Problem Set 1: OLS Review

EC 421: Introduction to Econometrics

Due *before* midnight on Wednesday, 29 January 2020

DUE Your solutions to this problem set are due *before* midnight on Wednesday, 29 January 2020. Your files must be uploaded to Canvas.

IMPORTANT You must submit **two files**:

1. your typed responses/answers to the question (in a Word file or something similar)
2. the R script you used to generate your answers. Each student must turn in her/his own answers.

README! The data† in this problem set come from the paper "Are Emily and George More Employable than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination" by Bertrand and Mullainathan (published in the *American Economic Review* (AER) in 2004).†† In their (very influential) paper, Bertrand and Mullainathan use a clever experiment to study the effects of race and gender in labor-market decisions by sending fake résumés to job listings. To isolate the effect of race and gender on employment decisions, Bertrand and Mullainathan randomize whether the résumé lists a typically African-American name or a typically White name—in addition to randomizing the suggested gender of the name.

OBJECTIVE This problem set has three purposes: (1) reinforce the econometrics topics we reviewed in class; (2) build your R toolset; (3) start building your intuition about causality within econometrics.

Problem 1: Getting started

Start here. We're going to set up R and read in the data

1a. Open up RStudio, start a R new script (File ➡ New file ➡ R Script).
You must hand in this script as part of your assignment.

1b. Load the the `pacman` package. Now use its function `p_load` to load the `tidyverse` package, *i.e.*,

```
# Load the 'pacman' package
library(pacman)
# Load the packages 'tidyverse' and 'here'
p_load(tidyverse, here)
```

Note: If `pacman` is not already installed on your computer, then you need to install it, *i.e.*, `install.packages("pacman")`. If `tidyverse` is not already installed, then `p_load(tidyverse)` will automatically install it for you—which is why we're using `pacman`. I'm also loading the `here` package.

1c. Download the dataset from Canvas. Save it in a helpful location. Remember this location.

1d. Read the data into R. Name your dataset `job_df`.

What are the dimensions of the dataset (numbers of rows and columns)?

Note: Let each row in this dataset represent a different résumé sent to a job posting. The table on the last page explains each of the variables.

1e. What are the names of the first five variables? *Hint:* `names(job_df)`

1f. What are the first four *first names* in the dataset (`first_name` variable)? *Hint:* `head(job_df$var, 10)` gives the first 10 observations of variable `var` in dataset `job_df`.

† The data that we use in the problem set contain a subset of the variables from the original paper.

†† Here's a link to an article on Medium that discussed their paper.

Problem 2: Analysis

Reviewing the basic analysis tools of econometrics.

`Important note:` When you regress (in OLS) a binary indicator variable (like `i_callback`) on an explanatory variable, your coefficients tells you **how the explanatory variable affects the probability** that the indicatory variable equals one. So if we regress `i_callback` on `n_jobs`, the coefficient on `n_jobs` tells us how the probability of a callback changes with each additional job listed on the résumé.

`2a.` What percentage of the résumés received a callback (`i_callback`)?

Hint: The mean of a binary indicator variable (*i.e.*, `mean(binary_variable)`) gives the percentage of times the variable equals one. Also: See the **Important note** above.

`2b.` Calculate the percentage of callbacks (*i.e.*, the mean of `i_callback`) for each sex (`sex`). Does it appear as though employers considered an applicant's sex when making callbacks? Explain.

Hint: `filter(job_df, sex == "f")` will select all observations (from the dataset `job_df`) where the variable `sex` takes the value `"f"` (female). Similarly `filter(job_df, sex == "f")$i_callback` will give you the values of `i_callback` for observations whose value of `sex` is `"f"`.

`2c.` What is the difference in the groups' mean callback rates (female *vs.* male)?

`2d.` Based upon the difference in percentages that we observe in **`2b.`** can we conclude that employers consider sex in hiring decisions?

