ACS and Google Trends (2010-2019) Cross-Sectional Data

Yu-Hsuan Liu

March 2nd, 2021

```
library(dplyr)
library(xtable)
library(prettyR)
library(ggplot2)
library(tidyr)
library(glue)
library(gtable)
library(grid)
library(gridExtra)
library(stargazer)
library(haven)
library(reshape2)
library(MatchIt)
library(maps)
library(plm)
library(multiwayvcov)
library(lmtest)
library(readxl)
library(janitor)
library(hablar)
library(psych)
library(psycho)
library(tidyverse)
library(geojsonio)
library(geojsonsf)
library(sp)
library(RColorBrewer)
library(rnaturalearth)
library(rnaturalearthdata)
library(sf)
library(broom)
library(mapproj)
library(ggpubr)
library(viridis)
library(pastecs)
library(kableExtra)
library(knitr)
library(MASS)



eval = TRUE
```

```r
options(scipen = 999)
options(xtable.comment = FALSE)
knitr::opts_chunk$set(cache = TRUE)
theme_set(theme_bw())
```

```r
## We dont use 2005-2009 data because there is no internet and vehicle data!

# read ACS 2005-2009
acs_05_09 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2005_2009.csv")

# read ACS 2006-2010
acs_06_10 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2006_2010.csv")

# read ACS 2007-2011
acs_07_11 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2007_2011.csv")

# read ACS 2008-2012
acs_08_12 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2008_2012.csv")

# read ACS 2009-2013
acs_09_13 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2009_2013.csv")

# read ACS 2010-2014
acs_10_14 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2010_2014.csv")

# read ACS 2011-2015
acs_11_15 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2011_2015.csv")

# read ACS 2012-2016
acs_12_16 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2012_2016.csv")

# read ACS 2013-2017
acs_13_17 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2013_2017.csv")

# read ACS 2014-2018
acs_14_18 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2014_2018.csv")

# read ACS 2015-2019
acs_15_19 <- read.csv("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/ACS_2015_2019.csv")
```

```r
# read needed ACS variables lists and IDs
acs_variables_id <- read_excel("C:/Users/tosea/Google-Trend/datasets/ACS(5 year estimates)/Variable Lists and IDs.xlsx")
print(xtable(acs_variables_id), scalebox = 0.8)
```

