Homework 4: Estimation

Econ 245

Overview

In this assignment we will be working with data downloaded from the US Department of Education's Campus Safety and Security (CSS) website. The data contain crimes from universities relating to liquor violations, weapon violations, and drug violations. We will not give detailed information about the column names, as this is often left to the researcher themselves. Hence, if you need clarification on a variable, we highly recommend you visit the website and find the answers yourself.

This homework assignment will mimic the early stages of a research project: finding data, tidying the data, and estimating some simple models. However, the purpose of this assignment is to introduce several packages that make estimation faster, and adjusting standard errors easier.

To Receive Credit

- Save the scripting file as assignment_4.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.

Grading on Coding Questions

Grading on the coding portion of the homework will come in two types of questions: *Public Questions* and *Private Questions*. Public Questions can be submitted as many times as you like to the autograder, and the autograder will give detailed feedback. On the other hand, Private Questions can be thought of as a mini quiz within the homework. While you still have as many times to upload your answer as you want, the autograder will not provide any feedback, and Professor Startz will not provide any guidance or assistance (but getting advice from classmates on Nectir or elsewhere is completely okay). Private Questions will be marked on the homework assignment.

Part 1: Cleaning

1. Load in the three datasets: `oncampusdiscipline111213.xls` (years 2011-2013), `oncampusdiscipline141516.xls` (years 2014-2016), and `oncampusdiscipline171819.xls` (years 2017-2019). The `readxl` package will be useful. Save each of these as tibbles named crime_11, crime_14, and crime_17 respectively.
2. We want to merge each of the data sets together and create a panel. To do this, you will need to (1) change the data from "wide" format to "long" format and (2) combine each of the data sets together. Here are a couple functions that may be useful: `tidyr::pivot_longer` and `tidyr::pivot_wider`. Save the final panel dataset as crime. See Table 1 for a preview of how the final data should look. A few things to keep in mind:

- Be certain to create a `year` column and make sure the `year` column is a `double`.

- Each crime (e.g. drug, weapon, and liquor) should have their own column.
- Do not add/remove rows from any of the data.
- The final tibble should have 100287 rows and 17 columns.
- Reading vignettes on unfamiliar packages and functions is very helpful.

Table 1: A preview of the final tibble in Question 2. Note that address, city, sector_cd, sector_desc, state, and zip have been omitted from this preview, however, they should still remain in the final tibble.

unitid_p	instnm	branch	men_total	women_total	total	year	weapon	drug	liquor	filter
100654001	Alabama A & M University	Main Campus	2268	2752	5020	2011	4	21	3	1
100654001	Alabama A & M University	Main Campus	2268	2752	5020	2012	1	47	6	1
100654001	Alabama A & M University	Main Campus	2268	2752	5020	2013	2	32	14	1
100654001	Alabama A & M University	Main Campus	2413	3446	5859	2014	2	109	29	1
100654001	Alabama A & M University	Main Campus	2413	3446	5859	2015	5	150	26	1
100654001	Alabama A & M University	Main Campus	2413	3446	5859	2016	2	92	11	1
100654001	Alabama A & M University	Main Campus	2364	3808	6172	2017	1	67	3	1
100654001	Alabama A & M University	Main Campus	2364	3808	6172	2018	4	159	6	1
100654001	Alabama A & M University	Main Campus	2364	3808	6172	2019	5	124	9	1
100663001	University of Alabama at Birmingham	Main Campus	7309	11259	18568	2011	2	43	127	1
100663001	University of Alabama at Birmingham	Main Campus	7309	11259	18568	2012	1	48	90	1
100663001	University of Alabama at Birmingham	Main Campus	7309	11259	18568	2013	0	44	136	1
100663001	University of Alabama at Birmingham	Main Campus	7590	11945	19535	2014	0	22	79	1
100663001	University of Alabama at Birmingham	Main Campus	7590	11945	19535	2015	0	23	115	1
100663001	University of Alabama at Birmingham	Main Campus	7590	11945	19535	2016	3	32	93	1
100663001	University of Alabama at Birmingham	Main Campus	8218	13862	22080	2017	2	62	118	1
100663001	University of Alabama at Birmingham	Main Campus	8218	13862	22080	2018	0	27	112	1
100663001	University of Alabama at Birmingham	Main Campus	8218	13862	22080	2019	3	26	93	1

3. We have created a random "treatment" dataset in which we randomly assigned a fake treatment year to each university. In practice, this could be thought of as a staggered adoption, where universities are treated at different points in time due to some policy change. Load in the `treatment.csv` file and save the tibble as treatment.

