Gun Series

Ryan Larson

11/24/2020

Base Panel Construction - ZCTA-Week Level

Hospital Data - ZCTA-Week level

```
hosp_zcta <- read_csv("minnepop_1620_agg_zipfull.csv") %>%
  rename(zipcode = Zipcode) %>%
  arrange(zipcode, year, weekofyr) %>%
  select(-`_chk`)
```

ZCTAs and ACS 5-Year Estimates

```
#adding in 5-year ACS data
census_api_key("ecda17575f4d914b502c70f2bae7a5f3d253792d")

year <- lst(2016, 2017, 2018, 2019)

acs <- map_dfr(
  year,
  ~ get_acs(geography = "zcta",
               variables = c("B01001_001E", "B03003_003E",
                             "B02001_003E", "B02001_002E",
                             "B02001_004E", "B02001_008E",
                             "B02001_005E", "B02001_006E",
                             "B02001_007E", "B11001_003E",
                             "B17001_002E", "B01002_001E",
                             "B09010_002E", "B06009_005E",
                             "B01001_002E", "B99233_005E"),
               output = "wide",
               survey = "acs5",
               year = .x),  .id = "year") %>%
  rename(total_pop = B01001_001E,
         white_pop = B02001_002E,
         black_pop = B02001_003E,
         na_pop = B02001_004E,
         asian_pop = B02001_005E,
         hpi_pop = B02001_006E,
         other_pop = B02001_007E,
```

```r
         biracial_pop = B02001_008E,
         hisp_pop = B03003_003E,
         ssi_snap = B09010_002E, #snap, ssi, public cash transfers
         med_age = B01002_001E,
         mar_fam = B11001_003E,
         povlevel = B17001_002E,
         bach_degree = B06009_005E,
         male = B01001_002E,
         nowork_12 = B99233_005E) %>%
  select(-ends_with("M", ignore.case = F), -GEOID) %>%
  mutate(zcta = str_sub(NAME, 6)) %>%
  select(-NAME) %>%
  select(zcta, everything()) %>%
  mutate(year = as.numeric(year)) %>%
  mutate_at(vars(-zcta, -year, -total_pop, -med_age), list(~(./total_pop)*100))

#linear imputation of 2020 until 2020 ACS release (12/9/2021)
acs_2020 <- acs %>%
  complete(zcta, year = 2016:2020) %>%
  group_by(zcta) %>%
  mutate_at(vars(-zcta, -year),
            funs(if(sum(!is.na(.))<2) {.} else{na_interpolation(., option = "linear")})) %>%
  filter(year==2020)

acs_imp <- acs %>%
  rbind(acs_2020) %>%
  mutate(zcta = as.numeric(zcta))

#joining to hospital data
hosp_panel <- hosp_zcta %>%
  left_join(acs_imp, by = c("zipcode"="zcta", "year"))

#SF geometries - get all ZCTAs
zcta <- get_acs(geography = "zcta",
                   variables = "B01001_001",
                   output = "wide",
                   year = 2019,
                   geometry = T,
                   survey = "acs5") %>%
  rename(zcta = GEOID,
         pop_2019 = B01001_001E) %>%
  select(-c(NAME, B01001_001M, pop_2019)) %>%
  mutate(zcta = as.numeric(zcta))
```

```
##   |                                                                              |
```

```r
#minneapolis shapefile (source: openminneapolis.gov)
mpls <- st_read("mpls_city-shp/16cdbbfa-ad10-493c-afaf-52b61f2e76e42020329-1-180h9ap.whbo.shp") %>%
   st_set_crs(st_crs(zcta))
```

```
## Reading layer ‘16cdbbfa-ad10-493c-afaf-52b61f2e76e42020329-1-180h9ap.whbo’ from data source ‘C:\Users
## Simple feature collection with 1 feature and 4 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -93.32911 ymin: 44.89059 xmax: -93.19433 ymax: 45.05125
## Geodetic CRS:  WGS 84
```

```
#joining to panel, filter to those ZCTAs intersecting MPLS
panel <- zcta %>%
  left_join(hosp_panel, by = c("zcta"="zipcode")) %>%
  filter(ifelse(lengths(st_intersects(., mpls)) > 0, 1, 0)==1 &
           zcta >=55401)

#creating date bookends
panel <- panel %>%
  group_by(zcta, year) %>%
  mutate(begin_date = ISOweek2date(paste(year, paste0("W", sprintf("%02d", weekofyr)), 1,sep = "-")),
         end_date = begin_date+weeks(1)-days(1))

#number of unique MPLS ZCTAs
n_zcta <- length(unique(panel$zcta))

#vector of intersecting ZCTAs for filtering downstream
zcta_universe <- unique(panel$zcta)
```

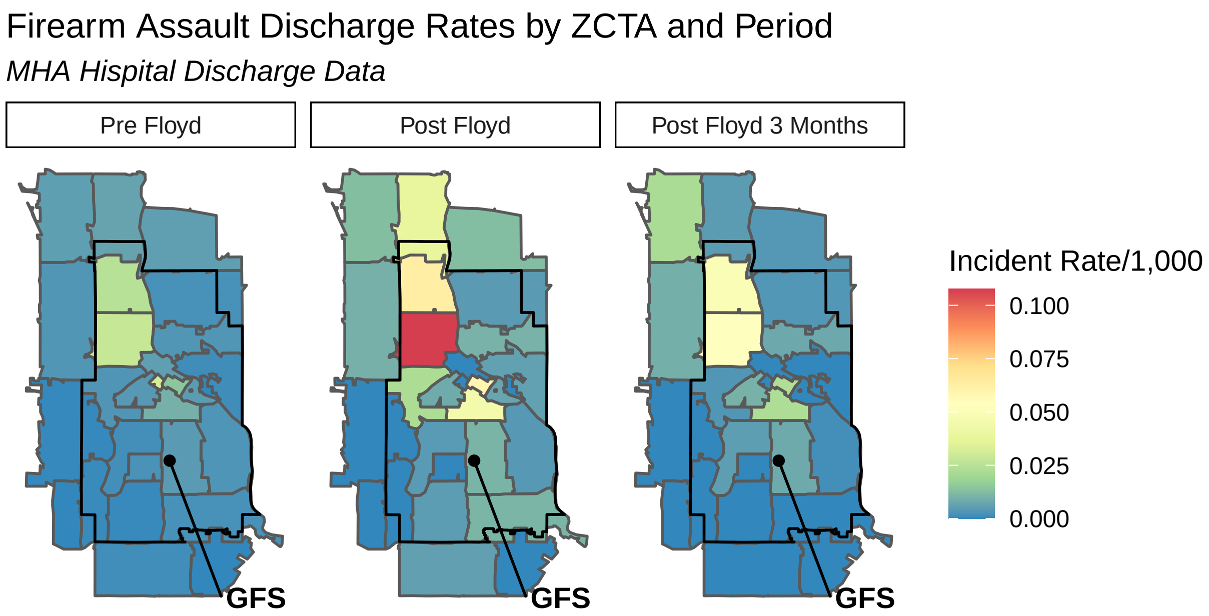
ZCTA-Week Level Police Data

```
#Minneapolis Police Department - Use of Force Dashboard
uof_spatial <- read_csv("Police_Use_Of_Force.csv")  %>%
  mutate(date=ymd_hms(ResponseDate),
         year=year(date),
         week=isoweek(date)) %>%
  select(OBJECTID, year, week, X, Y, Race) %>%
  st_as_sf(coords = c("X", "Y"), crs = "NAD83", remove=F) %>%
  mutate(intersection = as.integer(st_intersects(geometry, zcta)),
         zcta = ifelse(is.na(intersection), NA, zcta$zcta[intersection])) %>%
  st_drop_geometry() %>%
  filter(!is.na(zcta) & year >= 2016 & year <= 2020 & zcta %in% zcta_universe) %>%
  group_by(year, week, zcta, Race, .drop=F) %>%
  tally(name = "use_of_force") %>%
  filter(!is.na(Race) & Race!="not recorded") %>%
  ungroup() %>%
  complete(year, week, zcta=zcta_universe, Race, fill = list(use_of_force = 0)) %>%
  arrange(year, week, zcta, Race) %>%
  mutate(race = str_to_lower(Race)) %>%
  select(-Race) %>%
  pivot_wider(names_from = race,
              values_from = use_of_force,
              values_fill = 0,
              names_glue = "{race}_{.value}") %>%
  mutate(total_use_of_force = asian_use_of_force+black_use_of_force+`native american_use_of_force`+
           `other / mixed race_use_of_force`+`pacific islander_use_of_force`+unknown_use_of_force+
            white_use_of_force)

#MPD Stop Dashboard
```

```
stop_spatial <- read_csv("Police_Stop_Data.csv") %>%
  mutate(date=ymd_hms(responseDate),
         year=year(date),
         week=isoweek(date)) %>%
  select(OBJECTID, year, week, lat, long, race) %>%
  st_as_sf(coords = c("long", "lat"), crs = "NAD83", remove=F) %>%
  mutate(intersection = as.integer(st_intersects(geometry, zcta)),
         zcta = ifelse(is.na(intersection), NA, zcta$zcta[intersection])) %>%
  st_drop_geometry() %>%
  filter(!is.na(zcta) & year >= 2016& year <= 2020 & zcta %in% zcta_universe) %>%
  group_by(year, week, zcta, race, .drop=F) %>%
  tally(name = "police_stops") %>%
   filter(!is.na(race) & race!="not recorded") %>%
  ungroup() %>%
  complete(year, week, zcta=zcta_universe, race, fill = list(police_stops = 0)) %>%
  mutate(race = str_to_lower(race)) %>%
  arrange(year, week, zcta, race) %>%
  pivot_wider(names_from = race,
              values_from = police_stops,
              values_fill = 0,
              names_glue = "{race}_{.value}") %>%
  mutate(total_police_stops = asian_police_stops+black_police_stops+
         `east african_police_stops`+latino_police_stops+`native american_police_stops`+
           other_police_stops+unknown_police_stops+white_police_stops)

#Officer Involved Shootings - MPD
ois_spatial <- read_csv("Police_Officer_Involved_Shootings.csv") %>%
  mutate(date=ymd_hms(IncidentDate),
         year=year(date),
         week=isoweek(date)) %>%
  select(OBJECTID, year, week, CenterLatitude, CenterLongitude, SubjectOfForceRace) %>%
  rename(race = SubjectOfForceRace,
         lat = CenterLatitude,
         long = CenterLongitude) %>%
  st_as_sf(coords = c("long", "lat"), crs = "NAD83", remove=F) %>%
  mutate(intersection = as.integer(st_intersects(geometry, zcta)),
         zcta = ifelse(is.na(intersection), NA, zcta$zcta[intersection])) %>%
  st_drop_geometry() %>%
  filter(!is.na(zcta) & year >= 2016 & year <= 2020 & zcta %in% zcta_universe) %>%
  group_by(year, week, zcta, race, .drop=F) %>%
  tally(name = "police_shootings") %>%
  filter(!is.na(race) & race!="not recorded") %>%
  ungroup() %>%
  complete(year=2016:2021, week=1:53, zcta=zcta_universe, race, fill = list(police_shootings = 0)) %>%
  mutate(race = str_to_lower(race)) %>%
  arrange(year, week, zcta, race) %>%
  pivot_wider(names_from = race,
              values_from = police_shootings,
              values_fill = 0,
              names_glue = "{race}_{.value}") %>%
  mutate(total_police_shootings = asian_police_shootings+black_police_shootings+
         hispanic_police_shootings+other_police_shootings+
           unknown_police_shootings+white_police_shootings)
```

```
panel <- panel %>%
  left_join(uof_spatial, by = c("year", "weekofyr"="week", "zcta"="zcta")) %>%
  left_join(stop_spatial, by = c("year", "weekofyr"="week", "zcta"="zcta")) %>%
  left_join(ois_spatial, by = c("year", "weekofyr"="week", "zcta"="zcta"))

#creating period indicators for panel
panel <- panel %>%
  mutate(post_floyd = ifelse(begin_date >= as.Date("2020-05-25"), T, F),
         post_floyd_3 = ifelse(begin_date >= (as.Date("2020-05-25")+months(3)), T, F),
         stay_at_home = ifelse( begin_date >= as.Date("2020-03-28"), T, F),
         state_of_emerg = ifelse( begin_date >= as.Date("2020-03-13"), T, F),
         period = factor(case_when(
           post_floyd==F & post_floyd_3==F ~ "Pre Floyd",
           post_floyd==T & post_floyd_3==F ~ "Post Floyd",
           post_floyd==T & post_floyd_3==T ~ "Post Floyd 3 Months"),
           levels = c("Pre Floyd", "Post Floyd", "Post Floyd 3 Months"))) %>%
  group_by(zcta) %>%
  arrange(year, weekofyr) %>%
  mutate(t = row_number()) %>%
  ungroup()
```

```
#aggregate to zip-level over years
zip_level <- panel %>%
  group_by(zcta, period) %>%
  summarize(assault_tot = sum(assault_tot, na.rm = T),
            unintent_tot = sum(unintent_tot, na.rm = T),
            suicide_tot = sum(suicide_tot, na.rm = T),
            undeter_tot = sum(undeter_tot, na.rm = T),
            legal_tot = sum(legal_tot, na.rm = T),
            combined_tot = sum(combined_tot, na.rm = T),
            total_pop = sum(total_pop, na.rm = T)) %>%
  mutate(assault_incid_c = (assault_tot/total_pop)*1000,
         unintent_incid_c = (unintent_tot/total_pop)*1000,
         suicide_incid_c = (suicide_tot/total_pop)*1000,
         undeter_incid_c = (undeter_tot/total_pop)*1000,
         legal_incid_c = (legal_tot/total_pop)*1000,
         combined_incid_c = (combined_tot/total_pop)*1000) %>%
  ungroup() %>%
  st_drop_geometry() %>%
  left_join(zcta, by = "zcta")

#george floyd square
gfs <- geocode("George Floyd Square, Minneapolis", output = "latlon") %>%
  st_as_sf(coords = c("lon", "lat"), crs = "NAD83", remove=F) %>%
  mutate(name = "GFS")

ggplot() +
  geom_sf(data = zip_level, aes(geometry = geometry, fill = assault_incid_c)) +
  geom_sf(data = mpls, aes(geometry = geometry), color = "black", alpha = 0)+
  geom_sf(data = gfs, aes(geometry = geometry), color = "black")+
  geom_text_repel(data = gfs, aes(x=lon, y=lat, label = name),
                  fontface = "bold",
                  nudge_x = 1, nudge_y = -1)+
  facet_wrap(~period)+
  scale_fill_distiller(palette = "Spectral")+
  labs(title = "Firearm Assault Discharge Rates by ZCTA and Period",
       subtitle = "MHA Hispital Discharge Data",
       fill = "Incident Rate/1,000")+
  theme(axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank(),
  panel.background = element_blank(),
  panel.grid.major = element_line(colour="transparent"),
  plot.subtitle = element_text(face="italic"),
  strip.background = element_rect(fill = "white",
                colour = "black"))
```



