NYPD Shooting Data

C. Godsey

2023-04-01

NYPD Shooting Trends

Introduction

We're going to explore the provided data to see if we can identify any interesting trends in NYC shootings.

Data

We're using a data set published by the city of New York that gives per-incident shooting data for the entire city over several years. Here is a sample of the raw data:

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

```
## # A tibble: 6 x 21
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     LOC_OF_OCCUR_DESC PRECINCT
##          <dbl> <chr>      <time>     <chr>    <chr>                <dbl>
## 1    228798151 05/27/2021 21:30      QUEENS   <NA>                   105
## 2    137471050 06/27/2014 17:40      BRONX    <NA>                    40
## 3    147998800 11/21/2015 03:56      QUEENS   <NA>                   108
## 4    146837977 10/09/2015 18:30      BRONX    <NA>                    44
## 5     58921844 02/19/2009 22:58      BRONX    <NA>                    47
## 6    219559682 10/21/2020 21:36      BROOKLYN <NA>                    81
## # i 15 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   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>
```

Tidy

We tidy the data to include more specific temporal information. We aggregate the data to provide the number of incidents per month and year, as well as the month within the year.

```
shooting_data_raw <- shooting_data_raw_csv %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE)) %>%
  mutate(MONTH_DATE = floor_date(OCCUR_DATE, "month"),
         MONTH = month(OCCUR_DATE))
invisible(shooting_data_raw)
summary(shooting_data_raw)
```

```
##   INCIDENT_KEY         OCCUR_DATE           OCCUR_TIME           BORO
##  Min.   :  9953245   Min.   :2006-01-01   Length:27312      Length:27312
##  1st Qu.: 63860880   1st Qu.:2009-07-18   Class1:hms        Class :character
##  Median : 90372218   Median :2013-04-29   Class2:difftime   Mode  :character
##  Mean   :120860536   Mean   :2014-01-06   Mode  :numeric
##  3rd Qu.:188810230   3rd Qu.:2018-10-15
##  Max.   :261190187   Max.   :2022-12-31
##
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:27312       Min.   :  1.00   Min.   :0.0000    Length:27312
##  Class :character   1st Qu.: 44.00   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 68.00   Median :0.0000    Mode  :character
##                     Mean   : 65.64   Mean   :0.3269
##                     3rd Qu.: 81.00   3rd Qu.:0.0000
##                     Max.   :123.00   Max.   :2.0000
##                                      NA's   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:27312       Mode :logical           Length:27312
##  Class :character   FALSE:22046             Class :character
##  Mode  :character   TRUE :5266              Mode  :character
##
##
##
##
##    PERP_SEX          PERP_RACE         VIC_AGE_GROUP        VIC_SEX
##  Length:27312       Length:27312       Length:27312       Length:27312
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude
##  Length:27312       Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character   1st Qu.:1000028   1st Qu.:182834   1st Qu.:40.67
##  Mode  :character   Median :1007731   Median :194487   Median :40.70
##                     Mean   :1009449   Mean   :208127   Mean   :40.74
##                     3rd Qu.:1016838   3rd Qu.:239518   3rd Qu.:40.82
##                     Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                        NA's   :10
##    Longitude        Lon_Lat            MONTH_DATE             MONTH
##  Min.   :-74.25   Length:27312       Min.   :2006-01-01   Min.   : 1.000
##  1st Qu.:-73.94   Class :character   1st Qu.:2009-07-01   1st Qu.: 5.000
##  Median :-73.92   Mode  :character   Median :2013-04-01   Median : 7.000
##  Mean   :-73.91                      Mean   :2013-12-23   Mean   : 6.825
##  3rd Qu.:-73.88                      3rd Qu.:2018-10-01   3rd Qu.: 9.000
##  Max.   :-73.70                      Max.   :2022-12-01   Max.   :12.000
##  NA's   :10
```