	ACS ID	ACS Variable Labels
1	Geo_FIPS	FIPS
2	SE_A00001_001	Total Population
3	SE_A04001_003	Total Population: Not Hispanic or Latino: White Alone
4	SE_A04001_010	Total Population: Hispanic or Latino
5	SE_A04001_005	Total Population: Not Hispanic or Latino: American Indian and Alaska Native Alone
6	SE_A04001_006	Total Population: Not Hispanic or Latino: Asian Alone
7	SE_A04001_007	Total Population: Not Hispanic or Latino: Native Hawaiian and Other Pacific Islander Alone
8	SE_A04001_004	Total Population: Not Hispanic or Latino: Black or African American Alone
9	SE_A06001_003	Total Population: Foreign Born
10	SE_A10062B_001	Total Population in Occupied Housing Units: Renter Occupied
11	SE_A08001_003	Total: Moved within same county
12	SE_A08001_004	Total: Moved from different county within same state
13	SE_A08001_005	Total: Moved from different state
14	SE_A08001_006	Total: Moved from abroad
15	SE_A17002_006	Population 16 Years and Over: in Labor Force: Civilian: Unemployed
16	SE_A13002_002	Families: Income in 2006 below poverty level
17	SE_A02002_008	Total Population: Male: 25 to 34 Years
18	SE_A02002_007	Total Population: Male: 18 to 24 Years
19	SE_A02002_006	Total Population: Male: 15 to 17 Years
20	SE_A12003_002	Civilian Population 16 to 19 Years: Not High School Graduate, Not Enrolled (Dropped Out)
21	SE_A12003_001	Civilian Population 16 to 19 Years:
22	SE_A10009_007	Households: Households with one or More people under 18 Years: Family Households: Other Family (Single Parent): Female Householder, no husband present
23	SE_A10008_001	Households:
24	SE_A12001_002	Population 25 Years and Over: Less than High School
25	SE_A11001_006	Population 15 Years and Over: Divorced

```r
#Read walkcross file (county to dmas)
walkcross <- read.csv("county_dma_crosswalk_harvard.csv")

#process cross-sectional data 2010-2019
pre_process_ACS <- function(data, walkcross, variable_id, year){
  #To make ID as column names
  data <- data %>%
    row_to_names(row_number = 1)

  #Keep needed variables
  data <- data[,c(variable_id["ACS ID"])[[1]]]
  data <- data %>%
    rename(
      FIPS = Geo_FIPS,
      )
  #Make string to numberic
  data <- data %>% retype()

  #Aggregate county level data into DMAs

    ## Merge ACS with walkcross
  data_new <- merge(data, walkcross,by="FIPS")

    ## Drop columns not needed
  data_new = subset(data_new, select = -c(FIPS, STATE, COUNTY,  Harvard_DMA))

    ## Group_by DMAs
  data_DMA <- data_new %>%
    group_by(DMA) %>%
    summarise_each(funs(sum))

  return(data_DMA)
  }

pre_acs_15_19 <- pre_process_ACS(acs_15_19, walkcross, acs_variables_id, "2019")
```

```
## Warning: `summarise_each_()` was deprecated in dplyr 0.7.0.
## Please use `across()` instead.

## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```r
pre_acs_10_14 <- pre_process_ACS(acs_10_14, walkcross, acs_variables_id, "2014")

#add numeric variables
acs_10_19 <- (pre_acs_15_19[2:25] + pre_acs_10_14[2:25])/2
```

```r
#add DMA and ID
acs_10_19["DMA"] <- pre_acs_15_19$DMA
acs_10_19["DMAINDEX"] <- pre_acs_15_19$DMAINDEX
```

```r
#Write a function to process acs_10_19 (combining and calculating "rates")
post_process_ACS <- function(data_DMA, year){
  data_DMA["Percentage of Foreign Born"] <- data_DMA$SE_A06001_003/data_DMA$SE_A00001_001

    ## Percentage of MoveIn: (Moved)/Total Pop. Use function scale to standardize data
    ## as the data of mean == 0, sd == 1
  data_DMA["Percentage of MoveIn"] <-
    (data_DMA$SE_A08001_003 +
       data_DMA$SE_A08001_004 +
       data_DMA$SE_A08001_005 +
       data_DMA$SE_A08001_006 )/
       data_DMA$SE_A00001_001
   ## Alpha Test of Percentage of MoveIn
  print(alpha(data.frame("Move Within Same County" = data_DMA$SE_A08001_003,
                   "Move from different county within same state" = data_DMA$SE_A08001_004,
                   "Move from different state" = data_DMA$SE_A08001_005,
                   "Move from abroad" = data_DMA$SE_A08001_006)))

   ## Percentage of Renter
  data_DMA["Percentage of Renter"] <- data_DMA$SE_A10062B_001/data_DMA$SE_A00001_001


  ## Mobility Index
  ## Move In + Renter
  data_DMA["Mobility Index"] <-
    #standardized percentage of renter
    scale(data_DMA$SE_A10062B_001/data_DMA$SE_A00001_001) +
    #standardized percentage of movein
    scale((data_DMA$SE_A08001_003 +
       data_DMA$SE_A08001_004 +
       data_DMA$SE_A08001_005 +
       data_DMA$SE_A08001_006 )/
       data_DMA$SE_A00001_001)

  ## Alpha Test
  print(alpha(data.frame("Percentage of MoveIn" = scale(data_DMA["Percentage of MoveIn"]),
                   "Percentage of Renter" = scale(data_DMA["Percentage of Renter"]))))

   ## Concentrated Disadvantages Index
   ##(Unemployed + Female-headed Family + Poverty Family + Less than High School)
   ## (Using "scale" func to standardize)
  data_DMA["Concentrated Disadvantaged Index"] <-
    scale(data_DMA$SE_A17002_006/data_DMA$SE_A00001_001) +
    scale(data_DMA$SE_A10009_007/data_DMA$SE_A10008_001) +
    scale(data_DMA$SE_A13002_002/data_DMA$SE_A10008_001) +
    scale(data_DMA$SE_A12001_002/data_DMA$SE_A00001_001)

  data_DMA["Percentage of Unemployed"] <-
  data_DMA$SE_A17002_006/data_DMA$SE_A00001_001
```

```r
data_DMA["Percentage of Female Headed Family"] <-
data_DMA$SE_A10009_007/data_DMA$SE_A10008_001

data_DMA["Percentage of Poverty"] <-
data_DMA$SE_A13002_002/data_DMA$SE_A10008_001

 ## Alpha Test of Concentrated Disadvantages Index
print(alpha(data.frame("Unemployed" = scale(data_DMA$SE_A17002_006/data_DMA$SE_A00001_001),
                 "Female Headed Family" = scale(data_DMA$SE_A10009_007/data_DMA$SE_A10008_001),
                 "Family Income Below Poverty" = scale(data_DMA$SE_A13002_00/data_DMA$SE_A10008_001),
                 "Less than High School" = scale(data_DMA$SE_A12001_002/data_DMA$SE_A00001_001))))

 ## Heterogeneity Index (The Probability that two persons are in different race)
 ## = 1 - (The probability that two persons are in the same race)
data_DMA["Heterogeneity Index"] <-
  1 - (
    #percentage of Non-Hispanic White ^2
    (data_DMA$SE_A04001_003/data_DMA$SE_A00001_001)^2 +
      #percentage of Hispanic or Latino ^2
      (data_DMA$SE_A04001_010/data_DMA$SE_A00001_001)^2 +
      #percentage of Non-Hispanic Native and Indian ^2
      (data_DMA$SE_A04001_005/data_DMA$SE_A00001_001)^2 +
      #percentage of Non-Hispanic Asian ^2
      (data_DMA$SE_A04001_006/data_DMA$SE_A00001_001)^2 +
      #percentage of Non-Hispanic Pacific Islander ^2
      (data_DMA$SE_A04001_007/data_DMA$SE_A00001_001)^2 +
      #percentage of Non-Hispanic Black ^2
      (data_DMA$SE_A04001_004/data_DMA$SE_A00001_001)^2
  )

#Percentage of Young Males
  data_DMA["Percentage of Young Males"] <-
  (data_DMA$SE_A02002_006 +
     data_DMA$SE_A02002_006 +
     data_DMA$SE_A02002_006
   )/data_DMA$SE_A00001_001

#Percentage of Dropp Out
data_DMA["Percentage of Dropped Out"] <- data_DMA$SE_A12003_002/data_DMA$SE_A12003_001

#Percentage of Divorced
data_DMA["Percentage of Divorced"] <- data_DMA$SE_A11001_006/data_DMA$SE_A00001_001

#log population
data_DMA["Popualtion(logged)"] <- log(data_DMA$SE_A00001_001)

#insert year value
data_DMA$year = year

#Percentage of Less Than High School (Population 25 Years and Over: Less than High School)
data_DMA["Less than High School"] <- data_DMA$SE_A12001_002/data_DMA$SE_A00001_001

#Percentage of Non-Hispanic White
```

```r
  data_DMA["Percentage of White"] <- data_DMA$SE_A04001_003/data_DMA$SE_A00001_001

  #Percentage of Non-Hispanic Black
  data_DMA["Percentage of Black"] <- data_DMA$SE_A04001_004/data_DMA$SE_A00001_001

  #Percentage of Hispanic
  data_DMA["Percentage of Hispanic"] <- data_DMA$SE_A04001_010/data_DMA$SE_A00001_001

  return(data_DMA[c("DMA",
                    "DMAINDEX",
                    "year",
                    "Percentage of Foreign Born",
                    "Percentage of MoveIn",
                    "Percentage of Renter",
                    "Mobility Index",
                    "Concentrated Disadvantaged Index",
                    "Percentage of Unemployed",
                    "Percentage of Female Headed Family",
                    "Percentage of Poverty",
                    "Heterogeneity Index",
                    "Percentage of Young Males",
                    "Percentage of Dropped Out",
                    "Percentage of Divorced",
                    "Popualtion(logged)",
                    "Less than High School",
                    "Percentage of White",
                    "Percentage of Black",
                    "Percentage of Hispanic",
                    "SE_A00001_001")])
}
acs_10_19_dma <- post_process_ACS(acs_10_19, "2010-2019")
```

```
## Number of categories should be increased  in order to count frequencies.

##
## Reliability analysis
## Call: alpha(x = data.frame(`Move Within Same County` = data_DMA$SE_A08001_003,
##     `Move from different county within same state` = data_DMA$SE_A08001_004,
##     `Move from different state` = data_DMA$SE_A08001_005, `Move from abroad` = data_DMA$SE_A08001_00
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase  mean    sd median_r
##       0.75      0.97    0.96      0.89  31 0.0075 53939 74022     0.88
##
##  lower alpha upper     95% confidence boundaries
## 0.73 0.75 0.76
##
##  Reliability if an item is dropped:
##                                               raw_alpha std.alpha G6(smc)
## Move.Within.Same.County                            0.86      0.95    0.93
## Move.from.different.county.within.same.state      0.58      0.96    0.94
## Move.from.different.state                          0.66      0.96    0.95
## Move.from.abroad                                   0.76      0.96    0.94
##                                               average_r S/N alpha se    var.r
## Move.Within.Same.County                            0.87  20   0.0074 0.000019
```

```
## Move.from.different.county.within.same.state      0.89  24   0.0098 0.000631
## Move.from.different.state                         0.90  26   0.0087 0.000478
## Move.from.abroad                                  0.89  23   0.0092 0.000225
##                                              med.r
## Move.Within.Same.County                       0.87
## Move.from.different.county.within.same.state  0.88
## Move.from.different.state                     0.90
## Move.from.abroad                              0.88
##
##  Item statistics
##                                                n raw.r std.r r.cor r.drop
## Move.Within.Same.County                      211  0.99  0.97  0.96   0.93
## Move.from.different.county.within.same.state 211  0.95  0.95  0.93   0.92
## Move.from.different.state                    211  0.93  0.95  0.92   0.90
## Move.from.abroad                             211  0.93  0.96  0.94   0.93
##                                                mean     sd
## Move.Within.Same.County                      125291 179819
## Move.from.different.county.within.same.state  47362  62055
## Move.from.different.state                     34031  45109
## Move.from.abroad                               9073  18852

## Number of categories should be increased  in order to count frequencies.

##
## Reliability analysis
## Call: alpha(x = data.frame(`Percentage of MoveIn` = scale(data_DMA["Percentage of MoveIn"]),
##     `Percentage of Renter` = scale(data_DMA["Percentage of Renter"])))
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase               mean   sd
##       0.65      0.65    0.48      0.48 1.8 0.049 0.0000000000000046 0.86
##  median_r
##      0.48
##
##  lower alpha upper     95% confidence boundaries
## 0.55 0.65 0.74
##
##  Reliability if an item is dropped:
##                      raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r
## Percentage.of.MoveIn      0.48      0.48    0.23      0.48 0.91       NA     0
## Percentage.of.Renter      0.48      0.48    0.23      0.48 0.91       NA     0
##                      med.r
## Percentage.of.MoveIn  0.48
## Percentage.of.Renter  0.48
##
##  Item statistics
##                        n raw.r std.r r.cor r.drop               mean sd
## Percentage.of.MoveIn 211  0.86  0.86  0.59   0.48 0.0000000000000045  1
## Percentage.of.Renter 211  0.86  0.86  0.59   0.48 0.0000000000000048  1

## Number of categories should be increased  in order to count frequencies.

##
## Reliability analysis
## Call: alpha(x = data.frame(Unemployed = scale(data_DMA$SE_A17002_006/data_DMA$SE_A00001_001),
##     `Female Headed Family` = scale(data_DMA$SE_A10009_007/data_DMA$SE_A10008_001),
##     `Family Income Below Poverty` = scale(data_DMA$SE_A13002_00/data_DMA$SE_A10008_001),
```

```
##     `Less than High School` = scale(data_DMA$SE_A12001_002/data_DMA$SE_A00001_001)))
## 
##   raw_alpha std.alpha G6(smc) average_r S/N   ase                  mean   sd
##       0.87      0.87    0.87      0.62 6.4 0.016 0.0000000000000000059 0.84
##  median_r
##      0.61
## 
##  lower alpha upper     95% confidence boundaries
## 0.83 0.87 0.9 
## 
##  Reliability if an item is dropped:
##                             raw_alpha std.alpha G6(smc) average_r  S/N alpha se
## Unemployed                       0.92      0.92    0.90      0.80 11.9   0.0095
## Female.Headed.Family             0.78      0.78    0.78      0.55  3.6   0.0273
## Family.Income.Below.Poverty      0.77      0.77    0.73      0.53  3.4   0.0276
## Less.than.High.School            0.81      0.81    0.80      0.59  4.3   0.0238
##                             var.r med.r
## Unemployed                  0.005  0.84
## Female.Headed.Family        0.067  0.42
## Family.Income.Below.Poverty 0.030  0.51
## Less.than.High.School       0.048  0.51
## 
##  Item statistics 
##                               n raw.r std.r r.cor r.drop                  mean
## Unemployed                  211  0.68  0.68  0.49   0.47 -0.000000000000000140
## Female.Headed.Family        211  0.91  0.91  0.89   0.82  0.000000000000000073
## Family.Income.Below.Poverty 211  0.92  0.92  0.93   0.84  0.000000000000000249
## Less.than.High.School       211  0.87  0.87  0.84   0.76 -0.000000000000000161
##                             sd
## Unemployed                   1
## Female.Headed.Family         1
## Family.Income.Below.Poverty  1
## Less.than.High.School        1
```

```r
# Processing UCR Crime
ucr <- read.csv("C://Users//tosea//Google-Trend//datasets//UCR_Crime_ICPSR//ucr_offenses_known_yearly_1

#write a function to process UCR data
ucr_func <- function(ucr, year1, year2){
  #filter needed year range
  ucr_year <- ucr[(ucr$year >= year1) & (ucr$year <= year2), ]
  #keep needed variables
  ucr_year <- ucr_year[c("fips_state_county_code",
                               "population",
                               "actual_rape_total",
                               "actual_burg_total",
                               "actual_theft_total",
                               "actual_mtr_veh_theft_total")]

  #aggregate single years to a period needed
  ucr_year_fips <- ucr_year %>%
      group_by(fips_state_county_code) %>%
      summarise_each(funs(sum))
```

```r
  #rename FIPS
  ucr_year_fips <- ucr_year_fips%>%
      rename(
        FIPS = fips_state_county_code,
        )

  #merge with walkcross, prepare to group_by DMAs
  ucr_year_fips <- merge(ucr_year_fips, walkcross,by="FIPS")

  #Drop unneeded variables
  ucr_year_fips <- subset(ucr_year_fips,
                              select = -c(FIPS, STATE, COUNTY, DMAINDEX, Harvard_DMA))

  ## Group_by DMAs
  ucr_year_DMA <- ucr_year_fips %>%
      group_by(DMA) %>%
      summarise_each(funs(sum))

  #crime rate
  ucr_year_DMA_rate <- ucr_year_DMA[-1]/ucr_year_DMA$population


  #add DMA back to the data
  ucr_year_DMA_rate["DMA"] <- ucr_year_DMA$DMA

  #rename
  colnames(ucr_year_DMA_rate) <- c("POP", "UCR Rape", "UCR Burglary",
                                   "UCR Larceny", "UCR MVT", "DMA")

  return(ucr_year_DMA_rate)
}

ucr_2010_2019 <- ucr_func(ucr, 2010, 2019)[,-c(1)]
```

```r
#Read Google Trends data


gt_10_19 <- read.csv("10-19_gt_crime/GT_Crime_SingleKeywordsOrigin.csv")


gt_10_19 <- gt_10_19[c("dma_area",
           "MVT_10_19",
           "Burglary_10_19",
           "Larceny_10_19",
           "Rape_10_19")
           ] %>% rename(
             DMA = dma_area,
             )
```

```r
#control variables (Internet Usage and Median Vehicles per Family)
cv = read.csv("sima.csv")

#FIPS to DMAs
cv <- merge(cv, walkcross,by="FIPS")
```

```
#drop unneeded columns
cv <- subset(cv, select = -c(FIPS, STATE, COUNTY, DMAINDEX, Harvard_DMA))

# group by DMAs (in Stata it is collapese(sum))
cv_dma <- cv %>%
    group_by(DMA) %>%
    summarise_each(funs(sum))

# for median Vehicle, we need to use average (in Stata it is collapese(mean))
cv_dma_mean <- cv %>%
    group_by(DMA) %>%
    summarise_each(funs(mean))


############################################
#use cv year 2010-2019 for data 2010-2019
cv_dma["Internet Usage HH 10_19"] <-
  ((cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2011 +
     cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2012 +
     cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2013 +
     cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2014 +
     cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2015 +
     cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2016 +
     cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2017 +
     cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2018 +
     cv_dma$X..Households.Using...Internet..Any.Internet.Online.usage..2019
   )/9)/
  ((cv_dma$X..Households..HHs...2011 +
      cv_dma$X..Households..HHs...2012 +
      cv_dma$X..Households..HHs...2013 +
      cv_dma$X..Households..HHs...2014 +
      cv_dma$X..Households..HHs...2015 +
      cv_dma$X..Households..HHs...2016 +
      cv_dma$X..Households..HHs...2017 +
      cv_dma$X..Households..HHs...2018 +
      cv_dma$X..Households..HHs...2019)/9)


cv_dma["Median Vehicle HH 10_19"] <-
  (cv_dma_mean$Household..Median.Vehicles..2011 +
     cv_dma_mean$Household..Median.Vehicles..2012 +
     cv_dma_mean$Household..Median.Vehicles..2013 +
     cv_dma_mean$Household..Median.Vehicles..2014 +
     cv_dma_mean$Household..Median.Vehicles..2015 +
     cv_dma_mean$Household..Median.Vehicles..2016 +
     cv_dma_mean$Household..Median.Vehicles..2017 +
     cv_dma_mean$Household..Median.Vehicles..2018 +
     cv_dma_mean$Household..Median.Vehicles..2019
   )/9


cv_dma_10_19 <- cv_dma[c("DMA",
                   "Internet Usage HH 10_19",
```

```r
                   "Median Vehicle HH 10_19")]
```

```r
# Add Drug Mobility 2010-2018 into the Cross Sectional Data

drug_sup <- read.csv("datasets/substance_abuse_data_12_16/NCHS_Drug_Poisoning_Mortality_by_County__Unit

#filter years which are larger than and equal to 2010
drug_sup <- drug_sup %>%
  subset(drug_sup$Year >= 2010)

#average counts of all years
drug_sup_fips <- drug_sup[c(1,5)] %>%
  group_by(FIPS) %>%
  summarize_each(mean)

#rename
colnames(drug_sup_fips) <- c("FIPS", "Drug_Mortality_Count_10_18")

#from county to DMA
drug_sup_dma <- drug_sup_fips %>%
  merge(walkcross, by = "FIPS")

drug_sup_dma_10_18 <- drug_sup_dma[c("DMA", "Drug_Mortality_Count_10_18")] %>%
  group_by(DMA) %>%
  summarize_each(sum)
drug_temp <- merge(acs_10_19_dma, drug_sup_dma_10_18, by = "DMA")

#calculate the mortality rate
drug_temp$Drug.Mortality.Rate <-
  drug_temp$Drug_Mortality_Count_10_18/drug_temp$SE_A00001_001*1000000

drug_mortality_rate_10_18 <- drug_temp[c("DMA", "Drug.Mortality.Rate")]
```

```r
#law enforcement employments
law_enforce_10_14 <- retype(read.csv("datasets/police_2010_2019/law enforcement employments 10-14.csv")
law_enforce_15_19 <- retype(read.csv("datasets/police_2010_2019/law enforcement employments 15-19.csv")

colnames(law_enforce_10_14) <- c("FIPS", "law_enforcement_employments_10_14")
colnames(law_enforce_15_19) <- c("FIPS", "law_enforcement_employments_15_19")

law_enforce_10_19 <- merge(law_enforce_10_14, law_enforce_15_19, by = "FIPS")

law_enforce_10_19["law_enforcement_employments_10_19"] <-
  (law_enforce_10_19["law_enforcement_employments_10_14"]+
  law_enforce_10_19["law_enforcement_employments_15_19"])/2
law_enforce_10_19_dma <- merge(
  law_enforce_10_19, walkcross, by = "FIPS")



law_enforce_10_19_dma <- law_enforce_10_19_dma %>%
  dplyr::select("DMA", "law_enforcement_employments_10_19") %>%
  group_by(DMA) %>%
  summarize_each(sum) %>%
```

```r
  merge(acs_10_19_dma[c("DMA", "SE_A00001_001")], by = "DMA")

law_enforce_10_19_dma_per_thousand <- law_enforce_10_19_dma %>%
  mutate(law_enforcement_employments_10_19_per_thousand =
           law_enforcement_employments_10_19/SE_A00001_001*1000) %>%
  dplyr::select("DMA", "law_enforcement_employments_10_19_per_thousand")
```

```r
total_10_19 <-
  merge(
    merge(
      merge(
        merge(
          merge(
            gt_10_19, ucr_2010_2019, by = "DMA"),
          acs_10_19_dma,by = "DMA"),
        drug_mortality_rate_10_18, by = "DMA"),
      law_enforce_10_19_dma_per_thousand, by = "DMA"),
    cv_dma_10_19, by = "DMA")




rename_list <- c("DMA",
                 "GT.MVT",
                 "GT.Burglary",
                 "GT.Larceny",
                 "GT.Rape",
                 "UCR.Rape",
                 "UCR.Burglary",
                 "UCR.Larceny",
                 "UCR.MVT",
                 "DMA.INDEX",
                 "year",
                 "Percentage.of.Foreign.Born",
                 "Percentage.of.MoveIn",
                 "Percentage.of.Renter",
                 "Mobility.Index",
                 "Concentrated.Disadvantage.Index",
                 "Percentage.of.Unemployed",
                 "Percentage.of.Female.Headed.Family",
                 "Percentage.of.Poverty",
                 "Heterogeneity.Index",
                 "Percentage.of.Young.Males",
                 "Percentage.of.Dropped.Out",
                 "Percentage.of.Divorced",
                 "Population.logged",
                 "Less.than.High.School",
                 "Percentage.of.White",
                 "Percentage.of.Black",
                 "Percentage.of.Hispanic",
                 "Population",
                 "Drug.Mortality.Rate",
                 "Law.Enforce.per.Thousand",
                 "Internet.Usage.HH",
```

```r
                "Median.Vehicle.HH")

colnames(total_10_19) <- rename_list
```

```r
#get the 2010-2019 cross sectional data and drop na
final.data.2010_2019 <- total_10_19 %>%
  subset(select = -c(year)) %>%
  drop_na()
final.data.2010_2019$GT.Rape
```

```
##   [1] 56.242 56.185 59.476 45.131 75.924 52.012 55.151 68.141 47.714 47.296
##  [11] 62.028 59.304 67.456 87.058 55.490 56.536 70.484 53.867 57.264 58.470
##  [21] 59.242 61.026 60.262 58.149 56.333 61.881 47.095 59.968 61.970 56.847
##  [31] 53.405 82.919 88.833 67.827 49.750 62.103 58.915 56.103 58.349 60.377
##  [41] 51.194 55.323 49.579 54.810 61.841 60.776 79.119 60.105 60.681 70.179
##  [51] 57.496 68.183 74.952 49.579 58.202 50.516 64.716 46.085 45.095 47.105
##  [61] 64.661 32.571 64.788 62.667 48.831 59.554 63.702 50.780 55.177 60.400
##  [71] 56.621 51.258 59.637 58.274 44.669 56.391 56.611 50.143 62.196 54.095
##  [81] 53.452 54.796 49.690 67.823 39.776 36.806 43.571 60.329 48.381 83.308
##  [91] 62.048 68.315 78.742 65.151 63.113 65.700 53.319 67.863 94.720 52.079
## [101] 59.512 68.669 45.792 58.127 88.478 73.433 56.679
```

```r
#rescale GT and UCR crime rates into 0-100 after dropping NA
dependent_variable_col_names <- c("GT.MVT", "GT.Burglary", "GT.Larceny", "GT.Rape",
                                  "UCR.MVT", "UCR.Burglary", "UCR.Larceny", "UCR.Rape")

for (i in dependent_variable_col_names){
    final.data.2010_2019[[i]] <- final.data.2010_2019[[i]]/max(final.data.2010_2019[[i]])*100
  }


operated_data_and_variables <- final.data.2010_2019[,c("GT.MVT","GT.Burglary","GT.Larceny", "GT.Rape",
                                    "UCR.MVT","UCR.Burglary","UCR.Larceny", "UCR.Rape",
                                    "Concentrated.Disadvantage.Index",
                                    #"Percentage.of.Unemployed",
                                    #"Percentage.of.Female.Headed.Family",
                                    #"Percentage.of.Poverty",
                                    #"Percentage.of.MoveIn",
                                    #"Percentage.of.Renter",
                                    "Mobility.Index",
                                    "Heterogeneity.Index",
                                    "Percentage.of.Foreign.Born",
                                    #"Percentage.of.Black",
                                    #"Percentage.of.Hispanic",
                                    #"Percentage.of.White",
                                    "Percentage.of.Divorced",
                                    #"Less.than.High.School",
                                    "Percentage.of.Young.Males",
                                    "Drug.Mortality.Rate",
                                    #"Law.Enforce.per.Thousand",
                                    "Population.logged",
                                    "Internet.Usage.HH",
                                    "Median.Vehicle.HH")]
desc_10_19 <- round(t(stat.desc(operated_data_and_variables))[,c("mean", "std.dev", "max", "min")], 2)
```

```r
desc_table <- cbind.data.frame(desc_10_19)
colnames(desc_table) <- c("Mean", "SD", "Max", "Min")
list_of_row_names <- c("GT Motor Vehicle Theft", "GT Burglary", "GT Larceny", "GT Rape",
                          "UCR Motor Vehivle Theft", "UCR Burglary", "UCR Larceny", "UCR Rape",
                          "Concentrated Disadvantages Index",
                          #"% Unemployed",
                          #"% Female Headed Family",
                          #"% Poverty",
                          #"% Move In",
                          #"% Renter",
                          "Mobility Index",
                          "Heterogeneity Index",
                          "% Foreign Born",
                          #"% Black",
                          #"% Hispanic",
                          #"% White",
                          "% Divorced",
                          #"% Less than High School",
                          "% Young Males",
                          "Drug Mortality Rate",
                          #"Law Enforce per Thousand",
                          "Population(log)",
                          "% Internet Usage HH",
                          "Median Vehicle HH")

rownames(desc_table) <- list_of_row_names
```