4. Using `dplyr::left_join`, merge the treatment data with the crime data. Save the result as crime_panel.

5. Create a binary variable named `treatment` in the crime_panel tibble that is a 1 if the `treatment_year` variable is less than or equal to the `year` column, and 0 otherwise. In effect, we are considering this pseudo-policy to be enacted in `treatment_year` and continue on for all years after.

6. Create a new tibble named crime_cs. This tibble will be a cross section of crime_panel. Specifically, it will be the collapsed means across all years. To do this, use the `group_by` function to group by `unitid_p` and then `summarize` across the following columns: `men_total`, `women_total`, `weapon`, `drug`, `liquor`, `treatment_year`. Finally, `ungroup` once this transformation is performed. While `ungroup` is unnecessary in this scenario, it is a good habit to ungroup your columns after completing a task. Refer to Table 2 for an example of the output.

Table 2: Example of the cross sectional data for Question 1.6.

unitid_p	men_total	women_total	total	weapon	drug	liquor	treatment_year
100654001	2348.333	3335.333	5683.667	2.8888889	89.00000	11.88889	2014
100663001	7705.667	12355.333	20061.000	1.2222222	36.33333	107.00000	2018
100663002	7705.667	12355.333	20061.000	0.0000000	0.00000	0.00000	2015
100690001	248.500	365.500	614.000	0.0000000	0.00000	0.00000	2016
100706001	4847.333	3763.333	8610.667	0.3333333	19.33333	46.55556	2019

Part 2: Base R Regressions, Standard Errors, and Multiple Hypothesis Testing

To demonstrate some basic regression tools in `R`, we will utilize the cross sectional data created in problem 1.6. Consider the following model:

$$Y_u = \beta_0 + \beta_1 Men\_Total_u + \beta_2 Liquor_u + \beta_3 Drug_u + \beta_4 Weaopon_u + \epsilon_u \tag{1}$$

where Y_u will be an indicator equal to 1 if university u has a treatment year greater than 2017, and 0 otherwise. Since we randomly allocated treatment years, this model really has no meaning–it is only used here as a demonstration for performing certain tasks.

1. Create a new column named `treatment_2017` that is an indicator equal to 1 if the `treatment_year` is greater than 2017. Update the crime_cs tibble to reflect this.

2. First, estimate Equation 1 using `base::lm`. Note that the standard errors are not corrected for heteroskedasticity. Save this estimation model as ols.

3. Next, estimate Equation 1 using `base::lm`., but correct for heteroskedasticity using "HC1" standard errors. "HC1" standard errors are equivalent to typing "robust" in STATA. To do this, you will need to use the `lmtest::coeftest` function. Use the `tidy::broom` function to save the point estimates, standard errors, test statistics, and p-values as a tibble named ols_hetero1. See Table 3 for an example of the resulting tibble.

Table 3: Example of resulting tibble.

term	estimate	std.error	statistic	p.value
(Intercept)	0.2849253	0.0044321	64.2873052	0.0000000
weapon	0.0081086	0.0097286	0.8334751	0.4045911
liquor	0.0002254	0.0000932	2.4194602	0.0155562
men_total	-0.0000005	0.0000007	-0.6408254	0.5216466
drug	-0.0004017	0.0003105	-1.2937288	0.1957804

4. Similarly to problem 2.3, estimate Equation 1, using `base::lm`. This time, correct for heteroskedasticity using "HC3" standard errors. As before, use the `tidy::broom` function to save the point estimates, standard errors, test statistics, and p-values as a tibble named ols_hetero3. Notice that HC3 standard errors are larger. While not a theorem, this is generally the case.

5. Perform the following F-test:
$$\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$$
using the "HC3" standard errors. To do this, you will need to use the `car::linearHypothesis` function. Use the `tidy::broom` to save your result in a tibble named f_test.

