CS02 - Right-To-Carry

Yohan Kim, Ming Qiu, Kevin Lam, Khiem Pham

Introduction

Right to Carry Laws refer to laws that specify how citizens are allowed to carry concealed handguns when they're away from home without a permit.[1] These laws are state-dependent and can vary greatly from state to state.[1] There are 2 analyses that sought to explain the relation between RTC laws and violent crime: one by Mustard and Lott (1996), and the other by Donohue et al (2017).[2] For the Lott analysis, they concluded that RTC laws decreases the amount of violent crime.[2] In contrast, the Donohue analysis concluded that RTC laws increased the amount of violent crime.[2] One of the reasons for these contradicting conclusions is multicollinearity, which occurs when 2 or more seemingly independent variables are highly related to one another in a multiple regression model.[2] This could lead to inaccurate conclusions of the relationships estimated in an analysis.[3] For this case study, we'll investigate how multicollinearity affects the results we receive from our linear regression models.

Load packages

```
library(OCSdata)
library(tidyverse)
library(pdftools)
library(readxl)
library(tidyverse)
library(skimr)
library(ggrepel)
library(plm)
library(car)
library(rsample)
library(GGally)
library(ggcorrplot)
library(broom)
```

Question

- What is the effect of multicollinearity on coefficient estimates from linear regression models when analyzing right to carry laws and violence rates?

The Data

We'll be using the following datasets: Demographic (from Census), Crime (from FBI), RTC variables (from State session laws), Police Staffing (from FBI), Poverty Rate (from census), and Unemployment Rate (from Bureau Labor of Statistics).

```
#load_raw_data("ocs-bp-RTC-wrangling", outpath='.') # Only need to do once since it is just downloading the data
```

Data Import

Demographic data

```
#OCSdata::load_raw_data("ocs-bp-RTC-wrangling", outpath = '.')

dem_77_79 <- read_csv("data/raw/Demographics/Decade_1970/pe-19.csv", skip = 5)
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_number(),
##   `Year of Estimate` = col_double(),
##   `FIPS State Code` = col_character(),
##   `State Name` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
```

```
dem_80_89 <- list.files(recursive = TRUE,
                  path = "data/raw/Demographics/Decade_1980",
                  pattern = "*.csv",
                  full.names = TRUE) |>
  purrr::map(~read_csv(., skip=5))
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
##
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
##
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
##
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
##
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
##
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
##
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
##
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
```

```
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
##
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   `FIPS State and County Codes` = col_character(),
##   `Race/Sex Indicator` = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
```

```
dem_90_99 <- list.files(recursive = TRUE,
                  path = "data/raw/Demographics/Decade_1990",
                  pattern = "*.txt",
                  full.names = TRUE) |>
  map(~read_table2(., skip = 14))
```

```
## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female'
## => 'Female_1' [7], 'Male' => 'Male_2' [8], 'Female' => 'Female_2' [9], 'Male' =>
## 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4' [12], 'Female' =>
## 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' =>
## 'Male_6' [16], 'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' =>
## 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh90.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh91.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh92.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh93.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh94.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh95.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh96.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh97.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh98.txt'

## Warning: Duplicated column names deduplicated: 'Male' => 'Male_1' [6], 'Female' => 'Female_1' [7], 'Male' => '
Male_2' [8], 'Female' => 'Female_2' [9], 'Male' => 'Male_3' [10], 'Female' => 'Female_3' [11], 'Male' => 'Male_4'
[12], 'Female' => 'Female_4' [13], 'Male' => 'Male_5' [14], 'Female' => 'Female_5' [15], 'Male' => 'Male_6' [16],
'Female' => 'Female_6' [17], 'Male' => 'Male_7' [18], 'Female' => 'Female_7' [19]
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Year = col_double(),
##   e = col_character(),
##   Age = col_double(),
##   Male = col_double(),
##   Female = col_double(),
##   Male_1 = col_double(),
##   Female_1 = col_double(),
##   Male_2 = col_double(),
##   Female_2 = col_double(),
##   Male_3 = col_double(),
##   Female_3 = col_double(),
##   Male_4 = col_double(),
##   Female_4 = col_double(),
##   Male_5 = col_double(),
##   Female_5 = col_double(),
##   Male_6 = col_double(),
##   Female_6 = col_double(),
##   Male_7 = col_double(),
##   Female_7 = col_double()
## )
```

```
## Warning: 1 parsing failure.
## row col   expected    actual                                            file
##   1  -- 19 columns 1 columns 'data/raw/Demographics/Decade_1990/sasrh99.txt'
```

```
dem_00_10 <- list.files(recursive = TRUE,
                  path = "data/raw/Demographics/Decade_2000",
                  pattern = "*.csv",
                   full.names = TRUE) |>
    map(~read_csv(.))
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_double(),
##   NAME = col_character()
## )
## ℹ Use `spec()` for the full column specifications.
```

```
load("data/wrangled/DONOHUE_simulations.rda")
load("data/wrangled/LOTT_simulations.rda")
```

```
# read State Fips data from Excel file
STATE_FIPS <- readxl::read_xls("data/raw/State_FIPS_codes/state-geocodes-v2014.xls", skip = 5)
```

Police Staffing data

```
# read Police Staffing data from csv file
ps_data <- read_csv("https://campuspro-uploads.s3.us-west-2.amazonaws.com/0a1a45c4-dd64-4f91-a4e1-3847dbdfba5a/94
fc6045-a323-42b9-bd5e-2e225562d875/pe_1960_2018.csv",
                    col_types =  cols(male_total_ct = col_double(),
                                      female_total_ct = col_double()))
```

```
## Warning: 2874273 parsing failures.
##  row               col           expected actual
file
## 4115 female_officer_ct 1/0/T/F/TRUE/FALSE   8    'https://campuspro-uploads.s3.us-west-2.amazonaws.com/0a1a45c
4-dd64-4f91-a4e1-3847dbdfba5a/94fc6045-a323-42b9-bd5e-2e225562d875/pe_1960_2018.csv'
## 4115 officer_ct        1/0/T/F/TRUE/FALSE   37   'https://campuspro-uploads.s3.us-west-2.amazonaws.com/0a1a45c
4-dd64-4f91-a4e1-3847dbdfba5a/94fc6045-a323-42b9-bd5e-2e225562d875/pe_1960_2018.csv'
## 4115 civilian_ct       1/0/T/F/TRUE/FALSE   3    'https://campuspro-uploads.s3.us-west-2.amazonaws.com/0a1a45c
4-dd64-4f91-a4e1-3847dbdfba5a/94fc6045-a323-42b9-bd5e-2e225562d875/pe_1960_2018.csv'
## 4115 total_pe_ct       1/0/T/F/TRUE/FALSE   40   'https://campuspro-uploads.s3.us-west-2.amazonaws.com/0a1a45c
4-dd64-4f91-a4e1-3847dbdfba5a/94fc6045-a323-42b9-bd5e-2e225562d875/pe_1960_2018.csv'
## 4115 pe_ct_per_1000    1/0/T/F/TRUE/FALSE   2.00 'https://campuspro-uploads.s3.us-west-2.amazonaws.com/0a1a45c
4-dd64-4f91-a4e1-3847dbdfba5a/94fc6045-a323-42b9-bd5e-2e225562d875/pe_1960_2018.csv'
## .... ................. .................. ...... ............................................................
.............................................................................................
## See problems(...) for more details.
```

Unemployment data

1. We first use the function “list.files” to get file paths for all xlsx files from Unemployment folder. Next, we read the xlsx files at once then save them as a list of tibbles.
2. In order to get the name of the state each data is associated with, we read the same xlsx files again and search for the info within the specific cells then convert the list of state names to a vector and store it in ue_rate_names. We later assign each state name to its associated tibble.

```
#1
ue_rate_data <- list.files(recursive = TRUE,
                            path = "data/raw/Unemployment", # files are located in data/raw/Unemployment
                            pattern = "*.xlsx",             # look for xlsx files
                            full.names = TRUE) |>           # get full name
  map(~read_xlsx(., skip = 10))                             # read all files in once, skip the first 10 lines whi
ch are some info about the data

#2
ue_rate_names <- list.files(recursive = TRUE,
                            path = "data/raw/Unemployment",
                            pattern = "*.xlsx",
                            full.names = TRUE) %>%
  map(~read_xlsx(., range = "B4:B6")) %>% # look for cell B4-B6
  map(., c(1,2)) |>
  unlist() # convert the list to a vector
names(ue_rate_data) <- ue_rate_names # name each tibble
```

RTC data

- The code below reads the data from the pdf file “w23510.pd” using the function pdf_text() then store the data in variable DAWpaper.

```
DAWpaper <- pdf_text("data/raw/w23510.pdf")
```

Crime data

```
crime_data <- read_lines("data/raw/Crime/CrimeStatebyState.csv",
                         skip = 2, # we skip 2 lines since thow two lines are just comments of the dataset
                         skip_empty_rows = TRUE)
```

Read the data using read_lines function due to this function being more useful when handling spaces and / in colnames.

poverty data

```
poverty_rate_data <- readxl::read_xls("data/raw/Poverty/hstpov21.xls", skip = 2) # we skip 2 lines since thow two
lines are just comments of the dataset
```

```
## New names:
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * `` -> ...6
```

```
poverty_rate_data
```

```
## # A tibble: 2,177 × 6
##    `NOTE: Number in thous… ...2  ...3   ...4         ...5         ...6
##    <chr>                   <chr> <chr>  <chr>        <chr>        <chr>
##  1 2018                    <NA>  <NA>    <NA>        <NA>          <NA>
##  2 STATE                   Total Number "Standard\n… Percent      "Standard\ner…
##  3 Alabama                 4877  779    "65"         16           "1.3"
##  4 Alaska                  720   94     "9"          13.1         "1.2"
##  5 Arizona                 7241  929    "80"         12.80000000… "1.1000000000…
##  6 Arkansas                2912  462    "38"         15.9         "1.3"
##  7 California              39150 4664   "184"        11.9         "0.5"
##  8 Colorado                5739  521    "51"         9.099999999… "0.9000000000…
##  9 Connecticut             3413  348    "43"         10.19999999… "1.3"
## 10 Delaware                976   72     "9"          7.400000000… "1"
## # … with 2,167 more rows
```

Read the data using readxl function. This function allows us to read the function with automatically set the column be the first row of the xls file. We skip 4 rows since the first 4 rows are just description of the dataset.

Part III: Data Wrangling

Poverty

```
colnames(poverty_rate_data) <- c("STATE", "Total", "Number", "Number_se",
                                 "Percent", "Percent_se")
poverty_rate_data <- poverty_rate_data |>
  filter(STATE != "STATE") |>
  mutate(length_state = map_dbl(STATE, str_length)) |> # determine how long string in "STATE" column is
  filter(length_state < 100) |> # filter to only include possible state lengths
  mutate(STATE = str_replace(STATE, pattern = "D.C.",
                              replacement = "District of Columbia" ))
```

Above code is used to rename the columns. Some columns were having wrong column names (e.g. Standard error…4) due to readxl function error The code above is used to mutate the column type of it. It was originally set as character, which can cause error when performing calculation analysis with it.

```
year_values <- poverty_rate_data |>
  filter(str_detect(STATE, "[:digit:]")) |>
  distinct(STATE)
year_values <- rep(pull(year_values, STATE), each = 52) # repeat values from STATE column 52 times each
poverty_rate_data <- poverty_rate_data |>
  mutate(year_value = year_values) |>
  select(-length_state) |>
  filter(str_detect(STATE, "[:alpha:]"))
```

Above code is removing non-ordinary rows then add new column “Year” that represents each year for a row. As we can see from the original excel sheet, some rows show the year with the same column name. We get rid of that in the first part. Next, we need to show which year this row was in since we removed that. Thus we create a vector that consists of years, 51 same years with decreasing order. Reason it is 51 because we have total 50 states plus 1 district of columbia.

```
poverty_rate_data <- poverty_rate_data |>
  filter(year_value != "2017") |> # Remove 2017 since we already have same (but different value) dataset
  filter(year_value != "2013 (18)") |> # Remove 2013 (18) since we already have same (but different value) datase
t
  mutate(YEAR = str_sub(year_value, start = 1, end=4)) |>
  select(-c(Number, Number_se, Percent_se, Total, year_value)) |>
  rename("VALUE" = Percent) |>
  mutate(VARIABLE = "Poverty_rate",
         YEAR = as.numeric(YEAR),
         VALUE = as.numeric(VALUE))
poverty_rate_data
```

```
## # A tibble: 1,989 × 4
##    STATE                VALUE  YEAR VARIABLE
##    <chr>                <dbl> <dbl> <chr>
##  1 Alabama               16    2018 Poverty_rate
##  2 Alaska                13.1  2018 Poverty_rate
##  3 Arizona               12.8  2018 Poverty_rate
##  4 Arkansas              15.9  2018 Poverty_rate
##  5 California            11.9  2018 Poverty_rate
##  6 Colorado               9.1  2018 Poverty_rate
##  7 Connecticut           10.2  2018 Poverty_rate
##  8 Delaware               7.4  2018 Poverty_rate
##  9 District of Columbia  14.7  2018 Poverty_rate
## 10 Florida               13.7  2018 Poverty_rate
## # … with 1,979 more rows
```

Above code is to remove some tables (e.g. 2017 and 2013) that was duplicate to other ones. Not only that, we set our column name for percent be value, and chage the column value type.

Crime Data

```
crime_data <- crime_data[-((str_which(crime_data, "The figures shown in this column for the offense of rape were estimated using the legacy UCR definition of rape")-1):length(crime_data)+1)]
n_rows <- 2014-1977+1 # determine how many rows there are for each state
rep_cycle <- 4 + n_rows
rep_cycle_cut <- 2 + n_rows
colnames_crime <- (crime_data[4])
# specify which rows are to be deleted based on the file format
delete_rows <- c(seq(from = 2,
                      to = length(crime_data),
                      by = rep_cycle),
                 seq(from = 3,
                      to = length(crime_data),
                      by = rep_cycle),
                 seq(from = 4,
                      to = length(crime_data),
                      by = rep_cycle))
sort(delete_rows) # which rows are to be deleted
```

```
##   [1]    2    3    4   44   45   46   86   87   88  128  129  130  170  171  172
##  [16]  212  213  214  254  255  256  296  297  298  338  339  340  380  381  382
##  [31]  422  423  424  464  465  466  506  507  508  548  549  550  590  591  592
##  [46]  632  633  634  674  675  676  716  717  718  758  759  760  800  801  802
##  [61]  842  843  844  884  885  886  926  927  928  968  969  970 1010 1011 1012
##  [76] 1052 1053 1054 1094 1095 1096 1136 1137 1138 1178 1179 1180 1220 1221 1222
##  [91] 1262 1263 1264 1304 1305 1306 1346 1347 1348 1388 1389 1390 1430 1431 1432
## [106] 1472 1473 1474 1514 1515 1516 1556 1557 1558 1598 1599 1600 1640 1641 1642
## [121] 1682 1683 1684 1724 1725 1726 1766 1767 1768 1808 1809 1810 1850 1851 1852
## [136] 1892 1893 1894 1934 1935 1936 1976 1977 1978 2018 2019 2020 2060 2061 2062
## [151] 2102 2103 2104
```

Above code is to delete rows that are unnecessary to the dataset. Some rows have empty rows, which cause an error when importing to tibble. We have to remove those to ensure dataset does not have any unnecessary data.

```
crime_data[44:46] # we remove row 44-46 since those dataset is not necessary data that we need to look at
```

```
## [1] "\n,,National or state crime,,,,,,,"
## [2] "\n,,Violent crime,,,,,,,"
## [3] "\nYear,Population,Violent crime total,Murder and nonnegligent Manslaughter,Legacy rape /1,Revised rape /2
,Robbery,Aggravated assault,"
```

We use this line to ensure we deleted necessary lines. However, it seems like we did not delete correctly enough.

```
crime_data <- crime_data[-delete_rows]
# extract state labels from data
state_labels <- crime_data[str_which(crime_data, "Estimated crime in ")]
state_labels <- str_remove(state_labels, pattern = "Estimated crime in ")
state_label_order <- rep(state_labels, each = n_rows) # repeat n_rows times
crime_data <- crime_data[-str_which(crime_data, "Estimated crime")]
head(crime_data)
```

```
## [1] "1977,   3690000,      15293,         524,         929,,       3572,      10268 "
## [2] "1978,   3742000,      15682,         499,         954,,       3708,      10521 "
## [3] "1979,   3769000,      15578,         496,       1037,,       4127,       9918 "
## [4] "1980,   3861466,      17320,         509,       1158,,       5102,      10551 "
## [5] "1981,   3916000,      18423,         465,       1021,,       4952,      11985 "
## [6] "1982,   3943000,      17653,         417,       1026,,       4417,      11793 "
```