Table 1: Descriptive Statistics for 2010 to 2019

	Mean	SD	Max	Min
Crime Rates (Outcome Variables)				
GT Motor Vehicle Theft	57.91	13.69	100.00	23.74
GT Burglary	62.42	13.53	100.00	26.54
GT Larceny	53.23	10.22	100.00	31.26
GT Rape	62.66	11.39	100.00	34.39
UCR Motor Vehivle Theft	34.09	17.37	100.00	8.48
UCR Burglary	49.24	15.62	100.00	17.64
UCR Larceny	59.83	13.23	100.00	35.28
UCR Rape	50.97	15.42	100.00	21.37
Predictor Variables				
Concentrated Disadvantages Index	0.26	2.96	13.43	-5.06
Mobility Index	0.23	1.61	5.73	-2.45
Heterogeneity Index	0.45	0.15	0.71	0.11
% Foreign Born	0.10	0.07	0.32	0.01
% Divorced	0.09	0.01	0.12	0.06
% Young Males	0.06	0.00	0.08	0.05
Drug Mortality Rate	173.79	60.42	397.77	44.26
Control Variables				
Population(log)	14.39	0.82	16.87	10.80
% Internet Usage HH	0.82	0.01	0.84	0.79
Median Vehicle HH	2.15	0.13	2.41	1.63

