Week 3: Selection on Observables

Tanisha Mohapatra & Fernando Sanchez Monforte

HT 2025

1 Introduction

This week, we will look at one of the available tools for inference when the treatment of interest is not randomized – **matching**. In particular, we explore how relying on the **selection on observables** can boost (or not) our confidence in the effects of treatment that we uncover in such scenarios.

Although this approach is far from the 'gold standard' of RCTs, often, the scenarios in which we can adjust for treatment probability based on observable characteristics can have some advantages. These scenarios:

- Don't require researchers to conduct their own data collection (which might be costly, impractical, unethical or a combination thereof);
- Allow for a larger sample sizes;
- Can be used to study events that already happened, but remain of interest.

In this lab session, we will implement to concepts from this week's lecture[1]. Specifically:

1. Discuss the pitfalls of ignoring the observational nature of treatment assignment
2. Go over matching techniques and options
3. Compare treatment and control groups before and after matching
4. Analyze ATT before and after matching
5. Explore ATT's robustness to different matching choices

Before starting this seminar

1. Create a folder called `Lab3`.
2. Download the data and the empty RMarkdown file from Canvas
3. Save both in our `Lab3` folder.
4. Open the RMarkdown file.
5. Set your working directory using the `setwd()` function or by clicking on "More".
6. Prepare your R environment running the code below.

[1] This session builds on materials kindly shared by Tom O'Grady, Jack Blumenau, and Julia de Romémont

```r
# Let's get started!

## Global options:
knitr::opts_chunk$set(cache = TRUE, warning = FALSE, message = FALSE)

## We use pacman to install and load all packages we need:
# install.packages("pacman") # Uncomment if you don't have pacman installed yet!
library(pacman)
pacman::p_load(tidyverse,  # Cleaning data, plotting
               stargazer,  # Neat regression tables
               kableExtra, # Yet another package for tables
               tableone, # This one is for summary/balance tables
               MatchIt, # For matching procedures
               rgenoud, # Helps with optimization problems
               ggplot2, # Plots!
               sjPlot, # Other plots!
               foreign # Read different file formats
               )
```

2 MPs for Sale?

What is the monetary value of serving as an elected politician? It is firmly a part of received wisdom that politicians often use their positions in public office for self-serving purposes. There is much evidence that private firms profit from their connections with influential politicians, but evidence that politicians benefit financially *because of* their political positions is thin. Eggers and Hainmueller (2009) seek to address this question by asking: what are the financial returns to serving in parliament? They study data from the UK political system, and compare the wealth at the time of death for individuals who ran for office and won (MPs) to individuals who ran for office and lost (candidates) to draw causal inferences about the effects of political office on wealth.[2] The data from this study is in Rdata format, and provided that you have downloaded the data and placed it within the right folder, then it can be loaded using the `load()` function as follows:

```r
load("eggers_mps.Rdata")
# If you haven't saved your .rmd code file in the same folder:
#load("pathtodatafolder/eggers_mps.Rdata")
```

Files in `.Rdata` format are automatically loaded with a pre-assigned name (`mps` in this case). The dataset includes observations for 425 individuals. There are indicators for the main outcome of interest – the (log) wealth at the time of the individual's death (`lnrealgross`), and for the treatment – whether the individual was elected to parliament (`treated == 1`) or failed to win their election (`treated == 0`). The data also includes information on a rich set of covariates, as described below

[2] Eggers, A.C., and Hainmuller J. (2009). "MPs for Sale? Returns to Office in Postwar British Politics." *American Political Science Review*, 103 (4): 513-533

Table 1: Variables in ‘mps.RData’, from Eggers and Hainmueller (2009)