Panel Analysis

```
fe_model <- lm(assault_incid_c~t+state_of_emerg+stay_at_home+post_floyd+post_floyd_3+
                 as.factor(zcta), data = panel)
summary(fe_model)
```

```
##
## Call:
## lm(formula = assault_incid_c ~ t + state_of_emerg + stay_at_home +
##     post_floyd + post_floyd_3 + as.factor(zcta), data = panel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -4.343  -0.570  -0.257  -0.012 264.320
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.1877902  0.3205897   0.586  0.55805
## t                     0.0018890  0.0009776   1.932  0.05337 .
## state_of_emergTRUE   -0.4499546  0.6625353  -0.679  0.49707
## stay_at_homeTRUE      0.2395587  0.7274686   0.329  0.74193
## post_floydTRUE        0.9726047  0.4079601   2.384  0.01715 *
## post_floyd_3TRUE     -0.8145222  0.3282711  -2.481  0.01311 *
## as.factor(zcta)55402  2.0111402  0.4254101   4.728 2.32e-06 ***
## as.factor(zcta)55403  0.0171097  0.4254101   0.040  0.96792
## as.factor(zcta)55404  0.7702495  0.4254101   1.811  0.07024 .
## as.factor(zcta)55405 -0.0342657  0.4254101  -0.081  0.93580
## as.factor(zcta)55406 -0.1554189  0.4254101  -0.365  0.71487
## as.factor(zcta)55407  0.0580357  0.4254101   0.136  0.89149
## as.factor(zcta)55408 -0.2576650  0.4254101  -0.606  0.54474
## as.factor(zcta)55409 -0.2355336  0.4254101  -0.554  0.57983
## as.factor(zcta)55410 -0.4074094  0.4254101  -0.958  0.33825
## as.factor(zcta)55411  2.9395817  0.4254101   6.910 5.26e-12 ***
## as.factor(zcta)55412  2.3953377  0.4254101   5.631 1.86e-08 ***
## as.factor(zcta)55413 -0.0589963  0.4254101  -0.139  0.88971
## as.factor(zcta)55414 -0.3215257  0.4254101  -0.756  0.44979
## as.factor(zcta)55415  1.2368716  0.4254101   2.907  0.00365 **
## as.factor(zcta)55416 -0.4397597  0.4254101  -1.034  0.30130
## as.factor(zcta)55417 -0.2181726  0.4254101  -0.513  0.60807
## as.factor(zcta)55418 -0.2255753  0.4254101  -0.530  0.59595
## as.factor(zcta)55419 -0.3949362  0.4254101  -0.928  0.35325
## as.factor(zcta)55421  0.1182945  0.4254101   0.278  0.78097
## as.factor(zcta)55422 -0.0498435  0.4254101  -0.117  0.90673
## as.factor(zcta)55423 -0.3055180  0.4254101  -0.718  0.47267
## as.factor(zcta)55424 -0.4634631  0.4254101  -1.089  0.27599
## as.factor(zcta)55429  0.2377426  0.4254101   0.559  0.57628
## as.factor(zcta)55430  0.3594710  0.4254101   0.845  0.39814
## as.factor(zcta)55450 -0.4717190  0.4296206  -1.098  0.27225
## as.factor(zcta)55454  0.0086204  0.4254101   0.020  0.98383
## as.factor(zcta)55455 -0.4634631  0.4254101  -1.089  0.27599
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.869 on 7293 degrees of freedom
##   (10 observations deleted due to missingness)
## Multiple R-squared:  0.0337, Adjusted R-squared:  0.02946
## F-statistic: 7.949 on 32 and 7293 DF,  p-value: < 2.2e-16
```

```
fe_int_model <- lm(assault_incid_c~t+state_of_emerg+stay_at_home+post_floyd+post_floyd_3+as.factor(zcta
                     post_floyd:as.factor(zcta)+post_floyd_3:as.factor(zcta), data = panel)
```

```
summary(fe_int_model)
```

```
##
## Call:
## lm(formula = assault_incid_c ~ t + state_of_emerg + stay_at_home +
##     post_floyd + post_floyd_3 + as.factor(zcta) + post_floyd:as.factor(zcta) +
##     post_floyd_3:as.factor(zcta), data = panel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.771  -0.532  -0.260  -0.014 263.907
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            3.212e-01  3.385e-01   0.949  0.34261
## t                                      1.887e-03  9.761e-04   1.933  0.05328
## state_of_emergTRUE                    -4.498e-01  6.615e-01  -0.680  0.49653
## stay_at_homeTRUE                       2.396e-01  7.264e-01   0.330  0.74154
## post_floydTRUE                        -5.591e-01  1.376e+00  -0.406  0.68449
## post_floyd_3TRUE                      -3.019e-02  1.732e+00  -0.017  0.98610
## as.factor(zcta)55402                   2.291e+00  4.533e-01   5.053 4.45e-07
## as.factor(zcta)55403                  -1.100e-01  4.533e-01  -0.243  0.80824
## as.factor(zcta)55404                   4.259e-01  4.533e-01   0.939  0.34752
## as.factor(zcta)55405                  -2.027e-01  4.533e-01  -0.447  0.65485
## as.factor(zcta)55406                  -2.159e-01  4.533e-01  -0.476  0.63391
## as.factor(zcta)55407                  -6.121e-02  4.533e-01  -0.135  0.89261
## as.factor(zcta)55408                  -3.620e-01  4.533e-01  -0.799  0.42457
## as.factor(zcta)55409                  -2.683e-01  4.533e-01  -0.592  0.55398
## as.factor(zcta)55410                  -4.641e-01  4.533e-01  -1.024  0.30601
## as.factor(zcta)55411                   2.275e+00  4.533e-01   5.018 5.34e-07
## as.factor(zcta)55412                   1.956e+00  4.533e-01   4.314 1.62e-05
## as.factor(zcta)55413                  -1.892e-01  4.533e-01  -0.417  0.67651
## as.factor(zcta)55414                  -4.026e-01  4.533e-01  -0.888  0.37458
## as.factor(zcta)55415                   8.672e-01  4.533e-01   1.913  0.05581
## as.factor(zcta)55416                  -5.009e-01  4.533e-01  -1.105  0.26920
## as.factor(zcta)55417                  -3.139e-01  4.533e-01  -0.692  0.48867
## as.factor(zcta)55418                  -3.143e-01  4.533e-01  -0.693  0.48809
## as.factor(zcta)55419                  -4.810e-01  4.533e-01  -1.061  0.28875
## as.factor(zcta)55421                   2.827e-02  4.533e-01   0.062  0.95028
## as.factor(zcta)55422                  -1.879e-01  4.533e-01  -0.415  0.67850
## as.factor(zcta)55423                  -3.840e-01  4.533e-01  -0.847  0.39706
## as.factor(zcta)55424                  -5.279e-01  4.533e-01  -1.165  0.24424
## as.factor(zcta)55429                   2.147e-02  4.533e-01   0.047  0.96222
## as.factor(zcta)55430                   1.395e-01  4.533e-01   0.308  0.75830
## as.factor(zcta)55450                  -5.343e-01  4.585e-01  -1.165  0.24394
## as.factor(zcta)55454                  -3.355e-02  4.533e-01  -0.074  0.94100
## as.factor(zcta)55455                  -5.279e-01  4.533e-01  -1.165  0.24424
## post_floydTRUE:as.factor(zcta)55402   -2.291e+00  1.893e+00  -1.210  0.22614
## post_floydTRUE:as.factor(zcta)55403    9.612e-01  1.893e+00   0.508  0.61157
## post_floydTRUE:as.factor(zcta)55404    4.078e+00  1.893e+00   2.154  0.03123
## post_floydTRUE:as.factor(zcta)55405    2.443e+00  1.893e+00   1.291  0.19682
## post_floydTRUE:as.factor(zcta)55406    6.415e-01  1.893e+00   0.339  0.73464
## post_floydTRUE:as.factor(zcta)55407    1.107e+00  1.893e+00   0.585  0.55863
## post_floydTRUE:as.factor(zcta)55408    8.122e-01  1.893e+00   0.429  0.66782
## post_floydTRUE:as.factor(zcta)55409    2.683e-01  1.893e+00   0.142  0.88727
## post_floydTRUE:as.factor(zcta)55410    4.641e-01  1.893e+00   0.245  0.80630
## post_floydTRUE:as.factor(zcta)55411    8.491e+00  1.893e+00   4.486 7.36e-06
## post_floydTRUE:as.factor(zcta)55412    4.392e+00  