Above code is used to rename the specific row value that has “Estimated crime in…”

```
crime_data_sep <- read_csv(I(crime_data), col_names = FALSE) |>
  select(-X6)# remove random extra-comma column
```

```
## Warning: 1 parsing failure.
##  row col  expected    actual         file
## 1939  -- 8 columns 1 columns literal data
```

```
crime_data_sep <- crime_data_sep |>
  filter(!is.na(crime_data_sep))

# get column names for later
colnames(crime_data_sep) <- c("Year",
                              "Population",
                              "Violent_crime_total",
                              "Murder_and_nonnegligent_Manslaughter",
                              "Legacy_rape",
                              "Revised_rape",
                              "Robbery",
                              "Aggravated_assault")
# add column names in
crime_data_sep <- bind_cols(STATE = state_label_order, crime_data_sep)
```

Above code is to delete one column that was brought mistakenly when importing data, and we rename our columns.

```
crime_data <- crime_data_sep |>
  mutate(VARIABLE = "Viol_crime_count") |>
  rename("VALUE" = Violent_crime_total) |>
  rename("YEAR" = Year) |>
  select(YEAR,STATE, VARIABLE, VALUE)
crime_data
```

```
## # A tibble: 1,938 × 4
##     YEAR STATE   VARIABLE         VALUE
##    <dbl> <chr>   <chr>            <dbl>
##  1  1977 Alabama Viol_crime_count 15293
##  2  1978 Alabama Viol_crime_count 15682
##  3  1979 Alabama Viol_crime_count 15578
##  4  1980 Alabama Viol_crime_count 17320
##  5  1981 Alabama Viol_crime_count 18423
##  6  1982 Alabama Viol_crime_count 17653
##  7  1983 Alabama Viol_crime_count 16471
##  8  1984 Alabama Viol_crime_count 17204
##  9  1985 Alabama Viol_crime_count 18398
## 10  1986 Alabama Viol_crime_count 22616
## # … with 1,928 more rows
```

Above code is to rename some more, to be suitable when combining all data into one table/DF

State Fips

We wrangle the State Fips data we read in by making the labels more clear by renaming some of the variables columns, selecting only the columns that's most relevant to us, and then removing any invalid state codes.

```
# renaming State\n(FIPS) to STATEFP and Name to STATE
# select STATEFP and STATE columns
# remove all 00 from STATEFP to only have valid state codes
STATE_FIPS <- STATE_FIPS |>
  rename(STATEFP = `State\n(FIPS)`,
         STATE = Name) |>
  select(STATEFP, STATE) |>
  filter(STATEFP != "00")
STATE_FIPS
```

```
## # A tibble: 51 × 2
##    STATEFP STATE
##    <chr>   <chr>
##  1 09      Connecticut
##  2 23      Maine
##  3 25      Massachusetts
##  4 33      New Hampshire
##  5 44      Rhode Island
##  6 50      Vermont
##  7 34      New Jersey
##  8 36      New York
##  9 42      Pennsylvania
## 10 17      Illinois
## # … with 41 more rows
```

Demographics

```r
dem_77_79 <- dem_77_79 |>
  # Renaming column names for more clarity
  rename("race_sex" =`Race/Sex Indicator`) |>
  mutate(SEX = str_extract(race_sex, "male|female"),
        RACE = str_extract(race_sex, "Black|White|Other"))|>
  select(-`FIPS State Code`, -`race_sex`) |>
  rename("YEAR" = `Year of Estimate`,
        "STATE" = `State Name`) |>
  filter(YEAR %in% 1977:1979)
dem_77_79 <- dem_77_79 |>
  # Rearrange so there are more rows than columns
  pivot_longer(cols=contains("years"),
               names_to = "AGE_GROUP",
               values_to = "SUB_POP")
glimpse(dem_77_79)
```

```
## Rows: 16,524
## Columns: 6
## $ YEAR      <dbl> 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, …
## $ STATE     <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "Alab…
## $ SEX       <chr> "male", "male", "male", "male", "male", "male", "male", "mal…
## $ RACE      <chr> "White", "White", "White", "White", "White", "White", "White…
## $ AGE_GROUP <chr> "Under 5 years", "5 to 9 years", "10 to 14 years", "15 to 19…
## $ SUB_POP   <dbl> 98814, 113365, 123107, 135343, 126053, 111547, 100674, 81038…
```

This cleans up the data by creating new columns for sex and race and renaming the year and state columns for clarity. Then, it pivots the data to create more rows than columns for columns that contains “years”.

```r
dem_80_89 <- dem_80_89 |>
  map_df(bind_rows)
dem_80_89 <- dem_80_89 |>
  # Renaming columns for clarity
  rename("race_sex" =`Race/Sex Indicator`) |>
  mutate(SEX = str_extract(race_sex, "male|female"),
        RACE = str_extract(race_sex, "Black|White|Other"))|>
  # Exclude race_sex
  select( -`race_sex`) |>
  rename("YEAR" = `Year of Estimate`) |>
  rename("STATEFP_temp" = "FIPS State and County Codes") |>
  mutate(STATEFP = str_sub(STATEFP_temp, start = 1, end = 2)) |>
  left_join(STATE_FIPS, by = "STATEFP") |>
  select(-STATEFP)
```

map_df applies bind_rows to the data and then like the previous data, the new columns of sex and race are created and renamed. Left_join returns all rows from STATE_FIPS and then joins them by STATEFP.

```r
dem_80_89 <- dem_80_89 |>
  # Makes the dataset longer by increasing # of rows and decreasing # of columns
  pivot_longer(cols=contains("years"),
               names_to = "AGE_GROUP",
               values_to = "SUB_POP_temp") |>
  group_by(YEAR, STATE, AGE_GROUP, SEX, RACE) |>
  summarize(SUB_POP = sum(SUB_POP_temp), .groups="drop")
dem_80_89
```

```
## # A tibble: 55,080 × 6
##     YEAR STATE   AGE_GROUP      SEX    RACE  SUB_POP
##    <dbl> <chr>   <chr>          <chr>  <chr>   <dbl>
##  1  1980 Alabama 10 to 14 years female Black   50108
##  2  1980 Alabama 10 to 14 years female Other     805
##  3  1980 Alabama 10 to 14 years female White  109066
##  4  1980 Alabama 10 to 14 years male   Black   50768
##  5  1980 Alabama 10 to 14 years male   Other     826
##  6  1980 Alabama 10 to 14 years male   White  115988
##  7  1980 Alabama 15 to 19 years female Black   58428
##  8  1980 Alabama 15 to 19 years female Other     743
##  9  1980 Alabama 15 to 19 years female White  126783
## 10  1980 Alabama 15 to 19 years male   Black   56808
## # … with 55,070 more rows
```

This pivots the data into having more rows by columns that contain years and changes the column name to age_group with the values of SUB_POP_temp. Then the data is grouped and creates a new column SUB_POP using the sum of the data from SUB_POP_temp.

```
dem_90_99 <- dem_90_99 |>
  map_df(bind_rows)
# Renaming the column names
colnames(dem_90_99) <- c("YEAR", "STATEFP", "Age", "NH_W_M", "NH_W_F", "NH_B_M",
                         "NH_B_F", "NH_AIAN_M", "NH_AIAN_F", "NH_API_M", "NH_API_F",
                         "H_W_M", "H_W_F", "H_B_M", "H_B_F", "H_AIAN_M", "H_AIAN_F",
                         "H_API_M", "H_API_F")
dem_90_99 <- dem_90_99 |>
  drop_na() |>
  # Combining columns into new columns
  mutate(W_M = NH_W_M + H_W_M, W_F = NH_W_F + H_W_F,
         B_M = NH_B_M + H_B_M, B_F = NH_B_F + H_B_F,
         AIAN_M = NH_AIAN_M + H_AIAN_M, AIAN_F = NH_AIAN_F + H_AIAN_F,
         API_M = NH_API_M + H_API_M, API_F = NH_API_F + H_API_F) |>
  # Remove certain columns
  select(-starts_with("NH_"), -starts_with("H_"))
```

Originally, the data had rows for the column names and so this shortened the number of columns. Column names are changed and than new columns are created based on race and gender. Then columns that began with NH and H are removed.

```
dem_90_99 <- dem_90_99 |>
  # Create a new column AGE_GROUP with intervals of 5
  mutate(AGE_GROUP = cut(Age,
                         breaks = seq(0, 90, by=5),
                         right = FALSE, labels = pull(distinct(dem_77_79,AGE_GROUP), AGE_GROUP))) |>
  # Apply pivot_longer to columns except for Age
  select(-Age) |>
  pivot_longer(cols = c(starts_with("W_"),
                        starts_with("B_"),
                        starts_with("AIAN_"),
                        starts_with("API_")),
               names_to = "RACE",
               values_to = "SUB_POP_temp") |>
  # Create new columns
  mutate(SEX = case_when(str_detect(RACE, "_M") ~ "Male",
                         TRUE ~ "Female"),
         RACE = case_when(str_detect(RACE, "W_") ~ "White",
                          str_detect(RACE, "B_") ~ "Black",
                          TRUE ~ "Other"))
```

A new column called AGE_GROUP is added and then the original AGE column is removed. The columns are then pivoted and renamed to RACE. A new SEX column is added.

```
dem_90_99 <- dem_90_99 |>
  left_join(STATE_FIPS, by = "STATEFP") |>
  select(-STATEFP) |>
  group_by(YEAR, STATE, AGE_GROUP, SEX, RACE) |>
  summarize(SUB_POP = sum(SUB_POP_temp), .groups="drop")
glimpse(dem_90_99)
```

```
## Rows: 55,080
## Columns: 6
## $ YEAR      <dbl> 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, …
## $ STATE     <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "Alab…
## $ AGE_GROUP <fct> Under 5 years, Under 5 years, Under 5 years, Under 5 years, …
## $ SEX       <chr> "Female", "Female", "Female", "Male", "Male", "Male", "Femal…
## $ RACE      <chr> "Black", "Other", "White", "Black", "Other", "White", "Black…
## $ SUB_POP   <dbl> 45377, 1406, 92380, 46635, 1360, 98524, 46067, 1698, 92530, …
```

Then, to create similar columns like the previous decades data, left_join returns all rows from STATE_FIPS and then joins them by STATEFP. Then they're grouped to calculate the SUB_POP column.

```
dem_00_10 <- dem_00_10 |>
  map_df(bind_rows)
dem_00_10 <- dem_00_10 |>
  # Drops these columns
  select(-ESTIMATESBASE2000,-CENSUS2010POP) |>
  # Filters to get values that have data
  filter(NAME != "United States",
         SEX != 0,
         RACE != 0,
         AGEGRP != 0,
         ORIGIN == 0) |>
  select(-REGION, -DIVISION, -ORIGIN, -STATE) |>
  rename("STATE" = NAME,
         "AGE_GROUP" = AGEGRP)
```

The 2000s data was different, and we just removed unnecessary columns and filtered out data that was empty. Then columns were renamed to match the other wrangled data.

```
dem_00_10 <- dem_00_10 |>
  # Updates certain columns to be factors
  mutate(SEX = factor(SEX, levels = 1:2, labels = c("Male", "Female")),
         RACE = factor(RACE, levels = 1:6, labels = c("White", "Black", rep("Other",4))),
         AGE_GROUP = factor(AGE_GROUP, levels = 1:18,
                            labels = pull(distinct(dem_77_79,AGE_GROUP), AGE_GROUP)))
dem_00_10 <- dem_00_10 |>
  pivot_longer(cols=contains("ESTIMATE"), names_to = "YEAR", values_to = "SUB_POP_temp") |>
   mutate(YEAR = str_sub(YEAR, start=-4),
          YEAR = as.numeric(YEAR)) |>
  group_by(YEAR, AGE_GROUP, STATE, SEX, RACE) |>
  summarize(SUB_POP = sum(SUB_POP_temp), .groups = "drop")
glimpse(dem_00_10)
```

```
## Rows: 60,588
## Columns: 6
## $ YEAR      <dbl> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, …
## $ AGE_GROUP <fct> Under 5 years, Under 5 years, Under 5 years, Under 5 years, …
## $ STATE     <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "Alab…
## $ SEX       <fct> Male, Male, Male, Female, Female, Female, Male, Male, Male, …
## $ RACE      <fct> White, Black, Other, White, Black, Other, White, Black, Othe…
## $ SUB_POP   <dbl> 99527, 46595, 4487, 94473, 45672, 4431, 14765, 1039, 8572, 1…
```

This makes sure the labels are the same across data sets and have the same data so everything is formatted the same.

Population

```
pop_77_79 <- dem_77_79 |>
  # Grouping by year and state and finding the total population
  group_by(YEAR, STATE) |>
  summarize(TOT_POP = sum(SUB_POP), .groups = "drop")
pop_77_79
```

```
## # A tibble: 153 × 3
##     YEAR STATE                 TOT_POP
##    <dbl> <chr>                   <dbl>
##  1  1977 Alabama               3782571
##  2  1977 Alaska                 397220
##  3  1977 Arizona               2427296
##  4  1977 Arkansas              2207195
##  5  1977 California           22350332
##  6  1977 Colorado              2696179
##  7  1977 Connecticut           3088745
##  8  1977 Delaware               594815
##  9  1977 District of Columbia   681766
## 10  1977 Florida               8888806
## # … with 143 more rows
```

```
pop_80_89 <- dem_80_89 |>
  group_by(YEAR, STATE) |>
  summarize(TOT_POP = sum(SUB_POP), .groups = "drop")
pop_90_99 <- dem_90_99 |>
  group_by(YEAR, STATE) |>
  summarize(TOT_POP = sum(SUB_POP), .groups = "drop")
pop_00_10 <- dem_00_10 |>
  group_by(YEAR, STATE) |>
  summarize(TOT_POP = sum(SUB_POP), .groups = "drop")
```

Population data is stored in the demographic data, so we group the year and state data and sum it to create our population data. We do this for every decade.

Combine Demo and Population Data

```
dem_77_79 <- dem_77_79 |>
  left_join(pop_77_79, by = c("YEAR", "STATE")) |>
  # Create a new column of percentages
  mutate(PERC_SUB_POP = (SUB_POP/TOT_POP)*100) |>
  select(-SUB_POP, -TOT_POP) |>
    mutate(SEX = str_to_title(SEX))
dem_77_79
```

```
## # A tibble: 16,524 × 6
##     YEAR STATE   SEX   RACE  AGE_GROUP      PERC_SUB_POP
##    <dbl> <chr>   <chr> <chr> <chr>                 <dbl>
##  1  1977 Alabama Male  White Under 5 years          2.61
##  2  1977 Alabama Male  White 5 to 9 years           3.00
##  3  1977 Alabama Male  White 10 to 14 years         3.25
##  4  1977 Alabama Male  White 15 to 19 years         3.58
##  5  1977 Alabama Male  White 20 to 24 years         3.33
##  6  1977 Alabama Male  White 25 to 29 years         2.95
##  7  1977 Alabama Male  White 30 to 34 years         2.66
##  8  1977 Alabama Male  White 35 to 39 years         2.14
##  9  1977 Alabama Male  White 40 to 44 years         1.98
## 10  1977 Alabama Male  White 45 to 49 years         2.02
## # … with 16,514 more rows
```

We add in our population data and scale it to create percentage values. Then we remove columns we don't need anymore and create rename the sex column to be in all uppercase.

```
dem_80_89 <- dem_80_89 |>
  left_join(pop_80_89, by = c("YEAR", "STATE")) |>
  # Create a new column of percentages
  mutate(PERC_SUB_POP = (SUB_POP/TOT_POP)*100) |>
  select(-SUB_POP, -TOT_POP) |>
  mutate(SEX = str_to_title(SEX))
```

We repeat this process for Decade_1980.

```
dem_90_99 <- dem_90_99 |>
  left_join(pop_90_99, by = c("YEAR", "STATE")) |>
  # Create a new column of percentages
  mutate(PERC_SUB_POP = (SUB_POP/TOT_POP)*100) |>
  select(-SUB_POP, -TOT_POP)
dem_90_99
```

```
## # A tibble: 55,080 × 6
##     YEAR STATE   AGE_GROUP     SEX    RACE  PERC_SUB_POP
##    <dbl> <chr>   <fct>         <chr>  <chr>        <dbl>
##  1  1990 Alabama Under 5 years Female Black       1.12
##  2  1990 Alabama Under 5 years Female Other       0.0347
##  3  1990 Alabama Under 5 years Female White       2.28
##  4  1990 Alabama Under 5 years Male   Black       1.15
##  5  1990 Alabama Under 5 years Male   Other       0.0336
##  6  1990 Alabama Under 5 years Male   White       2.43
##  7  1990 Alabama 5 to 9 years  Female Black       1.14
##  8  1990 Alabama 5 to 9 years  Female Other       0.0419
##  9  1990 Alabama 5 to 9 years  Female White       2.29
## 10  1990 Alabama 5 to 9 years  Male   Black       1.16
## # … with 55,070 more rows
```

We repeat this process for Decade_1990 except the SEX column is already uppercase.

```
dem_00_10 <- dem_00_10 |>
  left_join(pop_00_10, by = c("YEAR", "STATE")) |>
  # Create a new column of percentages
  mutate(PERC_SUB_POP = (SUB_POP/TOT_POP)*100) |>
 select(-SUB_POP, -TOT_POP)
dem_00_10
```

```
## # A tibble: 60,588 × 6
##     YEAR AGE_GROUP     STATE   SEX    RACE  PERC_SUB_POP
##    <dbl> <fct>         <chr>   <fct>  <fct>        <dbl>
##  1  2000 Under 5 years Alabama Male   White       2.24
##  2  2000 Under 5 years Alabama Male   Black       1.05
##  3  2000 Under 5 years Alabama Male   Other       0.101
##  4  2000 Under 5 years Alabama Female White       2.12
##  5  2000 Under 5 years Alabama Female Black       1.03
##  6  2000 Under 5 years Alabama Female Other       0.0995
##  7  2000 Under 5 years Alaska  Male   White       2.35
##  8  2000 Under 5 years Alaska  Male   Black       0.165
##  9  2000 Under 5 years Alaska  Male   Other       1.37
## 10  2000 Under 5 years Alaska  Female White       2.26
## # … with 60,578 more rows
```

We repeat this process for Decade_2000.

