

Car Jacking MPLS - Tract

Ryan Larson

2022-10-28

Spatial Data

```
#MN tracts
tracts <- get_acs(geography = "tract",
                  state = "MN",
                  variables = "B01001_001E",
                  output = "wide",
                  survey = "acs5",
                  year = 2020,
                  geometry = T)
```

```
## |
```

```
#Minneapolis Shapefile
```

```
mpls <- st_read("Data/mpls_city-shp/16cdbbfa-ad10-493c-afaf-52b61f2e76e42020329-1-180h9ap.whbo.shp") %>%
  st_transform(st_crs(tracts))
```

```
## Reading layer `16cdbbfa-ad10-493c-afaf-52b61f2e76e42020329-1-180h9ap.whbo' from data source `C:\User
## using driver `ESRI Shapefile'
## 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
```

```
mpls_tract <- tracts %>%
  st_filter(mpls, .predicate = st_intersects) %>%
  mutate(GEOID = as.numeric(GEOID),
         tract_area = as.numeric(st_area(.)),
         tract_area_sqkm = tract_area*.000001,
         tract_area_sqmi = tract_area_sqkm*.386102,
         intersection_area = as.numeric(st_area(st_intersection(., mpls))),
         perc_intersection = intersection_area/tract_area*100) %>%
  filter(perc_intersection >= 2) %>%
  select(-"B01001_001M")
```

ACS Covariates

```
acs_17 <- get_acs(
  geography = "tract",
  variables = c("B01001_001E"),
  year = 2017,
```

```

state = "MN",
county = "Hennepin",
geometry = TRUE) %>%
select(estimate)

```

Getting data from the 2013-2017 5-year ACS

Downloading feature geometry from the Census website. To cache shapefiles for use in future sessions

```
## |
```

```

acs_18 <- get_acs(
  geography = "tract",
  variables = c("B01001_001E"),
  year = 2018,
  state = "MN",
  county = "Hennepin",
  geometry = TRUE
) %>%
select(estimate)

```

Getting data from the 2014-2018 5-year ACS

Downloading feature geometry from the Census website. To cache shapefiles for use in future sessions

```
## |
```

```

acs_19 <- get_acs(
  geography = "tract",
  variables = c("B01001_001E"),
  year = 2019,
  state = "MN",
  county = "Hennepin",
  geometry = TRUE
) %>%
select(estimate)

```

Getting data from the 2015-2019 5-year ACS

Downloading feature geometry from the Census website. To cache shapefiles for use in future sessions

```
## |
```

```

acs_20 <- get_acs(
  geography = "tract",
  variables = c("B01001_001E"),
  year = 2020,
  state = "MN",
  county = "Hennepin",
  geometry = TRUE
)

```

Getting data from the 2016-2020 5-year ACS

Downloading feature geometry from the Census website. To cache shapefiles for use in future sessions

```
library(tigris)
```

To enable caching of data, set `options(tigris_use_cache = TRUE)`

in your R script or .Rprofile.

```

hennepin_blocks <- blocks(
  "MN",
  "Hennepin",
  year = 2020
)

```

```
## |
```

```

#2017 ACS interpolation
acs_1720 <- interpolate_pw(
  from = acs_17,
  to = acs_20,
  to_id = "GEOID",
  weights = hennepin_blocks,
  weight_column = "POP20",
  crs = 26993,
  extensive = TRUE) %>%
  mutate(year = 2017)

```

```

#2018 ACS interpolation
acs_1820 <- interpolate_pw(
  from = acs_18,
  to = acs_20,
  to_id = "GEOID",
  weights = hennepin_blocks,
  weight_column = "POP20",
  crs = 26993,
  extensive = TRUE) %>%
  mutate(year = 2018)

```

```

#2019 ACS interpolation
acs_1920 <- interpolate_pw(
  from = acs_19,
  to = acs_20,
  to_id = "GEOID",
  weights = hennepin_blocks,
  weight_column = "POP20",
  crs = 26993,
  extensive = TRUE) %>%
  mutate(year = 2019)

```

```

pop_denoms <- acs_20 %>%
  st_transform(crs = 26993) %>%
  mutate(year = 2020) %>%
  select(-moe, -variable, -NAME) %>%
  rbind(acs_1720, acs_1820, acs_1920) %>%
  filter(GEOID %in% mpls_tract$GEOID)

```

```
#2021+2022: LOCF
```

```

pop_denom_21 <- pop_denoms %>%
  filter(year==2020) %>%
  select(GEOID, year, estimate) %>%

```

```

mutate(year = 2021)

pop_denom_22 <- pop_denoms %>%
  filter(year==2020) %>%
  select(GEOID, year, estimate) %>%
  mutate(year = 2022)

pop_denom_locf <- pop_denoms %>%
  rbind(pop_denom_21, pop_denom_22) %>%
  rename(total_pop = estimate) %>%
  mutate(GEOID = as.numeric(GEOID)) %>%
  st_drop_geometry()

#ACS 2020 L-2 covariates
acs_2020 <- get_acs(geography = "tract",
  state = "MN",
  variables = c("B01001_001E", "B03002_003E", "B03002_004E", "B03002_005E",
    "B03002_006E", "B03002_007E", "B03002_008E", "B03002_009E",
    "B03002_010E", "B03002_011E", "B03002_012E", "B23025_002E",
    "B23025_005E", "B17001_002E", "B19057_002E", "B11003_015E",
    "B06009_002E", "B06009_005E", "C24010_001E", "C24010_003E",
    "C24010_039E", "B11001_003E", "B01001_002E", "B05001_006E",
    "B01001_003E", "B01001_004E", "B01001_005E", "B01001_006E",
    "B01001_007E", "B01001_008E", "B01001_009E", "B01001_010E",
    "B01001_011E", "B01001_012E", "B01001_013E", "B01001_014E",
    "B01001_015E", "B01001_016E", "B01001_017E", "B01001_018E",
    "B01001_019E", "B01001_020E", "B01001_021E", "B01001_022E",
    "B01001_023E", "B01001_024E", "B01001_025E", "B01001_027E",
    "B01001_028E", "B01001_029E", "B01001_030E", "B01001_031E",
    "B01001_032E", "B01001_033E", "B01001_034E", "B01001_035E",
    "B01001_036E", "B01001_037E", "B01001_038E", "B01001_039E",
    "B01001_040E", "B01001_041E", "B01001_042E", "B01001_043E",
    "B01001_044E", "B01001_045E", "B01001_046E", "B01001_047E",
    "B01001_048E", "B01001_049E", "B07001_017E", "B25003_002E",
    "B05002_013E", "B19013_001E"),
  output = "wide",
  survey = "acs5",
  year = 2020) %>%
select(-ends_with("M", ignore.case = F)) %>%
rename(total_pop = B01001_001E, white_pop = B03002_003E, black_pop = B03002_004E,
  na_pop = B03002_005E, asian_pop = B03002_006E, hpi_pop = B03002_007E,
  other_pop = B03002_008E, biracial_pop = B03002_009E, biracial_other_pop = B03002_010E,
  biracial_three_pop = B03002_011E, hisp_pop = B03002_012E, total_ilf = B23025_002E,
  unemp = B23025_005E, povlevel = B17001_002E, pub_assist = B19057_002E,
  female_hh = B11003_015E, no_hs_dip = B06009_002E, bach_degree = B06009_005E,
  total_employed = C24010_001E, employed_mbsa_male = C24010_003E,
  employed_mbsa_female = C24010_039E, mar_fam = B11001_003E, male = B01001_002E,
  noncitizen = B05001_006E,
  age_m_5_under = B01001_003E, age_m_5_9 = B01001_004E, age_m_10_14 = B01001_005E,
  age_m_15_17 = B01001_006E, age_m_18_19 = B01001_007E, age_m_20 = B01001_008E,
  age_m_21 = B01001_009E, age_m_22_24 = B01001_010E, age_m_25_29 = B01001_011E,
  age_m_30_34 = B01001_012E, age_m_35_39 = B01001_013E, age_m_40_44 = B01001_014E,
  age_m_45_49 = B01001_015E, age_m_50_54 = B01001_016E, age_m_55_59 = B01001_017E,

```

```

age_m_60_61 = B01001_018E, age_m_62_64 = B01001_019E, age_m_65_66 = B01001_020E,
age_m_67_69 = B01001_021E, age_m_70_74 = B01001_022E, age_m_75_79 = B01001_023E,
age_m_80_84 = B01001_024E, age_m_85_plus = B01001_025E, age_f_5_under = B01001_027E,
age_f_5_9 = B01001_028E, age_f_10_14 = B01001_029E, age_f_15_17 = B01001_030E,
age_f_18_19 = B01001_031E, age_f_20 = B01001_032E, age_f_21 = B01001_033E,
age_f_22_24 = B01001_034E, age_f_25_29 = B01001_035E, age_f_30_34 = B01001_036E,
age_f_35_39 = B01001_037E, age_f_40_44 = B01001_038E, age_f_45_49 = B01001_039E,
age_f_50_54 = B01001_040E, age_f_55_59 = B01001_041E, age_f_60_61 = B01001_042E,
age_f_62_64 = B01001_043E, age_f_65_66 = B01001_044E, age_f_67_69 = B01001_045E,
age_f_70_74 = B01001_046E, age_f_75_79 = B01001_047E, age_f_80_84 = B01001_048E,
age_f_85_plus = B01001_049E, res_mob = B07001_017E,
own_hh = B25003_002E, foreign = B05002_013E,
med_hh_inc = B19013_001E) %>%
mutate(white_prop = white_pop/total_pop,
black_prop = black_pop/total_pop,
na_prop = na_pop/total_pop,
asian_prop = asian_pop/total_pop,
hpi_prop = hpi_pop/total_pop,
other_prop = other_pop/total_pop,
biracial_prop = (biracial_pop+biracial_other_pop+biracial_three_pop)/total_pop,
hisp_prop = hisp_pop/total_pop,
white_perc = 100*white_pop/total_pop,
black_perc = 100*black_pop/total_pop,
na_perc = 100*na_pop/total_pop,
asian_perc = 100*asian_pop/total_pop,
hpi_perc = 100*hpi_pop/total_pop,
other_perc = 100*other_pop/total_pop,
biracial_perc = 100*(biracial_pop+biracial_other_pop+biracial_three_pop)/total_pop,
hisp_perc = 100*hisp_pop/total_pop,
unemp_rate = 100*unemp/total_ilf,
pov_rate = 100*povlevel/total_pop,
pub_assist_rate = 100*pub_assist/total_pop,
female_hh_rate = 100*female_hh/total_pop,
no_hs_dip_rate = 100*no_hs_dip/total_pop,
bach_degree_rate = 100*bach_degree/total_pop,
employed_mbsa = employed_mbsa_male+employed_mbsa_female,
employed_mbsa_rate = 100*employed_mbsa/total_employed,
mar_fam_rate = 100*mar_fam/total_pop,
male_rate = 100*male/total_pop,
noncitizen_rate = 100*noncitizen/total_pop,
race_eth_hetero = 1-(white_prop^2+black_prop^2+na_prop^2+asian_prop^2+
hpi_prop^2+other_prop^2+other_prop^2+biracial_prop^2+hisp_prop^2),
age_below_18_perc = 100*(age_m_5_under+age_f_5_under+age_m_5_9+
age_f_5_9+age_m_10_14+age_f_10_14+age_m_15_17+
age_f_15_17)/total_pop,
age_19_29_perc = 100*(age_m_18_19+age_f_18_19+age_m_20+age_f_20+age_m_21+age_f_21+
age_m_22_24+age_f_22_24+age_m_25_29+age_f_25_29)/total_pop,
age_30_49_perc = 100*(age_m_30_34+age_f_30_34+age_m_35_39+age_f_35_39+
age_m_40_44+age_f_40_44+age_m_45_49+age_f_45_49)/total_pop,
age_50_69_perc = 100*(age_m_50_54+age_f_50_54+age_m_55_59+age_f_55_59+
age_m_60_61+age_f_60_61+age_m_62_64+age_f_62_64+
age_m_65_66+age_f_65_66+age_m_67_69+age_f_67_69)/total_pop,
age_70_plus_perc = 100*(age_m_70_74+age_f_70_74+age_m_75_79+age_f_75_79+

```

```

                                age_m_80_84+age_f_80_84+age_m_85_plus+age_f_85_plus)/total_pop,
    res_mob_rate = 100-100*res_mob/total_pop,
    own_hh_rate = 100*own_hh/total_pop,
    foreign_rate = 100*foreign/total_pop)

```

Getting data from the 2016-2020 5-year ACS

```

# 2020 Census 18+ Denominator
dc2020 <- get_decennial(
  geography = "tract",
  variables = c("P3_001N"),
  year = 2020,
  state = "MN",
  county = "Hennepin",
  geometry = F) %>%
mutate(GEOID = is.numeric(GEOID)) %>%
rename(total_pop = value) %>%
select(-GEOID)

```

Getting data from the 2020 decennial Census

Using the PL 94-171 Redistricting Data summary file

Note: 2020 decennial Census data use differential privacy, a technique that
introduces errors into data to preserve respondent confidentiality.

i Small counts should be interpreted with caution.

i See <https://www.census.gov/library/fact-sheets/2021/protecting-the-confidentiality-of-the-2020-cen>

This message is displayed once per session.

