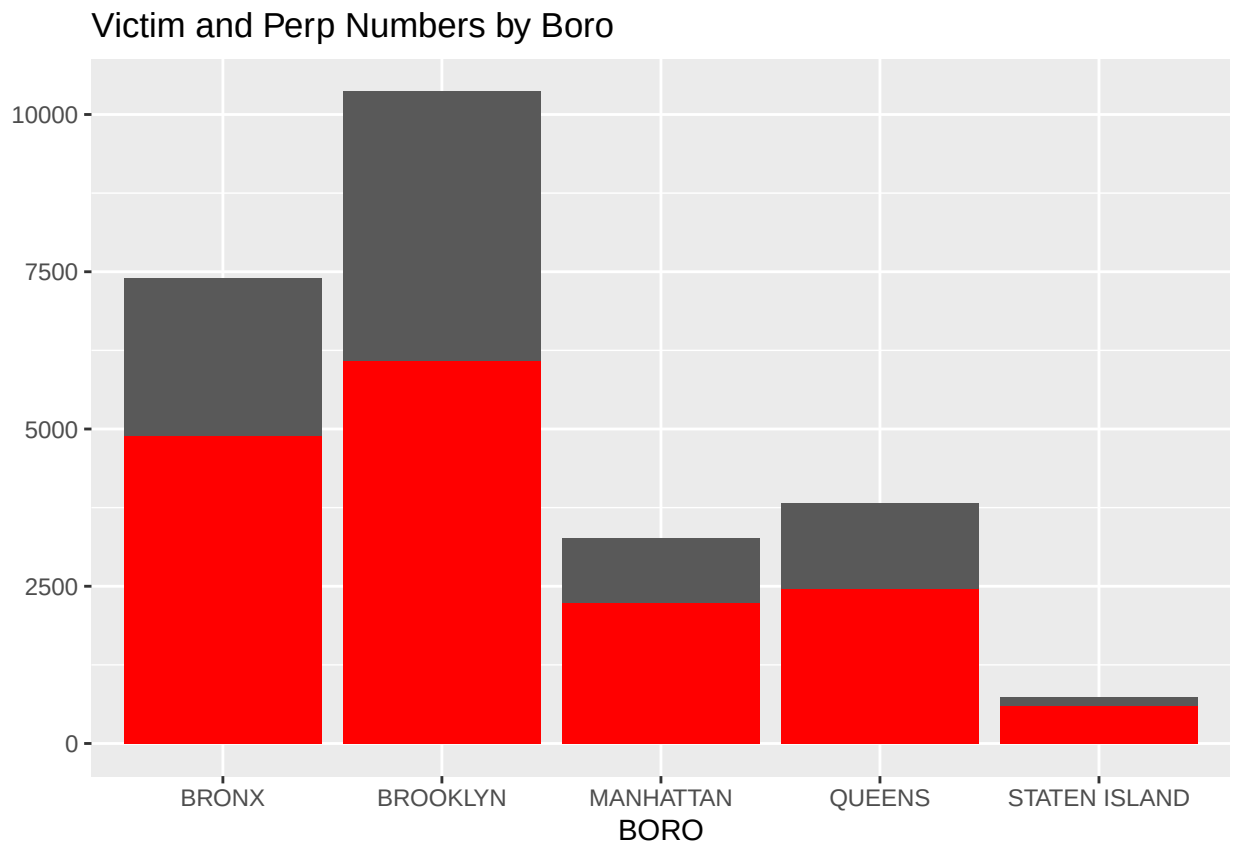
We will first group our data by BORO to see for each BORO, numbers of perpetrators and victims.

```
NYPD_by_boro <- NYPD %>%
  group_by(BORO) %>%
  summarize(perp_num = sum(!is.na(PERP_AGE_GROUP)), vic_num = n()) %>%
  ungroup()
NYPD_by_boro
```

```
## # A tibble: 5 x 3
##   BORO          perp_num vic_num
##   <chr>            <int>   <int>
## 1 BRONX             4890    7402
## 2 BROOKLYN          6074   10365
## 3 MANHATTAN         2235    3265
## 4 QUEENS            2462    3828
## 5 STATEN ISLAND      591     736
```

Now, let's plot by bar chat. Red bars show victims with perpetrators and grey bars show victims without perpetrators. Here we see in our 5 boros, Brooklyn has the most number of victims. It also has the most numbers of victim without finding perpetrators.

```
NYPD_by_boro %>%
  ggplot(aes(x = BORO)) +
  geom_bar(aes(y = vic_num), stat='identity') +
  geom_bar(aes(y = perp_num), stat='identity', fill = "red") +
  labs(title = str_c("Victim and Perp Numbers by Boro"), y = NULL)
```



Secondly, we would like to see for Brooklyn only, the trends of perpetrator numbers and victim numbers by year.

