ECON42720 Causal Inference and Policy Evaluation

4b Matching and Re-weighting

Ben Elsner (UCD)

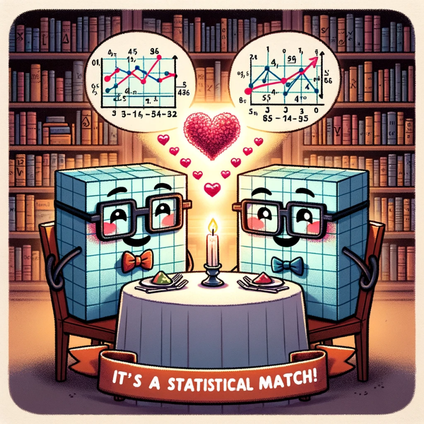
Approximate Matching

In most cases, we **cannot find a perfect match** for each treated unit

- Many **variables are continuous**
- We have many **covariates**
- . . . and **finite samples**

Approximate matching allows us to **match similar units**

Approximate Matching: Questions to Ask Yourself



1: Which **distance measure** to use?

- These determine how we measure similarity

2: How to turn **distance into matches**

- Which matches are "good enough"?
- Unique or multiple matches?
- Cut-off points (calipers), number of neighbours?

3: How do we **prune the data**?

- What do we do with units that are not matched?

4: Match **with or without replacement**?

- Do we allow control units to be matched to multiple treated units?

Approximate Matching

There are two **main methods for approximate matching:**

1. **Distance Matching** → minimise distance in X
2. **Propensity Score Matching** → match on likelihood of being treated

A third type of matching is **coarsened exact matching** (CEM)

It is also possible to **combine matching methods**

- Example: match exactly on some characteristics and approximately on others

Distance Matching: Questions to Ask

How do we measure **distance, i.e. the similarity** between treated and control units?

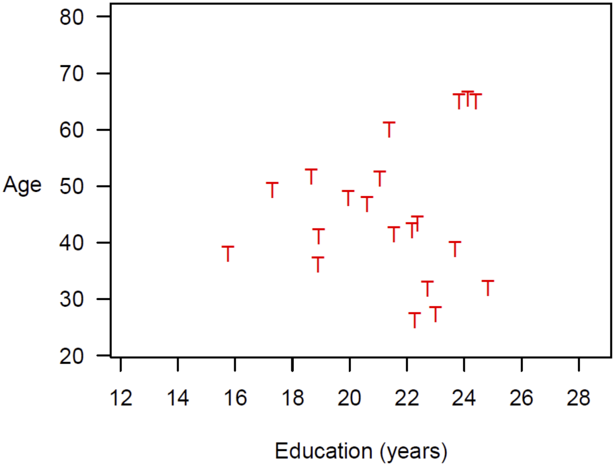
What is the **cut-off point** for a good match?

Do we consider **multiple matches** for each treated unit?

- If so, what criterion determines which unit is a match?
- And should **each control unit get the same weight**?

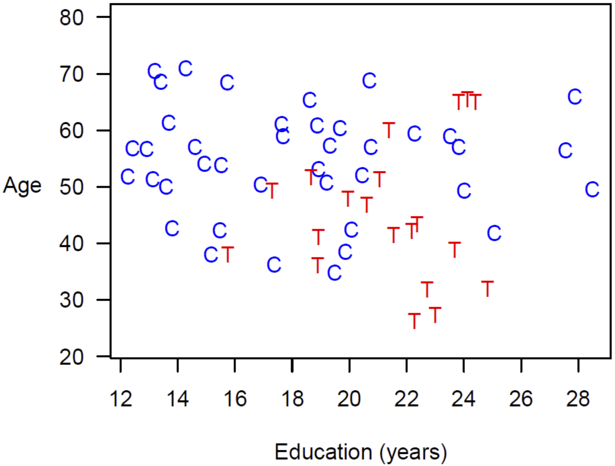
Distance Matching: Nearest Neighbour

Starting point: **treated units, with covariates age and age at which they left education**



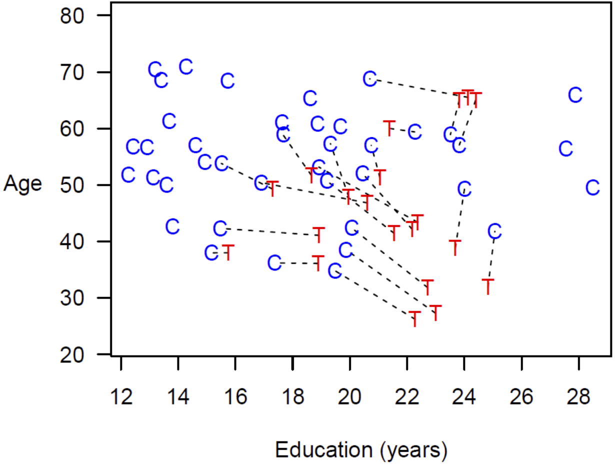
Distance Matching: Nearest Neighbour

Treated and control units are different w.r.t. education



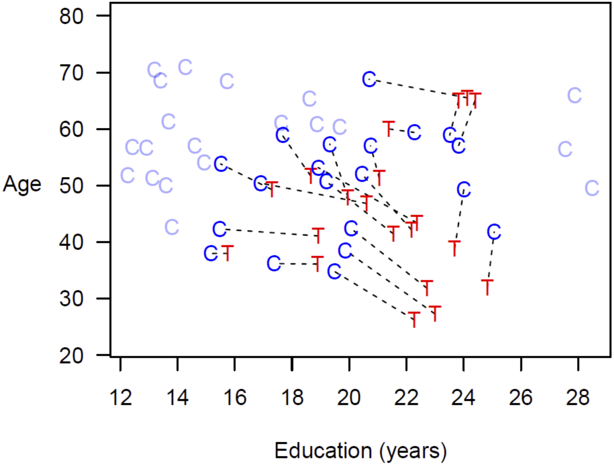
Distance Matching: Nearest Neighbour

For each treated unit, we **find the "closest" control unit** in terms of X (*Euclidean Distance*)