Variable	Description
lnrealgross	Wealth at the time of the individual’s death (logged)
treated	Was individual elected to parliament? 1 = Yes, 0 = No
labour	Labour Party member (1) or not (0)
tory	Conservative Party member (1) or not (0)
yob	Year of birth
yod	Year of death
female	Female (1) or not (0)
aristo	Individual holds an aristocratic title (1) or not (0)
scat_**	Variables about secondary education of the individual, all prefixed scat_ (see the paper for details)
ucat_**	Variables about university education of the individual, all prefixed ucat_ (see the paper for details)
oc_**	Variables about pre-treatment occupation of the individual, all prefixed oc_ (see the paper for details)

A. Getting started

Task 1

Estimate the average treatment effect using either a `t.test` or a bivariate linear regression. What is it? Is it significantly different from zero? Can we interpret this as an unbiased estimate of the causal effect of elected office on wealth?

Task 2

There are clearly many potentially confounding differences between those who are elected and those who are not elected, preventing us from inferring anything about a causal relationship between serving as an MP and wealth at the time of death. That is, it is likely that the differences estimated in the regression above are subject to selection bias. Let's test whether that's the case.

> *Hint*: You can check selected variables one by one, produce a table, plot the values, or approach it it in any other way

B. Matching

The main function we will use is `matchit()`, from the `MatchIt` package. The function takes a number of arguments, the most important of which are listed in the table below. Note that, depending on the matching method, not all of the argument will be needed and then they will take on their default values.

Table 2: Arguments to the `matchit()` function[3]

Argument	Description
`formula`	A two-sided formula in the form of `treatment_variable ~ covariates`
`data`	A data frame containing the variables in `formula`.
`method`	The matching method to be used. Available options are, among others, `"exact"`, `"nearest"` and `"genetic"`.
`distance`	If applicable, the distance measure to be used, like `"euclidean"` or `"mahalanobis"`[4]
`estimand`	Either `"ATT"` to calculate the average treatment effect for the treated (the default), `"ATE"` for the average treatment effect, or `"ATC"` for the average treatment effect for the controls.
`replace`	For methods that allow it, should matched observations be re-used? `TRUE` or `FALSE`
`ratio`	For methods that allow it, how many control units should be matched to each treated unit in k:1 matching?
`m.order`	For methods that allow it, the order that the matching takes place. Set to `"random"` when ties should be broken randomly.

[3] You can also explore the help file for the function by running '?matchit'.

[4] To learn more about this distance measure, see the discussion linked here.

Let's go back to our MPs dataset.

Task 3

Can we use **matching** to remove or at least reduce the bias? The idea here is to make treatment independent of the potential outcomes **conditional** on observable covariates. Implement matching as discussed in the lecture using the `matchit()` function from the `MatchIt` package.

To start, let's use exact 1:1 matching. Remember, this means that for every treated unit we have to find a non-treated (control) unit that is exactly the same on all observables we specify. We first will match the observables that were found to be most imbalanced: Conservative Party MP (`tory`), aristocratic title (`aristo`), and whether the individual received their secondary education from Eton (`scat_eto`).

Task 3.1 How many matched observations are there?

Task 3.2 What is the average treatment effect on the treated (ATT)?

> *Hint:* Use the command `match.out <- matchit([formula], [data], method ="exact",estimand = "ATT",ratio = 1)`. Matches can be extracted by summarizing the `match.out` object and the matched data can be extracted with `match.data()` to then estimate the ATT with `lm()`.
>
> *Hint:* If you are curious, try to find out who these individuals are in the `mps` data using `mps[which(match.out$weights==0),]`.

Task 4

Re-evaluate the balance between treated and control observations on gender, aristocratic title, and whether the individual received their secondary education from Eton. Do this for the raw data, and then for the matched data with by regressing each covariate on the treatment variable.

> *Hint:* For the matched data, you need to specify the `weights`.

Task 5

Apply the `summary()` function on the output from `matchit()` and create a plot of the standardised mean differences.

> *Hint:* Have a look at the function's help file with `?plot.summary.matchit`.