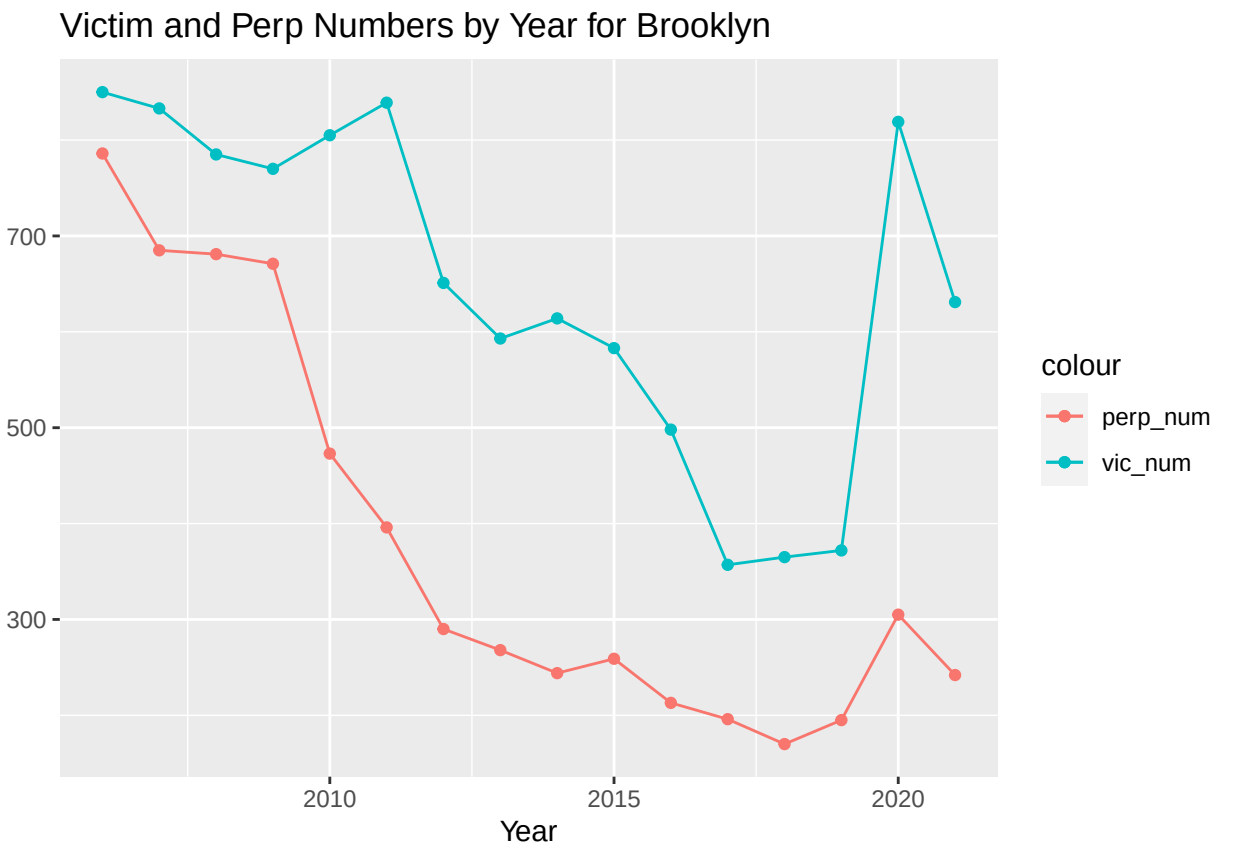
```
NYPD_by_year_Brooklyn <- NYPD %>%
  filter(BORO == "BROOKLYN") %>%
  mutate(Year = year(OCCUR_DATE)) %>%
  group_by(Year) %>%
  summarize(perp_num = sum(!is.na(PERP_AGE_GROUP)), vic_num = n()) %>%
  ungroup()
NYPD_by_year_Brooklyn
```

```
## # A tibble: 16 x 3
##     Year perp_num vic_num
##    <dbl>    <int>   <int>
##  1  2006      786     850
##  2  2007      685     833
##  3  2008      681     785
##  4  2009      671     770
##  5  2010      473     805
```

```
##  6  2011       396      839
##  7  2012       290      651
##  8  2013       268      593
##  9  2014       244      614
## 10  2015       259      583
## 11  2016       213      498
## 12  2017       196      357
## 13  2018       170      365
## 14  2019       195      372
## 15  2020       305      819
## 16  2021       242      631
```

