Matching Methods

Prof. Tzu-Ting Yang
楊子霆

Institute of Economics, Academia Sinica
中央研究院經濟研究所

March 26, 2026

Observational Studies

- Randomized Controlled Trial (RCT) is called the “gold standard” for causal inference
  - In a RCT, researcher can assign treatments randomly to the individuals
  - Therefore, **treatment status is unrelated to any observed and unobserved confounders**
    - ★ Treatment and control group should be similar in all characteristics
    - ★ Thus, we can use the observed outcomes of control group to approximate the counterfactual outcomes of treatment group

Observational Studies

- But implementing a randomized experiment in social science is very expensive and sometimes has ethical issues
- In social science, many empirical studies use **non-experimental data**
  - It means researchers **can NOT assign treatment**
- We call this type of empirical researches as **observational studies**

Observational Studies

- We want to **design** observational studies that **approximate experiments**:
  - “The planner of an observational study should always ask himself: How would the study be conducted if it were possible to do it by controlled experimentation” (Cochran 1965)
- How to do that?
  - Need to directly control for the observed variables
  - Use indirect methods to adjust for unobserved variables
  - Make “other thing equal” in observed and unobserved variables

Main Idea

Main Idea of Matching

- Assume all confounding factors are **observable** to researchers
- Matching is a way to eliminate selection bias
  - By constructing a control group with the same observable characteristics as the treatment group
- This can be accomplished by **matching** treated and untreated units with the same observable characteristics.

Main Idea of Matching

- **Example:**
  - We want to estimate the causal effect of job training program on worker's earnings
  - Suppose **age** is the only confounding factors that affect both earnings and job training decision
    - ★ Older workers are *less likely to seek job training*
    - ★ Older workers *also earn more* due to their experience

Matching: A Numerical Example

Trainees			Non-Trainees		
unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900
2	34	10200	2	50	31000
3	29	14400	3	30	21000
4	25	20800	4	27	9300
5	29	6100	5	54	41100
6	23	28600	6	48	29800
7	33	21900	7	39	42000
8	27	28800	8	28	8800
9	31	20300	9	24	25500
10	26	28100	10	33	15500
11	25	9400	11	26	400
12	27	14300	12	31	26600
13	29	12500	13	26	16500
14	24	19700	14	34	24200
15	25	10100	15	25	23300
16	43	10700	16	24	9700
17	28	11500	17	29	6200
18	27	10700	18	35	30200
19	28	16300	19	32	17800
			20	23	9500
			21	32	25900
Avg:	28.5	16426	Avg:	33	20724

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900			
2	34	10200	2	50	31000			
3	29	14400	3	30	21000			
4	25	20800	4	27	9300			
5	29	6100	5	54	41100			
6	23	28600	6	48	29800			
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900			
2	34	10200	2	50	31000			
3	29	14400	3	30	21000			
4	25	20800	4	27	9300			
5	29	6100	5	54	41100			
6	23	28600	6	48	29800			
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900			
2	34	10200	2	50	31000			
3	29	14400	3	30	21000			
4	25	20800	4	27	9300			
5	29	6100	5	54	41100			
6	23	28600	6	48	29800			
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000			
3	29	14400	3	30	21000			
4	25	20800	4	27	9300			
5	29	6100	5	54	41100			
6	23	28600	6	48	29800			
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000			
4	25	20800	4	27	9300			
5	29	6100	5	54	41100			
6	23	28600	6	48	29800			
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300			
5	29	6100	5	54	41100			
6	23	28600	6	48	29800			
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100			
6	23	28600	6	48	29800			
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800			
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000			
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800			
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500			
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500			
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400			
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600			
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500			
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200			
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300			
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300	15	25	23300
16	43	10700	16	24	9700			
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300	15	25	23300
16	43	10700	16	24	9700	1	43	20900
17	28	11500	17	29	6200			
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300	15	25	23300
16	43	10700	16	24	9700	1	43	20900
17	28	11500	17	29	6200	8	28	8800
18	27	10700	18	35	30200			
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300	15	25	23300
16	43	10700	16	24	9700	1	43	20900
17	28	11500	17	29	6200	8	28	8800
18	27	10700	18	35	30200	4	27	9300
19	28	16300	19	32	17800			
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

