# ECON42720 Causal Inference and Policy Evaluation

### Assignment 1: Selection on Observables

Ben Elsner

## About this Assignment

This assignment teaches you how to apply the methods of Lectures 1-4 of the course (DAGs, regression, potential outcomes, matching). It will also teach you how to run Monte Carlo simulations, a critical skill for anyone working with data.

- The assignment is due on **Monday, 25th March 2024, 23:59**.
- You can work on this assignments in groups of up to 4. I will not get involved in the group formation process. If you prefer to work on your own, you can do so.
- Please submit one assignment per group on Brightspace under *Assessment > Assignments*. Make sure that all the names of the group members are on the assignment.
- The assignment should be submitted as one pdf file. It should contain a write-up with your answers to the questions (including tables and figures) and the R code you used to generate the results. Do not use code chunks; instead, provide the entire code at the end of the document.
- Formatting: I recommend using Quarto, but it's not a must. I expect nicely formatted tables and figures (use ggplot2 or similar for figures, stargazer or similar for tables). Before AI became broadly available, students could get away with ugly tables and figures because producing nices ones takes time. Not any more.
- AI Policy. I expect that you work on the solutions yourself. Feel free to use AI to help with coding and editing, but I warn against using it to come up with solutions. Please include an AI statement that explains how you used AI in your assignment.
- I will provide feedback on Brightspace to the person who submitted the assignment. Please share the feedback and the grade with the group.
- Criteria for grading: correctness of the analysis and answers, quality of the write-up, quality of the code, quality of the tables and figures, and the overall presentation of the assignment. All members of the group receive the same grade. Your grade will be on a scale from 0 to 100.

- To calculate the overall assignment grade for this module, I will take the average across both assignments and convert them to a letter grade based on the Alternative Linear Grading Scale (40% pass).

# 1 Tasks

## 1.1 DAGs and Potential Outcomes

The government has run a pilot of providing free school meals to children in a number of schools in areas with a low average household income. There was no randomisation in this process; the government simply picked schools in some of the poorest neighbourhoods in the country. You have been asked to evaluate the impact of this policy on children's health and education outcomes. Before you ask for access to administrative data, you want to think about a research design.

a) Draw a DAG that represents the causal relationship between the treatment (free school meals), one outcome (health), at least 3 confounders and at least one mediator. Justify your choices with respect to the confounders and mediator(s) and the direction of causal paths.

b) Using the DAG, explain how you would estimate the causal effect of free school meals on health.

c) You present your research design and findings to the representative of a ministry. After praising you for your thorough analysis, they raise a concern that your analysis doesn't account for educational attainment. Quite remarkably, they remember from their econometrics class 23 years ago that one should include all relevant variables to avoid omitted variable bias. They want to you to include a measure of educational attainment that was taken one year after the start of the programme. Draw two simple DAGs with the treatment, the outcome, one unobserved confounder (SES) and educational attainment: 1) one where health is on the causal path between the treatment and educational attainment, and 2) one where educational attainment is on the causal path between the treatment and health. In both cases, explain why including educational attainment would lead to bias.

d) Based on your explanation of all the biases that could arise, you convinced the ministry that a randomised control trial (RCT) would be the best way to estimate the causal effect of free school meals on health. They agree and ask you to design the RCT. Explain briefly how you would design such an RCT such that it satisfies the identification assumptions for the causal effect of free school meals on health. You do not need to comment on the logistics (i.e. suppose you can measure health perfectly and all schools and students will take part in the experiment).

## 1.2 Empirical Analysis: Preparation and Data Inspection

We will perform an empirical analysis if the effect of free school meals based on simulated data. This may seem strange to you, but working with simulated data is important to understand

the behaviour of estimators and to compare different methods. Before we test our methods on real data, we want to know how well they work in a controlled environment.

a) Please generate the following dataset using random number generators in R and save it as a data frame or tibble:

- set the seed to 123 `set.seed(123)` so that we all work with the same data.
- 500 observations
- a binary treatment variable `D` that is randomly assigned to 20% of the observations
- a covariate `family_income` that is normally distributed with a mean of 50,000 and a standard deviation of 10,000. The mean of `family_income` is 20,000 lower for the treated group.
- a covariate `parent_education` that is normally distributed with a mean of 12 and a standard deviation of 3. The mean of `parent_education` is 4 lower for the treated group.
- an error term $\varepsilon$ that is normally distributed with a mean of 0 and a standard deviation of 5.