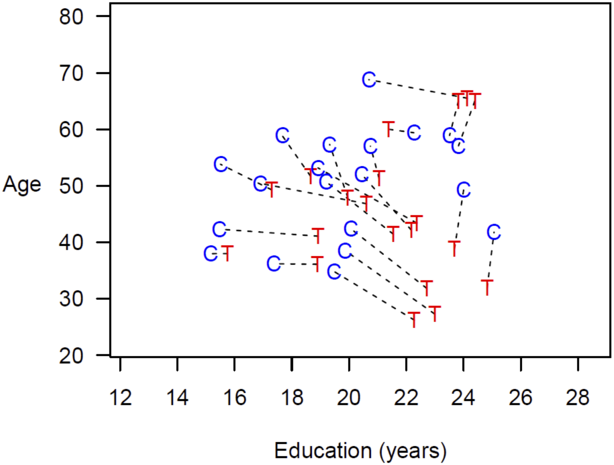
Distance Matching: Nearest Neighbour

Drop control units that are **not close enough** to any treated unit



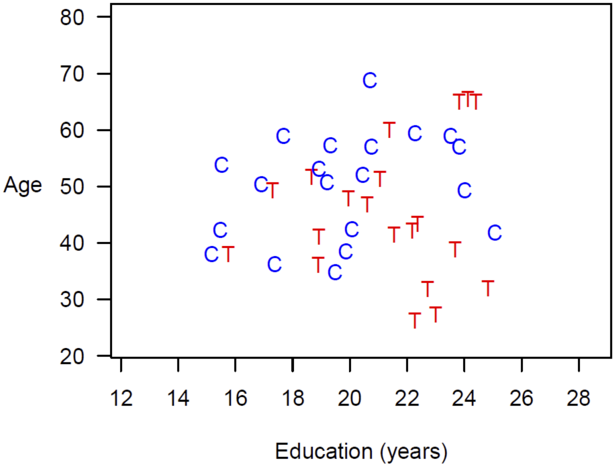
Distance Matching: Nearest Neighbour

Drop control units that are **not close enough** to any treated unit



Distance Matching: Nearest Neighbour

Our estimation sample:



Distance Matching: Nearest Neighbour

With **one covariate**, the distance is the Euclidean Distance

$$\begin{aligned} ||X_i - X_j|| &= \sqrt{(X_i - X_j)'(X_i - X_j)} \\ &= \sqrt{\sum_{n=1}^{k} (X_{ni} - X_{nj})^2} \end{aligned}$$

For each treated unit, we find the control unit with the **smallest distance** $||X_i - X_j||$

Multiple Covariates: Mahalanobis Distance

With **multiple covariates** $1, \ldots, k$, we take into account the variance-covariance matrix $\widehat{\Sigma}_X$ of the covariates

$$||X_i - X_j|| = \sqrt{(X_i - X_j)'\widehat{\Sigma}_X^{-1}(X_i - X_j)}$$

As before, for each treated unit, we find the control unit with the **smallest distance** $||X_i - X_j||$

Purpose of weighting with $\widehat{\Sigma}_X^{-1}$:

- Covariates become **scale-invariant**
- All distances are **measured in terms of standard deviations**

Nearest Neighbour Matching: Steps Involved

1. **Preprocess (Matching)**

- **Calculate the Mahalanobis Distance** $||X_i - X_j|| = \sqrt{(X_i - X_j)'\widehat{\Sigma}_X^{-1}(X_i - X_j)}$
- **Match** each treated unit to the **nearest control unit**
- Prune control units if unused
- Prune matches if Distance>caliper (i.e. if they exceed a certain distance)

2. **Estimation**: calculate difference in means or run a regression

Other Distance Matching Methods

k-Nearest-neighbour Matching (NNM)

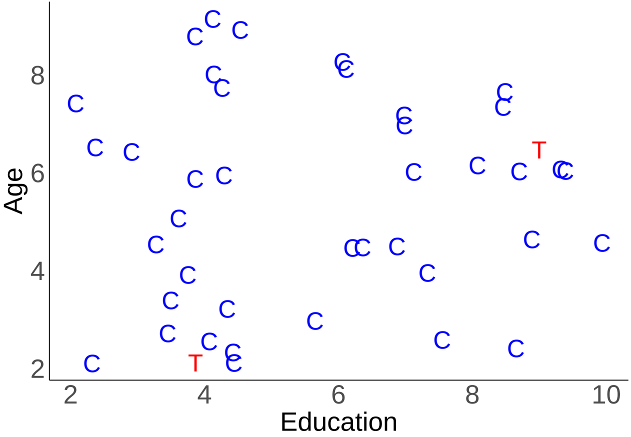
- Match with the nearest neighbour or the k **nearest neighbours** in terms of X
- Take the average of these neighbours as the counterfactual

Radius and Kernel Matching

- Match with **all control units** within a **certain radius of the treated unit**
- If all control units have equal weight, we call this **radius matching**
- If weights decay with distance, we call this **kernel matching**

