Homework 6: Table Making

Econ 245M

Overview

For this homework assignment, you will be replicating two tables from the paper (Anwar et. al 2022) on the consequences of unequal jury representation in Harris County, Texas. The goal of this assignment is to get you comfortable making publication ready tables *without any manual editing*. This is steep fixed cost, but it will save you hours in the future and will reduce the number of errors in your work. Moreover, knowing how to make publication-ready tables will give a good impression in your Econ 290 group (screenshots of output send a terrible signal!).

Before you begin, it is recommended that you read two vignettes: the kableExtra vignette and the modelsummary vignette. Each of these packages will be imperative to replicating the tables in the paper. If you do not read these vignettes, this homework will be near-impossible.

You will need two data sets for this homework which are available on the course website.

To Receive Credit

- Save the scripting file (i.e. your R program file) as assignment_6.R. Make sure your capitalization is correct as the autograder is case-sensitive.
- Make sure all changes to the original dataset are done within the R script.

Part 1: Coding Assignment

For this assignment, you will be turning in a PDF with your final tables to Gradescope in addition to your assignment_8.R script.

1. The goal of this question is to replicate Table 1 in (Anwar et. al 2022). This is a basic summary statistics table. To do this, you will need to use the data `zipcode_representation_analysis.dta`. You will need to use the following columns to create the table: `zip_pctwhite2000`, `zip_pctblack2000`, `zip_pctlaborforceold16`, `zip_medfaminc`, `zip_pctlesspov`, `zip_pct_hsgrad`, `zip_pctforeighborn` (yes that's spelled incorrectly), `zip_totalpop_2000`, and `zip_numover16`.

    a) Using the `haven::read_dta` function, read in the data and save the tibble with the name zipcodes.
    b) It's time to create the table. Read the modelsummary::datasummary webpage vignette. Using `modelsummary::datasummary` will be the easiest way to create this table. The basic strategy is to create a tibble in the format of what we see in the paper using `modelsummary::datasummary`. Then, pass this tibble into the `kableExtra::kbl` function and proceed to edit it with column headers and footnotes using other functions in the `kableExtra` package.
    c) Create a new column named `overrepresented` which is equal to "overrepresented" if the column `representation_pool` is greater than 1, and "underrepresented" otherwise.

d) Now, you will need to use `modelsummary::datasummary` to create the tibble you want. Make sure to set the `output` argument to `data.frame`. This will allow you to later pass in the tibble to the `kableExtra::kbl` function to edit using the excellent `kableExtra` functions. Your tibble should look similar to Table 1. Note that there are commas where the numbers get large. Since this may take you a long time to figure out how to get the correct format, here is the code to help:

```
mutate(across(-c(1), ~ifelse(row_number() == 6 |
                           row_number() == 10 |
                           row_number() == 11,
                         scales::comma(as.numeric(.)), .)))
```

e) Now that you have your desired tibble, you need to pass it through the `kableExtra::kbl` function. Note that you will want to use the `col.names` argument to change the column names, the `caption` argument to change the caption, set the `booktabs` argument to `TRUE` (this will make the table look like a publication table in PDF format), and set the `align` argument to `lccc` (stands for left-center-center-center-center) to align the columns correctly. Next, you will want to use the `kableExtra:add_header_above` function to create the column headers (you can use `\n` as a newline character within the string). Finally, you will need to use the `kableExtra::footnote` function to create the footnote. You will also want to set the argument `threeparttable=T` in the footnote function so the table does not bleed off the page. Your final table should look like Table 2.

Table 1: An example of how your tibble should appear in Question 1d. Note several columns are removed for presentation purposes.

	underrepresented / Mean	underrepresented / SD
Pool representation ratio	0.6	0.2
Percent White	26.4	20.6
Percent Black	26.3	25.0
Percent Hispanic	42.0	23.0
Percent labor force (over 16)	61.1	9.8
Zip code median family income	38,238.90	11,574.6
Percent income < poverty	17.7	8.3
Percent high school graduates	63.8	15.9
Percent foreign born	25.0	12.4
Total population in zip code	29,602.90	13,740.5
Population > age 16	20,984.60	9,612.0

Table 2: Census Characteristics of Zip Codes Undreprersented and Overrepresented in Harris County Jury Pool

	Underrepresented (<1) observations = 71 zip codes		Overrepresented (>1) observations =58 zip codes	
Zip code measure	Mean	SD	Mean	SD
Pool representation ratio	0.6	0.2	1.8	1.5
Percent White	26.4	20.6	64.8	19.4
Percent Black	26.3	25.0	10.1	13.0
Percent Hispanic	42.0	23.0	18.4	11.0
Percent labor force (over 16)	61.1	9.8	70.0	5.4
Zip code median family income	38,238.90	11,574.6	76,572.7	30,282.9
Percent income < poverty	17.7	8.3	5.6	3.9
Percent high school graduates	63.8	15.9	87.5	9.3
Percent foreign born	25.0	12.4	14.3	6.7
Total population in zip code	29,602.90	13,740.5	24,413.3	12,956.8
Population > age 16	20,984.60	9,612.0	17,893.1	9,107.5

Note:
Zip codes are classified as underrepresented and overrepresented based on their representation in the raw jury pool data relative to that in the census.