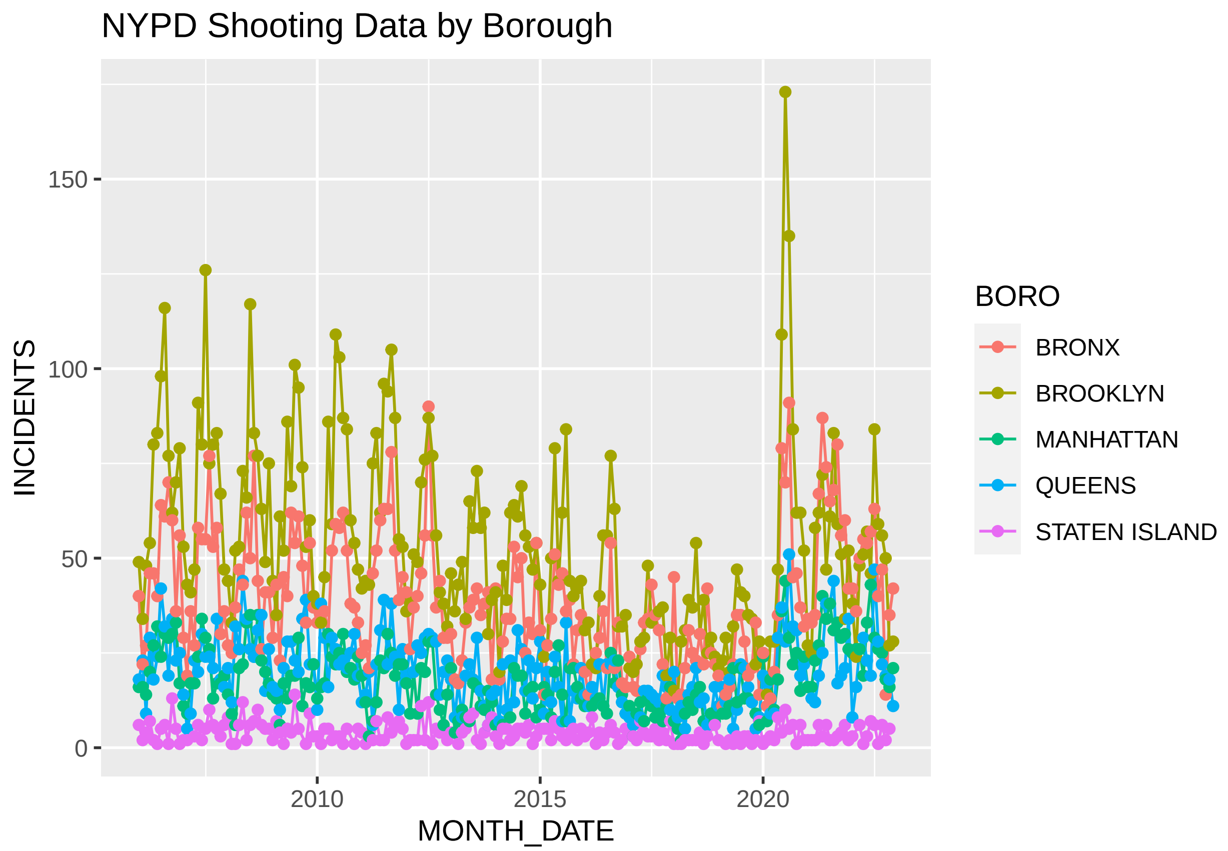
Here is some data that shows the aggregate incident count for each month in a year:

```
shooting_data_by_boro <- shooting_data_raw %>%
  summarise(.by = c(MONTH_DATE, BORO), INCIDENTS = n()) %>%
  arrange(MONTH_DATE) %>%
  mutate(NEW_INCIDENTS = INCIDENTS - lag(INCIDENTS))

shooting_data <- shooting_data_by_boro %>%
  summarise(.by = MONTH_DATE, INCIDENTS = sum(INCIDENTS))

print(shooting_data)
```

