BST 270: Individual Project

Reproducible Data Science: Police Settlements (FiveThirtyEight)

Shanta Murthy

Here we will be using an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

The article we will attempt to reproduce the results from is titled Cities Spend Millions On Police Misconduct Every Year. Here's Why It's So Difficult to Hold Departments Accountable.

The article has a corresponding data and code available for preprocessing which was used for reproducing select figures shown here, accessible on the corresponding original GitHub

Additionally, multiple levels of the processed files and separate codes for preprocessing steps that were conducted were provided for each city's data separately, including explanations of how the data was formatted and cleaned with respect to the available data for each location.

Sample outputs are provided in the output folder and the knitted file is contained in the parent directory.

Step 0. Load libraries, install if not already loaded

The packages that have been loaded are listed below and the sessionInfo() at the bottom of this document (in the Appendix) provides the version information for each of the packages. If any of these packages have not already been installed, please install first using install.packages() and then load the library with library() before proceeding.

Step 1. Load corresponding data files for all the following figures.

We will attempt to reproduce the barplot of the Cleveland settlement following Rice's death ("Figure 1"), comparison of misleading categorization between Cincinnati and Charleston ("Figure 2"), and Large settlements can skew a city's average payment ("Figure 3")) from the original FiveThirtyEight article on police settlements.

First, we will load the data files that have been provided in the corresponding FiveThirtyEight GitHub.

```
# For Figure 1 - intermediate file from Cleveland Police Department dataset,
 ↪  can be loaded as follows (with respect to this folder, this is where we
 ↪  have stored the intermediate file)
intermediate_cleveland <-
 ↪  readxl::read_xlsx(here::here('data/intermediate/clevelandstart.xlsx'))

#For Figures 2 and 3 - load each of the final data files from GitHub page.
# Define a list of cities with their respective paths and labels
city_info <- list(
  springfield     = c("springfield_edited.csv", "Springfield, MA"),
  milwaukee       = c("milwaukee_edited.csv", "Milwaukee"),
  los_angeles     = c("los_angeles_edited.csv", "Los Angeles"),
  san_francisco   = c("san_francisco_edited.csv", "San Francisco"),
  washington_dc   = c("DC_edited.csv", "Washington, D.C."),
  chicago         = c("chicago_edited.csv", "Chicago"),
  st_louis        = c("stlouis_edited.csv", "St. Louis"),
  baltimore       = c("baltimore_edited.csv", "Baltimore"),
  boston          = c("boston_edited.csv", "Boston"),
  cleveland       = c("cleveland_edited.csv", "Cleveland"),
  little_rock     = c("little_rock_edited.csv", "Little Rock"),
  new_orleans     = c("new_orleans_edited.csv", "New Orleans"),
  waterbury       = c("waterbury_edited.csv", "Waterbury, CT"),
  detroit         = c("detroit_edited.csv", "Detroit"),
  orlando         = c("orlando_edited.csv", "Orlando"),
  miami           = c("miami_edited.csv", "Miami"),
  paterson        = c("paterson_edited.csv", "Paterson, NJ"),
  atlanta         = c("atlanta_edited.csv", "Atlanta"),
  philadelphia    = c("philly_edited.csv", "Philadelphia"),
  baton_rouge     = c("baton_rouge_edited.csv", "Baton Rouge"),
  indianapolis    = c("indianapolis_edited.csv", "Indianapolis"),
  nyc             = c("new_york_edited.csv", "New York City"),
  cincinnati      = c("cincinnati_edited.csv", "Cincinnati"),
  columbia        = c("columbia_edited.csv", "Columbia"),
  north_charleston= c("north_charleston_edited.csv", "North Charleston, SC"),
  charleston      = c("charleston_edited.csv", "Charleston, SC"),
  memphis         = c("memphis_edited.csv", "Memphis, TN"),
  fort_lauderdale = c("fort_lauderdale_edited.csv", "Fort Lauderdale, FL"),
  roanoke         = c("roanoke_edited.csv", "Roanoke, VA"),
  cambridge       = c("cambridge_edited.csv", "Cambridge, MA"),
```

```
  richmond        = c("richmond_edited.csv", "Richmond, VA")
)

# Create an empty list to store data
city_datasets <- list()

# Save out file name and lab
for (city in names(city_info)) {
  filename <- city_info[[city]][1]
  label <- city_info[[city]][2]

  # Read and mutate final, processed data (stored in processed folder) to
   ↪  name city_label for column
  city_datasets[[city]] <- read.csv(here::here("data/processed", filename))
 ↪  %>%
    mutate(city_label = label)
}

# Assign each dataset to a variable - this will create a variable of each
 ↪  city name separately which will be used in Figure 2 and Figure 3
 ↪  generation
for (city in names(city_datasets)) {
  assign(city, city_datasets[[city]])
}

#These are now stored according to their respective city.
```