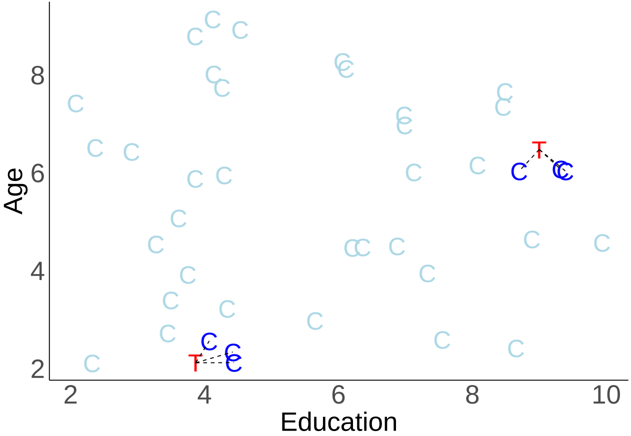
Nearest Neighbour Matching with $k = 3$

Suppose you have a dataset with 2 treated and many control units



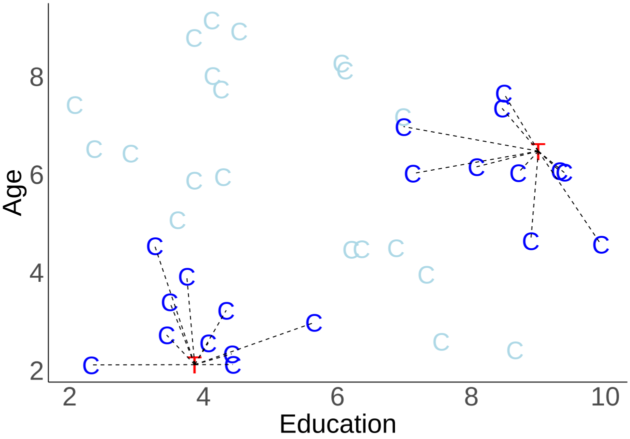
Nearest Neighbour Matching with $k = 3$

Now we select the three nearest neighbours for each treated unit; their average Y is the counterfactual for the treated unit



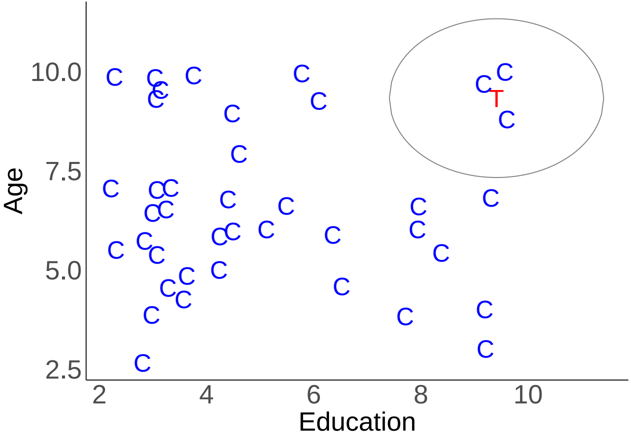
Nearest Neighbour Matching with $k = 10$

Now we select the ten nearest neighbours for each treated unit



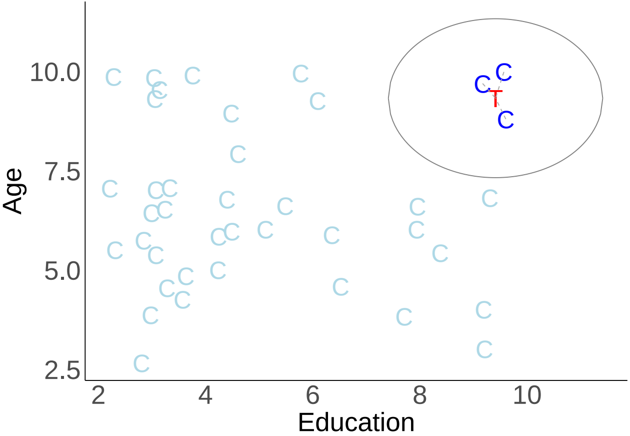
Radius Matching

Here the researcher specifies a **radius** ($r = 2$) around the treated unit

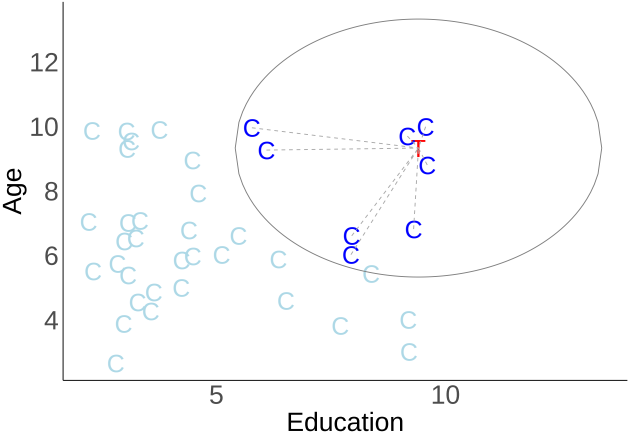


Radius Matching

Each **control unit within the radius has equal weight**

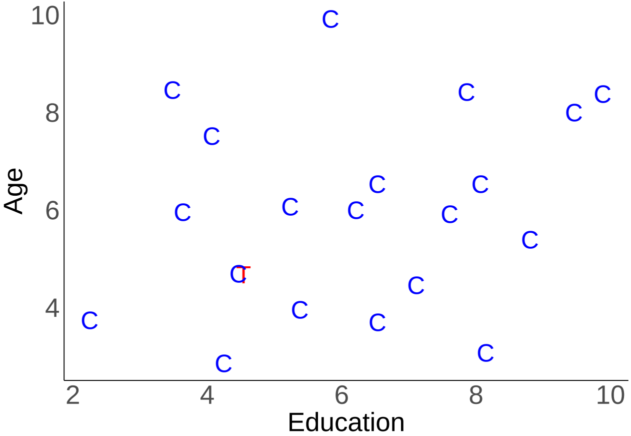


Radius Matching: Larger Radius ($r = 4$)