6. See how irritating fixing standard errors using `base::lm` is? Luckily, there are two fantastic solutions: the `modelsummary` package, and the `fixest` package. These packages allow for "on-the-fly" standard error adjustment (i.e., standard error adjustment is built-in). For this question, we will focus on the `modelsummary` package. First, the `modelsummary` package is most helpful for creating beautiful tables with a simple function call. Try running the following: `modelsummary(ols, stars = T)`. Notice how you can get a beautiful table with little effort. But now for the real magic: `modelsummary::modelsummary` has a `vcov` argument for standard error adjustment. In fact, you can instantly adjust your standard errors by passing in a vector of standard errors. In this problem, estimate the following model:
$$Liquor_u = \beta_0 + \beta_1 Men\_Total_u + \beta_2 Weapon_u + \beta_3 Drug_u + \epsilon_u$$
using `base::lm` and create a `modelsummary` table with the following standard error adjustments: "classical", "robust", "HC3", "bootstrap", "stata", "HC1", and "HC4". This should not be more than 1 or 2 lines of code. I highly recommend you read the quick-start website. Find a way to export this table so you can turn it in hard-copy. Also, notice that typing "robust" in `modelsummary::modelsummary` `vcov` argument is equivalent to the "HC3" standard errors, while "HC1" is equivalent to typing "stata".

Part 3: Fast Estimation with Fixest: Two-way Fixed Effects Model

The `fixest` package can be thought of as the computationally faster equivalent of STATA's popular `reghdfe` package. The benefit of the `fixest` package over using `base::lm` for regressions is strictly processing time. As fixed effects get higher in dimension, `base::lm` becomes too slow and may even freeze your computer. The equivalent of `base::lm` in `fixest` is the `fixest::feols` function. Read a short introduction to the `fixest` package here. Since this package will likely be the workhorse for the rest of your education, we recommend you read it thoroughly. The following problems will focus on using the `fixest` package in the common two-way fixed effects model.

1. Using `fixest::feols`, estimate the following model:
$$Y_{u,t} = \beta D_{u,t} + \phi \mathbb{X}_{u,t} + \gamma_u + \delta_t + \epsilon_{u,t}$$
where $Y_{u,t}$ is liquor offenses in university u in year t, $D_{u,t}$ is an indicator for treatment, γ_u are university fixed effects, δ_t are year fixed effects, $\mathbb{X}_{u,t}$ is a vector of covariates including `total` and a dummy for

whether a school is of the `sector_desc` "Public, 4-year or above" (call this `public`), and $\epsilon_{u,t}$ is the error term. Cluster the standard errors at the `unitid_p` level. Save this equation model as twfe_liquor.

2. Using the `broom::tidy` function, extract the tibble of point estimates, standard errors, p-values, and confidence intervals. Save this tibble as twfe_liquor_cluster. See Table 4 for an example of what your tibble should look like.

Table 4: TWFE clustered standard errors example output for question 3.2.

term	estimate	std.error	statistic	p.value	conf.low	conf.high
treatment	-0.0118420	0.1448950	-0.0817280	0.9348642	-0.2958309	0.2721470
total	0.0000103	0.0000689	0.1489742	0.8815761	-0.0001249	0.0001454
public	-0.2946182	0.4211447	-0.6995651	0.4842106	-1.1200467	0.5308104

3. Similar to problem 3.1, estimate the same model, but this time using the equivalent of STATA's robust standard errors (read the fixest vignette!).

4. Using the `broom::tidy` function, extract the tibble of point estimates, standard errors, p-values, and confidence intervals for the model estimated in 3.3. Save this tibble as twfe_liquor_robust. See Table 5 for an example of what your tibble should look like.

Table 5: TWFE robust standard errors example output for question 3.2.

term	estimate	std.error	statistic	p.value	conf.low	conf.high
treatment	-0.0118420	0.1322612	-0.0895348	0.9286571	-0.2710692	0.2473853
total	0.0000103	0.0000493	0.2082498	0.8350344	-0.0000864	0.0001069
public	-0.2946182	0.3346265	-0.8804389	0.3786242	-0.9504740	0.3612377

5. To showcase the power of `fixest::feols`, estimate the model in 3.1 but for `liquor`, `weapon`, and `drug` as the dependent variable. Do this *without* typing "feols" more than once (see here). Use the `modelsummary::modelsummary` function to make a table for these 3 regressions. Export it and find a way to turn it in hard-copy.