

Police Militarization Data Exercise

The dataset `data.csv` contains information on the degree of militarization of local police forces in the US, along with some demographic information. The variables are:

- `fips`: unique identifier for a given county (FIPS code)
- `year`: the observation year
- `county_name`: name of the county corresponding to the FIPS code
- `unemp`: county-level unemployment rate (in percent)
- `income`: county median household income in dollars
- `pov`: county poverty rate (in percent)
- `mil equip received`: total value of military equipment acquired by local police forces in a given county/year (in dollars)

The second dataset `data2.csv` contains the following variables:

- `county`: unique identifier for a given county (FIPS code)
- `year`: the observation year
- `pop`: population (number of inhabitants in given county/year)

Your task is the following:

- (i) Merge '`data2.csv`' with '`data.csv`' based on the county identifier (FIPS code) and the year. Make sure all the observations in `data.csv` are kept, even if no matching observation exists in '`data2.csv`'.
- (ii) Get familiar with the data by providing a very brief overview of the variables '`income`', '`pov`', '`pop`', and '`mil equip received`'. Where this may be the case, record anything that may be a potential issue.
- (iii) Compute the correlations between (i) `income` and `mil equip received` and (ii) `pov` and `mil equip received` - does it look like more material tends to go to richer counties?
- (iv) Create a variable '`cumulative equip received`' which contains the total value of all the equipment received by the county in question up until (and including) the year of the observation.
- (v) Create a new dataframe containing the average amount of equipment received per capita in every year and plot this series (with the year on the x-axis).
- (vi) Which county that never acquired military equipment since 2000 has made the largest initial equipment acquisition, and in what year?

In evaluating the project we will not only look at whether the submitted code is correct, but also at: (i) whether it is efficient and concise, (ii) whether the code is clean, easy-to-read, and well-commented, and (iii) how quickly the code was produced. If you spot any issue with the data, please address it as you deem fit.

To be able to evaluate (iii), we ask that you accurately record the total time spent on the project beginning when you first read through the instructions.