Pset4

Yu Hui, Janet Cao

2024-04-12

Objective: In Pset3 you estimated the ATE of the offer of training on post-training earnings using NSW experimental data. The Treatment-Control difference in sample averages indicates that the offer of training causes an additional $1,794 in terms of 1978 earnings. Variation in the cause/treatment is often observational in nature, instead of resulting from an RCT. In this Pset you utilize methods developed to estimate the effect of the offer of training using "observational data" and apply them to two datasets that Dehajia and Wahba constructed to mimic observational data.

Background: Consider the files `nswcps.csv` and `nswpsid.csv`. Each file contains a dataset. Each dataset combines two samples: 1) the treated sample from the Dehajia and Wahba's NSW data (i.e., 185 males offered NSW training in 1976-1977)[1]; and 2) a sample extracted from a large survey: a) in `nswcps.csv`, such sample is the Current Population Survey (CPS); b) in `nswpsid.csv`, such sample is the Panel Study of Income Dynamics (PSID). The samples in 2) contain data on a comparison group, that is, on subjects who (as far as we know) did not receive the NSW offer of training.[2] Specifically, the PSID sample (called **PSID-1**) consists of 2,490 male household heads under the age of 55 who are not retired; and, the CPS sample (called **CPS-1**) consists of 15,992 male household heads under the age of 55 who are not retired. The file `nswcps.csv` (respectively, `nswpsid.csv`) contains the treated individuals (from NSW-treated) along with the PSID (respectively, CPS) comparison individuals. The treatment indicator variable `treat` equals 1 for individuals in the NSW-treated sample and zero for the PSID (respectively, CPS) comparison individuals.

Part 1: Describe the Data (10 p)

Q1(4 p) Fill Table 1 columns 5 and 6 using, respectively, the data in `nswpsid.csv` and in `nswcps.csv`. Notes: You want to limit attention to observations with `treat=0`. You filled columns 3 and 4 in PSet 3.

```
#import data
df.psid<-read.csv("nswpsid.csv")
df.cps <-read.csv("nswcps.csv")

# summarizePSID
table.psid<-df.psid[df.psid$treat==0,]%>%
summarize_all(list(~mean(.)))%>%#summarizemean
mutate_all(list(~round(.,3)))%>% #limitdigits
t()%>% #transpose
```

[1] Dehejia and Wahba (1999) Causal Effects in Nonexperimental Studies: reevaluating the Evaluation of Training Programs, *JASA*, pp. 1053-1062. Dehejia and Wahba (2002) Propensity-score Matching Methods for Nonexperimental Causal Studies, *ReStat*, pp. 151-161.

[2] When working with observational data the untreated sample is more properly called a comparison group. Nevertheless it is common to use the terms control and comparison interchangeably, irrespective of whether the variation in the treatment indicator is induced by RA or not.

```
as.data.frame()
colnames(table.psid)<-c("psid_control")
table.psid<-tibble::rownames_to_column(table.psid,"varname")

#summarizeCPS
table.cps<-df.cps[df.cps$treat==0,]%>%
summarize_all(list(~mean(.)))%>%#summarizemean
mutate_all(list(~round(.,3)))%>% #limitdigits
t()%>% #transpose
as.data.frame()
colnames(table.cps)<-c("cps_control")
table.cps<-tibble::rownames_to_column(table.cps,"varname")
table<-merge(table.psid,table.cps,by="varname")
knitr::kable(table)
```

varname	psid_control	cps_control
age	34.851	33.225
black	0.251	0.074
edu	12.117	12.028
hisp	0.033	0.072
married	0.866	0.712
nodegree	0.305	0.296
re74	19428.745	14016.800
re75	19063.337	13650.804
re78	21553.921	14846.660
treat	0.000	0.000
u74	0.086	0.120
u75	0.100	0.109

Q2 (4 p) Briefly comment on the completed Table 1. Hint: Are the PSID-1 and CPS-1 samples “good” control groups?