Based on these variables, generate an outcome variable `health` as

$$\begin{aligned}\text{health} = 50 &+ 5 \cdot D \\ &+ 0.01 \cdot \text{family\_income} \\ &+ 0.5 \cdot \text{parent\_education} \\ &- 0.0005 \cdot \text{family\_income} \cdot \text{parent\_education} + \varepsilon\end{aligned}$$

b) Let's first inspect the data. Produce two plots: 1) a scatter plot of family income against health (y-axis), which separate regression lines and colours for treated and untreated observations, and 2) a scatter plot of parent education against health (y-axis), which separate regression lines and colours for treated and untreated observations. For each covariate, comment on common support.

c) Now produce separate density plots of each covariate, whereby each plot shows the distribution of the covariate for the treated and untreated group. Comment briefly on the difference between the distributions.

d) Using a logit model, estimate the propensity score for each unit. Plot the distributions of the propensity scores for the treated and untreated group in a density plot. Comment briefly on the difference between both distributions.

## 1.3 Empirical Analysis: Regression and Matching

a) Run the following regressions and report the results in a regression table:

1. A simple regression of health on the treatment variable `D`.
2. A regression of health on the treatment variable `D` and the covariate `family_income`.
3. A regression of health on the treatment variable `D` and the covariate `parent_education`.
4. A regression of health on the treatment variable `D`, the covariate `family_income` and the covariate `parent_education`.
5. A regression of health on the treatment variable `D`, the covariate `family_income`, the covariate `parent_education` and the interaction term between `family_income` and `parent_education`.

b) Consider the difference in the coefficient of the treatment between regressions 1 and 2. Using the omitted variable bias formula, explain why the coefficients differ.

c) Across all specifications, the coefficient of the treatment variable is closest to the assumed coefficient $\beta = 5$ regression 5. Explain why this is the case.

d) Perform propensity score matching (nearest neighbour) and obtain a matched dataset. Perform balancing tests for the covariates `family_income` and `parent_education` and comment on the results. I recommend the `MatchIt` package in R for this task.

e) Inspect the covariate balance visually. To do this, create density plots for each covariate that show the distribution of the covariate among treated and untreated units in the matched sample. Compare these to the distributions from 1.2 c).

f) Inspect the propensity score balance visually. To do this, create a density plot of the propensity scores for the treated and untreated units in the matched sample. Compare these to the distributions from 1.2 d).

g) Using the matched dataset, estimate the ATT and compare your result to the results from the regressions.

h) Perform tasks d)-g) again, but this time with coarsened exact matching (CEM). You can use the default procedure in `MatchIt` or similar packages; no need to manually specify any cutoffs. Compare the CEM results to the results from the propensity score matching.

i) Neither the PSM nor the CEM method give you the true ATT of $\beta = 5$. Why not?

## 1.4 Regression and Matching: Monte Carlo Simulation

Monte Carlo Simulations are often used to study the properties of estimators. With real data, we cannot easily assess the bias of an estimator because we don't know what the true value. Or we cannot look at the sampling distribution of an estimator, because we cannot draw repeated samples from the population. With Monte Carlo simulations, we can generate data with known properties, draw many samples, and analyse the properties of estimators. In the appendix (under `Hints`) you find more information on how Monte Carlo Simulations work and how to do them in R. One tip to give here is to first just simulate one data set and run the analysis. If that works, you can move to the simulation of many data sets.

a) Simulate 500 data sets. Each data set should contain the following variables. The data set should have the same structure as the data set you generated in 1.2 a), except that it has only 100 observations. In each replication, generate a matched data set based on Mahalanobis distance matching of the nearest neighbour. Then estimate the ATT using the matched data set.

b) Do the same as in a), but this time use Mahalanobis distance matching of the 10 nearest neighbours.

c) Produce a density plot of your estimates from a) and b) and explain why the two distributions look different.

