Lina's Analysis

Lina Cook

4/21/2022

```
# load R libraries
library(janitor)
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.0.5
```

```
library(tidyverse)
library(tidyr)
library(kableExtra)
library(reticulate)
```

```
## Warning: package 'reticulate' was built under R version 4.0.5
```

```
library(data.table)
library(XML)
```

```
## Warning: package 'XML' was built under R version 4.0.5
```

```
library(RSocrata)
library(nominatimlite)
library(spatialrisk)
library(maps)
library(ggmap)
library(mapdata)

# load Python libraries
#reticulate::py_install("pandas")
#reticulate::py_install("usaddress")
#reticulate::py_install("usaddress-scourgify", pip=TRUE)
```

```
# address objects
usaddress <- reticulate::import('usaddress')
scourgify <- reticulate::import('scourgify')

# FCN: normalize addresses

# function to usaddress-scourgify that normalizes addresses and returns
# a datatable of normalized addresses
normalize_address <- function(address_dat) {

  ## throw error if scourgify module is not imported
  if (! reticulate::py_module_available('scourgify')) {
    stop('The scourgify python module is not available in the default Python installation\n',
         'Install the usddress-scourgify module in R by calling:\n    reticulate::py_install("usaddress-
  }
```

```
  ## create object using purrr to specify how errors will be treated
  ##  in this case, errors will print "ERROR" across all address fields.
  poss_norm_addr <- purrr::possibly(.f = scourgify$normalize_address_record, otherwise = "ERROR")

  ## pass addresses to scourgify, which normalizes addresses
  ## according to US Post Office conventions.
  norm_addr <- furrr::future_map(address_dat, poss_norm_addr)

  ## bind list to dataframe for easier handling
  z_address_df <- as.data.frame(do.call(rbind, norm_addr))

  ## concatenate and edit output
  z_address_df <- z_address_df %>%
    ## trim postal codes
    mutate(postal_code = strtrim(z_address_df$postal_code, 5)) %>%
    ## concatenate
    unite(col = z_addr_cat, 1:5, sep = ", ", remove = FALSE, na.rm = TRUE) %>%
    ## move the concatenated, normalized address field to the first column
    dplyr::relocate(z_addr_cat, .before = address_line_1)

  ## remove NULL values
    ## from concatenated address field
    z_address_df$z_addr_cat <- gsub("NULL, ", "", z_address_df$z_addr_cat)
    ## from all other areas in dataframe
    z_address_df <- map_df(z_address_df, ~ gsub("NULL", "", .x))

  ## add raw address back on to results
  z_address_df <- cbind(address_dat, z_address_df) %>%
    rename(raw_addr = address_dat)

}
```

```