We have tested in PSet 3 that the experiment control group is significantly different from the treatment group in nodegree. But the PSID-1 and CPS-1 sample has a even larger difference in characteristics compared to the treatment group. The PSID and CPS sample are 8 to 9 years older, about 70% more likely to be married, 50% more likely to finish 12-year education,and have 2 more years of education. The NSW sample is 50% more likely to be black. The survey sample’s earnings in 1974, 1975, and 1978 are more than 10,000 dolars higher than the NSW sample. For the pre-determined variables, we can conclude that the the sample extracted from a large survey is different from the individuals placed in NSW training program. In general, the worker group selected to participate in RCT is under disadvantage in labor market compared to the whole population. Therefore, using survey sample as control of the treated group in NSW is not appropriate and will lead as to very biased (probably downwards) estimates of ATE.

Q3 (2 p) Why do you think that Dehajia and Wahba constructed their “observational datasets’ ’ by pulling together the treated sample from NSW and a sample of individuals drawn from either the PSID or the CPS data?

Although both PSID and CPS include information on whether an individual enrolled in a training course during the previous 12 months, this enrollment is not guaranteed to be random. Exclusively exploiting the observational variation in whether an individual enrolled in a training program will not lead to a causal interpretation. The treatment and control in this approach is unlikely to have the same potential outcome w/

or w/o treatment. The NSW experiment data ensures that the training assignment is random. Using these as the treatment group makes it more likely to identify ATE with some adjustment. For example, though the control group is drastically different to the treatment group in OPVs now, we can attempt propensity score matching approach to solve this concern.

Part 2: Regression-based Estimation of TEs (90 p)

You use the `nswpsid.csv` dataset to estimate the treatment effect (TE) of the offer of training via regression-based approaches associated with the following three specifications of the outcome equation:

$$
\begin{aligned}
re78_i &= \alpha + \rho D_i + u_i,\ i = 1, ..., 2675, && (1)\\
re78_i &= \alpha + \rho D_i + \mathbf{x}_i'\beta + u_i,\ i = 1, ..., 2675, && (2)\\
re78_i &= \rho D_i + g(\mathbf{x}_i) + u_i,\ i = 1, ..., 2675, && (3)
\end{aligned}
$$

Subscript i denotes an individual. Also: 1) $re78_i$ represents the data field `re78`; 2) D_i represents the data field `treat`; 3) $\mathbf{x}_i$ represents a $K \times 1$ vector of observed pre-determined variables (OPVs); and, 4) $g(\cdot)$ is an unknown and possibly non-linear function (i.e., a generalization of $\alpha + \beta'\mathbf{x}_i$). Table 2's column [1] references the regression specification. Column [2] gives the name of the approach. Column [3] indicates the regression coefficient of interest. You complete columns [4] and [5] with the estimate of the regression coefficient and its standard error (SE).}

Reference Model	Name of the Estimation Approach	Parameter of Interest	Estimate	SE
[1]	[2]	[3]	[4]	[5]
expression (1)	Treatment-Control Comparison (TCC)	ρ	-15204.776	1154.614
expression (2)	Regression-Adjusted Treatment-Control Comparison (Adj. TCC)	ρ	217.9438	655.6691
expression (3)	Double Machine Learning (DML)	ρ	-654.5	1121.8

Table 2: Treatment Effect Estimates Based on Three Regression-Based Approaches Applied to Observational Data.

Treatment Control Comparison Approach

Q4 (30 p) These questions pertain to the specification in expression (1) thus you obtain the Treatment-Control Comparison (TCC) Estimator of the treatment effect of the offer of training.

a) Estimate ρ. Programming Guidance: Use `stats::lm( )`. Say that your linear model is `m1 <- lm(re78 ~ treat, data = df)`. View the SEs of estimator $\hat{\rho}$ by using `summary(m1)$coefficients["treat", c("Estimate", "Std. Error"]`. View all SEs by using `lmtest::coeftest(m1, vcov. = vcov(m1))` which runs t-tests for each of the coefficients using the variance-covariance matrix estimated assuming **homoschedasticity**. Package `lmtest` allows you to perform z and t tests on estimated coefficients from, among others, method `lm( )`. It returns a coefficient matrix with columns containing the estimates, associated SEs, test statistics, and p-values.

```r
df <- df.psid
m1 <- lm(re78 ~ treat, data = df)
summary(m1)$coefficients["treat", c("Estimate", "Std. Error")]
coef1 <- summary(m1)$coefficients["treat", "Estimate"]
coeftest(m1, vcov. = vcov(m1))
```

```
##   Estimate Std. Error
## -15204.776   1154.614
##
## t test of coefficients:
##
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  21553.92     303.64  70.985 < 2.2e-16 ***
## treat       -15204.78    1154.61 -13.169 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

b) (10 p) Compute heteroschedasticity-robust SEs.\textcolor{gray}{Programming Guidance: There are multiple R packages to estimate the variance-covariance matrix of $(\hat{\alpha}, \hat{\rho})$ under general heteroschedasticity. Here are two ways. Option 1: Use `sandwich::vcovHC(m1, type = "HC0")` from package `sandwich`. Option 2: Use `car::hccm(m1, type = "hc0")` from package `car`. In both cases, the argument `type = "hc0"` (or `"HC0"`) tells R that you want to use the variance covariance matrix estimated using White's (1980) estimator, often referred to as HCE (heteroscedasticity-consistent estimator).

