

Estimation of average treatment effects

CONTENTS:

- **Regression methods**
- **Propensity score matching**

Regression methods

The **average treatment effect (ATE)** is a measure used to compare treatments or interventions in randomized experiments or evaluation of policy interventions. The ATE measures the difference in mean (average) outcomes between units assigned to treatment and units assigned to the control.

The literature on treatment effects relies on building a **counterfactual**, such that each individual has an outcome with and without treatment. Let Y_1 denote the outcome with treatment and Y_0 denote the outcome without treatment. Because the individuals cannot be in the two states, we cannot observe both Y_0 and Y_1 ; in fact, the problem we face is one of missing data. Let D be the binary treatment indicator, where $D=1$ denotes treatment and $D=0$ be otherwise, and let X denote a vector of observed individual characteristics used as conditioning variables.

The indicator of interest is then given by:

$$ATE \equiv E(Y_1 - Y_0 | x) \quad (1)$$

ATE is the expected effect of treatment on a randomly drawn from the population.

Another treatment effect of interest is the **average treatment effect on the treated (ATT)**:

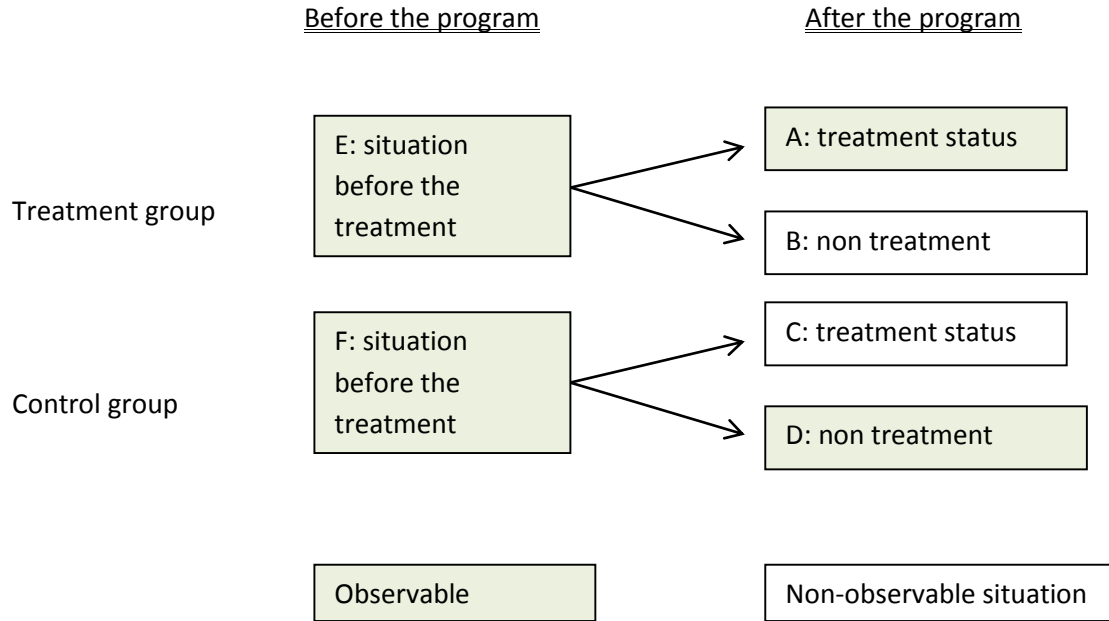
$$ATT \equiv E(Y_1 - Y_0 | x, D = 1) \quad (2)$$

which estimates the average impact of the program of those participating in it.

The missing data problem is the main issue in any evaluation exercise. In equation (2), Y_0 is not observed for program participants. Experimental designs provide direct evidence on the ATT parameter: data on program participants identifies the mean outcome in the treated state, $E(Y_1 | X, D = 1)$, while the randomized-out control group provides a direct estimate of $E(Y_0 | X, D = 1)$. In nonexperimental designs, in contrast, data on participants prior to entering the program or data on a comparison group who did not apply for the program are used as proxies for the missing counterfactual. **Selection bias** may arise when approximating this desired counterfactual outcome.