## Read in data
dat_311 <- read.csv("./data/dat_311_combined.csv")
## Make year variable, make created_date into date variable
dat_311 <- dat_311 %>%
  mutate(year = as.numeric(substr(created_date, start = 1, stop = 4)),
         month = as.numeric(substr(created_date, start = 6, stop = 7)),
         created_date = as.Date(created_date, format = "%Y-%m-%d"),
         latlong = paste0(round(latitude, 4), ", ",
                          round(longitude, 4)),
         after_2020 = case_when(
           year < 2020 ~ 0,
           year >= 2020 ~ 1))

## Read in Open Restaurant Applications
dat_or <- read.csv("./data/Open_Restaurant_Applications.csv")

## Make tibble of unique lat-longs from open restaurants
latlong_or <- tibble(
  latlong = paste0(round(dat_or$Latitude, 4), ", ",
                   round(dat_or$Longitude, 4))) %>%
  filter(latlong != "NA, NA") %>%
```

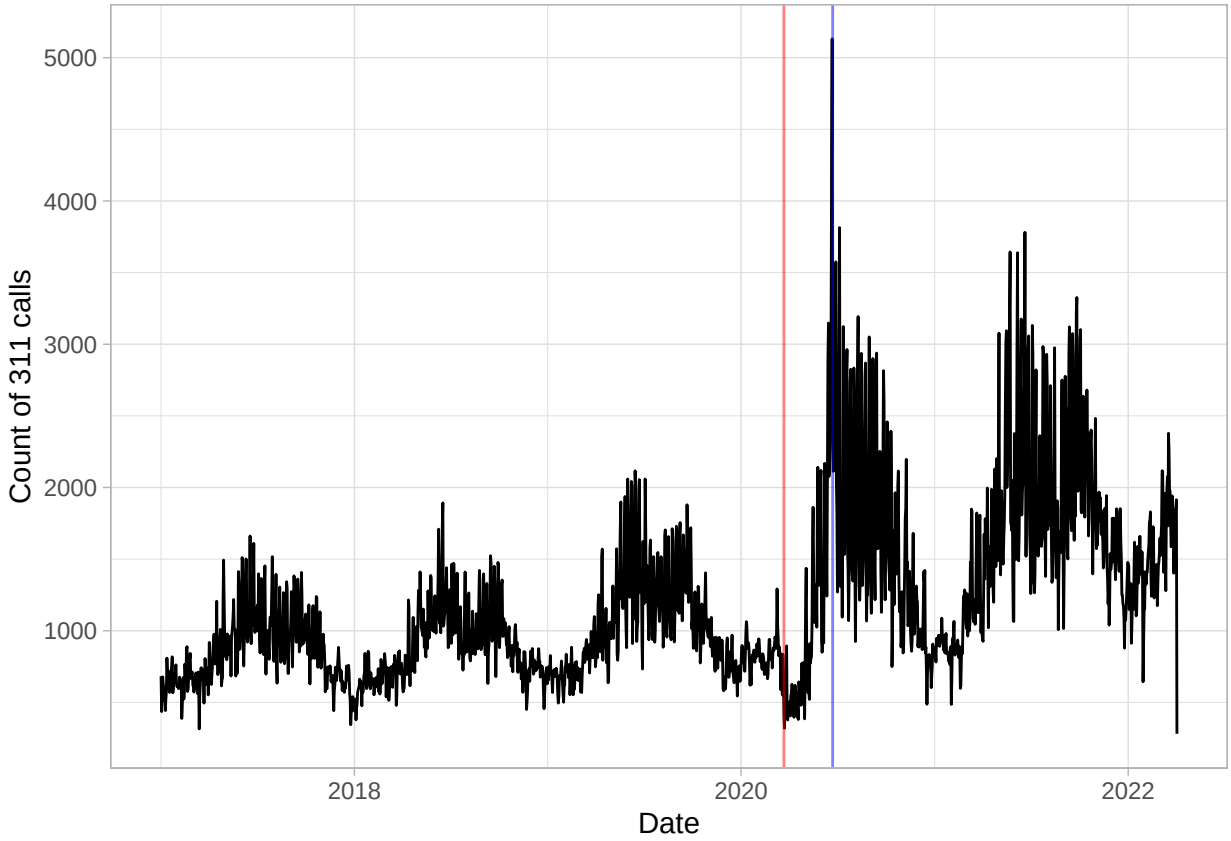
```
  unique()

## Evaluate if each 311 lat-long is in the unique O.R. lat-long's
dat_311$shed <- dat_311$latlong %in% latlong_or$latlong

## Create datasets of before or after 2020
dat_311_pre <- filter(dat_311, year < 2020)
dat_311_post <- filter(dat_311, year >= 2020)

## Create datasets of illegal parking from 311
dat_311_parking <- filter(dat_311, complaint_type == "Illegal Parking")
dat_311_no_park <- filter(dat_311, complaint_type != "Illegal Parking")

## Create datasets of sheds
dat_311_shed <- dat_311 %>%
  filter(shed == TRUE)

dat_311_no_shed <- dat_311 %>%
  filter(shed == FALSE)
```

```
## Summary Table
# dat_311_pre %>%
#   group_by(complaint_type) %>%
#   summarise(
#     count = n()
#   )
#
# dat_311_post %>%
#   group_by(complaint_type) %>%
#   summarise(
#     count = n()
#   )

table(dat_311$complaint_type, dat_311$after_2020)
```

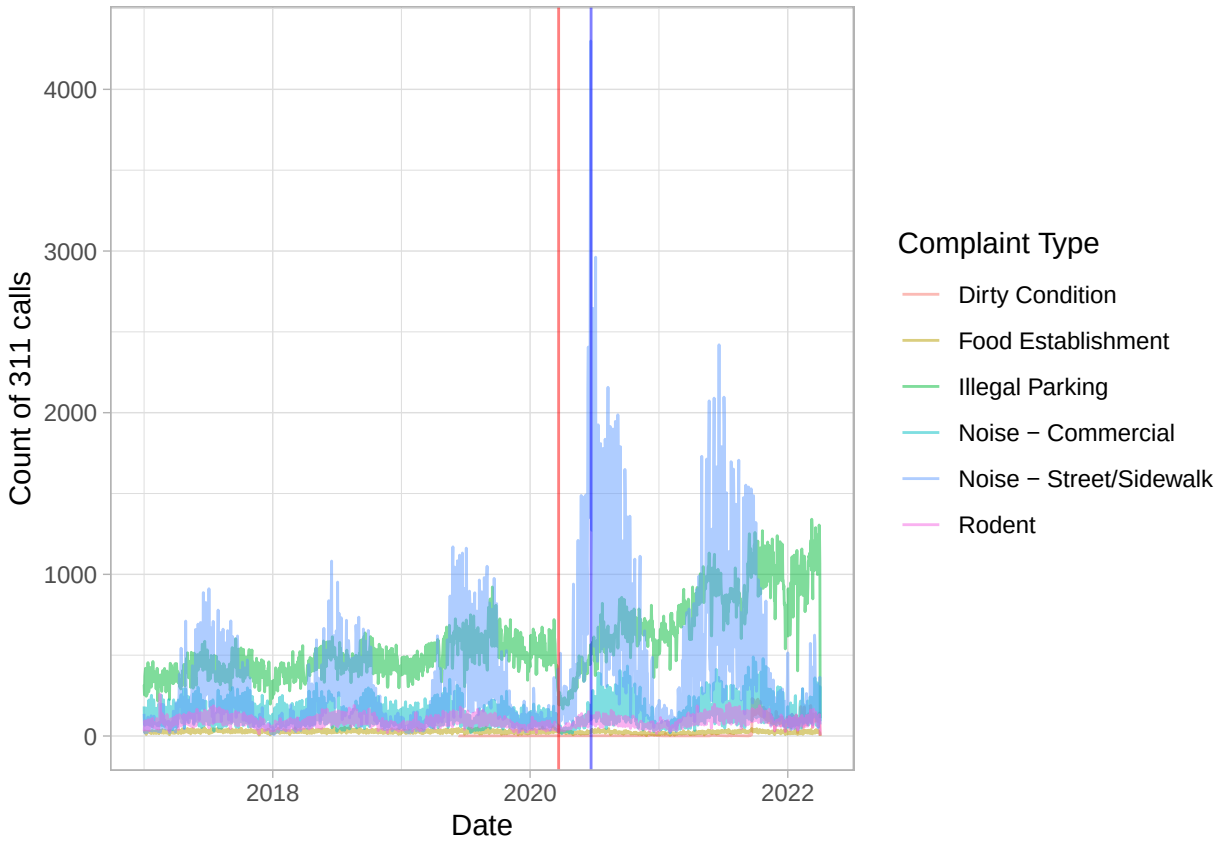
```
##
##                               0      1
##   Dirty Condition             3  21971
##   Food Establishment      31734  19303
##   Illegal Parking        507982 605095
##   Noise - Commercial     132673 110452
##   Noise - Street/Sidewalk 245199 405919
##   Rodent                  99250  74940
```

```
## Plot of overall number of 311 calls by day
count <- dat_311 %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_2020 = case_when(
           year < 2020 ~ 0,
           year >= 2020 ~ 1
         ))

ggplot(data = count, aes(x = created_date, y = n)) +
  geom_line() +
```

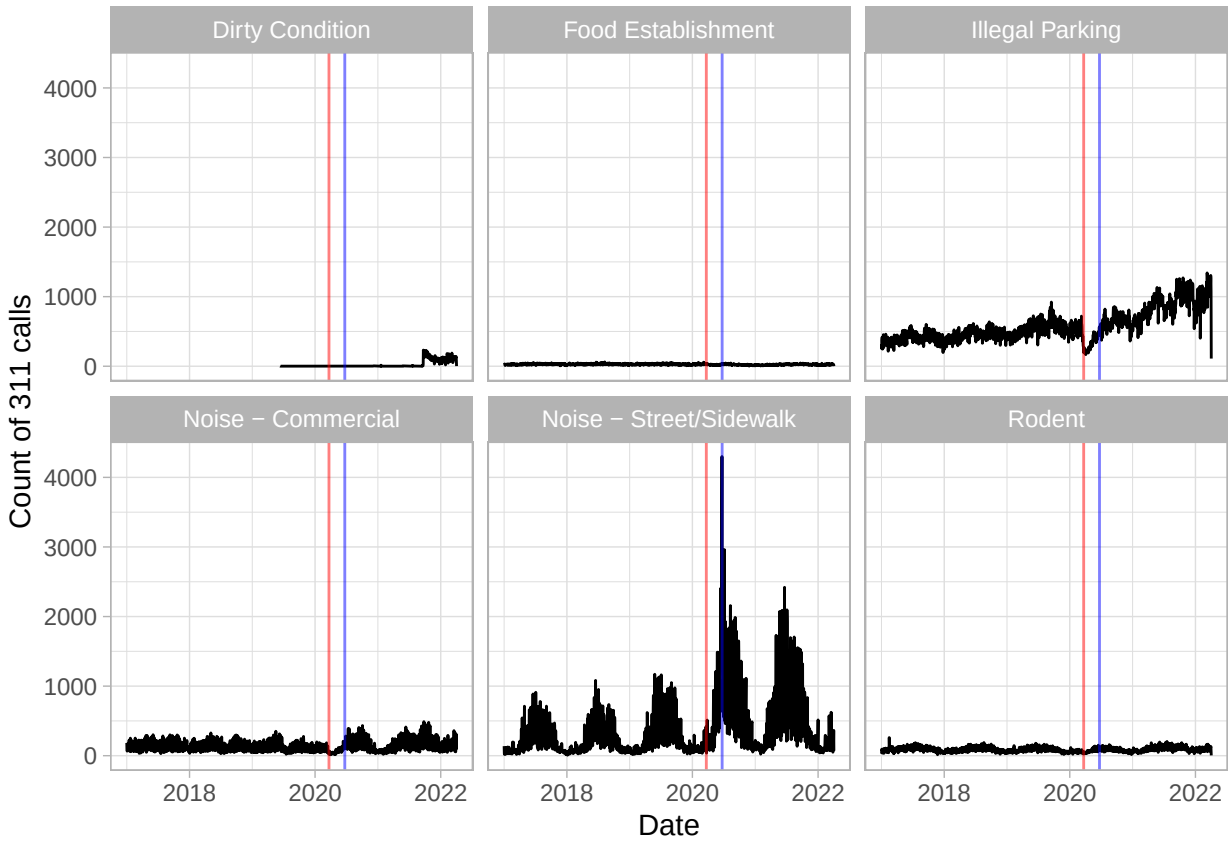
```
  geom_vline(xintercept = as.Date("03-22-2020", format = "%m-%d-%Y"),
             color = "red", alpha = 0.5) +
    geom_vline(xintercept = as.Date("06-22-2020", format = "%m-%d-%Y"),