```
# the variance covariance matrix estimated using White's (1980) estimator
vcovHC(m1, type = "HC0")
coeftest(m1, vcov.= vcovHC(m1, type = "HC0"))
```

```
##             (Intercept)     treat
## (Intercept)    97137.19 -97137.19
## treat         -97137.19 429901.96
##
## t test of coefficients:
##
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  21553.92     311.67  69.157 < 2.2e-16 ***
## treat       -15204.78     655.67 -23.190 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c) (2 p) Verify that $\hat{\rho}$ in equals $(\overline{re78}^{D=1} - \overline{re78}^{D=0})$, i.e., the difference between the average post-training earnings of the treated and of the control individuals. This fact explains the name of the estimator, and is consistent with what you derived in previous Psets.

```
mean.diff <- mean(df[df$treat==1, "re78"]) - mean(df[df$treat==0, "re78"])
mean.diff
coef1
```

```
## [1] -15204.78
## [1] -15204.78
```

d) (10 p) Intuitively explain why the TCC approach may not deliver a credible estimate of the average effect of the treatment of interest. **Hint: Use the result in to think about what this approach uses to proxy for the missing data, i.e., for the control units' mean of the potential outcome w/ treatment, and for the treated units' mean of the potential outcome w/out treatment.**

In the Treatment-Control Comparison (TCC) approach, the mean difference in outcome between the control and treatment group is taken as the average treatment effect (ATE). The ATE is defined as:

$$\begin{aligned}
\text{ATE} \equiv\ & \mathbb{E}[y_{1i} - y_{0i} \mid D_i = 1] \Pr(D_i = 1) + \mathbb{E}[y_{1i} - y_{0i} \mid D_i = 0] \Pr(D_i = 0) \\
=\ & [\mathbb{E}[y_{1i} \mid D_i = 1] - \mathbb{E}[y_{0i} \mid D_i = 1]] \Pr(D_i = 1) \\
& + [\mathbb{E}[y_{1i} \mid D_i = 0] - \mathbb{E}[y_{0i} \mid D_i = 0]] \Pr(D_i = 0) \\
=\ & [\mathbb{E}[y_i \mid D_i = 1] - \mathbb{E}[y_{0i} \mid D_i = 1]] \Pr(D_i = 1) \\
& + [\mathbb{E}[y_{1i} \mid D_i = 0] - \mathbb{E}[y_i \mid D_i = 0]] \Pr(D_i = 0).
\end{aligned} \tag{4}$$

This approach uses the control units' mean outcome without treatment to proxy the treated units' mean potential outcome without treatment, and uses treated units' mean outcome with treatment to proxy control units' mean potential outcome with treatment.

The reliability of the above actions strongly relies on the assumption that the treated react the same as the controlled when facing treatment. Usually, researchers provide evidence on this assumption by testing balance on pre-determined variables. Providing that two groups are not significantly different in observed pre-determined characteristics, researchers may argue that we can compare the two groups for ATE.

However, as we have discussed in Part 1, the treated group from the NSW experimental data and the control group from the PSID survey sample, are highly unbalanced. That is, the MIA assumption may fail in this context, indicating that $\mathbb{E}[y_{1i} \mid D_i = 1] \neq \mathbb{E}[y_{1i} \mid D_i = 0]$ and $\mathbb{E}[y_{0i} \mid D_i = 1] \neq \mathbb{E}[y_{0i} \mid D_i = 0]$. They are drastically different in almost all aspects like average age, education level, marriage status, and pre-treatment earnings. This explains why we obtain a highly unreliable ATE estimate of $\$-15204.78$ for the training.