Open Minneapolis Carjacking Data

```

#open minneapolis crime data 2019-4/7 (date of download)
cj_spatial <- read_csv("Car Jacking/crime_data.csv") %>%
  filter(Offense=="Carjacking - Subset of Robbery") %>% #filter carjackings
  mutate(date=ymd_hms(Occurred_Date),
         year=isoyear(date),
         week=isoweek(date)) %>%
  select(OBJECTID, year, week, Latitude, Longitude) %>%
  st_as_sf(coords = c("Longitude", "Latitude"), crs = "NAD83", remove=F) %>%
  st_join(mpls_tract) %>% #spatial join neighborhoods
  st_drop_geometry() %>%
  filter(!is.na(GEOID)) %>%
  group_by(year, week, GEOID, .drop=F) %>%
  tally(name = "car_jack") %>%
  ungroup() %>%
  complete(year, week, GEOID=mpls_tract$GEOID, fill = list(car_jack = 0)) %>%
  filter(!(year==2022 & week >= 14) & #removing unobserved/redundant completions
         !(year==2021 & week==53) & #removing unobserved/redundant completions
         !(year==2020 & week < 36)) %>% #removing weeks before Sept. 2020, isoweek 36
  arrange(GEOID, year, week) %>%
  left_join(mpls_tract, by = "GEOID") %>%
  left_join(pop_denom_locf, by = c("GEOID", "year")) %>%
  mutate(car_jack_rate = car_jack/total_pop*1000) %>%
  st_as_sf()

```

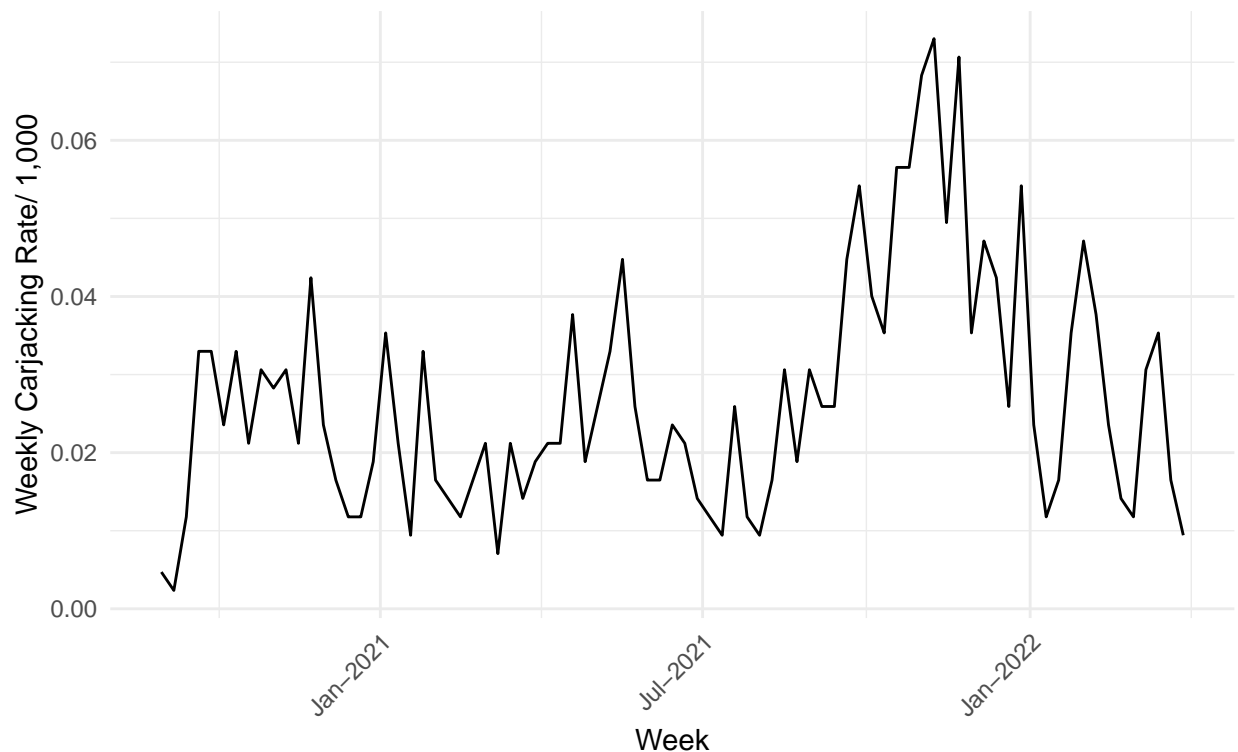
MPLS Carjackings by Week

```
#aggregate to week over tracts
cj_week <- cj_spatial %>%
  group_by(year, week) %>%
  summarize(car_jack = sum(car_jack, na.rm = T),
            total_pop = sum(total_pop, na.rm = T)) %>%
  mutate(begin_date = ISOweek2date(paste(year, paste0("W", sprintf("%02d", week)), 1, sep = "-")),
         end_date = begin_date+weeks(1)-days(1),
         car_jack_rate = car_jack/total_pop*1000)

## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.

ggplot(cj_week)+
  geom_line(aes(x=begin_date, y=car_jack_rate))+
  scale_x_date(date_labels = "%b-%Y",
              limits = c(min(cj_week$begin_date), max(cj_week$begin_date)))+
  labs(title = "Figure X: Weekly Minneapolis Carjackings, 8/31/2020-4/7/2022",
       subtitle = "openminneapolis Crime Data",
       x = "Week",
       y = "Weekly Carjacking Rate/ 1,000")+
  theme_minimal()+
  theme(axis.text.x=element_text(angle=45, hjust=1))
```

Figure X: Weekly Minneapolis Carjackings, 8/31/2020–4/7/2022
openminneapolis Crime Data



MPLS ZCTA Carjackings Map

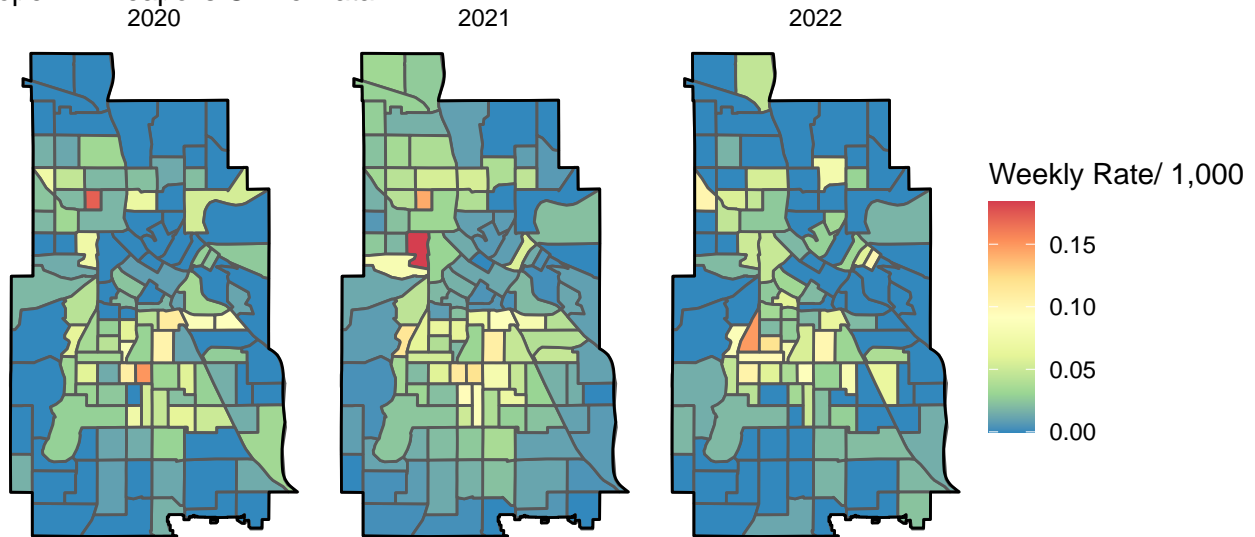
```
#aggregate to neighborhood-year level
cj_tract_year <- cj_spatial %>%
  group_by(GEOID, year) %>%
  summarize(car_jack = sum(car_jack, na.rm = T),
            total_pop = sum(B01001_001E, na.rm = T),
            car_jack_rate = car_jack/total_pop*1000)

## `summarise()` has grouped output by 'GEOID'. You can override using the
## `.groups` argument.

ggplot() +
  geom_sf(data = cj_tract_year, aes(geometry = geometry, fill = car_jack_rate)) +
  geom_sf(data = mpls, aes(geometry = geometry), color = "black", alpha = 0)+
  facet_wrap(~year)+
  scale_fill_distiller(palette = "Spectral")+
  labs(title = "Figure X: Minneapolis Carjacking Rates by Tract and Year",
       subtitle = "openminneapolis Crime Data",
       fill = "Weekly Rate/ 1,000")+
  theme_void()
```

Figure X: Minneapolis Carjacking Rates by Tract and Year

openminneapolis Crime Data



Expanded MPLS Carjacking (Crime Incidents) Data


```

cj_exp <- read_csv("Data/MPDdata_082422.csv") %>%
  mutate(date=mdy_hm(reporteddate),
         year=isoyear(date),
         week=isoweek(date)) %>%
  select(CaseNumber, year, week, latitude, longitude) %>%
  distinct(CaseNumber, .keep_all = TRUE) %>%
  drop_na(latitude, longitude) %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = "NAD83", remove=F) %>%
  st_join(mpls_tract) %>% #spatial join neighborhoods
  st_drop_geometry() %>%
  drop_na(GEOID) %>%
  group_by(year, week, GEOID, .drop=F) %>%
  tally(name = "car_jack") %>%
  ungroup() %>%
  complete(year, week, GEOID=mpls_tract$GEOID, fill = list(car_jack = 0)) %>%
  filter(!(year==2021 & week==53)) %>%
  arrange(GEOID, year, week) %>%
  left_join(mpls_tract, by = "GEOID") %>%
  left_join(dc2020, by = c("NAME")) %>%
  mutate(car_jack_rate = car_jack/total_pop*1000) %>%
  st_as_sf()

## Rows: 3894 Columns: 28
## -- Column specification -----
## Delimiter: ","
## chr (24): CaseNumber, dataset, closurecode, closurecode_MPD, reporteddate, c...
## dbl (4): precinct, latitude, longitude, age
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```

MPLS Carjackings by Week - MPD Extended Data

```

#aggregate to week over tracts
cj_exp_week <- cj_exp %>%
  group_by(year, week) %>%
  summarize(car_jack = sum(car_jack, na.rm = T),
            total_pop = sum(total_pop, na.rm = T)) %>%
  mutate(begin_date = ISOweek2date(paste(year, paste0("W", sprintf("%02d", week)), 1, sep = "-")),
         end_date = begin_date+weeks(1)-days(1),
         car_jack_rate = car_jack/total_pop*1000,
         pre_post_floyd = ifelse(end_date <= as.Date("2020-05-25"), 0, 1)) %>%
  filter(end_date <= as.Date("2022-08-20")) %>%
  mutate(c sma = forecast::ma(car_jack_rate, order=5, centre=TRUE),
         tsma = TTR::SMA(car_jack_rate, n=5))

## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
## Registered S3 method overwritten by 'quantmod': method from as.zoo.data.frame
## zoo

pre_mean <- mean(cj_exp_week$car_jack_rate[cj_exp_week$pre_post_floyd==0], na.rm = T)
post_mean <- mean(cj_exp_week$car_jack_rate[cj_exp_week$pre_post_floyd==1], na.rm = T)

```

```

c(pre_mean, post_mean)