1.893e+00   2.320  0.02034
## post_floydTRUE:as.factor(zcta)55413    1.191e+00  1.893e+00   0.629  0.52922
## post_floydTRUE:as.factor(zcta)55414    9.990e-01  1.893e+00   0.528  0.59761
## post_floydTRUE:as.factor(zcta)55415    5.067e+00  1.893e+00   2.677  0.00744
## post_floydTRUE:as.factor(zcta)55416    5.009e-01  1.893e+00   0.265  0.79126
## post_floydTRUE:as.factor(zcta)55417    1.388e+00  1.893e+00   0.734  0.46325
## post_floydTRUE:as.factor(zcta)55418    7.857e-01  1.893e+00   0.415  0.67803
## post_floydTRUE:as.factor(zcta)55419    9.918e-01  1.893e+00   0.524  0.60027
## post_floydTRUE:as.factor(zcta)55421    1.221e+00  1.893e+00   0.645  0.51877
## post_floydTRUE:as.factor(zcta)55422    1.146e+00  1.893e+00   0.605  0.54503
## post_floydTRUE:as.factor(zcta)55423    9.743e-01  1.893e+00   0.515  0.60672
## post_floydTRUE:as.factor(zcta)55424    5.279e-01  1.893e+00   0.279  0.78029
## post_floydTRUE:as.factor(zcta)55429    1.259e+00  1.893e+00   0.665  0.50605
## post_floydTRUE:as.factor(zcta)55430    3.662e+00  1.893e+00   1.935  0.05303
## post_floydTRUE:as.factor(zcta)55450    5.343e-01  1.894e+00   0.282  0.77787
## post_floydTRUE:as.factor(zcta)55454    7.461e-01  1.893e+00   0.394  0.69342
## post_floydTRUE:as.factor(zcta)55455    5.279e-01  1.893e+00   0.279  0.78029
## post_floyd_3TRUE:as.factor(zcta)55402 -5.000e-14  2.450e+00   0.000  1.00000
## post_floyd_3TRUE:as.factor(zcta)55403  1.419e-01  2.450e+00   0.058  0.95383
## post_floyd_3TRUE:as.factor(zcta)55404 -2.237e+00  2.450e+00  -0.913  0.36123
## post_floyd_3TRUE:as.factor(zcta)55405 -1.892e+00  2.450e+00  -0.772  0.44005
## post_floyd_3TRUE:as.factor(zcta)55406 -2.601e-01  2.450e+00  -0.106  0.91545
## post_floyd_3TRUE:as.factor(zcta)55407 -2.324e-01  2.450e+00  -0.095  0.92443
## post_floyd_3TRUE:as.factor(zcta)55408  7.504e-02  2.450e+00   0.031  0.97557
## post_floyd_3TRUE:as.factor(zcta)55409 -5.454e-14  2.450e+00   0.000  1.00000
## post_floyd_3TRUE:as.factor(zcta)55410 -4.725e-14  2.450e+00   0.000  1.00000
## post_floyd_3TRUE:as.factor(zcta)55411 -5.421e+00  2.450e+00  -2.213  0.02695
## post_floyd_3TRUE:as.factor(zcta)55412 -1.411e+00  2.450e+00  -0.576  0.56481
## post_floyd_3TRUE:as.factor(zcta)55413 -2.226e-01  2.450e+00  -0.091  0.92761
## post_floyd_3TRUE:as.factor(zcta)55414 -5.965e-01  2.450e+00  -0.243  0.80766
## post_floyd_3TRUE:as.factor(zcta)55415 -3.626e+00  2.450e+00  -1.480  0.13889
## post_floyd_3TRUE:as.factor(zcta)55416 -3.764e-14  2.450e+00   0.000  1.00000
## post_floyd_3TRUE:as.factor(zcta)55417 -1.074e+00  2.450e+00  -0.439  0.66102
## post_floyd_3TRUE:as.factor(zcta)55418 -1.048e-01  2.450e+00  -0.043  0.96590
## post_floyd_3TRUE:as.factor(zcta)55419 -5.108e-01  2.450e+00  -0.208  0.83485
## post_floyd_3TRUE:as.factor(zcta)55421 -8.608e-01  2.450e+00  -0.351  0.72534
## post_floyd_3TRUE:as.factor(zcta)55422 -2.660e-02  2.450e+00  -0.011  0.99134
## post_floyd_3TRUE:as.factor(zcta)55423 -5.903e-01  2.450e+00  -0.241  0.80960
## post_floyd_3TRUE:as.factor(zcta)55424 -2.575e-14  2.450e+00   0.000  1.00000
## post_floyd_3TRUE:as.factor(zcta)55429  9.103e-01  2.450e+00   0.372  0.71024
## post_floyd_3TRUE:as.factor(zcta)55430 -3.309e+00  2.450e+00  -1.351  0.17687
## post_floyd_3TRUE:as.factor(zcta)55450 -2.563e-14  2.450e+00   0.000  1.00000
## post_floyd_3TRUE:as.factor(zcta)55454 -7.126e-01  2.450e+00  -0.291  0.77118
## post_floyd_3TRUE:as.factor(zcta)55455 -1.532e-14  2.450e+00   0.000  1.00000
##
## (Intercept)
## t                                      .
## state_of_emergTRUE
## stay_at_homeTRUE
## post_floydTRUE
## post_floyd_3TRUE
## as.factor(zcta)55402                  ***
## as.factor(zcta)55403
## as.factor(zcta)55404
## as.factor(zcta)55405
## as.factor(zcta)55406
## as.factor(zcta)55407
## as.factor(zcta)55408
## as.factor(zcta)55409
## as.factor(zcta)55410
## as.factor(zcta)55411                  ***
## as.factor(zcta)55412                  ***
## as.factor(zcta)55413
## as.factor(zcta)55414
## as.factor(zcta)55415                  .
## as.factor(zcta)55416
## as.factor(zcta)55417
## as.factor(zcta)55418
## as.factor(zcta)55419
## as.factor(zcta)55421
## as.factor(zcta)55422
## as.factor(zcta)55423
## as.factor(zcta)55424
## as.factor(zcta)55429
## as.factor(zcta)55430
## as.factor(zcta)55450
## as.factor(zcta)55454
## as.factor(zcta)55455
## post_floydTRUE:as.factor(zcta)55402
## post_floydTRUE:as.factor(zcta)55403
## post_floydTRUE:as.factor(zcta)55404   *
## post_floydTRUE:as.factor(zcta)55405
## post_floydTRUE:as.factor(zcta)55406
## post_floydTRUE:as.factor(zcta)55407
## post_floydTRUE:as.factor(zcta)55408
## post_floydTRUE:as.factor(zcta)55409
## post_floydTRUE:as.factor(zcta)55410
## post_floydTRUE:as.factor(zcta)55411   ***
## post_floydTRUE:as.factor(zcta)55412   *
## post_floydTRUE:as.factor(zcta)55413
## post_floydTRUE:as.factor(zcta)55414
## post_floydTRUE:as.factor(zcta)55415   **
## post_floydTRUE:as.factor(zcta)55416
## post_floydTRUE:as.factor(zcta)55417
## post_floydTRUE:as.factor(zcta)55418
## post_floydTRUE:as.factor(zcta)55419
## post_floydTRUE:as.factor(zcta)55421
## post_floydTRUE:as.factor(zcta)55422
## post_floydTRUE:as.factor(zcta)55423
## post_floydTRUE:as.factor(zcta)55424
## post_floydTRUE:as.factor(zcta)55429
## post_floydTRUE:as.factor(zcta)55430   .
## post_floydTRUE:as.factor(zcta)55450
## post_floydTRUE:as.factor(zcta)55454
## post_floydTRUE:as.factor(zcta)55455
## post_floyd_3TRUE:as.factor(zcta)55402
## post_floyd_3TRUE:as.factor(zcta)55403
## post_floyd_3TRUE:as.factor(zcta)55404
## post_floyd_3TRUE:as.factor(zcta)55405
## post_floyd_3TRUE:as.factor(zcta)55406
## post_floyd_3TRUE:as.factor(zcta)55407
## post_floyd_3TRUE:as.factor(zcta)55408
## post_floyd_3TRUE:as.factor(zcta)55409
## post_floyd_3TRUE:as.factor(zcta)55410
## post_floyd_3TRUE:as.factor(zcta)55411 *
## post_floyd_3TRUE:as.factor(zcta)55412
## post_floyd_3TRUE:as.factor(zcta)55413
## post_floyd_3TRUE:as.factor(zcta)55414
## post_floyd_3TRUE:as.factor(zcta)55415
## post_floyd_3TRUE:as.factor(zcta)55416
## post_floyd_3TRUE:as.factor(zcta)55417
## post_floyd_3TRUE:as.factor(zcta)55418
## post_floyd_3TRUE:as.factor(zcta)55419
## post_floyd_3TRUE:as.factor(zcta)55421
## post_floyd_3TRUE:as.factor(zcta)55422
## post_floyd_3TRUE:as.factor(zcta)55423
## post_floyd_3TRUE:as.factor(zcta)55424
## post_floyd_3TRUE:as.factor(zcta)55429
## post_floyd_3TRUE:as.factor(zcta)55430
## post_floyd_3TRUE:as.factor(zcta)55450
## post_floyd_3TRUE:as.factor(zcta)55454
## post_floyd_3TRUE:as.factor(zcta)55455
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.862 on 7239 degrees of freedom
##   (10 observations deleted due to missingness)
## Multiple R-squared:  0.04378,    Adjusted R-squared:  0.03242
## F-statistic: 3.854 on 86 and 7239 DF,  p-value: < 2.2e-16
```

```
#map of post_floyd coefficients by zip? color just significant from 0?
```

