FLS 6415: Class 10 Homework

November, 2017

Remember to answer all the questions in R markdown and produce a PDF. Email your completed homework (R markdown file and PDF) to jonnyphillips@gmail.com by midnight the night before class. Remember to refer to the example code from this week and the last couple of weeks for coding guidance.

Load the Boas and Hidalgo dataset `combined_data.RDS` using `readRDS()`. There are many tools and settings that can be adjusted when using matching - we will try and repicate the Boas and Hidalgo results using more user-friendly tools than in their own code, so we will not recover identical values.

1. The treatment variable in Boas and Hidalgo is whether a councillor that applied for a media licence received approval before the 2004 election (`treat`). The outcome variable is the councillor's vote share in the 2004 elections (`pctVV`). Conduct and interpret a basic linear regression of the outcome on treatment with no controls.

2. 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?

3. One 'control for confouding' approach is to use a regression. Conduct the regression of the oucome (`pctVV`) on treatment (`treat`), controlling for `male` and compare the results to your answer to Q1.

4. An alternative approach to overcome confounding is matching: removing control units from the dataset until we have balance on gender. Let's try to implement one-to-one exact matching MANUALLY to illustrate the process:
a. Create separate datasets for treated and control units (`d_treat` and `d_control`).
b. Create a new empty column in the treated dataset (eg. call it matched_control_unit_pctVV) where we are going to save the outcome for the matched control unit. b. Create a `for` loop so we can focus on each *treated* unit (each row of your treated dataset `d_treat`) separately.
c. Within your for loop, for one treated unit, identify its gender, and filter the control units dataset (`d_control`) to only include control units with the same gender as the current treated unit.
d. There are lots of control units with the same gender as our treated unit, so we can pick any of them as our matched unit. *Hint:* Use `sample_n(1)` to randomly pick one row of your control unit data.
e. Now, for this matched control unit, select only the outcome variable and save that value to the new column you made in the *treated* dataset (eg. `d_treat[i,"matched_control_unit_pctVV"]`). Make sure you save it only to the row of the current treated unit you are analysing in your `for` loop.

5. With this matched dataset, we now have balance on gender by definition. To estimate the treatment effect using a difference-in-means calculate the difference in outcomes between your treated and matched control units in the treated units dataset (i.e. between the `pctVV` and `matched_control_unit_pctVV` variables). Then average these differences to get an average treatment effect and interpret the result. Also conduct a t-test to get a p-value.

6. We can also use post-matching regression to do the analysis stage. To do so, we need to reorganize our dataset to include rows for both the treated and matched control units, with their corresponding outcomes.
a. Select only the 'pctVV' and 'matched_control_unit_pctVV' columns.
b. Transform the data from wide to long format using `gather`. Your dataset should now have 622 rows. c. Use `mutate` with `ifelse` to recode the treatment column values from 'pctVV' and 'matched_control_unit_pctVV' to 1 and 0 respectively. d. Conduct the simple linear regression of the outcome on treatment for this dataset.
e. Compare the results to your answer in Q5.

7. To match on continuous variables, we can't match 'exactly' so we do 'nearest' neighbour matching. Another potential confounder is the size of the electorate (`log.valid.votes`). Use

`matchit` to construct a datset of treated and control units matched on (only) the size of the electorate with nearest neighbour matching. In your matched data, how many treated and control units are there? *Hint:* Access the units summary using `output$nn`, where `output` is the result of `matchit`.

8. Check that this matching has improved balance by comparing the average difference in `log.valid.votes` in the original dataset and in the matched dataset. *Hint:* Remember you can use `match.data(output_of_matchit)` to get the matched dataset.

9. To see which units have been matched, plot two layers on a graph of the confounder `log.valid.votes` against the outcome `pctVV`. One layer should be the full dataset - make this a specific size, shape and alpha (transparency). The second layer should be the matched dataset from Q8 - make this a different size, shape and alpha so that the matched units stand out. Group and colour each layer by treatment status.

10. Use a simple least squares regression to estimate the effect of the treatment ('treat') on the outcome ('pctVV') in the matched dataset. Interpret the result.

11. Conduct the same regression as in Q8 but add a control for `log.valid.votes`, the variable we already matched on. Does this change the estimate of the treatment effect compared to your answer in Q8? Why/why not?

12. Now lets 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 for matching.

```
covars <- c("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")

covars_formula <- paste0(covars, collapse = " + ")
```

13. For your matched dataset from Q12, report a before-matching and after-matching balance table similar to the first four columns of Boas and Hidalgo's Table 3. Their first two columns do not report the difference in means between treated and control, but the *standardized* difference in means. This requires calculating the difference in means and then dividing it by the standard deviation of the treated group:

a. In your matched dataset, calculate the standard deviation of the treated variable for each of the 44 covariates. *Hint:* `summarise_at` is useful to calculate the same summary statistic for the list of covariates provided above.

b. In your matched dataset, calculate the difference in means between treated and control group for each covariate.

c. Divide this difference in means by the standard deviation of the treated group for each variable as calculated in (a.), and multiply by 100.

d. For the third and fourth columns of Boas and Hidalgo, calulate the p-value of the difference between treated and control groups. *Hint:* Try something like `d %>% summarise_at(covars,funs(t.test(.[treat==0], .[treat==1])$p.value))`.

e. Now combine the columns for standardized differences in means and p-values for the full and matched datasets into a single table. *Hint:* Use `gather()` to turn wide rows into long columns, and `left_join` to combine the columns into a single table.

14. Implement the regression of the outcome on treatment (excluding controls) on the matched dataset and compare your answer to that of Boas and Hidalgo in Table 4.

15. One risk with the nearest-neighbour method is that the control unit can still be far away from the treated unit if there are no good matches, so imbalance can still be a problem. We can put a 'caliper' on the matching method so that it only includes 'close' matches within a specific number of standard deviations. Re-run the matching process from Q12 but with a caliper of 0.01 standard deviations, and then conduct the regression from Q14. Note how the number of units changes and interpret if the result is any different.

16. Boas and Hidalgo use genetic matching, which is a complex automated process. We can implement genetic matching easily by loading the package `rgenoud` and choosing `method="genetic"` in the matching process in `matchit`. Then run the same regression again and compare the results to those in Q14 and in Boas and Hidalgo. *Hint:* Genetic matching might take 10-20 minutes, so in your R chunk options set `cache=TRUE` so that it doesn't re-run every time you knit to PDF. Also set results='hide' in the chunk options to prevent lots of messages printing to your PDF and put your table in a separate chunk with results='asis'.

17. As with regression, these estimates only have a causal interpretation if matching is successful at producing balance on ALL of the confounding covariates. Provide an example of a confounding variable which would bias Boas and Hidalgo's estimates despite their matching procedure.