             color = "blue", alpha = 0.5) +
  theme_light() +
  labs(x = "Date", y = "Count of 311 calls")
```




```
## Plot of overall 311 calls per day by complaint type
count2 <- dat_311 %>%
  group_by(complaint_type) %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_2020 = case_when(
           year < 2020 ~ 0,
           year >= 2020 ~ 1
         ))

ggplot(data = count2, aes(x = created_date, y = n, color = complaint_type)) +
  geom_line(alpha = 0.5) +
  geom_vline(xintercept = as.Date("03-22-2020", format = "%m-%d-%Y"),
             color = "red", alpha = 0.5) +
  geom_vline(xintercept = as.Date("06-22-2020", format = "%m-%d-%Y"),
             color = "blue", alpha = 0.5) +
  theme_light() +
  labs(x = "Date", y = "Count of 311 calls",
       color = "Complaint Type")
```




```
## Plot of overall 311 calls per day by complaint type
ggplot(count2, aes(created_date, n)) +
  geom_line() +
  geom_vline(xintercept = as.Date("03-22-2020", format = "%m-%d-%Y"),
             color = "red", alpha = 0.5) +
  geom_vline(xintercept = as.Date("06-22-2020", format = "%m-%d-%Y"),
             color = "blue", alpha = 0.5) +
  theme_light() +
  facet_wrap(~ complaint_type) +
  labs(x = "Date", y = "Count of 311 calls")
```




```
table(dat_311_pre$shed)
```

```
##
##  FALSE   TRUE
## 933929  82912
```

```
table(dat_311_pre$shed)/nrow(dat_311_pre)*100
```

```
##
##    FALSE     TRUE
## 91.84612  8.15388
```

```
table(dat_311_post$shed)
```

```
##
##   FALSE    TRUE
## 1150365   87315
```

```
table(dat_311_post$shed)/nrow(dat_311_post)*100
```

```
##
##     FALSE      TRUE
## 92.945269  7.054731
```

```
round(table(dat_311_post$complaint_type,
            dat_311_post$shed), 3)
```

```
##
##                           FALSE   TRUE
##   Dirty Condition         20660   1311
```

```
##   Food Establishment       13309   5994
##   Illegal Parking         588415  16680
##   Noise - Commercial       69082  41370
##   Noise - Street/Sidewalk 387047  18872
##   Rodent                   71852   3088
```

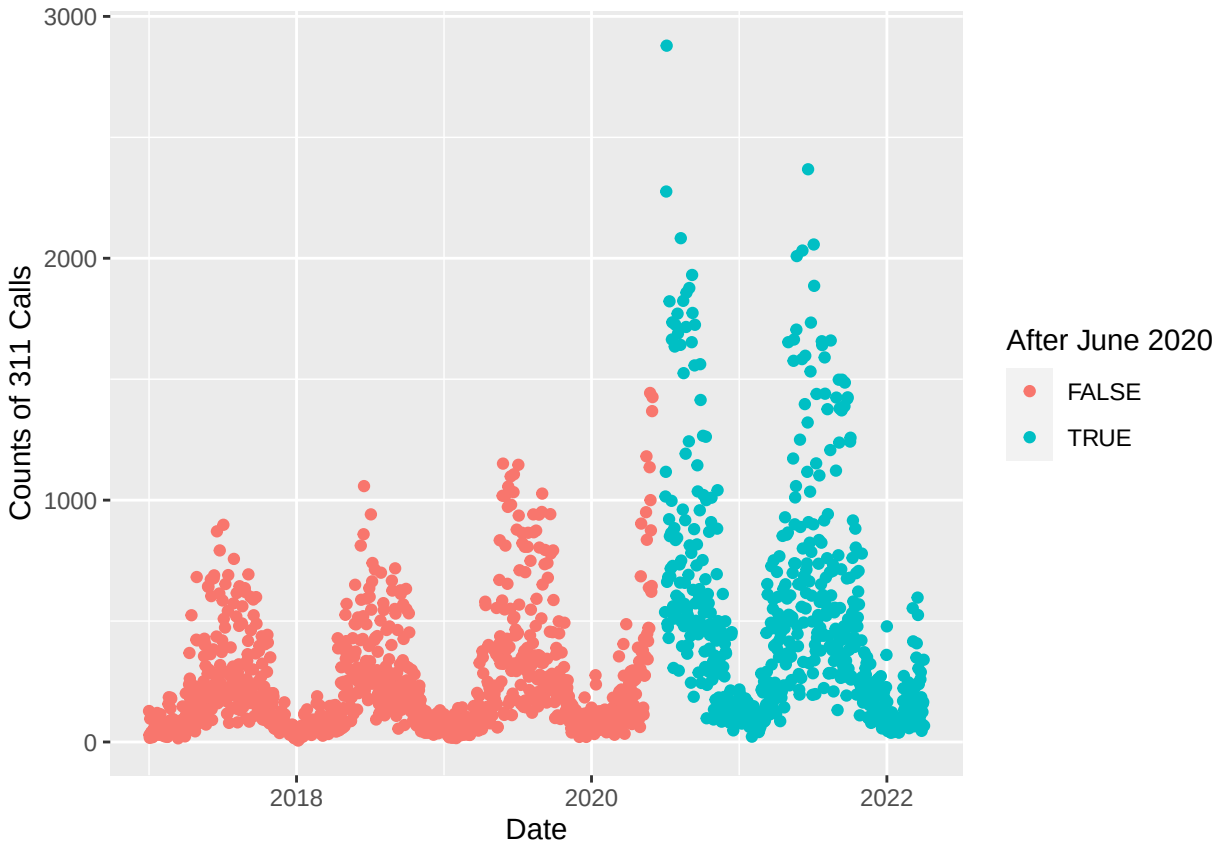
```
18872/(387047+18872)
```

```
## [1] 0.04649203
```

```
round(table(dat_311_pre$complaint_type,
            dat_311_pre$shed)/nrow(dat_311_pre)*100, 3)
```

```
##
##                            FALSE   TRUE
##   Dirty Condition          0.000  0.000
##   Food Establishment       2.310  0.810
##   Illegal Parking         48.499  1.458
##   Noise - Commercial       8.411  4.636
##   Noise - Street/Sidewalk 23.130  0.983
##   Rodent                   9.495  0.266
```

```
test <- dat_311_no_shed %>%