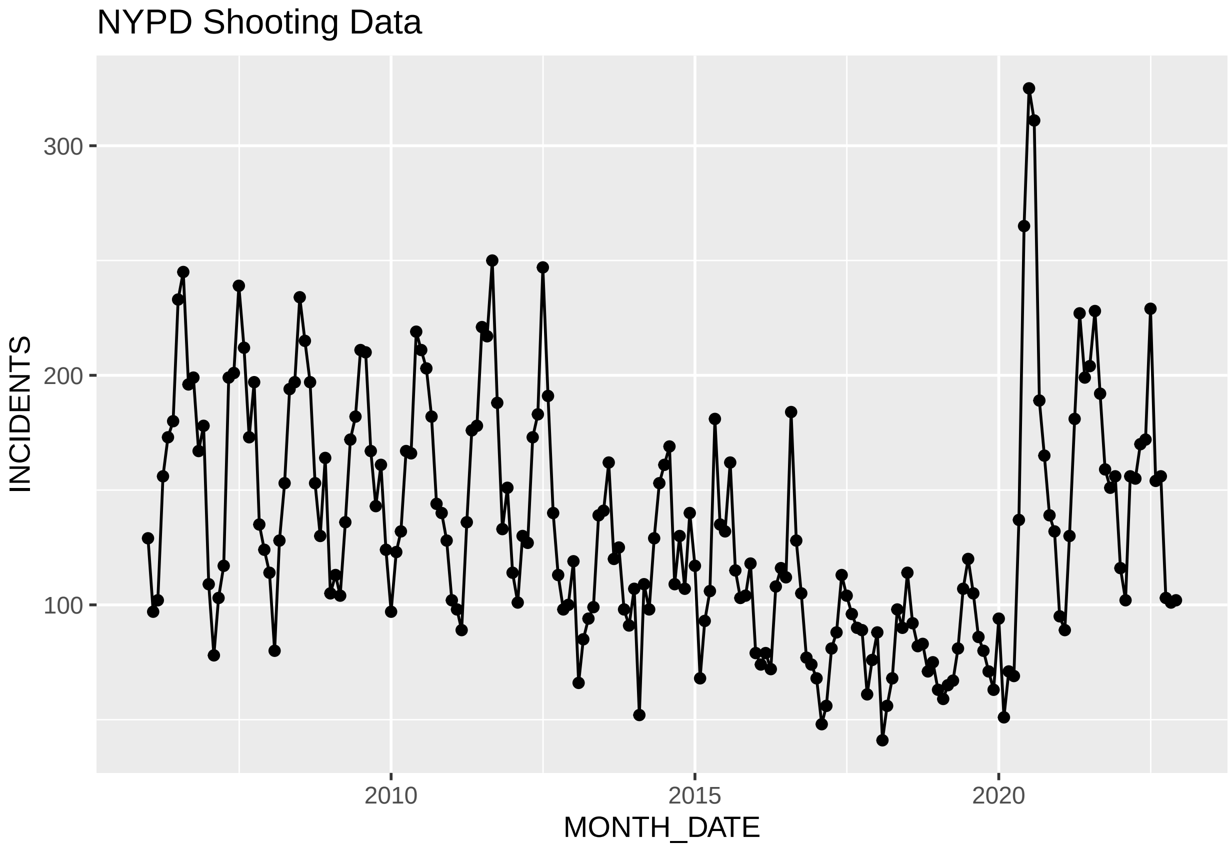
```
## # A tibble: 204 x 2
##    MONTH_DATE INCIDENTS
##    <date>         <int>
##  1 2006-01-01       129
##  2 2006-02-01        97
##  3 2006-03-01       102
##  4 2006-04-01       156
##  5 2006-05-01       173
##  6 2006-06-01       180
##  7 2006-07-01       233
##  8 2006-08-01       245
##  9 2006-09-01       196
## 10 2006-10-01       199
## # i 194 more rows
```