2. For this question, the goal is to recreate the regression output in Table 2 of (Anwar et. al 2022). To do this, you will need to estimate regressions using the `fixest` package, and then follow a similar workflow as before: use `modelsummary::modelsummary` to get a tibble of your desired table, and then pass this tibble into `kableExtra::kbl` to customize. You will need to use the `jury_analysis_extract.dta` data for this question. The necessary columns to use here are `max_length_prob`, `pool_sh_quartile1`, `pool_sh_quartile4`, `def_race`, and `life_sent_elig`.

   a) Using the `haven::read_dta` function, read in the data and save it as jury.
   b) You will need to estimate six regressions using `fixest::feols`. You will not be adjusting standard errors. You will be regressing each column in order. First, regress `max_length_prob` on `pool_sh_quartile1` and `pool_sh_quartile4`, while filtering `def_race == 1` (for White defendants only). Save this object as white_length_prob.
   c) Similar to part b, estimate the same equation, but filter to only `def_race == 2` (for Black defendants). Save this object as black_length_prob.
   d) Similar to parts a and b, estimate the same equation, but filter `def_race == 1` and `life_sent_elig == 1`. Save this object as white_length_prob_lse.
   e) Similar to part d, estimate the same equation but filter `def_race == 2` and `life_sent_elig == 1`. Save this object as black_length_prob_lse.
   f) Now estimate a new regression of `life_sent` on `pool_sh_quartile1` and `pool_sh_quartile4` while filtering `def_race == 1` and `life_sent_elig == 1`. Save this object as white_lse.
   g) Similar to part f, estimate the same regression but filter `def_race == 2` and `life_sent_elig == 1`. Save this object as black_lse.
   h) Now it is time to make the table. You will need to read the vignettes outlined above. In particular, the modelsummary vignette will be helpful. As mentioned earlier, you will follow a similar workflow as Question 1. Therefore, you will first create the tibble with the desired information using `modelsummary::modelsummary` function while setting the `output` argument to `data.frame`. The following code provides you with an outline for getting the correct tibble:

```
modelsummary(list(), ## fill in
         coef_map = , ## fill in
         gof_omit = , ## fill in
         gof_map = , ## fill in
         output = "data.frame",
         fmt = ) %>% ## fill in
  select(-part, -statistic) %>% ## drops unnecessary columns
  mutate(term = ifelse(row_number() == 2 |
                     row_number() == 4 |
                     row_number() == 6, " ", term)) %>% ## inserts blanks in rows
  add_row(term = "Full sample or life sent. eligible (LSE)?",
      `Model 1` = "Full",
      `Model 2` = "Full",
      `Model 3` = "LSE",
      `Model 4` = "LSE",
      `Model 5` = "LSE",
      `Model 6` = "LSE",
      .before = 7) %>% ## adds a row with following information
  add_row(term = "Defendant race",
      `Model 1` = "White",
      `Model 2` = "Black",
      `Model 3` = "White",
      `Model 4` = "Black",
      `Model 5` = "White",
      `Model 6` = "Black",
      .before = 8) ## adds a row with following information
```

i) Finally, using `kableExtra::kbl` recreate Table 3. You will need to change the `col.names`, `booktabs`, `align`, and `caption` arguments. Furthermore, you will need to use the `kableExtra::add_header_above` and `kableExtra::foonote` functions.

Table 3: Relationship between Jury Pool Representation and Trial Outcomes

	Sentence length (years with aquittals=0)				Convicted of a life setence	
	(1)	(2)	(3)	(4)	(5)	(6)
Proportion of pool in Q1	21.15	1.51	38.25	3.23	-0.03	0.02
	(52.52)	(25.75)	(68.85)	(29.13)	(0.79)	(0.31)
Proportion of pool in Q4	-14.20	59.45	35.91	92.06	-0.27	0.88
	(46.23)	(25.24)	(62.00)	(29.36)	(0.71)	(0.31)
Constant	28.03	15.62	23.06	12.64	0.27	-0.10
	(20.59)	(10.97)	(26.89)	(12.63)	(0.31)	(0.13)
Full sample or life sent. eligible (LSE)?	Full	Full	LSE	LSE	LSE	LSE
Defendant race	White	Black	White	Black	White	Black
Observations	166	551	104	436	104	436
R^2	0.0025	0.0118	0.0047	0.0261	0.0016	0.0216

Note:
All regressions use OLS and standard errors are in parentheses. Michael was able to do this.