Problem Set 1: OLS Review

EC 421: Introduction to Econometrics

Solutions

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.

Answer Done. Free points (as long as you handed in your R script).

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.

Answer More free points.

† 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.

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.

Answer Setup

```
# Read in the data
job_df = read_csv(file = here("001-data.csv"))
# Dimensions of the dataset with dim
dim(job_df)
```

```
## [1] 4626   13
```

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

Answer A few options

```
# Using 'head'
names(job_df) %>% head(5)
```

```
## [1] "i_callback" "n_jobs"     "n_expr"     "i_military" "i_computer"
```

```
# Using indexes
job_df[, 1:5] %>% names()
```

```
## [1] "i_callback" "n_jobs"     "n_expr"     "i_military" "i_computer"
```

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`.

Answer Three ways to do it:

```
# Using head
head(job_df$first_name, 4)
```

```
## [1] "Jamal"   "Tanisha" "Carrie"  "Anne"
```

```
# Using indexes
job_df[1:4,"first_name"]
```

```
## # A tibble: 4 x 1
##   first_name
##   <chr>
## 1 Jamal
## 2 Tanisha
## 3 Carrie
## 4 Anne
```

```
# Using head and select
job_df %>% head(4) %>% select(first_name)
```

```
## # A tibble: 4 x 1
##   first_name
##   <chr>
## 1 Jamal
## 2 Tanisha
## 3 Carrie
## 4 Anne
```

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.

Answer

```
# The mean of callback
mean(job_df$i_callback)
```

```
## [1] 0.08084738
```

Thus, approximately 8.1 percent of résumés received callbacks.

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"`.

Answer

One method:

```
# Percentage for Black
filter(job_df, sex == "f")$i_callback %>% mean()
```

```
## [1] 0.08345041
```

```
# Percentage for White
filter(job_df, sex == "m")$i_callback %>% mean()
```

```
## [1] 0.07216495
```

Alternative method:

```
job_df %>% group_by(sex) %>% summarize(mean(i_callback))
```

```
## # A tibble: 2 x 2
##   sex   `mean(i_callback)`
##   <chr>              <dbl>
## 1 f                 0.0835
## 2 m                 0.0722
```

Approximately 8.3 percent of résumés with implicitly female names received callbacks, while 7.2 percent of résumés with male-sounding names received callbacks.

This difference is consistent with gender discrimination (employers considering sex in hiring), but we do not know if this difference is statistically significant.

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

Answer

```
# Percentage for Black
mean_f <- filter(job_df, sex == "f")$i_callback %>% mean()
# Percentage for White
mean_m <- filter(job_df, sex == "m")$i_callback %>% mean()
# Difference:
mean_f - mean_m
```

```
## [1] 0.01128546
```

Female-sounding names had a 1.1-percentage point higher callback rate.

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

Answer No. We have shown a difference in callbacks for women and men, but we do not know if this difference is statistically meaningful (significant).

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.

Answer

```
# Percentage for everyone
mean_all ← job_df$i_callback %>% mean()
# Number: Black
n_f ← filter(job_df, sex == "f") %>% nrow()
# Number: White
n_m ← filter(job_df, sex == "m") %>% nrow()
# The Z statistic
z_stat ← (mean_f - mean_m) / sqrt(mean_all * (1 - mean_all) * (1/n_f + 1/n_m))
# The p value
2 * pnorm(abs(z_stat), lower.tail = F)
```

```
## [1] 0.2355672
```

For H_0: equal callback rates *vs.* H_A: callback rates were not equal, we **fail to reject the null hypothesis** at the 5-percent level, since our p-value is approximately 0.236.

Note: I opted for a two-sided Z test here, since we are testing unequal proportions. A t test (testing two means) would be fine, though maybe not technically correct. You could also test a one-side hypothesis if your null was that discrimination pointed in a specific direction (which it likely was).

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**?

Answer

Simple linear regression...

```
# Regression
reg_2f ← lm(i_callback ~ i_female, data = job_df)
# Results
reg_2f %>% tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   0.0722   0.00835      8.65 7.20e-18
## 2 i_female      0.0113   0.00952      1.19 2.36e- 1
```

The coefficient on `i_female` does indeed match the difference in callback rate across female- and male-sounding first names.