Explore

We plot this data per borough to see if there's anything interesting there:



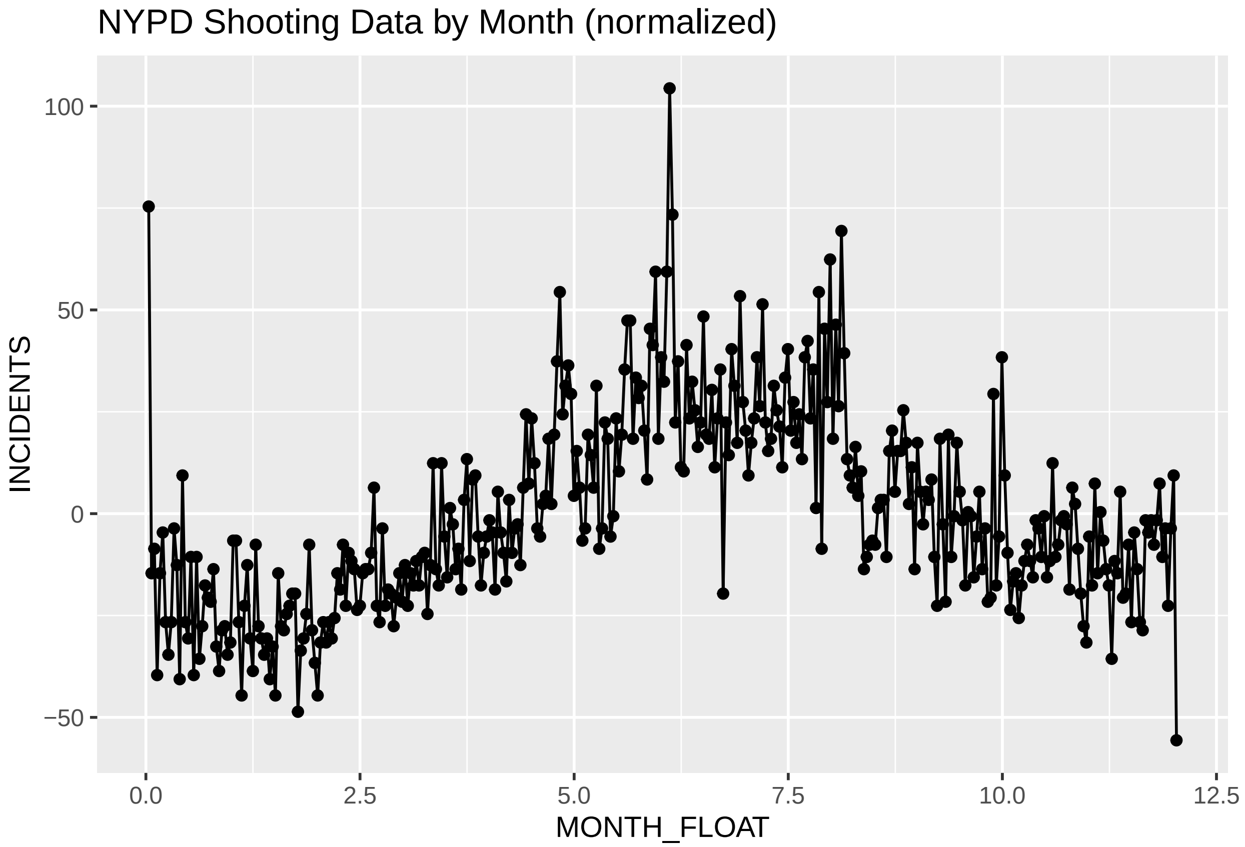
Ultimately all boroughs seem to follow a similar cyclic pattern over time. Let's look at the aggregate data for the entire city:



Analysis

We can more clearly see that there is indeed a yearly cycle. Shootings seem to happen the most in summer! Let's look at all years aggregated together (as the mean) to see the number of incidents per month. Let's also normalize this data, as at this point, we're interested in the relative difference for the months. Instead of aggregating on month of year, we'll aggregate on day of year.