## [1] 0.004835884 0.030361407

post_mean/pre_mean

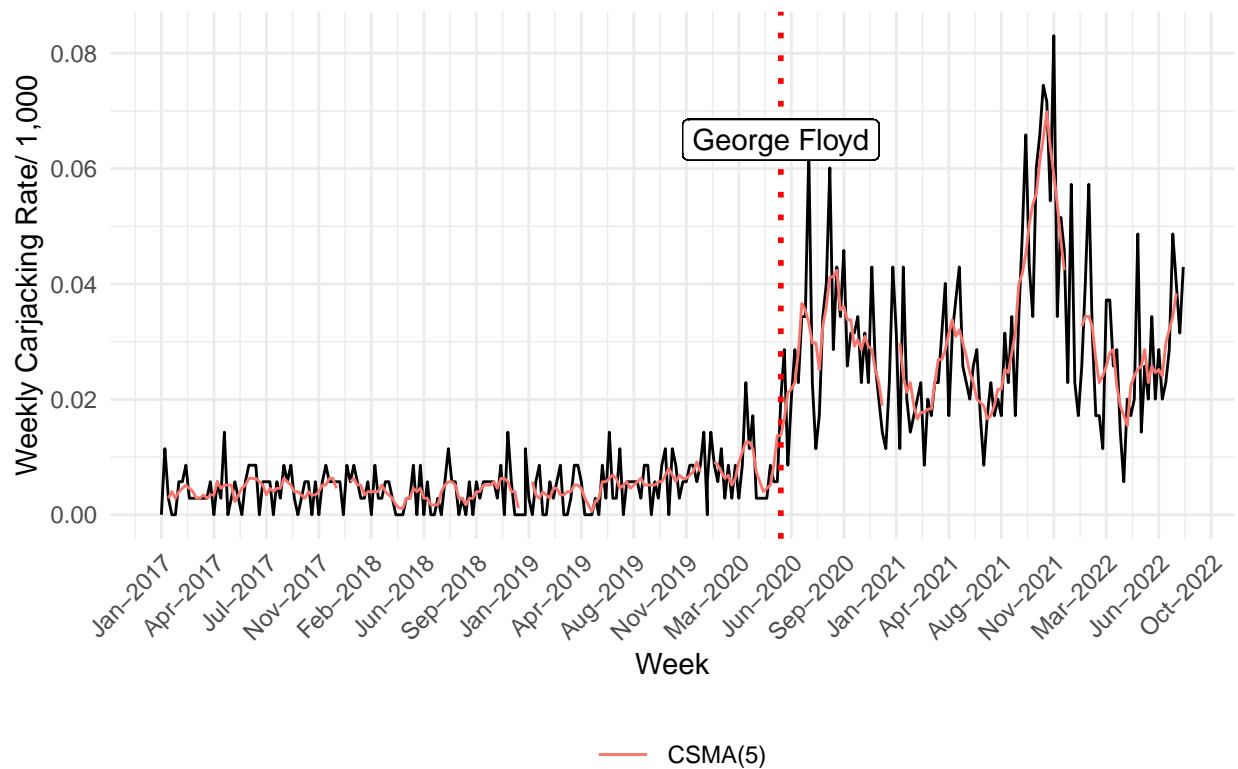
## [1] 6.278358

ggplot(cj_exp_week)+
  geom_line(aes(x=begin_date, y=car_jack_rate))+
  scale_x_date(date_labels = "%b-%Y", date_breaks = "15 weeks",
               limits = c(min(cj_exp_week$begin_date), max(cj_exp_week$begin_date)))+
  geom_vline(xintercept=cj_exp_week$begin_date[cj_exp_week$year==2020 &
                                                cj_exp_week$week==isoweek(date("2020-05-25"))],
             linetype="dotted", color="red", size=1)+
  geom_label(aes(x=cj_exp_week$begin_date[cj_exp_week$year==2020 &
                                                cj_exp_week$week==isoweek(date("2020-05-25"))],
                 y=0.065),
             label = "George Floyd", show.legend = FALSE)+
  labs(title = "Figure 1: Weekly Minneapolis Carjackings, 1/1/2017-8/20/2022",
       x = "Week",
       y = "Weekly Carjacking Rate/ 1,000",
       color = NULL)+
  geom_line(aes(x=begin_date, y=csma, color = "CSMA(5)"))+
  theme_minimal()+
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  theme(legend.key.size = unit(0.8, "cm"), legend.position = "bottom")

## Warning: Removed 4 row(s) containing missing values (geom_path).

```

Figure 1: Weekly Minneapolis Carjackings, 1/1/2017–8/20/2022



```
ggsave(filename = "Car Jacking/Figures for PAA/fig1.png", bg="white", width = 10, height = 8)
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```

Structural Change in Carjacking

```
library(strucchange)
```

```
## Loading required package: sandwich
```

```
##
```

```
## Attaching package: 'strucchange'
```

```
## The following object is masked from 'package:stringr':
```

```
##
```

```
## boundary
```

```
cj_exp_week <- cj_exp_week %>%
```

```
  ungroup() %>%
```

```
  mutate(t = row_number())
```

```
sctest(cj_exp_week$car_jack_rate~cj_exp_week$t,
```

```
  type = "Chow",
```

```
  point = 180)
```

```
##
```

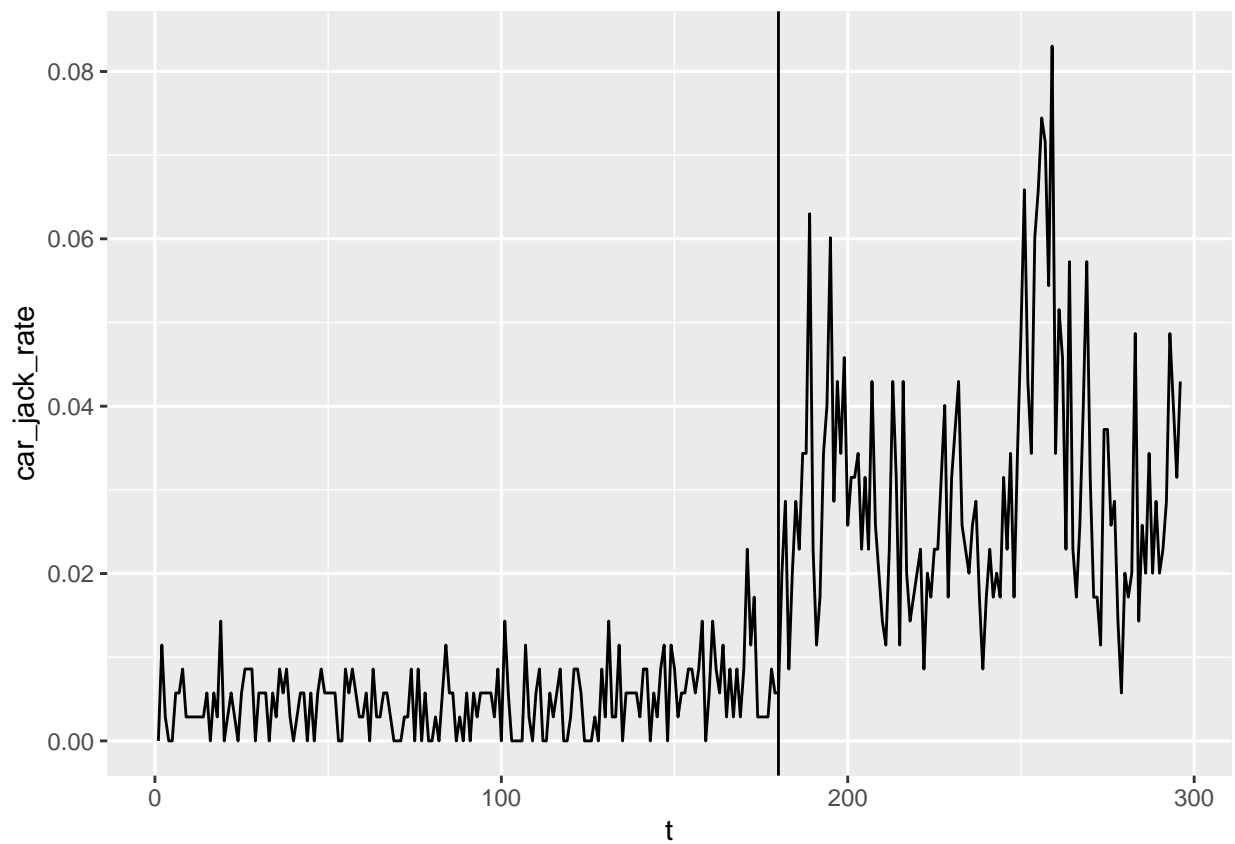
```
## Chow test
```

```
##
## data:  cj_exp_week$car_jack_rate ~ cj_exp_week$t
## F = 49.567, p-value < 2.2e-16

breakpoints(car_jack_rate~t,
            data = cj_exp_week,
            breaks = 1)

##
## Optimal 2-segment partition:
##
## Call:
## breakpoints.formula(formula = car_jack_rate ~ t, breaks = 1,
## data = cj_exp_week)
##
## Breakpoints at observation number:
## 180
##
## Corresponding to breakdates:
## 0.6081081

ggplot(cj_exp_week)+
  geom_line(aes(x=t, y=car_jack_rate))+
  geom_vline(aes(xintercept = 180))
```



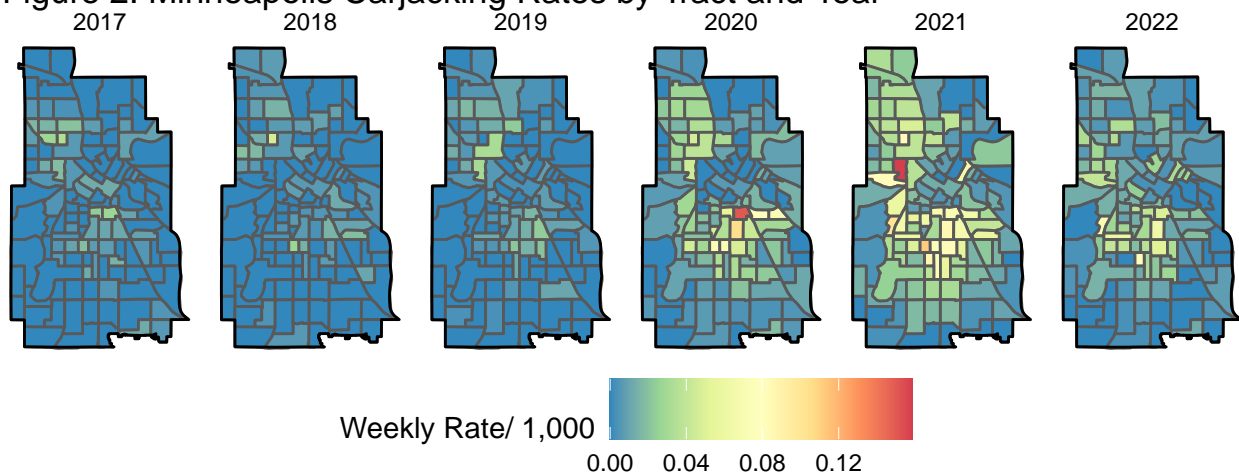
MPLS ZCTA Carjackings Map - MPD Extended Data

```
#aggregate to neighborhood-year level
cj_exp_tract_year <- cj_exp %>%
  group_by(GEOID, year) %>%
  summarize(car_jack = sum(car_jack, na.rm = T),
            total_pop = sum(B01001_001E, na.rm = T),
            car_jack_rate = car_jack/total_pop*1000) %>%
  mutate(GEOID = as.character(GEOID))

## `summarise()` has grouped output by 'GEOID'. You can override using the
## `.groups` argument.

ggplot() +
  geom_sf(data = cj_exp_tract_year, aes(geometry = geometry, fill = car_jack_rate)) +
  geom_sf(data = mpls, aes(geometry = geometry), color = "black", alpha = 0)+
  facet_grid(~year)+
  scale_fill_distiller(palette = "Spectral")+
  labs(title = "Figure 2: Minneapolis Carjacking Rates by Tract and Year",
       fill = "Weekly Rate/ 1,000")+
  theme_void()+
  theme(legend.key.size = unit(0.8, "cm"), legend.position = "bottom")
```

Figure 2: Minneapolis Carjacking Rates by Tract and Year



```
ggsave(filename = "Car Jacking/Figures for PAA/fig2.png", bg="white", width = 10, height = 8)
```

MPLS Murder (Crime Incidents) Data

```
#pre-pims
mpd_2016 <- read_csv("Data/Police_Incidents_2016.csv")

## Rows: 20155 Columns: 20
## -- Column specification -----
## Delimiter: ","
## chr (12): PublicAddress, CCN, Precinct, ReportedDate, BeginDate, Offense, D...
## dbl (7): FID, ControlNbr, GBSID, Lat, Long, X, Y
## time (1): Time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

mpd_2017 <- read_csv("Data/Police_Incidents_2017.csv")

## Rows: 22085 Columns: 20
## -- Column specification -----
## Delimiter: ","
## chr (12): PublicAddress, CCN, Precinct, ReportedDate, BeginDate, Offense, D...
## dbl (7): FID, ControlNbr, GBSID, Lat, Long, X, Y
## time (1): Time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

mpd_2018a <- read_csv("Data/Police_Incidents_2018.csv")

## Rows: 7350 Columns: 20
## -- Column specification -----
## Delimiter: ","
## chr (12): PublicAddress, CCN, Precinct, ReportedDate, BeginDate, Offense, D...
## dbl (7): FID, ControlNbr, GBSID, Lat, Long, X, Y
## time (1): Time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

#pims
mpd_2018b <- read_csv("Data/Police_Incidents_2018_PIMS.csv")

## Rows: 11603 Columns: 23
## -- Column specification -----
## Delimiter: ","
## chr (13): publicaddress, caseNumber, precinct, reportedDate, beginDate, repo...
## dbl (10): X, Y, reportedTime, beginTime, centergbssid, centerLong, centerLat,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

mpd_2019 <- read_csv("Data/Police_Incidents_2019.csv")

## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```

## Rows: 22934 Columns: 23
## -- Column specification -----
## Delimiter: ","
## chr (12): publicaddress, caseNumber, reportedDate, beginDate, reportedDateTi...
## dbl (11): X, Y, precinct, reportedTime, beginTime, centergbid, centerLong, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
mpd_2020 <- read_csv("Data/Police_Incidents_2020.csv")

## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 24136 Columns: 23
## -- Column specification -----
## Delimiter: ","
## chr (12): publicaddress, caseNumber, reportedDate, beginDate, reportedDateTi...
## dbl (11): X, Y, precinct, reportedTime, beginTime, centergbid, centerLong, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
mpd_2021 <- read_csv("Data/Police_Incidents_2021.csv")

## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 24755 Columns: 23
## -- Column specification -----
## Delimiter: ","
## chr (12): publicaddress, caseNumber, reportedDate, beginDate, reportedDateTi...
## dbl (11): X, Y, precinct, reportedTime, beginTime, centergbid, centerLong, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
mpd_2022 <- read_csv("Data/Police_Incidents_2022.csv")

## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 19555 Columns: 23
## -- Column specification -----
## Delimiter: ","
## chr (12): publicaddress, caseNumber, reportedDate, beginDate, reportedDateTi...
## dbl (11): X, Y, precinct, reportedTime, beginTime, centergbid, centerLong, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```