Control Fnc Approach

Q5: (20 p) These questions pertain to the specification in expression (2) thus you obtain the Regression-Adjusted Treatment-Control Comparison (Adj. TCC) Estimator of the treatment effect of the offer of training.

a) (10 p) Add to the model estimated in ?? the following OPVs as regression covariates: `age`, `agesq`, `edu`, `nodegree`, `black`, `hisp`, `re74`, and `re75`. Report $\hat{\rho}$ and its heteroschedasticity-robust SE. Programming Guidance: Add column `agesq` (age squared) to your dataframe using, e.g., `dplyr::mutate( )`.

```
df <- mutate(df, agesq=age*age)
m2 <- lm(re78 ~ treat + age + agesq + edu + nodegree + black + hisp + re74 + re75, data = df)
rho2 <- summary(m2)$coefficients["treat","Estimate"]
rho2
```

```
## [1] 217.9438
```

```
se2 <- coeftest(m2, vcov.= vcovHC(m2, type = "HC0"))["treat", "Std. Error"]
se2
```

```
## [1] 766.4444
```

b) (10 p) Intuitively explain why the Adj. TCC approach may be regarded as an improvement over the TCC approach when it comes to credible identification/estimation of average treatment effects.

As mentioned above, the problem with the Treatment-Control Comparison (TCC) approach lies in the fact that the control group is nowhere near balanced compared to the treatment group. It is intuitive that these observed pre-determined variables (OPVs) can be strongly related to income. Without controlling for these variables, the estimates mix treatment effects with the earning effects contributed by differences in OPVs for the two groups.

By adding the OPVs like *age*, *agesq*, *edu*, *nodegree*, *black*, *hisp*, *re74*, and *re75*, we can partial out the effects of the observed variables on the outcome earnings. To be more specific, the estimates are closer to the Average Treatment Effect (ATE) because we are comparing the difference between the treatment and control group on the residual of outcome earnings after regressing on all the OPVs. This is an improvement (as we can see the coefficient estimate is now 217.9438, at least positive) since we observe sharp differences in OPVs between the PSID survey sample and the NSW sample.

In mathematical language, we are assuming a weaker assumption, the Conditional Mean Independence Assumption (CMIA):

$$\mathbb{E}[y_{ji} \mid D_i, x_i] = \mathbb{E}[y_{ji} \mid x_i] \quad \forall j = 0, 1; \quad \forall x_i. \tag{5}$$

For each x cell, we estimate the Conditional Average Treatment Effect (CATE).

For OPVs that do not predict treatment, adding them can account for some variation in $re78_i$, hence increasing the precision of the estimator of $\hat{\rho}$. The Standard Error (SE) of ATE will decrease and we can better discriminate among competing hypotheses concerning the Treatment Effect (TE).

Partialing Out

Q6 (20 p) Consider again the specification in expression 2 estimated in Q2. Here, you implement two procedures, as detailed below, to verify the "partialling-out" interpretation of OLS coefficients in MLRM.

(a) (8 p) Procedure A: (i) (4 p) First Stage: Regress treat on a constant and the OPVs listed. ii. (4p) Second Stage: Regress re78on a constant and the residuals from the previous question

```r
# First Stage
 s1 <- lm(treat ~ age + agesq + edu + nodegree + black + hisp + re74 + re75, data = df)
 df$r1 <- resid(s1) # residual
 # Second Stage
 s2 <- lm(re78 ~ r1, data = df)
 rho2a <- summary(s2)$coefficients["r1","Estimate"]
 rho2a
```

```
## [1] 217.9438
```

(b) (8 p) Procedure B:

(i) (4 p)(0 p) First Stage: Same as Q6ai

(ii) (4 p)First Stage: Regress `re78` on a constant and the OPVs listed in Q5a; obtain the residuals.