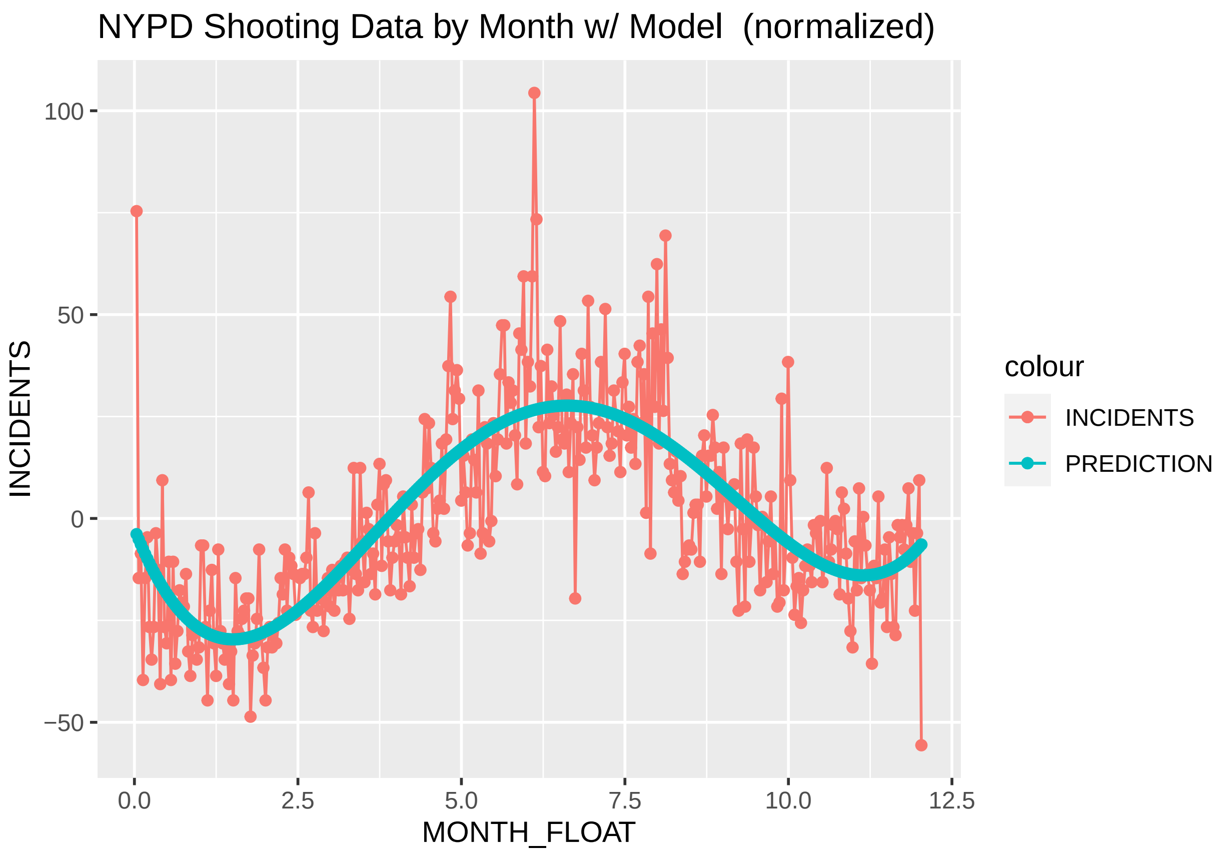
```
shooting_data_by_yday <- shooting_data_raw %>%
  mutate(MONTH = month(OCCUR_DATE), YEAR_DAY = yday(OCCUR_DATE)) %>%
  summarise(.by = c(YEAR_DAY), INCIDENTS = n()) %>%
  arrange(YEAR_DAY) %>%
  mutate(MONTH_FLOAT = YEAR_DAY/365*12) %>%
  mutate(INCIDENTS = INCIDENTS-mean(INCIDENTS))
```