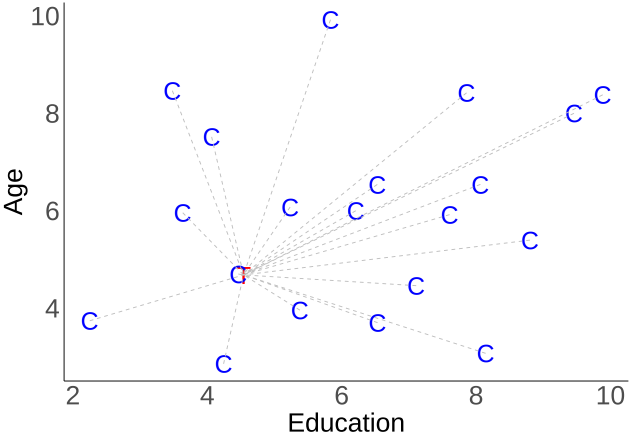
Kernel Matching

Consider the following sample



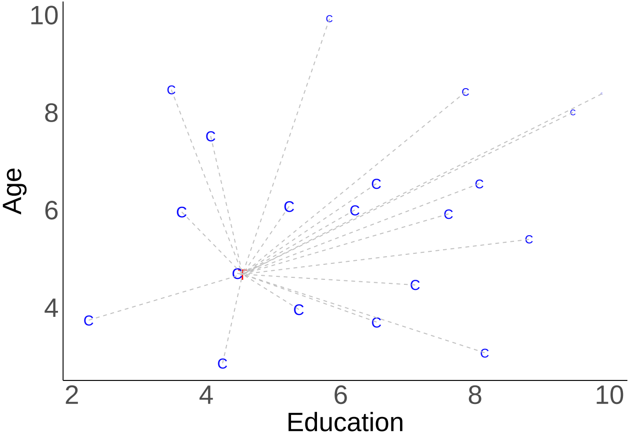
Kernel Matching

Now suppose each control unit gets equal weight



Kernel Matching

Now let's use an Epanechnikov Kernel: further away ⇒ smaller weight



Kernel Matching

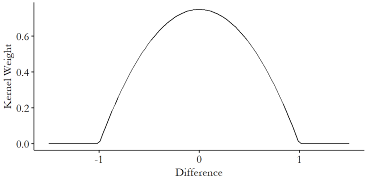
We want to create a weighted average by applying a kernel function

$$\bar{Y} = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} K(X_i) Y_i}{\sum_{i=1}^{n} K(X_i)}$$

There are many Kernel functions; they are typically concave and assign the highest weight to the smallest distance.

Example: Epanechnikov kernel

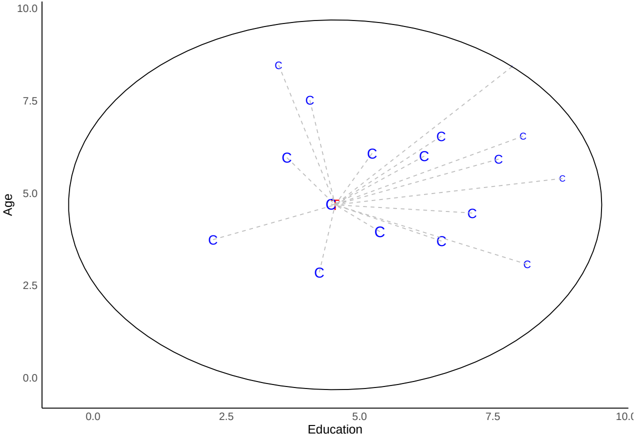
$$K(X) = \frac{3}{4}(1 - X^2)$$



K(X) is only defined between -1 and 1

Kernel Matching Within a Radius

We often use **kernel weighting within a radius** or the set of *k* nearest neighbours



More vs Better: the Bias-Variance Tradeoff

We often need to decide between unique matches and multiple control units

Researchers need to **solve a bias-variance trade-off**

Unique matches:

- Matches are precise but few → **low bias, high variance**

Weighted average of multiple control units

- Find many matches, but these are imprecise → **high bias, low variance**

Coarsened Exact Matching

Idea of CEM:

- Coarsen X (for example different age groups)
- Perform exact matching based on coarsened data

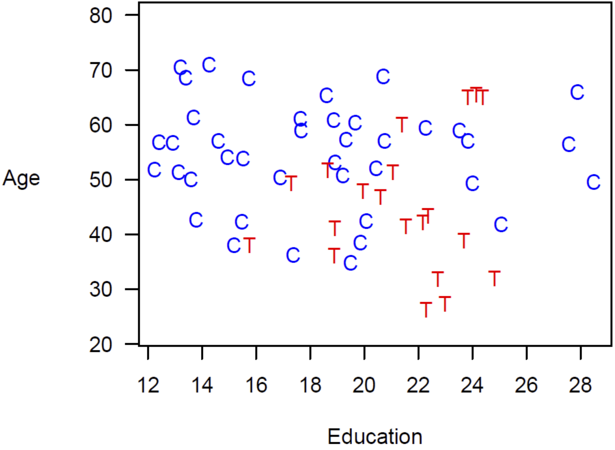
Advantage: **easy and fast**

Disadvantages:

- researcher **degrees of freedom** (categories are chosen by the researcher)
- curse of dimensionality (few categories: many but imprecise matches; many categories: few but more precise matches)