Combine: Demo Data

```
dem <- bind_rows(dem_77_79,
                 dem_80_89,
                 dem_90_99,
                 dem_00_10)
glimpse(dem)
```

```
## Rows: 187,272
## Columns: 6
## $ YEAR         <dbl> 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, 197…
## $ STATE        <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "A…
## $ SEX          <chr> "Male", "Male", "Male", "Male", "Male", "Male", "Male", "…
## $ RACE         <chr> "White", "White", "White", "White", "White", "White", "Wh…
## $ AGE_GROUP    <chr> "Under 5 years", "5 to 9 years", "10 to 14 years", "15 to…
## $ PERC_SUB_POP <dbl> 2.6123502, 2.9970356, 3.2545853, 3.5780690, 3.3324688, 2.…
```

We put the data sets together into a single data set called dem. They will all have the same column names now with data across every year.

Demographic Data (Donohue)

```
# Creates the age groups into a variable
DONOHUE_AGE_GROUPS <- c("15 to 19 years",
                        "20 to 24 years",
                        "25 to 29 years",
                        "30 to 34 years",
                        "35 to 39 years")
dem_DONOHUE <- dem |>
  # We filter by matching values in age_group and our new age groups and filtering specifically for males
  filter(AGE_GROUP %in% DONOHUE_AGE_GROUPS,
             SEX == "Male") |>
  # Collapse factor levels into these age groups
  mutate(AGE_GROUP = fct_collapse(AGE_GROUP, "20 to 39 years"=c("20 to 24 years",
                                                                "25 to 29 years",
                                                                "30 to 34 years",
                                                                "35 to 39 years")))
```

Following the Donohue paper, we create the age groups they used and create a new column called AGE_GROUP that groups anyone from 20 to 39 years old together.

```
dem_DONOHUE <- dem_DONOHUE |>
  mutate(AGE_GROUP = str_replace_all(string = AGE_GROUP,
                                     pattern = " ",
                                     replacement = "_")) |> # separate with "_"
  group_by(YEAR, STATE, RACE, SEX, AGE_GROUP) |>
  summarize(PERC_SUB_POP = sum(PERC_SUB_POP), .groups = "drop")
dem_DONOHUE
```

```
## # A tibble: 10,404 × 6
##     YEAR STATE   RACE  SEX   AGE_GROUP      PERC_SUB_POP
##    <dbl> <chr>   <chr> <chr> <chr>                 <dbl>
##  1  1977 Alabama Black Male  15_to_19_years       1.55
##  2  1977 Alabama Black Male  20_to_39_years       3.04
##  3  1977 Alabama Other Male  15_to_19_years       0.0178
##  4  1977 Alabama Other Male  20_to_39_years       0.0642
##  5  1977 Alabama White Male  15_to_19_years       3.58
##  6  1977 Alabama White Male  20_to_39_years      11.1
##  7  1977 Alaska  Black Male  15_to_19_years       0.163
##  8  1977 Alaska  Black Male  20_to_39_years       0.968
##  9  1977 Alaska  Other Male  15_to_19_years       1.12
## 10  1977 Alaska  Other Male  20_to_39_years       2.73
## # … with 10,394 more rows
```

Here we clean the up the data by replacing the data in the AGE_GROUP column by separating them with a _ instead of a space. Then we use group_by to find the percentage of the sub population that each group makes like they did in the Donohue paper.

```
dem_DONOHUE <- dem_DONOHUE |>
  unite(col = "VARIABLE", RACE, SEX, AGE_GROUP, sep = "_") |> #separate with "_"
  rename("VALUE" = PERC_SUB_POP)
dem_DONOHUE
```

```
## # A tibble: 10,404 × 4
##     YEAR STATE   VARIABLE                    VALUE
##    <dbl> <chr>   <chr>                       <dbl>
##  1  1977 Alabama Black_Male_15_to_19_years  1.55
##  2  1977 Alabama Black_Male_20_to_39_years  3.04
##  3  1977 Alabama Other_Male_15_to_19_years  0.0178
##  4  1977 Alabama Other_Male_20_to_39_years  0.0642
##  5  1977 Alabama White_Male_15_to_19_years  3.58
##  6  1977 Alabama White_Male_20_to_39_years 11.1
##  7  1977 Alaska  Black_Male_15_to_19_years  0.163
##  8  1977 Alaska  Black_Male_20_to_39_years  0.968
##  9  1977 Alaska  Other_Male_15_to_19_years  1.12
## 10  1977 Alaska  Other_Male_20_to_39_years  2.73
## # … with 10,394 more rows
```

Again following the Donohue paper, we combine RACE, SEX, AGE_GROUP into a new column VARIABLE. Each year and state row has a new combination under VARIABLE with its associated VALUE.

Demographic Data (Lott)

```
LOTT_AGE_GROUPS_NULL <- c("Under 5 years",
                          "5 to 9 years")
dem_LOTT <- dem |>
  #We use the "!" with the filter method to skip over any age groups that we specified in the LOTT_AGE_GROUPS_NUL
L, which are ages 9 and under.
  filter(!(AGE_GROUP %in% LOTT_AGE_GROUPS_NULL) )|>
  mutate(AGE_GROUP = fct_collapse(AGE_GROUP,
                                  "10 to 19 years"=c("10 to 14 years", "15 to 19 years"),
                                  "20 to 29 years"=c("20 to 24 years", "25 to 29 years"),
                                  "30 to 39 years"=c("30 to 34 years", "35 to 39 years"),
                                  "40 to 49 years"=c("40 to 44 years", "45 to 49 years"),
                                  "50 to 64 years"=c("50 to 54 years", "55 to 59 years",
                                                     "60 to 64 years"),
                                  "65 years and over"=c("65 to 69 years", "70 to 74 years",
                                                        "75 to 79 years", "80 to 84 years",
                                                        "85 years and over")))
```

The Lott paper included a wider age range, so here we include from ages 10 and older. The age groups are separated by decades, such as 10 to 19, but from 50 years old it becomes every 5 years. Like 50 to 54 years old, 55 to 59 years old, etc.

```
dem_LOTT <- dem_LOTT |>
  mutate(AGE_GROUP = str_replace_all(AGE_GROUP, " ", "_")) |>
  group_by(YEAR, STATE, RACE, SEX, AGE_GROUP) |>
  summarize(PERC_SUB_POP = sum(PERC_SUB_POP), .groups = "drop") |>
  # the VARIABLE column is created by combining RACE, SEX, and AGE_GROUP and is separated by "_".
  unite(col = "VARIABLE", RACE, SEX, AGE_GROUP, sep = "_") |>
  rename("VALUE" = PERC_SUB_POP)
glimpse(dem_LOTT)
```

```
## Rows: 62,424
## Columns: 4
## $ YEAR     <dbl> 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1…
## $ STATE    <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", "Alaba…
## $ VARIABLE <chr> "Black_Female_10_to_19_years", "Black_Female_20_to_29_years",…
## $ VALUE    <dbl> 3.01067713, 2.32860137, 1.29295656, 1.18231753, 1.73263106, 1…
```

Here we change the dem_LOTT data to match our other data sets. The LOTT column for AGE_GROUP is changed to match the other data sets. Finally, we rename PERC_SUB_POP to VALUE.

Combine: Population Data

```
population_data <- bind_rows(pop_77_79,
                             pop_80_89,
                             pop_90_99,
                             pop_00_10)
population_data <- population_data |>
  mutate(VARIABLE = "Population") |>
  # Value is renamed to TOT_POP.
  rename("VALUE" = TOT_POP)
```

Another data set for population data is created with a new column called Population.

Police Staffing

We wrangle the Police Staffing data we read in by filtering the data to years between 1977 and 2014, creating officer_total by combining male_total_ct and female total_ct, and create an officer_state_total column that is the sum of all the values in officer_total. This will clean up our data and make it easier to work with.

```
# filter data so data_year is between 1977 and 2014
# create new variable officer_total, which is combining male_total_ct and female_total_ct, apply those values to
all columns with total_count, replace all NA values with 0
# select data_year, pub_agency_name, state_abbr_ and officer_total, group data_year and state_abbr to perform pro
cesses on them together
# summarize data and create officer_state_total column that is sum of all values in officer_total
ps_data <- ps_data |>
  filter(data_year >= 1977,
         data_year <= 2014) |>
  mutate(across(.cols =contains("total_ct"), ~replace_na(., 0)),
         officer_total = male_total_ct + female_total_ct) |>
  select(data_year,
         pub_agency_name,
         state_abbr,
         officer_total) |>
  group_by(data_year, state_abbr) |>
  summarize(officer_state_total = sum(officer_total), .groups = "drop")
glimpse(ps_data)
```

```
## Rows: 2,242
## Columns: 3
## $ data_year           <dbl> 1977, 1977, 1977, 1977, 1977, 1977, 1977, 1977, 19…
## $ state_abbr          <chr> "AK", "AL", "AR", "AS", "AZ", "CA", "CO", "CT", "C…
## $ officer_state_total <dbl> 544, 7380, 3344, 0, 6414, 65596, 7337, 6051, 0, 47…
```

We now want to remove the territories that are included in the data.

```
# create vector of all non-states
# use vector to filter out all the values in the data that match with value in vector
state_of_interest_NULL <- c("AS", "GM", "CZ", "FS", "MP", "OT", "PR", "VI")
ps_data <- ps_data |>
  filter(!(state_abbr %in% state_of_interest_NULL))
```

We create a new tibble with the state abbreviations and adjust a few values and columns to make them easier to work with.

```
# use state abbreviations for the data
# create tibble with values state_abbr using state.abb values and STATE using the state.name values
# replace all NE values to NB in state_abbr
# add DC as state_abbr and District of Columbia as STATE to a new row.
state_abb_data <- tibble("state_abbr" = state.abb, "STATE" = state.name)
state_abb_data <- state_abb_data |>
  mutate(state_abbr = str_replace(string = state_abbr,
                                  pattern = "NE",
                                  replacement = "NB")) |>
  add_row(state_abbr = "DC", STATE = "District of Columbia")
```

We combine ps_data and state_abb data and make a few adjustments to column names to make them easier to work with.

```
# left join state_abb_data to ps_data, adding values that state_abb_data have in common with ps_data, adding miss
ing values otherwise
# drop state_abbr from selection
# rename YEAR to data_year, VALUE to officer_state_total, and change VARIABLE values to officer_state_total
ps_data <- ps_data |>
  left_join(state_abb_data, by = "state_abbr") |>
  select(-state_abbr) |>
  rename(YEAR = "data_year",
         VALUE = "officer_state_total") |>
  mutate(VARIABLE = "officer_state_total")
ps_data
```

```
## # A tibble: 1,938 × 4
##     YEAR VALUE STATE                VARIABLE
##    <dbl> <dbl> <chr>                <chr>
##  1  1977   544 Alaska               officer_state_total
##  2  1977  7380 Alabama              officer_state_total
##  3  1977  3344 Arkansas             officer_state_total
##  4  1977  6414 Arizona              officer_state_total
##  5  1977 65596 California           officer_state_total
##  6  1977  7337 Colorado             officer_state_total
##  7  1977  6051 Connecticut          officer_state_total
##  8  1977  4751 District of Columbia officer_state_total
##  9  1977  1018 Delaware             officer_state_total
## 10  1977 24588 Florida              officer_state_total
## # … with 1,928 more rows
```

We now want to scale by population to compare between states. For scaling, we'll multiply the VALUE values by 100,000, then divide by Population_temp.

```
# create new data frame denominator_temp (population_data but excludes VARIABLE column and rename Population_temp
to VALUE)
# left join denominator_temp to ps_data by STATE AND YEAR, adding STATE and YEAR values that are the same in ps_d
ata, add missing values if the values aren't the same
# mutate VALUE values to scale it, multiplying values by 100,000 and dividing it by Population_temp
# change VARIABLE value to police_per_100k_lag and remove the Population_temp column from selection
denominator_temp <- population_data |>
  select(-VARIABLE) |>
  rename("Population_temp"=VALUE)
ps_data <- ps_data |>
  left_join(denominator_temp, by=c("STATE","YEAR")) |>
  mutate(VALUE = (VALUE * 100000) / Population_temp) |>
  mutate(VARIABLE = "police_per_100k_lag") |>
  select(-Population_temp)
```

Unemployment

- We convert the list in ue_rate_data to a single tibble and add a new column "STATE" that states the state name of the tibble that each row comes from. We also rename some columns and create a new column named VARIABLE.

```
ue_rate_data <- ue_rate_data |>
  map_df(bind_rows, .id = "STATE") |> # bind all the tibbles together and save the output as a dataframe, create
a new col named "STATE"
  select(STATE, Year, Annual) |> # select col STATE, Year, Annual
  rename("YEAR" = Year,    # rename Year col
         "VALUE" = Annual) |> # rename Annual col
  mutate(VARIABLE = "Unemployment_rate") # create a col named VARIABLE, store string "Unemployment_rate" in each
row
```

RTC

1. We first get the RTC related data from DAWpaper.
2. We split the text and drop unnecessary rows.
3. We combine it by row into a tibble, we later add the column names to the tibble p_62 and drop the unnecessary rows.
4. We convert the values in RTC_LAW_YEAR to numeric. We store the changes into the new variable RTC that only contains the column STATE and RTC_LAW_YEAR.

```
#1
DAWpaper_p_62 <- DAWpaper[[62]] #get related data

#2
p_62 <- DAWpaper_p_62 |>
  str_split("\n") |> # split string on space
  unlist() |>
  as_tibble() |>
  slice(-(1:2)) |> #drop row 1 and 2
  rename(RTC = value) |> # rename tibble name
  slice(-c(53:54)) |>  # remove empty lines
  mutate(RTC = str_replace_all(RTC, "\\s{40,}", "|N/A|"), #separate N/A value with |, fill missing value with N/A
         RTC = str_trim(RTC, side = "left"), # remove whitespace on left side
         RTC = str_replace_all(RTC, "\\s{2,15}", "|")) #split by |

#3
p_62 <- pull(p_62, RTC) |>
  str_split( "\\|{1,}")  # split data on "|" symbol
# get the tibble!
p_62 <- as_tibble(do.call(rbind, p_62)) # rbind and not bind_cols here b/c we have no column names yet
# add col names
colnames(p_62) <- c("STATE",
                    "E_Date_RTC",
                    "Frac_Yr_Eff_Yr_Pass",
                    "RTC_Date_SA")
p_62 <- p_62 |>
  slice(-c(1, 53:nrow(p_62))) # remove unnecessary and empty rows

#4
RTC <- p_62 |>
  select(STATE, RTC_Date_SA) |>
  rename(RTC_LAW_YEAR = RTC_Date_SA) |> # rename col RTC_Date_SA
  mutate(RTC_LAW_YEAR = as.numeric(RTC_LAW_YEAR)) |> # convert value to numeric
  mutate(RTC_LAW_YEAR = case_when(RTC_LAW_YEAR == 0 ~ Inf,
                              TRUE ~ RTC_LAW_YEAR)) # make sure values in RTC_LAW_YEAR are > 0
```