```

pre_pims_base <- mpd_2016 %>%
  rbind(mpd_2017) %>%
  rbind(mpd_2018a) %>%
  rename(reportedDate = ReportedDate,
         centerLong = Long,
         centerLat = Lat) %>%
  select(FID, centerLong, centerLat, Offense, reportedDate) %>%
  rename(OBJECTID = FID,
         X = centerLong,
         Y = centerLat,
         offense = Offense)

post_pims_base <- mpd_2018b %>%
  rbind(mpd_2019) %>%
  rbind(mpd_2020) %>%
  rbind(mpd_2021) %>%
  rbind(mpd_2022) %>%
  select(OBJECTID, X, Y, offense, reportedDate)

mpd <- pre_pims_base %>%
  rbind(post_pims_base)

#aggregate homicides to tract-week
homicide <- mpd %>%
  mutate(date=ymd_hms(reportedDate),
         year=isoyear(date),
         week=isoweek(date)) %>%
  filter(offense=="MURDR" & year!=2016 & year!=2015) %>% #filter homicides
  select(OBJECTID, year, week, Y, X) %>%
  st_as_sf(coords = c("X", "Y"), crs = "NAD83", remove=F) %>%
  st_join(mpls_tract) %>% #spatial join neighborhoods
  st_drop_geometry() %>%
  filter(!is.na(GEOID)) %>%
  group_by(year, week, GEOID, .drop=F) %>%
  tally(name = "homicide") %>%
  ungroup() %>%
  complete(year, week, GEOID=mpls_tract$GEOID, fill = list(homicide = 0)) %>%
  filter(!(year==2021 & week==53)) %>%
  arrange(GEOID, year, week) %>%
  left_join(mpls_tract, by = "GEOID") %>%
  left_join(dc2020, by = c("NAME")) %>%
  mutate(homicide_rate = homicide/total_pop*1000) %>%
  st_as_sf()

```

MPLS Murder by Week

```

#aggregate to week over tracts
homicide_week <- homicide %>%
  group_by(year, week) %>%
  summarize(homicide = sum(homicide, na.rm = T),
            total_pop = sum(total_pop, na.rm = T)) %>%
  mutate(begin_date = ISOweek2date(paste(year, paste0("W", sprintf("%02d", week)), 1, sep = "-")),

```



```

    end_date = begin_date+weeks(1)-days(1),
    homicide_rate = homicide/total_pop*1000,
    pre_post_floyd = ifelse(end_date <= as.Date("2020-05-25"), 0, 1)) %>%
filter(end_date <= as.Date("2022-08-20")) %>%
mutate(csma = forecast::ma(homicide_rate, order=5,centre=TRUE),
       tsma = TTR::SMA(homicide_rate, n=5))

## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.

pre_mean <- mean(homicide_week$homicide_rate[homicide_week$pre_post_floyd==0], na.rm = T)
post_mean <- mean(homicide_week$homicide_rate[homicide_week$pre_post_floyd==1], na.rm = T)

c(pre_mean, post_mean)

## [1] 0.001749826 0.005331759

post_mean/pre_mean

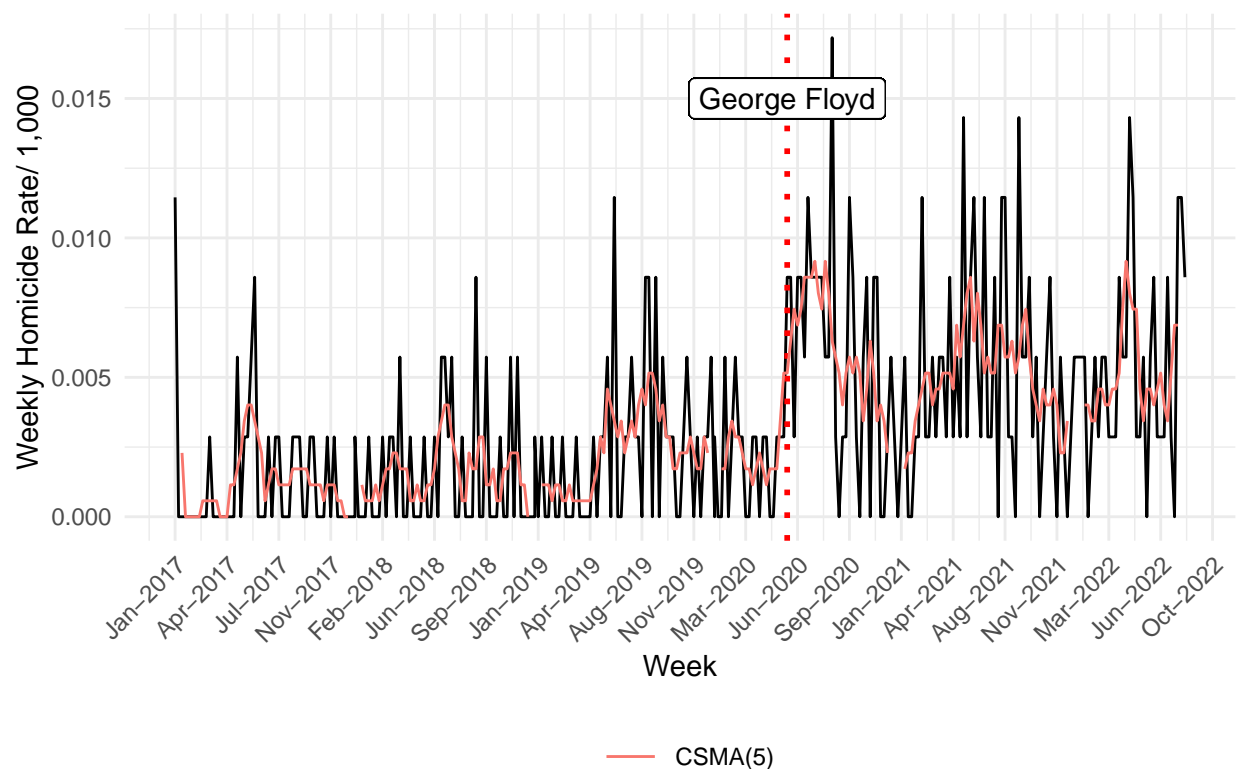
## [1] 3.047022

ggplot(homicide_week)+
  geom_line(aes(x=begin_date, y=homicide_rate))+
  scale_x_date(date_labels = "%b-%Y", date_breaks = "15 weeks",
               limits = c(min(homicide_week$begin_date), max(homicide_week$begin_date)))+
  labs(title = "Figure 3: Weekly Minneapolis Homicide, 1/1/2017-8/20/2022",
       x = "Week",
       y = "Weekly Homicide Rate/ 1,000",
       color = NULL)+
  geom_vline(xintercept=homicide_week$begin_date[homicide_week$year==2020 &
                                                  homicide_week$week==isoweek(date("2020-05-25"))],
             linetype="dotted", color="red", size=1)+
  geom_label(aes(x=homicide_week$begin_date[homicide_week$year==2020 &
                                                  homicide_week$week==isoweek(date("2020-05-25"))],
                 y=0.015),
             label = "George Floyd", show.legend = FALSE)+
  theme_minimal()+
  geom_line(aes(x=begin_date, y=csma, color = "CSMA(5)"))+
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  theme(legend.key.size = unit(0.8, "cm"),legend.position = "bottom")

## Warning: Removed 4 row(s) containing missing values (geom_path).

```

Figure 3: Weekly Minneapolis Homicide, 1/1/2017–8/20/2022



```
ggsave(filename = "Car Jacking/Figures for PAA/fig3.png", bg="white", width = 10, height = 8)
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```

Structural Change in Carjacking

```
homicide_week <- homicide_week %>%
  ungroup() %>%
  mutate(t = row_number())

sctest(homicide_week$homicide_rate~homicide_week$t,
  type = "Chow",
  point = 180)

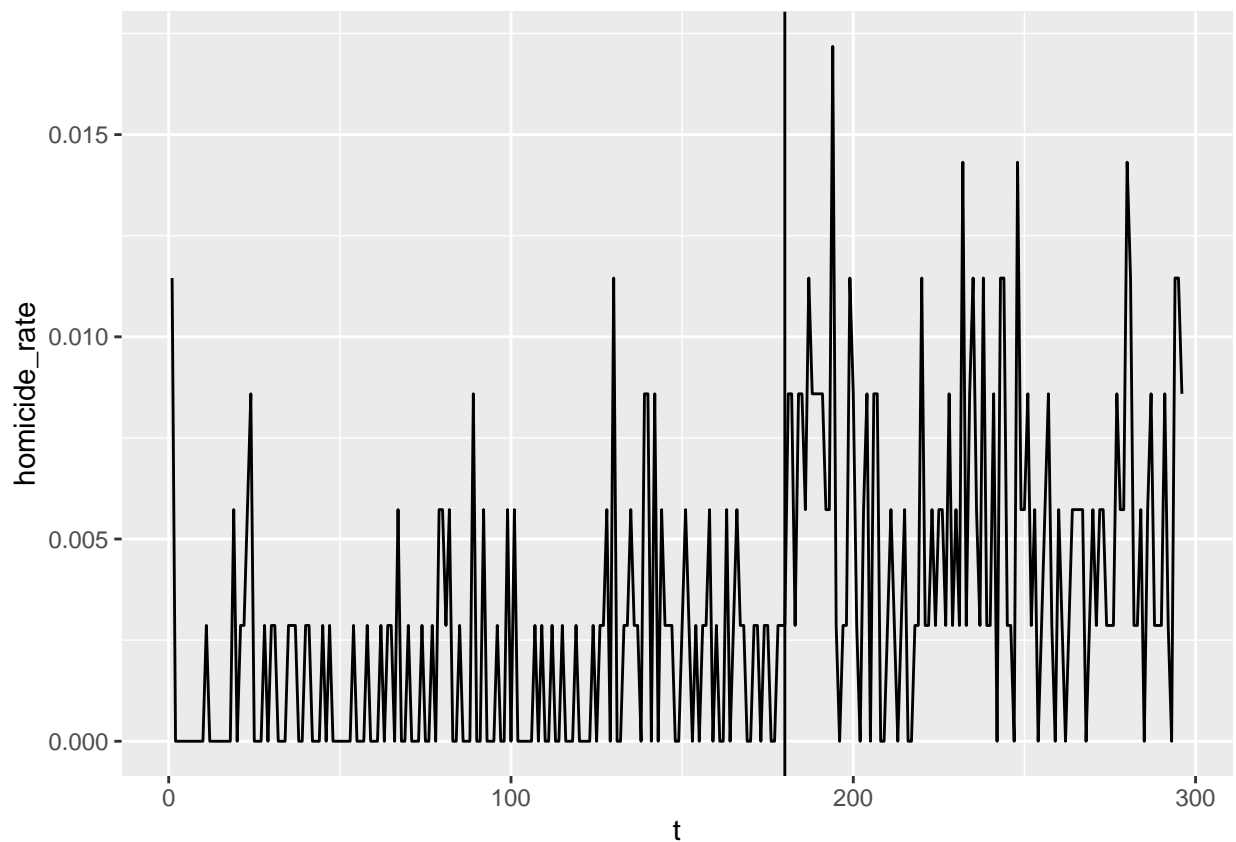
##
## Chow test
##
## data: homicide_week$homicide_rate ~ homicide_week$t
## F = 10.643, p-value = 3.454e-05

breakpoints(homicide_rate~t,
  data = homicide_week,
  breaks = 1)

##
## Optimal 2-segment partition:
```

```
##
## Call:
## breakpoints.formula(formula = homicide_rate ~ t, breaks = 1,
##   data = homicide_week)
##
## Breakpoints at observation number:
## 180
##
## Corresponding to breakdates:
## 0.6081081
```

```
ggplot(homicide_week)+
  geom_line(aes(x=t, y=homicide_rate))+
  geom_vline(aes(xintercept = 180))
```