Model

This definitely seems to show that shootings happen more often in the summer! Let's model the monthly data using a 4th order polynomial regression model.

```
model <- lm(data=shooting_data_by_yday,
            INCIDENTS ~ MONTH_FLOAT + I(MONTH_FLOAT^2) +
              I(MONTH_FLOAT^3) + I(MONTH_FLOAT^4))

predictions <- model %>% predict(shooting_data_by_yday)
```



The model seems good! Let's print the coefficients so we can successfully model the relative difference in shootings per month for the city of New York.

```
##
## Call:
## lm(formula = INCIDENTS ~ MONTH_FLOAT + I(MONTH_FLOAT^2) + I(MONTH_FLOAT^3) +
##     I(MONTH_FLOAT^4), data = shooting_data_by_yday)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.234  -9.264  -1.240   8.076  79.231
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.537709   4.051857  -0.626    0.532
## MONTH_FLOAT      -40.636647   4.639690  -8.758   <2e-16 ***
## I(MONTH_FLOAT^2)  18.351104   1.560714  11.758   <2e-16 ***
## I(MONTH_FLOAT^3)  -2.344897   0.194223 -12.073   <2e-16 ***
## I(MONTH_FLOAT^4)   0.091272   0.007986  11.429   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.25 on 361 degrees of freedom
## Multiple R-squared:  0.6038, Adjusted R-squared:  0.5994
## F-statistic: 137.5 on 4 and 361 DF,  p-value: < 2.2e-16
```

Conclusion

We definitely see a cyclic pattern in shootings in NYC, with them most often to occur in summer. This model should help to show relative difference in shootings from month to month. The model will effectively provide a scalar value that can be used to forecast shootings. For example, if you know the shooting count in January of this year, you can use the model to get the scale between January and whatever month you are interested in. If `n` is the number of shootings by the end of January, the forecast for February's total shootings can be obtained with `y = n * f(2)/f(1)`.

Bias may have occurred in this analysis based on the author's prior knowledge of cyclic patterns in murders. This may have unduly influenced the course of analysis in this project.

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.2 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## 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_1.1.1
##  [5] purrr_1.0.1     readr_2.1.4     tidyr_1.3.0     tibble_3.2.1
##  [9] ggplot2_3.4.1   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0 xfun_0.38        haven_2.5.2      colorspace_2.1-0
##  [5] vctrs_0.6.1      generics_0.1.3   htmltools_0.5.5  yaml_2.3.7
##  [9] utf8_1.2.3       rlang_1.1.0      pillar_1.9.0     glue_1.6.2
## [13] withr_2.5.0      DBI_1.1.3        bit64_4.0.5      dbplyr_2.3.2
## [17] modelr_0.1.11    readxl_1.4.2     lifecycle_1.0.3  munsell_0.5.0
## [21] gtable_0.3.3     cellranger_1.1.0 rvest_1.0.3      evaluate_0.20
## [25] labeling_0.4.2   knitr_1.42       tzdb_0.3.0       fastmap_1.1.1
## [29] curl_5.0.0       parallel_4.1.2   fansi_1.0.4      highr_0.10
## [33] broom_1.0.4      scales_1.2.1     backports_1.4.1  vroom_1.6.1
## [37] jsonlite_1.8.4   farver_2.1.1     bit_4.0.5        fs_1.6.1
## [41] hms_1.1.3        digest_0.6.31    stringi_1.7.12   grid_4.1.2
## [45] cli_3.6.1        tools_4.1.2      magrittr_2.0.3   crayon_1.5.2
## [49] pkgconfig_2.0.3  xml2_1.3.3       reprex_2.0.2     timechange_0.2.0
## [53] rmarkdown_2.21   httr_1.4.5       rstudioapi_0.14  R6_2.5.1
## [57] compiler_4.1.2
```