  filter(complaint_type == "Noise - Street/Sidewalk",
         created_date < as.Date("06-01-2020", format = "%m-%d-%Y") |
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y")) %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_june2020 = case_when(
           created_date < as.Date("06-01-2020", format = "%m-%d-%Y") ~ FALSE,
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y") ~ TRUE
         ))

ggplot(data = test,
       aes(x = created_date, y = n, color = as.factor(after_june2020))) +
  geom_point() +
  #geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2), color = "black") +
  #geom_segment(aes(x = x3, y = y3, xend = x4, yend = y4), color = "black") +
  labs(x = "Date", y = "Counts of 311 Calls", color = "After June 2020")#,
```




```
       #title = "RD of Street Noise Complaints")
```

```
## RD 1: Full data (no parking)
# counts ~ date + after_2020
counts_shed <- dat_311_no_park %>%
  filter(shed == TRUE,
         created_date < as.Date("06-01-2020", format = "%m-%d-%Y") |
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y")) %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_june2020 = case_when(
           created_date < as.Date("06-01-2020", format = "%m-%d-%Y") ~ FALSE,
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y") ~ TRUE
         ))

lm1 <- lm(n ~ created_date * after_june2020, data = counts_shed)
summary(lm1)
```

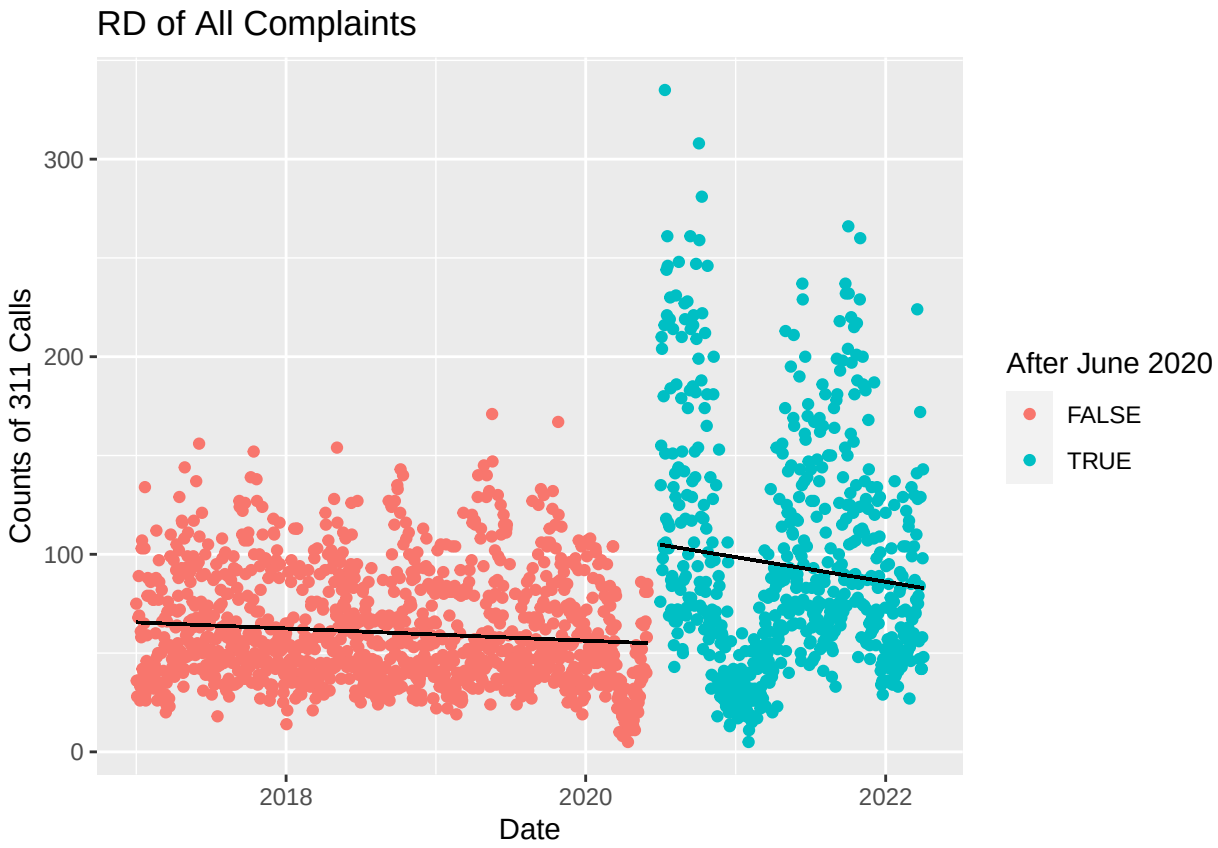
```
##
## Call:
## lm(formula = n ~ created_date * after_june2020, data = counts_shed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -92.241 -25.232  -8.932  22.073 230.815
##
## Coefficients:
```

```
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     210.686615  57.190997   3.684 0.000236 ***
## created_date                     -0.008455   0.003214  -2.630 0.008597 **
## after_june2020TRUE              521.705625 173.001136   3.016 0.002599 **
## created_date:after_june2020TRUE  -0.025585   0.009276  -2.758 0.005865 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.86 on 1885 degrees of freedom
## Multiple R-squared:  0.1392, Adjusted R-squared:  0.1379
## F-statistic: 101.6 on 3 and 1885 DF,  p-value: < 2.2e-16
```

```
lm1$coefficients[1] + (as.Date("2017-01-01") - as.Date("1970-01-01"))*lm1$coefficients[2]
```

```
## Time difference of 65.54707 days
```

```
## plot lm1
hold <- as.Date("2017-01-01") - as.Date("1970-01-01")
n_prior <- length(which(counts_shed$after_june2020 == FALSE))
n_post <- length(which(counts_shed$after_june2020 == TRUE))

x1 <- as.Date("01-01-2017", format = "%m-%d-%Y")
x2 <- as.Date("05-31-2020", format = "%m-%d-%Y")
y1 <- lm1$coefficients[1] + hold*lm1$coefficients[2]