`2e.` Without running a regression, conduct a statistical test for the difference in the two groups' average callback rates (*i.e.*, test that the proportion of callbacks is equal for the two groups).

Hint: Back to your statistics class—difference in proportions (a Z test) or means (a t test). It doesn't really matter which one you choose.

`2f.` Now regress `i_callback` (whether the résumé generated a callback) on `i_female` (whether the résumé's name implied a female applicant). Report the coefficient on `i_female`. Does it match the difference that you found in **2c**?

`2g.` Conduct a t test for the coefficient on `i_female` in the regression above in **2f**. Write our your hypotheses (both H_0 and H_A), the test statistic, the result of your test (*i.e.*, reject or fail to reject H_0), and your conclusion.

`2h.` Now regress `i_callback` (whether the résumé generated a callback) on `i_female`, `n_expr` (years of experience), and the interaction between `i_female` and `n_expr`. Interpret the estimates for all of the coefficients (both the meaning of the coefficients and whether they are statistically significant).

Hint: In R, `lm(y ~ x1 + x2 + x1:x2, data = job_df)` regresses `y` on `x1`, `x2`, and the interaction between `x1` and `x2` (all from the dataset `job_df`).

`2i.` Now create a new dataset that is the subset of job applications from names that are tyically interpreted as from African-American applicants. Repeat all of **2h** on this new dataset.

Hint: To take the subset of the dataset `job_df` that use names often interpreted as African American:

```
# Subset the job-application data to names interpreted as African American
aa_df = filter(job_df, i_black == 1)
```

Problem 3: Thinking about causality

Now for the big picture.

This project by Bertrand and Mullainathan took a decent amount of time and effort—finding job listings, generating fake résumés, responding to the listings, *etc.* It probably would have been much quicker/cheaper/easier to just go out and get data from job applicants—whether they received callbacks and their sexes. So why didn't they take the easier, cheaper, and quicker route?

To answer this question, we are going to consider the model

$$\text{Callback}_i = \beta_0 + \beta_1 \text{Female}_i + u_i \tag{3.0}$$

and think about omitted-variable bias.

3a. If we go out, collect data on job applicants, and estimate the model in (3.0) using OLS, *i.e.*,

$$\text{Callback}_i = \hat{\beta}_0 + \hat{\beta}_1 \text{Female}_i + e_i \tag{3.1}$$

we should be concerned about omitted-variable bias. Explain why this is the case **and** provide at least one example of an omitted variable that could bias our estimates in (3.1).

3b. To avoid this potential bias, Bertrand and Mullainathan ran an experiment in which they randomized applicants' names on the résumés—thus randomly assigning the (implied) sex of the job applicants. How does this randomization help Bertrand and Mullainathan avoid omitted variables bias?

In other words, why are we less concerned about omitted variable bias in the following estimated model

$$\text{Callback}_i = \hat{\beta}_0 + \hat{\beta}_1 (\text{Randomized Female})_i + w_i \tag{3.2}$$

while we were concerned about bias in (3.1)?

Description of variables and names

Variable	Description
`i_callback`	Binary variable (0,1) for whether the resume received a callback.
`n_jobs`	Number of previous jobs listed on the application.
`n_expr`	Number of years of experience listed on the application.
`i_military`	Binary variable for whether the application included military status.
`i_computer`	Binary variable for whether the application included computer skills.
`first_name`	The first name listed on the application.
`sex`	The implied sex of the first name on the application (`'f'` or `'m'`).
`i_female`	Binary indicator for whether the implied sex was female.
`i_male`	Binary indicator for whether the implied sex was male.
`race`	The implied race of the first name on the application (`'b'` or `'w'`).
`i_black`	Binary indicator for whether the implied race was African American.
`i_white`	Binary indicator for whether the implied race was White.
`i_secretary`	Binary indicator for whether the job was for secretarial work.

In general, I've tried to stick with a naming convention. Variables that begin with `i_` denote binary indicatory variables (taking on the value of `0` or `1`). Variables that begin with `n_` are numeric variables.