Problem Set 2

Your name here

2025-02-10

0.1 Preliminaries

In what follows, we will demonstrate the use and compare different methods to estimate causal effects in observational data, under the conditional ignorability of the treatment assumption. Let's denote units $i, \dots, N$, with treatment $D \in \{0,1\}$, and potential outcomes Y_d. In particular, as is common in this literature, we want to estimate an Average Treatment Effect on the Treated (ATT), defined as: $E[\tau_i|D=1] = E[Y_{1i}|D=1] - E[Y_{0i}|D=1]$.

We also observe a series of pre-treatment variables X_i. To estimate the ATT from observational data, in addition to SUTVA or consistency of the potential outcomes, we would need to assume the following:

- (Mean) Conditional independence of the control potential outcome and the treatment:

$$E[Y_{0i}] \perp\!\!\!\perp D_i|X_i$$

- Positivity or common support

$$0 < P(D|X) < 1$$

Under these conditions, we can estimate conditional treatment effects $E[Y_{1i} - Y_{0i}|X=x]$ for each X, and then recover the ATT using the adjustment formula:

$$\sum_{X_{D=1}} \mathbb{E}[\hat{\tau}_i|X]P[X_{D=1}]$$

Where $X_{D=1}$ represent the covariate distribution among the treated.

0.2 Data

For this exercise, we will use the classical data in LaLonde (1986). This dataset has been widely used to evaluate the performance of different methods to estimate causal effects using observational data, since it has the unique feature of providing an experimental component that could be used as a benchmark, and an observational component that can be used to test the proposed method.

The experimental portion of the data comes from the National Supported Work Demonstration (NSW), a labor training program that randomly selected participants to treatment ($N_t = 185$) and control ($N_c = 260$) conditions. The observational data, in addition to the treated group from the experiment, includes information about additional subjects from the Panel Survey of Income Dynamics (PSID), who are potential observational controls ($N_{c,obs} = 2490$).

The data includes 10 raw covariates: age, education (years), race (white, black, hispanic), marriage status (married/not), high school diploma (yes/no), and both previous earnings and unemployment (during 1974 and 1975). The outcome of interest is earnings in 1978. To better illustrate the difference between methods, and following Hainmueller (2012), we will use an expanded version of this covariates, including their squares and every two-way interaction (65 covariates in total).

```r
library("cobalt") # Package to assess covariate balance
```

```
 cobalt (Version 4.5.5, Build Date: 2024-04-02)
```

```r
library("causalsens") # lalonde experimental vs lalonde observational
library("tidyverse") # Just for data manipulation
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.2

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```
####################
### DATA LOADING ###
####################

# First, let's load the experimental dataset (lalonde.exp)
data(lalonde.exp, package="causalsens")
# Then also the observational dataset (lalonde.psid)
data(lalonde.psid, package="causalsens")

########################
### BASELINE BALANCE ###
########################
library("ebal") # Entropy balance (and matrixmaker function)
```

```
##
## ebal Package: Implements Entropy Balancing.

## See http://www.stanford.edu/~jhain/ for additional information.
```

```
# Let's expand the matrix of covariates
lalonde.exp.covs <- lalonde.exp |> select(-"treat",-"re78") |> matrixmaker()
lalonde.obs.covs <- lalonde.psid |> select(-"treat",-"re78") |> matrixmaker()
# Then list the covariates we will be using in the analysis
covs_raw <- lalonde.exp |> select(-re78, -treat) |> names()
covs <- names(lalonde.obs.covs)

# Create a single dataset with both obs and experimental data
# Just to plot balance in both
lalonde.mix <-
  rbind(lalonde.exp.covs |>
          mutate(treat = lalonde.exp$treat, wexp=1, wobs=ifelse(treat==1,1,0)),
        lalonde.obs.covs[lalonde.psid$treat==0,] |> # Observational controls
        mutate(treat=0, wexp=0, wobs=1)) # Weights for obs data
#names(lalonde.mix)

# Create "loveplot" to show covariate distribution in both samples
```

