Capstone Exercise

Reproducing Edelman, Luca, and Svirsky (AEJ Applied 2017)

February 28, 2018

Introduction

For the capstone exercise, you will be applying what you've learned over the past several modules to conduct a replication of a recent journal article. Specifically, you will be reproducing the main results from "Racial Discrimination in the Sharing Economy: Evidence from a Field Experiment," by Benjamin Edelman, Michael Luca, and Dan Svirsky and published in the *American Economic Journal: Applied Economics* last April.

In this article, Edelman, Luca, and Svirsky conduct an experiment wherein they apply for Airbnb apartments using guest names that have distinctively white or African American names. Using this experiment, they then investigate racial discrimination based on a number of host and location characteristics.

Data Preparation

Preliminaries

- To begin the replication, download the paper and data from here.
- Create a new Project for the exercise in RStudio with version-control (preferably connected to a GitHub repository).
  - You should put the data from Edelman, Luca, and Svirsky in something like a *data* subfolder in the project directory.
- Within the project, create a new RMarkdown document.
- As you go through each step, either add in the instructions found here or add your comments explaining what you are doing.
- Commit (and Push) your work every 15 minutes or so.

Importing Data

- If you are not working inside of RMarkdown / RStudio Project, set the work directory.
- Import the data set "main_data.csv"

Recode missing values and convert to a tibble

- Using a map or for-loop, change text to lower case using the tolower() function.
- Using a map or for-loop, recode the following values to missing throughout the dataset: "\\N", "Null", "-1".
- Then convert the dataframe to a tibble.

Rename the variables

Import the "datanames" csv file and assign it as the column names of the main dataset. - You may want to change the format of *datanames* to matrix after you import it.

Convert columns to correct class

Using for-loops, change the class of columns in the main dataset as follows:

- **Covert to numeric columns:** 3-6, 10-14, 19-21, 39-46, and 49.
- **Convert to factor columns:** 1, 7-9, 15-17, 23-33, 36-38, and 47.

Set reference groups

- For the variable *guest_race*, set the reference group to the value "white".
- For the variable *guest_gender*, set the reference group to the value "male".

Create a guest_name by city variable to identify individual guests

For clustering of standard errors in the regression analysis, create a variable *namebycity* that concatenates the values from *guest_first_name* and *city*.

- Use the paste() function to do this.

Import and merge survey results

- Import the file "name_survey_results.xlsx"
- Again apply tolower() to the *guest_first_name* variable.
- Merge in additional variables from this dataset for observations from the main dataset, using the key *guest_first_name*.

Change the values of *guest_race_continuous*

Change the value of *guest_race_continuous* by subtracting one from it's current value, so that it's range is 0 to 1 instead of 1 to 2.

Make host race and sex variables

Create the following indicator variables:

- *host_race_black* equal to 1 if the host's race is "black" according to the *host_race* variable.
- *host_race_white* equal to 1 if the host's race is "white" according to the *host_race* variable.
- *host_male* equal to 1 if the host's race is "m" according to the *host_gender* variable.

Make a categorical host age variable

Make a categorical host age variable, *host_age_cat*, with values as follows:

- Value of 0 if *host_age* is equal to any of "young","young/uu", "uu/young", "young/na", or "na/young".
- Value of 1 if *host_age* is equal to any of "middle/young", or "young/middle".
- Value of 2 if *host_age* is equal to any of "middle","middle/uu","uu/middle","middle/na", or "na/middle".

- Value of 3 if *host_age* is equal to any of "middle/old" or "old/middle".
- Value of 4 if *host_age* is equal to any of "old/middle","old","old/uu","uu/old","old/na", or "na/old".