Step 2. Figure 1: 'Cleveland's settlement amounts rose after Rice's death'

Using intermediate data file corresponding to the Cleveland dataset, we are able to conduct the preprocessing steps described in cleveland_oh.R corresponding script with function to generate plot. As the original GitHub states, they were provided settlement data for cases paid out between 2010 and 2020. In this calendar year corresponds to the year the settlement was paid out and there was no information regarding the type of misconduct, so this was not used in the filtering process. The authors stated that while some cases included settlements that were paid out in multiple installments over more than 1 year, the data has been defined as amount Cleveland paid per year rather than the number of cases settled in each year. The preprocessing steps have been copied into the script sourced below, and we were able to conduct the same preprocessing and get the same final outputs using the checks provided in the original script. Thus the preprocessing steps appear to be consistent with those that the

authors had used. We have additionally added the steps for visualizing the data which was not provided in the published FiveThirtyEight script, and visualize it below.

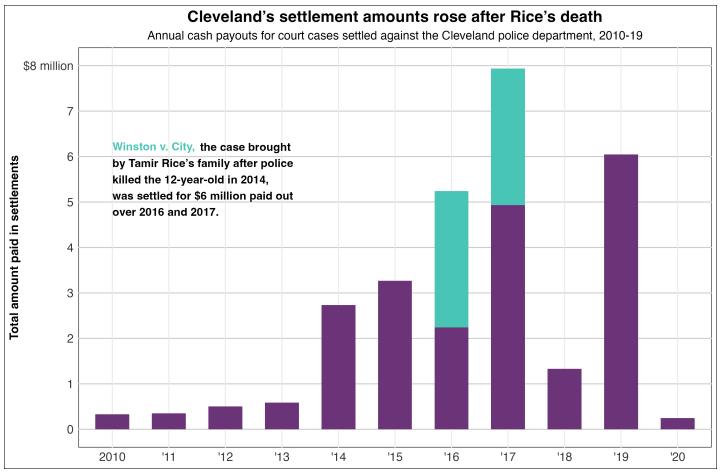
```
#Use intermediate_cleveland file to process data appropriately (described in
 ↪  corresponding R script)
source(here::here('helper_functions/Figure1_policesettlements.R'))

#Visualize figure 1 plot

F1 <- figure1(intermediate_cleveland)
```

```
[1] "There are 142 rows missing closed date"
[1] "There are 0 rows missing calendar year"
[1] "There are 0 rows missing amount awarded"
[1] "There are 0 rows with amount awarded = 0"
[1] "There are 0 rows missing docket number"
[1] "Total number of cases"
[1] 142
[1] "Total amount awarded"
```

```
ggsave(filename = here::here('plots/', "Figure1_output.png"),
       plot = F1,
       bg = "white", width = 10, height = 6.5, units = "in")
knitr::include_graphics(here::here("plots", "Figure1_output.png"))
```



From the visualization we see that the size of the bars appears to be consistent with what is represented in the original article, and thus the authors provided the necessary information

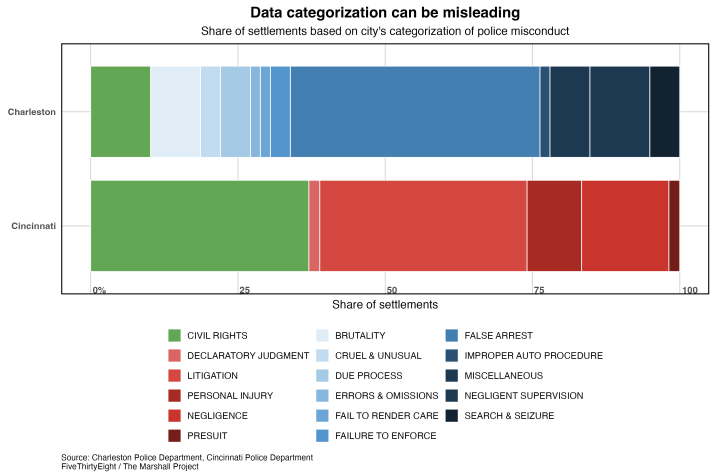
to reproduce the figure. Note that as described in the original article, the Cleveland Police Department from which the data was originally collected from did not convey descriptions of the misconduct that led to the settlements shown above, so there may be other cases that vary in severity still included in the annual totals, such as 'payments for car accidents' (example described from article).