Regression methods

When having cross-section or panel data we can estimate the average treatment effects using traditional estimators. Those estimators are the difference-in-difference (also known as diff-in-diff), before-after and cross-section estimators. The following illustration may help to understand which treatment or non-treatment situations we can observe and which ones are necessary to use when implementing these estimators. Note that the letters A through F represent the sample averages for the individuals in each specific situation.



Difference-in-difference

The diff-in-diff estimator results from the difference in the before-after differences of the outcomes of both program participants and nonparticipants (treated versus non-treated). It uses both pre- and post-program data. This estimator accounts for economy-wide effects but is still sensitive to the choice of the baseline period (pre-program data). Following the figure above, the diff-in-diff estimator is defined as (A-E)-(D-F).

In a regression setup, the equation for examining the impact policy change is equal to

$$y = \beta_0 + \beta_1 t + \beta_2 D + \beta_3 (t * D) + error \quad (3)$$

where t denotes a dummy variable for the post-policy change period and D equals to one for those in the treatment group and zero otherwise.. The **parameter of interest** is β_3 that is a dummy variable equal to unity for those observations in the treatment group in the post-policy change period.

Before-After estimator

The before-after estimator uses pre-program data of the treated group to impute counterfactual outcomes for program participants. This estimator, however, ignores economy-wide effects and is sensitive to the choice of the baseline period¹. Following the figure above, the before-after estimator is defined as (A-E). In equation (3), the **parameters of interest** are β_1 and β_3 .

Cross-section estimator

¹ When evaluating training programs the choice of the baseline period is very important. There is an empirical regularity called Ashenfelter's dip that reflects the fact that the mean earnings of program participants decline during the period just prior to participation. The most likely explanation is that the trainees had a bad year (e.g. they lost their job) and this is what causes them to enter the training.

The cross-section estimator only uses data from the post-program period. The outcomes of the comparison group ($D=0$) are used to impute the counterfactual outcomes for the treated group ($D=1$). This estimator assumes that the economy-wide effects have the same impact on both the treated and the non-treated. Following the figure above, the cross section estimator is defined as (A-D). In equation (3), the **parameters of interest** are β_2 and β_3 .

Conventional estimators have been widely applied to conditioning on eligibility status (i.e. X defined as a vector of variables that determine eligibility), but as indicated by some authors (e.g., Heckman and Smith, The Economic Journal 1999), this may result in a comparison group that does not represent the desired counterfactual outcome and could lead to substantial bias when evaluating a program. They stress the importance of exploiting information regarding the factors that drive program participation in order to better capture the underlying choices leading to differences in unobserved variables between participants and nonparticipants.

EXAMPLE 1 (Length of Time on Workers' Compensation): The data on INJURY.RAW contains cross sectional individual data on 7150 individuals. Meyer, Viscusi and Durbin (MVD) (1995) study the length of time (weeks) that an injured worker receives worker's compensation. On July 15, 1980, Kentucky raised the cap on weekly earnings that were covered by worker's compensation. An increase in the cap has no effect on the benefit for low-income workers, but it makes it less costly for a high-income worker to stay in worker's compensation. Therefore, the control group is low-income workers and the treatment group is the high-income workers. Using random samples, MVD are able to test whether more generous workers' compensation causes people to stay out of work longer. The dummy variable for observations after the policy change is *afchnge* and *highearn* is the dummy variable for high earners. To implement the difference-in-difference approach, the following equation is estimated:

$$\log(\text{durat}) = \beta_0 + \beta_1 \text{afchnge} + \beta_2 \text{highearn} + \beta_3 (\text{afchnge} * \text{highearn}) \quad (4)$$

```
. use http://fmwww.bc.edu/ec-p/data/wooldridge/injury
. regress ldurat afchnge highearn afhigh
```