y2 <- y1 + n_prior*lm1$coefficients[2]

x3 = as.Date("07-01-2020", format = "%m-%d-%Y")
x4 = as.Date("04-03-2022", format = "%m-%d-%Y")
y3 = y2 + (hold+n_prior+30)*lm1$coefficients[4] + lm1$coefficients[3]
y4 = y3 + n_post*(lm1$coefficients[2] + lm1$coefficients[4])


ggplot(data = counts_shed, aes(x = created_date, y = n, color = as.factor(after_june2020))) +
  geom_point() +
  geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2), color = "black") +
  geom_segment(aes(x = x3, y = y3, xend = x4, yend = y4), color = "black") +
  labs(x = "Date", y = "Counts of 311 Calls", color = "After June 2020",
       title = "RD of All Complaints")
```

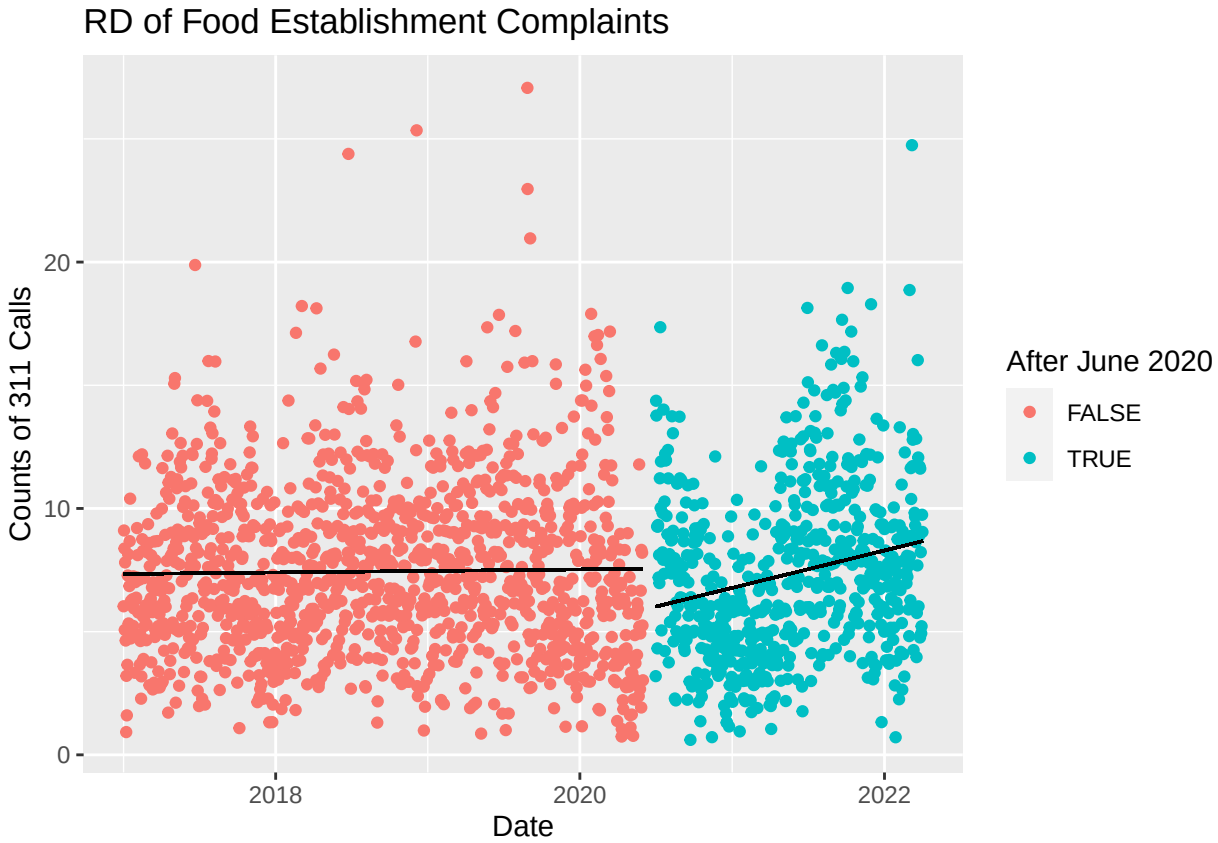



```
## RD 2: Food Establishment

counts_shed_food <- dat_311_shed %>%
  filter(complaint_type == "Food Establishment",
         created_date < as.Date("06-01-2020", format = "%m-%d-%Y") |
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y")) %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_june2020 = case_when(
           created_date < as.Date("06-01-2020", format = "%m-%d-%Y") ~ FALSE,
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y") ~ TRUE
         ))

lm2 <- lm(n ~ created_date * after_june2020, data = counts_shed_food)
summary(lm2)
```

```
##
## Call:
## lm(formula = n ~ created_date * after_june2020, data = counts_shed_food)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4275 -2.4408 -0.4234  1.8816 19.4959
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.445e+00  4.824e+00   0.921    0.357
```

```
## created_date                       1.687e-04  2.711e-04   0.622    0.534
## after_june2020TRUE                -7.525e+01  1.463e+01  -5.145 2.95e-07 ***
## created_date:after_june2020TRUE    3.997e-03  7.842e-04   5.098 3.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.443 on 1880 degrees of freedom
## Multiple R-squared:  0.01709,    Adjusted R-squared:  0.01552
## F-statistic:  10.9 on 3 and 1880 DF,  p-value: 4.28e-07
```

```r
## plot lm2
n_prior <- length(which(counts_shed_food$after_june2020 == FALSE))
n_post <- length(which(counts_shed_food$after_june2020 == TRUE))

y1 <- lm2$coefficients[1] + hold*lm2$coefficients[2]
y2 <- y1 + n_prior*lm2$coefficients[2]

y3 = y2 + (hold+n_prior+30)*lm2$coefficients[4] + lm2$coefficients[3]
y4 = y3 + n_post*(lm2$coefficients[2] + lm2$coefficients[4])


ggplot(data = counts_shed_food,
       aes(x = created_date, y = n, color = as.factor(after_june2020))) +
  geom_jitter() +
  geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2), color = "black") +