Step 3. Figure 2: Comparison of allegations between Cincinnati versus Charleston: 'Data categorization can be misleading'

This figure represents discrepancies and cautions readers from comparing data from police departments of different states due to their differences in categorization of the corresponding allegations for the settlements. Using the final files for Cincinnati and Charleston shared by the authors of the FiveThirtyEight article, who had processed the data provided from their respective city police departments, we loaded those tables and are sourcing the code with stacked barplot visual below. The authros described that for Cincinnati cases closed between 2010-2020, some of the allegation categorization were ambiguous, with labels like 'litigation' and 'negligence'. For the Charleston dataset, the authors stated that they received PDFs from 2010-2019, a slightly different time frame, and manually transferred some of the data with "Law Enforcement" within the label. The manual transferring of the data is a bit concerning for the possibility of errors, and additionally, the authors have noted that for this dataset there is uncertainty of whether the data includes both claims and settlements. The corresponding data dictionaries are listed for Cincinnati, OH and Charleston, SC, respectively.

```
source(here::here('helper_functions/Figure2_cincinnati_vs_charleston.R'))
F2 <- figure2(cincinnati, charleston)
ggsave(filename = here::here('plots/', "Figure2_output_barplot.png"),
       plot = F2,
       bg = "white", width = 12, height = 8, units = "in")

knitr::include_graphics(here::here("plots", "Figure2_output_barplot.png"))
```



The plot was not entirely reproduced the same way, including differences in the aesthetic additions in the original image, such as captioning the comparison between Charleston and Cincinnati that says that by the length of the bar it appears that Cincinnati would have more civil rights. Additionally the grouping of the legend variables. However, using geom_bar(stat = "count", position = "fill") parameters allowed full expansion such that the settlement categories altogether covered 100%, and each of the bars and their sizes visually appear similar to the original image. As the authors described how they filtered the data and it was clear that the visualized stacked variable was summary_allegations, which are colored similarly as described to the legend, quantitatively the figure represents the same information. However, the final plot we have generated does not bring attention to the categories that are only found in one of the cities versus both of the cities as the original figure does, and as the authors noted in the paper that the naming and sub-classifications or detailed levels in each city varied. The percentages were also added using a manual annotation tool to equally space out the quarter percentages and it is unclear how these visuals were made if done in R, as this code has not been provided by the authors.

Step 4. Figure 3: Comparison of average and median incomes per city in study

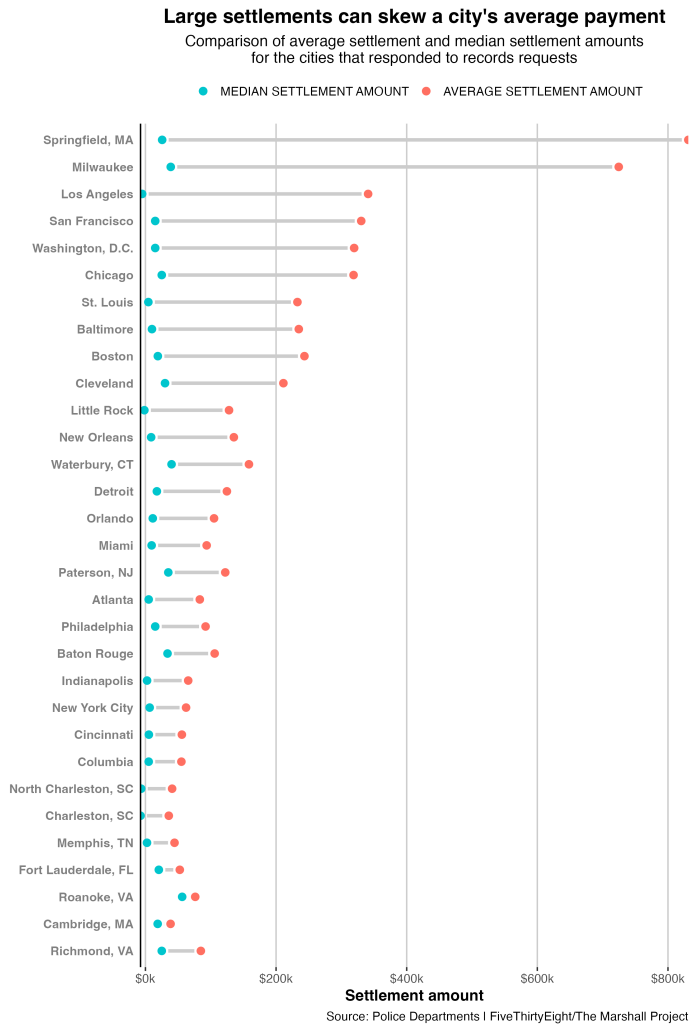
The last figure that was attempted to be reproduced was the final figure of the article, labeled "Large settlements can skew a city's average payment". The final processed data collected from 31 city police departments was loaded (obtained from FiveThirtyEight's GitHub), and the corresponding csv names are stored in city_info. As the article states, the authors asked for detailed information of data dictionaries in the public record requests (conducted in partnership between FiveThirtyEight and The Marshall Project), but they only received a detailed data dictionary from New York City. The authors conveyed that there was ambiguity in handling the multi-year totals for the calculation of settlement amounts, as well as how Additionally, the authors left scratching our heads about how to understand the data underneath the multi-year totals. The authors stated that they wanted to convey that the settlement amount (arbitrarily