Combine Donohue

```
# combine after all that wrangling!
DONOHUE_DF <- bind_rows(dem_DONOHUE,
                        ue_rate_data,
                        poverty_rate_data,
                        crime_data,
                        population_data,
                        ps_data)
DONOHUE_DF
```

```
## # A tibble: 20,247 × 4
##     YEAR STATE   VARIABLE                    VALUE
##    <dbl> <chr>   <chr>                       <dbl>
##  1  1977 Alabama Black_Male_15_to_19_years  1.55
##  2  1977 Alabama Black_Male_20_to_39_years  3.04
##  3  1977 Alabama Other_Male_15_to_19_years  0.0178
##  4  1977 Alabama Other_Male_20_to_39_years  0.0642
##  5  1977 Alabama White_Male_15_to_19_years  3.58
##  6  1977 Alabama White_Male_20_to_39_years 11.1
##  7  1977 Alaska  Black_Male_15_to_19_years  0.163
##  8  1977 Alaska  Black_Male_20_to_39_years  0.968
##  9  1977 Alaska  Other_Male_15_to_19_years  1.12
## 10  1977 Alaska  Other_Male_20_to_39_years  2.73
## # … with 20,237 more rows
```

```
# to wide format!
DONOHUE_DF <- DONOHUE_DF |>
  pivot_wider(names_from = "VARIABLE",
            values_from = "VALUE")
DONOHUE_DF |>
  slice_sample(n = 10) |>
  glimpse()
```

```
## Rows: 10
## Columns: 13
## $ YEAR                       <dbl> 1988, 2018, 1981, 1991, 2010, 1991, 2017, 20…
## $ STATE                      <chr> "Missouri", "Nevada", "California", "Utah", …
## $ Black_Male_15_to_19_years <dbl> 0.50329307, NA, 0.41601509, 0.04475318, 0.29…
## $ Black_Male_20_to_39_years <dbl> 1.6416060, NA, 1.3847336, 0.1783355, 0.96940…
## $ Other_Male_15_to_19_years <dbl> 0.05492223, NA, 0.33295042, 0.17737611, 0.74…
## $ Other_Male_20_to_39_years <dbl> 0.2202596, NA, 1.3153436, 0.6406534, 2.84158…
## $ White_Male_15_to_19_years <dbl> 3.222288, NA, 3.650934, 4.262839, 2.845963, …
## $ White_Male_20_to_39_years <dbl> 13.79221, NA, 15.27256, 14.83294, 10.79685, …
## $ Unemployment_rate          <dbl> 5.7, 4.4, 7.4, 4.7, 12.2, 2.7, 3.7, 5.9, 12.…
## $ Poverty_rate               <dbl> 12.7, 13.0, 13.3, 12.9, 16.3, 9.5, 11.4, 17.…
## $ Viol_crime_count           <dbl> 28393, NA, 208485, 5077, 164133, 5330, NA, 1…
## $ Population                 <dbl> 5081731, NA, 24285898, 1771941, 37349363, 15…
## $ police_per_100k_lag        <dbl> 255.8971, NA, 250.1452, 298.6555, 327.8369, …
```

```
# add in RTC data!
DONOHUE_DF <- DONOHUE_DF |>
  left_join(RTC , by = c("STATE")) |>
  mutate(RTC_LAW = case_when(YEAR >= RTC_LAW_YEAR ~ TRUE,
                              TRUE ~ FALSE)) |>
 drop_na() # drop rows with missing information
DONOHUE_DF |>
  slice_sample(n = 10) |>
  glimpse()
```

```
## Rows: 10
## Columns: 15
## $ YEAR                       <dbl> 1998, 2004, 2010, 1992, 2003, 2002, 1998, 19…
## $ STATE                      <chr> "Oklahoma", "Arizona", "New Hampshire", "Sou…
## $ Black_Male_15_to_19_years <dbl> 0.40793801, 0.16522174, 0.06508404, 1.365309…
## $ Black_Male_20_to_39_years <dbl> 1.1635351, 0.6354287, 0.2377048, 4.6109011, …
## $ Other_Male_15_to_19_years <dbl> 0.46129964, 0.45914977, 0.15955843, 0.039354…
## $ Other_Male_20_to_39_years <dbl> 1.3399699, 1.4680833, 0.6051221, 0.1651680, …
## $ White_Male_15_to_19_years <dbl> 3.240566, 3.081556, 3.384826, 2.315741, 3.88…
## $ White_Male_20_to_39_years <dbl> 11.16761, 12.33916, 11.01591, 11.50471, 13.1…
## $ Unemployment_rate          <dbl> 4.3, 5.0, 5.8, 6.7, 5.6, 5.6, 5.9, 6.1, 6.0,…
## $ Poverty_rate               <dbl> 14.1, 14.4, 6.5, 19.0, 10.2, 9.5, 15.4, 10.2…
## $ Viol_crime_count           <dbl> 18053, 28952, 2204, 34029, 3362, 49578, 2298…
## $ Population                 <dbl> 3339478, 5652404, 1316759, 3600576, 1363380,…
## $ police_per_100k_lag        <dbl> 306.9941, 363.3675, 272.1075, 282.6770, 284.…
## $ RTC_LAW_YEAR               <dbl> 1996, 1995, 1959, 1997, 1990, 1989, Inf, 200…
## $ RTC_LAW                    <lgl> TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, …
```

```
# filter to only data where RTC laws were adopted between 1980-2010
# have crime data pre- and post-adotion this way
baseline_year <- min(DONOHUE_DF$YEAR)
censoring_year <- max(DONOHUE_DF$YEAR)
DONOHUE_DF <- DONOHUE_DF |>
  mutate(TIME_0 = baseline_year,
         TIME_INF = censoring_year) |>
  filter(RTC_LAW_YEAR > TIME_0)
```

```
# calculate violent crime rate; put population/crime on log scale
DONOHUE_DF <- DONOHUE_DF |>
  mutate(Viol_crime_rate_1k = (Viol_crime_count*1000)/Population,
         Viol_crime_rate_1k_log = log(Viol_crime_rate_1k),
         Population_log = log(Population))
```

Combine LOTT

```
# Combine LOTT dataset into one data frame variable
LOTT_DF <- bind_rows(dem_LOTT,
                     ue_rate_data,
                     poverty_rate_data,
                     crime_data,
                     population_data,
                     ps_data) |>
  pivot_wider(names_from = "VARIABLE",
              values_from = "VALUE") |>
  left_join(RTC , by = c("STATE")) |>
  mutate(RTC_LAW = case_when(YEAR >= RTC_LAW_YEAR ~ TRUE,
                             TRUE ~ FALSE)) |>
   drop_na()
```

```
# We remove dataset that does not correlate with our RTC_LAW_YEAR (so that RTC_LAW_YEAR and Year have same year t
o directly compare)
baseline_year <- min(LOTT_DF$YEAR)
censoring_year <- max(LOTT_DF$YEAR)
LOTT_DF <- LOTT_DF |>
  mutate(TIME_0 = baseline_year,
         TIME_INF = censoring_year) |>
  filter(RTC_LAW_YEAR > TIME_0)
```

```
# we create rate by 1000 (we use this to answer our question)
LOTT_DF <- LOTT_DF |>
  mutate(Viol_crime_rate_1k = (Viol_crime_count*1000)/Population,
         Viol_crime_rate_1k_log = log(Viol_crime_rate_1k),
         Population_log = log(Population))
LOTT_DF
```

```
## # A tibble: 1,364 × 50
##     YEAR STATE      Black_Female_10_to_… Black_Female_20_to… Black_Female_30_to…
##    <dbl> <chr>                     <dbl>               <dbl>               <dbl>
##  1  1980 Alaska                   0.264               0.443              0.201
##  2  1980 Arizona                  0.287               0.278              0.165
##  3  1980 Arkansas                 1.82                1.50               0.842
##  4  1980 California               0.780               0.815              0.581
##  5  1980 Colorado                 0.352               0.388              0.245
##  6  1980 Delaware                 1.87                1.68               1.14
##  7  1980 District …               6.53                7.54               5.18
##  8  1980 Florida                  1.50                1.37               0.912
##  9  1980 Georgia                  2.90                2.78               1.85
## 10  1980 Hawaii                   0.0930              0.215              0.0776
## # … with 1,354 more rows, and 45 more variables:
## #   Black_Female_40_to_49_years <dbl>, Black_Female_50_to_64_years <dbl>,
## #   Black_Female_65_years_and_over <dbl>, Black_Male_10_to_19_years <dbl>,
## #   Black_Male_20_to_29_years <dbl>, Black_Male_30_to_39_years <dbl>,
## #   Black_Male_40_to_49_years <dbl>, Black_Male_50_to_64_years <dbl>,
## #   Black_Male_65_years_and_over <dbl>, Other_Female_10_to_19_years <dbl>,
## #   Other_Female_20_to_29_years <dbl>, Other_Female_30_to_39_years <dbl>,
## #   Other_Female_40_to_49_years <dbl>, Other_Female_50_to_64_years <dbl>,
## #   Other_Female_65_years_and_over <dbl>, Other_Male_10_to_19_years <dbl>,
## #   Other_Male_20_to_29_years <dbl>, Other_Male_30_to_39_years <dbl>,
## #   Other_Male_40_to_49_years <dbl>, Other_Male_50_to_64_years <dbl>,
## #   Other_Male_65_years_and_over <dbl>, White_Female_10_to_19_years <dbl>,
## #   White_Female_20_to_29_years <dbl>, White_Female_30_to_39_years <dbl>,
## #   White_Female_40_to_49_years <dbl>, White_Female_50_to_64_years <dbl>,
## #   White_Female_65_years_and_over <dbl>, White_Male_10_to_19_years <dbl>,
## #   White_Male_20_to_29_years <dbl>, White_Male_30_to_39_years <dbl>,
## #   White_Male_40_to_49_years <dbl>, White_Male_50_to_64_years <dbl>,
## #   White_Male_65_years_and_over <dbl>, Unemployment_rate <dbl>,
## #   Poverty_rate <dbl>, Viol_crime_count <dbl>, Population <dbl>,
## #   police_per_100k_lag <dbl>, RTC_LAW_YEAR <dbl>, RTC_LAW <lgl>, TIME_0 <dbl>,
## #   TIME_INF <dbl>, Viol_crime_rate_1k <dbl>, Viol_crime_rate_1k_log <dbl>,
## #   Population_log <dbl>
```

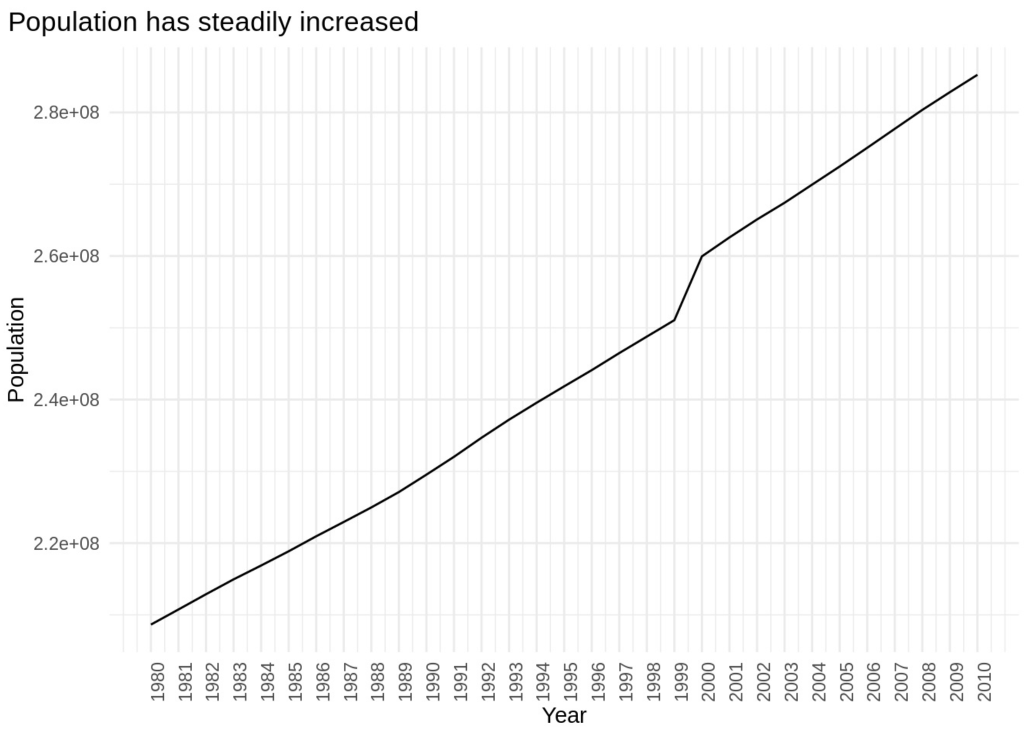
Save data

```
save(dem_77_79, dem_80_89, dem_90_99, dem_00_10, #demographic data
     STATE_FIPS, # codes for states
     ps_data, # police staffing data
     ue_rate_data, # unemployment data
     poverty_rate_data, # poverty data
     crime_data, # crime data
     DAWpaper,
     file = "data/imported_data_rtc.rda")
```

This saves the imported data into a file called “imported_data_rtc.rda” inside the data directory.

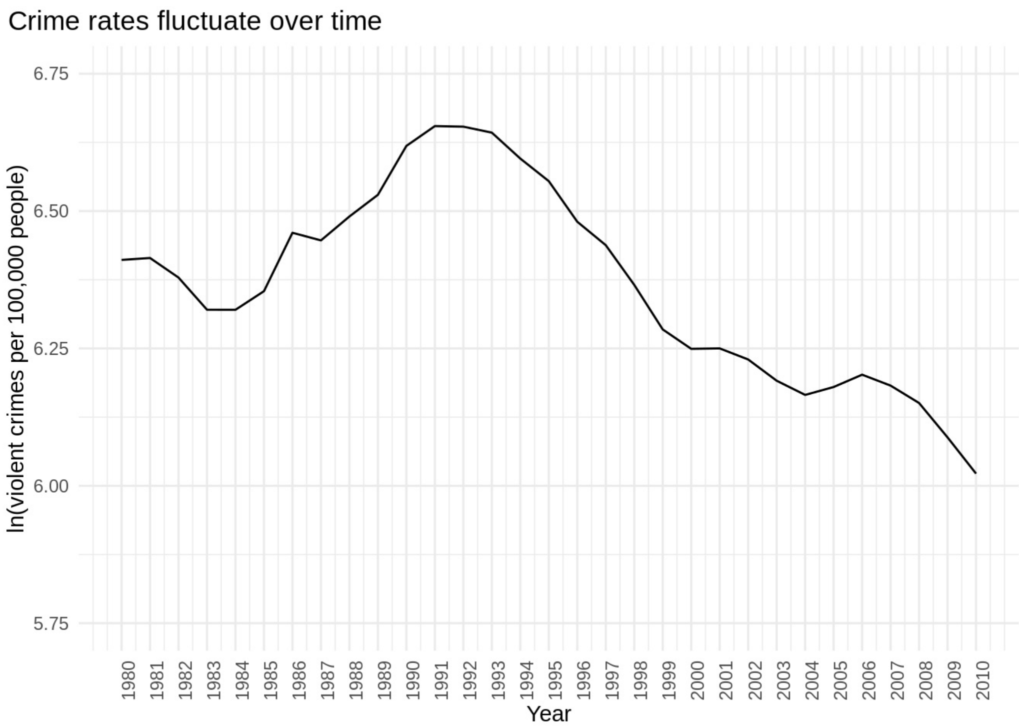
Exploratory Data Analysis

```
df <- DONOHUE_DF |>
  group_by(YEAR) |>
  summarise(Population = sum(Population))

ggplot(df, aes(x = YEAR, y = Population)) +
  geom_line() +
  scale_x_continuous(
    breaks = seq(1980, 2010, by = 1),
    limits = c(1980, 2010),
    labels = c(seq(1980, 2010, by = 1))
  ) +
  labs(
    title = "Population has steadily increased",
    x = "Year",
    y = "Population"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90),
        plot.title.position = "plot")
```



Above code has two parts: one creating data frame that consists of population by each year from DONOHUE dataset, and one that uses that dataframe to create a graph (using ggplot). As we can see from the graph, the population has steady increased from 1980 to 2010.