  geom_segment(aes(x = x3, y = y3, xend = x4, yend = y4), color = "black") +
  labs(x = "Date", y = "Counts of 311 Calls", color = "After June 2020",
       title = "RD of Food Establishment Complaints")
```

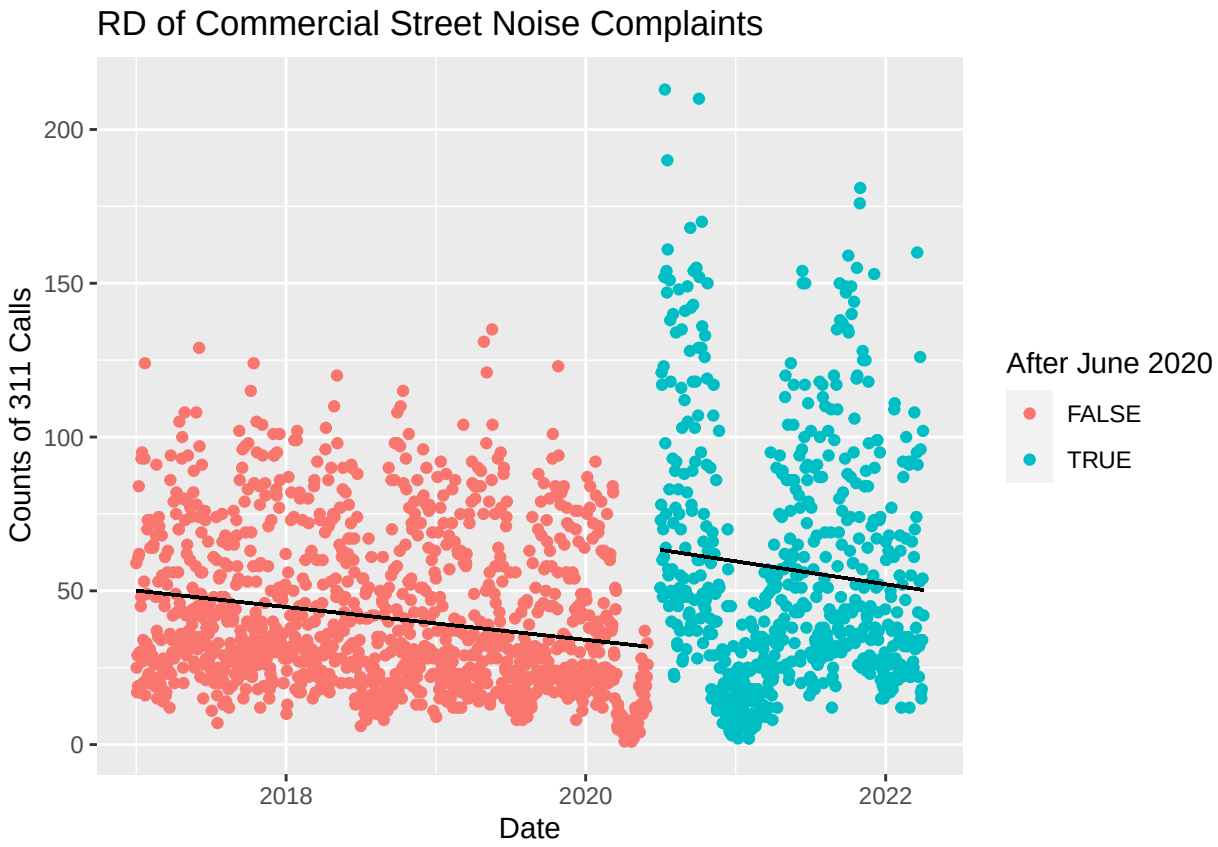


```r
## RD 3: Noise - Commercial

counts_shed_comm <- dat_311_shed %>%
  filter(complaint_type == "Noise - Commercial",
         created_date < as.Date("06-01-2020", format = "%m-%d-%Y") |
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y")) %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_june2020 = case_when(
           created_date < as.Date("06-01-2020", format = "%m-%d-%Y") ~ FALSE,
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y") ~ TRUE
         ))

lm3 <- lm(n ~ created_date * after_june2020, data = counts_shed_comm)
summary(lm3)
```

```
##
## Call:
## lm(formula = n ~ created_date * after_june2020, data = counts_shed_comm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.975 -21.891  -9.346  17.224 150.408
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    301.318166  43.660846   6.901 7.01e-12 ***
## created_date                    -0.014636   0.002454  -5.964 2.93e-09 ***
## after_june2020TRUE             136.346187 131.733523   1.035    0.301
## created_date:after_june2020TRUE  -0.005688   0.007063  -0.805    0.421
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.1 on 1882 degrees of freedom
## Multiple R-squared:  0.0729, Adjusted R-squared:  0.07143
## F-statistic: 49.33 on 3 and 1882 DF,  p-value: < 2.2e-16
```

```r
## plot lm3
n_prior <- length(which(counts_shed_comm$after_june2020 == FALSE))
n_post <- length(which(counts_shed_comm$after_june2020 == TRUE))

y1 <- lm3$coefficients[1] + hold*lm3$coefficients[2]
y2 <- y1 + n_prior*lm3$coefficients[2]

y3 = y2 + (hold+n_prior+30)*lm3$coefficients[4] + lm3$coefficients[3]
y4 = y3 + n_post*(lm3$coefficients[2] + lm3$coefficients[4])


ggplot(data = counts_shed_comm,
       aes(x = created_date, y = n, color = as.factor(after_june2020))) +
  geom_point() +
  geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2), color = "black") +
  geom_segment(aes(x = x3, y = y3, xend = x4, yend = y4), color = "black") +
  labs(x = "Date", y = "Counts of 311 Calls", color = "After June 2020",
```

```
       title = "RD of Commercial Street Noise Complaints")
```




```
## RD 4: Noise - Street/Sidewalk

counts_shed_street <- dat_311_shed %>%
  filter(complaint_type == "Noise - Street/Sidewalk",
         created_date < as.Date("06-01-2020", format = "%m-%d-%Y") |
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y")) %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_june2020 = case_when(