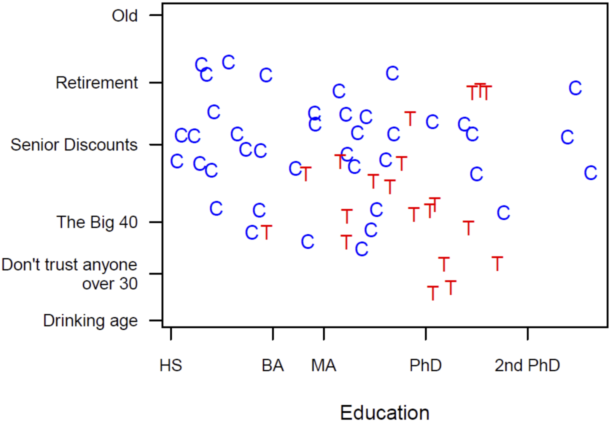
Coarsened Exact Matching

Starting point: same as before



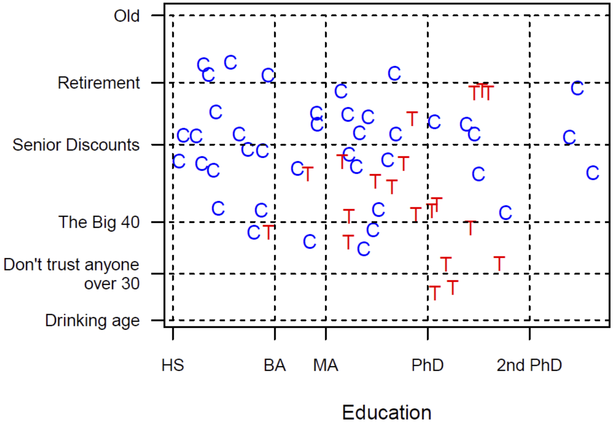
Coarsened exact matching

Coarsen: **divide variables into categories**



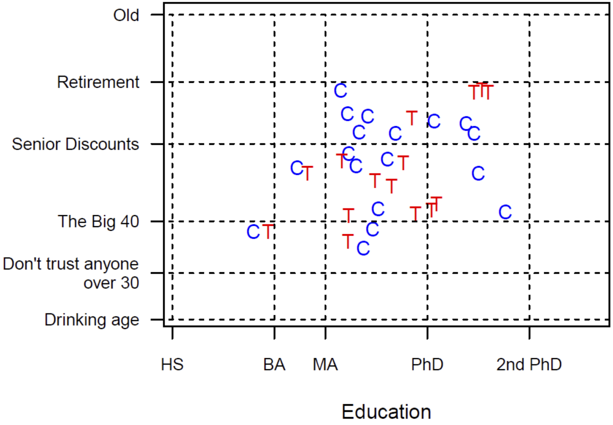
Coarsened exact matching

Now see **which cells contain treated and control units**



Coarsened exact matching

We **find matches within cells**



Coarsened exact matching

We **find matches within cells**

We need to **take a stand** regarding **matching within the cells**

- nearest neighbour or k nearest neighbours
- with or without replacement
- kernel distance function (usually not necessary)

Propensity Score Matching

Idea: predict the **probability that a given unit is treated** based on X

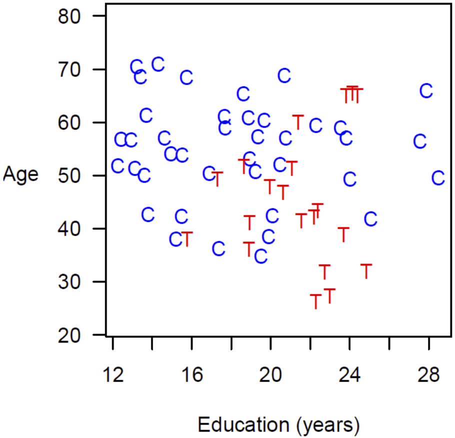
- The probability $Pr(D_i = 1|X)$ is called the **propensity score**

Match units with a similar probability of being treated (**propensity score**)