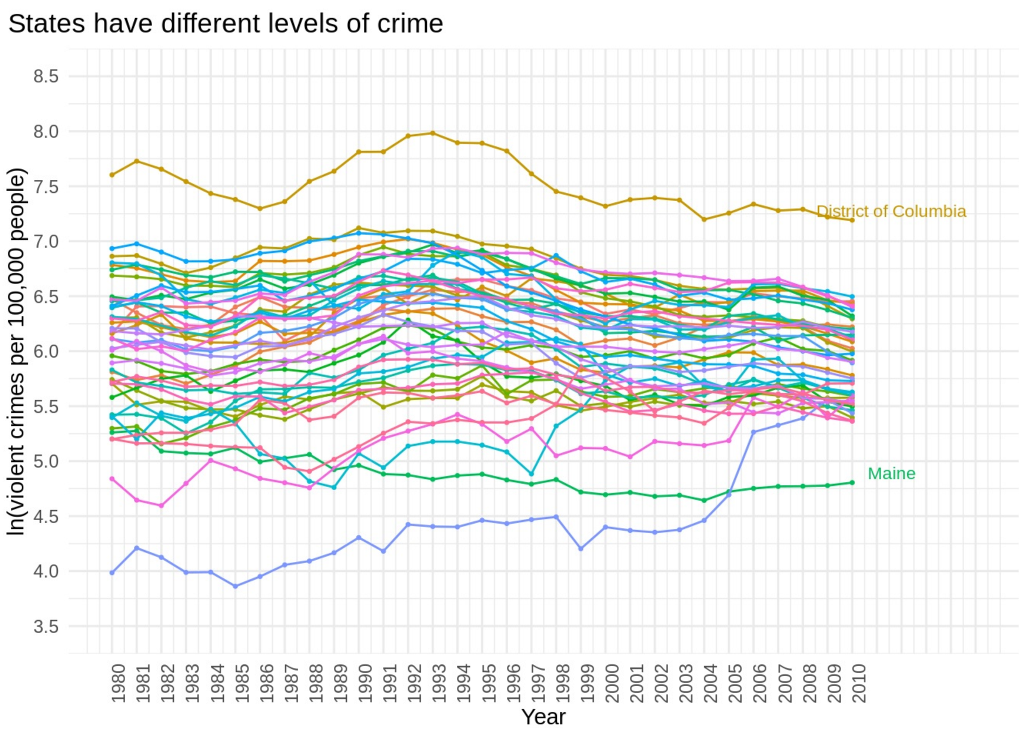
```r
df <- DONOHUE_DF |>
  group_by(YEAR) |>
  summarize(
    Viol_crime_count = sum(Viol_crime_count),
    Population = sum(Population),
    .groups = "drop"
  ) |>
  mutate(Viol_crime_rate_100k_log = log((Viol_crime_count * 100000) / Population))

df |>
  ggplot(aes(x = YEAR, y = Viol_crime_rate_100k_log)) +
  geom_line() +
  scale_x_continuous(
    breaks = seq(1980, 2010, by = 1),
    limits = c(1980, 2010),
    labels = c(seq(1980, 2010, by = 1))
  ) +
  scale_y_continuous(
    breaks = seq(5.75, 6.75, by = 0.25),
    limits = c(5.75, 6.75)
  ) +
  labs(
    title = "Crime rates fluctuate over time",
    x = "Year",
    y = "ln(violent crimes per 100,000 people)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90), plot.title.position = "plot")
```



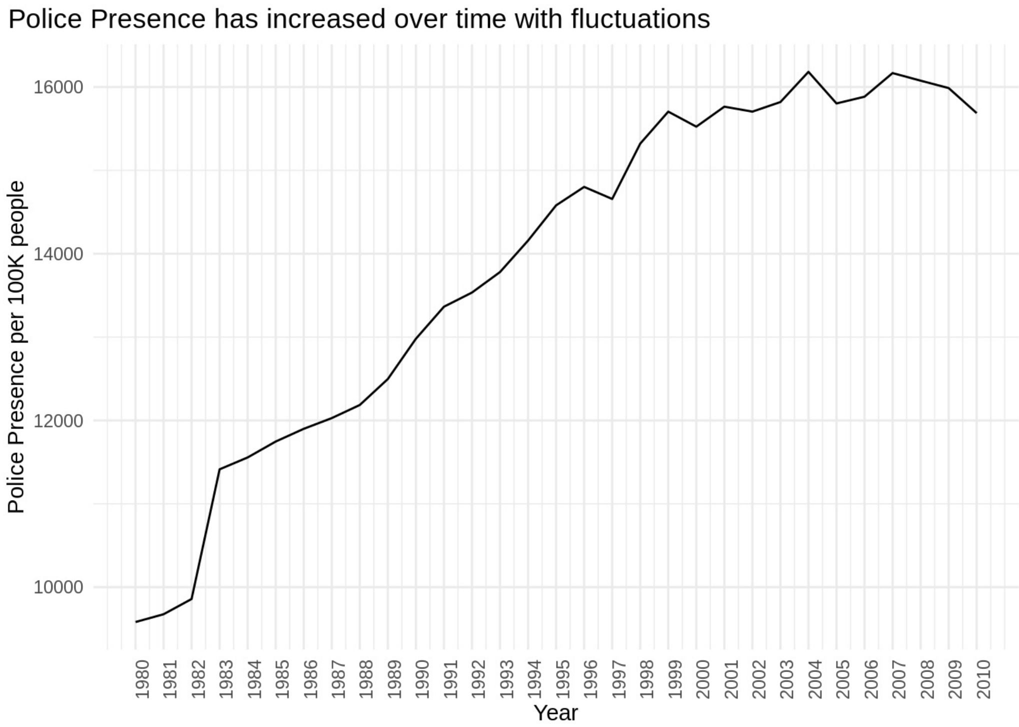
Above code has two parts: one that sets dataframe contains data of violent crime rate number (using mutate() function) grouped by year, and another part that uses that data frame to create a graph of violent crimes per 10,000 people per year. As we can see from the graph above, the crime rate starts to decrease from 1980 to 1984, but then increases until 1992, then it starts to decreases again until 2004, then increase for little bit until 2006, then starts to decrease again. In all, we can conclude that the crime rate fluctuates over the years.

```
p <- DONOHUE_DF |>
  mutate(Viol_crime_rate_100k_log = log((Viol_crime_count * 100000) / Population)) |>
  ggplot(aes(x = YEAR, y = Viol_crime_rate_100k_log, color = STATE)) +
  geom_point(size = 0.5) +
  geom_line(aes(group = STATE),
    size = 0.5,
    show.legend = FALSE
  ) +
  geom_text_repel(data = DONOHUE_DF |>
      mutate(Viol_crime_rate_100k_log = log((Viol_crime_count * 100000) / Population)) |>
      filter(YEAR == last(YEAR)),
      aes(label = STATE,x = YEAR, y = Viol_crime_rate_100k_log),
      size = 3, alpha = 1, nudge_x = 1, direction = "y",
      hjust = 1, vjust = 1, segment.size = 0.25, segment.alpha = 0.25,
      force = 1, max.iter = 9999)

p +
  guides(color = "none") +
  scale_x_continuous(
    breaks = seq(1980, 2015, by = 1),
    limits = c(1980, 2015),
    labels = c(seq(1980, 2010, by = 1), rep("", 5))
  ) +
  scale_y_continuous(
    breaks = seq(3.5, 8.5, by = 0.5),
    limits = c(3.5, 8.5)
  ) +
  labs(
    title = "States have different levels of crime",
    x = "Year", y = "ln(violent crimes per 100,000 people)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90), plot.title.position = "plot")
```

```
## Warning: ggrepel: 42 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
p <- DONOHUE_DF |>
  group_by(YEAR) |>
  summarise(Police = sum(police_per_100k_lag)) |>
  ggplot(aes(x = YEAR, y = Police)) +
  geom_line() +
  scale_x_continuous(
    breaks = seq(1980, 2010, by = 1),
    limits = c(1980, 2010),
    labels = c(seq(1980, 2010, by = 1))
  )

p +
  labs(
    title = "Police Presence has increased over time with fluctuations",
    x = "Year",
    y = "Police Presence per 100K people"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90),
        plot.title.position = "plot")
```



As we can see from the graph above, the police staffing has increased over the years.

Data Analysis

We are using Panel Linear Regression

```
d_panel_DONOHUE <- pdata.frame(DONOHUE_DF, index = c("STATE", "YEAR"))
class(d_panel_DONOHUE)
```

```
## [1] "pdata.frame" "data.frame"
```

```
slice_head(d_panel_DONOHUE, n=3)
```

```
##             YEAR  STATE Black_Male_15_to_19_years Black_Male_20_to_39_years
## Alaska-1980 1980 Alaska                 0.1670456                 0.9933775
## Alaska-1981 1981 Alaska                 0.1732299                 1.0028219
## Alaska-1982 1982 Alaska                 0.1737069                 1.0204445
##             Other_Male_15_to_19_years Other_Male_20_to_39_years
## Alaska-1980                  1.129782                  2.963329
## Alaska-1981                  1.124441                  2.974775
## Alaska-1982                  1.069821                  3.015071
##             White_Male_15_to_19_years White_Male_20_to_39_years
## Alaska-1980                  3.627805                  18.28852
## Alaska-1981                  3.558261                  18.12821
## Alaska-1982                  3.391844                  18.10666
##             Unemployment_rate Poverty_rate Viol_crime_count Population
## Alaska-1980               9.6          9.6             1919     404680
## Alaska-1981               9.4          9.0             2537     418519
## Alaska-1982               9.9         10.6             2732     449608
##             police_per_100k_lag RTC_LAW_YEAR RTC_LAW TIME_0 TIME_INF
## Alaska-1980            194.7218         1995   FALSE   1980     2010
## Alaska-1981            200.2299         1995   FALSE   1980     2010
## Alaska-1982            191.0553         1995   FALSE   1980     2010
##             Viol_crime_rate_1k Viol_crime_rate_1k_log Population_log
## Alaska-1980           4.742018               1.556463       12.91085
## Alaska-1981           6.061851               1.802015       12.94448
## Alaska-1982           6.076404               1.804413       13.01613
```

Above code is grouping state and year and other columns like race with age from the DONOHUE dataset and we are converting into data frame format.

```
DONOHUE_OUTPUT <- plm(Viol_crime_rate_1k_log ~
                      RTC_LAW +
                      White_Male_15_to_19_years +
                      White_Male_20_to_39_years +
                      Black_Male_15_to_19_years +
                      Black_Male_20_to_39_years +
                      Other_Male_15_to_19_years +
                      Other_Male_20_to_39_years +
                      Unemployment_rate +
                      Poverty_rate +
                      Population_log +
                      police_per_100k_lag,
                      effect = "twoways",
                      model = "within",
                      data = d_panel_DONOHUE)
```

Above code is creating a variable type of PLM with selected columns from p_panel_DONOHUE (Data Frame)

```
DONOHUE_OUTPUT_TIDY <- tidy(DONOHUE_OUTPUT, conf.int = 0.95)
DONOHUE_OUTPUT_TIDY
```

```
## # A tibble: 11 × 7
##    term                estimate std.error statistic  p.value conf.low conf.high
##    <chr>                  <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1 RTC_LAWTRUE          0.0240    0.0170      1.42  1.56e- 1 -9.19e-3  0.0573
##  2 White_Male_15_to_1…  0.0104    0.0282      0.367 7.13e- 1 -4.49e-2  0.0656
##  3 White_Male_20_to_3…  0.0293    0.0100      2.93  3.50e- 3  9.68e-3  0.0490
##  4 Black_Male_15_to_1… -0.0597    0.0578     -1.03  3.02e- 1 -1.73e-1  0.0536
##  5 Black_Male_20_to_3…  0.123     0.0195      6.34  3.17e-10  8.53e-2  0.162
##  6 Other_Male_15_to_1…  0.674     0.114       5.92  4.15e- 9  4.51e-1  0.897
##  7 Other_Male_20_to_3… -0.304     0.0383     -7.95  4.21e-15 -3.79e-1 -0.229
##  8 Unemployment_rate   -0.0171    0.00498    -3.42  6.36e- 4 -2.68e-2 -0.00729
##  9 Poverty_rate        -0.00760   0.00299    -2.54  1.12e- 2 -1.35e-2 -0.00174
## 10 Population_log      -0.211     0.0617     -3.42  6.55e- 4 -3.32e-1 -0.0899
## 11 police_per_100k_lag  0.000559  0.000140    4.00  6.72e- 5  2.85e-4  0.000833
```

```
DONOHUE_OUTPUT_TIDY$Analysis <- "Analysis Donohue"
```

```
LOTT_variables <- LOTT_DF |>
  select(RTC_LAW,
         contains(c("White", "Black", "Other")),
         Unemployment_rate,
         Poverty_rate,
         Population_log,
         police_per_100k_lag) |>
  colnames()
LOTT_fmla <- as.formula(paste("Viol_crime_rate_1k_log ~", paste(LOTT_variables, collapse = " + ")
))
LOTT_fmla
```

```
## Viol_crime_rate_1k_log ~ RTC_LAW + White_Female_10_to_19_years +
##     White_Female_20_to_29_years + White_Female_30_to_39_years +
##     White_Female_40_to_49_years + White_Female_50_to_64_years +
##     White_Female_65_years_and_over + White_Male_10_to_19_years +
##     White_Male_20_to_29_years + White_Male_30_to_39_years + White_Male_40_to_49_years +
##     White_Male_50_to_64_years + White_Male_65_years_and_over +
##     Black_Female_10_to_19_years + Black_Female_20_to_29_years +
##     Black_Female_30_to_39_years + Black_Female_40_to_49_years +
##     Black_Female_50_to_64_years + Black_Female_65_years_and_over +
##     Black_Male_10_to_19_years + Black_Male_20_to_29_years + Black_Male_30_to_39_years +
##     Black_Male_40_to_49_years + Black_Male_50_to_64_years + Black_Male_65_years_and_over +
##     Other_Female_10_to_19_years + Other_Female_20_to_29_years +
##     Other_Female_30_to_39_years + Other_Female_40_to_49_years +
##     Other_Female_50_to_64_years + Other_Female_65_years_and_over +
##     Other_Male_10_to_19_years + Other_Male_20_to_29_years + Other_Male_30_to_39_years +
##     Other_Male_40_to_49_years + Other_Male_50_to_64_years + Other_Male_65_years_and_over +
##     Unemployment_rate + Poverty_rate + Population_log + police_per_100k_lag
```

Above code is used to group and create a formula variable to linearly group crime rate with other variables used with it.

```
d_panel_LOTT <- pdata.frame(LOTT_DF, index = c("STATE", "YEAR"))
LOTT_OUTPUT <- plm(LOTT_fmla,
                   model = "within",
                   effect = "twoways",
                   data = d_panel_LOTT
)
```

Above code is creating PLM model using grouped variable that was created above.

```
LOTT_OUTPUT_TIDY <- tidy(LOTT_OUTPUT, conf.int = 0.95)
LOTT_OUTPUT_TIDY
```

```
## # A tibble: 41 × 7
##    term                 estimate std.error statistic  p.value conf.low conf.high
##    <chr>                   <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1 RTC_LAWTRUE          -0.0518     0.0162    -3.20  1.39e- 3  -0.0835   -0.0201
##  2 White_Female_10_to_…  0.636      0.149      4.26  2.24e- 5   0.343     0.929
##  3 White_Female_20_to_…  0.00698    0.0670     0.104 9.17e- 1  -0.124     0.138
##  4 White_Female_30_to_…  0.261      0.0813     3.21  1.38e- 3   0.101     0.420
##  5 White_Female_40_to_…  0.0168     0.0814     0.206 8.37e- 1  -0.143     0.176
##  6 White_Female_50_to_… -0.459      0.0625    -7.35  3.60e-13  -0.582    -0.337
##  7 White_Female_65_yea…  0.156      0.0469     3.33  9.04e- 4   0.0641    0.248
##  8 White_Male_10_to_19… -0.583      0.143     -4.07  4.92e- 5  -0.863    -0.302
##  9 White_Male_20_to_29…  0.0639     0.0623     1.03  3.05e- 1  -0.0582    0.186
## 10 White_Male_30_to_39… -0.200      0.0859    -2.33  2.01e- 2  -0.368    -0.0315
## # … with 31 more rows
```