[1] N = 107 DMAs. [2] GT = Google Trends Crime Estimates. [3] HH = Household [4] Concentrated Disadvantaged Index combines the normalized percentage of unemployments, the percentage of female-headed family, the percentage of poverty, and the percentage of less than high school education(alpha = .865). [5] Mobility Index combines the normalized percentage of moved in(moved within same county, moved from different county within same state, moved from different state, and moved from abroad) and the percentage of renters(alpha = .645). [6] Heterogeneity Index is the probability of randomly choosen two individuals in the DMA, and they would be different races.

```r
###Correlation Matrix Function
corstars <-function(x, method=c("pearson", "spearman"), removeTriangle=c("upper", "lower"),
                     result=c("none", "html", "latex")){
    #Compute correlation matrix
    require(Hmisc)
    x <- as.matrix(x)
    correlation_matrix<-rcorr(x, type=method[1])
    R <- correlation_matrix$r # Matrix of correlation coeficients
    p <- correlation_matrix$P # Matrix of p-value

    ## Define notions for significance levels; spacing is important.
    mystars <- ifelse(p < .01, "**", ifelse(p < .05, "*\ ", "\ \ "))

    ## trunctuate the correlation matrix to two decimal
    R <- format(round(cbind(rep(-1.11, ncol(x)), R), 2))[,-1]

    ## build a new matrix that includes the correlations with their apropriate stars
    Rnew <- matrix(paste(R, mystars, sep=""), ncol=ncol(x))
    diag(Rnew) <- paste(diag(R), " ", sep="")
    rownames(Rnew) <- colnames(x)
    colnames(Rnew) <- paste(colnames(x), "", sep="")

    ## remove upper triangle of correlation matrix
    if(removeTriangle[1]=="upper"){
      Rnew <- as.matrix(Rnew)
      Rnew[upper.tri(Rnew, diag = TRUE)] <- ""
      Rnew <- as.data.frame(Rnew)
    }

    ## remove lower triangle of correlation matrix
    else if(removeTriangle[1]=="lower"){
      Rnew <- as.matrix(Rnew)
      Rnew[lower.tri(Rnew, diag = TRUE)] <- ""
      Rnew <- as.data.frame(Rnew)
    }

    ## remove last column and return the correlation matrix
    Rnew <- cbind(Rnew[1:length(Rnew)-1])
    if (result[1]=="none") return(Rnew)
    else{
      if(result[1]=="html") print(xtable(Rnew), type="html")
      else print(xtable(Rnew), type="latex")
    }
}
```

```r
#Correlation Matrix

cor_table <- corstars(operated_data_and_variables)
```

```
## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
## The following object is masked from 'package:psych':
##
##     describe
## The following object is masked from 'package:prettyR':
##
##     describe
## The following objects are masked from 'package:xtable':
##
##     label, label<-
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
#insert index at the end of rowname
cor_table["new"] <- c("(1)", "(2)", "(3)", "(4)",
                        "(5)", "(6)", "(7)", "(8)",
                        "(9)", "(10)", "(11)", "(12)",
                        "(13)", "(14)", "(15)", "(16)",
                        "(17)", "(18)")
cor_table <- cor_table[,c(18, 1:17)]


#insert index as colnames
colnames(cor_table) <- c(" ", "(1)", "(2)", "(3)", "(4)",
                        "(5)", "(6)", "(7)", "(8)",
                        "(9)", "(10)", "(11)", "(12)",
                        "(13)", "(14)", "(15)", "(16)",
                        "(17)")



rownames(cor_table) <- c("GT MVT",
                         "GT Burglary",
                         "GT Larceny",
                         "GT Rape",
                         "UCR MVT",
                         "UCR Burglary",
                         "UCR Larceny",
                         "UCR Rape",
                         "CD Index",
                          #"% Unemployed",
                          #"% Female Headed Family",
                          #"% Poverty",
                          #"% Move In",
                          #"% Renter",
                          "Mobility Index",
                          "Heterogeneity Index",
```

```
                        "% Foreign Born",
                        #"% Black",
                        #"% Hispanic",
                        #"% White",
                        "% Divorced",
                        #"% Less than High School",
                        "% Young Males",
                        "Drug Mortality Rate",
                        #"Law Enforce per Thousand",
                        "Population(log)",
                        "% Internet Usage HH",
                        "Median Vehicle HH")
```

Table 2: Correlation Matrix of All Measures

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
GT MVT	(1)																	
GT Burglary	(2)	0.29**																
GT Larceny	(3)	0.81**	0.41**															
GT Rape	(4)	-0.09	0.23*	0.10														
UCR MVT	(5)	0.81**	0.27**	0.76**	-0.02													
UCR Burglary	(6)	0.42**	0.46**	0.56**	0.47**	0.52**												
UCR Larceny	(7)	0.32**	0.33**	0.40**	0.26**	0.45**	0.70**											
UCR Rape	(8)	0.08	-0.04	0.19*	0.26**	0.16	0.31**	0.29**										
CD Index	(9)	0.25**	0.30**	0.51**	0.36**	0.36**	0.53**	0.36**	-0.01									
Mobility Index	(10)	0.48**	0.30**	0.55**	-0.09	0.50**	0.29**	0.32**	0.17	0.25*								
Heterogeneity Index	(11)	0.55**	0.19*	0.51**	-0.20*	0.55**	0.34**	0.31**	-0.08	0.32**	0.54**							
% Foreign Born	(12)	0.41**	0.01	0.44**	-0.52**	0.41**	-0.13	-0.03	-0.22*	0.28**	0.40**	0.57**						
% Divorced	(13)	0.04	0.16	0.06	0.29**	-0.06	0.29**	0.17	0.28**	-0.14	0.08	-0.20*	-0.42**					
% Young Males	(14)	0.17	0.02	0.29**	0.09	0.25**	0.16	0.10	0.16	0.45**	-0.09	0.00	0.18	-0.45**				
Drug Mortality Rate	(15)	0.05	-0.15	0.02	0.19	-0.04	0.07	-0.07	-0.01	-0.06	-0.18	-0.24*	-0.29**	0.43**	-0.24*			
Population(log)	(16)	0.27**	-0.24*	0.10	-0.42**	0.16	-0.20*	-0.14	-0.26**	-0.17	-0.12	0.35**	0.47**	-0.34**	0.10	0.01		
% Internet Usage HH	(17)	0.35**	-0.10	0.20*	-0.53**	0.35**	-0.16	-0.02	-0.11	-0.20*	0.28**	0.47**	0.47**	-0.50**	0.28**	-0.26**	0.56**	
Median Vehicle HH	(18)	0.10	-0.13	0.00	-0.17	0.18	-0.07	-0.09	0.12	-0.38**	-0.03	-0.16	-0.19	-0.16	0.30**	-0.12	0.07	0.47**

Note: [1] ** = p < .01, * = p < .05; [2] GT = Google Tredns Crime Estimates [3] MVT = Motor Vehicle Theft [4] CD = Concentrated Disadvantages [5] HH = Household

```r
#write a OLS regression function (for one period 2010-2019)
lm_func <- function(data, y, control_Vehicle){
  data <- drop_na(data)
  if (control_Vehicle == FALSE){
    lm_model <- lm(data = data,
                   data[[y]] ~ Concentrated.Disadvantage.Index +
                       #Percentage.of.Unemployed +
                       #Percentage.of.Female.Headed.Family +
                       #Percentage.of.Poverty +
                       #Percentage.of.MoveIn +
                       #Percentage.of.Renter +
                       Mobility.Index +
                       Heterogeneity.Index +
                       Percentage.of.Foreign.Born +
                       #Percentage.of.Black +
                       #Percentage.of.Hispanic +
                       #Percentage.of.White +
                       Percentage.of.Divorced +
                       #Less.than.High.School +
                       Percentage.of.Young.Males +
                       Drug.Mortality.Rate +
                       #Law.Enforce.per.Thousand +
                       Population.logged +
                       Internet.Usage.HH)
  }
  else{
    lm_model <- lm(data = data,
                   data[[y]] ~ Concentrated.Disadvantage.Index +
                     #Percentage.of.Unemployed +
                       #Percentage.of.Female.Headed.Family +
                       #Percentage.of.Poverty +
                       #Percentage.of.MoveIn +
                       #Percentage.of.Renter +
                       Mobility.Index +
                       Heterogeneity.Index +
                       Percentage.of.Foreign.Born +
                       #Percentage.of.Black +
                       #Percentage.of.Hispanic +
                       #Percentage.of.White +
                       Percentage.of.Divorced +
                       #Less.than.High.School +
                       Percentage.of.Young.Males +
                       Drug.Mortality.Rate +
                       #Law.Enforce.per.Thousand +
                       Population.logged +
                       Internet.Usage.HH +
                       Median.Vehicle.HH)
  }
  return(lm_model)
}
```

```r
#GT_MVT 2010-2019 OLS model
GT_MVT_lm <- lm_func(final.data.2010_2019,
                     "GT.MVT",
```

```
                     control_Vehicle = TRUE)

#UCR_MVT 2010-2019 OLS model
UCR_MVT_lm <- lm_func(final.data.2010_2019,
                        "UCR.MVT",
                        control_Vehicle = TRUE)

#GT_Burglary 2010-2019 OLS model
GT_Burglary_lm <- lm_func(final.data.2010_2019,
                     "GT.Burglary",
                     control_Vehicle = FALSE)

#UCR_Burglary 2010-2019 OLS model
UCR_Burglary_lm <- lm_func(final.data.2010_2019,
                        "UCR.Burglary",
                        control_Vehicle = FALSE)

#GT_Larceny 2010-2019 OLS model
GT_Larceny_lm <- lm_func(final.data.2010_2019,
                     "GT.Larceny",
                     control_Vehicle = FALSE)

#UCR_Larceny 2010-2019 OLS model"\\%
UCR_Larceny_lm <- lm_func(final.data.2010_2019,
                        "UCR.Larceny",
                        control_Vehicle = FALSE)

#GT_Rape 2010-2019 OLS model
GT_Rape_lm <- lm_func(final.data.2010_2019,
                     "GT.Rape",
                     control_Vehicle = FALSE)

#UCR_Burglary 2010-2019 OLS model
UCR_Rape_lm <- lm_func(final.data.2010_2019,
                        "UCR.Rape",
                        control_Vehicle = FALSE)
```

```
# Test RMSE of each model
library(Metrics)

ucr_mvt_rmse <- rmse(UCR_MVT_lm$fitted.values, final.data.2010_2019$UCR.MVT) #11.38
gt_mvt_rmse <- rmse(GT_MVT_lm$fitted.values, final.data.2010_2019$GT.MVT) #9.27
ucr_burglary_rmse <- rmse(UCR_Burglary_lm$fitted.values, final.data.2010_2019$UCR.Burglary) #9.78
gt_burglary_rmse <- rmse(GT_Burglary_lm$fitted.values, final.data.2010_2019$GT.Burglary) #11.82
ucr_larceny_rmse <- rmse(UCR_Larceny_lm$fitted.values, final.data.2010_2019$UCR.Larceny) #10.80
gt_larceny_rmse <- rmse(GT_Larceny_lm$fitted.values, final.data.2010_2019$GT.Larceny) #6.2
ucr_rape_rmse <- rmse(UCR_Rape_lm$fitted.values, final.data.2010_2019$UCR.Rape) #12.72
gt_rape_rmse <- rmse(GT_Rape_lm$fitted.values, final.data.2010_2019$GT.Rape) #7.32
total_rmse <- c(gt_mvt_rmse, ucr_mvt_rmse, gt_burglary_rmse, ucr_burglary_rmse,
                gt_larceny_rmse, ucr_larceny_rmse, gt_rape_rmse, ucr_rape_rmse)
```

```r
list_of_variable_showing_names <- c(#"GT MVT", "GT Burglary", "GT Larceny", "GT Rape",
                        #"UCR MVT", "UCR Burglary", "UCR Larceny", "UCR Rape",
                        "Concentrated Disadvantages Index",
                               #"\\% Unemployed",
                               #"\\% Female Headed Family",
                               #"\\% Poverty",
                               #"\\% Move In",
                               #"\\% Renter",
                               "Mobility Index",
                               "Heterogeneity Index",
                               "\\% Foreign Born",
                               #"\\% Black",
                               #"\\% Hispanic",
                               #"\\% White",
                               "\\% Divorced",
                               #"\\% Less than High School",
                               "\\% Young Males",
                               "Drug Mortality Rate",
                               #"Law Enforce per Thousand",
                               "Population(log)",
                               "\\% Internet Usage HH",
                               "Median Vehicle HH")

stargazer(GT_MVT_lm, UCR_MVT_lm,
          GT_Burglary_lm, UCR_Burglary_lm,
          GT_Larceny_lm, UCR_Larceny_lm,
          GT_Rape_lm, UCR_Rape_lm,
          title = "OLS Model of Google Trends and UCR Crime Estimation on Crime Factors, 2010-2019",
          omit.stat = c("rsq", "ll", "ser"),
          no.space = TRUE,
          table.placement = "h!",
          notes = c("GT = Google Trends Crime Estimates; MVT = Motor Vehicle Theft; HH = Household"),
          notes.align = "l",
          dep.var.labels.include = F,
          column.labels = c("GT MVT", "UCR MVT", "GT Burglary", "UCR Burglary",
                            "GT Larceny", "UCR Larceny", "GT Rape", "UCR Rape"),
          covariate.labels = list_of_variable_showing_names,
          column.sep.width = "1.5pt",
```

```
          font.size = "small",
          add.lines = list(c("RMSE", round(total_rmse, 3))),
          df = F)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: Wed, Apr 07, 2021 - 12:33:07 AM

Table 3: OLS Model of Google Trends and UCR Crime Estimation on Crime Factors, 2010-2019

	Dependent variable:							
	GT MVT	UCR MVT	GT Burglary	UCR Burglary	GT Larceny	UCR Larceny	GT Rape	UCR Rape
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Concentrated Disadvantages Index	0.249	1.473^{*}	1.306^{*}	2.400^{***}	0.713^{*}	1.514^{**}	1.415^{***}	-2.199^{***}
	(0.629)	(0.772)	(0.761)	(0.629)	(0.399)	(0.695)	(0.471)	(0.819)
Mobility Index	2.115^{**}	2.283^{**}	1.116	1.360	2.175^{***}	1.473	1.059	3.431^{***}
	(0.929)	(1.140)	(1.168)	(0.965)	(0.612)	(1.066)	(0.723)	(1.256)
Heterogeneity Index	37.965^{***}	45.244^{***}	9.614	46.928^{***}	15.012^{**}	25.039^{*}	6.539	18.174
	(11.262)	(13.829)	(14.204)	(11.743)	(7.442)	(12.968)	(8.797)	(15.277)
% Foreign Born	38.617^{*}	55.566^{**}	-21.505	-103.708^{***}	27.669^{**}	-65.104^{***}	-101.215^{***}	-37.459
	(21.205)	(26.038)	(25.626)	(21.186)	(13.427)	(23.396)	(15.872)	(27.562)
% Divorced	295.069^{**}	209.529	246.283^{*}	409.481^{***}	263.368^{***}	272.380^{**}	-3.387	326.483^{**}
	(116.151)	(142.625)	(147.321)	(121.794)	(77.191)	(134.505)	(91.248)	(158.451)
% Young Males	629.508^{*}	399.942	-65.218	629.337^{*}	768.450^{***}	234.538	324.840	$1{,}975.800^{***}$
	(342.629)	(420.723)	(403.073)	(333.230)	(211.195)	(368.007)	(249.654)	(433.523)
Drug Mortality Rate	0.045^{**}	0.049^{**}	-0.046^{*}	-0.001	0.027^{**}	-0.026	0.002	-0.005
	(0.019)	(0.024)	(0.025)	(0.020)	(0.013)	(0.022)	(0.015)	(0.026)
Population(log)	2.610	0.608	-1.713	1.147	0.715	0.709	1.504	-1.336
	(1.838)	(2.257)	(2.324)	(1.921)	(1.218)	(2.122)	(1.439)	(2.499)
% Internet Usage HH	-113.433	-137.726	58.222	-36.420	22.520	109.698	-396.893^{**}	-490.958^{*}
	(213.590)	(262.273)	(253.140)	(209.277)	(132.636)	(231.118)	(156.789)	(272.264)
Median Vehicle HH	26.760^{**}	57.946^{***}						
	(11.379)	(13.973)						
Constant	-39.525	-65.890	25.831	-25.623	-61.225	-80.666	351.428^{***}	317.114
	(155.478)	(190.916)	(189.842)	(156.947)	(99.470)	(173.327)	(117.584)	(204.184)
RMSE	9.267	11.379	11.824	9.776	6.196	10.796	7.324	12.718
Observations	107	107	107	107	107	107	107	107
Adjusted R^2	0.489	0.522	0.158	0.568	0.595	0.265	0.544	0.250
F Statistic	11.146^{***}	12.568^{***}	3.205^{***}	16.482^{***}	18.301^{***}	5.247^{***}	15.037^{***}	4.924^{***}

Note: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$
GT = Google Trends Crime Estimates; MVT = Motor Vehicle Theft; HH = Household