Estimate the ATT based on the matched dataset

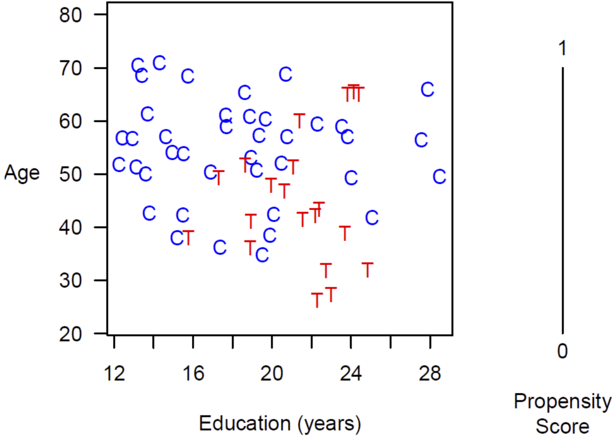
Propensity score matching

Starting point: treated and control units



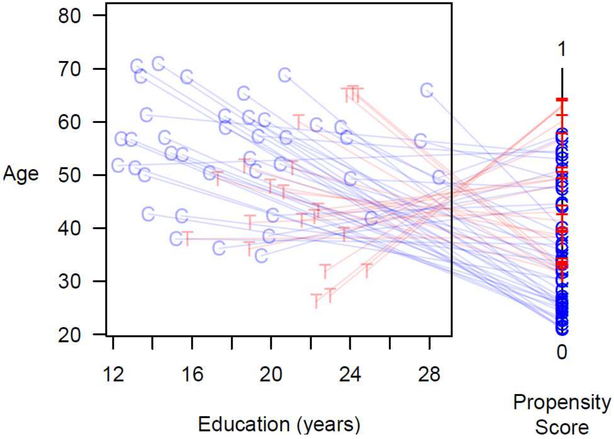
Propensity score matching

For each unit, we want to predict the probability that it is treated based on X



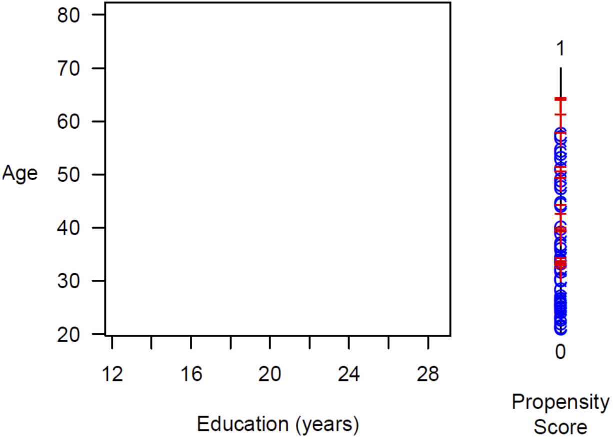
Propensity score matching

Let's do this: predict the probability of being treated



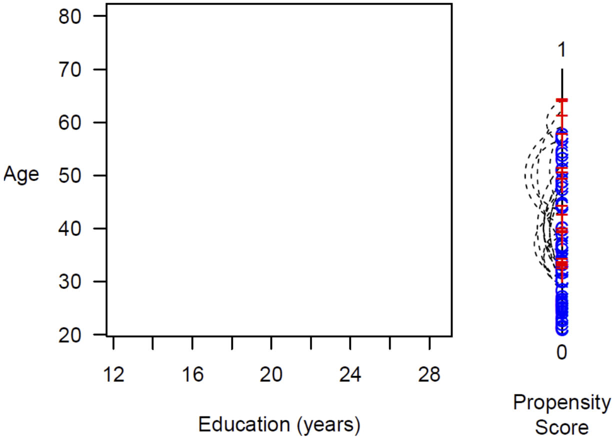
Propensity score matching

For each treated unit, find the control unit with the closest propensity score



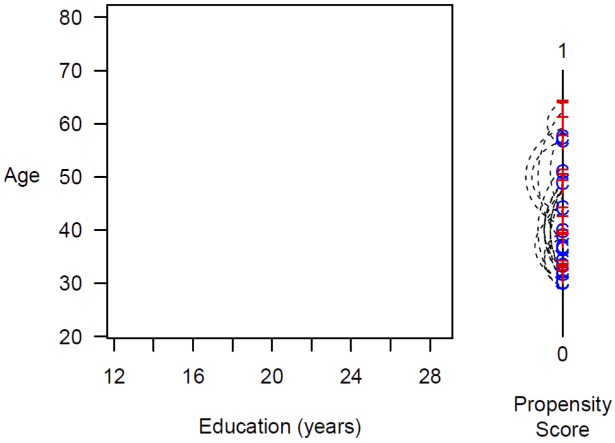
Propensity score matching

Prune control units that have not been matched



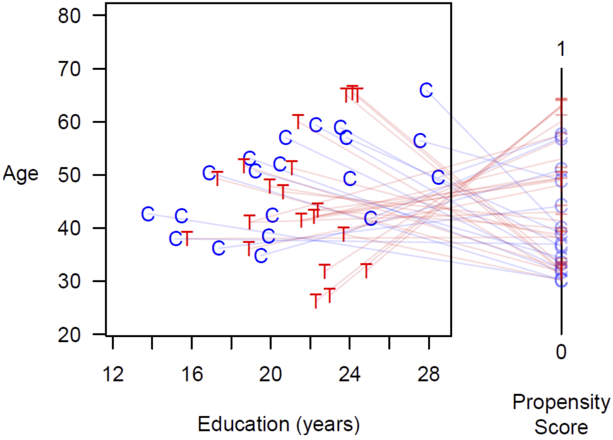
Propensity score matching

The result is your matched dataset



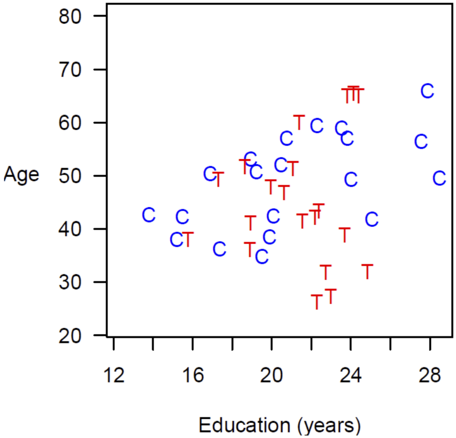
Propensity score matching

The result is your matched dataset



Propensity score matching

We can now regress the outcome on the treatment status



Estimation of the propensity score

The propensity score must satisfy the **balancing property**. It implies:

- Observations with the **same propensity score have the same distribution of observable covariates** independently of treatment status;
- For a given propensity score **assignment to treatment is random**, hence treated and control units are on average observationally identical.

Estimation of the propensity score

Can use any **standard probability model** to estimate the propensity score, e.g. a logit model:

$$Pr\{D_i = 1|X_i\} = \frac{e^{\lambda h(X_i)}}{1 + e^{\lambda h(X_i)}},$$

where $h(X_i)$ is a **function of covariates with linear and higher order terms.**

Inverse Probability Weighting (IPW)

IPW is based on the propensity score. Ingredients:

- The propensity score of being treated: $p(X)$
- The propensity score of beign untreated: $1 - p(X)$

Units are weighted by the **inverse propensity score** of THEIR treatment status

Why does this work?

