Car Jacking MPLS - Tract

Ryan Larson

2023-03-13

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 and Denominators

```
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.
```

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

```
##   |                                                                              |
```

```r
#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) %>%
```

```r
  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-cens
## This message is displayed once per session.
```

Expanded MPLS Carjacking (Crime Incidents) Data

```
cj_exp <- read_csv("Car Jacking/MPDdata_2017to2022.csv") %>%
   mutate(date=mdy_hm(reporteddate),
         year=isoyear(date),
         week=isoweek(date)) %>%
  select(casenumber, year, week, latitude, longitude) %>%
  distinct(casenumber, .keep_all = TRUE) %>% #collapsing to incident-level
  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: 3779 Columns: 28
## -- Column specification --------------------------------------------------------
```

```
## Delimiter: ","
## chr (23): casenumber, dataset, closurecode, closurecode_mpd, reporteddate, c...
## dbl  (4): precinct, latitude, longitude, age
## lgl  (1): dateofbirth
##
## 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("2022-12-31")) %>%
  ungroup() %>%
  mutate(csma = 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 0.030500004
```

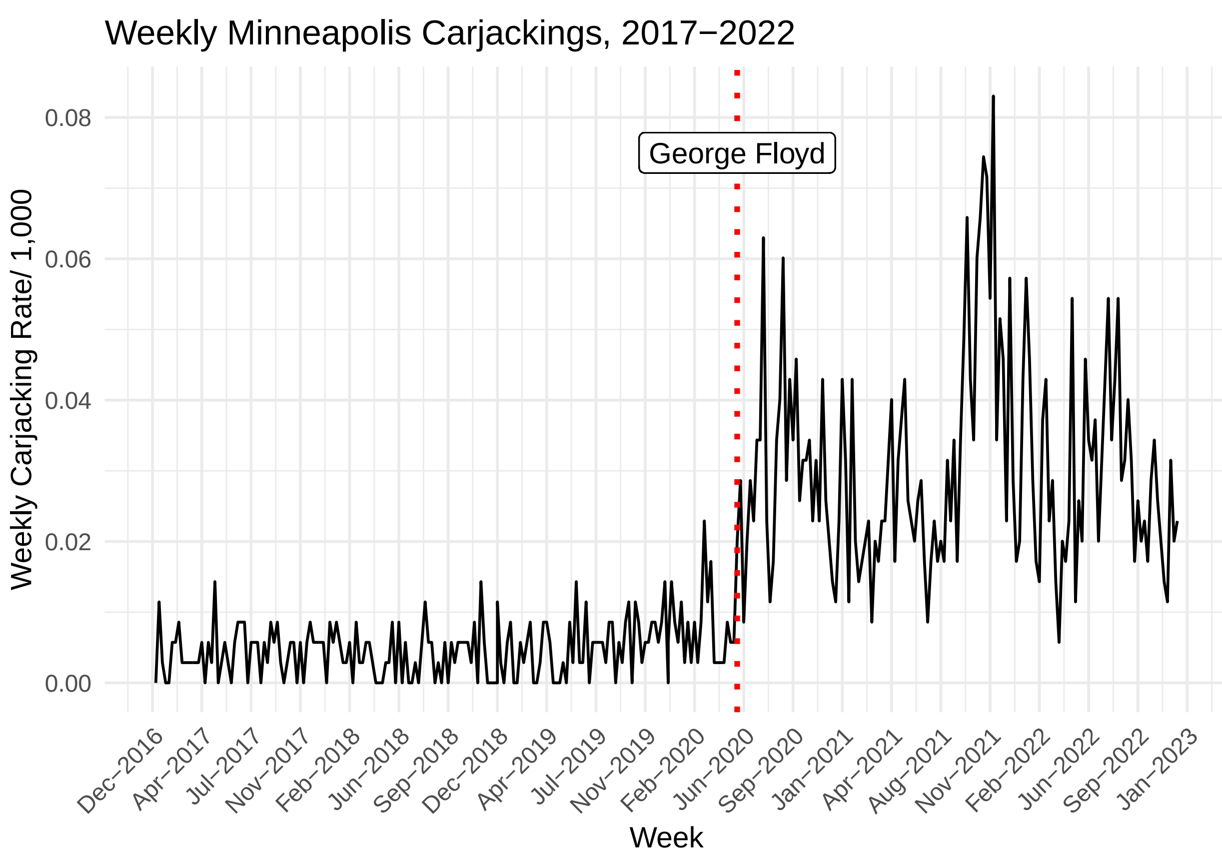
```
post_mean/pre_mean
```

```
## [1] 6.307018
```

```
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=0.075),
             label = "George Floyd", show.legend = FALSE)+
  labs(title = "Weekly Minneapolis Carjackings, 2017-2022",
       x = "Week",
       y = "Weekly Carjacking Rate/ 1,000",
       color = NULL)+
```

```r
  #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: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.

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

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

## Warning: Use of `cj_exp_week$week` is discouraged.
## i Use `week` instead.
```




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

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

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

## Warning: Use of `cj_exp_week$week` is discouraged.
## i Use `week` instead.
```

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 = 51.174, p-value < 2.2e-16
```

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

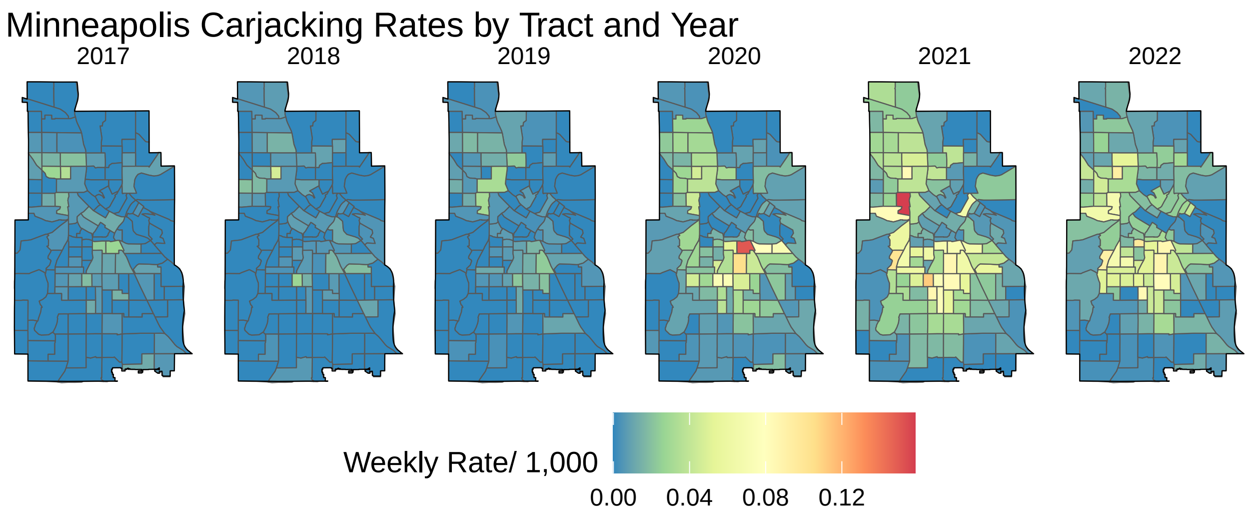
```
##
##   Optimal 2-segment partition:
##
## Call:
## breakpoints.formula(formula = car_jack_rate ~ t, breaks = 1,
##     data = cj_exp_week)
##
## Breakpoints at observation number:
## 180
##
## Corresponding to breakdates:
## 0.5714286
```

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 = "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")
```




```
ggsave(filename = "Car Jacking/Figures for publication/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.
```

```r
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.
```

```r
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.
```

```r
#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, centergbsid, 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.
```

```r
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, centergbsid, 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.
```

```r
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, centergbsid, 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, centergbsid, 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")
```

```
## Rows: 26526 Columns: 23
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (13): publicaddress, caseNumber, precinct, reportedDate, beginDate, repo...
## dbl  (9): X, Y, OBJECTID, reportedTime, beginTime, centerLong, centerLat, ce...
## lgl  (1): centergbsid
##
## 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) %>%
```

```r
  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

```r
#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("2022-12-31")) %>%
  ungroup() %>%
  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.
```

```r
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)
```