           created_date < as.Date("06-01-2020", format = "%m-%d-%Y") ~ FALSE,
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y") ~ TRUE
         ))

lm4 <- lm(n ~ created_date * after_june2020, data = counts_shed_street)
summary(lm4)
```

```
##
## Call:
## lm(formula = n ~ created_date * after_june2020, data = counts_shed_street)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.982  -5.885  -2.087   3.856  71.395
##
## Coefficients:
```

```
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -8.741e+01  1.714e+01  -5.099 3.76e-07 ***
## created_date                     5.457e-03  9.633e-04   5.665 1.70e-08 ***
## after_june2020TRUE               7.268e+02  5.166e+01  14.068  < 2e-16 ***
## created_date:after_june2020TRUE -3.825e-02  2.770e-03 -13.808  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.16 on 1860 degrees of freedom
## Multiple R-squared:  0.2951, Adjusted R-squared:  0.294
## F-statistic: 259.6 on 3 and 1860 DF,  p-value: < 2.2e-16
```

```r
## plot lm4
n_prior <- length(which(counts_shed_street$after_june2020 == FALSE))
n_post <- length(which(counts_shed_street$after_june2020 == TRUE))

y1 <- lm4$coefficients[1] + hold*lm4$coefficients[2]
y2 <- y1 + n_prior*lm4$coefficients[2]

y3 = y2 + (hold+n_prior+30)*lm4$coefficients[4] + lm4$coefficients[3]
y4 = y3 + n_post*(lm4$coefficients[2] + lm4$coefficients[4])


ggplot(data = counts_shed_street,
       aes(x = created_date, y = n, color = as.factor(after_june2020))) +
  geom_point() +
  geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2), color = "black") +
  geom_segment(aes(x = x3, y = y3, xend = x4, yend = y4), color = "black") +
  labs(x = "Date", y = "Counts of 311 Calls", color = "After June 2020",
       title = "RD of Street Noise Complaints")
```


```
## RD 5: Rodent

counts_shed_rodent <- dat_311_shed %>%
  filter(complaint_type == "Rodent",
         created_date < as.Date("06-01-2020", format = "%m-%d-%Y") |
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y")) %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_june2020 = case_when(
           created_date < as.Date("06-01-2020", format = "%m-%d-%Y") ~ FALSE,
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y") ~ TRUE
         ))

lm5 <- lm(n ~ created_date * after_june2020, data = counts_shed_rodent)
summary(lm5)
```

```
##
## Call:
## lm(formula = n ~ created_date * after_june2020, data = counts_shed_rodent)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2002 -1.5862 -0.4863  1.1191 13.5991
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -5.196e+00  2.937e+00  -1.769  0.07705 .