Time Series Construction - Week Level

Aggregate Hospital Panel to Week-Level

```
#panel to week-level, aggregating over ZCTAs
hosp_series <- panel %>%
  group_by(year, weekofyr) %>%
  summarize(assault_tot = sum(assault_tot, na.rm = T),
            unintent_tot = sum(unintent_tot, na.rm = T),
            suicide_tot = sum(suicide_tot, na.rm = T),
            undeter_tot = sum(undeter_tot, na.rm = T),
            legal_tot = sum(legal_tot, na.rm = T),
            combined_tot = sum(combined_tot, na.rm = T),
            total_pop = sum(total_pop, na.rm = T)) %>%
  mutate(assault_incid_c = (assault_tot/total_pop)*1000,
         unintent_incid_c = (unintent_tot/total_pop)*1000,
         suicide_incid_c = (suicide_tot/total_pop)*1000,
         undeter_incid_c = (undeter_tot/total_pop)*1000,
         legal_incid_c = (legal_tot/total_pop)*1000,
         combined_incid_c = (combined_tot/total_pop)*1000) %>%
  ungroup() %>%
  mutate(week_id = row_number()) %>%
  st_drop_geometry()
```

Police Data Week-Level

```
#Minneapolis Police Department - Use of Force Dashboard
uof <- read_csv("Police_Use_Of_Force.csv") %>%
  mutate(date=ymd_hms(ResponseDate),
         year=year(date),
         week=isoweek(date)) %>%
  group_by(year, week, .drop=F) %>%
  tally(name = "use_of_force") %>%
  arrange(year, week) %>%
  ungroup() %>%
  select(year, week, everything())

#merge onto series
series <- hosp_series %>%
  left_join(uof, by=c("year", "weekofyr"="week")) %>%
  mutate(use_of_force_rate = (use_of_force/total_pop)*1000)

#MPD Officer Involved Shootings
ois <- read_csv("Police_Officer_Involved_Shootings.csv") %>%
  mutate(date=ymd_hms(IncidentDate),
         year=year(date),
         week=isoweek(date)) %>%
  group_by(year, week, .drop=F) %>%
  tally(name = "off_inv_shooting") %>%
  arrange(year, week) %>%
  ungroup() %>%
  select(year, week, everything())

#merge onto series
series <- series %>%
  left_join(ois, by=c("year", "weekofyr"="week")) %>%
  mutate(off_inv_shooting = ifelse(is.na(off_inv_shooting), 0, off_inv_shooting),
         off_inv_shooting_rate = (off_inv_shooting/total_pop)*1000)


#Minneapolis Police Department - Police Stops Dashboard
stop <- read_csv("Police_Stop_Data.csv") %>%
  mutate(date=ymd_hms(responseDate),
         year=year(date),
         week=isoweek(date)) %>%
  group_by(year, week, .drop=F) %>%
  tally(name = "police_stops")

#merge onto series
series <- series %>%
  left_join(stop, by = c("year", "weekofyr"="week")) %>%
  mutate(police_stops = ifelse(is.na(police_stops), 0, police_stops),
         police_stop_rate = (police_stops/total_pop)*1000)

#New York Times COVID Case/Mortality Data
covid_hennepin <- nytcovcounty %>%
  mutate(week = isoweek(date),
         year = year(date)) %>%
  filter(county=="Hennepin" & state=="Minnesota" & year >=2019) %>%
  group_by(year, week, .drop=F) %>%
  summarize(covid_cases = sum(cases, na.rm = T),
            covid_deaths = sum(deaths, na.rm = T))

#filling 0s for pre-covid series
series <- series %>%
  left_join(covid_hennepin, by = c("year", "weekofyr"="week")) %>%
  mutate_at(vars(c(covid_cases, covid_deaths)), ~ifelse(is.na(.), 0, .))

#creating date variable
series <- series %>%
  mutate(begin_date = ISOweek2date(paste(year, paste0("W", sprintf("%02d", weekofyr)), 1,sep = "-")),
         end_date = begin_date+weeks(1)-days(1))
```

Weather Data

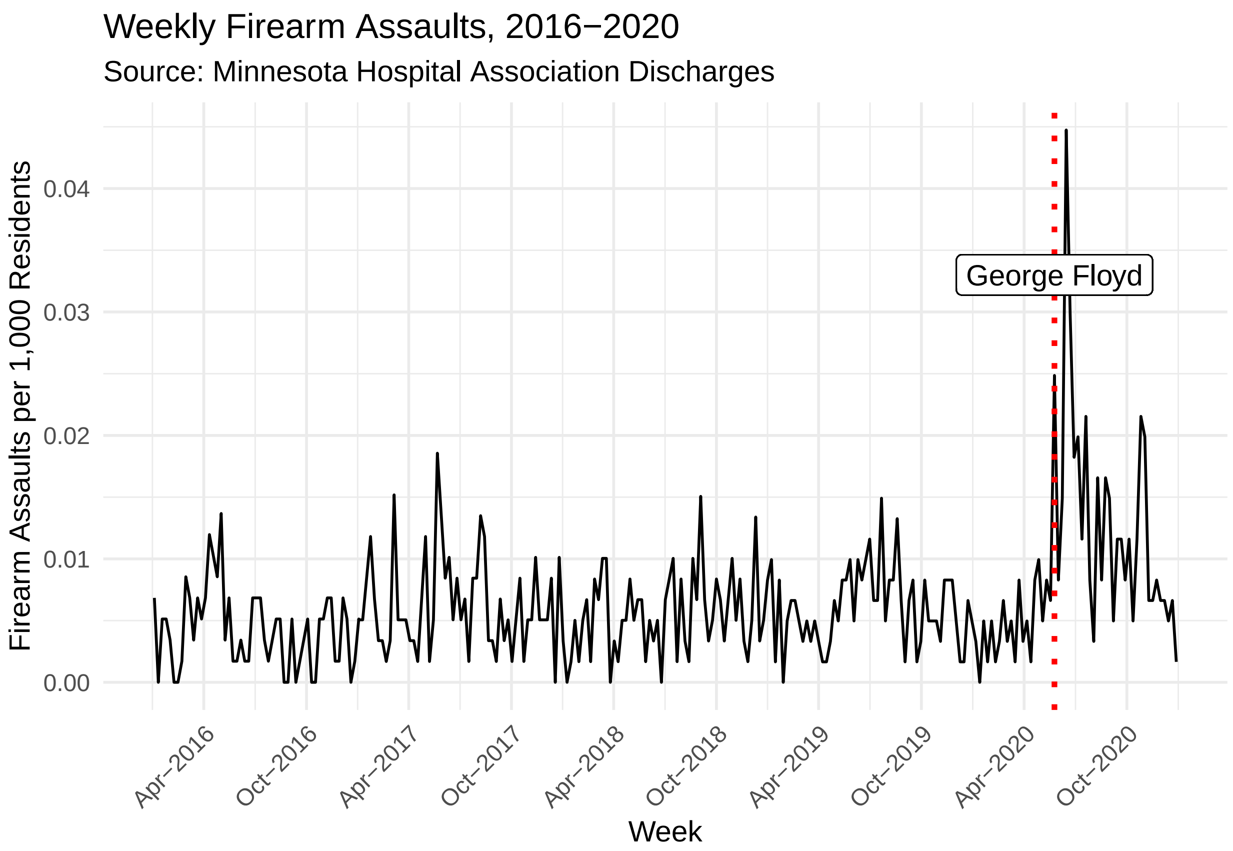
```
#weatherData
noaa_token <- "nrEVQsKalyfOauURFnZPpNZQwbAiPaSh"

#list of ncdc stations with coverage over MPLS ZCTAs
stations <- ncdc_stations(extent = as.vector(st_bbox(zcta[zcta$zcta %in% zcta_universe,]))[order(c(2,1,
              token = noaa_token, startdate = "2016-01-01", enddate = "2020-12-31")

ncdc_datasets(stationid=stations$data$id, token=noaa_token)

datatypes <- c('PRCP','SNOW', 'TMAX')

begin_date <- as.list(series$begin_date)

weather <- map_df(.x = begin_date,
              ~ ncdc(datasetid='GHCND',
              datatypeid=datatypes,
              stationid=ncdc_stations(extent = as.vector(st_bbox(zcta[zcta$zcta %in% zcta_universe,]))[
              token = noaa_token, startdate = .x, enddate = .x+weeks(1))$data$id,
              startdate = .x,
              enddate = .x+weeks(1),
              limit=1000,
              token = noaa_token)$data %>%
  group_by(date, datatype) %>%
  summarize(value = mean(value, na.rm = T)) %>%
  pivot_wider(id_cols = date, names_from = datatype, values_from = value))
```