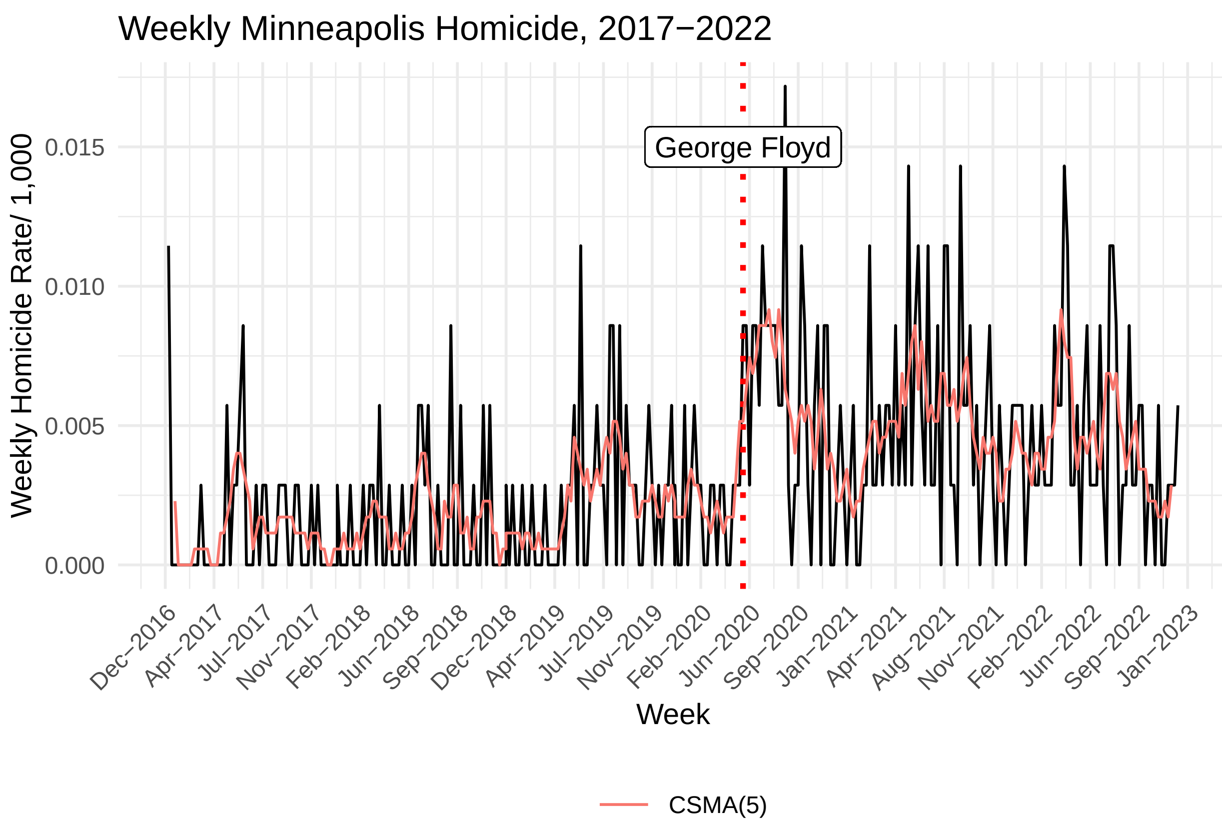
```
## [1] 0.001749826 0.004984354
```

```
post_mean/pre_mean
```

```
## [1] 2.848485
```

```
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 = "Weekly Minneapolis Homicide, 2017-2022",
       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: Use of `homicide_week$begin_date` is discouraged.
## i Use `begin_date` instead.

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

## Warning: Use of `homicide_week$week` is discouraged.
## i Use `week` instead.

## Warning: Removed 4 rows containing missing values (`geom_line()`).
```




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

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

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

## Warning: Use of `homicide_week$week` is discouraged.
## i Use `week` instead.

## Warning: Removed 4 rows containing missing values (`geom_line()`).
```

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 = 16.772, p-value = 1.21e-07
```

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

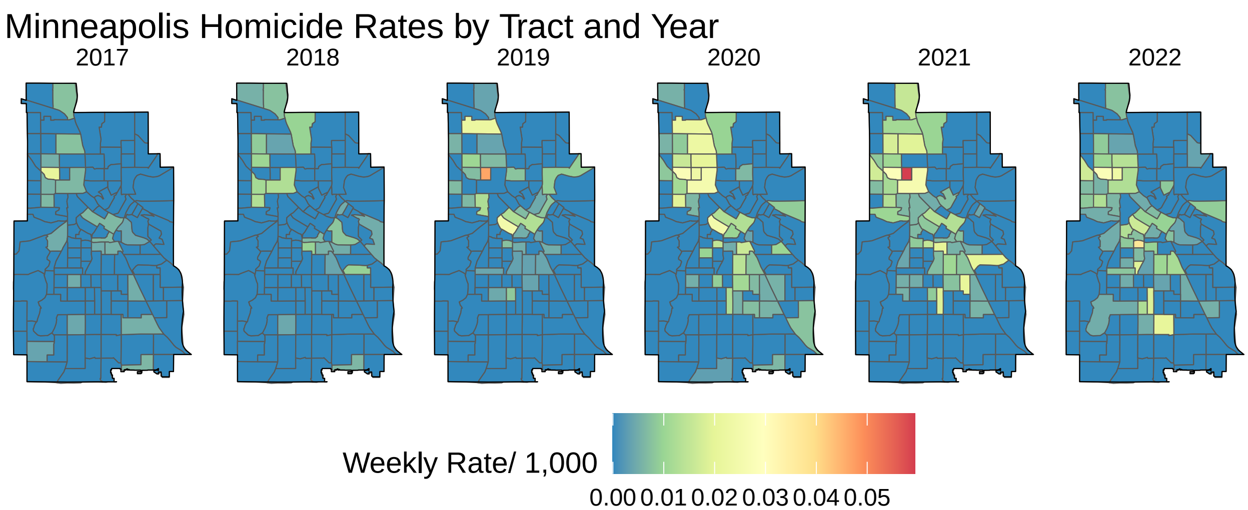
```
##
##   Optimal 2-segment partition:
##
## Call:
## breakpoints.formula(formula = homicide_rate ~ t, breaks = 1,
##     data = homicide_week)
##
## Breakpoints at observation number:
## 180
##
## Corresponding to breakdates:
## 0.5714286
```

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 = "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")
```




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

Dispersion of Change from 2017-2019 to 2020-2021

Car Jacking

```
crimedispersion <- 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]
```

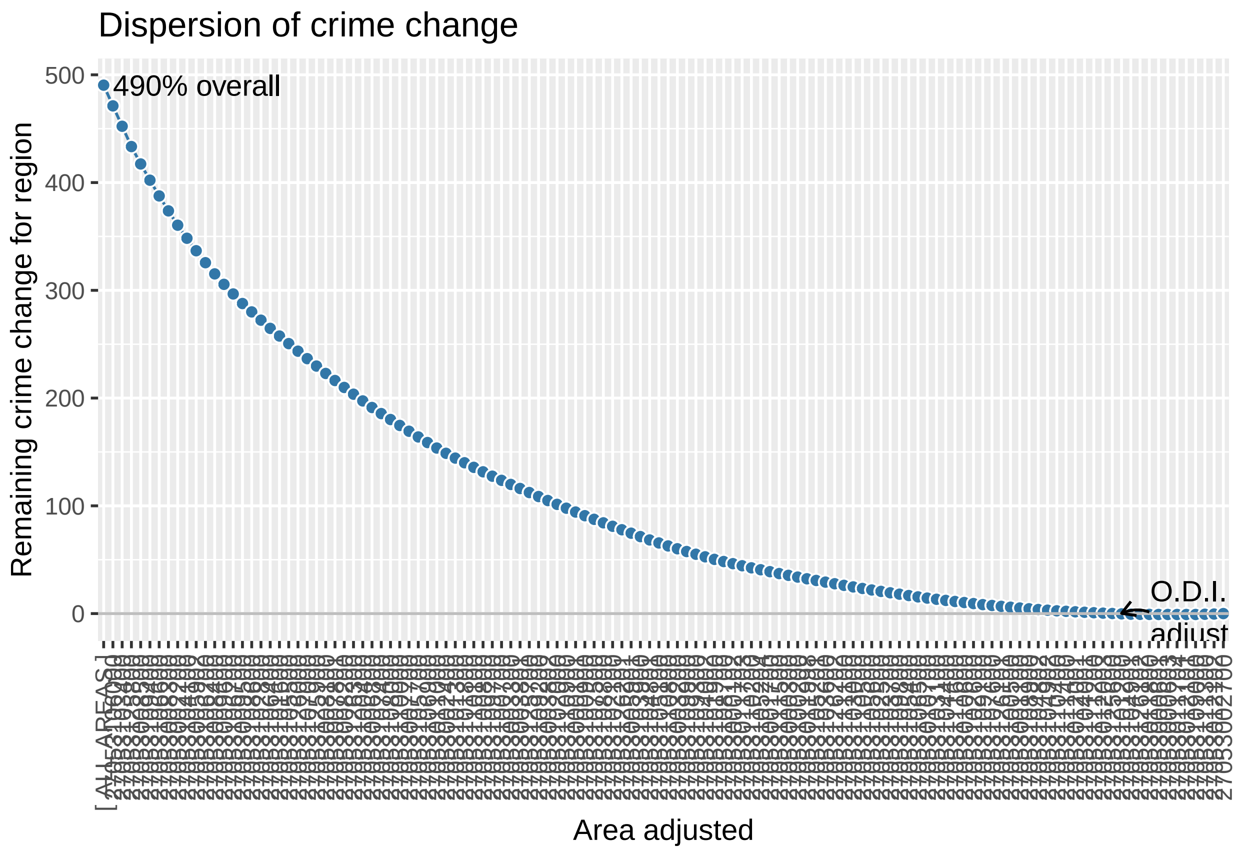
```
    chg_Rt1_Rt2 <- count_Rt2 - count_Rt1
    pct_Rt1_Rt2 <- (chg_Rt1_Rt2 / count_Rt1) *100
    named_areas <- df1$unit[row_to_remove]
  }