```

```
## created_date                       4.490e-04  1.649e-04   2.723  0.00654 **
## after_june2020TRUE                -4.674e+01  8.603e+00  -5.433 6.35e-08 ***
## created_date:after_june2020TRUE    2.545e-03  4.613e-04   5.516 4.00e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.981 on 1710 degrees of freedom
## Multiple R-squared:  0.135,  Adjusted R-squared:  0.1334
## F-statistic: 88.93 on 3 and 1710 DF,  p-value: < 2.2e-16
```

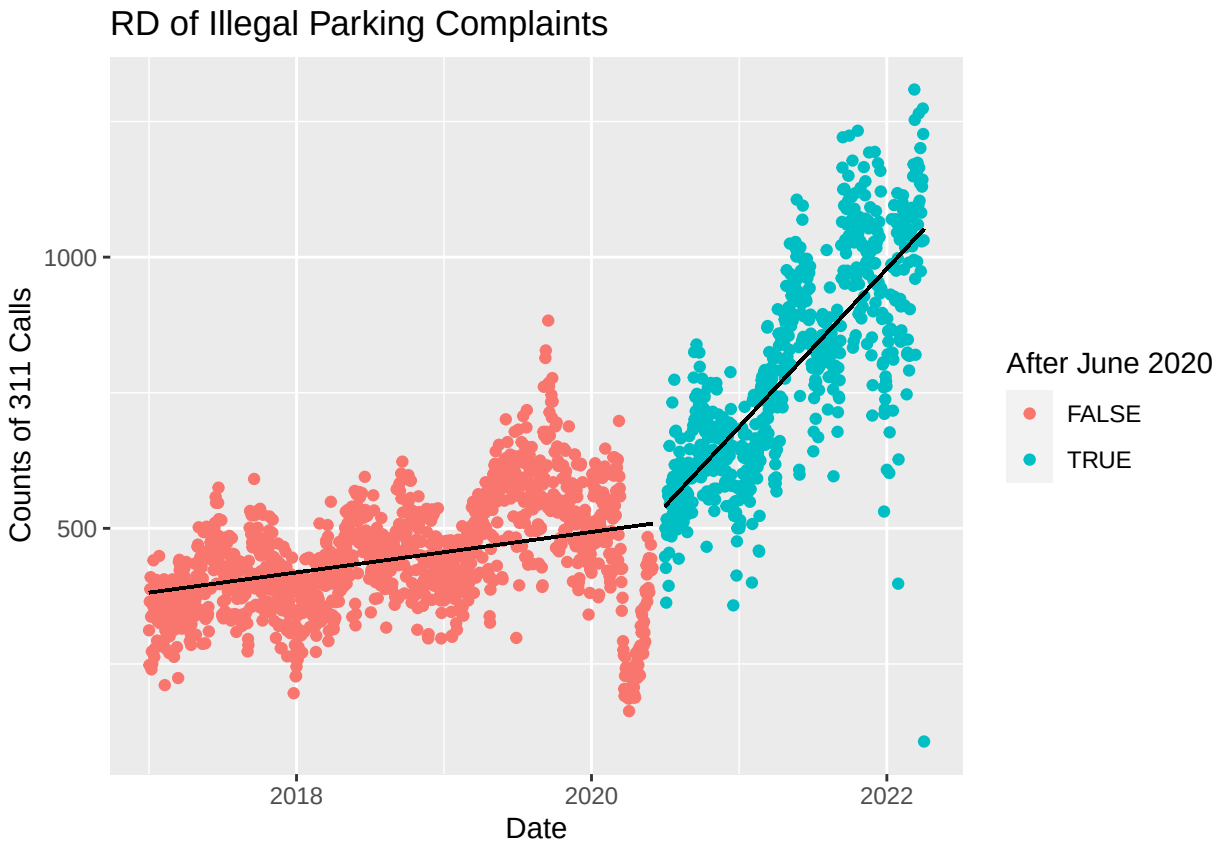
```r
## plot lm5
n_prior <- length(which(counts_shed_rodent$after_june2020 == FALSE))
n_post <- length(which(counts_shed_rodent$after_june2020 == TRUE))

y1 <- lm5$coefficients[1] + hold*lm5$coefficients[2]
y2 <- y1 + n_prior*lm5$coefficients[2]

y3 = y2 + (hold+n_prior+30)*lm5$coefficients[4] + lm5$coefficients[3]
y4 = y3 + n_post*(lm5$coefficients[2] + lm5$coefficients[4])


ggplot(data = counts_shed_rodent,
       aes(x = created_date, y = n, color = as.factor(after_june2020))) +
  geom_jitter() +
  geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2), color = "black") +
  geom_segment(aes(x = x3, y = y3, xend = x4, yend = y4), color = "black") +
  labs(x = "Date", y = "Counts of 311 Calls", color = "After June 2020",
       title = "RD of Rodent Complaints")
```

```r
## RD 6: Only Parking, shed = FALSE

counts_shed_parking <- dat_311_parking %>%
  filter(shed == FALSE,
         created_date < as.Date("06-01-2020", format = "%m-%d-%Y") |
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y")) %>%
  count(created_date) %>%
  mutate(year = as.numeric(substr(created_date, 1, 4)),
         after_june2020 = case_when(
           created_date < as.Date("06-01-2020", format = "%m-%d-%Y") ~ FALSE,
           created_date > as.Date("06-30-2020", format = "%m-%d-%Y") ~ TRUE
         ))

lm6 <- lm(n ~ created_date * after_june2020, data = counts_shed_parking)
summary(lm6)
```

```
##
## Call:
## lm(formula = n ~ created_date * after_june2020, data = counts_shed_parking)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -947.11  -60.36    1.02   68.73  400.71
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -1.371e+03  1.533e+02  -8.944   <2e-16 ***
## created_date                     1.021e-01  8.615e-03  11.849   <2e-16 ***
## after_june2020TRUE              -1.277e+04  4.637e+02 -27.541   <2e-16 ***
## created_date:after_june2020TRUE  6.941e-01  2.486e-02  27.921   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 109.5 on 1885 degrees of freedom
## Multiple R-squared:  0.7525, Adjusted R-squared:  0.7521
## F-statistic:  1911 on 3 and 1885 DF,  p-value: < 2.2e-16
```

```r
## plot lm6
n_prior <- length(which(counts_shed_parking$after_june2020 == FALSE))
n_post <- length(which(counts_shed_parking$after_june2020 == TRUE))

y1 <- lm6$coefficients[1] + hold*lm6$coefficients[2]
y2 <- y1 + n_prior*lm6$coefficients[2]

y3 = y2 + (hold+n_prior+30)*lm6$coefficients[4] + lm6$coefficients[3]
y4 = y3 + n_post*(lm6$coefficients[2] + lm6$coefficients[4])


ggplot(data = counts_shed_parking,
       aes(x = created_date, y = n, color = as.factor(after_june2020))) +
  geom_point() +
  geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2), color = "black") +
  geom_segment(aes(x = x3, y = y3, xend = x4, yend = y4), color = "black") +
  labs(x = "Date", y = "Counts of 311 Calls", color = "After June 2020",
```

```
       title = "RD of Illegal Parking Complaints")
```




```
## Make map of NYC using lat-long: not great
# x = rows = latitude, from = 40.49912, to = 40.91346
# y = columns = longitude, from = -74.25453, to = -73.7006

states <- map_data("state")
NY <- subset(states, region %in% c("new york"))
NYC <- filter(NY, long <= -73.7006 & long >= -74.25453 &
                lat <= 40.91346 & lat >= 40.49912)
counties <- map_data("county")
NY_county <- subset(counties, region == "new york")
remove(states, counties)


## Map plot, not looking the best if I'm honest
ggplot(data = NYC, mapping = aes(x = long, y = lat, group = group)) +
  coord_fixed(1.3) +
  geom_polygon(color = "black", fill = NA) +
  theme_void() +
  geom_polygon(data = NY_county, #aes(fill = sqrt(residence_count)),
               color = "black")
```