0.3 Balance

A first interesting inspection of the data is to assess the overall covariate balance in the different samples. As we can see in the first figure, in the experimental sample all the covariates are very well balanced among treated and controls. In the observational sample, on the other hand, treated and control subjects are very different, in particular with respect to their previous earning trajectories.

```r
# The bal.table function takes the raw data and weights
# And outputs a table of standardized difference between treated and controls
# Love.plot takes a bal.table object as inputs
# And transform it into a plot to facilitate presentation

bal.tab(f.build("treat", covs), data=lalonde.mix,
        estimand = "ATT", m.threshold =.1,
        disp.var.ratio=FALSE, disp.means=TRUE,
        weights = c("wexp","wobs")) |> # Weigght for experimental and observational sample
  love.plot(abs=FALSE,
            sample.names =
              c("Combined","Experimental","Observational")) +
  labs(title= " ") + theme(legend.position = "top")
```

```
Warning: Large mean differences detected; you may not be using standardized
mean differences for continuous variables.
```

```
Warning: Standardized mean differences and raw mean differences are present in the same plot
Use the `stars` argument to distinguish between them and appropriately label the x-axis.
```

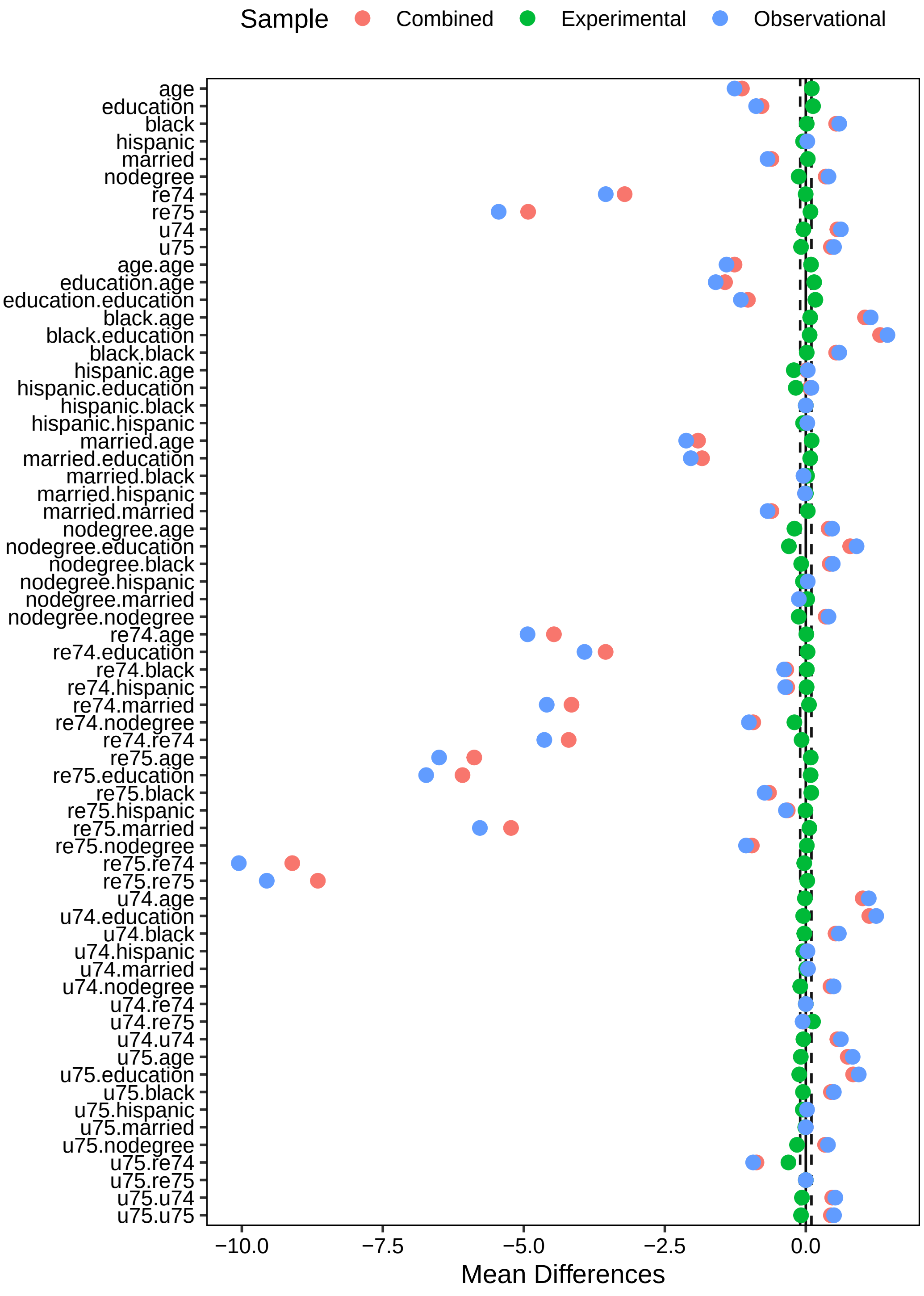

Figure 1: Covariate Balance by sample

0.4 Treatment effect

A second aspect that is worth noting are the estimated returns of the program, as we can see in the next figure. The experimental benchmark is around 1,600 and 1,800 USD, depending if we use a simple difference in means or the Lin estimator (a regression model with all raw covariates and their interactions with the treatment).

The observational difference in means, as expected from the huge covariate imbalance, is completely off: -15,205 USD. The multiple regression results with the observational data are more interesting, however. Although the point estimate is way below the point estimate from the experimental benchmark (suggesting only a 4USD return), the confidence interval resulting from the regression model overlaps with the benchmark returns. That's it, the regression model does a good job in accounting for the uncertainty associated to estimate the effect of the program, providing a wide and uninformative confidence interval.