  # 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("2022-12-31")) %>%
  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 <- crimedispersion(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 publication/fig5.png", bg="white", width = 10, height = 8)
```

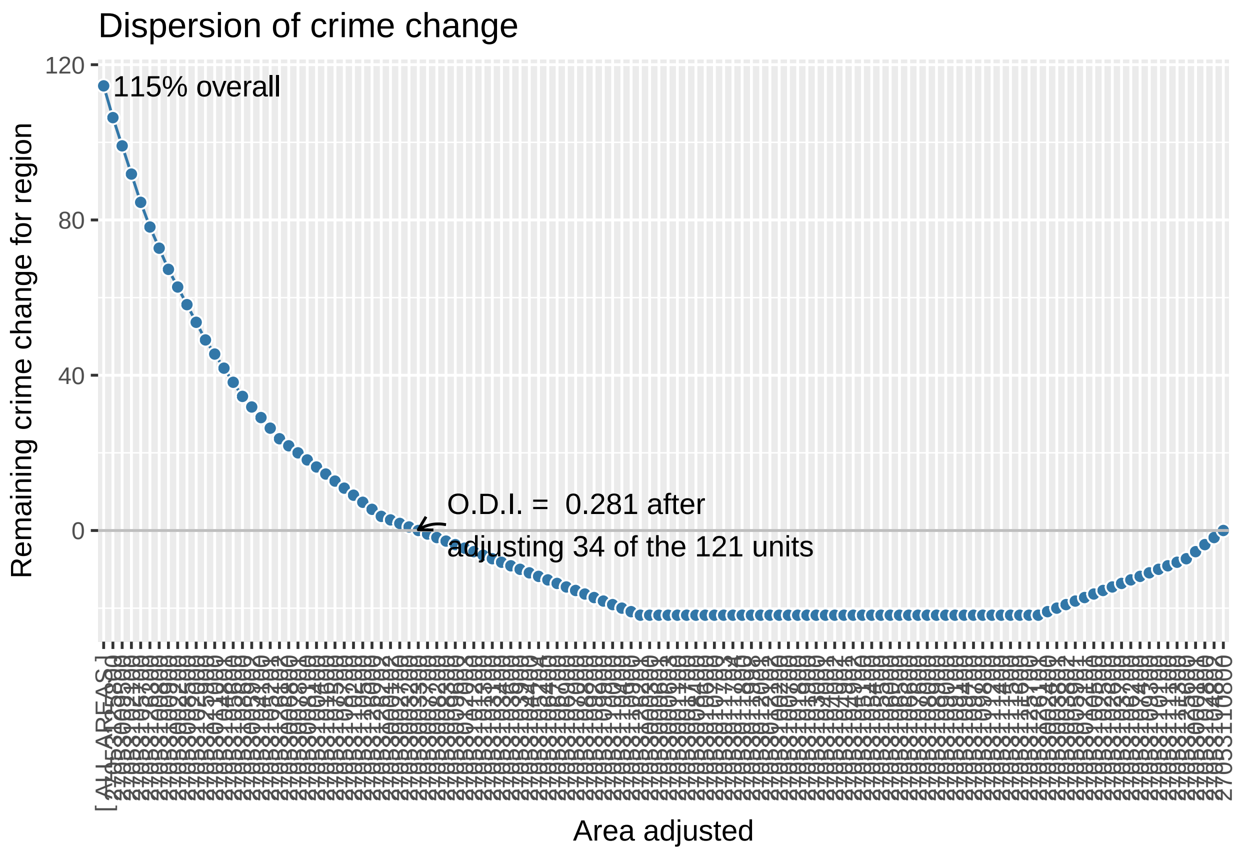
Homicide

```
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 publication/fig6.png", bg="white", width = 10, height = 8)
```

Spatial Correlation *Change* in Carjackings and Homicide

Carjacking

```
library(sfdep)
library(spdep)
```

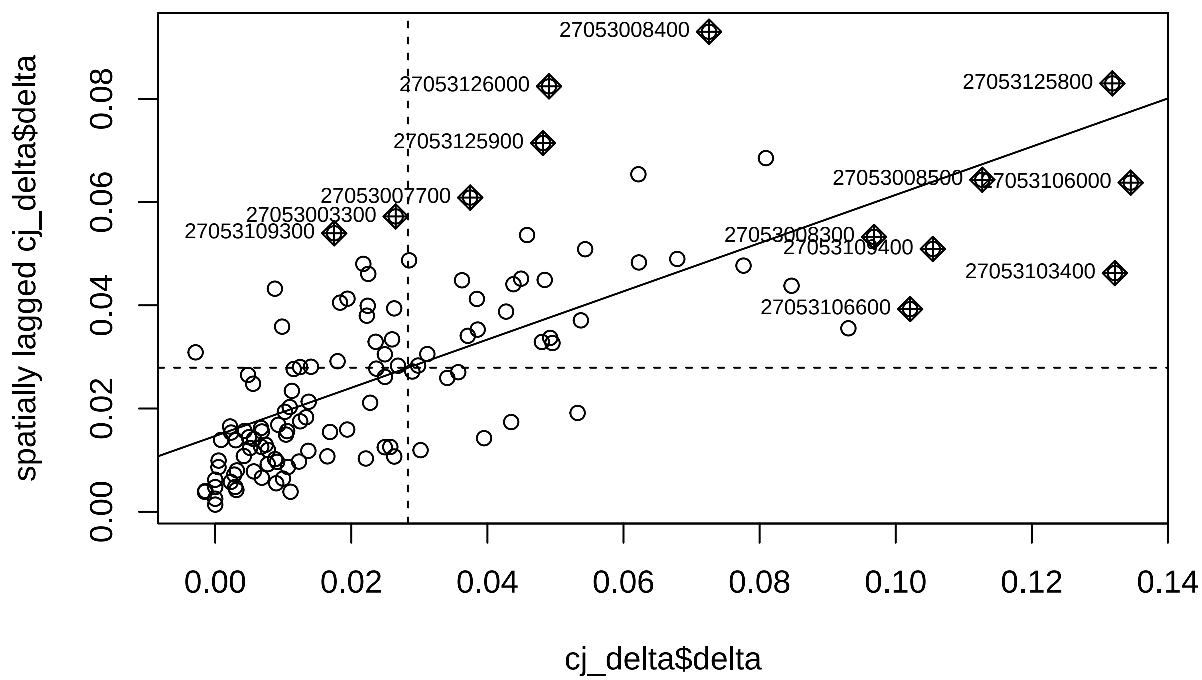
```
## Loading required package: spData

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
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()
```

```r
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)
```

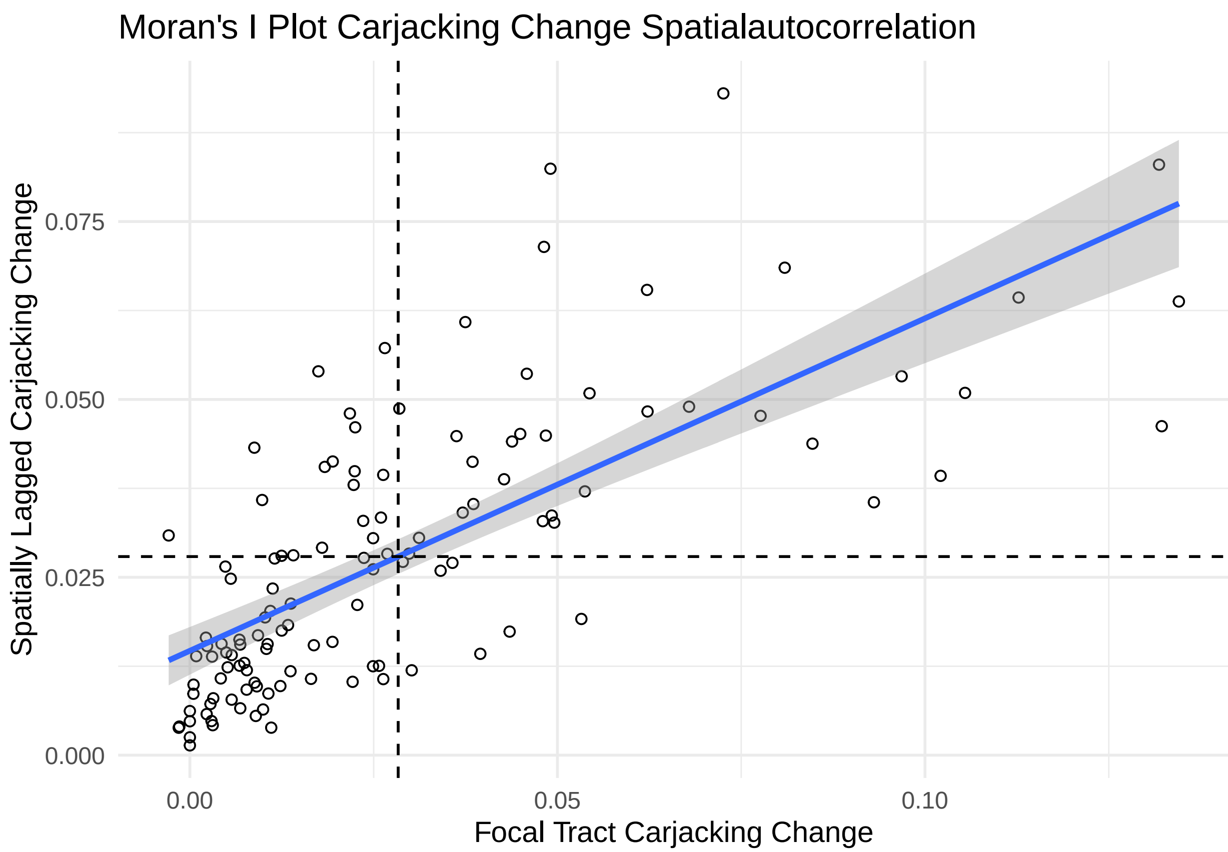
```
##
##  Moran I test under randomisation
##
## data:  x
## weights: listw
##
## Moran I statistic standard deviate = 8.9895, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.467232307      -0.008333333       0.002798653
```

```r
mp <- moran.plot(cj_delta$delta, nb2listw(nb),
                 labels = as.character(cj_delta$GEOID))
```