```r
#write a plot function
plot_texts <- function(df, title){
  df <- drop_na(df)
  mvt_lm <- lm(data = df, UCR.MVT ~ GT.MVT)
  burglary_lm <- lm(data = df, UCR.Burglary ~ GT.Burglary)
  larceny_lm <- lm(data = df, UCR.Larceny ~ GT.Larceny)
  rape_lm <- lm(data = df, UCR.Rape ~ GT.Rape)
  #_________________________________________________
  #plot cook's distance in MVT
  MVT_text_plot1 <- ggplot(df, aes(x = GT.MVT,
                                   y = UCR.MVT,
                                   label = DMA,
                                   size = cooks.distance(mvt_lm),
                                   col = (cooks.distance(mvt_lm) > 4/(
                                     length(final.data.2010_2019$DMA) - 1 - 1))&(
                                     GT.MVT > median(df$GT.MVT)) & (
                                       UCR.MVT < median(df$UCR.MVT)
                                       ))) +
  geom_text(vjust = 2) +
  geom_point() +
  geom_abline(intercept = coef(lm(data = df, UCR.MVT~GT.MVT))[1],
              slope = coef(lm(data = df, UCR.MVT~GT.MVT))[2],color = "gray28")+
  geom_hline(aes(yintercept=median(df$UCR.MVT), linetype= " "), col = "firebrick3", show.legend = FALSE)
  geom_vline(aes(xintercept=median(df$GT.MVT), linetype= " "), col = "firebrick3", show.legend = FALSE)
  scale_linetype_manual("Median", values = c(2,2)) +
  ylab("UCR MVT Rate (0 to 100 scale)") +
    xlab("GT MVT Estimates") +
    scale_color_manual("High\nInfluence\n&\nUnderreported\nZone",
                       values = c("TRUE" = "gold2",
                                  "FALSE" = "lightblue")) +
    guides(color = FALSE) +
    scale_size("Cook’s\nDistance") +
    xlim(0, 120) +
    ylim(-10, 110)
  #_________________________________________________
  #plot cook's distance in Burglary
  burglary_text_plot <- ggplot(df, aes(x = GT.Burglary,
                                   y = UCR.Burglary,
                                   label = DMA,
                                   size = cooks.distance(burglary_lm),
                                   col = (cooks.distance(burglary_lm) > 4/(
                                     length(final.data.2010_2019$DMA) - 1 - 1))&(
                                     GT.Burglary > median(df$GT.Burglary)) & (
                                       UCR.Burglary < median(df$UCR.Burglary)
                                       ))) +
  geom_text(vjust = 2) +
  geom_point()  +
  geom_abline(intercept = coef(lm(data = df, UCR.Burglary~GT.Burglary))[1],
              slope = coef(lm(data = df, UCR.Burglary~GT.Burglary))[2],color = "gray28")+
  geom_hline(aes(yintercept=median(df$UCR.Burglary), linetype= " "), col = "firebrick3", show.legend = F
  geom_vline(aes(xintercept=median(df$GT.Burglary), linetype= " "), col = "firebrick3", show.legend = FA
  scale_linetype_manual("Median", values = c(2,2)) +
  ylab("UCR Burglary Rate (0 to 100 scale)") +
```

```
  xlab("GT Burglary Estimates") +
  scale_color_manual("High\nInfluence\n&\nUnderreported\nZone",
                     values = c("TRUE" = "gold2",
                                "FALSE" = "lightblue")) +
  guides(color = FALSE) +
  scale_size("Cook's\nDistance") +
  xlim(0, 120) +
  ylim(-10, 110)

#________________________________________________
#plot cook's distance in Larceny
larceny_text_plot <- ggplot(df, aes(x = GT.Larceny,
                            y = UCR.Larceny,
                            label = DMA,
                            size = cooks.distance(larceny_lm),
                            col = (cooks.distance(larceny_lm) > 4/(
                              length(final.data.2010_2019$DMA) - 1 - 1))&(
                              GT.Larceny > median(df$GT.Larceny)) & (
                                UCR.Larceny < median(df$UCR.Larceny)
                                ))) +
geom_text(vjust = 2) +
geom_point() +
geom_abline(intercept = coef(lm(data = df, UCR.Larceny~GT.Larceny))[1],
            slope = coef(lm(data = df, UCR.Larceny~GT.Larceny))[2],color = "gray28")+
geom_hline(aes(yintercept=median(df$UCR.Larceny), linetype= " "), col = "firebrick3", show.legend = F
geom_vline(aes(xintercept=median(df$GT.Larceny), linetype= " "), col = "firebrick3", show.legend = FA
scale_linetype_manual("Median", values = c(2,2)) +
ylab("UCR Larceny Rate (0 to 100 scale)") +
xlab("GT Larceny Estimates") +
  scale_color_manual("High\nInfluence\n&\nUnderreported\nZone",
                     values = c("TRUE" = "gold2",
                                "FALSE" = "lightblue")) +
  guides(color = FALSE) +
  scale_size("Cook's\nDistance") +
  xlim(0, 120) +
  ylim(-10, 110)

#________________________________________________
#plot cook's distance in Rape
rape_text_plot <- ggplot(df, aes(x = GT.Rape,
                            y = UCR.Rape,
                            label = DMA,
                            size = cooks.distance(rape_lm),
                            col = (cooks.distance(rape_lm) > 4/(
                              length(final.data.2010_2019$DMA) - 1 - 1))&(
                              GT.Rape > median(df$GT.Rape)) & (
                                UCR.Rape < median(df$UCR.Rape)
                                ))) +
geom_text(vjust = 2) +
geom_point() +
geom_abline(intercept = coef(lm(data = df, UCR.Rape~GT.Rape))[1],
            slope = coef(lm(data = df, UCR.Rape~GT.Rape))[2],color = "gray28")+
geom_hline(aes(yintercept=median(df$UCR.Rape), linetype= " "), col = "firebrick3", show.legend = FALSE
```

```
geom_vline(aes(xintercept=median(df$GT.Rape), linetype= " "), col = "firebrick3", show.legend = FALSE
scale_linetype_manual("Median", values = c(2,2)) +
scale_size("Cook's\nDistance") +
ylab("UCR Rape Rate (0 to 100 scale)") +
  xlab("GT Rape Estimates") +
  scale_color_manual("High\nInfluence\n&\nUnderreported\nZone",
                     values = c("TRUE" = "gold2",
                                "FALSE" = "lightblue")) +
  guides(color = FALSE) +
  xlim(0, 120) +
  ylim(-10, 110)
#______________________________________________
#just to get the legend
just_to_get_the_legend1 <- ggplot(df, aes(x = GT.Rape,
                                 y = UCR.Rape,
                                 label = DMA,
                                 size = cooks.distance(rape_lm),
                                 col = (cooks.distance(rape_lm) > 4/(
                                   length(final.data.2010_2019$DMA) - 1 - 1))&(
                                   GT.Rape > median(df$GT.Rape)) & (
                                     UCR.Rape < median(df$UCR.Rape)
                                     ))) +
geom_text(vjust = 2) +
geom_point() +
geom_hline(aes(yintercept=median(df$UCR.Rape), linetype= " "), col = "firebrick3", show.legend = FALSE
geom_vline(aes(xintercept=median(df$GT.Rape), linetype= " "), col = "firebrick3", show.legend = FALSE
scale_linetype_manual("Median", values = c(2,2)) +
labs(title = "Rape") +
scale_size("Cook's\nDistance") +
  scale_color_manual("High Influence & In the Underreported Zone",
                     values = c("TRUE" = "gold2",
                                "FALSE" = "lightblue")) +
  guides(size = FALSE) +
  xlim(0, 120) +
  ylim(-10, 110) +theme(legend.position = "top")