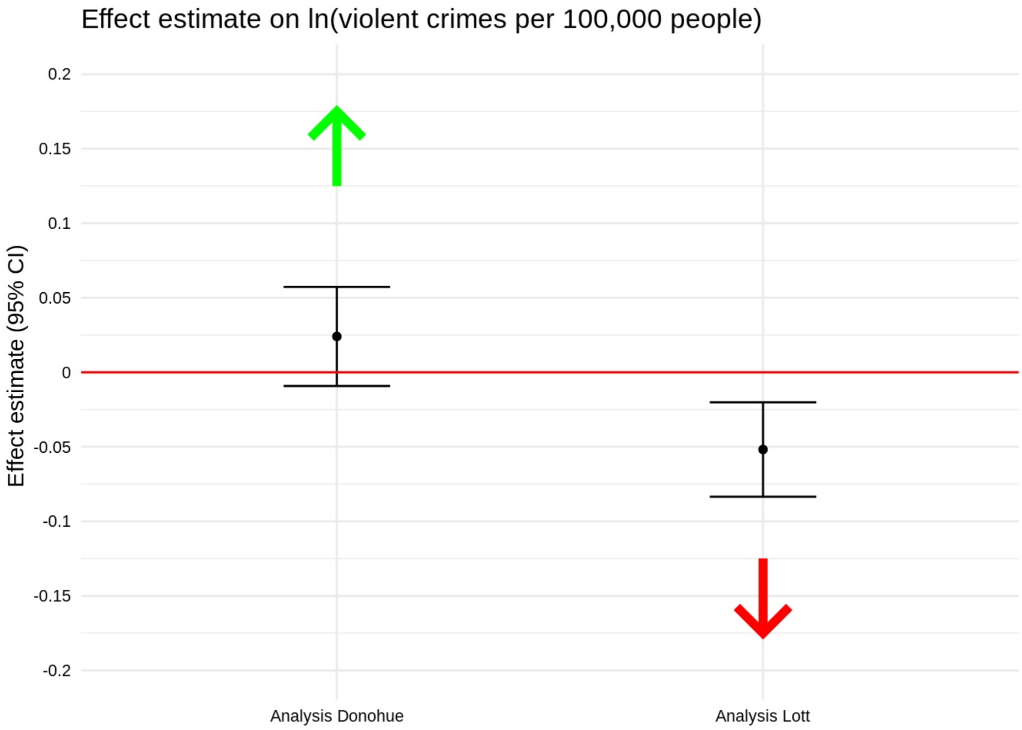
```
LOTT_OUTPUT_TIDY$Analysis <- "Analysis Lott"
```

```
comparing_analyses <- DONOHUE_OUTPUT_TIDY |>
  bind_rows(LOTT_OUTPUT_TIDY) |>
  filter(term == "RTC_LAWTRUE")
comparing_analyses
```

```
## # A tibble: 2 × 8
##   term      estimate std.error statistic p.value conf.low conf.high Analysis
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl> <chr>
## 1 RTC_LAWT…   0.0240    0.0170      1.42 0.156   -0.00919    0.0573 Analysis Do…
## 2 RTC_LAWT…  -0.0518    0.0162     -3.20 0.00139 -0.0835    -0.0201 Analysis Lo…
```

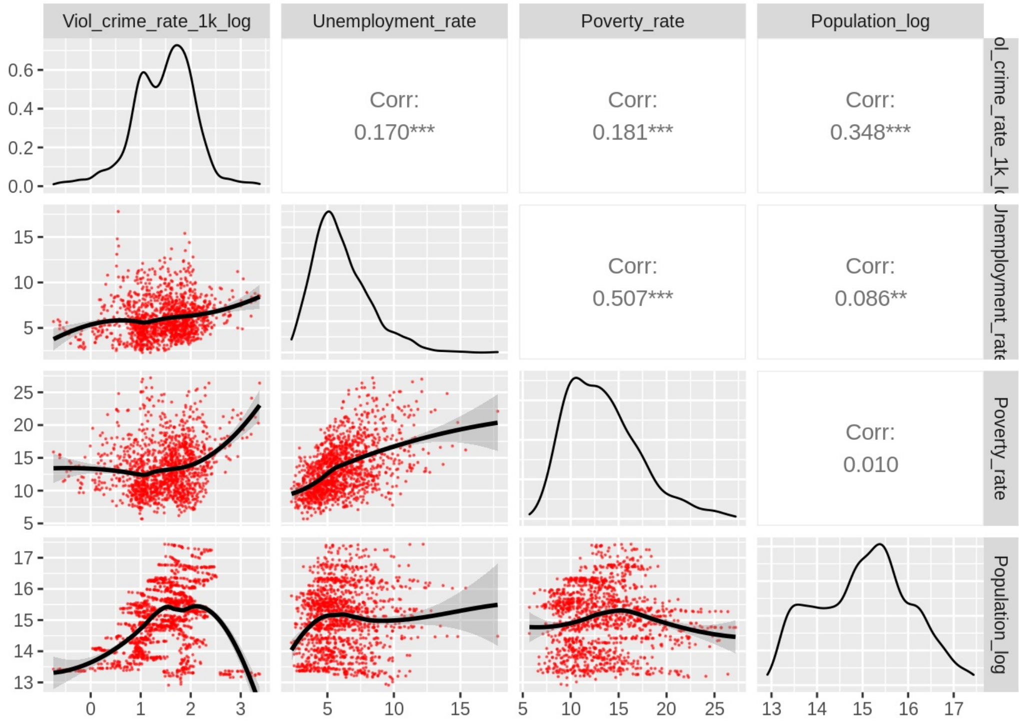
We are grouping DONOHUE and LOTT dataset and view statistical comparison (e.g. standard error, p value, etc)

```
ggplot(comparing_analyses) +
  geom_point(aes(x = Analysis, y = estimate)) +
  geom_errorbar(aes(x = Analysis, ymin = conf.low, ymax = conf.high), width = 0.25) +
  geom_hline(yintercept = 0, color = "red") +
  scale_y_continuous(
    breaks = seq(-0.2, 0.2, by = 0.05),
    labels = seq(-0.2, 0.2, by = 0.05),
    limits = c(-0.2, 0.2)
  ) +
  geom_segment(aes(x = 1, y = 0.125, xend = 1, yend = 0.175),
    arrow = arrow(angle = 45, ends = "last", type = "open"),
    size = 2, color = "green", lineend = "butt", linejoin = "mitre"
  ) +
  geom_segment(aes(x = 2, y = -0.125, xend = 2, yend = -0.175),
    arrow = arrow(angle = 45, ends = "last", type = "open"),
    size = 2, color = "red", lineend = "butt", linejoin = "mitre"
  ) +
  theme_minimal() +
  theme(
    axis.title.x = element_blank(),
    axis.text = element_text(size = 8, color = "black")
  ) +
  labs(
    title = "Effect estimate on ln(violent crimes per 100,000 people)",
    y = "  Effect estimate (95% CI)"
  )
```



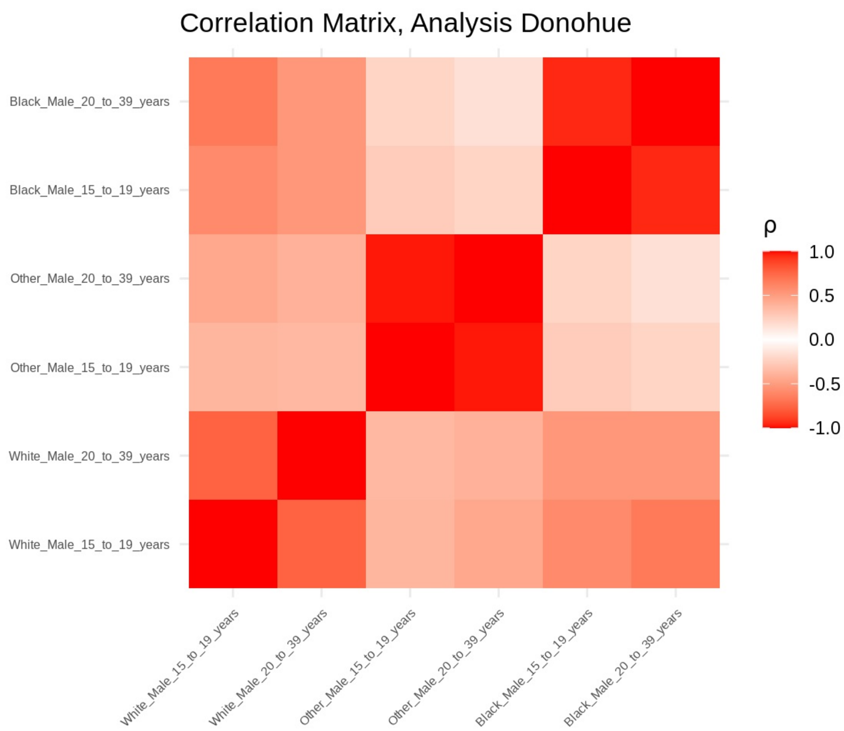
This plots the coefficient estimates for both Donohue and Lott. The red horizontal line at 0 indicates that there is no effect. Since the confidence interval for Donohue overlaps with the line at 0, suggesting that it is likely that the coefficient estimate for Donohue increases, and that we are 95% confident is in within that range.

```
DONOHUE_DF |>
  select(RTC_LAW,
         Viol_crime_rate_1k_log,
         Unemployment_rate,
         Poverty_rate,
         Population_log) |>
  ggpairs(columns = c(2:5),
          lower = list(continuous = wrap("smooth_loess",
                                         color = "red",
                                         alpha = 0.5,
                                         size = 0.1)))
```



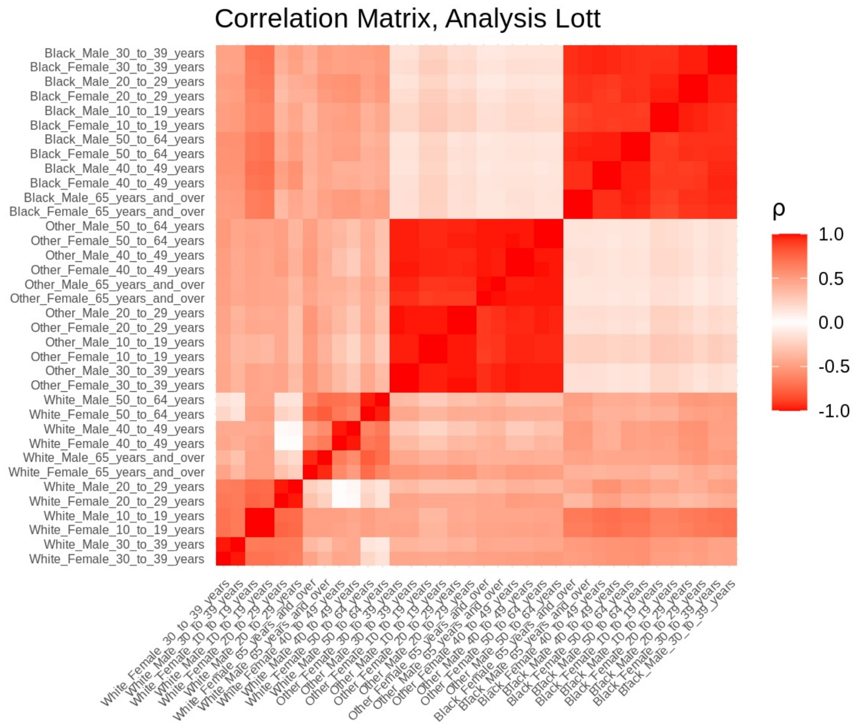
The pairplot above helps us diagnose multicollinearity. The graphs along the diagonal show the distributions of the variables specified on the top. The other graphs show a scatterplot between the variables specified in the row and column. Unemployment rate and poverty rate appear to have similar trends with the graph skewed to the right. Note that they also have the highest correlation as well.

```
cor_DONOHUE_dem <- cor(DONOHUE_DF |> select(contains("_years")))
ggcorrplot(cor_DONOHUE_dem,
  tl.cex = 6,
  hc.order = TRUE,
  colors = c(
    "red",
    "white",
    "red"
  ),
  outline.color = "transparent",
  title = "Correlation Matrix, Analysis Donohue",
  legend.title = expression(rho)
)
```



This heatmap shows that there is a high correlation within race in our Donohue data, with race having a stronger correlation than age. This suggests collinearity of our predictors.

```
cor_LOTT_dem <- cor(LOTT_DF |> select(contains("_years")))
corr_mat_LOTT <- ggcorrplot(cor_LOTT_dem,
  tl.cex = 6,
  hc.order = TRUE,
  colors = c(
    "red",
    "white",
    "red"
  ),
  outline.color = "transparent",
  title = "Correlation Matrix, Analysis Lott",
  legend.title = expression(rho)
)
corr_mat_LOTT
```



This heatmap of the Lott data are clustering by age more than gender. The heatmap can be divided into blocks by race. For instance in the y-axis, the blocks are black on top, other in the middle, and white on bottom.

```
## split data
set.seed(124)
DONOHUE_splits <- d_panel_DONOHUE |> loo_cv()
DONOHUE_splits
```

```
## # Leave-one-out cross-validation
## # A tibble: 1,364 × 2
##    splits           id
##    <list>           <chr>
##  1 <split [1363/1]> Resample1
##  2 <split [1363/1]> Resample2
##  3 <split [1363/1]> Resample3
##  4 <split [1363/1]> Resample4
##  5 <split [1363/1]> Resample5
##  6 <split [1363/1]> Resample6
##  7 <split [1363/1]> Resample7
##  8 <split [1363/1]> Resample8
##  9 <split [1363/1]> Resample9
## 10 <split [1363/1]> Resample10
## # … with 1,354 more rows
```

We use leave one out cross validation (loo_cv()) to check the stability of the coefficient estimates under changes of the data. This takes out one row of the data and does it 1364 times, to get every possible row out of the data.

```
DONOHUE_subsets <- map(pull(DONOHUE_splits, splits), training)
glimpse(DONOHUE_subsets[[1]])
```

```
## Rows: 1,363
## Columns: 20
## $ YEAR                      <fct> 1980, 1981, 1982, 1983, 1984, 1985, 1986, 19…
## $ STATE                     <fct> Alaska, Alaska, Alaska, Alaska, Alaska, Alas…
## $ Black_Male_15_to_19_years <pseries> 0.1670456, 0.1732299, 0.1737069, 0.17095…
## $ Black_Male_20_to_39_years <pseries> 0.9933775, 1.0028219, 1.0204445, 1.03127…
## $ Other_Male_15_to_19_years <pseries> 1.1297816, 1.1244412, 1.0698208, 0.98828…
## $ Other_Male_20_to_39_years <pseries> 2.963329, 2.974775, 3.015071, 3.008048, …
## $ White_Male_15_to_19_years <pseries> 3.627805, 3.558261, 3.391844, 3.222002, …
## $ White_Male_20_to_39_years <pseries> 18.28852, 18.12821, 18.10666, 17.90600, …
## $ Unemployment_rate         <pseries> 9.6, 9.4, 9.9, 9.9, 9.8, 9.7, 10.9, 10.3…
## $ Poverty_rate              <pseries> 9.6, 9.0, 10.6, 12.6, 9.6, 8.7, 11.4, 12…
## $ Viol_crime_count          <pseries> 1919, 2537, 2732, 2940, 3108, 3031, 3046…
## $ Population                <pseries> 404680, 418519, 449608, 488423, 513697, …
## $ police_per_100k_lag       <pseries> 194.7218, 200.2299, 191.0553, 364.2335, …
## $ RTC_LAW_YEAR              <pseries> 1995, 1995, 1995, 1995, 1995, 1995, 1995…
## $ RTC_LAW                   <pseries> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE…
## $ TIME_0                    <pseries> 1980, 1980, 1980, 1980, 1980, 1980, 1980…
## $ TIME_INF                  <pseries> 2010, 2010, 2010, 2010, 2010, 2010, 2010…
## $ Viol_crime_rate_1k        <pseries> 4.742018, 6.061851, 6.076404, 6.019373, …
## $ Viol_crime_rate_1k_log    <pseries> 1.556463, 1.802015, 1.804413, 1.794983, …
## $ Population_log            <pseries> 12.91085, 12.94448, 13.01613, 13.09894, …
```