```r
ggplot(mp, aes(x=x, y=wx)) +
  geom_point(shape=1) +
```

```r
    geom_smooth(formula=y ~ x, method="lm") +
    geom_hline(yintercept=mean(mp$wx), lty=2) +
    #geom_hline(yintercept=0, lty=2) +
    geom_vline(xintercept=mean(mp$x), lty=2) +
    #geom_vline(xintercept=0, lty=2) +
    theme_minimal() +
    labs(title = "Moran's I Plot Carjacking Change Spatialautocorrelation",
         x = "Focal Tract Carjacking Change",
         y = "Spatially Lagged Carjacking Change")
```

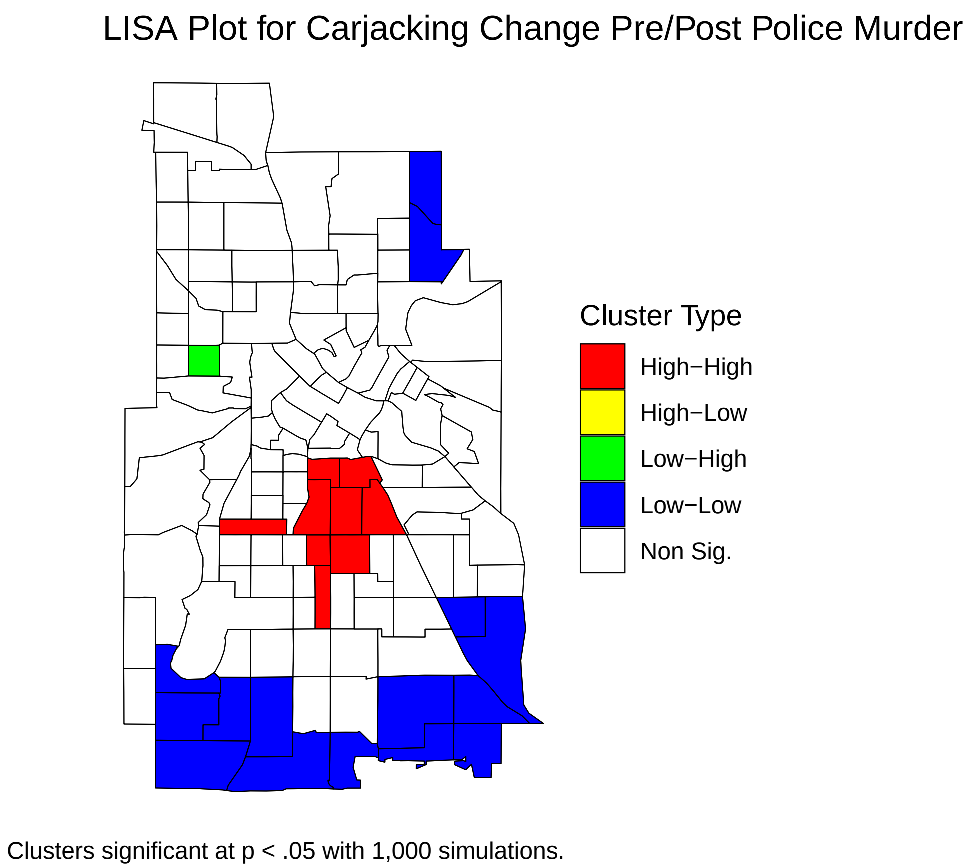



```r
global_moran_test(
  cj_delta$car_jack_rate_1,
  nb,
  wt,
  alternative = "greater",
  randomization = TRUE)
```

```
##
##  Moran I test under randomisation
##
## data:  x
## weights: listw
##
## Moran I statistic standard deviate = 9.7351, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
```

```
## Moran I statistic       Expectation          Variance
##       0.506295621      -0.008333333       0.002794527
```

```
#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_lisa_rate <-  local_moran(cj_delta$car_jack_rate_1,
                       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 = "LISA Plot for Carjacking Change Pre/Post Police Murder",
       fill = "Cluster Type",
       caption = "Clusters significant at p < .05 with 1,000 simulations.")
```




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

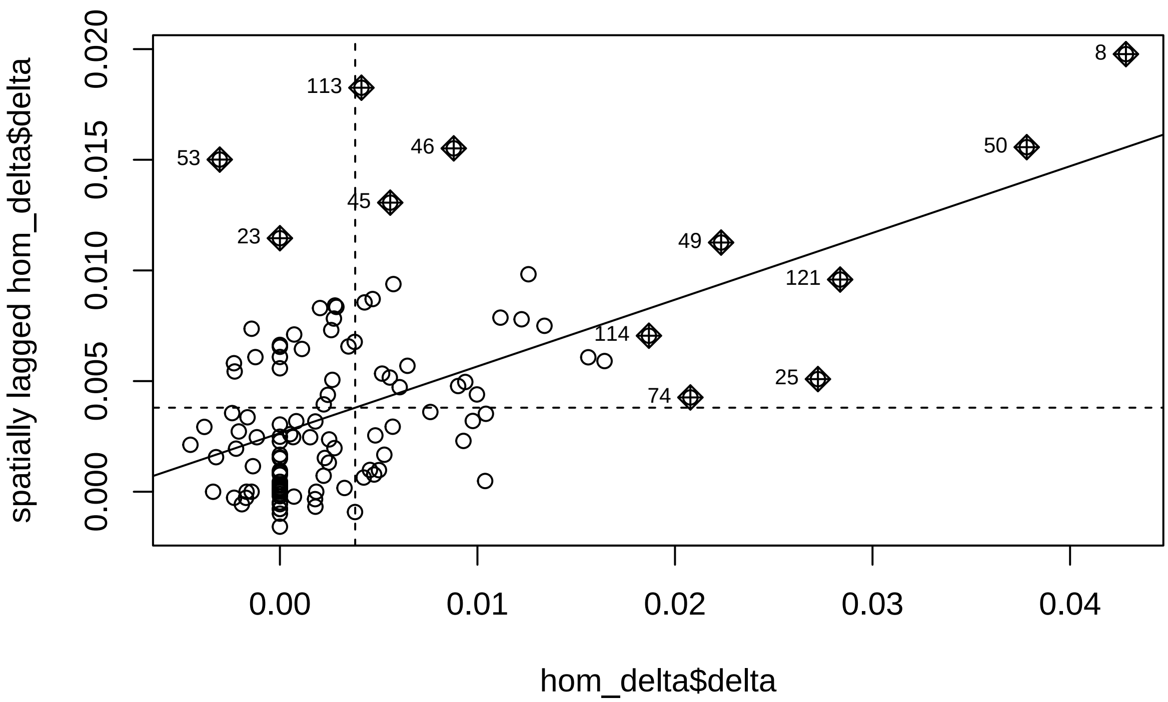
Homicide

```
hom_delta <- prepost_hom %>%
  mutate(delta = homicide_rate_1-homicide_rate_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)
```

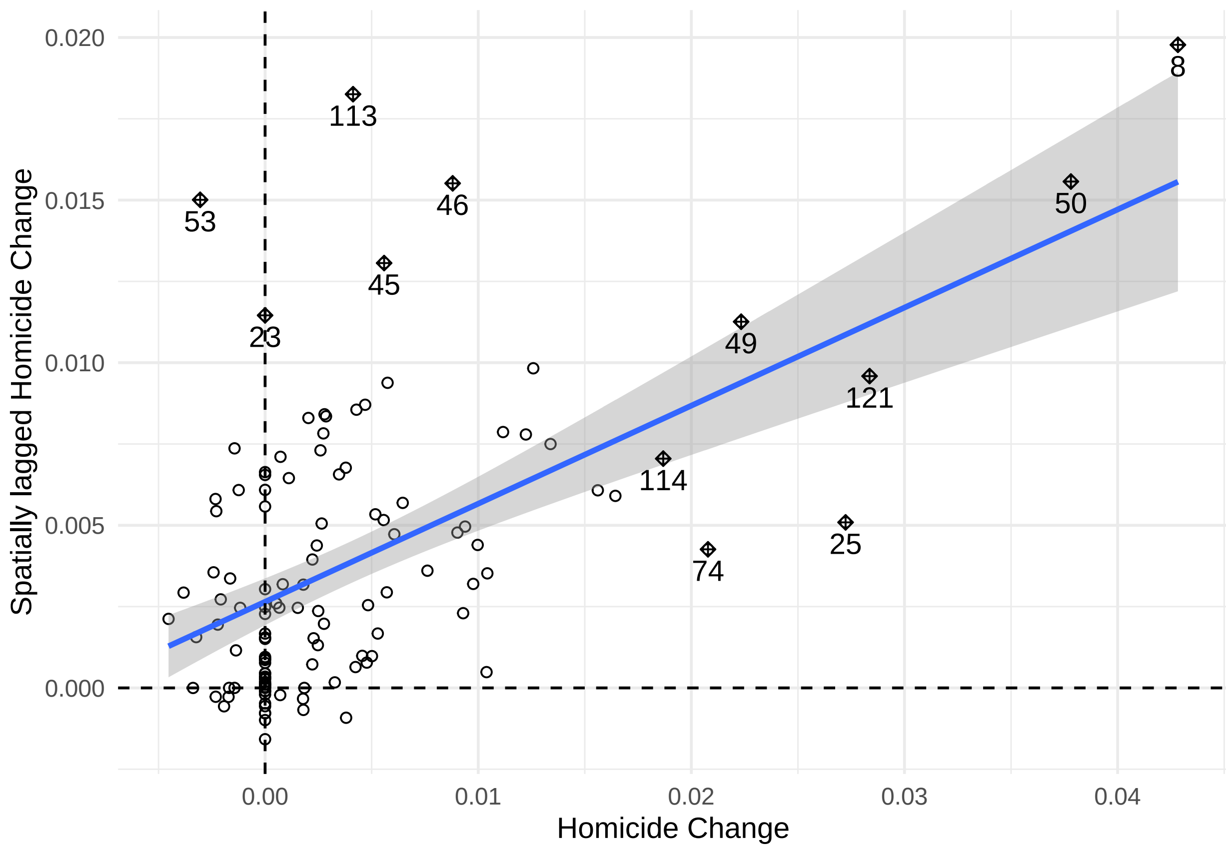
```
##
##  Moran I test under randomisation
##
## data:  x
## weights: listw
```