#legends of Median lines
just_to_get_the_legend2 <- ggplot(df, aes(x = GT.Rape,
                                 y = UCR.Rape,
                                 label = DMA
                                     )) +
geom_hline(aes(yintercept=median(df$UCR.Rape), linetype= " "), col = "firebrick3") +
scale_linetype_manual("Median", values = c(2,2)) + theme(legend.position = "top")

#legends of Regression line
just_to_get_the_legend3 <- ggplot(data = df, aes(x = GT.Rape, y = UCR.Rape)) +
geom_point()+
geom_abline(aes(intercept = coef(lm(data = df, UCR.Rape~GT.Rape))[1],
            slope = coef(lm(data = df, UCR.Rape~GT.Rape))[2], color = ""),
            show_guide = TRUE) +
scale_color_manual(name = "Regression Line", values=c("gray28")) + theme(legend.position = "top")
#_____________________________________________
#blank plot
```

```
  blankPlot <- ggplot()+geom_blank(aes(1,1)) + cowplot::theme_nothing()

  #______________________________________________
  #put all the cook's distance plots together
  legend_1 <- get_legend(just_to_get_the_legend1)
  legend_2 <- get_legend(just_to_get_the_legend2)
  legend_3 <- get_legend(just_to_get_the_legend3)
  final_plot <- grid.arrange( MVT_text_plot1, burglary_text_plot, larceny_text_plot, rape_text_plot,
                               blankPlot, legend_1, legend_3, legend_2, blankPlot, ncol=6,  nrow = 3,
                              widths = c(0.3,0.5,1,1,0.5,0.3),
                              heights = c(2 , 2 , 0.5),
                              layout_matrix = rbind(c(1,1,1,2,2,2), c(3,3,3,4,4,4),c(5,6,6,7,8,9)),
                             top = textGrob(
                               title,hjust = 0.5, vjust = 0.5,
                               gp=gpar(fontsize = 15,font=2)))
  return(final_plot)
}




#"lm" dma text scatter plot
png(file="Scatter_Plot_Cook_Distance_GT_UCR.png", width = 30, height = 17, unit = "cm",
    res = 200)
plot_texts(final.data.2010_2019, "Figure 5: Scatter Plot and Cook's Distance of Google Trends and UCR C
```

```
## Warning: `show_guide` has been deprecated. Please use `show.legend` instead.

## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `df$GT.Rape` is discouraged. Use `GT.Rape` instead.

## Warning: Use of `df$UCR.Rape` is discouraged. Use `UCR.Rape` instead.

## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `df$GT.Rape` is discouraged. Use `GT.Rape` instead.

## Warning: Use of `df$UCR.Rape` is discouraged. Use `UCR.Rape` instead.

## Warning: Use of `df$UCR.Rape` is discouraged. Use `UCR.Rape` instead.

## Warning: Use of `df$GT.Rape` is discouraged. Use `GT.Rape` instead.

## Warning: Use of `df$UCR.Rape` is discouraged. Use `UCR.Rape` instead.

## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `df$GT.MVT` is discouraged. Use `GT.MVT` instead.

## Warning: Use of `df$UCR.MVT` is discouraged. Use `UCR.MVT` instead.

## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `df$GT.MVT` is discouraged. Use `GT.MVT` instead.

## Warning: Use of `df$UCR.MVT` is discouraged. Use `UCR.MVT` instead.

## Warning: Use of `df$UCR.MVT` is discouraged. Use `UCR.MVT` instead.

## Warning: Use of `df$GT.MVT` is discouraged. Use `GT.MVT` instead.
```

```
## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.
## Warning: Use of `df$GT.Burglary` is discouraged. Use `GT.Burglary` instead.
## Warning: Use of `df$UCR.Burglary` is discouraged. Use `UCR.Burglary` instead.
## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.
## Warning: Use of `df$GT.Burglary` is discouraged. Use `GT.Burglary` instead.
## Warning: Use of `df$UCR.Burglary` is discouraged. Use `UCR.Burglary` instead.
```

```
## Warning: Use of `df$UCR.Burglary` is discouraged. Use `UCR.Burglary` instead.
## Warning: Use of `df$GT.Burglary` is discouraged. Use `GT.Burglary` instead.
## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.
## Warning: Use of `df$GT.Larceny` is discouraged. Use `GT.Larceny` instead.
## Warning: Use of `df$UCR.Larceny` is discouraged. Use `UCR.Larceny` instead.
## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.
## Warning: Use of `df$GT.Larceny` is discouraged. Use `GT.Larceny` instead.
## Warning: Use of `df$UCR.Larceny` is discouraged. Use `UCR.Larceny` instead.
```

```
## Warning: Use of `df$UCR.Larceny` is discouraged. Use `UCR.Larceny` instead.
## Warning: Use of `df$GT.Larceny` is discouraged. Use `GT.Larceny` instead.
## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.
## Warning: Use of `df$GT.Rape` is discouraged. Use `GT.Rape` instead.
## Warning: Use of `df$UCR.Rape` is discouraged. Use `UCR.Rape` instead.
## Warning: Use of `final.data.2010_2019$DMA` is discouraged. Use `DMA` instead.
## Warning: Use of `df$GT.Rape` is discouraged. Use `GT.Rape` instead.
## Warning: Use of `df$UCR.Rape` is discouraged. Use `UCR.Rape` instead.
```

```
## Warning: Use of `df$UCR.Rape` is discouraged. Use `UCR.Rape` instead.
## Warning: Use of `df$GT.Rape` is discouraged. Use `GT.Rape` instead.
## TableGrob (4 x 6) "arrange": 10 grobs
##     z     cells    name                grob
## 1   1 (2-2,1-3) arrange      gtable[layout]
## 2   2 (2-2,4-6) arrange      gtable[layout]
## 3   3 (3-3,1-3) arrange      gtable[layout]
## 4   4 (3-3,4-6) arrange      gtable[layout]
## 5   5 (4-4,1-1) arrange      gtable[layout]
## 6   6 (4-4,2-3) arrange   gtable[guide-box]
## 7   7 (4-4,4-4) arrange   gtable[guide-box]
## 8   8 (4-4,5-5) arrange   gtable[guide-box]
## 9   9 (4-4,6-6) arrange      gtable[layout]
## 10 10 (1-1,1-6) arrange text[GRID.text.502]
```

```
dev.off()
```

```
## pdf
##   2
```

```
#Test the residual of GT and UCR, and test what causes the residuals
#write a function to process residual test
test_ucr_gt_residuals <- function(data, GT, UCR, year){
  data2 <- drop_na(data)
  data2_lm <- lm(data = data2, data2[[GT]] ~ data2[[UCR]])
  data2$residuals <- data2_lm$residuals
  data2_resdiual <- lm(data = data2, residuals ~ Concentrated.Disadvantage.Index +
                       #Percentage.of.Unemployed +
                       #Percentage.of.Female.Headed.Family +
                       #Percentage.of.Poverty +
                       #Percentage.of.MoveIn +
                       #Percentage.of.Renter +
                       Mobility.Index +
                       Heterogeneity.Index +
                       Percentage.of.Foreign.Born +
                       #Percentage.of.Black +
                       #Percentage.of.Hispanic +
                       #Percentage.of.White +
                       Percentage.of.Divorced +
                       #Less.than.High.School +
                       Percentage.of.Young.Males +
                       Drug.Mortality.Rate +
                       #Law.Enforce.per.Thousand +
                       Population.logged +
                       Internet.Usage.HH +
                       Median.Vehicle.HH)
  return(summary(data2_resdiual))
}
test_ucr_gt_residuals(final.data.2010_2019, "GT.MVT", "UCR.MVT", "2010_2019")
```

```
##
## Call:
## lm(formula = residuals ~ Concentrated.Disadvantage.Index + Mobility.Index +
##     Heterogeneity.Index + Percentage.of.Foreign.Born + Percentage.of.Divorced +
##     Percentage.of.Young.Males + Drug.Mortality.Rate + Population.logged +
##     Internet.Usage.HH + Median.Vehicle.HH, data = data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.234  -4.123   0.130   3.426  25.043
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -33.46762  121.40124  -0.276    0.783
## Concentrated.Disadvantage.Index  -0.69306    0.49105  -1.411    0.161
## Mobility.Index                    0.65450    0.72504   0.903    0.369
## Heterogeneity.Index               9.01888    8.79350   1.026    0.308
## Percentage.of.Foreign.Born        3.06737   16.55741   0.185    0.853
## Percentage.of.Divorced          161.01836   90.69358   1.775    0.079 .
## Percentage.of.Young.Males       373.63701  267.53316   1.397    0.166
## Drug.Mortality.Rate               0.01422    0.01511   0.941    0.349
## Population.logged                 2.22126    1.43511   1.548    0.125
## Internet.Usage.HH               -25.32029  166.77632  -0.152    0.880
## Median.Vehicle.HH               -10.31189    8.88521  -1.161    0.249
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.639 on 96 degrees of freedom
## Multiple R-squared:  0.1715, Adjusted R-squared:  0.08519
## F-statistic: 1.987 on 10 and 96 DF,  p-value: 0.04296
```

```
test_ucr_gt_residuals(final.data.2010_2019, "GT.Rape", "UCR.Rape", "2010_2019")
```

```
##
## Call:
## lm(formula = residuals ~ Concentrated.Disadvantage.Index + Mobility.Index +
##     Heterogeneity.Index + Percentage.of.Foreign.Born + Percentage.of.Divorced +
##     Percentage.of.Young.Males + Drug.Mortality.Rate + Population.logged +
##     Internet.Usage.HH + Median.Vehicle.HH, data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.7263  -4.6550  -0.4919   4.4464  29.7986
##
## Coefficients:
##                                   Estimate  Std. Error t value    Pr(>|t|)
## (Intercept)                     228.093613  123.974106   1.840     0.06888 .
## Concentrated.Disadvantage.Index   1.801064    0.501455   3.592     0.00052 ***
## Mobility.Index                    0.419998    0.740411   0.567     0.57187
## Heterogeneity.Index               2.741767    8.979863   0.305     0.76078
## Percentage.of.Foreign.Born      -95.324958   16.908314  -5.638 0.000000173 ***
## Percentage.of.Divorced          -65.746735   92.615654  -0.710     0.47949
## Percentage.of.Young.Males       -31.204968  273.203016  -0.114     0.90930
## Drug.Mortality.Rate               0.003237    0.015429   0.210     0.83426
## Population.logged                 1.728525    1.465520   1.179     0.24113
## Internet.Usage.HH              -285.374338  170.310829  -1.676     0.09707 .
## Median.Vehicle.HH                -2.417323    9.073514  -0.266     0.79049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.801 on 96 degrees of freedom
## Multiple R-squared:  0.5435, Adjusted R-squared:  0.4959
## F-statistic: 11.43 on 10 and 96 DF,  p-value: 0.000000000001141
```