MPLS ZCTA Murder Map - MPD Extended Data

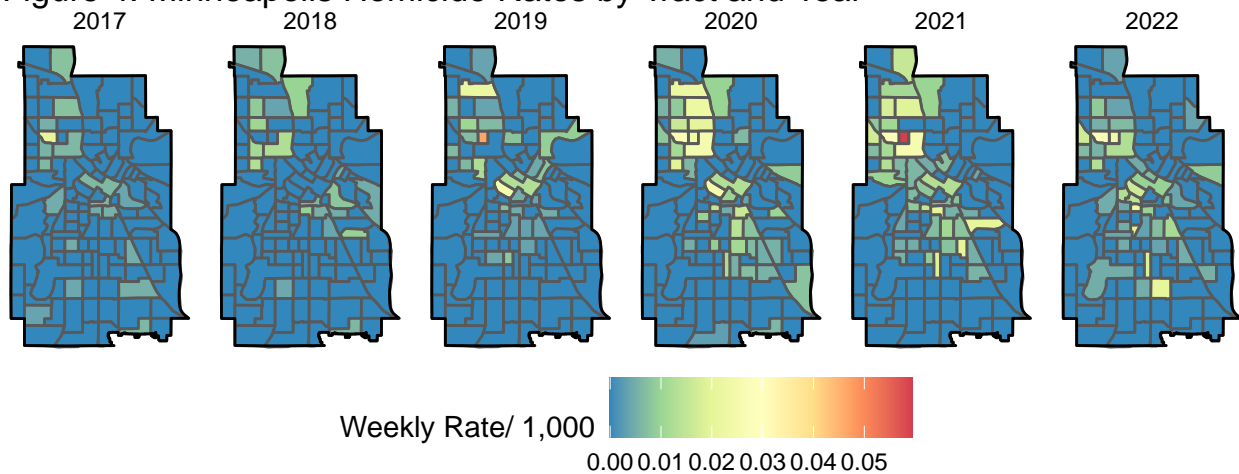
```
#aggregate to neighborhood-year level
homicide_tract_year <- homicide %>%
  group_by(GEOID, year) %>%
  summarize(homicide = sum(homicide, na.rm = T),
            total_pop = sum(B01001_001E, na.rm = T),
            homicide_rate = homicide/total_pop*1000) %>%
  mutate(GEOID = as.character(GEOID))
```

```
## `summarise()` has grouped output by 'GEOID'. You can override using the
```

```
## `.groups` argument.
```

```
ggplot() +
  geom_sf(data = homicide_tract_year, aes(geometry = geometry, fill = homicide_rate)) +
  geom_sf(data = mpls, aes(geometry = geometry), color = "black", alpha = 0)+
  facet_grid(~year)+
  scale_fill_distiller(palette = "Spectral")+
  labs(title = "Figure 4: Minneapolis Homicide Rates by Tract and Year",
       fill = "Weekly Rate/ 1,000")+
  theme_void() +
  theme(legend.key.size = unit(0.8, "cm"), legend.position = "bottom")
```

Figure 4: Minneapolis Homicide Rates by Tract and Year



```
ggsave(filename = "Car Jacking/Figures for PAA/fig4.png", bg="white", width = 10, height = 8)
```

Dispersion of Change from 2017-2019 to 2020-2021

Car Jacking

```
crimedisersion <- function
(data1, unitID, time1, time2, method = "match") {

  # define variables to limit build warnings
  adjusted <- Ut1 <- Ut2 <- Rt1 <- Rt2 <- chg <- pct <- NULL

  # ERROR CHECKING. Has user passed a data frame?
  if (!is.data.frame(data1)) {
```

```

    stop("The input data specified is not a data.frame object. Please fix.")
  }

  # Build a local data.frame and populate with passed arguments
  source_rows <- nrow(data1)
  df1 <- data.frame(matrix(ncol = 3, nrow = source_rows))
  colnames(df1) <- c("unit", "time1", "time2")
  df1$unit <- data1[, unitID]
  df1$time1 <- data1[, time1]
  df1$time2 <- data1[, time2]
  if (method == "remove") {
    analysisMethod <- "remove"
  } else {
    analysisMethod <- "match"
  }
}

# ERROR CHECKING. Did user pass numeric columns where needed?
try (df1$time1 <- as.numeric(df1$time1), silent = TRUE)
try (df1$time2 <- as.numeric(df1$time2), silent = TRUE)

if (!class(df1$time1)[1] == "numeric") {
  stop("The time1 field is not a numeric object. Please fix.")
}
if (!class(df1$time2)[1] == "numeric") {
  stop("The time2 field is not a numeric object. Please fix.")
}

# MORE ERROR CHECKING:
# What if the user has NA or missing data?
# What if the crime problem is decreasing?
# Fun tasks for later...

# Set up parameters -----

# Set up initial parameters
count_Rt1 <- sum(df1$time1)
count_Rt2 <- sum(df1$time2)
chg_Rt1_Rt2 <- count_Rt2 - count_Rt1
pct_Rt1_Rt2 <- (chg_Rt1_Rt2 / count_Rt1) *100

# Add the field that has the volume of change, and order by it
df1 <- df1 %>%
  mutate (diff = time2 - time1) %>%
  mutate (diffPct = 100*(diff/time1)) %>%
  arrange(desc(diff))

# Grab some basic statistics here
numPositive <- length(which(df1$diff > 0))
numNeutral <- length(which(df1$diff == 0))
numNegative <- length(which(df1$diff < 0))

```

```

# Create the new data frame to hold the result
df2 <- data.frame(matrix(ncol = 8, nrow = 0))
colnames(df2) <- c("unit", "adjusted", "Ut1", "Ut2", "Rt1", "Rt2", "chg", "pct")
df2 <- df2 %>%
  mutate(unit = as.character(unit)) %>%
  mutate(adjusted = as.numeric(adjusted)) %>%
  mutate(Ut1 = as.numeric(Ut1)) %>%
  mutate(Ut2 = as.numeric(Ut2)) %>%
  mutate(Rt1 = as.numeric(Rt1)) %>%
  mutate(Rt2 = as.numeric(Rt2)) %>%
  mutate(chg = as.numeric(chg)) %>%
  mutate(pct = as.numeric(pct))

# set up the initial row in the result data frame
df2 <- df2 %>% add_row(unit = "[ ALL AREAS ]", adjusted = 0,
                      Ut1 = 0, Ut2 = 0,
                      Rt1 = count_Rt1, Rt2 = count_Rt2,
                      chg = chg_Rt1_Rt2, pct = pct_Rt1_Rt2)

gain_from_row_removal <- row_to_remove <- NULL

# Loop through each row of the data
for (master_loop in 1:(source_rows)){

  df1 <- df1 %>% # order the data frame
    arrange(desc(diff))

  if (analysisMethod == "match"){
    ### 'Zero change the row' approach
    count_Rt1_temp <- count_Rt1
    count_Rt2_temp <- count_Rt2 - df1$diff[master_loop]
    pct_Rt1_Rt2 <- ((count_Rt1_temp - count_Rt2_temp) / count_Rt1) * 100
  }
  else { #analysisMethod == "remove"
    ### 'Remove entire row' approach, including remove t1 value
    count_Rt1_temp <- count_Rt1 - df1$time1[master_loop]
    count_Rt2_temp <- count_Rt2 - df1$time2[master_loop]
    pct_Rt1_Rt2 <- ((count_Rt1_temp - count_Rt2_temp) / count_Rt1) * 100
  }

  row_to_remove <- 1 # Always row 1, but this is a legacy from
  # when I used a different approach...
  # Here, the row we are removing is
  # stored in row_to_remove

  if (analysisMethod == "remove"){
    ### Remove entire row approach
    # This approach removes the impact of the area by subtracting
    # both Rt1 and Rt2
    count_Rt1 <- count_Rt1 - df1$time1[row_to_remove]
    count_Rt2 <- count_Rt2 - df1$time2[row_to_remove]
    chg_Rt1_Rt2 <- count_Rt2 - count_Rt1
  }
}

```

```

pct_Rt1_Rt2 <- (chg_Rt1_Rt2 / count_Rt1) *100
named_areas <- df1$unit[row_to_remove]
}

if (analysisMethod == "match"){
  ##### Zero change the row approach, as if Rt2 == Rt1 in the row
  #   The best row to remove is has been exhaustively calculated
  #   Here, the row we are removing is stored in row_to_remove
  count_Rt1 <- count_Rt1
  count_Rt2 <- count_Rt2 - df1$diff[row_to_remove]
  chg_Rt1_Rt2 <- count_Rt2 - count_Rt1
  pct_Rt1_Rt2 <- (chg_Rt1_Rt2 / count_Rt1) *100
  named_areas <- df1$unit[row_to_remove]
}

# Add result to the output data frame
df2 <- df2 %>% add_row(unit = named_areas, adjusted = master_loop,
                      Ut1 = df1$time1[row_to_remove], Ut2 = df1$time2[row_to_remove],
                      Rt1 = count_Rt1, Rt2 = count_Rt2,
                      chg = chg_Rt1_Rt2, pct = pct_Rt1_Rt2)

# Adjust the row we just used in one of two ways:
# 1. remove the actual row entirely
if (analysisMethod == "remove"){
  df1 <-df1[-c(row_to_remove), ]
}

#2. adjust the Rt2 to match Rt1 resulting in a zero diff
# but show that diff as < lowest diff in the data set so that
# the program does not stall with too many zeros
if (analysisMethod == "match"){
  df1$time2[row_to_remove] <- df1$time1[row_to_remove]
  df1$diff[row_to_remove] <- -999 # this should be changed to always less than
  # the lowest diff score in the data set
}

} # end master_loop

# Calculate ODI and NCDI indices -----
NumContributed <- length(which(df2$chg > 0))
ODI <- NumContributed / source_rows
NCDI <- (numPositive - NumContributed) / source_rows
ODI.text <- paste("O.D.I. = ", format(ODI, digits = 3), "after \nadjusting",
                  NumContributed, "of the", source_rows, "units")

# Tidy up names for data frame -----

df2 <- df2 %>%
  rename(unit_t1 = Ut1, unit_t2 = Ut2, region_t1 = Rt1, region_t2 = Rt2)

# Plot -----

df3 <- df2

```

```

plot.adjustment <- ""
if (nrow(df3) > 151) {
  df3 <- df3[1:151, ]
  plot.adjustment <- "Plot only shows first\n100 areas adjusted"
}

p <- ggplot(df3, aes(x=reorder(unit, adjusted), y=pct, group = 1)) +
  geom_line(color="#3277a8") +
  geom_point(shape=21, color="white", fill="#3277a8", size=2) +
  geom_hline(color="grey", yintercept=0) +
  labs(title="Dispersion of crime change",
       x="Area adjusted", y = "Remaining crime change for region") +
  annotate(
    geom = "curve", x = NumContributed+4, y = 1.5,
    xend = NumContributed+1, yend = 0.2,
    curvature = .2, arrow = arrow(length = unit(2, "mm"))
  ) +
  annotate(geom = "text", x = NumContributed+4.1, y = 1.5,
          label = ODI.text, hjust = "left") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +

  annotate(geom = "text", x = 2, y = df2$pct[1],
          label = paste0(format(df2$pct[1], digits = 3), "% overall"), hjust = "left") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

if (plot.adjustment != "") {
  p <- p +
    annotate(geom = "text", x = 100, y = df3$pct[1]-1, label = plot.adjustment, hjust = "right")
}

p

# Create return list -----

output <- list(df2, p, NumContributed, ODI, NCDI)
return(output)
}

prepost_cj <- cj_exp %>%
  mutate(begin_date = ISOweek2date(paste(year, paste0("W", sprintf("%02d", week))), 1, sep = "--"),
         end_date = begin_date+weeks(1)-days(1),
         pre_post_floyd = ifelse(end_date <= as.Date("2020-05-25"), 0, 1)) %>%
  filter(end_date <= as.Date("2022-08-20")) %>%
  group_by(GEOID, pre_post_floyd) %>%
  summarize(car_jack = sum(car_jack, na.rm = T),
            total_pop = sum(total_pop, na.rm = T)) %>%
  mutate(car_jack_rate = car_jack/total_pop*1000) %>%
  select(GEOID, pre_post_floyd, car_jack, car_jack_rate) %>%
  st_drop_geometry() %>%
  pivot_wider(names_from = pre_post_floyd, values_from = c(car_jack, car_jack_rate)) %>%