```
#######################
### BASELINE MODELS ###
#######################
library("estimatr") # Lin estimator

# Bivariate regression on experimental data: 1,794 USD
exp_biv <- lm(re78 ~ treat, data=lalonde.exp)
# Lin estimator (treatment fully interacted with covs) on experimental data: 1,583 USD
exp_lin <- lm_lin(f.build("re78", "treat"), covariates = f.build("", covs_raw), data=lalonde
# Bivariate linear regression on observational data: -15,205 USD
obs_biv <- lm(re78 ~ treat, data=lalonde.psid)
# Multiple regression with controls on observational data: 4.159 USD
obs_reg <- lm(f.build("re78", c("treat", covs_raw)), data=lalonde.psid)
# Combine results (Store in a matrix for plotting)
level_order <- c("Experimental (diff-in-means)","Experimental (Lin estimator)",
             "Observational (Regression)","Observational (diff-in-means)")

# Compile the resulting coefficients into a single matrix
coeffs <-
  tibble(estimator =
           factor((c("Experimental (diff-in-means)",
             "Experimental (Lin estimator)",
             "Observational (Regression)",
             "Observational (diff-in-means)")), levels=level_order),
         coeff = c(exp_biv$coefficients[2],
                   exp_lin$coefficients[2],
                   obs_reg$coefficients[2],
                   obs_biv$coefficients[2]),
         se = c(summary(exp_biv)$coefficients[2,2],
                exp_lin$std.error[2],
                summary(obs_reg)$coefficients[2,2],
                summary(obs_biv)$coefficients[2,2]))

# Create the annotated plot
ggplot(coeffs, aes(x=coeff, y=estimator)) +
  geom_point() +
  # Error bars (Confidence intervals)
  geom_errorbarh(aes(x=coeff, y=estimator,
                     xmin=coeff-2*se, xmax=coeff+2*se),
                 height=0) +
  geom_vline(xintercept = 0, linetype="dashed", color="red") +
  # Annotations (point estimates)
  annotate("text", x = -2500, y = 1, label = "1,794 USD") +
  annotate("text", x = -2500, y = 2, label = "1,583 USD") +
  annotate("text", x = -4000, y = 3, label = "4.159 USD") +
  annotate("text", x = -10000, y = 4, label = "-15,205 USD") +
  # Labels
  theme_classic() +
  labs(title = "Estimated return of the NSW training program",
       subtitle = "(Outcome: Earnings in 1978)",
       x = " ", y= " ")
```

```
Warning in geom_errorbarh(aes(x = coeff, y = estimator, xmin = coeff - 2 * :
Ignoring unknown aesthetics: x
```