- Weights "create" similar observations in terms of X
- Treated observations with similar X as untreated get a high weight because they are similar

Inverse Probability Weighting and the ATE

It can be shown that IPW identifies the ATE in the population:

$$\begin{aligned}\Delta = E\left[\mu_1(X) - \mu_0(X)\right] &= E\left[\frac{E[Y \cdot D \mid X] \cdot D}{p(X)} - \frac{E[Y \cdot (1-D) \mid X] \cdot (1-D)}{1-p(X)}\right] \\ &= E\left[\frac{Y \cdot D}{p(X)} - \frac{Y \cdot (1-D)}{1-p(X)}\right]\end{aligned}$$

Inverse Probability Weighting and the ATT

The ATT is identified by

$$\Delta_{D=1} = E\left[\frac{Y \cdot D}{\Pr(D=1)} - \frac{Y \cdot (1-D) \cdot p(X)}{(1-p(X)) \cdot \Pr(D=1)}\right]$$

IPW Weights Example

Observation	Treated	PS ($p(x)$)	weight
1	1	0.6	$\frac{1}{p(x)} = \frac{1}{0.6} = 1.67$
2	0	0.6	$\frac{1}{1-p(x)} = \frac{1}{0.4} = 2.5$
3	1	0.9	$\frac{1}{p(x)} = \frac{1}{0.9} = 1.11$
4	0	0.9	$\frac{1}{1-p(x)} = \frac{1}{0.1} = 10$

Here, observations 2 and 4 (untreated) get fairly large weights because they have similar X to typical treated units

Observation 3 (treated) gets a low weight because it is dissimilar to most untreated units

Matching and Causal Inference

Matching is NOT a causal identification strategy

- Neither is regression
- It is a **data reduction/pre-processing** technique

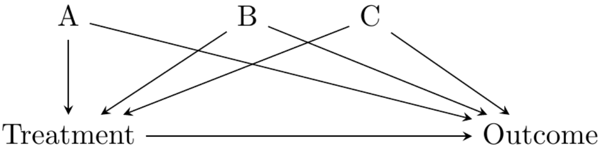
It helps us to achieve **balance on observables** X

Causal identification rests on the **conditional independence assumption**

- given X, D should be as good as randomly assigned
- i.e. X has to capture all confounders

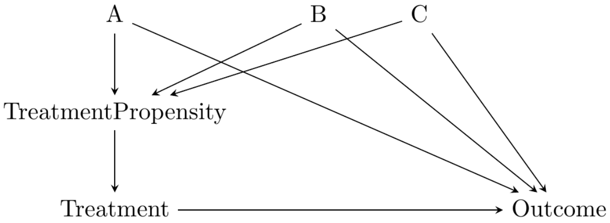
Matching and Causal Inference: a DAG

If this is the correct DAG, our **matching needs to account for A, B and C**



PSM and Causal Inference

With PSM, the additional assumption is that the **propensity score is correctly specified and closes all backdoor paths**



Matching Example: Broockman (2013)

Research question: are black politicians more likely to help black citizens even if the incentives are low?

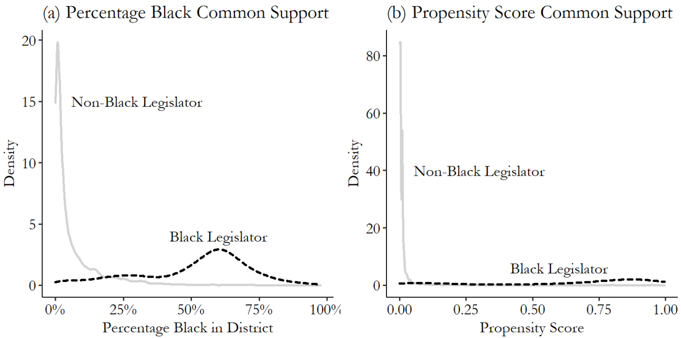
Methodology: audit study; sent emails to U.S. state legislators; asking them to help them sign up for unemployment benefits

Experimental variation:

- Sender with black vs. white name
- Sender lives in same district as legislator or far away

Matching: white and black legislators with similar characteristics

Step 1: Check for Common Support



Left: share of black voters in the district

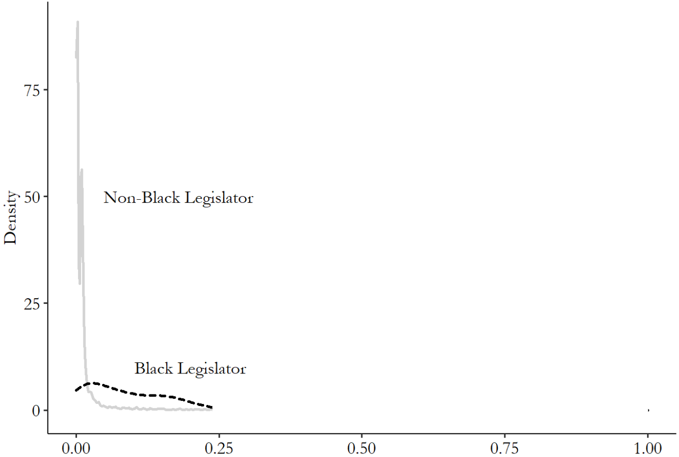
Right: distribution of propensity scores

- **propensity score** *p* (probability of being black) based on share of black voters, median household income, legislator is a democrat

Step 1: Common Support

Problem: areas with high p have no white legislators, areas with low p have no black legislators

- Solution: **prune areas without common support** (at least 10 control obs within a .02 bin)