Takes the DONOHUE_splits we created and splits it into training data. It is then mapped across all the splits we created. There will end up being 1364 different Donohue subsets.

```
fit_nls_on_bootstrap_DONOHUE <- function(subset) {
  plm(Viol_crime_rate_1k_log ~ RTC_LAW +
        White_Male_15_to_19_years +
        White_Male_20_to_39_years +
        Black_Male_15_to_19_years +
        Black_Male_20_to_39_years +
        Other_Male_15_to_19_years +
        Other_Male_20_to_39_years +
        Unemployment_rate +
        Poverty_rate +
        Population_log +
        police_per_100k_lag,
      data = data.frame(subset),
      index = c("STATE", "YEAR"),
      model = "within",
      effect = "twoways")
}
```

This creates a function that can be used for every Donohue subset we created.

```
# Code takes a long time to run
# subsets_models_DONOHUE <- map(DONOHUE_subsets, fit_nls_on_bootstrap_DONOHUE)
# subsets_models_DONOHUE <- subsets_models_DONOHUE |>
#   map(tidy)
```

We use the map function to use our function across every Donohue subsets, and save it into subsets_models_DONOHUE. We cache this chunk so that when we rerun this chunk without changing anything, it will load much faster.

```
# File already saved
# save(subsets_models_DONOHUE,
#   file = "data/wrangled/DONOHUE_simulations.rda")
```

Here we also save the subsets for loading later.

```
set.seed(124)
LOTT_splits <- d_panel_LOTT |> loo_cv()
LOTT_subsets <- map(pull(LOTT_splits, splits), training)
```

We repeat the splitting for the Lott data.

```
fit_nls_on_bootstrap_LOTT <- function(split) {
  plm(LOTT_fmla,
      data = data.frame(split),
      index = c("STATE", "YEAR"),
      model = "within",
      effect = "twoways"
  )
}
```

```r
# Code takes a long time to run
# subsets_models_LOTT <- map(LOTT_subsets, fit_nls_on_bootstrap_LOTT)
#subsets_models_LOTT <- subsets_models_LOTT |>
#  map(tidy)
```

```r
# File already saved
# save(subsets_models_LOTT,
#   file = "data/wrangled/LOTT_simulations.rda")
```

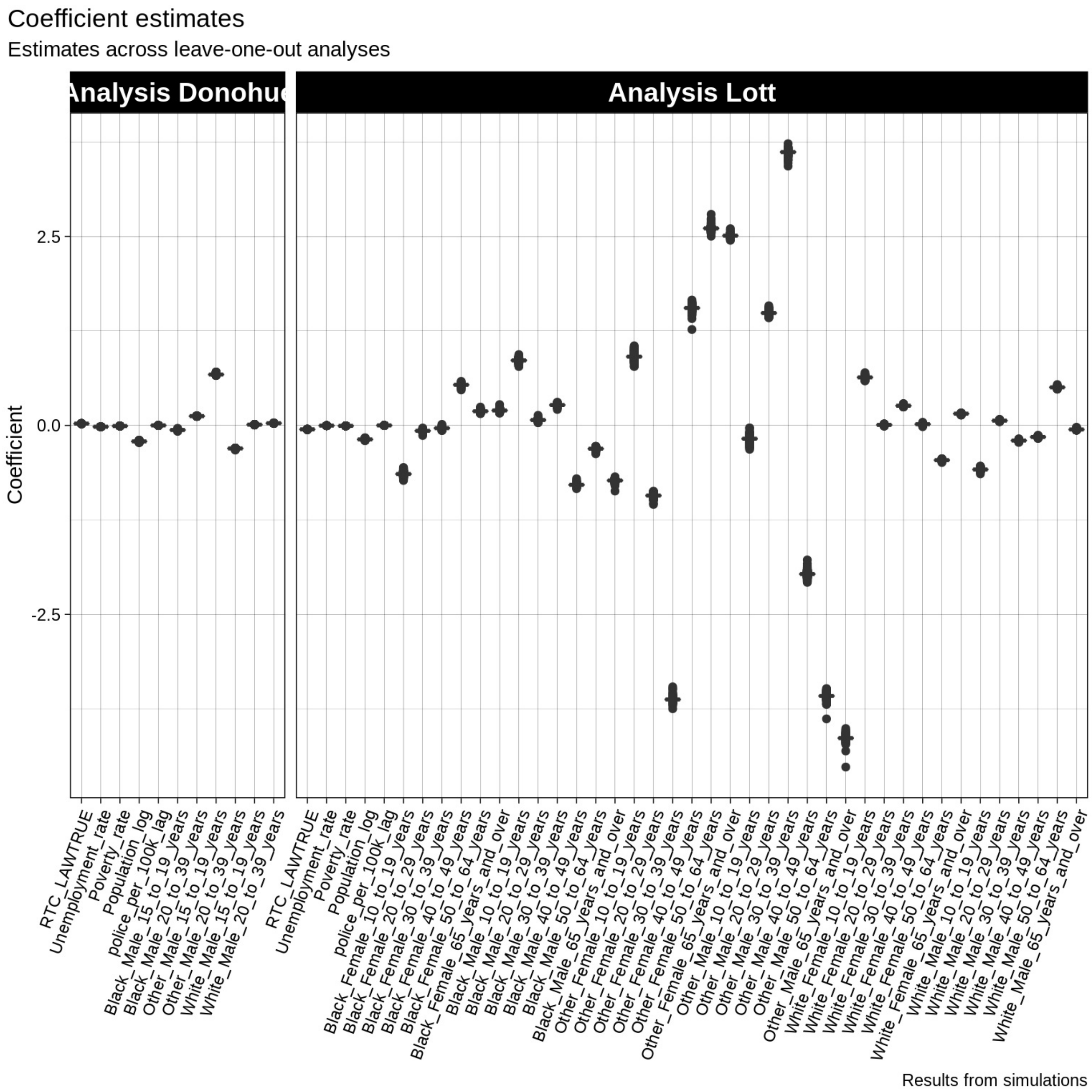
```r
names(subsets_models_DONOHUE) <- paste0("DONOHUE_", seq_len(length(subsets_models_DONOHUE)))
names(subsets_models_LOTT) <-
  paste0("LOTT_", 1:length(subsets_models_LOTT))
```

Here we clean up the results by setting the names of the subsets models with its corresponding data set name and adding a number to the end starting from 1 to the length of the subset.

```r
simulations_DONOHUE <- subsets_models_DONOHUE |>
  bind_rows(.id = "ID") |>
  mutate(Analysis = "Analysis Donohue")
simulations_LOTT <- subsets_models_LOTT |>
  bind_rows(.id = "ID") |>
  mutate(Analysis = "Analysis Lott")
simulations <- bind_rows(simulations_DONOHUE, simulations_LOTT)
# order for easier comparison
simulations <- simulations |>
  mutate(term = factor(term,
    levels = c(
      str_subset(unique(pull(simulations, term)), "years", negate = TRUE),
      sort(str_subset(unique(pull(simulations, term)), "years")))))
```

This binds together all the subsets into their respective datasets, and then creates a new column called Analysis for each data set. Then, they are combined into a single data frame and sorted using levels.

```r
simulations |>
  ggplot(aes(x = term, y = estimate)) +
  geom_boxplot() +
  facet_grid(. ~ Analysis, scale = "free_x", space = "free", drop = TRUE) +
  labs(title = "Coefficient estimates",
       subtitle = "Estimates across leave-one-out analyses",
       x = "Term",
       y = "Coefficient",
       caption = "Results from simulations") +
  theme_linedraw() +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_text(angle = 70, hjust = 1),
        strip.text.x = element_text(size = 14, face = "bold"),
        plot.title.position="plot")
```

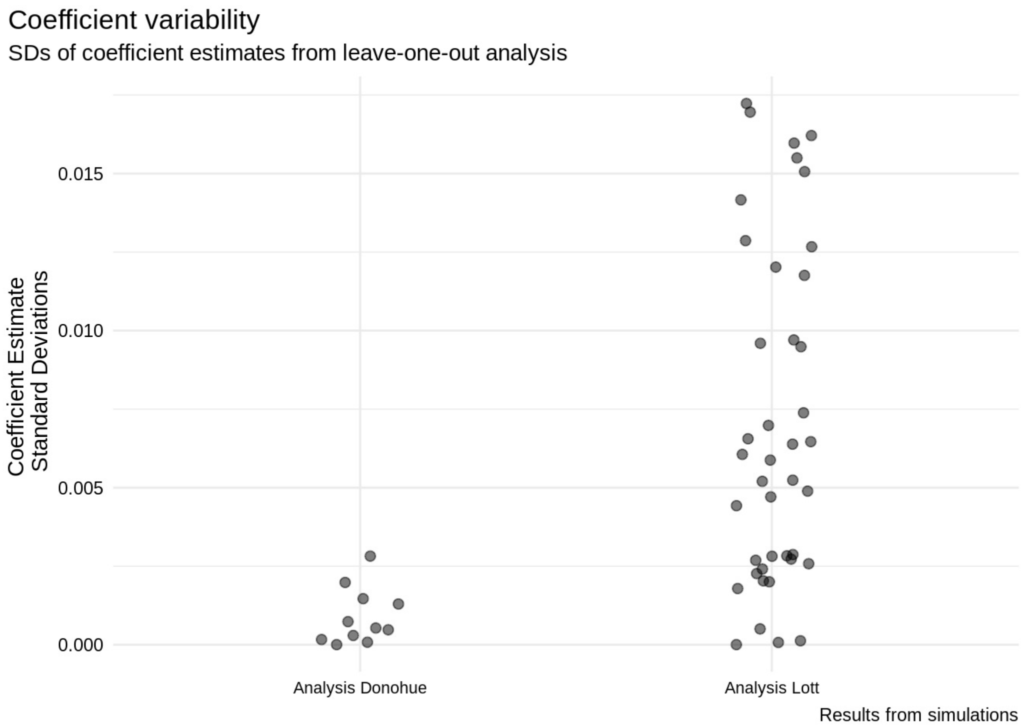


This plots the coefficient estimate with the 95% confidence interval. Coefficients with 0 effect on the outcome are plotted at 0. In the Donohue analysis, most predictors have close to 0 effect on crime. In the Lott analysis, the larger coefficients have a wider confidence interval. This also shows there is much greater variability if we remove one sample at a time with the estimate we got.

```
coeff_sd <- simulations |>
  group_by(Analysis, term) |>
  summarize("SD" = sd(estimate))
```

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

```
coeff_sd |>
  ggplot(aes(x = Analysis, y = SD)) +
  geom_jitter(width = 0.1, alpha = 0.5, size = 2) +
  labs(title = "Coefficient variability",
       subtitle = "SDs of coefficient estimates from leave-one-out analysis",
       x = "Term",
       y = "Coefficient Estimate \n Standard Deviations",
       caption = "Results from simulations") +
  theme_minimal() +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_text(size = 8, color = "black"),
        axis.text.y = element_text(color = "black"),
        plot.title.position="plot")
```



This shows the variability of the coefficients using standard deviation after the leave-one-out analysis. There is much more variance in the Lott analysis, likely due to their wider age range and inclusion of females.

Calculating VIF (Donohue)

```
# create model matrix
lm_DONOHUE_data <- as.data.frame(model.matrix(DONOHUE_OUTPUT))
# define model
lm_DONOHUE_data <- lm_DONOHUE_data |>
  mutate(Viol_crime_rate_1k_log = plm::Within(pull(
    d_panel_DONOHUE, Viol_crime_rate_1k_log
  )), effect = "twoways")
```

Here we define the models for the Donohue data.

```
# specify model
lm_DONOHUE <- lm(Viol_crime_rate_1k_log ~
RTC_LAWTRUE +
  White_Male_15_to_19_years +
  White_Male_20_to_39_years +
  Black_Male_15_to_19_years +
  Black_Male_20_to_39_years +
  Other_Male_15_to_19_years +
  Other_Male_20_to_39_years +
  Unemployment_rate +
  Poverty_rate +
  Population_log +
  police_per_100k_lag,
data = lm_DONOHUE_data
)
# calculate VIF
vif_DONOHUE <- vif(lm_DONOHUE)
```

We create the linear model for Donohue and then calculate the VIF from our model and store it into a variable.

```
# combine into nice table
vif_DONOHUE <- vif_DONOHUE |>
  as_tibble() |>
  cbind(names(vif_DONOHUE)) |>
  as_tibble()
colnames(vif_DONOHUE) <- c("VIF", "Variable")
vif_DONOHUE
```

```
## # A tibble: 11 × 2
##      VIF Variable
##    <dbl> <chr>
##  1  1.11 RTC_LAWTRUE
##  2  1.15 White_Male_15_to_19_years
##  3  1.72 White_Male_20_to_39_years
##  4  1.34 Black_Male_15_to_19_years
##  5  1.66 Black_Male_20_to_39_years
##  6  1.58 Other_Male_15_to_19_years
##  7  1.52 Other_Male_20_to_39_years
##  8  1.23 Unemployment_rate
##  9  1.27 Poverty_rate
## 10  1.17 Population_log
## 11  1.21 police_per_100k_lag
```

Then we clean the results into a table where each predictor has a VIF associated with it the a new column.

Calculating VIF (Lott)

```
lm_LOTT_data <- as.data.frame(model.matrix(LOTT_OUTPUT))
lm_LOTT_data <- lm_LOTT_data |>
  mutate(Viol_crime_rate_1k_log = plm::Within(pull(d_panel_LOTT, Viol_crime_rate_1k_log), effect = "twoways")) |>
  rename(RTC_LAW = RTC_LAWTRUE)
lm_LOTT <- lm(LOTT_fmla, data = lm_LOTT_data)
```

We repeat this process for the Lott data.

```
vif_LOTT <- vif(lm_LOTT)
vif_LOTT
```

```
##                        RTC_LAW    White_Female_10_to_19_years
##                       1.614063                     120.641561
##    White_Female_20_to_29_years    White_Female_30_to_39_years
##                      47.179533                      49.551323
##    White_Female_40_to_49_years    White_Female_50_to_64_years
##                      36.309285                      34.164792
## White_Female_65_years_and_over      White_Male_10_to_19_years
##                      11.584389                     119.516433
##      White_Male_20_to_29_years      White_Male_30_to_39_years
##                      41.760782                      70.830662
##      White_Male_40_to_49_years      White_Male_50_to_64_years
##                      30.870611                      49.953811
##   White_Male_65_years_and_over    Black_Female_10_to_19_years
##                      12.580172                     341.839459
##    Black_Female_20_to_29_years    Black_Female_30_to_39_years
##                     107.596490                      79.668249
##    Black_Female_40_to_49_years    Black_Female_50_to_64_years
##                      97.687605                      66.320239
## Black_Female_65_years_and_over      Black_Male_10_to_19_years
##                      50.153256                     326.610953
##      Black_Male_20_to_29_years      Black_Male_30_to_39_years
##                      91.643419                      90.370825
##      Black_Male_40_to_49_years      Black_Male_50_to_64_years
##                      91.099806                      63.992052
##   Black_Male_65_years_and_over    Other_Female_10_to_19_years
##                      38.106324                     148.187495
##    Other_Female_20_to_29_years    Other_Female_30_to_39_years
##                      67.071204                      54.884463
##    Other_Female_40_to_49_years    Other_Female_50_to_64_years
##                     227.299921                     133.599401
## Other_Female_65_years_and_over      Other_Male_10_to_19_years
##                      83.898466                     156.792520
##      Other_Male_20_to_29_years      Other_Male_30_to_39_years
##                      55.711146                      63.459660
##      Other_Male_40_to_49_years      Other_Male_50_to_64_years
##                     250.840801                     177.496397
##   Other_Male_65_years_and_over              Unemployment_rate
##                      53.771794                       1.507412
##                   Poverty_rate                 Population_log
##                       1.412285                       3.393272
##            police_per_100k_lag
##                       1.732919
```

```
vif_LOTT <- vif(lm_LOTT)|>
  as_tibble() |>
  cbind(names(vif_LOTT)) |>
  as_tibble()
colnames(vif_LOTT) <- c("VIF", "Variable")
vif_LOTT
```

```
## # A tibble: 41 × 2
##       VIF Variable
##     <dbl> <chr>
##  1   1.61 RTC_LAW
##  2 121.   White_Female_10_to_19_years
##  3  47.2  White_Female_20_to_29_years
##  4  49.6  White_Female_30_to_39_years
##  5  36.3  White_Female_40_to_49_years
##  6  34.2  White_Female_50_to_64_years
##  7  11.6  White_Female_65_years_and_over
##  8 120.   White_Male_10_to_19_years
##  9  41.8  White_Male_20_to_29_years
## 10  70.8  White_Male_30_to_39_years
## # … with 31 more rows
```