Source	SS	df	MS			
Model	193.919839	3	64.6399463	Number of obs =	7150	
Residual	12047.1903	7146	1.68586486	F(3, 7146) =	38.34	
Total	12241.1101	7149	1.71228285	Prob > F =	0.0000	
				R-squared =	0.0158	
				Adj R-squared =	0.0154	
				Root MSE =	1.2984	

ldurat	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
afchnge	.0236351	.0397008	0.60	0.552	-.0541902	.1014604
highearn	.2151955	.0433612	4.96	0.000	.1301948	.3001962
afhigh	.1883498	.062794	3.00	0.003	.065255	.3114445
_cons	1.199336	.0271091	44.24	0.000	1.146194	1.252478

$\hat{\beta}_3=0.1888$ which implies that the average duration of workers' compensation increased by 18.88% percent due to higher earnings cap. The coefficient on *afchnge* is small and statistically insignificant: as is expected, the increase in the earnings cap had no effect on duration for low-earnings workers. The coefficient on *highearn* shows that, even in the absence of any change in earnings cap, high earners spent much more time, 21.51% on worker's compensation.

Propensity score matching

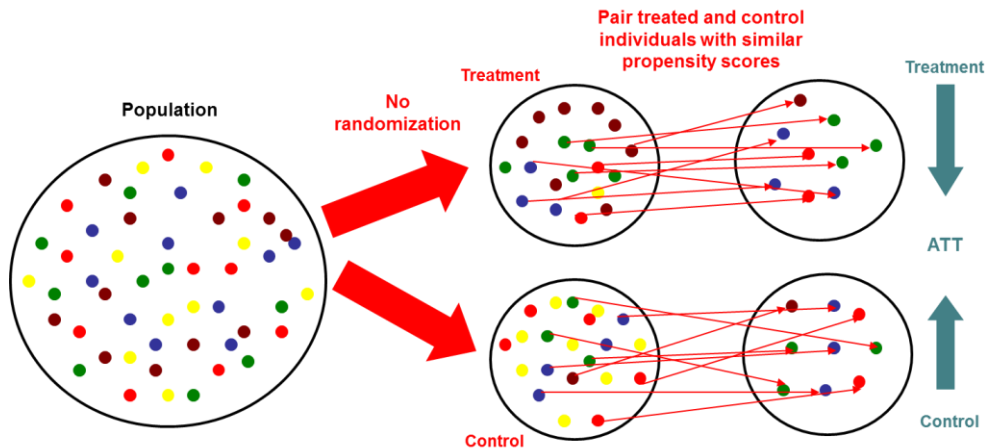
In observational studies, by definition there are no experimental controls. Therefore, there is no direct counterpart of the ATT calculated as a mean difference between the outcomes of the treated and non-treated groups. In other words, the counterfactual is not identified. As a substitute we may obtain data from a set of potential comparison units that are not necessarily drawn from the same population as the treated units, but for whom the observable characteristics, **X**, match to those of the treated units up to some selected degree of closeness.

The average outcome for the untreated matched group identifies the mean counterfactual outcome for the treated group in the absence of the treatment. This approach solves the evaluation problem by assuming that selection is unrelated to the untreated outcome, conditional on **X**. To make this approach operational it is necessary to define the matching criteria.

The **propensity score** is the conditional probability of receiving treatment given **X**, denoted by $p(X)$

$$\Pr(D=1|X)=p(X) \tag{5}$$

where **X** is a vector of observable characteristics that drive program participation.



Propensity score matching therefore avoids the “curse of dimensionality” associated to trying to match participants and non-participants on every possible characteristic when **X** is very large. The key to PSM is to identify the factors determining program participation.