```



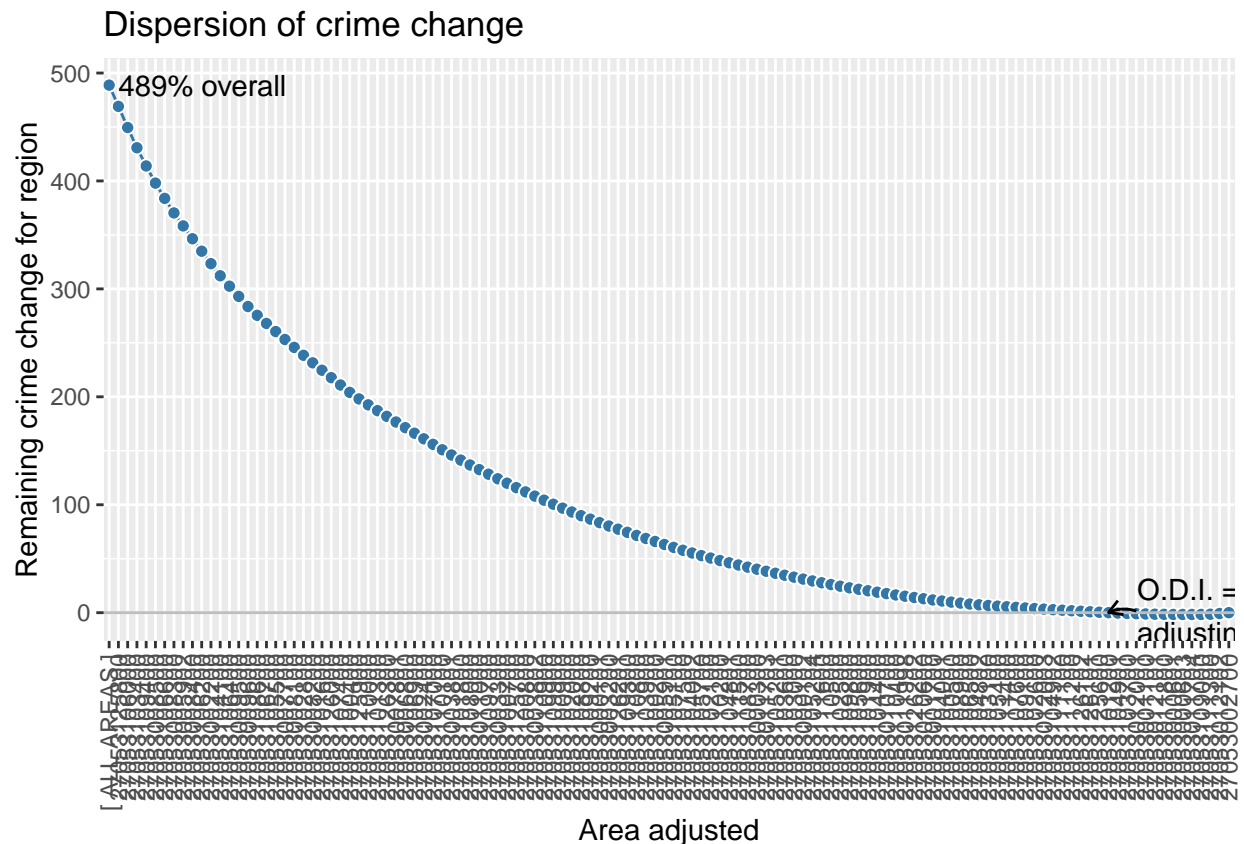
```
mutate(GEOID = as.character(GEOID))

## `summarise()` has grouped output by 'GEOID'. You can override using the
## `.groups` argument.

output <- crimedisposition(as.data.frame(prepost_cj), 'GEOID', 'car_jack_rate_0', 'car_jack_rate_1')

ouput_data <- output[[1]]
n_remove <- output[[3]]
odi <- output[[4]] #ratio of n removed to n overall
ncdi <- output[[5]] #ratio of areas not contributing to overall increase but still increase to overall

output[[2]]
```



```
ggsave(filename = "Car Jacking/Figures for PAA/fig5.png", bg="white", width = 10, height = 8)
```

Homicide

```
prepost_hom <- homicide %>%
  mutate(begin_date = ISOweek2date(paste(year, paste0("W", sprintf("%02d", week)), 1, sep = "-")),
         end_date = begin_date+weeks(1)-days(1),
         pre_post_floyd = ifelse(end_date <= as.Date("2020-05-25"), 0, 1)) %>%
  group_by(GEOID, pre_post_floyd) %>%
  summarize(homicide = sum(homicide, na.rm = T),
         total_pop = sum(total_pop, na.rm = T)) %>%
  mutate(homicide_rate = homicide/total_pop*1000) %>%
```

```
st_drop_geometry() %>%
select(GEOID, pre_post_floyd, homicide, homicide_rate) %>%
pivot_wider(names_from = pre_post_floyd, values_from = c(homicide, homicide_rate)) %>%
mutate(GEOID = as.character(GEOID))
```

`summarise()` has grouped output by 'GEOID'. You can override using the
`.groups` argument.

```
output_homicide <- crimedispersion(as.data.frame(prepost_hom), 'GEOID', 'homicide_0', 'homicide_1')
```

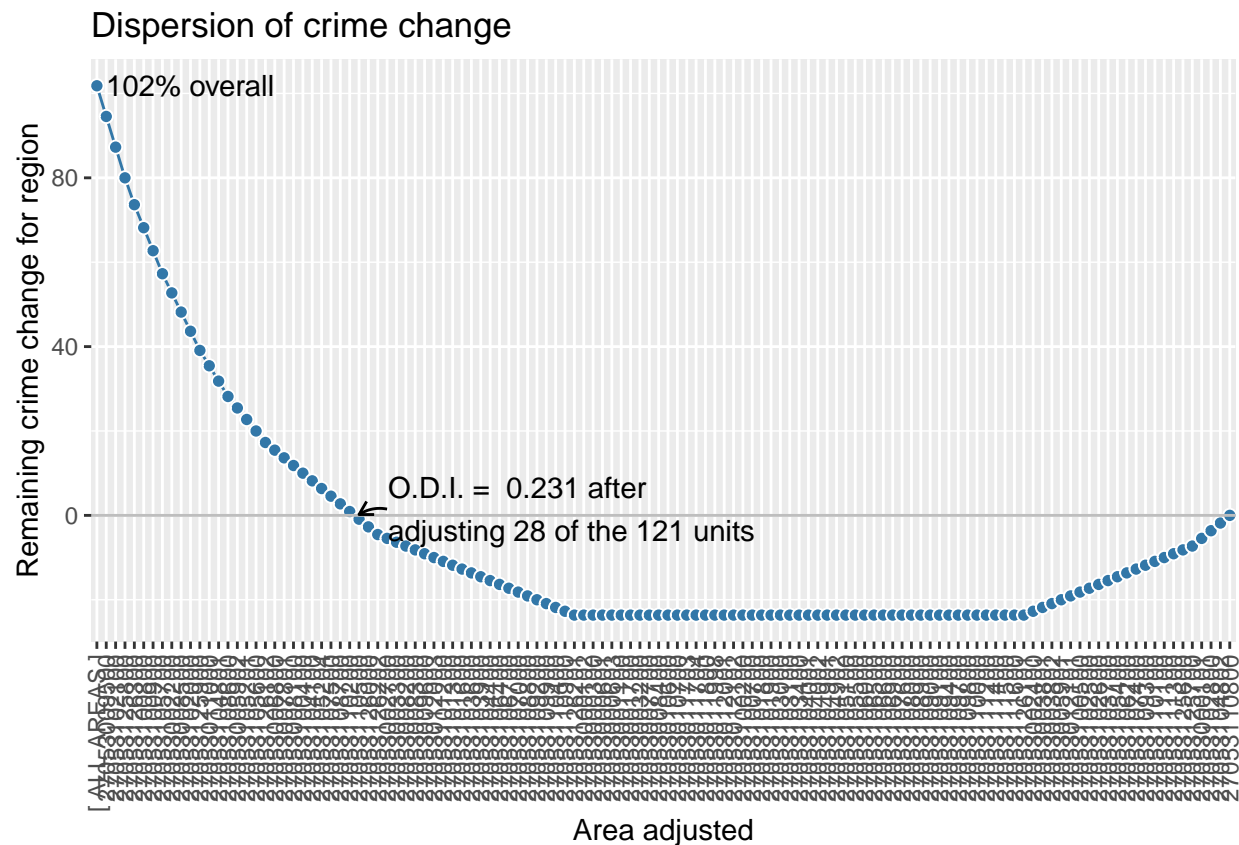
```
ouput_data <- output_homicide[[1]]
```

```
n_remove <- output_homicide[[3]]
```

```
odi <- output_homicide[[4]] #ratio of n removed to n overall
```

```
ncdi <- output_homicide[[5]] #ratio of areas not contributing to overall increase but still increase to
```

```
output_homicide[[2]]
```



```
ggsave(filename = "Car Jacking/Figures for PAA/fig6.png", bg="white", width = 10, height = 8)
```

Spatial Correlation *Change* in Carjackings and Homicide

Carjacking

```
library(sfdep)
```

```
cj_delta <- prepost_cj %>%
```

```

mutate(delta = car_jack_rate_1-car_jack_rate_0,
        GEOID = as.numeric(GEOID)) %>%
left_join(mpls_tract, by = "GEOID") %>%
st_as_sf()

nb <- st_contiguity(cj_delta, queen=TRUE)
wt <- st_weights(nb, style = "W")

global_moran_test(
  cj_delta$delta,
  nb,
  wt,
  alternative = "greater",
  randomization = TRUE)

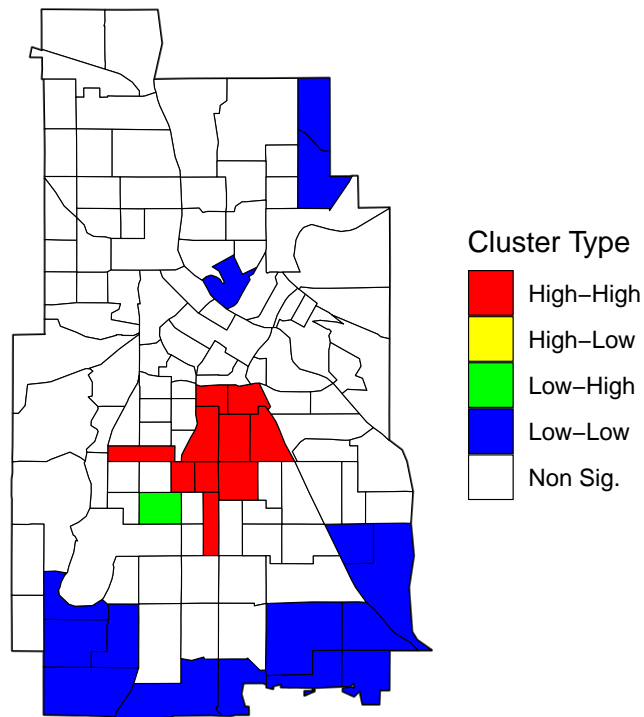
##
## Moran I test under randomisation
##
## data: x
## weights: listw
##
## Moran I statistic standard deviate = 8.736, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.453555803      -0.008333333      0.002795420

#LISA
cj_lisa <- local_moran(cj_delta$delta,
                      nb = nb,
                      wt = wt,
                      nsim = 1000,
                      iseed = set.seed(7188)) %>%
mutate(mean_p = ifelse(p_ii_sim <= 0.05, as.character(pysal), "Non Sig."),
       mean_p = factor(mean_p, levels = c("High-High", "High-Low", "Low-High",
                                           "Low-Low", "Non Sig.")))

cj_delta %>%
  cbind(cj_lisa) %>%
  ggplot(aes(fill = mean_p)) +
  geom_sf() +
  geom_sf(lwd = 0.2, color = "black") +
  theme_void() +
  scale_fill_manual(values = c("red", "yellow", "green", "blue", "white"), drop = FALSE)+
  labs(title = "Figure 7: LISA Plot for Carjacking Change Pre/Post Police Murder",
       fill = "Cluster Type",
       caption = "Clusters significant at p < .05 with 1,000 simulations.")

```

Figure 7: LISA Plot for Carjacking Change Pre/Post Police Mu



Clusters significant at $p < .05$ with 1,000 simulations.

```
ggsave(filename = "Car Jacking/Figures for PAA/fig7.png", bg="white", width = 10, height = 8)
```

Homicide

```
hom_delta <- prepost_hom %>%
  mutate(delta = homicide_1-homicide_0,
          GEOID = as.numeric(GEOID)) %>%
  left_join(mpls_tract, by = "GEOID") %>%
  st_sf()

nb <- st_contiguity(hom_delta, queen=TRUE)
wt <- st_weights(nb, style = "W")

global_moran_test(
  hom_delta$delta,
  nb,
  wt,
  alternative = "greater",
  randomization = TRUE)

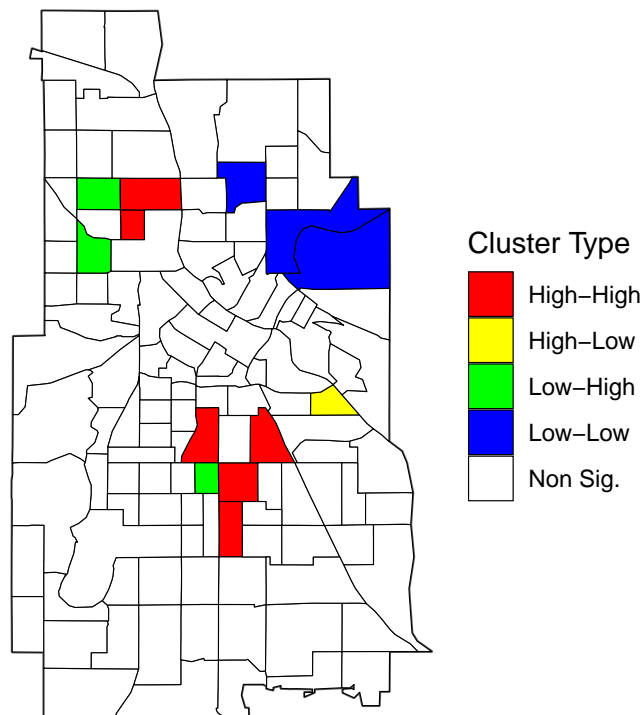
##
## Moran I test under randomisation
##
## data: x
## weights: listw
```

```
##
## Moran I statistic standard deviate = 3.6635, p-value = 0.0001244
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.185646900      -0.008333333      0.002803644

#LISA
hom_lisa <- local_moran(hom_delta$delta,
                        nb = nb,
                        wt = wt,
                        nsim = 1000,
                        iseed = set.seed(7188)) %>%
  mutate(mean_p = ifelse(p_ii_sim <= 0.05, as.character(pysal), "Non Sig.))

hom_delta %>%
  cbind(hom_lisa) %>%
  ggplot(aes(fill = mean_p)) +
  geom_sf() +
  geom_sf(lwd = 0.2, color = "black") +
  theme_void() +
  scale_fill_manual(values = c("red", "yellow", "green", "blue", "white"))+
  labs(title = "Figure 8: LISA Plot for Homicide Change Pre/Post Police Murder",
       fill = "Cluster Type",
       caption = "Clusters significant at p < .05 with 1,000 simulations.")
```