Sunset Data

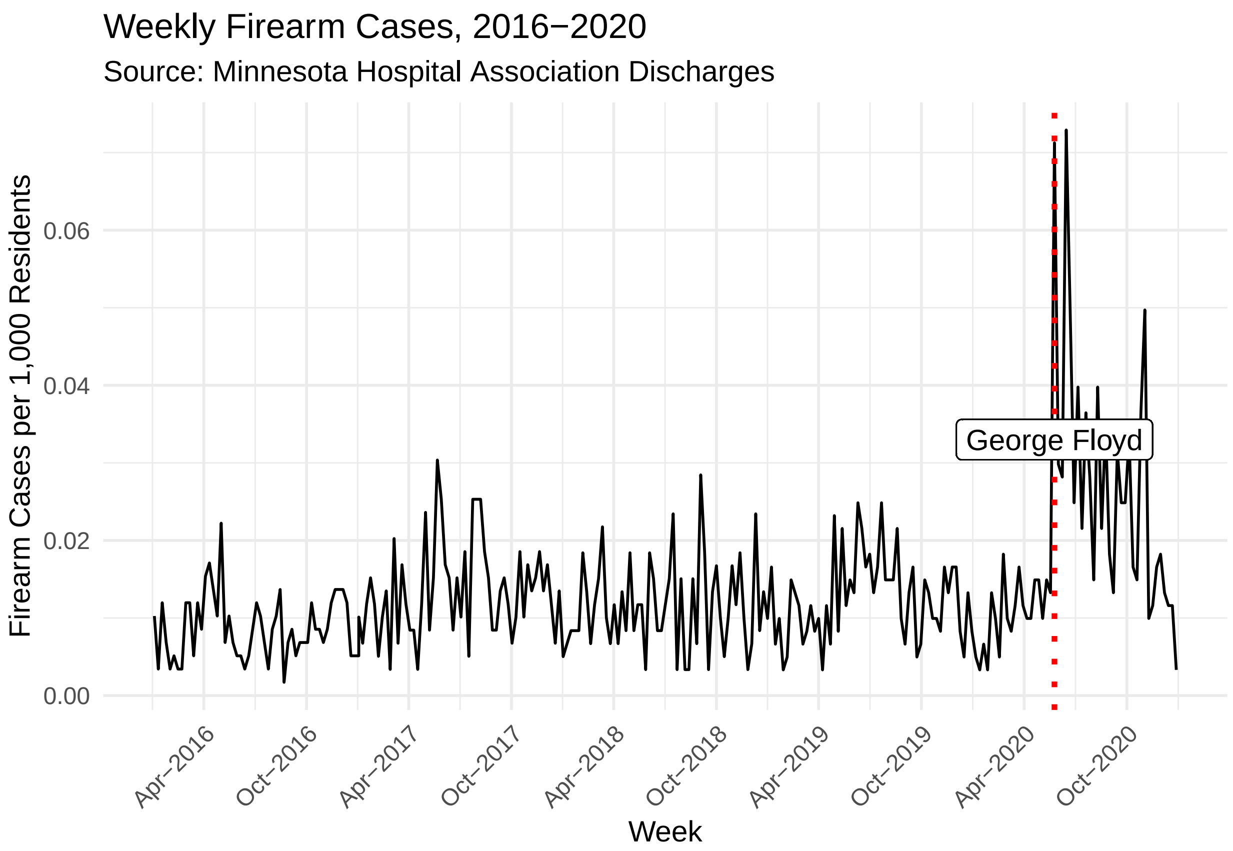
School Data

Time Series Vizualization

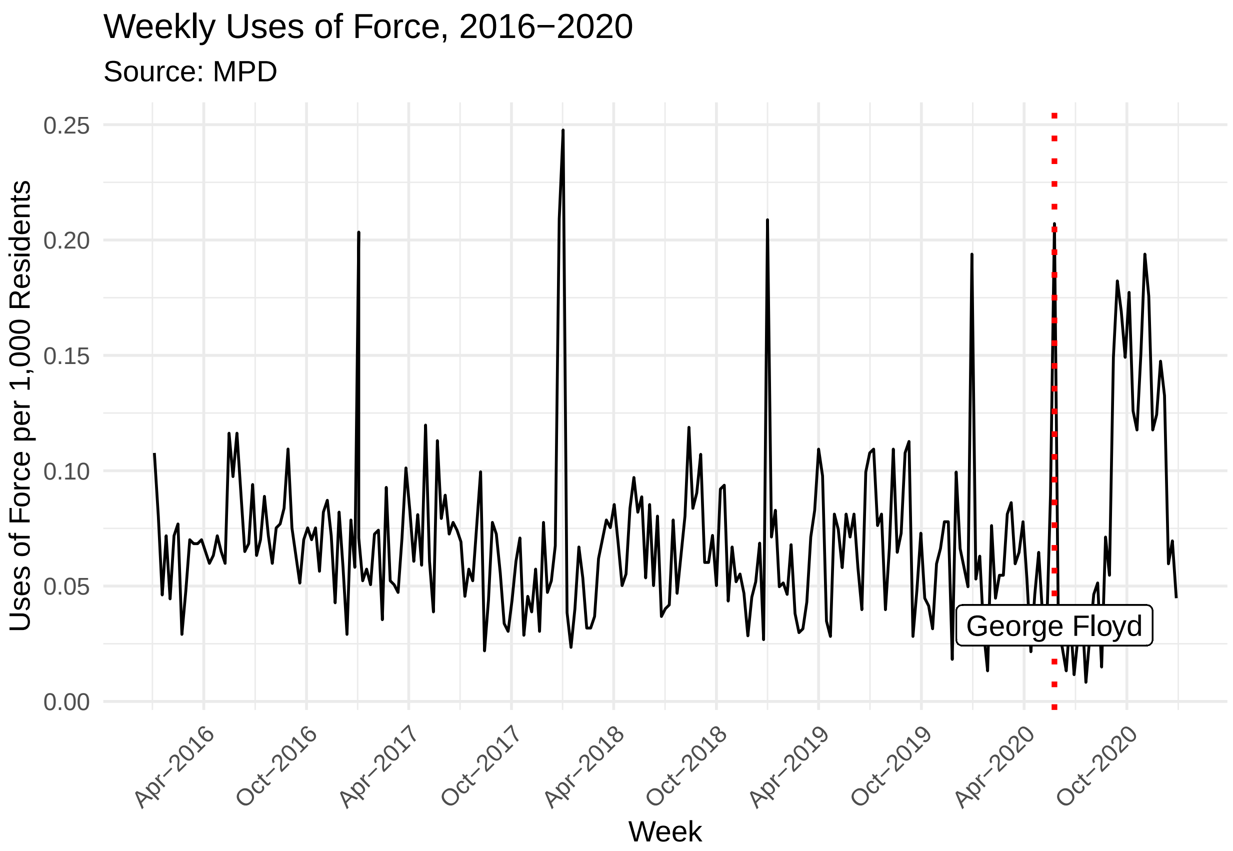
```
ggplot(series)+
  geom_line(aes(x=begin_date, y=assault_incid_c))+
  scale_x_date(date_labels = "%b-%Y", date_breaks = "6 months")+
  geom_vline(xintercept=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25
              linetype="dotted", color="red", size=1)+
  geom_label(aes(x=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25"))],
                 y=0.033),
             label = "George Floyd", show.legend = FALSE)+
  labs(title = "Weekly Firearm Assaults, 2016-2020",
       subtitle = "Source: Minnesota Hospital Association Discharges",
       x = "Week",
       y = "Firearm Assaults per 1,000 Residents")+
  theme_minimal()+
  theme(axis.text.x=element_text(angle=45, hjust=1))
```



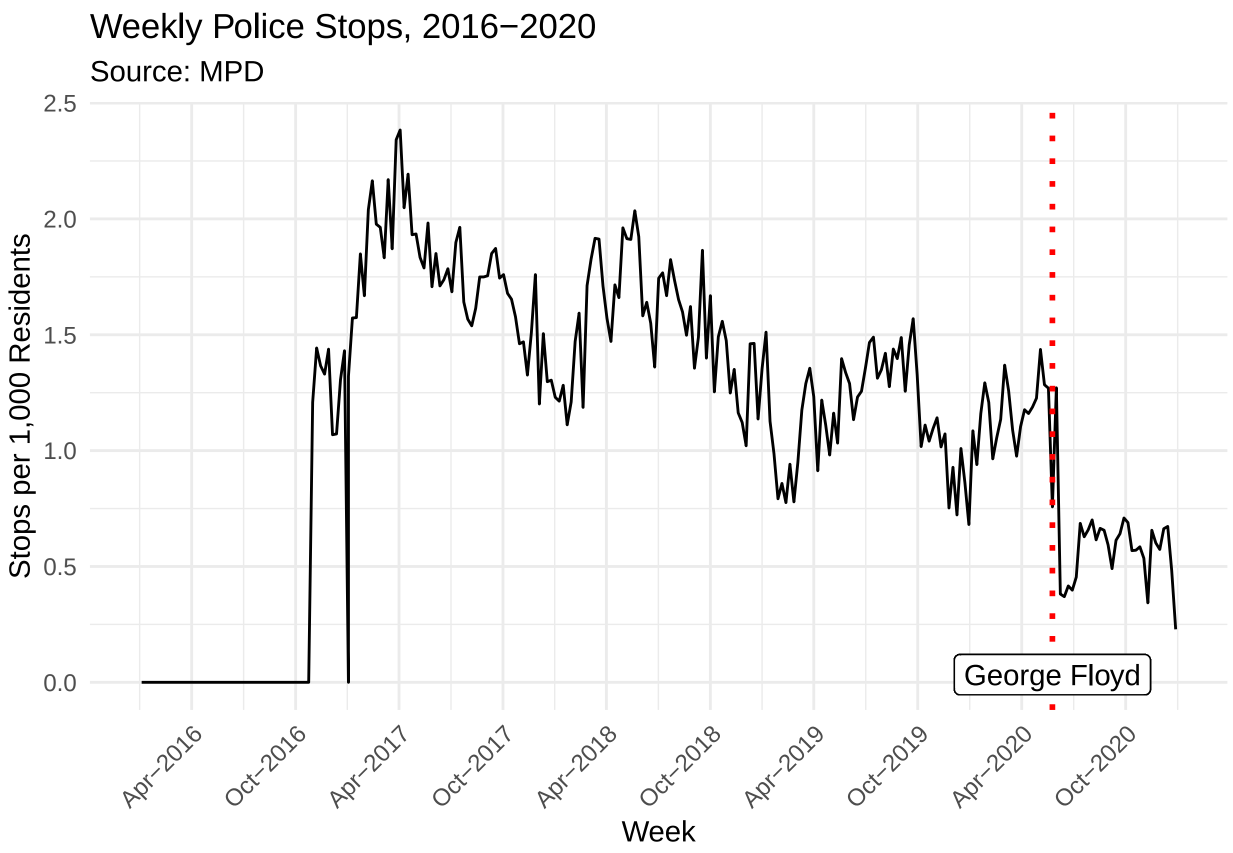
```
ggplot(series)+
  geom_line(aes(x=begin_date, y=combined_incid_c))+
  scale_x_date(date_labels = "%b-%Y", date_breaks = "6 months")+
  geom_vline(xintercept=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25
              linetype="dotted", color="red", size=1)+
  geom_label(aes(x=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25"))],
                 y=0.033),
             label = "George Floyd", show.legend = FALSE)+
  labs(title = "Weekly Firearm Cases, 2016-2020",
       subtitle = "Source: Minnesota Hospital Association Discharges",
       x = "Week",
       y = "Firearm Cases per 1,000 Residents")+
  theme_minimal()+
  theme(axis.text.x=element_text(angle=45, hjust=1))
```



```
ggplot(series)+
  geom_line(aes(x=begin_date, y=use_of_force_rate))+
  scale_x_date(date_labels = "%b-%Y", date_breaks = "6 months")+
  geom_vline(xintercept=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25
              linetype="dotted", color="red", size=1)+
  geom_label(aes(x=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25"))],
                 y=0.033),
             label = "George Floyd", show.legend = FALSE)+
  labs(title = "Weekly Uses of Force, 2016-2020",
       subtitle = "Source: MPD",
       x = "Week",
       y = "Uses of Force per 1,000 Residents")+
  theme_minimal()+
  theme(axis.text.x=element_text(angle=45, hjust=1))
```