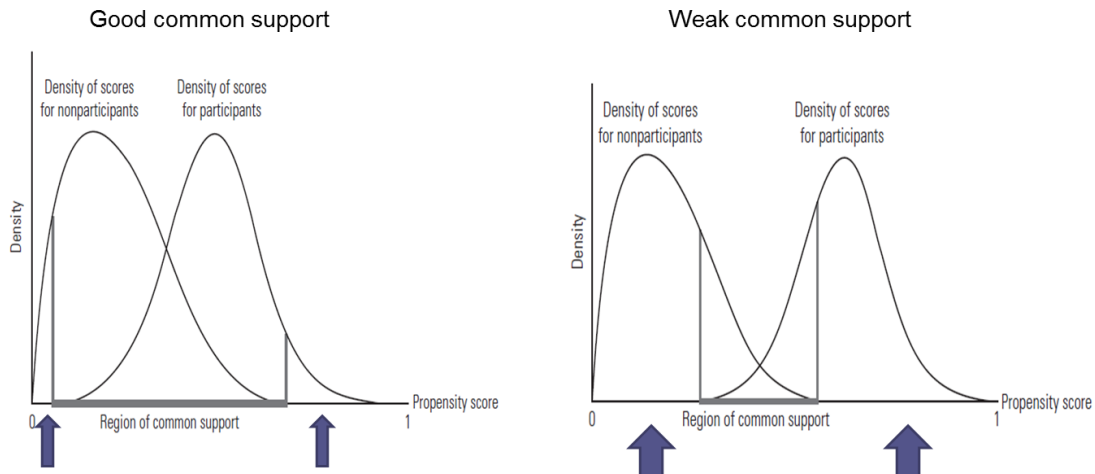
By correctly accounting for factors that drive program participation, potential unobserved differences between participants and non-participants are likely to be reduced (reduce selection bias). Also, it is recommended to have a reasonable amount of non-participants to match with “alike” participants.

Steps to apply propensity score matching:

1. **Estimate a model of program participation:** pool the samples of participants and non-participants and estimate a model of program participation (**D**) as a function of all variables **X** in the data that are likely to determine participation $P(D=1|X)=g(X\beta)$ where $g(\cdot)$ is a distribution function and β are the set of parameters to be estimated. After the participation equation is estimated, predicted values of the

probability of participation can be derived. These predicted probabilities are the propensity scores. These scores can be estimated by Probit, Logit, Linear Probability or semi-parametric (Single Index) models.

2. **Defining the region of common support and balance the sets:** The region of common support needs to be defined where distributions of the propensity score for the treatment and comparison group overlap. Some of the nonparticipants may have to be excluded because they have a propensity score outside the range found for the treatment sample (typically too low); the same for participants (typically too high).



3. **Matching participants to non-participants:** Different criteria can be used to assign participants to nonparticipants on the basis of the estimated propensity scores. The choice of the particular matching technique (algorithm) may affect the estimated impacts through the weights assigned. PSM algorithms differ on the weights they assign to members in the comparison group. There is a tradeoff between bias and variance when matching with and without replacement. Matching with replacement increases the average quality of the matches but increases the variance of the estimator by reducing the number of distinct nonparticipant observations.

The different **matching algorithms** are:

1. **Nearest neighbor (NN) matching:** each treatment unit is matched to the comparison unit with the closest propensity score. You can also use the five or ten closest comparison units. When using more comparison units you also trade reduced variance (resulting from using more information to construct the counterfactual for each participant) for increased bias (resulting from using, on average, poorer matches).
2. **Caliper or radius matching:** since the distance between participant i and comparison j can be substantial, a caliper can be introduced that imposes a tolerance of maximum distance allowed $||p_i - p_j||$. It may be difficult to know a priori which tolerance level is reasonable.
3. **Stratification or interval matching:** This procedure partitions the common support into different intervals and calculates the program's impact within each interval (mean difference in outcomes).
4. **Kernel and local linear matching:** These are nonparametric matching estimators that use a weighted average of all nonparticipants to construct the counterfactual match for each participant. The weight is based on the distance between the controls and the treatment ("closest" controls are given a higher weight).