```
test_ucr_gt_residuals(final.data.2010_2019, "GT.Burglary", "UCR.Burglary", "2010_2019")
```

```
##
## Call:
## lm(formula = residuals ~ Concentrated.Disadvantage.Index + Mobility.Index +
##     Heterogeneity.Index + Percentage.of.Foreign.Born + Percentage.of.Divorced +
##     Percentage.of.Young.Males + Drug.Mortality.Rate + Population.logged +
##     Internet.Usage.HH + Median.Vehicle.HH, data = data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.735  -6.715   1.646   7.523  25.220
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                       -55.14177  189.46268  -0.291   0.7716
## Concentrated.Disadvantage.Index     0.14169    0.76635   0.185   0.8537
## Mobility.Index                      0.72136    1.13153   0.638   0.5253
## Heterogeneity.Index               -10.39066   13.72342  -0.757   0.4508
## Percentage.of.Foreign.Born         12.15999   25.84003   0.471   0.6390
## Percentage.of.Divorced             89.62365  141.53932   0.633   0.5281
## Percentage.of.Young.Males        -168.31141  417.52086  -0.403   0.6878
## Drug.Mortality.Rate                -0.04631    0.02358  -1.964   0.0524 .
## Population.logged                  -2.35113    2.23967  -1.050   0.2965
## Internet.Usage.HH                 159.28920  260.27650   0.612   0.5420
## Median.Vehicle.HH                 -12.89051   13.86654  -0.930   0.3549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.92 on 96 degrees of freedom
## Multiple R-squared:  0.1119, Adjusted R-squared:  0.01941
## F-statistic:  1.21 on 10 and 96 DF,  p-value: 0.2946
```

```r
test_ucr_gt_residuals(final.data.2010_2019, "GT.Larceny", "UCR.Larceny", "2010_2019")
```

```
##
## Call:
## lm(formula = residuals ~ Concentrated.Disadvantage.Index + Mobility.Index +
##     Heterogeneity.Index + Percentage.of.Foreign.Born + Percentage.of.Divorced +
##     Percentage.of.Young.Males + Drug.Mortality.Rate + Population.logged +
##     Internet.Usage.HH + Median.Vehicle.HH, data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.2437  -3.0494  -0.2753   2.8537  28.3644
##
## Coefficients:
##                                   Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                        5.75619  105.62984   0.054   0.95665
## Concentrated.Disadvantage.Index    0.58782    0.42726   1.376   0.17208
## Mobility.Index                     1.49173    0.63085   2.365   0.02006 *
## Heterogeneity.Index                9.57708    7.65113   1.252   0.21371
## Percentage.of.Foreign.Born        59.51359   14.40642   4.131 0.0000771 ***
## Percentage.of.Divorced           170.75301   78.91146   2.164   0.03296 *
## Percentage.of.Young.Males        463.44062  232.77758   1.991   0.04933 *
## Drug.Mortality.Rate                0.03599    0.01315   2.738   0.00737 **
## Population.logged                  0.78995    1.24867   0.633   0.52848
## Internet.Usage.HH               -149.58667  145.11019  -1.031   0.30520
## Median.Vehicle.HH                 20.55680    7.73092   2.659   0.00918 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.647 on 96 degrees of freedom
## Multiple R-squared:  0.5445, Adjusted R-squared:  0.497
## F-statistic: 11.47 on 10 and 96 DF,  p-value: 0.000000000001037
```

```r
plot_correlation <- function(df, title){
  mvtp <- round(cor.test(df$GT.MVT,df$UCR.MVT)$p.value,3)
  burglaryp <- round(cor.test(df$GT.Burglary,df$UCR.Burglary)$p.value,3)
  larcenyp <- round(cor.test(df$GT.Larceny,df$UCR.Larceny)$p.value,3)
```

```r
  rapep <- round(cor.test(df$GT.Rape,df$UCR.Rape)$p.value,3)
  mvtp <- ifelse(mvtp == 0, 0.001, mvtp)
  burglaryp <- ifelse(burglaryp == 0, 0.001,burglaryp )
  larcenyp <- ifelse(larcenyp == 0, 0.001,larcenyp )
  rapep <- ifelse(rapep == 0, 0.001,rapep )
  df <- drop_na(df)
  gt_ucr <- ggplot(df) +
  geom_point(aes(x = GT.MVT, y = UCR.MVT), alpha = 0.3, size = 1, color = "brown") +
  geom_smooth(aes(x = GT.MVT, y = UCR.MVT), method = "lm") +
  geom_text(x=40, y=70, label = paste(
    "r = ", sprintf("%.3f", cor(df$GT.MVT,df$UCR.MVT,use = "complete.obs")),
    ", p-value < ", mvtp)) +
  labs(title = "Motor Vehicle Theft") +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank())
  xlim(0, 100)

  gt_ucr_burg <- ggplot(df)+
    geom_point(aes(x = GT.Burglary, y = UCR.Burglary), alpha = 0.3, size = 1) +
    geom_smooth(aes(x = GT.Burglary, y = UCR.Burglary), method = "lm")+
    geom_text(x=40, y=76, label = paste("r = ",sprintf("%.3f",cor(df$GT.Burglary,df$UCR.Burglary,use =
    ", p-value < ", burglaryp))+
    labs(title = "Burglary") +
    theme(axis.title.y = element_blank(), axis.title.x = element_blank())
    xlim(20, 100)

  gt_ucr_lar <- ggplot(df)+
    geom_point(aes(x = GT.Larceny, y = UCR.Larceny), alpha = 0.3, size = 1, color = "red")+
    geom_smooth(aes(x = GT.Larceny, y = UCR.Larceny), method = "lm")+
    geom_text(x=50, y=82, label = paste("r = ",sprintf("%.3f", cor(df$GT.Larceny,df$UCR.Larceny,use = "
    ", p-value < ", larcenyp))+
    labs(title = "Larceny") +
    theme(axis.title.y = element_blank(), axis.title.x = element_blank())
    xlim(20, 100)

  gt_ucr_rape <- ggplot(df)+
    geom_point(aes(x = GT.Rape, y = UCR.Rape), alpha = 0.3, size = 1, color = "darkgreen")+
    geom_smooth(aes(x = GT.Rape, y = UCR.Rape), method = "lm") +
    geom_text(x=55, y=80, label = paste("r = ",sprintf("%.3f", cor(df$GT.Rape, df$UCR.Rape, use = "comp
    ", p-value < ", rapep))+
    labs(title = "Rape") +
    theme(axis.title.y = element_blank(), axis.title.x = element_blank())
    xlim(20, 100)

  grid.arrange(gt_ucr, gt_ucr_burg, gt_ucr_lar, gt_ucr_rape,
               ncol=2, top = textGrob(
               title, hjust = 0.5, vjust = 0.5,
               gp=gpar(fontsize=16,font=2)),
             left = textGrob("UCR Crime Rates", rot = 90, vjust = 1,
                             gp=gpar(fontsize = 15,fontface="bold")),
             bottom = textGrob("Google Trends Crime Estimates",
                               gp=gpar(fontsize = 15,fontface="bold")))
}
```

```r
png(file="correlation_10_19_plot.png", width = 30, height = 17, unit = "cm",
    res = 100)
plot_correlation(final.data.2010_2019,
                 "Figure 1, UCR, and Google Trends Correlation Scatter Plots")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
library(MASS)

#Cook's Distance Plot Function
cook_distance <- function(lm_model, text_name_column, crimetype){
  cooks_data <- data.frame(cooks.distance(lm_model), hatvalues(lm_model), studres(lm_model), text_name_c
  colnames(cooks_data) <- cbind("cooks_dist", "hat_values", "studres", "DMA")

  ## Plot

  ggplot(cooks_data, aes(x = hat_values, y = studres,
         size = cooks_dist,
         col = cooks_dist > 4/(nrow(cooks_data) - 1 - 1),
         label = cooks_data$DMA)) +
    geom_point() +
    geom_text(vjust = 2)  +
    geom_vline(xintercept = 2 * (lm_model$rank - 1 + 1)/nrow(cooks_data),
               linetype = 2) +
    geom_hline(yintercept = c(-4, 4), linetype = 2) +
    scale_color_manual("High\nInfluence",
                       values = c("TRUE" = "gold2",
                                  "FALSE" = "gray97")) +
    scale_size("Cook's\nDistance") + theme_bw() +
    ggtitle(paste(crimetype)) +
    theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
    xlim(-0.28, 0.45) +
    ylim(-4.8, 4.8)
}

#Arrange Cook's Distance Plot Function
test_lm <- lm(data = final.data.2010_2019, UCR.MVT ~ GT.MVT)
test_lm$rank
```

```
## [1] 2
```

```r
arrange_cook <- function(df, title){
  df <- drop_na(df)
  ucr_gt_mvt_lm <- lm(data = df, UCR.MVT ~ GT.MVT)
  ucr_gt_mvt_cook_plot <- cook_distance(ucr_gt_mvt_lm, df$DMA, "Motor Vehicle Theft")+
    theme(legend.position="none")
```

```r
  ucr_gt_larceny_lm <- lm(data = df, UCR.Larceny ~ GT.Larceny)
  ucr_gt_larceny_cook_plot <- cook_distance(ucr_gt_larceny_lm, df$DMA, "Larceny")+
    theme(legend.position="none")

  ucr_gt_burglary_lm <- lm(data = df, UCR.Burglary ~ GT.Burglary)
  ucr_gt_burglary_cook_plot <- cook_distance(ucr_gt_burglary_lm, df$DMA, "Burglary")+
    theme(legend.position="none")

  ucr_gt_rape_lm <- lm(data = df, UCR.Rape ~ GT.Rape)
  ucr_gt_rape_cook_plot <- cook_distance(ucr_gt_rape_lm, df$DMA, "Rape")+
    theme(legend.position="none")

  legend <- get_legend(cook_distance(ucr_gt_rape_lm, df$DMA, "Rape"))

  final_plot <- grid.arrange(arrangeGrob(ucr_gt_mvt_cook_plot,
                           ucr_gt_burglary_cook_plot,
                           ucr_gt_larceny_cook_plot,
                           ucr_gt_rape_cook_plot, nrow = 2),
                           legend,
                           widths=c(8, 1),
                           heights=c(200, 1),
                           ncol=2, top = textGrob(title, hjust = 0.5 , vjust = 0.5,
                                                  gp=gpar(fontsize=16,font=2)),
                           left = textGrob("Studentized Residuals", rot = 90, vjust = 1,
                             gp=gpar(fontsize = 15,fontface="bold")),
                           bottom = textGrob("Hat Values", hjust = 0.9,
                             gp=gpar(fontsize = 15,fontface="bold")))
  return(final_plot)
}

png(file="cook_10_19_plot.png", width = 30, height = 17, unit = "cm",
    res = 100)
arrange_cook(final.data.2010_2019,
             "Figure 6: Influence Plot of Google Trends and UCR Crimes, 2010 to 2019")
```

```
## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.