Make binary variables for other host characteristics:

Create the following binary variables:

- *ten_reviews* indicating whether or not *number_of_reviews* is greater than or equal to 10.
- *five_star_property* indicating whether or not *apt_rating* is equal to five.
- *multiple_listings* indicating whether or not *number_of_listings* is greater than 1.
- *shared_property* indicating whether *property_setup* is ***either*** "private room" or "shared room".
- *shared_bathroom* for the conditions that *bathrooms* is less than 1.5 and the property is shared according to your variable above.
- *has_cleaning_fee* indicating whether *cleaning_fee* is not missing.
- *strict_cancellation* indicating whether *cancellation_policy* is equal to "strict".
- *young* indicating whether *host_age_cat* is equal to zero.
- *middle* indicating whether *host_age_cat* is equal to one or two.
- *old* indicating whether *host_age_cat* is equal to three or four.

Crate a simplified host response variable

Create a new variable *simplified_response* that has the following values:

- "No Response" if *host_response* is equal to NA.
- "Yes" if *host_response* is equal to 1.
- "No" if *host_response* is equal to 0.
- "Conditional Yes" if *host_response* is equal to 4, 5,6, 7 or 8.
- "Conditional No" if *host_response* is equal to 2,3, 9, 10, or 11.

Create a binary host response variable

Create a new variable *yes* that that is equal to:

- 1 if if *host_response* is equal to 1,4, or 6.
- 0 if if *host_response* is equal to 0,2,3,7,8,9,10,11,12, or if *host_response* is missing.

Drop observations in Tampa and Atlanta

The experiment could not be completed in Tampa or Atlanta, so drop the observations where *city* is equal to either of these two values.

Merge in data on past guests

- Import the dataset "hosts.dta" and add in variables from this dataset to the observations from the main dataset using the key *host_id*.

Main Analysis

Reproduce estimates from Table 2: The Impact of Race on Likelihood of Acceptance

- Perform separate regressions corresponding to each of the columns of Table 2 and save the regressions objects
  - "Guest is African-American" is captured by the *host_race* variable.
- For the first regression:
  - Obtain the cluster-robust standard errors and test-statistics using the function cluster.vcov from the multiwayvcov package.
  - Cluster on *namebycity*
  - The syntax of cluster.vcov is:

```
cluster_obj <- cluster.vcov(reg_object, cluster=data$clustervar)
```

- – Print a *tidy-ed* version of the estimates from each regression using the cluster-robust standard errors.
- After the first regression:
  - create a function that takes a regression objects, obtains the clustered-standard errors, performs t-tests using the clustered standard errors, and then saves the tidy-ed version of those estimates.
  - Use the function to get the estimates from columns 2 and 3.

Reproduce Figure 2: Host Responses by Race

Create a grouped bar plot of host responses by Race, as in Figure 2 of Edelman, Luca, and Svirsky.

- First create a summary data frame that counts the number of observations grouped by *guest_race* and *simplified_response*.
- Then create a bar plot that is ***grouped*** by specifying *fill* color according to *guest_race* inside of the base aesthetic, with the argument `position="dodge"` inside of **geom_bar** (otherwise you'd get a stacked bar plot).

[Bonus!] Table 5. Are Effects Driven by Host Characteristics?

Reproduce columns 4 and 5 from Table 5 (again using your helper function for cluster-robust test statistics).

- "Host has 1+ reviews from an African American guest" is represented by the *any_black* variable.

Table 6: Are Effects Driven by Location Characteristics?

Data Preperation

Make a variable that lists the number of properties within the census tract

- Using the *group_by* and *summarize* function, first create a variable that tallies the number of Airbnb listings in each tract using the summary condition:

```
tract_listings = sum(latitude > 0)
```

- Use a join operation to add this data to the main dataset.

Generate Price Variables

Generate *price_geq_median* indicating whether or not the apartment price is greater than equal to the median of apartment prices, according to *price*.

Generate racial composition of Census tract variables

Create the racial composition variables as follows:

- *white_proportion* equal to the variable *whites* divided by *population*.
- *black_proportion* equal to the variable *blacks* divided by *population*.
- *asian_proportion* equal to the variable *asians* divided by *population*.
- *hispanic_proportion* equal to the variable *hispanics* divided by *population*.

Analysis

Reproduce columns 1 through 3 of Table 6.