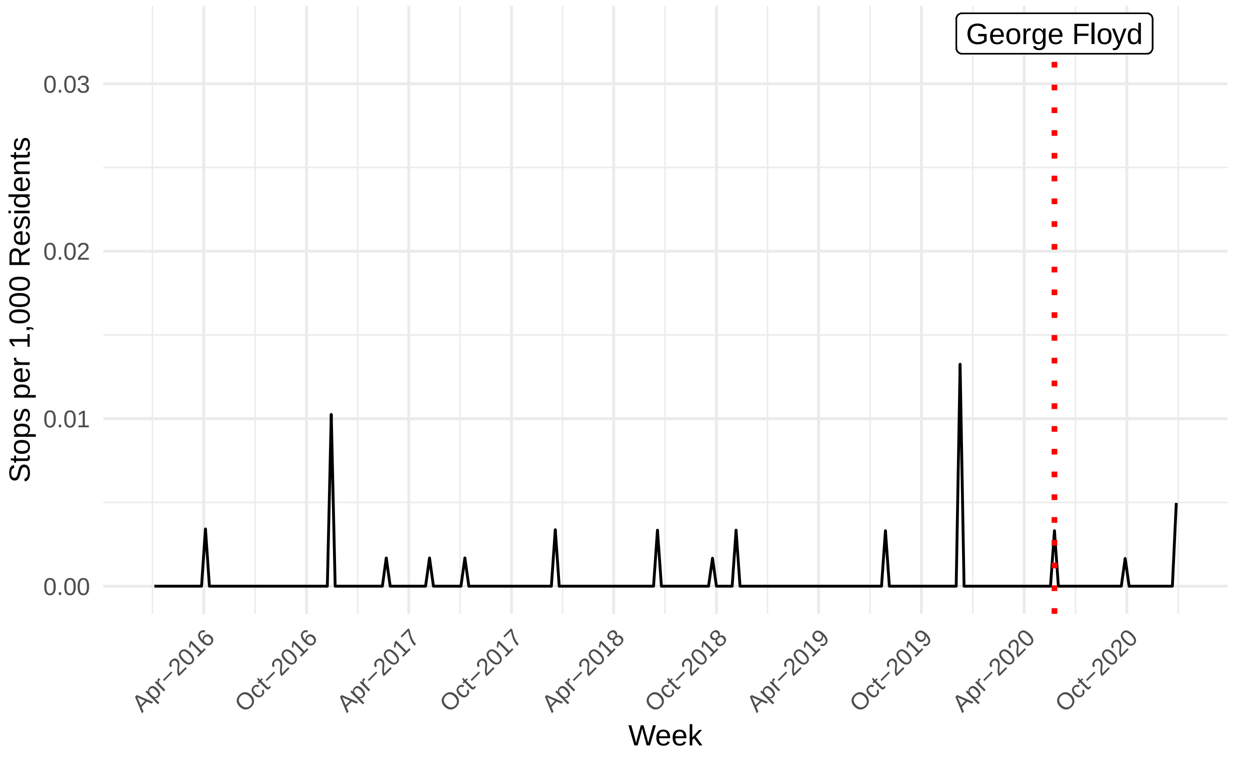


```
ggplot(series)+
  geom_line(aes(x=begin_date, y=police_stop_rate))+
  scale_x_date(date_labels = "%b-%Y", date_breaks = "6 months")+
  geom_vline(xintercept=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25
              linetype="dotted", color="red", size=1)+
  geom_label(aes(x=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25"))],
                 y=0.033),
             label = "George Floyd", show.legend = FALSE)+
  labs(title = "Weekly Police Stops, 2016-2020",
       subtitle = "Source: MPD",
       x = "Week",
       y = "Stops per 1,000 Residents")+
  theme_minimal()+
  theme(axis.text.x=element_text(angle=45, hjust=1))
```



```
ggplot(series)+
  geom_line(aes(x=begin_date, y=off_inv_shooting_rate))+
  scale_x_date(date_labels = "%b-%Y", date_breaks = "6 months")+
  geom_vline(xintercept=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25
             linetype="dotted", color="red", size=1)+
  geom_label(aes(x=series$begin_date[series$year==2020 & series$weekofyr==isoweek(date("2020-05-25"))],
                 y=0.033),
             label = "George Floyd", show.legend = FALSE)+
  labs(title = "Weekly Officer Involved Shootings, 2016-2020",
       subtitle = "Source: MPD",
       x = "Week",
       y = "Stops per 1,000 Residents")+
  theme_minimal()+
  theme(axis.text.x=element_text(angle=45, hjust=1))
```

```
## Warning: Use of `series$begin_date` is discouraged. Use `begin_date` instead.

## Warning: Use of `series$year` is discouraged. Use `year` instead.

## Warning: Use of `series$weekofyr` is discouraged. Use `weekofyr` instead.
```



Weekly Officer Involved Shootings, 2016−2020

Source: MPD

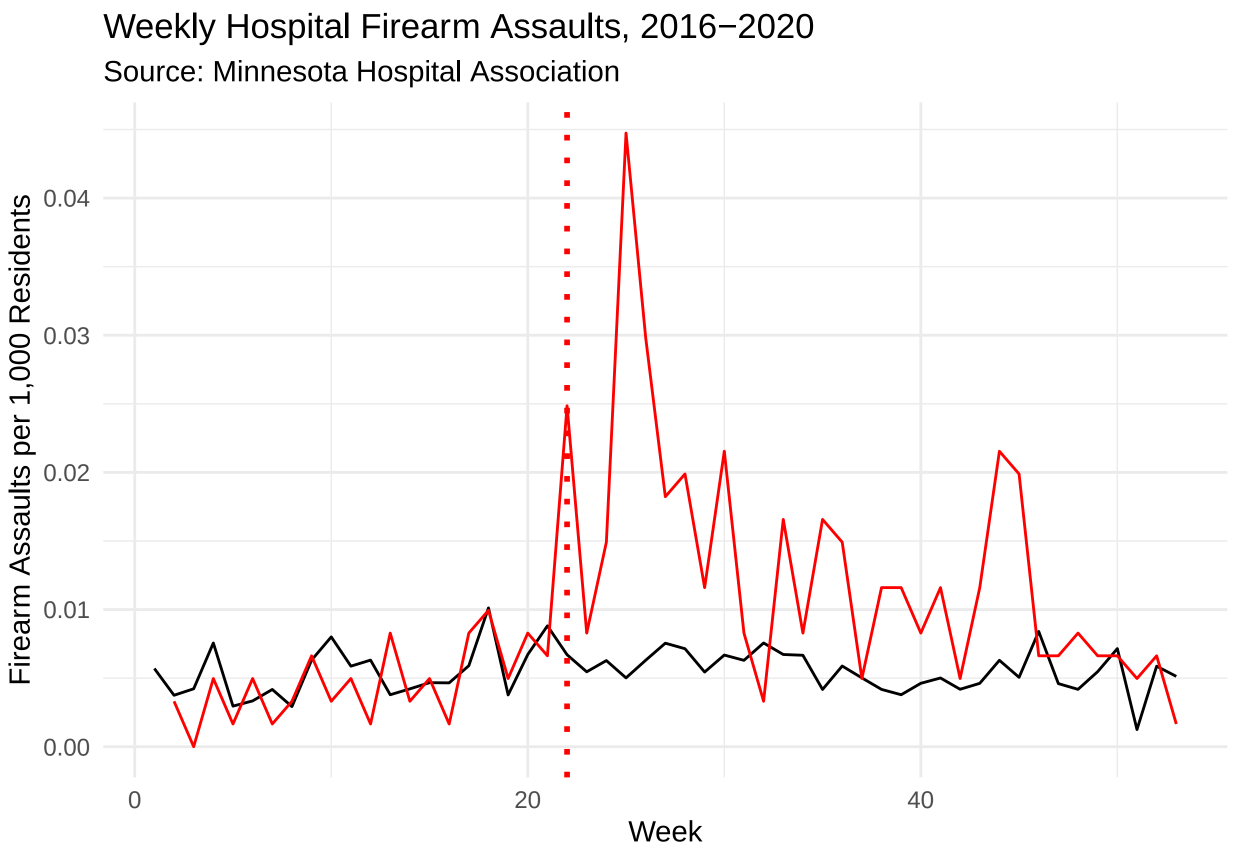
Time Series Analysis

```
window_df <- function(df, outcome, end.date, start.date, end.pre.date) {

  ### Function to generate windowed data frame objects
  ###   for a full time period (start through end)
  ###   and for a pre-treatment period (start through pre)
  ### Inputs
  ###  df: data frame to be windowed
  ###  outcome: column number of a crime or arrest category (integer)
  ###  end.date: last date for full time period (date format)
  ###  start.date: first date for full time period (date format)
  ###  end.pre.date: last date for pre-treatment period (date format)
  ### Returns:
  ###  list containing 2 data frame objects

  ## select variables from data frame and filter

  df.windowed.pre <- select(df, begin_date, year, weekofyr, y = outcome) %>%
                     dplyr::filter(begin_date >= start.date,
                            begin_date <= end.pre.date)
  df.windowed.post <- select(df, begin_date, year, weekofyr, y = outcome) %>%
                     dplyr::filter(begin_date >= end.pre.date)
  df.windowed.all <- select(df, begin_date, year, weekofyr, y = outcome) %>%
```

```r
                     dplyr::filter(begin_date >= start.date,
                            begin_date <= end.date)
  df.pre.agg <- df.windowed.pre %>%
    group_by(weekofyr) %>%
    summarize(y = mean(y, na.rm = T))


  ## return list

  list.df <- list(df.windowed.pre, df.windowed.post, df.windowed.all, df.pre.agg)

  return(list.df)
}

pre_2020 <- window_df(series,
               outcome = "assault_incid_c",
               end.date = "2020-12-31",
               start.date = "2016-01-01",
               end.pre.date = "2020-01-01")
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(outcome)‘ instead of ‘outcome‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```r
ggplot()+
  geom_line(data = as.data.frame(pre_2020[[4]]), aes(x=weekofyr, y=y))+
  geom_line(data = as.data.frame(pre_2020[[2]]), aes(x=weekofyr, y=y), color = "red")+
  geom_vline(xintercept=isoweek("2020-05-25"),
              linetype="dotted", color="red", size=1)+
   labs(title = "Weekly Hospital Firearm Assaults, 2016-2020",
       subtitle = "Source: Minnesota Hospital Association ",
       x = "Week",
       y = "Firearm Assaults per 1,000 Residents")+
  theme_minimal()
```