Task 6

Rematch the data, this time expanding the list of covariates to include all of the schooling, university and occupation categories. Use exact matching again. What is the ATT? How many observations remain in the matched data?

C. Alternative specifications

Task 7

Eggers and Hainmueller run the matching analysis separately for Labour and Conservative candidates. Do this now and try to replicate the 3^{rd} and 6^{th} columns from Table 3 of their paper (see below). As you will see in the paper, they use M=1 genetic matching but you can get quite close to this result using a nearest neighbour 1:1 matching with replacement and mahalanobis distance. Adapt the code given below to do this. In their analysis, they do not only match on the categorical variables we have used so far, they also use the continuous variables for year of birth and year of death (`yob`,`yod`). Include these in the new matching procedure.

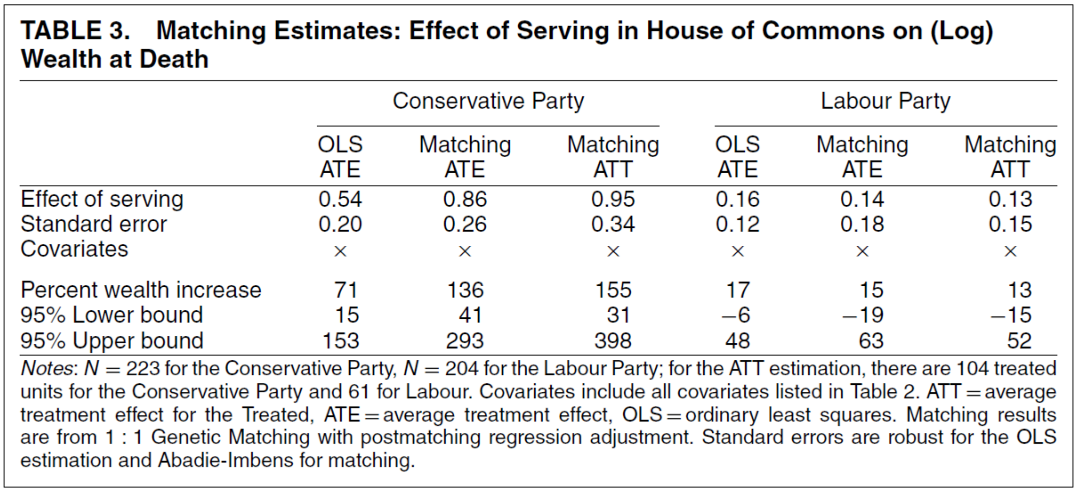

TABLE 3. Matching Estimates: Effect of Serving in House of Commons on (Log) Wealth at Death

	Conservative Party			Labour Party		
	OLS ATE	Matching ATE	Matching ATT	OLS ATE	Matching ATE	Matching ATT
Effect of serving	0.54	0.86	0.95	0.16	0.14	0.13
Standard error	0.20	0.26	0.34	0.12	0.18	0.15
Covariates	×	×	×	×	×	×
Percent wealth increase	71	136	155	17	15	13
95% Lower bound	15	41	31	−6	−19	−15
95% Upper bound	153	293	398	48	63	52

Notes: $N = 223$ for the Conservative Party, $N = 204$ for the Labour Party; for the ATT estimation, there are 104 treated units for the Conservative Party and 61 for Labour. Covariates include all covariates listed in Table 2. ATT = average treatment effect for the Treated, ATE = average treatment effect, OLS = ordinary least squares. Matching results are from 1 : 1 Genetic Matching with postmatching regression adjustment. Standard errors are robust for the OLS estimation and Abadie-Imbens for matching.

Figure 1: Table 3 from: Eggers and Hainmuellert (2009)

Task 8

Optional: For those of you who are interested, you can replicate the genetic matching approach that they use in the paper by using `method = "genetic"`. You should also provide a high number to `pop.size`, for instance 1000. However, this means it will take a while to run. You can decrease the number if you want, but note that this means that the estimates may be quite imprecise.