5. **Conditional diff-in-diff matching:** Assumes that there are systematic differences between participant and nonparticipant outcomes even after conditioning on the probability of participation. So differencing the outcomes between the pre- and post-program period eliminates the bias component.
6. **Biased-corrected matching:** It is an alternative NN matching estimator that uses multi-dimensional covariates (instead of a propensity score). Adjusts for the bias resulting from discrepancies in the covariates between the matched individuals and their matches. Based on Abadie and Imbens (NBER 2002).
7. **“Hybrid method”:** Consists in matching each treated observation with a member of the comparison group using an explicit algorithm that controls, for example, for same gender, education level and location. For example, we do not want to end up matching two individuals living in different cities. So after controlling for similarity in certain characteristics (categorical variables), we match based on proximity in terms of the propensity score.

EXAMPLE 2 (Effects of Job Training on Earnings): The data in JTRAIN2.RAW is from a job training experiment in the 1970’s. The response variable is real earnings in 1978, measured in thousands of dollars. Real earnings are zero for men who did not work during the year. Training began two years prior to 1978. The elements on X are earnings in 1974 (*re74*) and 1975 (*re75*), age in quadratic form (*agesq*), a binary high school indicator (*nodegree*), marital status (*married*) and binary variables for black and Hispanic. We will introduce an example on how to implement propensity score matching in Stata using the *pscore* command².

The model on program participation in this case:

$$train = \beta_0 + \beta_1 agesq + \beta_2 nodegree + \beta_3 married + \beta_4 black + \beta_5 hisp + \beta_6 re74 + \beta_7 re75 \quad (6)$$

Once estimated with *pscore*, the propensity scores are directly derivated and saved.³

² Another important command for implementing propensity score matching in Stata is *psmatch2*. In Stata13 the teffects *psmatch* command was introduced. The teffects *psmatch* command has an advantage over *psmatch2* in that it takes into account the fact that propensity scores are estimated rather than known when calculating standard errors.

³ Note that the default estimation is done using a probit model

```
. use http://fmwww.bc.edu/ec-p/data/wooldridge/jtrain2
. pscore train agesq nodegree married black hisp re74 re75 , pscore(p)
```

```
*****
Algorithm to estimate the propensity score
*****
```

The treatment is train

=1 if assigned to job training	Freq.	Percent	Cum.
0	260	58.43	58.43
1	185	41.57	100.00
Total	445	100.00	

Estimation of the propensity score

```
Iteration 0: log likelihood = -302.1
Iteration 1: log likelihood = -294.07647
Iteration 2: log likelihood = -294.06753
Iteration 3: log likelihood = -294.06753
```

```
Probit regression                               Number of obs =      445
                                                LR chi2(7)         =      16.06
                                                Prob > chi2        =      0.0245
Log likelihood = -294.06753                    Pseudo R2         =      0.0266
```

train	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
agesq	.0000631	.0001449	0.44	0.663	-.0002209 .0003471
nodegree	-.4416419	.1485242	-2.97	0.003	-.7327439 -.1505398
married	.0911747	.169307	0.54	0.590	-.240661 .4230103
black	-.1446176	.2271606	-0.64	0.524	-.5898443 .3006091
hisp	-.5002755	.307429	-1.63	0.104	-1.102825 .1022743
re74	-.0189585	.0159389	-1.19	0.234	-.0501982 .0122812
re75	.0371704	.027059	1.37	0.170	-.0158642 .090205
_cons	.2205207	.253164	0.87	0.384	-.2756715 .7167129

Description of the estimated propensity score

Estimated propensity score				
Percentiles	Smallest			
1%	.2392116	.1638605		
5%	.2482821	.1997582		
10%	.3368791	.2258122	Obs	445
25%	.3658126	.2387806	Sum of Wgt.	445
50%	.3861172		Mean	.4155312
		Largest	Std. Dev.	.0934451
75%	.4611962	.6473616		
90%	.5542987	.6481746	Variance	.008732
95%	.5879249	.6690131	Skewness	.5058576
99%	.6350283	.6739108	Kurtosis	2.973849

```
*****
Step 1: Identification of the optimal number of blocks
Use option detail if you want more detailed output
*****
```