# 2 Some Hints

## 2.1 Using a random number generator

We often want to generate random numbers, for example to run simulations or to randomly assign a treatment to units. When using random number generators, it is important to set a seed so that the results are reproducible. Without a seed, the random number generator will produce different numbers every time you run the code. Here is an example of how to set a seed and generate random numbers in R:

```r
set.seed(123)

# draw one random number from a standard normal distribution
rnorm(1, mean = 0, sd = 1)

# draw a random number from a binomial distribution (0 or 1) with a probability of 0.5
rbinom(1, 1, 0.5)

# Let's create a dataset with 100 observations and 3 variables

    n <- 100 # number of observations
    D <- rbinom(n, 1, 0.5) # 50% are treated here
    X1 <- rnorm(n, mean=100, sd=15) # a normally distributed covariate
    X2 <- rnorm(n, mean=50, sd=10) # another normally distributed covariate

    data <- data.frame(D, X1, X2) # create a data frame
    head(data)

# Now suppose you want to create an outcome through a data-generating process

    # generate an error term
    data$error <- rnorm(n, mean=0, sd=5)

    # generate the outcome from a DGP (data-generating process)
    data$Y <- 5 + 3 * data$D + 0.1 * data$X1 + 0.5 * data$X2 + data$error


# Let's run a regression
    reg <- lm(Y ~ D + X1 + X2, data = data)
    summary(reg)
```

```r
# The estimates you get here differ from the DGP because of sampling variation
```

## 2.2 Running a logit regression

You can run a logit regression using the `glm` function. Here is an example:

```r
library(stargazer)

# Simulate data with 100 observations and 2 covariates
set.seed(123) # Set seed for reproducibility
n <- 100
X1 <- rnorm(n)
X2 <- rnorm(n)
D <- rbinom(n, 1, 0.5) # 50% are treated here
data <- data.frame(X1, X2, D)

# Run a logit regression

logit_model <- glm(D ~ X1 + X2, data = data, family = binomial(link = "logit"))
stargazer(logit_model, type = "latex", header=FALSE) # Print the results
```

## 2.3 Running a Monte Carlo Simulation

Running a Monte Carlo simulation is actually quite easy. It typically follows the following steps:

1. Specify a data-generating process (DGP) that generates the data you want to analyse, for example a linear regression model.
2. Set the number of replications you want to run. In each replication, you will draw a new sample from the DGP and estimate the model.
3. In each replication, draw a new sample from the DGP, estimate the model and save the estimates. Repeat this step for the number of replications you specified.
4. Analyse the results of the Monte Carlo simulation. This could be the average of the estimates, the standard deviation, or the entire distribution.

Here is a simple example of how to run a Monte Carlo simulation in R which allows us to assess omitted variable bias and the sampling distribution of the OLS estimator.

```r
library(ggplot2)
```

```r
# STEP 1: set parameters

set.seed(123) # Set seed for reproducibility

reps <- 500   # Set the number of replications
n <- 100      # Set the sample size in each replication


# STEP 2:
# Write a function that generates the data and estimates the model
# This function will be called in each replication

simulate <- function(n) {

  # Generate the data
    error <- rnorm(n)
    D <- rbinom(n, 1, 0.5) # 50% are treated here
    X1 <- 0.4*rnorm(n) + 0.6*D # assume X1 is correlated with the treatment
    Y <- 5 + 3 * D + 0.5 * X1 + error # Specify the DGP
    data <- data.frame(X1, D, Y)

  # Estimate the model (we do not include X1 here)
  reg <- lm(Y ~ D, data = data)

  # Return the coefficient of the treatment variable
  return(coef(reg)[2])
}


# STEP 3: Run the Monte Carlo simulation
estimates <- replicate(reps, simulate(n))


# STEP 4: Analyse the results
mean(estimates) # The average of the estimates
sd(estimates)   # The standard deviation of the estimates

# Plot the sampling distribution of the OLS estimator

ggplot(data.frame(estimates), aes(x = estimates)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue", color = "black") +
```

```
  labs(title = "Sampling Distribution of the OLS Estimator", x = "Estimate", y = "Frequenc
  theme_minimal()
```