(iii) (4 p)(4 p) Second Stage: Regress the residuals from Q6bii on the residuals from Q6Bi.

```
# First Stage
s1a <- lm(treat ~ age + agesq + edu + nodegree + black + hisp + re74 + re75, data = df)
df$r1 <- resid(s1) # residual
s1b <- lm(re78 ~ age + agesq + edu + nodegree + black + hisp + re74 + re75, data = df)
df$r2 <- resid(s1b) # residual of s1b
# Second Stage
s2b <- lm(r2 ~ r1, data = df)
rho2b <- summary(s2b)$coefficients["r1","Estimate"]
rho2b
```

```
## [1] 217.9438
```

(c) (4p) Verify that the estimates of the slope coefficient from Q3(a)ii and Q3(b)iii are numerically identical to $\hat{\rho}$ obtained in Q2a. Use this fact to give meaning to the expression "partialling-out" interpretation of OLSin a MLRM.

```
rho2
```

```
## [1] 217.9438
```

```
rho2a
```

```
## [1] 217.9438
```

```
rho2b
```

```
## [1] 217.9438
```

The three estimates are numerically identical. According to the background, OLS is essentially partialling-out the effects of a constant on both D_i and y_i.

If we extend the constant to a series of OPVs, MLRM is equivalent to (1) partialling out the OPVs' effects on D_i and then (2) regressing y_i on the residuals from the previous step. This is why the Q2a (MLRM) estimate equals to the Q3a estimate.

Since in Q3a, we have partialed out OPVs' effects on D_i before checking D_i's effects on y_i, the estimate does not contain the effects of OPVs. So, if we also partial out the OPVs effects on y_i, the regression of residuals on residuals (Q3b) is identical to Q3a. We have shown that this conclusion holds.

DoubleML

Q4 (20 p) Consider the partially-linear specification in expression. Here you estimate ρ via the the Double Machine Learning (DML) estimation procedure of Robinson (1988)\footnote{Robinson, P. M. (1988). Root-N-consistent semi-parametric regression. Econometrica 56, 931-54.

(a) (2 p) Install four R packages: DoubleML, data.table, mlr3, and mlr3learners.

```
# install.packages("DoubleML")
# install.packages("data.table")
# install.packages("mlr3")
# install.packages("mlr3learners")
# install.packages("ranger")
```

(b) (2 p) If your data is not already a `data.table` object convert it. Programming Guidance: Assuming that your dataframe is called `df`, use `dt <- data.table::as.data.table(df)`. `data.table` is an extension of `data.frame` and allows for fast manipulation of very large data.

```
# install.packages("data.table")
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:zoo':
##
##     yearmon, yearqtr

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
dt <- as.data.table(df)
```

(c) (2 p) Collect all the original OPVs in a list named, for example, `pretreat_colnames`. Note: Henceforth when we refer to these OPVs in mathematical expressions we use the notation x_i.

```
pretreat_colnames <- c("age", "edu", "nodegree", "black", "hisp", "married","u74", "u75", "re74", "re75
```

(d) (2 p) Specify data and variables for the causal model by running the script:

```
dml_data_psid <- DoubleML::DoubleMLData$new(dt,
                            y_col = "re78",
                            d_cols = "treat",
                            x_cols = pretreat_colnames)
```

(e) (2 p) Suppress messages from the `mlr3` package by adding `lgr::get_logger("mlr3")$set_threshold("warn")` to your script.

```r
lgr::get_logger("mlr3")$set_threshold("warn")
```

(f) (2 p) Here you mimic the first stage of Procedure B in Q3b. Namely, you specify the model for the two regression functions $l(\mathbf{x}) = E[\texttt{re78}_i|\mathbf{x}_i = \mathbf{x}]$ and $m(\mathbf{x}) = E[\texttt{treat}_i|\mathbf{x}_i = \mathbf{x}]$. In Q3b you used a linear-in-parameter model and a priori decided which OPVs to include and which transformations to apply to the OPVs to include (e.g., you excluded `u74`, you used both `age` and `agesq`, you left as-is the other included OPVs). Instead here you do not a priori exclude any OPVs, and you use flexible models, which accommodate complex non-linearities. Run the script:

```r
# Specify a RF model as the learner model for l(x)=E[re78|X=x]
ml_l_rf <- mlr3::lrn("regr.ranger")
# Specify a RF model as the learner model for m(x)=E[treat|X=x]
ml_m_rf <- mlr3::lrn("classif.ranger")
```

The above script uses a Random Forest (RF) model for both conditional expectations functions.[3]