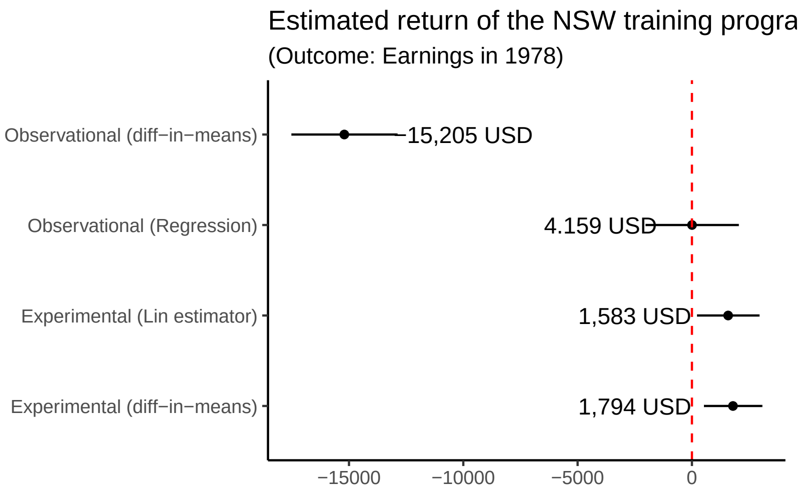

Figure 2: Estimated returns of the training program.

1 Overlap

Note: for this section, you may want to use the `cobalt` package in `R`. See reference here

Now it's your turn. First, we will assess overlap in the covariate distribution. Since it is not easy to do it directly with many covariates, you will need to estimate a *propensity score*. This is, a model for $P(D_i|X_i)$.

(a) Estimate a logistic regression with D as the outcome and the `covs_raw` as the predictors.

```
# Here your code
```

(b) Estimate an alternative logistic regression, with D as th outcome and `covs` as the predictors.

```
# Here your code
```

(c) Now, from the model in (a), get both the predicted probabilities ($\in [0, 1]$) and the logits ($\in [-\infty, +\infty]$). Then create a density plot overlying the distribution of the predicted probabilities for treated and the *observational* control units. Do the same with the logits.

```
# Here your code
```

(d) Now, from the model in (b), get both the predicted probabilities ($\in [0, 1]$) and the logits ($\in [-\infty, +\infty]$). Then create a density plot overlying the distribution of the predicted probabilities for treated and the *observational* control units. Do the same with the logits.

```
# Here your code
```

(e) What can you conclude from the previous exercise?

2 Matching

For this section, you are asked to use the `MatchIt` package in `R` to estimate the Average Treatment Effect on the Treated using a matching approach. You can find documentation and tutorials here.

(a) Select one distance measure and one matching method. Explain how they work, and justify why do you think these are the most sensible choices.

(b) Perform the matching procedure using the `matchit` routine. Show your results (i.e., print the matching object)

```
# Here your code
```

(c) Assess the balance obtained: i. using `summary` on your matching object; ii. `plot` the distribution of the propensity scores using the `type = "jitter"` option; iii. `plot` the balance of covariates using the `type = "density"` option; iv. `plot` the balance of covariates calling the function on the `summary` of your matching object. Interpret all outputs.

```
# Here your code
```

(d) Estimate the treatment effect, and interpret the results.

```
# Here your code
```

2.1 Extra credit

(e) Repeat b) to d) using an alternative propensity score specification, and a different matching method.

(f) How do results compare with the previous? Assuming you *cannot* see the experimental benchmark, what estimate would you prefer and why?

3 Weighting

For this section, you are asked to use the `WeightIt` package in `R` to estimate the Average Treatment Effect on the Treated using a weighting approach. You can find documentation and tutorials here.

(a) Select two balancing methods, one based on the direct estimation of the propensity score (balance *in expectation*), and one directly targeting balance *in sample*. Explain how they work, and justify why do you think these are the most sensible choices.

```
# Here your code
```

(b) Perform the weighting procedure using the `weightit` routine. Show your results (i.e., print the output object)

```
# Here your code
```

(c) Assess the balance obtained: i. using `summary` on your weighting object; ii. using `bal.tab` on your weighting object; iii. creating a plot showing mean balance for all covariates, using the `cobalt::love.plot()` function. Interpret all outputs.

```
# Here your code
```

(d) Estimate the treatment effect, and interpret your results.

```
# Here your code
```

3.1 Extra credit

(e) Repeat b) to d) using an alternative weighting method.

(f) How do results compare with the previous? Assuming you *cannot* see the experimental benchmark, what estimate would you prefer and why?