Figure 8: LISA Plot for Homicide Change Pre/Post Police Murc



Clusters significant at $p < .05$ with 1,000 simulations.

```
ggsave(filename = "Car Jacking/Figures for PAA/fig8.png", bg="white", width = 10, height = 8)
```

RE CJ Models

```

cj_exp_year_acs <- cj_exp_tract_year %>%
  left_join(acs_2020, by = c("GEOID")) %>%
  ungroup() %>%
  mutate(yearzero = year-2017,
         pov_rate_center = pov_rate-mean(pov_rate, na.rm = T),
         white_perc_center = scale(white_perc, center = T, scale = F),
         black_perc_center = scale(black_perc, center = T, scale = F),
         anyjack = ifelse(car_jack==0, 0, 1)) %>%
  drop_na()

cj_exp_prepost <- cj_exp %>%
  mutate(begin_date = ISOweek2date(paste(year, paste0("W", sprintf("%02d", week)), 1, sep = "-")),
         end_date = begin_date+weeks(1)-days(1),
         post_floyd = as.numeric(begin_date >= as.Date("2020-05-25")),
         post_floyd_3 = as.numeric(begin_date >= as.Date("2020-05-25")+months(3)),
         # stay_at_home = as.numeric(begin_date >= as.Date("2020-03-28")) &
         # state_of_emerg = as.numeric(begin_date >= as.Date("2020-03-13")),
         period = factor(case_when(
           post_floyd==0 & post_floyd_3==0 ~ "Pre-Killing",
           post_floyd>=1 & post_floyd_3==0 ~ "0-3 Months Post-Killing",
           post_floyd>=1 & post_floyd_3>=1 ~ "3+ Months Post-Killing"),
         levels = c("Pre-Killing", "0-3 Months Post-Killing", "3+ Months Post-Killing")),
         GEOID = as.character(GEOID),
         anyjack = ifelse(car_jack==0, 0, 1),
         t = 1:length(car_jack_rate)) %>%
  left_join(acs_2020, by = c("GEOID")) %>%
  drop_na()

library(lme4)
library(lmerTest)

library(lavaan)

cd_model_1 <- ' cd =~ unemp_rate + pov_rate + female_hh_rate + no_hs_dip_rate + black_perc
               black_perc =~ unemp_rate'

cfa_cd <- cfa(cd_model_1, data = cj_exp_prepost, std.lv = T)
modificationindices(cfa_cd)

```

##	lhs op	rhs	mi	epc	sepc.lv	sepc.all	sepc.nox
## 13	unemp_rate ~~	pov_rate	776.997	6.514	6.514	0.184	0.184
## 14	unemp_rate ~~	female_hh_rate	640.565	-1.195	-1.195	-0.195	-0.195
## 15	unemp_rate ~~	no_hs_dip_rate	10.933	0.327	0.327	0.028	0.028
## 16	pov_rate ~~	female_hh_rate	414.268	-2.638	-2.638	-0.118	-0.118
## 17	pov_rate ~~	no_hs_dip_rate	62.778	2.032	2.032	0.048	0.048
## 18	pov_rate ~~	black_perc	107.800	-8.340	-8.340	-0.141	-0.141
## 19	female_hh_rate ~~	no_hs_dip_rate	88.342	0.548	0.548	0.075	0.075

```
## 20 female_hh_rate ~~      black_perc 703.770  5.041   5.041   0.494   0.494
## 21 no_hs_dip_rate ~~      black_perc 291.676 -7.088  -7.088  -0.367  -0.367
```

```
summary(cfa_cd, fit.measures=TRUE, standardized = T)
```

```
## lavaan 0.6-12 ended normally after 32 iterations
##
##      Estimator                      ML
##      Optimization method          NLMINB
##      Number of model parameters      11
##
##      Number of observations          38040
##
## Model Test User Model:
##
##      Test statistic                  1433.656
##      Degrees of freedom                4
##      P-value (Chi-square)             0.000
##
## Model Test Baseline Model:
##
##      Test statistic                  96858.000
##      Degrees of freedom              10
##      P-value                         0.000
##
## User Model versus Baseline Model:
##
##      Comparative Fit Index (CFI)      0.985
##      Tucker-Lewis Index (TLI)         0.963
##
## Loglikelihood and Information Criteria:
##
##      Loglikelihood user model (H0)     -600873.408
##      Loglikelihood unrestricted model (H1) -600156.580
##
##      Akaike (AIC)                     1201768.815
##      Bayesian (BIC)                    1201862.826
##      Sample-size adjusted Bayesian (BIC) 1201827.868
##
## Root Mean Square Error of Approximation:
##
##      RMSEA                            0.097
##      90 Percent confidence interval - lower 0.093
##      90 Percent confidence interval - upper 0.101
##      P-value RMSEA <= 0.05              0.000
##
## Standardized Root Mean Square Residual:
##
##      SRMR                             0.026
##
## Parameter Estimates:
##
##      Standard errors                  Standard
##      Information                      Expected
##      Information saturated (h1) model  Structured
```

```
##
## Latent Variables:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   cd =~
##     unemp_rate      3.520   0.023 154.568  0.000   3.520   0.749
##     pov_rate        8.350   0.067 123.785  0.000   8.350   0.591
##     female_hh_rate   2.148   0.013 162.986  0.000   2.148   0.738
##     no_hs_dip_rate   4.664   0.026 176.182  0.000   4.664   0.782
##     black_perc      17.533   0.075 234.988  0.000  17.533   0.959
##
## Covariances:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   .unemp_rate ~~
##     .black_perc      -9.425   0.237 -39.771  0.000  -9.425  -0.583
##
## Variances:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   .unemp_rate        9.680   0.100  97.059  0.000   9.680   0.439
##   .pov_rate       129.663   0.973 133.264  0.000 129.663   0.650
##   .female_hh_rate    3.867   0.032 122.544  0.000   3.867   0.456
##   .no_hs_dip_rate   13.811   0.121 114.574  0.000  13.811   0.388
##   .black_perc       26.971   1.022  26.403  0.000  26.971   0.081
##   cd                1.000                1.000   1.000
cd_predict <- as.vector(lavPredict(cfa_cd, newdata = cj_exp_prepost))
cj_exp_prepost$conc_dis <- cd_predict

re <- lmer(car_jack_rate~t+post_floyd+post_floyd_3+conc_dis+
  age_19_29_perc+age_30_49_perc+age_50_69_perc+
  age_70_plus_perc+ post_floyd:conc_dis+post_floyd_3:conc_dis+
  (1|GEOID),
  data = cj_exp_prepost)

## Warning: Some predictor variables are on very different scales: consider
## rescaling

## Warning: Some predictor variables are on very different scales: consider
## rescaling

summary(re)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
##   car_jack_rate ~ t + post_floyd + post_floyd_3 + conc_dis + age_19_29_perc +
##     age_30_49_perc + age_50_69_perc + age_70_plus_perc + post_floyd:conc_dis +
##     post_floyd_3:conc_dis + (1 | GEOID)
##   Data: cj_exp_prepost
##
## REML criterion at convergence: -74814
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.291 -0.242 -0.121  0.005 33.376
```



```

##
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   GEOID    (Intercept) 0.000132 0.01149
##   Residual              0.008113 0.09007
## Number of obs: 38040, groups: GEOID, 120
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    5.052e-02  1.473e-02 1.132e+02   3.430 0.000842 ***
## t              1.379e-07  1.079e-07 1.130e+02   1.278 0.203926
## post_floyd     2.486e-02  2.281e-03 3.792e+04  10.895 < 2e-16 ***
## post_floyd_3   -1.500e-03  2.319e-03 3.792e+04  -0.647 0.517690
## conc_dis       6.992e-04  1.827e-03 1.253e+02   0.383 0.702547
## age_19_29_perc -4.491e-04  1.509e-04 1.130e+02  -2.976 0.003577 **
## age_30_49_perc -4.090e-04  2.465e-04 1.130e+02  -1.659 0.099904 .
## age_50_69_perc -1.056e-03  2.793e-04 1.130e+02  -3.779 0.000253 ***
## age_70_plus_perc -5.403e-04  3.368e-04 1.130e+02  -1.604 0.111424
## post_floyd:conc_dis  1.850e-02  2.320e-03 3.792e+04   7.975 1.56e-15 ***
## post_floyd_3:conc_dis -8.651e-03  2.359e-03 3.792e+04  -3.668 0.000245 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr) t      pst_fl pst__3 cnc_ds a_19_2 a_30_4 a_50_6 a_70__
## t      -0.021
## post_floyd -0.011 -0.005
## post_flyd_3 0.000 -0.003 -0.913
## conc_dis   -0.725 0.050 0.000 0.000
## ag_19_29_pr -0.924 -0.052 0.000 0.000 0.680
## ag_30_49_pr -0.878 -0.108 0.000 0.000 0.591 0.747
## ag_50_69_pr -0.741 -0.125 0.001 0.000 0.604 0.719 0.444
## ag_70_pls_p -0.303 -0.036 0.000 0.000 0.191 0.278 0.282 -0.119
## pst_flyd:c_ 0.000 0.000 0.000 0.000 -0.092 0.000 0.000 0.000 0.000
## pst_fly_3:_ 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##      pst:_
## t
## post_floyd
## post_flyd_3
## conc_dis
## ag_19_29_pr
## ag_30_49_pr
## ag_50_69_pr
## ag_70_pls_p
## pst_flyd:c_
## pst_fly_3:_ -0.913
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
re_logit_cd <- glmer(anyjack ~ t+post_floyd+post_floyd_3+conc_dis+
  age_19_29_perc+age_30_49_perc+age_50_69_perc+
  age_70_plus_perc+ post_floyd:conc_dis+post_floyd_3:conc_dis+
  (1|GEOID),
  data = cj_exp_prepost, family = binomial)

```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 3.43449 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable:
## - Rescale variables?;Model is nearly unidentifiable: large eigenvalue ratio
## - Rescale variables?
```

```
summary(re_logit_cd)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: anyjack ~ t + post_floyd + post_floyd_3 + conc_dis + age_19_29_perc +
## age_30_49_perc + age_50_69_perc + age_70_plus_perc + post_floyd:conc_dis +
## post_floyd_3:conc_dis + (1 | GEOID)
## Data: cj_exp_prepost
##
##      AIC      BIC    logLik deviance df.resid
## 10380.8 10483.3 -5178.4 10356.8   38028
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.7191 -0.2141 -0.1329 -0.0772  21.2478
##
## Random effects:
## Groups Name      Variance Std.Dev.
## GEOID (Intercept) 0.4228   0.6502
## Number of obs: 38040, groups:  GEOID, 120
##
## Fixed effects:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.500e+00  9.026e-01  -3.877 0.000106 ***
## t               1.337e-05  8.136e-06   1.644 0.100256
## post_floyd      1.786e+00  1.288e-01  13.866 < 2e-16 ***
## post_floyd_3    5.340e-02  1.144e-01   0.467 0.640649
## conc_dis        5.760e-01  1.181e-01   4.878 1.07e-06 ***
## age_19_29_perc  -8.335e-03  9.180e-03  -0.908 0.363887
## age_30_49_perc   1.447e-03  1.493e-02   0.097 0.922832
## age_50_69_perc  -6.457e-02  1.718e-02  -3.759 0.000171 ***
## age_70_plus_perc -1.858e-02  2.026e-02  -0.917 0.359044
## post_floyd:conc_dis -8.203e-02  1.031e-01  -0.795 0.426430
## post_floyd_3:conc_dis -2.557e-01  9.589e-02  -2.667 0.007655 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) t      pst_fl pst__3 cnc_ds a_19_2 a_30_4 a_50_6 a_70__
## t              -0.096
## post_floyd     -0.040  0.010
## post_flyd_3    -0.001  0.001 -0.803
## conc_dis       -0.727  0.137  0.125  0.004
## ag_19_29_pr    -0.924  0.004 -0.002  0.000  0.667
## ag_30_49_pr    -0.867 -0.103 -0.002  0.000  0.570  0.749
```

```

## ag_50_69_pr -0.733 -0.096 -0.004 0.000 0.563 0.718 0.446
## ag_70_pls_p -0.276 -0.021 -0.004 -0.003 0.164 0.258 0.263 -0.141
## pst_flyd:c_ 0.020 -0.008 -0.450 0.347 -0.223 0.001 0.002 0.001 0.005
## pst_fly_3:_ 0.002 -0.017 0.329 -0.408 -0.013 0.001 0.002 0.001 0.009
##          pst:_
## t
## post_floyd
## post_flyd_3
## conc_dis
## ag_19_29_pr
## ag_30_49_pr
## ag_50_69_pr
## ag_70_pls_p
## pst_flyd:c_
## pst_fly_3:_ -0.802
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 3.43449 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?
## Model is nearly unidentifiable: large eigenvalue ratio
## - Rescale variables?