(g) (2 p) Here you initialize & parametrize the model object which you later use to perform estimation. Run the script:

```r
# Set seeds for cross-fitting
set.seed(3251)
# Set the DML specification
obj_dml_plr <- DoubleML::DoubleMLPLR$new(dml_data_psid, # from Q4d
                                         ml_l = ml_l_rf, ml_m = ml_m_rf,
                                         n_folds = 2,
                                         score = "partialling out",
                                         apply_cross_fitting = TRUE)
```

The above script: (i) utilizes the data object generated in Q4d, namely `dml_data_psid`; (ii) utilizes the models for the first stage regressions picked in Q4d, namely `ml_l_rf` and `ml_m_rf`; (iii) specifies that we want to split the sample into 2 parts (`n_folds = 2`), and (iv) that we want to use the "partialling out'' approach to estimate causal impacts (`score = "partialling out"`), and (v) that we want to apply cross-fitting (`apply_cross_fitting = TRUE`).

(h) (2 p) Here you fit the DML model defined in . Run the script:

```r
obj_dml_plr$fit()
obj_dml_plr
```

[3] You do not need to know what a RFM is. Think of this approach as a way to flexibly estimate the form of a function of many variables. If you want to learn more about these approaches consider taking ECMA 31350 in Winter 2024.

```
## ================= DoubleMLPLR Object ==================
##
##
## ------------------ Data summary      ------------------
## Outcome variable: re78
## Treatment variable(s): treat
## Covariates: age, edu, nodegree, black, hisp, married, u74, u75, re74, re75
## Instrument(s):
## No. Observations: 2675
##
## ------------------ Score & algorithm ------------------
## Score function: partialling out
## DML algorithm: dml2
##
## ------------------ Machine learner   ------------------
## ml_l: regr.ranger
## ml_m: classif.ranger
##
## ------------------ Resampling        ------------------
## No. folds: 2
## No. repeated sample splits: 1
## Apply cross-fitting: TRUE
##
## ------------------ Fit summary       ------------------
##  Estimates and significance testing of the effect of target variables
##       Estimate. Std. Error t value Pr(>|t|)
## treat    -560.5     1006.4  -0.557    0.578
```

At a high level the above script implements all of the following operations: (i) fits the two models for the first stage selected in Q7f, (ii) gets residuals, (iii) regresses the residuals for the outcome variables onto the residuals for the treatment indicator to obtain the DML estimate of ρ in expression (3). Note: You specified `n_folds = 2` and requested `apply_cross_fitting = TRUE` in Q7g thus the 2-stage estimation procedure proceed as follows. First the entire data is split into two sub-samples, call them A and B (hence the term 2 folds). Sample A is used to fit the 1st stage models. These fitted models are used to compute residuals in sample B and these residuals are used to fit the 2nd stage model using only data in sample B. Denote the resulting estimate $\hat{\rho}_{AB}$. Then the samples are swapped (hence the term “cross fitting’ ’).[4] That is, sample B is used to fit the 1st stage models. Sample A is used to fit the 2nd stage model. Denote the resulting estimate $\hat{\rho}_{BA}$. The DML estimate is the average of $\hat{\rho}_{AB}$ and $\hat{\rho}_{BA}$.

The DML estimate of ATE is -560.5 (SE 1006.4).

(i)(4 p) Take a look at the output, i.e., at the object `obj_dml_plr`. How does the DML estimate of average treatment effect compare to the estimates based on specifications (1) and (2)?

The Double Machine Learning (DML) model estimates the average treatment effect (ATE) at -560.5 with a standard error of 1006.4, a figure that falls between the significant negative estimate from the basic Specification 1 and the positive but smaller estimate from Specification 2.

Neither the DML estimate nor that from Specification 2 are statistically significant at a 5% level, indicating the data does not conclusively show a treatment effect different from zero. This may suggest an imbalance in OPVs between treated and untreated groups or other complexities in the data.

[4] Cross-fitting is implemented to eliminate the bias from overfitting resulting from the fact that the two conditional mean functions $l(\cdot)$ and $m(\cdot)$ are estimated via ML models, in our case the RF models specified in **??**.

Specification 1, while statistically significant, is limited by its simplicity and may fail to address confounding factors. Conversely, Specification 2 accounts for OPVs but lacks the DML's sophistication in modeling non-linear effects, possibly leading to different results.