And we will plot our second graph. Number of victims are higher from 2006 to 2011, getting lower in the following years and return to peak on 2020. Number of perpetrators are almost following the same trends, and for 2020, there are lots of victims without perpetrators comparing to other years.

```
NYPD_by_year_Brooklyn %>%
  ggplot(aes(x = Year, y = perp_num)) +
  geom_line(aes(color = "perp_num")) +
  geom_point(aes(color = "perp_num")) +
  geom_line(aes(y = vic_num, color = "vic_num")) +
  geom_point(aes(y = vic_num, color = "vic_num")) +
  labs(title = str_c("Victim and Perp Numbers by Year for Brooklyn"), y = NULL)
```



We could model our data for perp numbers vs victim numbers. It shows 2017 had the least number of victims and 2006 had the most.

```
mod <- lm(perp_num ~ vic_num, data = NYPD_by_year_Brooklyn)
summary(mod)
```

```
##
## Call:
## lm(formula = perp_num ~ vic_num, data = NYPD_by_year_Brooklyn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -230.78  -95.59  -39.99   95.35  221.95
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -211.2890   135.2018  -1.563 0.140424
## vic_num        0.9122     0.2017   4.521 0.000479 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 138.5 on 14 degrees of freedom
## Multiple R-squared:  0.5935, Adjusted R-squared:  0.5645
## F-statistic: 20.44 on 1 and 14 DF,  p-value: 0.0004792
```

```
NYPD_by_year_Brooklyn %>% slice_min(vic_num)
```

```
## # A tibble: 1 x 3
##    Year perp_num vic_num
##   <dbl>    <int>   <int>
## 1  2017      196     357
```

```
NYPD_by_year_Brooklyn %>% slice_max(vic_num)
```

```
## # A tibble: 1 x 3
##    Year perp_num vic_num
##   <dbl>    <int>   <int>
## 1  2006      786     850
```

We can predict number of perpetrator by using our model.

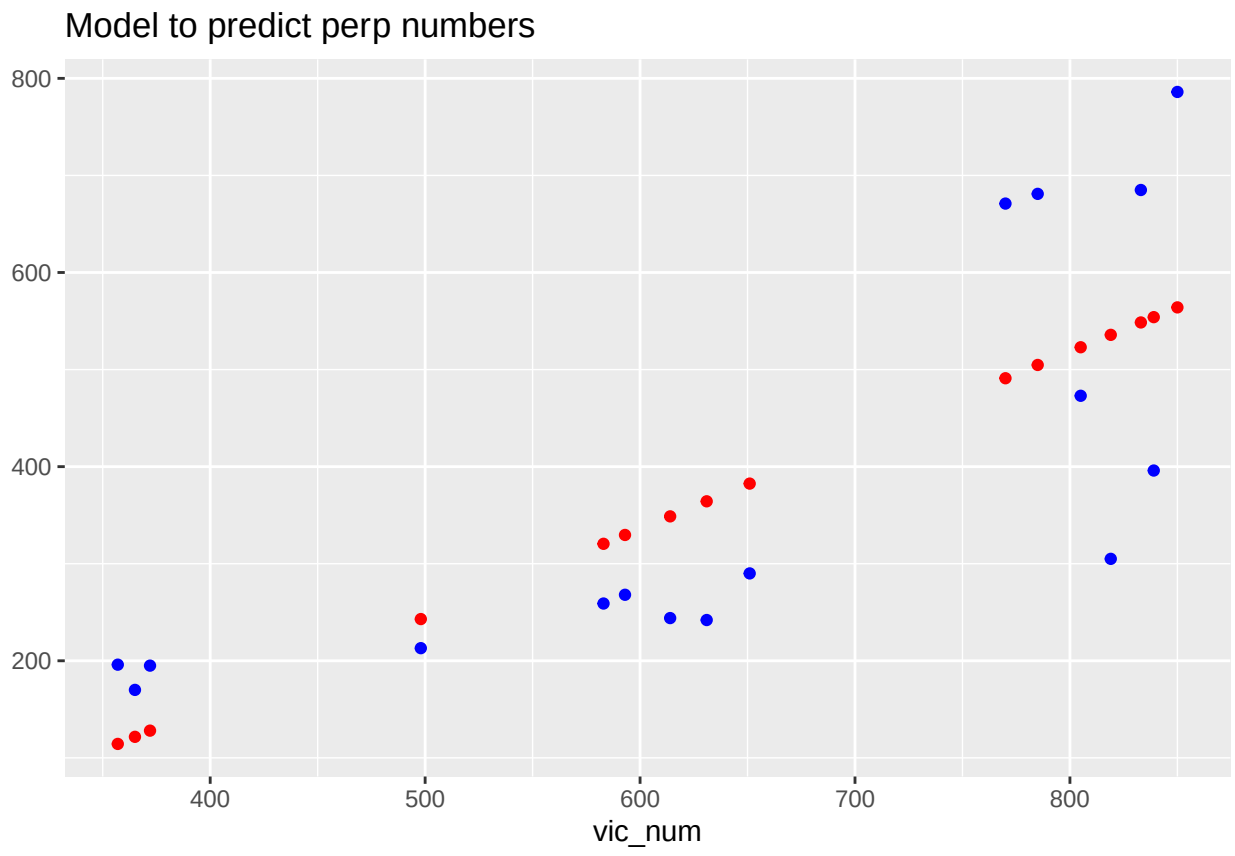
```
NYPD_by_year_Brooklyn_pred <- NYPD_by_year_Brooklyn %>% mutate(pred = predict(mod))
NYPD_by_year_Brooklyn_pred
```

```
## # A tibble: 16 x 4
##     Year perp_num vic_num  pred
##    <dbl>    <int>   <int> <dbl>
##  1  2006      786     850  564.
##  2  2007      685     833  549.
##  3  2008      681     785  505.
##  4  2009      671     770  491.
##  5  2010      473     805  523.
##  6  2011      396     839  554.
```

```
##  7  2012      290     651  383.
##  8  2013      268     593  330.
##  9  2014      244     614  349.
## 10  2015      259     583  321.
## 11  2016      213     498  243.
## 12  2017      196     357  114.
## 13  2018      170     365  122.
## 14  2019      195     372  128.
## 15  2020      305     819  536.
## 16  2021      242     631  364.
```