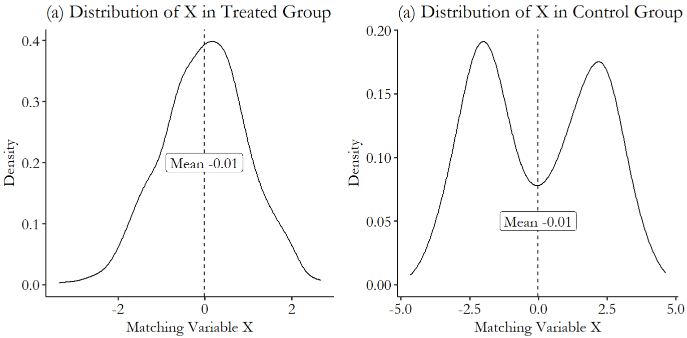
Balance in Broockman (2013)

Broockman performs Mahalanobis (nearest neighbour) matching; here is the balancing table

	Before Matching	After Matching
Median Household Income		
Mean Treatment	3.33	3.333
Mean Control	4.435	3.316
Std. Mean Diff	-97.057	1.455
t-test p-value	<.0001	0.164
Black Percent		
Mean Treatment	0.517	0.515
Mean Control	0.063	0.513
Std. Mean Diff	224.74	1.288
t-test p-value	<.0001	0.034
Legislator is a Democrat		
Mean Treatment	0.978	0.978
Mean Control	0.501	0.978
Std. Mean Diff	325.14	0
t-test p-value	<.0001	1

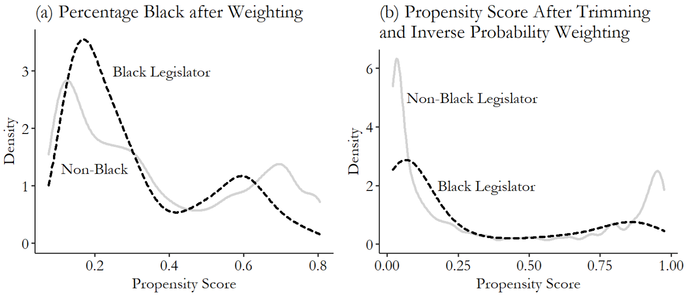
Careful when Performing Balancing Tests

Focusing just on mean differences can be deceptive. Consider these two distributions:



Always check the **full distribution** of covariates before and after matching

Full Distribution of X and p after IPW



Not perfect, but not so bad either. . .

Data preparation in R

We use the excellent `Matching` package in R. A great alternative is `MatchIt`

```r
library(Matching)
library(causaldata)
library(tidyverse)

br <- causaldata::black_politicians

# Outcome
Y <- br %>%
  pull(responded)
# Treatment
D <- br %>%
  pull(leg_black)
# Matching variables
# Note select() is also in the Matching package, so we specify dplyr
X <- br %>%
  dplyr::select(medianhhincom, blackpercent, leg_democrat) %>%
  as.matrix()
```

Mahalanobis distance matching in R

```r
# Set weight=2 for Mahalanobis distance
M <- Match(Y, D, X, Weight = 2, caliper = 1)

# See treatment effect estimate
summary(M)
```

```
##
## Estimate...  -0.0073462
## AI SE......  0.072683
## T-stat.....  -0.10107
## p.val......  0.91949
##
## Original number of observations..............  5593
## Original number of treated obs...............  364
## Matched number of observations...............  363
## Matched number of observations  (unweighted).  405
##
## Caliper (SDs)........................................   1 1 1
## Number of obs dropped by ’exact’ or ’caliper’  1
```

Mahalanobis distance matching in R

Previous slide: the estimate -0.007346 means that black legislators were 0.7 percentage points less likely to respond to emails

This effect is not statistically significant

Comparison with OLS

Table 2

	Dependent variable:	
	responded	
	(1)	(2)
leg_black	−0.032	−0.035
	(0.027)	(0.039)
medianhhincom		0.014***
		(0.005)
blackpercent		0.081
		(0.063)
leg_democrat		−0.039***
		(0.014)
Constant	0.425***	0.377***
	(0.007)	(0.025)
Observations	5,593	5,593

Comparison with OLS

The covariate plots showed that there is little common support

Matching rests on comparable observations with common support in X

OLS uses observations without common support; this explains the difference in the estimates

Matching and Re-Weighting: Conclusion

Matching and re-weighting are **data subsetting techniques**

- They help to **achieve balance on observables** X
- They are particularly useful when there is **little common support**

Matching and re-weighting are not causal identification strategies

- we need to rely on the **conditional independence assumption** to identify causal effects
- whether this holds depends on the context
- even the best matching procedure in the world cannot fix a bad research design

Matching and Re-Weighting: Conclusion

There are **many matching procedures**

- It is easy to get lost in the details
- The most important thing is to **achieve balance on observables** X
- When choosing a method, **keep the bias-variance tradeoff in mind**

Showing robustness to different matching procedures is very important

Matching in R

R has many packages for matching; most of them do similar things but have their strengths and weaknesses. Here are some very good ones:

- `Matching`
- `MatchIt`
- `cem` for Coarsened Exact Matching
- `optmatch`

`MatchIt` covers all the bases

References

Broockman, David E. 2013. Black Politicians Are More Intrinsically Motivated to Advance Blacks' Interests: A Field Experiment Manipulating Political Incentives. *American Journal of Political Science*, **57**(3), 521–536.

APPENDIX

benjamin.elsner@ucd.ie

www.benjaminelsner.com

Sign up for office hours

YouTube Channel

@ben_elsner

LinkedIn

Contact

Prof. Benjamin Elsner
University College Dublin
School of Economics
Newman Building, Office G206
benjamin.elsner@ucd.ie

Office hours: book on Calendly