```
##
## Moran I statistic standard deviate = 6.0187, p-value = 8.791e-10
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.301559446      -0.008333333       0.002651046
```

```
mp <- moran.plot(hom_delta$delta, nb2listw(nb))
```




```
ggplot(mp, aes(x=x, y=wx)) +
  geom_point(shape=1) +
    geom_smooth(formula=y ~ x, method="lm") +
     #geom_hline(yintercept=mean(mp$wx), lty=2) +
    geom_hline(yintercept=0, lty=2) +
    #geom_vline(xintercept=mean(mp$x), lty=2) +
    geom_vline(xintercept=0, lty=2) +
    theme_minimal() +
    geom_point(data=mp[mp$is_inf,], aes(x=x, y=wx), shape=9) +
    geom_text(data=mp[mp$is_inf,], aes(x=x, y=wx, label=labels, vjust=1.5)) +
    xlab("Homicide Change") +
    ylab(paste0("Spatially lagged ", "Homicide Change"))
```

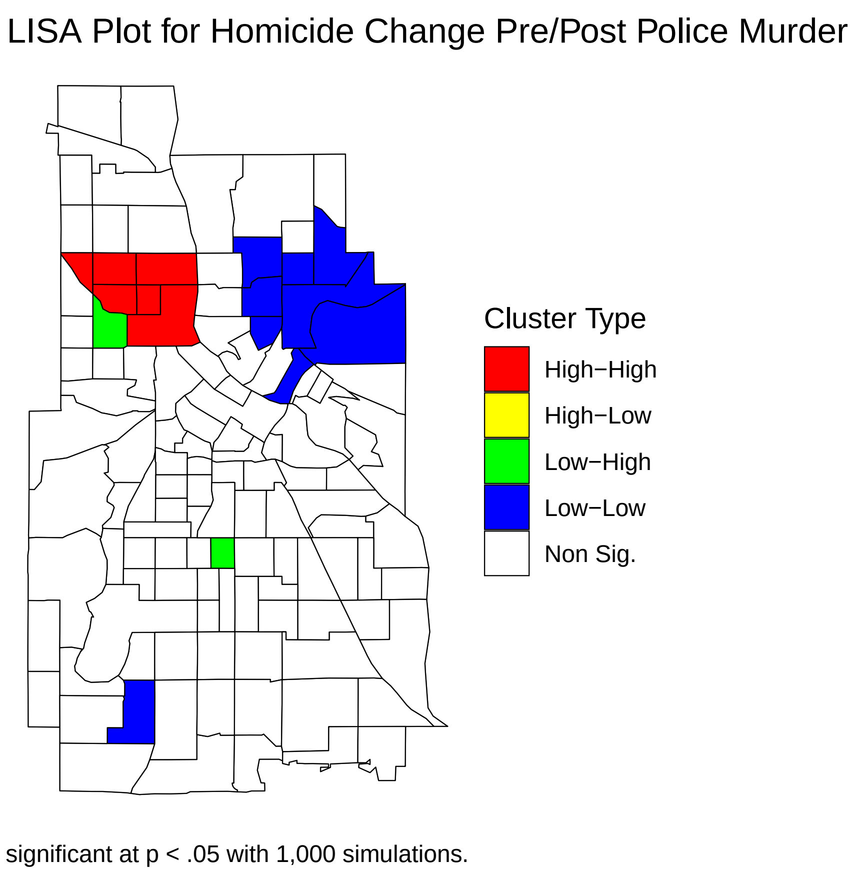



```r
global_moran_test(
  hom_delta$homicide_1,
  nb,
  wt,
  alternative = "greater",
  randomization = TRUE)
```

```
##
##  Moran I test under randomisation
##
## data:  x
## weights: listw
##
## Moran I statistic standard deviate = 6.725, p-value = 8.777e-12
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.347800619      -0.008333333       0.002804380
```

```r
#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."),
         mean_p = factor(mean_p, levels = c("High-High", "High-Low", "Low-High",
```

```r
                                           "Low-Low", "Non Sig.")))

hom_lisa_rate <-  local_moran(hom_delta$homicide_1,
                       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.")))

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"), drop = F)+
  labs(title = "LISA Plot for Homicide Change Pre/Post Police Murder",
       fill = "Cluster Type",
       caption = "Clusters significant at p < .05 with 1,000 simulations.")
```




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

RE CJ Models

```r
cj_exp_prepost <- cj_exp %>%
  group_by(GEOID) %>%
  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)),
         weeks_post = as.numeric(begin_date-as.Date("2020-05-25"))/7,
         t_post_floyd = ifelse(weeks_post >=0,
                               weeks_post,
                               0),
        # 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")) %>%
  select(-med_hh_inc) %>%
  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 943.629  7.286   7.286    0.198    0.198
## 14     unemp_rate ~~ female_hh_rate 704.567 -1.273  -1.273   -0.200   -0.200
## 15     unemp_rate ~~ no_hs_dip_rate   6.434  0.255   0.255    0.021    0.021
## 16       pov_rate ~~ female_hh_rate 443.948 -2.705  -2.705   -0.122   -0.122
## 17       pov_rate ~~ no_hs_dip_rate  58.088  1.936   1.936    0.046    0.046
## 18       pov_rate ~~     black_perc 110.356 -8.406  -8.406   -0.144   -0.144
## 19 female_hh_rate ~~ no_hs_dip_rate 105.301  0.598   0.598    0.082    0.082
## 20 female_hh_rate ~~     black_perc 719.519  5.183   5.183    0.514    0.514
## 21 no_hs_dip_rate ~~     black_perc 296.794 -7.281  -7.281   -0.382   -0.382
```

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

```
## lavaan 0.6.14 ended normally after 32 iterations
##
##   Estimator                                         ML
```

```
##   Optimization method                           NLMINB
##   Number of model parameters                        11
##
##   Number of observations                         38357
##
## Model Test User Model:
##
##   Test statistic                              1610.419
##   Degrees of freedom                                 4
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                             96458.724
##   Degrees of freedom                                10
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.983
##   Tucker-Lewis Index (TLI)                       0.958
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)            -606558.430
##   Loglikelihood unrestricted model (H1)    -605753.221
##
##   Akaike (AIC)                             1213138.861
##   Bayesian (BIC)                           1213232.962
##   Sample-size adjusted Bayesian (SABIC)    1213198.004
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.102
##   90 Percent confidence interval - lower         0.098
##   90 Percent confidence interval - upper         0.107
##   P-value H_0: RMSEA <= 0.050                    0.000
##   P-value H_0: RMSEA >= 0.080                    1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.029
##
## 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        3.428    0.023  148.576    0.000    3.428    0.726
##     pov_rate          8.295    0.067  123.883    0.000    8.295    0.590
```

```
##     female_hh_rate    2.155    0.013  164.248    0.000    2.155    0.740
##     no_hs_dip_rate    4.667    0.026  177.181    0.000    4.667    0.783
##     black_perc       17.544    0.074  236.523    0.000   17.544    0.960
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .unemp_rate ~~
##    .black_perc       -9.264    0.237  -39.009    0.000   -9.264   -0.555
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .unemp_rate       10.512    0.103  101.640    0.000   10.512    0.472
##    .pov_rate        128.929    0.964  133.780    0.000  128.929    0.652
##    .female_hh_rate    3.844    0.031  122.692    0.000    3.844    0.453
##    .no_hs_dip_rate   13.746    0.120  114.796    0.000   13.746    0.387
##    .black_perc       26.463    1.011   26.187    0.000   26.463    0.079
##     cd                1.000                               1.000    1.000
```

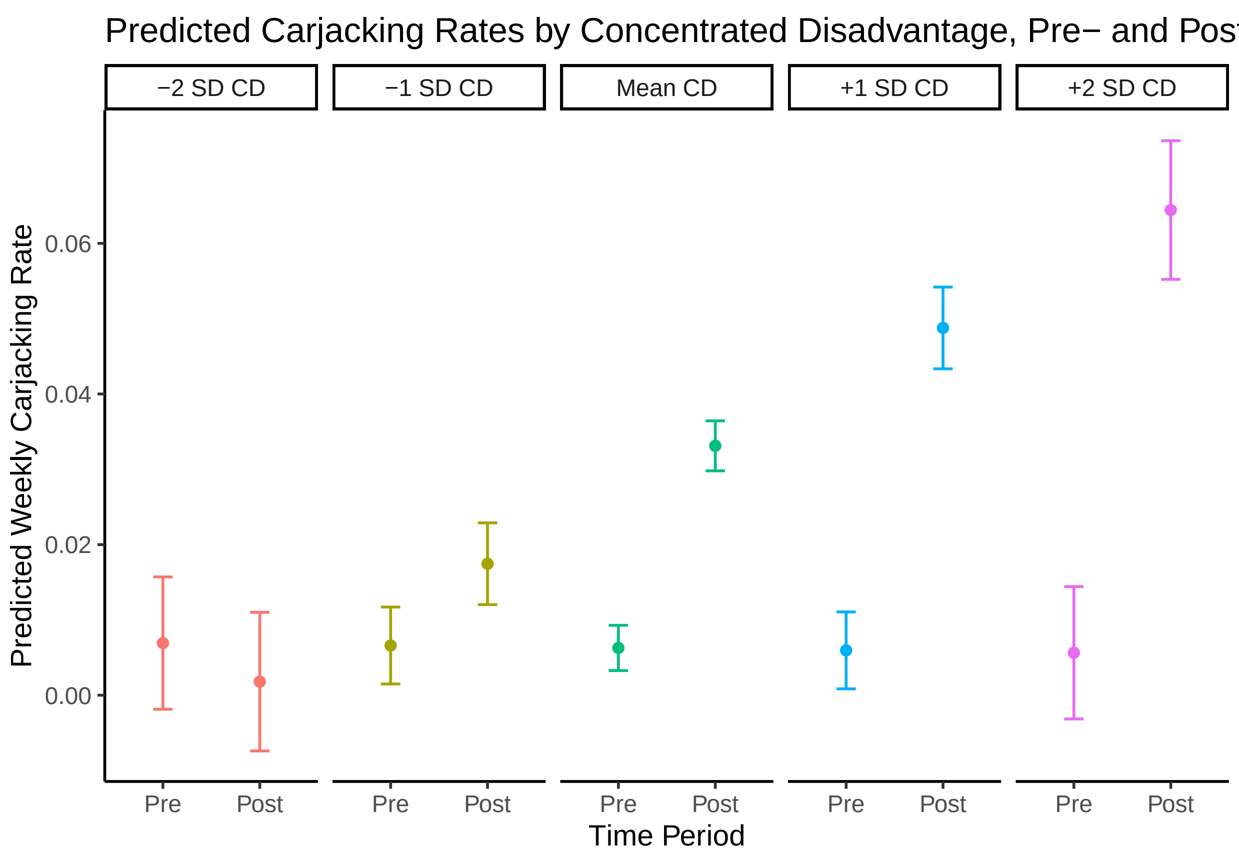
```
cd_predict <- as.vector(lavPredict(cfa_cd, newdata = as.data.frame(cj_exp_prepost)))
cj_exp_prepost$conc_dis <- cd_predict

