FLS 6415: Replication 8 - Matching

May 2020

To be submitted (code + answers) by midnight, Thursday 14th May.

First read the paper by Boas and Hidalgo (2011) on the course website. For this replication we will focus on the *second half* of their paper, not the initial RDD but the matching analysis of how possession of a radio licence affects the mayor's vote share in the next election.

The replication data is in the file *Boas_Hidalgo.csv*. A list of the most important variables is also provided below.

Variable	Description
pctVV	The councillor's vote share in the 2004 elections
treat	Whether a councillor that applied for a media licence received approval before the 2004 election
male	Councillor is male
log.valid.votes	Log of the size of the electorate (proxied by valid votes)

1. What is treatment? What is control? What is the outcome?

2. Why do Boas and Hidalgo not use an experiment or natural experiment to estimate the effect of possessing a radio licence?

3. Conduct and interpret a basic linear regression of the outcome on treatment with no controls.

4. One potential confounding variable is gender (this could affect the chances of an application being approved if there is bias in the Ministry, and the candidate's vote share if there is bias among voters). Is there balance across control and treatment groups on the `male` variable?

5. One way of controlling for gender is to add it as a control variable to your regression in Q3. Interpret the result.

6. An alternative approach is to use matching. Let's try to do one-to-one exact matching on gender *manually*. There are 311 treated units but 1144 control units in your data, so one-to-one matching means *throwing away* 833 control units.
(a) Split your data into four different datasets: treated males, treated females, control males and control females;
(b) How many treated males do you have? Reduce your dataset of control males so you have only the same number as the number of treated males - since they are exactly matched on gender it doesn't matter which you pick so choose which ones to keep/drop randomly;
(c) Do the same for control females - reduce the number of control females to the same as the number of treated females;
(d) Join your four datasets back together to make one dataset (this will be smaller than the original dataset as we threw some data away);
(e) Check for balance in gender on the new dataset - it should be perfectly balanced, right?

7. Using the matched dataset from Q6, conduct two analyses of the difference in outcomes between treated and control groups. One using a difference-in-means t-test and one using a simple linear regression. Interpret the results.

8. To match on continuous or multiple variables it's easier to use `matchit`.

(a) Return to your original full dataset and, using nearest neighbour matching, match only on the size of the electorate (*log.valid.votes*).

(b) How many units are matched? Why this number?

(c) Conduct a simple balance t-test on the size of the electorate for the full dataset and for your matched dataset (you can recover it with `match.data(output_of_matchit)`). How does balance change after matching?

9. Let's see which units were dropped by our matching method in Q8. For the full (unmatched) dataset, create a graph of the size of the electorate against the outcome variable. Colour the points according to treatment status. Make this layer semi-transparent (adjust the 'alpha' of your graph in R) if you can so we can see all the points. Finally, add another layer to your graph showing the same variables for the *matched* data but with a different shape so we can distinguish them. What does this graph tell you about which units were matched?

10. Using the matched dataset from Q8, conduct two analyses of the difference in outcomes between treated and control groups. One using a difference-in-means t-test and one using a simple linear regression. Interpret the results.

11. Now let's include all of the matching variables that Boas and Hidalgo use, and use nearest neighbour matching in `matchit` to construct a matched dataset. Use the list of matching variables provided below to conduct nearest neighbour matching.

"occBlue.collar", "occEducation", "occGovernment", "occMedia", "occNone", "occOther", "occPolitician", "occWhite.collar", "lat", "long", "ran.prior", "incumbent", "log.valid.votes", "party.prior.pctVV", "prior.pctVV", "elec.year", "match.partyPCB", "match.partyPC.do.B", "match.partyPDT", "match.partyPFL", "match.partyPL", "match.partyPMDB", "match.partyPMN", "match.partyPP", "match.partyPPS", "match.partyPSB", "match.partyPSC", "match.partyPSDB", "match.partyPSDC", "match.partyPSL", "match.partyPT", "match.partyPTB", "match.partyPV", "uf.rs", "uf.sp", "yob", "eduMore.than.Primary..Less.than.Superior", "eduSome.Superior.or.More", "log.total.assets", "pt_pres_1998", "psdb_2000", "hdi_2000", "income_2000", "log.num.apps"

12. Using your matched dataset from Q11, conduct a simple linear regression of the outcome on treatment. Interpret the results and compare them to the result in the first column of Table 4 in Boas and Hidalgo (2011) (it probably won't be the same, see the next questions).

13. With lots of variables it's impossible to get perfect balance on all variables, there are just too many dimensions and too few units. One option to control for 'residual confounding' is to include the matching variables as control variables in our analysis regression. How does this change your estimated treatment effect from Q12?

14. One risk with nearest-neighbour matching is that the control unit can still be far away from the treated unit if there are no good matches. Re-run the matching process from Q11 but with a caliper of 0.01 standard deviations, and then re-run the regression from Q12 (no controls). How does the number of units and the result change?

15. Another problem with nearest neighbour matching is that it is 'greedy' - the first matches might make it harder to match well later. Boas and Hidalgo use genetic matching, which is a complex automated process to try and get the best 'overall' matches for the full dataset. Run genetic matching process with the same variables and then run your regression (with no controls) again. *Note:* Genetic matching might take 10-20 minutes.