Weekly Hospital Firearm Assaults, 2016−2020

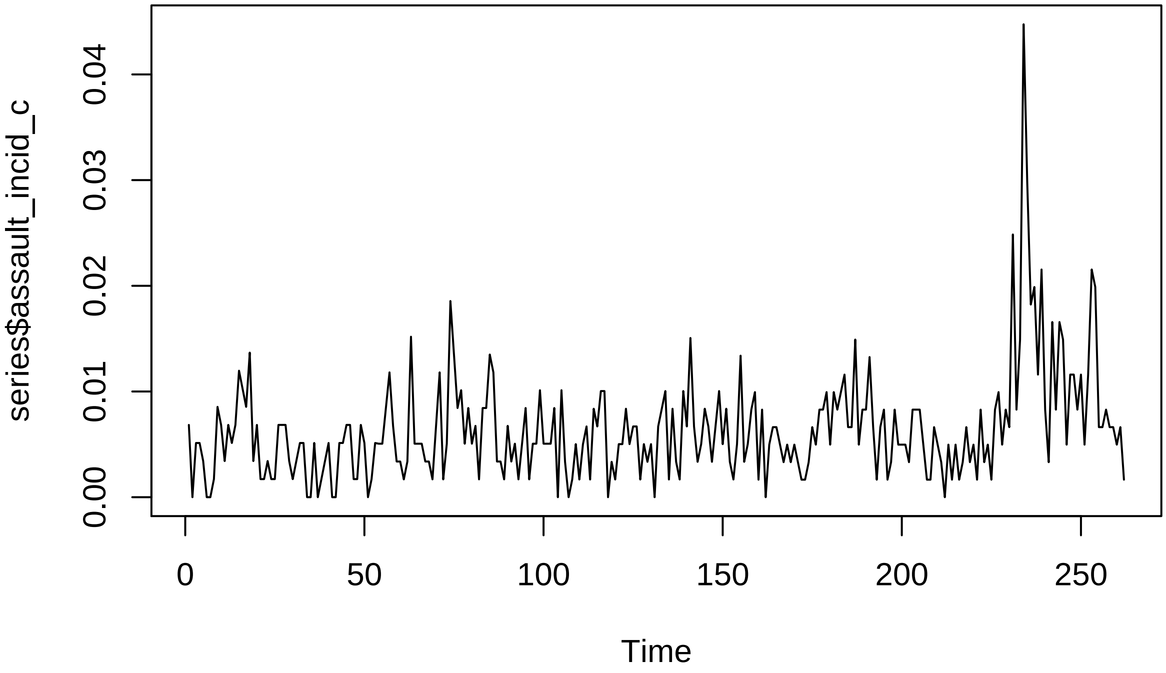
Source: Minnesota Hospital Association

```
#moregan-pally intervention least squares models
intervention.models <- function(dataset, i, end.date, start.date, end.pre.date) {

  ### Estimate five linear models from windowed dataset, return as a list
  ###   calls on window_df() function
  ### Inputs
  ###  dataset: data frame with data to be modeled
  ###  outcome: column number of a crime or arrest category (integer)
  ###  end.date: last date for full time period (date format)
  ###  start.date: first date for full time period (date format)
  ###  end.pre.date: last date for pre-treatment period (date format)
  ### Returns:
  ###  list containing five linear models and mean predicted value for
  ###   the last 52 weeks of the pre-Ferguson period

  ## set up data, specifying outcome as i and setting ending date for window
  ##  use window.df.and.zoo function defined elsewhere

  w <- window_df(dataset, i, end.date, start.date, end.pre.date)
  df.windowed.pre <- w[[1]]
  df.windowed.all <- w[[2]]

  df.windowed.pre.original <- df.windowed.pre

  ## create linear time variables
```

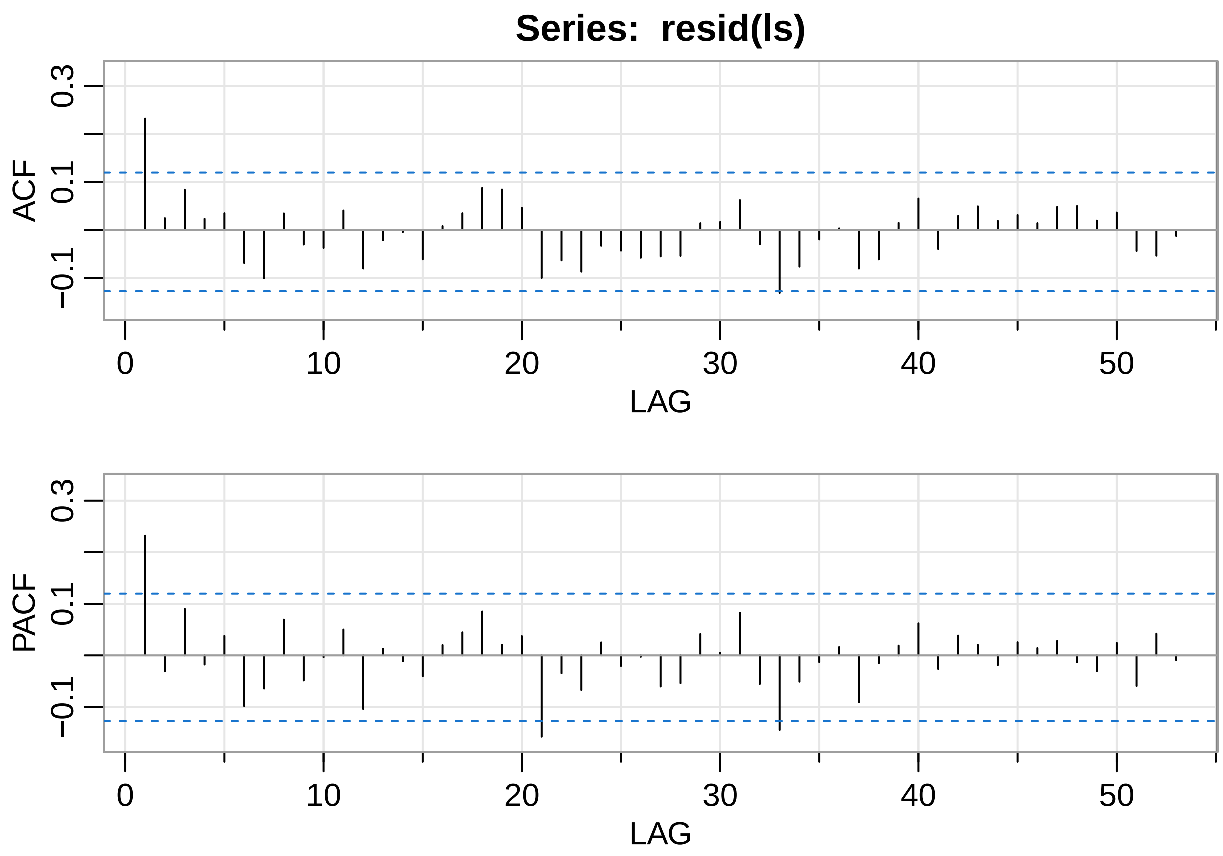
```r
df.windowed.pre$t <- 1:length(df.windowed.pre$y)
df.windowed.all$t <- 1:length(df.windowed.all$y)

## create linear spline for temperature

parameterize.spline <- function(x, c) ifelse (x > c, x - c, 0)
tmax.f.knots <- c(0, 50, 60, 70, 80)
df.windowed.pre$tmax.f.spline <- outer(df.windowed.pre$tmax.f,
                                       tmax.f.knots, parameterize.spline)
df.windowed.all$tmax.f.spline <- outer(df.windowed.all$tmax.f,
                                       tmax.f.knots, parameterize.spline)

## create spike and period type variables

#post.floyd
df.windowed.all$post.floyd <- as.numeric(df.windowed.all$week.first >= as.Date("2020-05-25"))

#three-month post.floyd
df.windowed.all$post.floyd.3 <- as.numeric(df.windowed.all$week.first >= as.Date("2020-05-25")+months

#stay at home order - covid
df.windowed.all$stay.at.home <- as.numeric(df.windowed.all$week.first >= as.Date("2020-03-28") &
#state of emergency - covid
 df.windowed.all$state.of.emerg <- as.numeric(df.windowed.all$week.first >= as.Date("2020-03-13"))

## specify model, estimate, and store selected output

naive.int.model.formula <- as.formula(paste("y ~ t + state.of.emerg + stay.at.home +
                                            post.floyd+post.floyd.3"))
covariate.model.formula <- as.formula(paste("y ~ t + tmax.f.spline + snow.in +
                                            precip.in + dark.before.12 + school"))
full.int.model.formula <- as.formula(paste("y ~ t +
                                           state.of.emerg + stay.at.home +
                                            post.floyd + post.floyd.3 +
                                           tmax.f.spline + snow.in +
                                           precip.in + dark.before.12 + school"))
constrained.int.model.formula <- as.formula(paste("y.diff ~
                                                  state.of.emerg + stay.at.home +
                                            post.floyd + post.floyd.3"))

ls.naive.int <- lm(naive.int.model.formula, df.windowed.all)
ls.pre <- lm(covariate.model.formula, df.windowed.pre)
ls.all <- lm(covariate.model.formula, df.windowed.all)
ls.full.int <- lm(full.int.model.formula, df.windowed.all)
df.windowed.all$y.predicted <- predict(ls.pre, df.windowed.all,
                                       type = "response")
df.windowed.all$y.diff <- df.windowed.all$y - df.windowed.all$y.predicted
ls.constrained.int <- lm(constrained.int.model.formula, df.windowed.all)




## return list
```

```
  return(list(ls.pre, ls.all, ls.naive.int, ls.constrained.int, ls.full.int))
}
```

```
plot.ts(series$assault_incid_c)
```

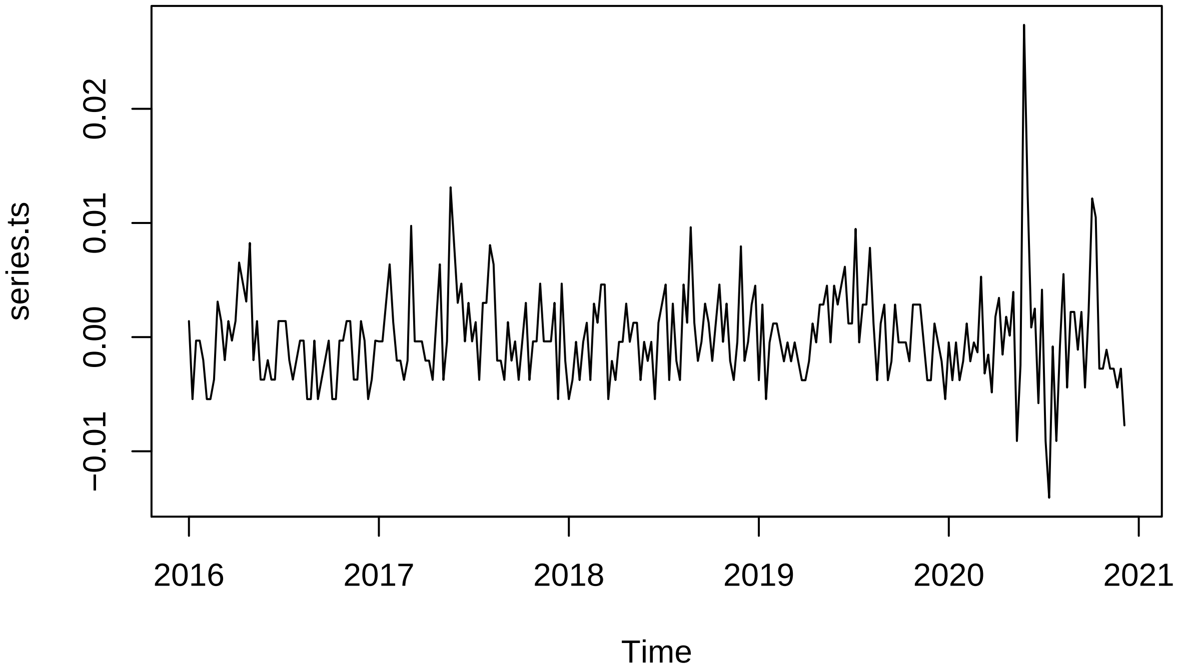


```
#post-floyd
series$post_floyd <- as.numeric(series$begin_date >= as.Date("2020-05-25"))

#three-month post-floyd
series$post_floyd_3 <- as.numeric(series$begin_date >= as.Date("2020-05-25")+months(3))

#stay at home order - covid
series$stay_at_home <- as.numeric(series$begin_date  >= as.Date("2020-03-28") &
#state of emergency - covid
series$state_of_emerg <- as.numeric(series$begin_date  >= as.Date("2020-03-13"))

ls <- lm(assault_incid_c~state_of_emerg+stay_at_home+post_floyd+post_floyd_3, data = series)

acf2(resid(ls), max.lag = 53)
```