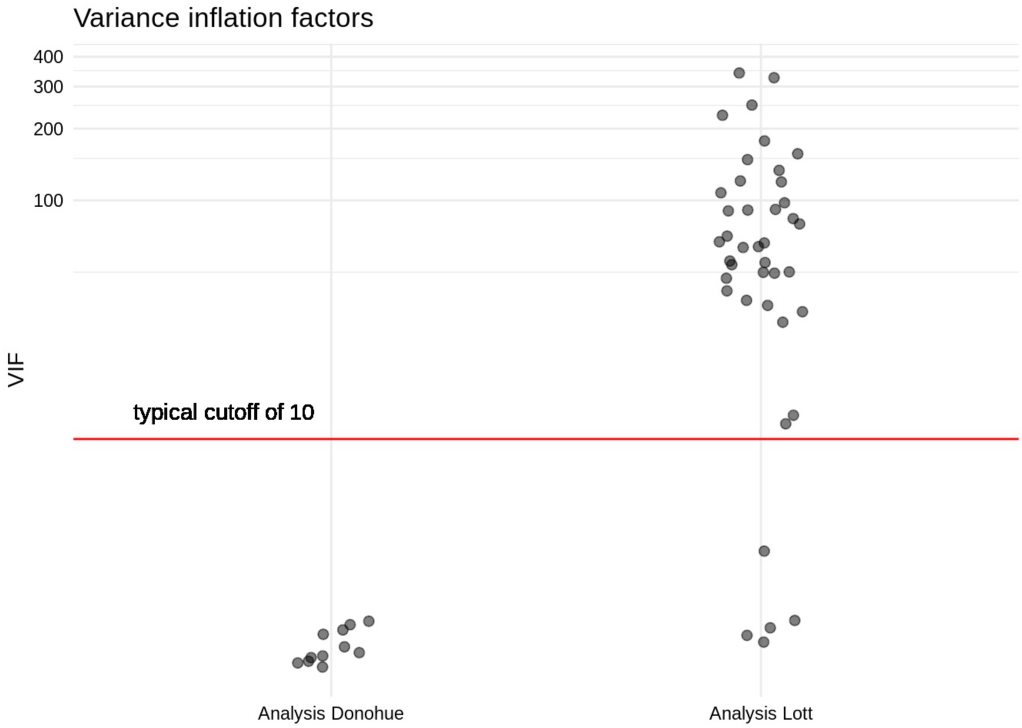
```
# clean up names
vif_LOTT |>
  mutate(Variable = str_replace(string = Variable,
                                pattern = "RTC_LAW",
                                replacement = "RTC_LAWTRUE"))
```

```
## # A tibble: 41 × 2
##       VIF Variable
##     <dbl> <chr>
##  1   1.61 RTC_LAWTRUE
##  2 121.   White_Female_10_to_19_years
##  3  47.2  White_Female_20_to_29_years
##  4  49.6  White_Female_30_to_39_years
##  5  36.3  White_Female_40_to_49_years
##  6  34.2  White_Female_50_to_64_years
##  7  11.6  White_Female_65_years_and_over
##  8 120.   White_Male_10_to_19_years
##  9  41.8  White_Male_20_to_29_years
## 10  70.8  White_Male_30_to_39_years
## # … with 31 more rows
```

```
vif_DONOHUE <- vif_DONOHUE |>
  mutate(Analysis = "Analysis Donohue")
vif_LOTT <- vif_LOTT |>
  mutate(Analysis = "Analysis Lott")
vif_df <- bind_rows(vif_DONOHUE, vif_LOTT) # combine analysis of DONOHUE_DF, LOTT_DF
```

Here we combine our VIF results from Donohue and Lott into a single data frame after creating a new column in each, specifying which analysis they are from.

```
vif_df |>
  ggplot(aes(x = Analysis, y = VIF)) +
  geom_jitter(width = 0.1, alpha = 0.5, size = 2) +
  geom_hline(yintercept = 10, color = "red") +
  geom_text(aes(.75, 13, label = "typical cutoff of 10")) +
  coord_trans(y = "log10") +
  labs(title = "Variance inflation factors") +
  theme_minimal() +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_text(color = "black"),
        axis.text.y = element_text(color = "black"))
```



The cutoff of 10 indicates that any feature in our model with a value greater than 10 is multicollinear with something else in the dataset. The majority of predictors in the Lott analysis have multicollinearity, meaning most of the predictors are likely to have inaccurate coefficients since we are unsure about if the estimate is actually for that predictor or a combination of predictors.

Results

Multicollinearity can lead to uncertainty of whether our predictors are giving us accurate coefficient estimates. From our leave one out analysis, we found that there were certain age groups that had a very high or low coefficient estimate, especially in the Lott data. Those predictors are unstable and causing our linear regression models to become less accurate. From our VIF calculations, we found that the majority of the Lott data suffered from multicollinearity using a cutoff of 10. Many subsets of the datasets are well above 10, thus it indicates that there are a lot of subsets have a multicollinearity problem.

Extended Analysis

Question

We found out that there are a lot of variables that are well above 10 (cutoff line) of VIF for LOTT dataset, thus we want to ask the question: does creating a new dataset that combining some age range help minimizing the effect of collinearly?

```
# modify age_group
dem_new <- dem |>
  filter(!(AGE_GROUP %in% LOTT_AGE_GROUPS_NULL) )|> # remove age group "Under 5 years" and "5 to 9 years"
  mutate(AGE_GROUP = fct_collapse(AGE_GROUP,
                               "10 to 64 years"=c("10 to 14 years", "15 to 19 years", "20 to 24 years",
                                                   "25 to 29 years", "30 to 34 years", "35 to 39 years",
                                                   "40 to 44 years", "45 to 49 years", "50 to 54 years",
                                                   "55 to 59 years", "60 to 64 years"),
                               "65 years and over"=c("65 to 69 years", "70 to 74 years",
                                                      "75 to 79 years", "80 to 84 years",
                                                      "85 years and over")))
```

The above code creates age_group for NEW_DF. Unlike the LOTT_DF data, we would have 2 age groups, “10 to 14 years” and “65 years and over”.

```
dem_new <- dem_new |>
  mutate(AGE_GROUP = str_replace_all(AGE_GROUP, " ", "")) |> # remove white space
  group_by(YEAR, STATE, RACE, SEX, AGE_GROUP) |>
  summarize(PERC_SUB_POP = sum(PERC_SUB_POP), .groups = "drop") |>
  unite(col = "VARIABLE", RACE, SEX, AGE_GROUP, sep = "") |> #combining RACE, SEX, and AGE_GROUP and make it a ne
w col
  rename("VALUE" = PERC_SUB_POP)
```

Similar to the process of demographic data for LOTT, above code changes the dem_new data to match our other data sets.

```
# form new dataset NEW_DF
NEW_DF <- bind_rows(dem_new,
                    ue_rate_data,
                    poverty_rate_data,
                    crime_data,
                    population_data,
                    ps_data) |>
  pivot_wider(names_from = "VARIABLE",
              values_from = "VALUE") |>
  left_join(RTC , by = c("STATE")) |> # combine RTC data
  mutate(RTC_LAW = case_when(YEAR >= RTC_LAW_YEAR ~ TRUE,
                             TRUE ~ FALSE)) |>
   drop_na()
```

The code above combines all the subsets and save the output in NEW_DF.

```
NEW_DF <- NEW_DF |>
  mutate(Viol_crime_rate_1k = (Viol_crime_count*1000)/Population, # calculate crime rate per 1k people
         Viol_crime_rate_1k_log = log(Viol_crime_rate_1k), # calculate log of crime rate per 1k people
         Population_log = log(Population)) # calculate log of pop
```

Above code gets the crime rate per 1000 people, its log, and the log for population.

```
# formula for New df
new_LOTT_variables <- NEW_DF |> # get variables from NEW_DF
  select(RTC_LAW,
         contains(c("White", "Black", "Other")),
         Unemployment_rate,
         Poverty_rate,
         Population_log,
         police_per_100k_lag) |>
  colnames()
new_LOTT_fmla <- as.formula(paste("Viol_crime_rate_1k_log ~", paste(new_LOTT_variables, collapse = " + ") #add va
riables to formula using paste()
))
new_LOTT_fmla
```

```
## Viol_crime_rate_1k_log ~ RTC_LAW + WhiteFemale10to64years + WhiteFemale65yearsandover +
##     WhiteMale10to64years + WhiteMale65yearsandover + BlackFemale10to64years +
##     BlackFemale65yearsandover + BlackMale10to64years + BlackMale65yearsandover +
##     OtherFemale10to64years + OtherFemale65yearsandover + OtherMale10to64years +
##     OtherMale65yearsandover + Unemployment_rate + Poverty_rate +
##     Population_log + police_per_100k_lag
```

The code above forms new formula for NEW_DF.

```
# model fitting
new_d_panel_LOTT <- pdata.frame(NEW_DF, index = c("STATE", "YEAR"))
new_LOTT_OUTPUT <- plm(new_LOTT_fmla,
                   model = "within",
                   effect = "twoways",
                   data = new_d_panel_LOTT
)
```

Above code fits the panel linear regression model with NEW_DF data.

```
# specify model
new_lm_LOTT_data <- as.data.frame(model.matrix(new_LOTT_OUTPUT))
new_lm_LOTT_data <- new_lm_LOTT_data |>
  mutate(Viol_crime_rate_1k_log = plm::Within(pull(new_d_panel_LOTT, Viol_crime_rate_1k_log), effect = "twoways")
) |>
  rename(RTC_LAW = RTC_LAWTRUE)
new_lm_LOTT <- lm(new_LOTT_fmla, data = new_lm_LOTT_data)
```

```
# calculate VIF
new_vif_LOTT <- vif(new_lm_LOTT)
```

```r
# clean up
new_vif_LOTT <- new_vif_LOTT |>
  as_tibble() |>
  cbind(names(new_vif_LOTT)) |>
  as_tibble()
colnames(new_vif_LOTT) <- c("VIF", "Variable")
# show cleaned table
new_vif_LOTT
```

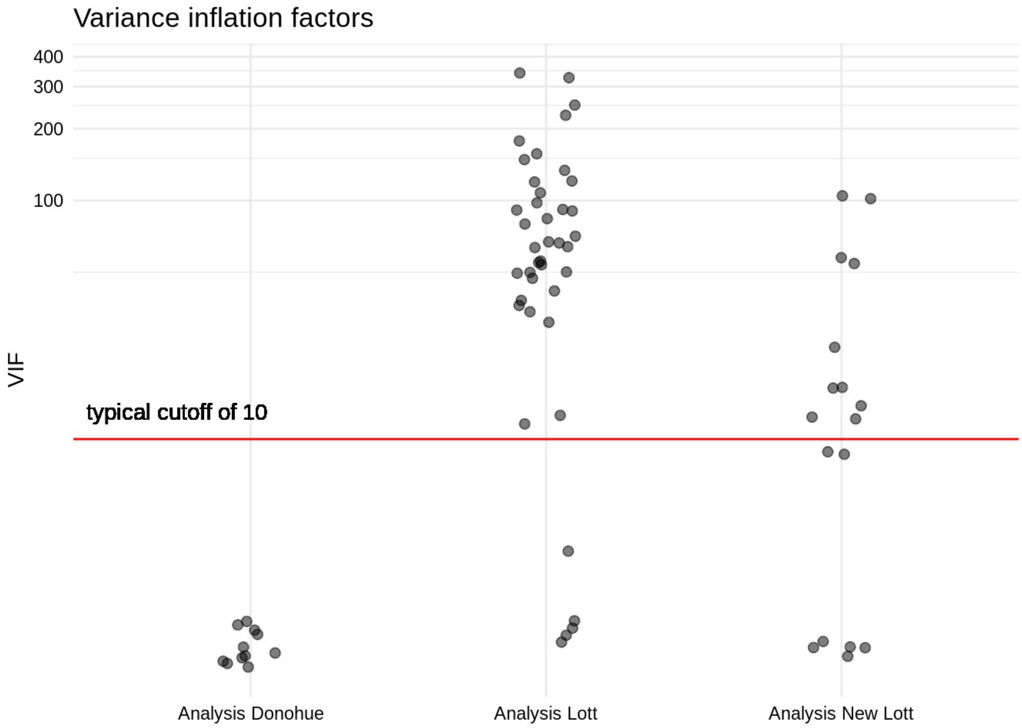
```
## # A tibble: 17 × 2
##       VIF Variable
##     <dbl> <chr>
##  1   1.34 RTC_LAW
##  2  16.5  WhiteFemale10to64years
##  3   8.64 WhiteFemale65yearsandover
##  4  24.3  WhiteMale10to64years
##  5   8.85 WhiteMale65yearsandover
##  6  54.4  BlackFemale10to64years
##  7  12.2  BlackFemale65yearsandover
##  8  57.6  BlackMale10to64years
##  9  12.4  BlackMale65yearsandover
## 10 105.   OtherFemale10to64years
## 11  16.4  OtherFemale65yearsandover
## 12 102.   OtherMale10to64years
## 13  13.8  OtherMale65yearsandover
## 14   1.34 Unemployment_rate
## 15   1.35 Poverty_rate
## 16   1.42 Population_log
## 17   1.23 police_per_100k_lag
```

The code above clean up the table and make it more readable.

```r
# across analysis
new_vif_LOTT <- new_vif_LOTT |>
  mutate(Analysis = "Analysis New Lott") # analysis for NEW_DF
new_vif_df <- bind_rows(vif_DONOHUE, vif_LOTT, new_vif_LOTT) # combine analysis of DONOHUE_DF, LOTT_DF and NEW_DF
```

The code above binds the VIF analysis of LOTT_DF, DONOHUE_DF and the analysis of NEW_DF together and store it in new_vif_df

```r
# plotting the comparison of new df, lott df, and donohue df
new_vif_df |>
  ggplot(aes(x = Analysis, y = VIF)) +
  geom_jitter(width = 0.1, alpha = 0.5, size = 2) +
  geom_hline(yintercept = 10, color = "red") +
  geom_text(aes(.75, 13, label = "typical cutoff of 10")) +
  coord_trans(y = "log10") +
  labs(title = "Variance inflation factors") +
  theme_minimal() +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_text(color = "black"),
        axis.text.y = element_text(color = "black"))
```



Above code shows the plot for VIF values of NEW_DF compares to the VIF of LOTT_DF and DONOHUE_DF. Based on the graph, we can see that although more than half of the variables in NEW_DF still suffer from collinearity (VIF > 10), there are large improvements on the NEW_DF because more variables have the VIF values less than 10 than the LOTT_DF, and NEW_DF has increased number of variables that have the VIF closer to 10.

Discussion of Extended Analysis

We came into eliminating the effect of collinearity on some variables in LOTT_DF by combining some age range. Based on our analysis, we can say that the change we made on age_group decreases the strong correlation within race and helps reducing the interfere from collinearity on the accuracy of a model, but there may be more factors that we did not encounter of. For instance, some particular age group might have had lower VIF scores than other particular age group, but by simply combining every age group, though we have lowered the VIF scores for age factor, we might not have made our LOTT_DF be more accurate than before. This suggests that future work is required to lower the effect that approaches the level of effect on DONOHUE_DF.

Conclusion

In this Case studies, we were able to import, wrangle, and make analysis with various dataset that is related with Right to carry guns, population size, demographics, etc. Specifically, we created to see relevance between several dataset like Police Staffing, Crime rate, etc. In our EDA, we found that the crime rates fluctuated over the years with an overall decrease, and the police presence steadily increased overall. Our analysis of the effect estimates of the Donohue and Lott datasets led us to diagnose for multicollinearity. After using leave-one-out analysis and doing VIF calculations, we found that the majority of the Lott dataset suffered from multicollinearity, leading to less accurate linear models. Some of the age groups affected the multicollinearity more than others, so we decided to reduce the impact by grouping them together. We grouped all the ages from 10 to 64 together and after running another analysis, we found a smaller effect of multicollinearity on our new datasets.

References

(1) https://www.nraila.org/get-the-facts/right-to-carry-and-concealed-carry/ (https://www.nraila.org/get-the-facts/right-to-carry-and-concealed-carry/)

(2) https://www.opencasestudies.org/ocs-bp-RTC-wrangling/ (https://www.opencasestudies.org/ocs-bp-RTC-wrangling/)

(3) https://www.investopedia.com/terms/m/multicollinearity.asp (https://www.investopedia.com/terms/m/multicollinearity.asp)