#predicted probability/rate plots
  #over time
  #stratified by CD (mean, +1SD, -1SD, +2SD, -2SD)


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

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## car_jack_rate ~ t + post_floyd + t_post_floyd + conc_dis + age_19_29_perc +
##     age_30_49_perc + age_50_69_perc + age_70_plus_perc + post_floyd:conc_dis +
##     t_post_floyd:conc_dis + (1 | GEOID)
##    Data: cj_exp_prepost
##
## REML criterion at convergence: -71411.7
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -1.255 -0.254 -0.120  0.018 31.639
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  GEOID    (Intercept) 0.0001701 0.01304
##  Residual             0.0090060 0.09490
## Number of obs: 38357, groups:  GEOID, 121
##
## Fixed effects:
```

```
##                         Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)            5.038e-02  1.649e-02  1.162e+02   3.054  0.00280 **
## t                      1.490e-05  1.238e-05  3.823e+04   1.204  0.22858
## post_floyd             2.684e-02  1.954e-03  3.823e+04  13.736  < 2e-16 ***
## t_post_floyd          -1.775e-05  2.237e-05  3.823e+04  -0.794  0.42745
## conc_dis               4.779e-04  2.031e-03  1.261e+02   0.235  0.81436
## age_19_29_perc        -4.629e-04  1.690e-04  1.150e+02  -2.738  0.00716 **
## age_30_49_perc        -3.520e-04  2.709e-04  1.150e+02  -1.299  0.19652
## age_50_69_perc        -1.033e-03  3.113e-04  1.150e+02  -3.318  0.00121 **
## age_70_plus_perc      -6.118e-04  3.773e-04  1.150e+02  -1.622  0.10760
## post_floyd:conc_dis    1.598e-02  1.630e-03  3.823e+04   9.805  < 2e-16 ***
## t_post_floyd:conc_dis -2.725e-05  1.897e-05  3.823e+04  -1.436  0.15097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) t      pst_fl t_pst_ cnc_ds a_19_2 a_30_4 a_50_6 a_70__
## t           -0.068
## post_floyd   0.026 -0.573
## t_post_flyd  0.038 -0.553 -0.223
## conc_dis    -0.724  0.000  0.000  0.000
## ag_19_29_pr -0.936  0.000  0.000  0.000  0.695
## ag_30_49_pr -0.884  0.000  0.000  0.000  0.597  0.767
## ag_50_69_pr -0.749  0.000  0.000  0.000  0.618  0.724  0.435
## ag_70_pls_p -0.300  0.000  0.000  0.000  0.190  0.279  0.276 -0.127
## pst_flyd:c_  0.000  0.000  0.000  0.000 -0.129  0.000  0.000  0.000  0.000
## t_pst_fly:_  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
##             pst_:_
## t
## post_floyd
## t_post_flyd
## conc_dis
## ag_19_29_pr
## ag_30_49_pr
## ag_50_69_pr
## ag_70_pls_p
## pst_flyd:c_
## t_pst_fly:_ -0.792
```

```r
library(emmeans)

pred_raw <- emmeans(re, c("conc_dis", "post_floyd"),
                    at = list(conc_dis = c(-2, -1, 0, 1, 2),
                              post_floyd = c(0,1))) %>%
  as.data.frame() %>%
  mutate(conc_dis = factor(case_when(
    conc_dis==-2~"-2 SD CD",
    conc_dis==-1~"-1 SD CD",
    conc_dis==0~"Mean CD",
    conc_dis==1~"+1 SD CD",
    conc_dis==2~"+2 SD CD"),
    levels = c("-2 SD CD",
               "-1 SD CD",
               "Mean CD",
```

```
                "+1 SD CD",
                "+2 SD CD")),
    post_floyd = factor(case_when(
    post_floyd==1~"Post",
    post_floyd==0~"Pre"),
    levels = c("Pre", "Post")))

ggplot(pred_raw, aes(x = post_floyd, y = emmean, color = conc_dis))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin=asymp.LCL, ymax=asymp.UCL), width=0.2)+
  labs(title = "Predicted Carjacking Rates by Concentrated Disadvantage, Pre- and Post-Killing",
       y = "Predicted Weekly Carjacking Rate",
       x = "Time Period")+
  theme_classic()+
  facet_grid(~conc_dis)+
  guides(color = "none")
```




```
ggsave(filename = "Car Jacking/Figures for publication/fig9.png", bg="white", width = 10, height = 8)



re_logit_cd <- glmer(anyjack ~ t+post_floyd+t_post_floyd+conc_dis+
             age_19_29_perc+age_30_49_perc+age_50_69_perc+
             age_70_plus_perc+ post_floyd:conc_dis+t_post_floyd:conc_dis+
             (1|GEOID),
```

```r
                 data = cj_exp_prepost, family = binomial)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00861892 (tol = 0.002, component 1)
```

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