The final number of blocks is 4

This number of blocks ensures that the mean propensity score is not different for treated and controls in each blocks

```
*****
Step 2: Test of balancing property of the propensity score
Use option detail if you want more detailed output
*****
```

The balancing property is satisfied

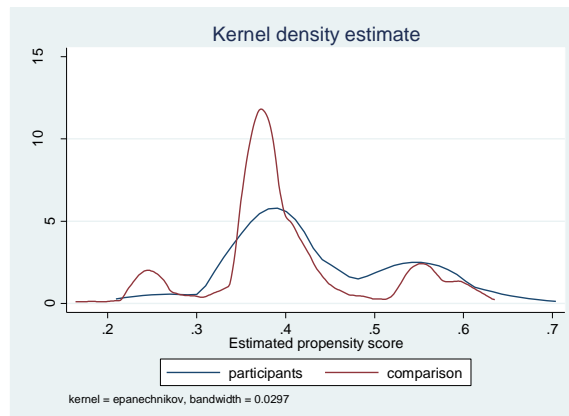
This table shows the inferior bound, the number of treated and the number of controls for each block

Inferior of block of pscore	=1 if assigned to job training		Total
	0	1	
0	2	0	2
.2	165	85	250
.4	85	90	175
.6	8	10	18
Total	260	185	445

```
*****
End of the algorithm to estimate the pscore
*****
```

Univariate kernel density estimations of the estimated probability of participation $P(Z)$ reveal that the common support for both participants ($D=1$) and nonparticipants ($D=0$) is not small:

kdensity p if (train==1), plot(kdensity p if (train==0)) legend (label (1 "participants") label (2 "comparison"))



Finally, we must choose the matching algorithm to calculate the ATT. In this case we present here two different matching algorithms: nearest neighbor matching and radius matching.

```
. attnd re78 train , pscore(p)
```

The program is searching the nearest neighbor of each treated unit.
 This operation may take a while.

ATT estimation with Nearest Neighbor Matching method
 (random draw version)
 Analytical standard errors

n. treat.	n. contr.	ATT	Std. Err.	t
185	172	2.416	0.746	3.238

Note: the numbers of treated and controls refer to actual nearest neighbour matches

```
. attr re78 train , pscore(p)
```

The program is searching for matches of treated units within radius.
 This operation may take a while.

ATT estimation with the Radius Matching method
 Analytical standard errors

n. treat.	n. contr.	ATT	Std. Err.	t
185	260	1.889	0.682	2.769

Note: the numbers of treated and controls refer to actual matches within radius

When using the nearest neighbor matching algorithm results suggest that job training is estimated to increase earnings by about US\$ 2416. This number is very high when compared with results obtained when using a radius matching estimator: US\$ 1889.

EXERCISE 1

- a) Write the observed outcome Y in terms of Y_1 , Y_0 and D , where D is the job training (treatment) indicator and $Y_1(Y_0)$ are the outcomes with (without) job training.
- b) Explain the meaning of $E(Y_0 | D = 1) < E(Y_0 | D = 0)$.

EXERCISE 2 (Manual implementation in Stata of propensity score matching)

Use the data in JTRAIN2.RAW (use <http://fmwww.bc.edu/ec-p/data/wooldridge/jtrain2>) for this question.

- a) Run a logit on equation (6) and execute the *predict p2* post estimation command, what do we obtain?
- b) Execute the command line *pscore train black hisp married nodegree re74 re75 agesq, pscore(p) logit*. Compare *p2* and *p*. What do you conclude?
- c) Obtain an estimator of program evaluation based on an OLS regression by regressing *re78* on a constant, D and the propensity score obtained in a).
- d) Obtain an estimator of program evaluation based on an OLS regression by regressing *re78* on a constant, D and *black hisp married nodegree re74 re75 agesq*. Comment on the results in c) and d).