#build in police covariates
#what other covariates do we need here?
#crude model - no post-treatment control
#businesses - crime generators
#percent single males
#percent "isolated" youth
#similar story with homicide?
#FE models

cj_exp_prepost <- cj_exp_prepost %>%
  mutate(GEOID = as.numeric(GEOID)) %>%
  left_join(homicide, by = c("GEOID", "year", "week")) %>%
  mutate(anyhom = ifelse(homicide==0, 0, 1))

re_homicide <- lmer(homicide_rate~t+post_floyd+post_floyd_3+conc_dis+
  age_19_29_perc+age_30_49_perc+age_50_69_perc+
  age_70_plus_perc+ post_floyd:conc_dis+post_floyd_3:conc_dis+
  (1|GEOID),
  data = cj_exp_prepost)

## Warning: Some predictor variables are on very different scales: consider
## rescaling

## Warning: Some predictor variables are on very different scales: consider
## rescaling

summary(re_homicide)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:

```

```

## homicide_rate ~ t + post_floyd + post_floyd_3 + conc_dis + age_19_29_perc +
##   age_30_49_perc + age_50_69_perc + age_70_plus_perc + post_floyd:conc_dis +
##   post_floyd_3:conc_dis + (1 | GEOID)
## Data: cj_exp_prepost
##
## REML criterion at convergence: -131908.3
##
## Scaled residuals:
##   Min      1Q  Median      3Q      Max
## -1.398 -0.100 -0.030 -0.007  45.410
##
## Random effects:
##   Groups   Name      Variance Std.Dev.
##   GEOID    (Intercept) 1.617e-05 0.004021
##   Residual              1.810e-03 0.042550
## Number of obs: 38040, groups: GEOID, 120
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    9.667e-03  5.489e-03 1.133e+02   1.761   0.0809 .
## t              1.817e-08  4.021e-08 1.130e+02   0.452   0.6522
## post_floyd     7.919e-03  1.078e-03 3.792e+04   7.348 2.06e-13 ***
## post_floyd_3   -4.939e-03  1.096e-03 3.792e+04  -4.508 6.56e-06 ***
## conc_dis       1.617e-03  6.910e-04 1.331e+02   2.340   0.0208 *
## age_19_29_perc -6.446e-05  5.623e-05 1.130e+02  -1.146   0.2541
## age_30_49_perc -1.635e-04  9.186e-05 1.130e+02  -1.780   0.0777 .
## age_50_69_perc  1.222e-05  1.041e-04 1.130e+02   0.117   0.9068
## age_70_plus_perc -2.764e-04  1.255e-04 1.130e+02  -2.203   0.0296 *
## post_floyd:conc_dis  9.761e-03  1.096e-03 3.792e+04   8.906 < 2e-16 ***
## post_floyd_3:conc_dis -6.798e-03  1.114e-03 3.792e+04  -6.102 1.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) t      pst_fl pst__3 cnc_ds a_19_2 a_30_4 a_50_6 a_70__
## t              -0.021
## post_floyd     -0.014 -0.004
## post_flyd_3    0.000 -0.003 -0.913
## conc_dis       -0.714  0.050  0.000  0.000
## ag_19_29_pr    -0.923 -0.052  0.000  0.000  0.669
## ag_30_49_pr    -0.878 -0.108  0.000  0.000  0.582  0.747
## ag_50_69_pr    -0.741 -0.125  0.000  0.000  0.595  0.719  0.444
## ag_70_pls_p    -0.303 -0.036  0.000  0.000  0.188  0.278  0.282 -0.119
## pst_flyd:c_     0.000  0.000  0.000  0.000 -0.114  0.000  0.000  0.000  0.000
## pst_fly_3:_     0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
##              pst_:_
## t
## post_floyd
## post_flyd_3
## conc_dis
## ag_19_29_pr
## ag_30_49_pr
## ag_50_69_pr
## ag_70_pls_p

```

```

## pst_flyd:c_
## pst_fly_3:_ -0.913
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
re_logit_cd_homicide <- glmer(anyhom ~ t+post_floyd+post_floyd_3+conc_dis+
  age_19_29_perc+age_30_49_perc+age_50_69_perc+
  age_70_plus_perc+ post_floyd:conc_dis+post_floyd_3:conc_dis+
  (1|GEOID),
  data = cj_exp_prepost, family = binomial)

## Warning: Some predictor variables are on very different scales: consider
## rescaling

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues

summary(re_logit_cd)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: anyjack ~ t + post_floyd + post_floyd_3 + conc_dis + age_19_29_perc +
## age_30_49_perc + age_50_69_perc + age_70_plus_perc + post_floyd:conc_dis +
## post_floyd_3:conc_dis + (1 | GEOID)
## Data: cj_exp_prepost
##
##      AIC      BIC   logLik deviance df.resid
## 10380.8 10483.3 -5178.4 10356.8    38028
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.7191 -0.2141 -0.1329 -0.0772  21.2478
##
## Random effects:
##      Groups Name      Variance Std.Dev.
##      GEOID (Intercept) 0.4228   0.6502
## Number of obs: 38040, groups:  GEOID, 120
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.500e+00  9.026e-01  -3.877 0.000106 ***
## t               1.337e-05  8.136e-06   1.644 0.100256
## post_floyd      1.786e+00  1.288e-01  13.866 < 2e-16 ***
## post_floyd_3    5.340e-02  1.144e-01   0.467 0.640649
## conc_dis        5.760e-01  1.181e-01   4.878 1.07e-06 ***
## age_19_29_perc  -8.335e-03  9.180e-03  -0.908 0.363887
## age_30_49_perc   1.447e-03  1.493e-02   0.097 0.922832
## age_50_69_perc  -6.457e-02  1.718e-02  -3.759 0.000171 ***
## age_70_plus_perc -1.858e-02  2.026e-02  -0.917 0.359044
## post_floyd:conc_dis -8.203e-02  1.031e-01  -0.795 0.426430
## post_floyd_3:conc_dis -2.557e-01  9.589e-02  -2.667 0.007655 **
## ---

```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) t          pst_fl pst__3 cnc_ds a_19_2 a_30_4 a_50_6 a_70__
## t          -0.096
## post_floyd -0.040  0.010
## post_flyd_3 -0.001  0.001 -0.803
## conc_dis   -0.727  0.137  0.125  0.004
## ag_19_29_pr -0.924  0.004 -0.002  0.000  0.667
## ag_30_49_pr -0.867 -0.103 -0.002  0.000  0.570  0.749
## ag_50_69_pr -0.733 -0.096 -0.004  0.000  0.563  0.718  0.446
## ag_70_pls_p -0.276 -0.021 -0.004 -0.003  0.164  0.258  0.263 -0.141
## pst_flyd:c_  0.020 -0.008 -0.450  0.347 -0.223  0.001  0.002  0.001  0.005
## pst_fly_3:_  0.002 -0.017  0.329 -0.408 -0.013  0.001  0.002  0.001  0.009
##          pst:_
## t
## post_floyd
## post_flyd_3
## conc_dis
## ag_19_29_pr
## ag_30_49_pr
## ag_50_69_pr
## ag_70_pls_p
## pst_flyd:c_
## pst_fly_3:_ -0.802
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 3.43449 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
## - Rescale variables?
## Model is nearly unidentifiable: large eigenvalue ratio
## - Rescale variables?

class(re) <- "lmerMod"
class(re_logit_cd) <- "lmerMod"

stargazer(re, re_logit_cd,
  title = "Interrupted Time Series Models of Carjackings, MPLS 2017-2022",
  covariate.labels = c("T", "Post-Killing", "Post-Killing 3 Months",
    "Conc. Dis.", "Age 19-29", "Age 30-49",
    "Age 50-69", "Age 70+",
    "Post-Killing X Conc. Dis.",
    "Post-Killing 3 Months X Conc. Dis."),
  header = F,
  dep.var.caption = "Carjacking",
  dep.var.labels = c("Rate per 1,000", "Any Carjacking"),
  model.names = FALSE,
  column.labels = c("RE HLM", "RE Logit"),
  report = "vcs",
  ci=TRUE,
  ci.level=0.95,
  ci.separator = "|",
  notes = "95\\% Confidence Intervals in parentheses",

```

```

single.row = F,
omit.stat = c("adj.rsq", "aic", "bic"),
#star.cutoffs = c(.05, .01, .001), star.char = c("?", "**", "***"),
add.lines = list(c("SD(Tract)", .008, .650),
                 c("SD(Residual)", .067, "")),
notes.append = F)

```

Table 1: Interrupted Time Series Models of Carjackings, MPLS 2017-2022

	Carjacking	
	Rate per 1,000 RE HLM (1)	Any Carjacking RE Logit (2)
T	0.00000 (−0.00000 0.00000)	0.00001 (−0.00000 0.00003)
Post-Killing	0.025 (0.020 0.029)	1.786 (1.533 2.038)
Post-Killing 3 Months	−0.002 (−0.006 0.003)	0.053 (−0.171 0.278)
Conc. Dis.	0.001 (−0.003 0.004)	0.576 (0.345 0.807)
Age 19-29	−0.0004 (−0.001 −0.0002)	−0.008 (−0.026 0.010)
Age 30-49	−0.0004 (−0.001 0.0001)	0.001 (−0.028 0.031)
Age 50-69	−0.001 (−0.002 −0.001)	−0.065 (−0.098 −0.031)
Age 70+	−0.001 (−0.001 0.0001)	−0.019 (−0.058 0.021)
Post-Killing X Conc. Dis.	0.019 (0.014 0.023)	−0.082 (−0.284 0.120)
Post-Killing 3 Months X Conc. Dis.	−0.009 (−0.013 −0.004)	−0.256 (−0.444 −0.068)
Constant	0.051 (0.022 0.079)	−3.500 (−5.269 −1.731)
SD(Tract)	0.008	0.65
SD(Residual)	0.067	
Observations	38,040	38,040
Log Likelihood	37,407.000	−5,178.390

Note:

95% Confidence Intervals in parentheses

RE CJ Models

```
class(re_homicide) <- "lmerMod"
class(re_logit_cd_homicide) <- "lmerMod"

stargazer(re_homicide, re_logit_cd_homicide,
  title = "Interrupted Time Series Models of Homicide, MPLS 2017-2022",
  covariate.labels = c("T", "Post-Killing", "Post-Killing 3 Months",
    "Conc. Dis.", "Age 19-29", "Age 30-49",
    "Age 50-69", "Age 70+",
    "Post-Killing X Conc. Dis.",
    "Post-Killing 3 Months X Conc. Dis."),
  header = F,
  dep.var.caption = "Homicide",
  dep.var.labels = c("Rate per 1,000", "Any Homicide"),
  model.names = FALSE,
  column.labels = c("RE HLM", "RE Logit"),
  report = "vcs",
  ci=TRUE,
  ci.level=0.95,
  ci.separator = "|",
  notes = "95\\% Confidence Intervals in parentheses",
  single.row = F,
  omit.stat = c("adj.rsq", "aic", "bic"),
  #star.cutoffs = c(.05, .01, .001), star.char = c("*", "**", "***"),
  add.lines = list(c("SD(Tract)", .003, .031),
    c("SD(Residual)", .065, "-")),
  notes.append = F)
```


Table 2: Interrupted Time Series Models of Homicide, MPLS 2017-2022

	Homicide	
	Rate per 1,000 RE HLM	Any Homicide RE Logit
	(1)	(2)
T	0.00000 (−0.00000 0.00000)	0.00002 (−0.00000 0.00004)
Post-Killing	0.008 (0.006 0.010)	1.449 (0.966 1.932)
Post-Killing 3 Months	−0.005 (−0.007 −0.003)	−0.573 (−1.033 −0.112)
Conc. Dis.	0.002 (0.0003 0.003)	0.915 (0.566 1.264)
Age 19-29	−0.0001 (−0.0002 0.00005)	0.016 (−0.011 0.043)
Age 30-49	−0.0002 (−0.0003 0.00002)	0.012 (−0.032 0.056)
Age 50-69	0.00001 (−0.0002 0.0002)	0.017 (−0.035 0.070)
Age 70+	−0.0003 (−0.001 −0.00003)	−0.043 (−0.107 0.020)
Post-Killing X Conc. Dis.	0.010 (0.008 0.012)	0.145 (−0.175 0.465)
Post-Killing 3 Months X Conc. Dis.	−0.007 (−0.009 −0.005)	−0.182 (−0.487 0.123)
Constant	0.010 (−0.001 0.020)	−7.125 (−9.722 −4.528)
SD(Tract)	0.003	0.031
SD(Residual)	0.065	-
Observations	38,040	38,040
Log Likelihood	65,954.170	−1,661.554

Note:

95% Confidence Intervals in parentheses