## Warning: Use of `cooks_data$DMA` is discouraged. Use `DMA` instead.
```

```
## TableGrob (4 x 3) "arrange": 5 grobs
##   z     cells    name              grob
## 1 1 (2-2,2-2) arrange    gtable[arrange]
## 2 2 (2-2,3-3) arrange  gtable[guide-box]
## 3 3 (1-1,2-3) arrange text[GRID.text.929]
## 4 4 (4-4,2-3) arrange text[GRID.text.930]
## 5 5 (1-4,1-1) arrange text[GRID.text.931]
```

```
dev.off()
```

```
## pdf
##   2
```

```
#VIF Test, VIF need to be less than 4
library(caret)
```

```
##
## Attaching package: 'caret'

## The following objects are masked from 'package:Metrics':
##
##     precision, recall

## The following object is masked from 'package:survival':
##
##     cluster

## The following object is masked from 'package:purrr':
##
##     lift
```

```
stargazer(cbind(car::vif(GT_MVT_lm), car::vif(UCR_MVT_lm)),cbind(
                car::vif(GT_Burglary_lm),
                car::vif(UCR_Burglary_lm),
                car::vif(GT_Larceny_lm),
                car::vif(UCR_Larceny_lm),
                car::vif(GT_Rape_lm),
                car::vif(UCR_Rape_lm)))
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Apr 07, 2021 - 12:33:13 AM

Table 4

Concentrated.Disadvantage.Index	3.834	3.834
Mobility.Index	2.481	2.481
Heterogeneity.Index	2.994	2.994
Percentage.of.Foreign.Born	2.623	2.623
Percentage.of.Divorced	2.187	2.187
Percentage.of.Young.Males	3.003	3.003
Drug.Mortality.Rate	1.514	1.514
Population.logged	2.512	2.512
Internet.Usage.HH	4.695	4.695
Median.Vehicle.HH	2.357	2.357

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Apr 07, 2021 - 12:33:13 AM

Table 5

Concentrated.Disadvantage.Index	3.485	3.485	3.485	3.485	3.485	3.485
Mobility.Index	2.435	2.435	2.435	2.435	2.435	2.435
Heterogeneity.Index	2.956	2.956	2.956	2.956	2.956	2.956
Percentage.of.Foreign.Born	2.377	2.377	2.377	2.377	2.377	2.377
Percentage.of.Divorced	2.184	2.184	2.184	2.184	2.184	2.184
Percentage.of.Young.Males	2.579	2.579	2.579	2.579	2.579	2.579
Drug.Mortality.Rate	1.512	1.512	1.512	1.512	1.512	1.512
Population.logged	2.493	2.493	2.493	2.493	2.493	2.493
Internet.Usage.HH	4.093	4.093	4.093	4.093	4.093	4.093

```
library(psych)
library("GPArotation")
efa_gt_2010_2015 <- fa(final.data.2010_2019[c(12:15,19,23,29:31)], nfactors = 3,rotate = "oblimin",fm="
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done

## In smc, smcs < 0 were set to .0

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done

## In smc, smcs < 0 were set to .0

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done

## In smc, smcs < 0 were set to .0

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done
```

```
print(efa_gt_2010_2015$loadings,cutoff = 0.3)
```

```
##
## Loadings:
##                                  MR1    MR2    MR3
## Percentage.of.MoveIn              0.933
## Percentage.of.Renter              0.566         0.466
## Mobility.Index                    0.948
## Concentrated.Disadvantage.Index                 0.891
## Heterogeneity.Index                      0.558  0.425
## Population.logged                        0.828
## Drug.Mortality.Rate
## Law.Enforce.per.Thousand                        0.450
## Internet.Usage.HH                        0.749
##
##                  MR1   MR2   MR3
## SS loadings    2.319 1.715 1.529
```

```
## Proportion Var 0.258 0.191 0.170
## Cumulative Var 0.258 0.448 0.618
```

```r
print("Eigen values of the common factor solution: ")
```

```
## [1] "Eigen values of the common factor solution: "
```

```r
print(efa_gt_2010_2015$values,cutoff = 0.3)
```

```
## [1]  3.04793545  1.56689262  1.13921929  0.25697012  0.07417425  0.01599903
## [7] -0.07434362 -0.11656168 -0.16624016
```

```r
efa_gt_2010_2015 <- fa(final.data.2010_2019[c(12:15,19,23,29:31)], nfactors = 6,rotate = "oblimin",fm="r
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done

## In smc, smcs < 0 were set to .0

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done

## In smc, smcs < 0 were set to .0

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done

## In smc, smcs < 0 were set to .0

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done

## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done
```

```r
print(efa_gt_2010_2015$loadings,cutoff = 0.3)
```

```
##
## Loadings:
##                                MR6    MR1    MR2    MR4    MR3    MR5
## Percentage.of.MoveIn            1.007
## Percentage.of.Renter                   0.869
## Mobility.Index                  0.545  0.535
## Concentrated.Disadvantage.Index                              0.889
## Heterogeneity.Index                                                 0.905
## Population.logged                             0.955
## Drug.Mortality.Rate                                  -0.387
## Law.Enforce.per.Thousand                                     0.412
## Internet.Usage.HH                                     0.921
##
##                  MR6   MR1   MR2   MR4   MR3   MR5
## SS loadings    1.328 1.112 1.073 1.012 0.974 0.951
## Proportion Var 0.148 0.124 0.119 0.112 0.108 0.106
## Cumulative Var 0.148 0.271 0.390 0.503 0.611 0.717
```

```r
print("Eigen values of the common factor solution: ")
```

```
## [1] "Eigen values of the common factor solution: "
```

```r
print(efa_gt_2010_2015$values,cutoff = 0.3)
```

```
## [1]  3.17466991648  1.80261149617  1.25701752678  0.42431843669  0.26809444818
## [6]  0.17553334599  0.00009536098  0.00000686875 -0.00285073714
```

```r
efa_gt_2010_2015 <- fa(final.data.2010_2019[c(12:15,19,23,29:31)], nfactors = 9,rotate = "oblimin",fm="r
```

```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done
## In smc, smcs < 0 were set to .0
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done
## In smc, smcs < 0 were set to .0
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was done
## In smc, smcs < 0 were set to .0
## Warning in GPFoblq(L, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPFoblq. 1000 iterations used.
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done
## In factor.scores, the correlation matrix is singular, an approximation is used
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done
```

```
print(efa_gt_2010_2015$loadings,cutoff = 0.3)
```

```
##
## Loadings:
##                                  MR1    MR6    MR2    MR8    MR3    MR5
## Percentage.of.MoveIn                    1.000
## Percentage.of.Renter             0.957
## Mobility.Index                   0.591  0.512
## Concentrated.Disadvantage.Index                              0.872
## Heterogeneity.Index
## Population.logged                              0.946
## Drug.Mortality.Rate
## Law.Enforce.per.Thousand                                            0.662
## Internet.Usage.HH                                     0.893
##                                  MR4    MR7    MR9
## Percentage.of.MoveIn
## Percentage.of.Renter
## Mobility.Index
## Concentrated.Disadvantage.Index
## Heterogeneity.Index                     0.321
## Population.logged
## Drug.Mortality.Rate              0.566
## Law.Enforce.per.Thousand
## Internet.Usage.HH
##
##                  MR1   MR6   MR2   MR8   MR3   MR5   MR4   MR7   MR9
## SS loadings    1.292 1.267 0.953 0.858 0.775 0.518 0.341 0.110 0.000
## Proportion Var 0.144 0.141 0.106 0.095 0.086 0.058 0.038 0.012 0.000
## Cumulative Var 0.144 0.284 0.390 0.486 0.572 0.629 0.667 0.679 0.679
```

```
print("Eigen values of the common factor solution: ")
```

```
## [1] "Eigen values of the common factor solution: "
```

```
print(efa_gt_2010_2015$values,cutoff = 0.3)
```

```
## [1]  3.150955146  1.717284098  1.230393541  0.437435395  0.257475738
## [6]  0.091610261  0.065746794  0.004782341 -0.002850567
```

```r
res <- cor(final.data.2010_2019[3:10,12:31])

write.csv(round(res, 3),file="GT_ACS_cormatrix.csv")
```

```r
library(ggplot2)
library(gridExtra)

h1 <- ggplot(final.data.2010_2019, aes(x=GT.MVT)) +
  geom_histogram() +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
  labs(title = "Motor Vehicle Theft") +
  xlim(0,100)

h2 <- ggplot(final.data.2010_2019, aes(x=GT.Burglary))+
  geom_histogram() +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
  labs(title = "Burglary") +
  xlim(0,100)

h3 <- ggplot(final.data.2010_2019, aes(x=GT.Larceny)) +
  geom_histogram() +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
  labs(title = "Larceny") +
  xlim(0,100)

h4 <- ggplot(final.data.2010_2019, aes(x=GT.Rape)) +
  geom_histogram() +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
  labs(title = "Rape") +
  xlim(0,100)

gt_hist_title <- "Figure 2, Histogram of Google Trends Crime Estimates"

png(file="hist_GT_2010_2019.png", width = 30, height = 17, unit = "cm",
    res = 200)
grid.arrange(h1, h2, h3, h4,
          ncol = 2, nrow = 2,
          top = textGrob(gt_hist_title, hjust = 0.5 , vjust = 0.5,
                         gp=gpar(fontsize=15,font=2)),
          left = textGrob("Counts", rot = 90, vjust = 1, gp=gpar(fontsize = 15,fontface="bold")),
          bottom = textGrob("Google Trends Crime Estimates Distributions",
                            gp=gpar(fontsize = 15,fontface="bold")))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing missing values (geom_bar).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing missing values (geom_bar).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing missing values (geom_bar).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
h5 <- ggplot(final.data.2010_2019, aes(x=UCR.MVT)) +
  geom_histogram() +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
  labs(title = "Motor Vehicle Theft") +
  xlim(0,100)

h6 <- ggplot(final.data.2010_2019, aes(x=UCR.Burglary)) +
  geom_histogram() +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
  labs(title = "Burglary") +
  xlim(0,100)

h7 <- ggplot(final.data.2010_2019, aes(x=UCR.Larceny)) +
  geom_histogram() +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
  labs(title = "Larceny") +
  xlim(0,100)

h8 <- ggplot(final.data.2010_2019, aes(x=UCR.Rape))  +
  geom_histogram() +
  theme(axis.title.y = element_blank(), axis.title.x = element_blank()) +
  labs(title = "Rape") +
  xlim(0,100)

ucr_hist_title <- "Figure 3, Histogram of Uniform Crime Report Crime Rates"

png(file="hist_UCR_2010_2019.png", width = 30, height = 17, unit = "cm",
    res = 200)
grid.arrange(h5, h6, h7, h8, ncol = 2,
          top = textGrob(ucr_hist_title, hjust = 0.5, vjust = 0.5,
                         gp=gpar(fontsize = 15,font=2)),
          left = textGrob("Counts", rot = 90, vjust = 1, gp=gpar(fontsize = 15,fontface="bold")),
          bottom = textGrob("Uniform Crime Raport Crime Rates Distributions",
                            gp=gpar(fontsize = 15,fontface="bold")))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing missing values (geom_bar).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing missing values (geom_bar).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing missing values (geom_bar).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing missing values (geom_bar).
```

```
dev.off()
```

```
## pdf
##   2
```

```
write.csv(final.data.2010_2019,file="final_data_2010_2019.csv")
```