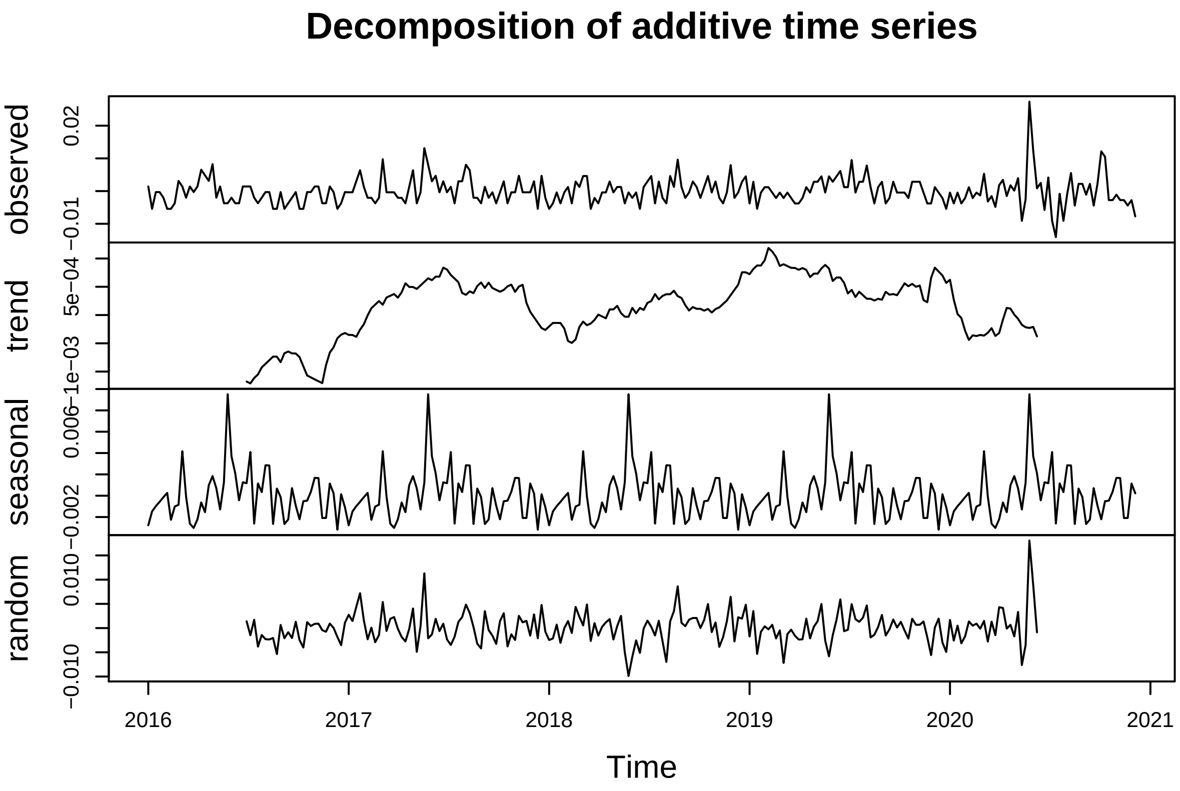


```
##      [,1]  [,2] [,3]  [,4] [,5]  [,6]  [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.23  0.02 0.08  0.02 0.03 -0.07 -0.10 0.03 -0.03 -0.04  0.04 -0.08 -0.02
## PACF 0.23 -0.03 0.09 -0.02 0.04 -0.10 -0.06 0.07 -0.05  0.00  0.05 -0.10  0.01
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.00 -0.06  0.01  0.03  0.09  0.08  0.05 -0.10 -0.06 -0.09 -0.03 -0.04
## PACF -0.01 -0.04  0.02  0.04  0.09  0.02  0.04 -0.16 -0.03 -0.07  0.03 -0.02
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF  -0.06 -0.05 -0.05  0.01  0.02  0.06 -0.03 -0.13 -0.08 -0.02  0.00 -0.08
## PACF  0.00 -0.06 -0.05  0.04  0.01  0.08 -0.06 -0.14 -0.05 -0.01  0.02 -0.09
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49]
## ACF  -0.06  0.01  0.07 -0.04  0.03  0.05  0.02  0.03  0.01  0.05  0.05  0.02
## PACF -0.02  0.02  0.06 -0.03  0.04  0.02 -0.02  0.03  0.01  0.03 -0.01 -0.03
##      [,50] [,51] [,52] [,53]
## ACF   0.04 -0.04 -0.05 -0.01
## PACF  0.02 -0.06  0.04 -0.01
```

```
series.ts <- ts(resid(ls), frequency = 53, start = c(2016,1))
plot(series.ts)
```



```
plot(decompose(series.ts))
```



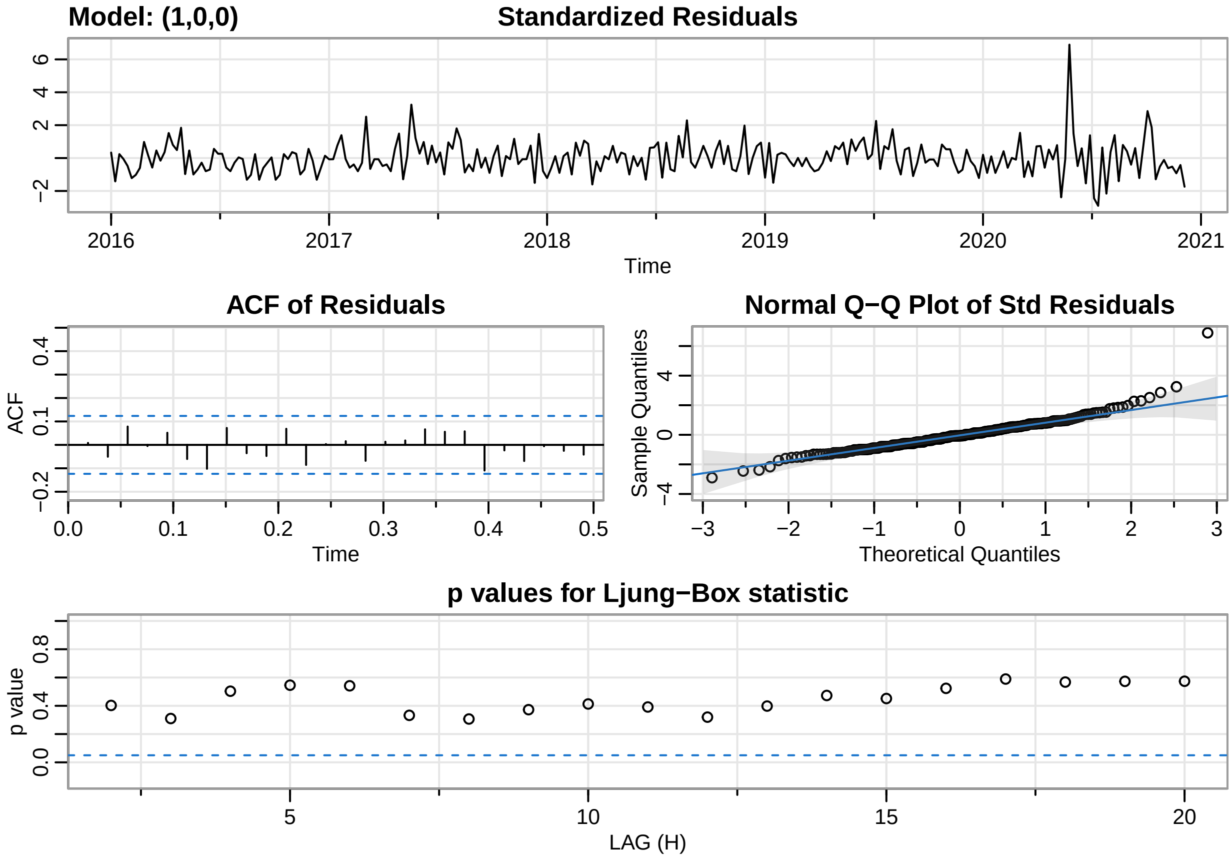
```
#testing for stationarity
adf.test(series.ts) #alt hyp is stationarity; no diff needed
```

```
## Warning in adf.test(series.ts): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  series.ts
## Dickey-Fuller = -6.2107, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
sarima(ts(series$assault_incid_c, frequency = 53, start = c(2016,1)),p=1, d=0, q=0,
                                          xreg = cbind(series$state_of_emerg,
                                                       series$stay_at_home,
                                                       series$post_floyd,
                                                       series$post_floyd_3))
```

```
## initial  value -5.474431
## iter   2 value -5.502527
## iter   3 value -5.502667
## iter   4 value -5.502760
## iter   5 value -5.502760
## iter   6 value -5.502761
## iter   6 value -5.502761
## iter   6 value -5.502761
## final  value -5.502761
## converged
## initial  value -5.504345
## iter   2 value -5.504346
## iter   3 value -5.504347
## iter   4 value -5.504347
## iter   4 value -5.504347
## iter   4 value -5.504347
## final  value -5.504347
## converged
```



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1  intercept    xreg1   xreg2   xreg3    xreg4
##       0.2364     0.0054  -0.0033  0.0044  0.0151  -0.0078
## s.e.  0.0606     0.0004   0.0026  0.0028  0.0028   0.0018
##
```

```
## sigma^2 estimated as 1.655e-05:  log likelihood = 1070.38,  aic = -2126.75
##
## $degrees_of_freedom
## [1] 256
##
## $ttable
##           Estimate     SE t.value p.value
## ar1         0.2364 0.0606  3.9015  0.0001
## intercept   0.0054 0.0004 14.9348  0.0000
## xreg1      -0.0033 0.0026 -1.2466  0.2137
## xreg2       0.0044 0.0028  1.5983  0.1112
## xreg3       0.0151 0.0028  5.4468  0.0000
## xreg4      -0.0078 0.0018 -4.2391  0.0000
##
## $AIC
## [1] -8.117382
##
## $AICc
## [1] -8.116125
##
## $BIC
## [1] -8.022045
```