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.

Answer

H_0: $\beta_1 = 0$ and H_A: $\beta_1 \neq 0$, where β_1 is the coefficient for the effect of sex on the probability a résumé received a callback.

The point estimate for this coefficient is 0.011. Its associated t statistic is 1.19, which has a p-value of approximately 0.236.

We fail to reject the null hypothesis at the 5-percent level. We conclude that we cannot reject $\beta_1 = 0$, meaning we cannot reject "no difference between women and men".

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`).

Answer

```
# Regression with interaction
reg_2h ← lm(i_callback ~ i_female + n_expr + i_female:n_expr, data = job_df)
# Results
reg_2h %>% tidy()
```

```
## # A tibble: 4 x 5
##   term              estimate std.error statistic     p.value
##   <chr>                <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)      0.0769     0.0147      5.23  0.000000177
## 2 i_female        -0.0299     0.0170     -1.76  0.0787
## 3 n_expr          -0.000635   0.00161    -0.394 0.694
## 4 i_female:n_expr  0.00522    0.00185     2.82  0.00483
```

The coefficient on `i_female` has changed signs, relative to the previous estimate—suggesting that a female-sounding name reduced the probability of a callback by approximately 3 percentage points **for low-experience individuals**. This effect is not significant at the 5-percent level but it is significant at the 10-percent level.

The coefficient on the number of years of experience (`n_expr`) is close to zero and does not differ significantly from zero. This coefficient estimates the effect of experience on the probability of receiving a callback **for men**.

The coefficient on the interaction between the female indicator variable (`i_female`) and the experience variable (`n_expr`) tests whether the effect of experience on the callback rate differed between female and male résumés. The point estimate is statistically significant and suggests that for each additional year of experience, a woman's probability of receiving a callback increases by 0.5 percent.

In other words, we detect a significant difference in the effect of experience between women and men.

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)
```

Answer

```
# Take subset
aa_df = filter(job_df, i_black == 1)
# Regression with interaction
reg_2h ← lm(i_callback ~ i_female + n_expr + i_female:n_expr, data = aa_df)
# Results
reg_2h %>% tidy()
```

```
## # A tibble: 4 x 5
##   term            estimate std.error statistic   p.value
##   <chr>              <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)      0.0776    0.0191       4.05 0.0000522
## 2 i_female        -0.0482    0.0220      -2.19 0.0284
## 3 n_expr          -0.00266   0.00213     -1.25 0.212
## 4 i_female:n_expr  0.00733   0.00241      3.04 0.00243
```

The coefficient on `i_female` suggests that African American women with **wiht low-experience individuals** are almost 5-percentage points less likely to receive a callback, relative to African American men. This effect is statistically significant at the 5-percent level.

The coefficient on the number of years of experience (`n_expr`) is close to zero and does not differ significantly from zero. This coefficient estimates the effect of experience on the probability of receiving a callback for African American men.

The coefficient on the interaction between the female indicator variable (`i_female`) and the experience variable (`n_expr`) tests whether the effect of experience on the callback rate differed between female and male résumés (for names typically interpreted as African-American). The point estimate is statistically significant and suggests that for each additional year of experience, an African American woman's probability of receiving a callback increases by 0.7 percent.

In other words, we detect a significant difference in the effect of experience between African American women and men.

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).

Answer

We should be concerned about omitted-variable bias because there likely many variables that affect whether individuals received callback **and** are correlated with sex. If this is the case, then our estimates for β_1 will be biased.

Several possibilities: social connections, education, college major

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)?

Answer

Because Bertrand and Mullainathan randomize the implied sex (and race) on each (fake) résumé (along with the other variables), the sex (and race) variable in their study are uncorrelated with the other variables that affect callbacks. Thus, even if we omit 'important' variables (for predicting callback), they are uncorrelated with our variable of interest (sex), and thus they will not cause omitted-variable bias.

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.