And comparing with the actual number of perpetrator using plot comparison, blue dots are actual number of perp and red dots are estimated.

```
NYPD_by_year_Brooklyn_pred %>%
  ggplot() +
  geom_point(aes(x = vic_num, y = perp_num), color = "blue") +
  geom_point(aes(x = vic_num, y = pred), color = "red") +
  labs(title = str_c("Model to predict perp numbers"), y = NULL)
```



Step 4 - Add Bias Identification

Our model illustrates the actual number of preps are very close to the predicted number of preps. There is a positive connections between number of victims and number of perpetrators.

The potential bias on my analysis would be the data accuracy on the data source. The analysis that I made was based on considering all NA in perp columns meaning the police didn't found perp for those victims, but it could possible those shooting cases are closed and perps are found, they just don't want to public the perp info because of privacy issue.

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.0  timechange_0.1.1 forcats_0.5.2    stringr_1.5.0
##  [5] dplyr_1.0.10     purrr_0.3.5      readr_2.1.3      tidyr_1.2.1
##  [9] tibble_3.1.8     ggplot2_3.4.0    tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.2.1    digest_0.6.30       utf8_1.2.2
##  [4] R6_2.5.1            cellranger_1.1.0    backports_1.4.1
##  [7] reprex_2.0.2        evaluate_0.18       highr_0.9
## [10] httr_1.4.4          pillar_1.8.1        rlang_1.0.6
## [13] googlesheets4_1.0.1 curl_4.3.3          readxl_1.4.1
## [16] rstudioapi_0.14     rmarkdown_2.18      labeling_0.4.2
## [19] googledrive_2.0.0   bit_4.0.5           munsell_0.5.0
## [22] broom_1.0.1         compiler_4.2.2      modelr_0.1.10
## [25] xfun_0.35           pkgconfig_2.0.3     htmltools_0.5.3
## [28] tidyselect_1.2.0    fansi_1.0.3         crayon_1.5.2
## [31] tzdb_0.3.0          dbplyr_2.2.1        withr_2.5.0
## [34] grid_4.2.2          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_3.4.1           stringi_1.7.8
## [43] vroom_1.6.0         farver_2.1.1        fs_1.5.2
## [46] xml2_1.3.3          ellipsis_0.3.2      generics_0.1.3
## [49] vctrs_0.5.1         tools_4.2.2         bit64_4.0.5
## [52] glue_1.6.2          hms_1.1.2           parallel_4.2.2
## [55] fastmap_1.1.0       yaml_2.3.6          colorspace_2.0-3
## [58] gargle_1.2.1        rvest_1.0.3         knitr_1.41
## [61] haven_2.5.1
```