decided based on how dates and settlement and misconduct categorizations were defined per each city, and uncertainty about the completeness of records) is hard to make clear conclusions or interpret, as we lack information of the specific settlements, and the measures of center of the settlements may not indicate the severity of the case.

```
#Create dataframe of combined data from each of the cities
full <- bind_rows(springfield, milwaukee, los_angeles, san_francisco,
 ↪  washington_dc,
                   chicago, st_louis, baltimore, boston, cleveland,
 ↪  little_rock,
                   new_orleans, waterbury, detroit, orlando, miami,
 ↪  paterson, atlanta,
                   philadelphia, baton_rouge, indianapolis, nyc,
 ↪  cincinnati, columbia,
                   north_charleston, charleston, memphis, fort_lauderdale,
 ↪  roanoke,
                   cambridge, richmond)

# Combine all datasets into a single dataframe
source(here::here('helper_functions/Figure3_output_segmentplot.R'))
F3 <- figure3(all_data = full)
ggsave(filename = here::here('plots/', "Figure3_output_segmentplot.png"),
       plot = F3,
       bg = "white", width = 9, height = 13, units = "in")

knitr::include_graphics(here::here("plots",
 ↪  "Figure3_output_segmentplot.png"))
```



As shown above, we have attempted to reproduce the final figure from the article. Note that outside of aesthetics, we do notice that although most values appear to be similar to the original article for the median and average settlement amount, we do see a difference for the gap between Richmond, VA median and average settlement amounts. Reviewing the corresponding original github readme file for the Richmond dataset, the authors mentioned that seven follow-up cases were added, which may have been included after the article got published. It is unclear which cases contributed to a smaller difference between median and average. Additionally, some cases, like Roanoke and Cambridge had very close or the same median and average values when computed from the shared final files. However, the final plot included in the article shows half of the "blue" and "red" dots near each other. I did not figure out how to reproduce these elements, and attempted to make the plot similar by using a position nudge() function within ggplot. While the final files were provided, it is clear that there is some subjectivity in the inclusion of certain cases, and uncertainty in defining the time

frames.

Appendix. Session Info

```
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: aarch64-apple-darwin20
Running under: macOS Sonoma 14.5

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  L

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] scales_1.3.0    tinytex_0.54    ggplot2_3.5.1   stringr_1.5.1
[5] lubridate_1.9.4 dplyr_1.1.4     tidyr_1.3.1     here_1.0.1
[9] readxl_1.4.3

loaded via a namespace (and not attached):
 [1] gtable_0.3.6      jsonlite_1.8.9    compiler_4.4.2    tidyselect_1.2.1
 [5] textshaping_0.4.1 systemfonts_1.1.0 yaml_2.3.10       fastmap_1.2.0
 [9] R6_2.5.1          labeling_0.4.3    generics_0.1.3    knitr_1.49
[13] tibble_3.2.1      munsell_0.5.1     rprojroot_2.0.4   pillar_1.10.1
[17] rlang_1.1.4       stringi_1.8.4     xfun_0.50         timechange_0.3.0
[21] cli_3.6.3         withr_3.0.2       magrittr_2.0.3    digest_0.6.37
[25] grid_4.4.2        rstudioapi_0.17.1 lifecycle_1.0.4   vctrs_0.6.5
[29] evaluate_1.0.1    glue_1.8.0        farver_2.1.2      cellranger_1.1.0
[33] ragg_1.3.3        colorspace_2.1-1  rmarkdown_2.29    purrr_1.0.2
[37] tools_4.4.2       pkgconfig_2.0.3   htmltools_0.5.8.1
```