```r
summary(re_logit_cd)
```

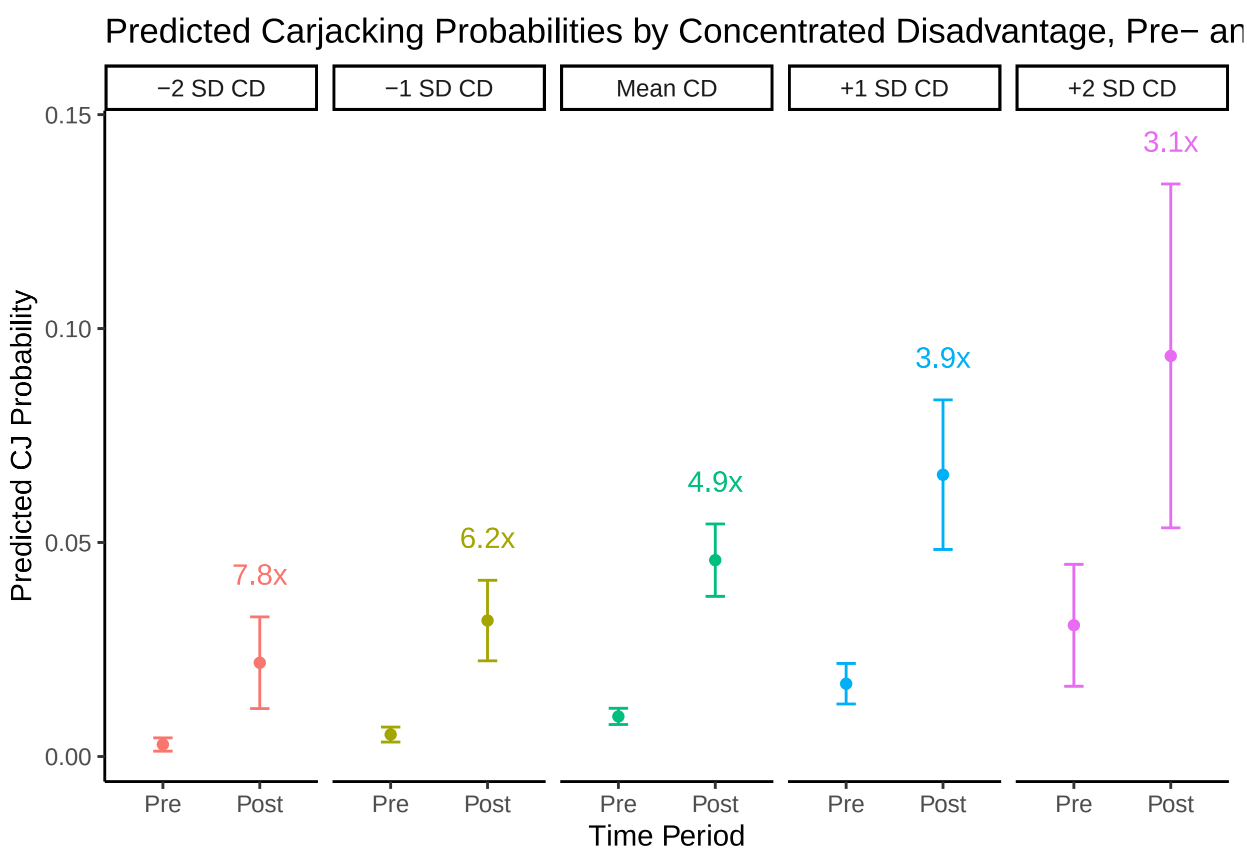
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: anyjack ~ t + post_floyd + t_post_floyd + conc_dis + age_19_29_perc +
##     age_30_49_perc + age_50_69_perc + age_70_plus_perc + post_floyd:conc_dis +
##     t_post_floyd:conc_dis + (1 | GEOID)
##    Data: cj_exp_prepost
##
##      AIC      BIC   logLik deviance df.resid
##  11182.8  11285.4  -5579.4  11158.8    38345
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7196 -0.2197 -0.1384 -0.0726 21.5221
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  GEOID  (Intercept) 0.4674   0.6837
## Number of obs: 38357, groups:  GEOID, 121
##
## Fixed effects:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -4.4947951  0.9185632  -4.893 9.92e-07 ***
## t                      0.0038399  0.0011363   3.379 0.000727 ***
## post_floyd             1.6256029  0.1310456  12.405  < 2e-16 ***
## t_post_floyd          -0.0031597  0.0013849  -2.281 0.022519 *
## conc_dis               0.6378147  0.1190326   5.358 8.40e-08 ***
## age_19_29_perc        -0.0028888  0.0093469  -0.309 0.757271
## age_30_49_perc         0.0149162  0.0149808   0.996 0.319402
## age_50_69_perc        -0.0491073  0.0173871  -2.824 0.004738 **
## age_70_plus_perc      -0.0147769  0.0208016  -0.710 0.477472
## post_floyd:conc_dis   -0.2215576  0.0776478  -2.853 0.004326 **
## t_post_floyd:conc_dis -0.0011664  0.0007388  -1.579 0.114378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) t      pst_fl t_pst_ cnc_ds a_19_2 a_30_4 a_50_6 a_70__
## t           -0.124
## post_floyd   0.048 -0.699
## t_post_flyd  0.101 -0.820  0.335
## conc_dis    -0.714  0.002  0.122  0.000
## ag_19_29_pr -0.928  0.000 -0.002  0.001  0.675
## ag_30_49_pr -0.878  0.000 -0.001  0.001  0.584  0.766
```

```
## ag_50_69_pr -0.739  0.000 -0.002  0.000  0.581  0.720  0.431
## ag_70_pls_p -0.278  0.000 -0.005 -0.001  0.167  0.262  0.264 -0.147
## pst_flyd:c_  0.026 -0.003 -0.303  0.119 -0.298  0.001  0.001  0.000  0.011
## t_pst_fly:_  0.001  0.000  0.127 -0.172 -0.006 -0.001 -0.001 -0.001  0.004
##             pst_:_
## t
## post_floyd
## t_post_flyd
## conc_dis
## ag_19_29_pr
## ag_30_49_pr
## ag_50_69_pr
## ag_70_pls_p
## pst_flyd:c_
## t_pst_fly:_ -0.631
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00861892 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
## Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

```
pred_prob_raw <- emmeans(re_logit_cd, c("conc_dis", "post_floyd"),
                    at = list(conc_dis = c(-2, -1, 0, 1, 2),
                              post_floyd = c(0,1)),
                    trans = "response") %>%
  as.data.frame() %>%
  mutate(conc_dis = factor(case_when(
    conc_dis==-2~"-2 SD CD",
    conc_dis==-1~"-1 SD CD",
    conc_dis==0~"Mean CD",
    conc_dis==1~"+1 SD CD",
    conc_dis==2~"+2 SD CD"),
    levels = c("-2 SD CD",
               "-1 SD CD",
               "Mean CD",
               "+1 SD CD",
               "+2 SD CD")),
    post_floyd = factor(case_when(
    post_floyd==1~"Post",
    post_floyd==0~"Pre"),
    levels = c("Pre", "Post")))

ratios <- pred_prob_raw %>%
  select(conc_dis, post_floyd, prob) %>%
  pivot_wider(names_from = post_floyd, values_from = prob) %>%
  mutate(ratio = Post/Pre)

pred_prob_raw <- pred_prob_raw %>%
  left_join(ratios, by = "conc_dis") %>%
  mutate(ratio = ifelse(post_floyd=="Pre", NA_integer_, round(ratio,1)),
         ratio = ifelse(is.na(ratio), ratio, paste0(ratio, "x")))

ggplot(pred_prob_raw, aes(x = post_floyd, y = prob, color = conc_dis))+
```

```
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin=asymp.LCL, ymax=asymp.UCL), width=0.2)+
  labs(title = "Predicted Carjacking Probabilities by Concentrated Disadvantage, Pre- and Post-Killing"
       y = "Predicted CJ Probability",
       x = "Time Period")+
  theme_classic()+
  facet_grid(~conc_dis)+
  guides(color = "none")+
  geom_text(aes(x = post_floyd, y = asymp.UCL+.01, label = ratio))
```

```
## Warning: Removed 5 rows containing missing values (`geom_text()`).
```




```
ggsave(filename = "Car Jacking/Figures for publication/fig10.png", bg="white", width = 10, height = 8)
```

```
## Warning: Removed 5 rows containing missing values (`geom_text()`).
```

```
homicide <- homicide %>% st_drop_geometry()

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+t_post_floyd+conc_dis+
             age_19_29_perc+age_30_49_perc+age_50_69_perc+
             age_70_plus_perc+ post_floyd:conc_dis+t_post_floyd:conc_dis+
```

```r
            (1|GEOID),
          data = cj_exp_prepost)
summary(re_homicide)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## homicide_rate ~ t + post_floyd + t_post_floyd + conc_dis + age_19_29_perc +
##     age_30_49_perc + age_50_69_perc + age_70_plus_perc + post_floyd:conc_dis +
##     t_post_floyd:conc_dis + (1 | GEOID)
##    Data: cj_exp_prepost
##
## REML criterion at convergence: -131581.4
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -1.095 -0.108 -0.033 -0.007 44.509
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  GEOID    (Intercept) 1.527e-05 0.003907
##  Residual             1.879e-03 0.043348
## Number of obs: 38357, groups:  GEOID, 121
##
## Fixed effects:
##                         Estimate Std. Error         df t value Pr(>|t|)
## (Intercept)            7.689e-03  5.403e-03  1.174e+02   1.423  0.15735
## t                      1.165e-05  5.653e-06  3.823e+04   2.061  0.03929 *
## post_floyd             4.206e-03  8.925e-04  3.823e+04   4.713 2.45e-06 ***
## t_post_floyd          -3.297e-05  1.022e-05  3.823e+04  -3.226  0.00126 **
## conc_dis               1.681e-03  6.778e-04  1.371e+02   2.480  0.01434 *
## age_19_29_perc        -5.925e-05  5.524e-05  1.150e+02  -1.073  0.28567
## age_30_49_perc        -1.355e-04  8.853e-05  1.150e+02  -1.530  0.12871
## age_50_69_perc         2.834e-05  1.017e-04  1.150e+02   0.279  0.78100
## age_70_plus_perc      -2.832e-04  1.233e-04  1.150e+02  -2.297  0.02343 *
## post_floyd:conc_dis    6.338e-03  7.444e-04  3.823e+04   8.514  < 2e-16 ***
## t_post_floyd:conc_dis -3.527e-05  8.666e-06  3.823e+04  -4.070 4.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) t      pst_fl t_pst_ cnc_ds a_19_2 a_30_4 a_50_6 a_70__
## t           -0.095
## post_floyd   0.036 -0.573
## t_post_flyd  0.052 -0.553 -0.223
## conc_dis    -0.707  0.000  0.000  0.000
## ag_19_29_pr -0.934  0.000  0.000  0.000  0.681
## ag_30_49_pr -0.881  0.000  0.000  0.000  0.585  0.767
## ag_50_69_pr -0.747  0.000  0.000  0.000  0.605  0.724  0.435
## ag_70_pls_p -0.299  0.000  0.000  0.000  0.186  0.279  0.276 -0.127
## pst_flyd:c_  0.000  0.000  0.000  0.000 -0.177  0.000  0.000  0.000  0.000
## t_pst_fly:_  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
##             pst_:_
## t
```

```
## post_floyd
## t_post_flyd
## conc_dis
## ag_19_29_pr
## ag_30_49_pr
## ag_50_69_pr
## ag_70_pls_p
## pst_flyd:c_
## t_pst_fly:_ -0.792
```

```
re_logit_cd_homicide <- glmer(anyhom ~ t+post_floyd+t_post_floyd+conc_dis+
             age_19_29_perc+age_30_49_perc+age_50_69_perc+
             age_70_plus_perc+ post_floyd:conc_dis+t_post_floyd:conc_dis+
             (1|GEOID),
                  data = cj_exp_prepost, family = binomial)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.267118 (tol = 0.002, component 1)

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

```
summary(re_logit_cd_homicide)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: anyhom ~ t + post_floyd + t_post_floyd + conc_dis + age_19_29_perc +
##     age_30_49_perc + age_50_69_perc + age_70_plus_perc + post_floyd:conc_dis +
##     t_post_floyd:conc_dis + (1 | GEOID)
##    Data: cj_exp_prepost
##
##      AIC      BIC   logLik deviance df.resid
##   3457.5   3560.2  -1716.8   3433.5    38345
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -0.338 -0.092 -0.061 -0.042 35.080
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  GEOID  (Intercept) 0.6719   0.8197
## Number of obs: 38357, groups:  GEOID, 121
##
## Fixed effects:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -7.959083   1.318054  -6.039 1.56e-09 ***
## t                      0.004882   0.001890   2.583  0.00981 **
## post_floyd             0.777797   0.245234   3.172  0.00152 **
## t_post_floyd          -0.007360   0.002811  -2.618  0.00884 **
## conc_dis               0.957280   0.177431   5.395 6.84e-08 ***
## age_19_29_perc         0.019009   0.013565   1.401  0.16112
## age_30_49_perc         0.027486   0.021594   1.273  0.20308
## age_50_69_perc         0.026644   0.025918   1.028  0.30394
## age_70_plus_perc      -0.047473   0.032278  -1.471  0.14136
```

```
## post_floyd:conc_dis    0.162234   0.137858   1.177  0.23927
## t_post_floyd:conc_dis -0.002923   0.001564  -1.869  0.06164 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) t      pst_fl t_pst_ cnc_ds a_19_2 a_30_4 a_50_6 a_70__
## t           -0.148
## post_floyd   0.044 -0.599
## t_post_flyd  0.099 -0.672 -0.001
## conc_dis    -0.738  0.001  0.129  0.007
## ag_19_29_pr -0.913 -0.001  0.002 -0.001  0.689
## ag_30_49_pr -0.851 -0.001  0.002 -0.001  0.579  0.724
## ag_50_69_pr -0.693 -0.001  0.001  0.000  0.563  0.666  0.335
## ag_70_pls_p -0.256  0.000  0.004 -0.006  0.134  0.241  0.252 -0.202
## pst_flyd:c_  0.033 -0.001 -0.468  0.295 -0.289 -0.003 -0.003 -0.001 -0.009
## t_pst_fly:_  0.000  0.000  0.297 -0.404 -0.018  0.003  0.002  0.000  0.014
##             pst_:_
## t
## post_floyd
## t_post_flyd
## conc_dis
## ag_19_29_pr
## ag_30_49_pr
## ag_50_69_pr
## ag_70_pls_p
## pst_flyd:c_
## t_pst_fly:_ -0.652
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.267118 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
## Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

```r
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", "T Post-Killing",
                               "Conc. Dis.", "Age 19-29", "Age 30-49",
                               "Age 50-69", "Age 70+",
                               "Post-Killing X Conc. Dis.",
                               "T Post-Killing 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)", .013, .674),
                           c("SD(Residual)",  .094, "")),
          notes.append = F)
```

Spatial Panel Models

```
library(splm)
library(plm)
```

```
##
## Attaching package: 'plm'

## The following object is masked from 'package:lavaan':
##
##     nobs

## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
nb <- st_contiguity(mpls_tract, queen=TRUE)
wt <-  st_weights(nb, style = "W", allow_zero = T)
w <- recreate_listw(nb, wt)

cj_exp_prepost <- cj_exp_prepost %>%
  group_by(GEOID) %>%
  arrange(year, week) %>%
  mutate(WEEKID = row_number())

#write_csv(cj_exp_prepost, "Car Jacking/cj_exp_prepost.csv")

cj_exp_prepost_panel <- pdata.frame(cj_exp_prepost, index = c("GEOID", "WEEKID"), drop.index = F)

slmtest(car_jack_rate~t+post_floyd+t_post_floyd+
             conc_dis+
             age_19_29_perc+age_30_49_perc+age_50_69_perc+
             age_70_plus_perc+ post_floyd:conc_dis+t_post_floyd:conc_dis,
        data=cj_exp_prepost_panel,
        listw =w,
        model = "random",
        test = "lml")
```

```
##
##  LM test for spatial lag dependence
##
## data:  formula (random transformation)
## LM = 1.0311, df = 1, p-value = 0.3099
## alternative hypothesis: spatial lag dependence
```

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

	Carjacking	
	Rate per 1,000 RE HLM	Any Carjacking RE Logit
	(1)	(2)
T	0.00001 (−0.00001|0.00004)	0.004 (0.002|0.006)
Post-Killing	0.027 (0.023|0.031)	1.626 (1.369|1.882)
T Post-Killing	−0.00002 (−0.0001|0.00003)	−0.003 (−0.006|−0.0004)
Conc. Dis.	0.0005 (−0.004|0.004)	0.638 (0.405|0.871)
Age 19-29	−0.0005 (−0.001|−0.0001)	−0.003 (−0.021|0.015)
Age 30-49	−0.0004 (−0.001|0.0002)	0.015 (−0.014|0.044)
Age 50-69	−0.001 (−0.002|−0.0004)	−0.049 (−0.083|−0.015)
Age 70+	−0.001 (−0.001|0.0001)	−0.015 (−0.056|0.026)
Post-Killing X Conc. Dis.	0.016 (0.013|0.019)	−0.222 (−0.374|−0.069)
T Post-Killing X Conc. Dis.	−0.00003 (−0.0001|0.00001)	−0.001 (−0.003|0.0003)
Constant	0.050 (0.018|0.083)	−4.495 (−6.295|−2.694)
SD(Tract)	0.013	0.674
SD(Residual)	0.094	
Observations	38,357	38,357
Log Likelihood	35,705.860	−5,579.381
Note:	95% Confidence Intervals in parentheses	

```r
#SAR Linear CJ
sar <- spml(car_jack_rate~t+post_floyd+t_post_floyd+
             conc_dis+
             age_19_29_perc+age_30_49_perc+age_50_69_perc+
             age_70_plus_perc+ post_floyd:conc_dis+t_post_floyd:conc_dis,
                      data = cj_exp_prepost,
             index = c("GEOID"),
            effect="individual",
             model="random",
             listw = w,
            lag=T,
            spatial.error="none")
summary(sar)
```

```
## Warning in sqrt(diag(object$vcov.arcoef)): NaNs produced
```

```
## ML panel with spatial lag, random effects
##
## Call:
## spreml(formula = formula, data = data, index = index, w = listw2mat(listw),
##     w2 = listw2mat(listw2), lag = lag, errors = errors, cl = cl)
##
## Residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.12788 -0.02415 -0.01091  0.00015 -0.00335  3.04165
##
## Error variance parameters:
##      Estimate Std. Error t-value  Pr(>|t|)
## phi 0.0178066  0.0028774  6.1884 6.079e-10 ***
##
## Spatial autoregressive coefficient:
##         Estimate Std. Error t-value Pr(>|t|)
## lambda 0.0084483        NaN     NaN      NaN
##
## Coefficients:
##                          Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)            5.0394e-02  1.6084e-02  3.1332 0.0017290 **
## t                      1.4758e-05  1.2374e-05  1.1926 0.2330308
## post_floyd             2.6618e-02  1.9536e-03 13.6247 < 2.2e-16 ***
## t_post_floyd          -1.7571e-05  2.2370e-05 -0.7855 0.4321814
## conc_dis               4.7086e-04  1.9828e-03  0.2375 0.8122918
## age_19_29_perc        -4.6335e-04  1.6484e-04 -2.8110 0.0049392 **
## age_30_49_perc        -3.5226e-04  2.6419e-04 -1.3334 0.1824035
## age_50_69_perc        -1.0343e-03  3.0353e-04 -3.4076 0.0006554 ***
## age_70_plus_perc      -6.1286e-04  3.6786e-04 -1.6660 0.0957136 .
## post_floyd:conc_dis    1.5984e-02  1.6296e-03  9.8089 < 2.2e-16 ***
## t_post_floyd:conc_dis -2.7238e-05  1.8971e-05 -1.4358 0.1510713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#SAR LPM CJ
sar_lpm <- spml(anyjack~t+post_floyd+t_post_floyd+
             conc_dis+
             age_19_29_perc+age_30_49_perc+age_50_69_perc+
             age_70_plus_perc+ post_floyd:conc_dis+t_post_floyd:conc_dis,
```

```
                      data = cj_exp_prepost,
             index = c("GEOID"),
            effect="individual",
             model="random",
             listw = w,
            lag=T,
            spatial.error="none")
summary(sar_lpm)
```

```
## ML panel with spatial lag, random effects
##
## Call:
## spreml(formula = formula, data = data, index = index, w = listw2mat(listw),
##     w2 = listw2mat(listw2), lag = lag, errors = errors, cl = cl)
##
## Residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.19665 -0.06206 -0.01820  0.00511 -0.00435  1.01439
##
## Error variance parameters:
##      Estimate Std. Error t-value  Pr(>|t|)
## phi 0.0209545  0.0029535  7.0948 1.296e-12 ***
##
## Spatial autoregressive coefficient:
##          Estimate Std. Error t-value Pr(>|t|)
## lambda -0.0046216  0.0014756  -3.132 0.001736 **
##
## Coefficients:
##                          Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)           -6.0659e-02  3.4825e-02 -1.7418  0.081542 .
## t                      1.9325e-05  2.4992e-05  0.7733  0.439372
## post_floyd             6.0843e-02  3.9457e-03 15.4199 < 2.2e-16 ***
## t_post_floyd           3.9053e-07  4.5181e-05  0.0086  0.993103
## conc_dis               3.7355e-03  4.2816e-03  0.8724  0.382964
## age_19_29_perc         6.7027e-04  3.5704e-04  1.8773  0.060479 .
## age_30_49_perc         1.8523e-03  5.7223e-04  3.2370  0.001208 **
## age_50_69_perc        -3.6193e-04  6.5745e-04 -0.5505  0.581973
## age_70_plus_perc       3.5468e-04  7.9680e-04  0.4451  0.656225
## post_floyd:conc_dis    2.5927e-02  3.2912e-03  7.8775 3.339e-15 ***
## t_post_floyd:conc_dis -7.7275e-05  3.8316e-05 -2.0168  0.043717 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```