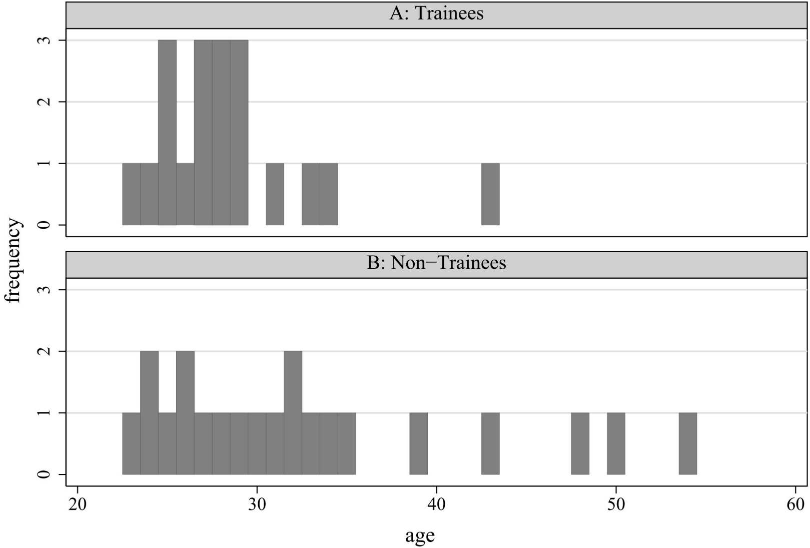
Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300	15	25	23300
16	43	10700	16	24	9700	1	43	20900
17	28	11500	17	29	6200	8	28	8800
18	27	10700	18	35	30200	4	27	9300
19	28	16300	19	32	17800	8	28	8800
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:		

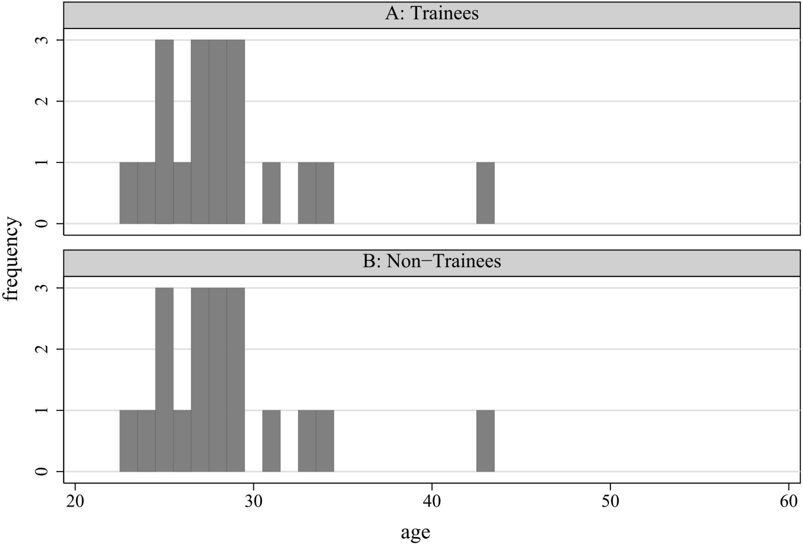
Matching: A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300	15	25	23300
16	43	10700	16	24	9700	1	43	20900
17	28	11500	17	29	6200	8	28	8800
18	27	10700	18	35	30200	4	27	9300
19	28	16300	19	32	17800	8	28	8800
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:	28.5	13982

Age Distribution: Before Matching



Age Distribution: After Matching



Treatment Effect Estimates

Difference in average earnings between trainees and non-trainees:

- Before matching:

$$16426 - 20724 = -4298$$

- After matching:

$$16426 - 13982 = 2444$$

Identification

Conditional Independence Assumption

Intuition

From the previous example, we know:

- A naive comparison mixes up the **effect of job training** with the **effect of age**.

Solution

Match by age: Once we compare people of the **same age**, the confounding effect of age disappears.

Conditional Independence Assumption

Formal Definition

Conditional Independence Assumption (CIA)

$$(\mathrm{Y}_i^1, \mathrm{Y}_i^0) \perp\!\!\!\perp D_i \mid X_i$$

"Given X_i, the potential outcomes are independent of treatment assignment."

- $(\mathrm{Y}_i^1, \mathrm{Y}_i^0)$: potential outcomes (with/without treatment)
- D_i: treatment indicator ($1 =$ treated, $0 =$ control)
- X_i: observable characteristics (covariates)
- Conditional on X_i, treatment assignment becomes **"as good as random"**
- This assumption is also called **selection on observables** —all confounding comes from observable X_i, not hidden factors

Conditional Independence Assumption

Counterfactual Logic

	Trained $(D = 1)$	**Not Trained** $(D = 0)$
Observed	$\mathrm{E}[Y_i^1 \mid X_i{=}40, D_i{=}1]$	$\mathrm{E}[Y_i^0 \mid X_i{=}40, D_i{=}0]$
Counterfactual	$\mathrm{E}[Y_i^0 \mid X_i{=}40, D_i{=}1]$	$\mathrm{E}[Y_i^1 \mid X_i{=}40, D_i{=}0]$

Under CIA, red cells (counterfactuals) can be filled in using the **observed outcomes of the other group**:

- $\mathrm{E}[Y_i^0 \mid X_i{=}40, D_i{=}1] = \mathrm{E}[Y_i^0 \mid X_i{=}40, D_i{=}0]$
  - What would **trainees** have earned *without* training?
    → Estimated by observed earnings of **non-trainees** of the same age
- $\mathrm{E}[Y_i^1 \mid X_i{=}40, D_i{=}0] = \mathrm{E}[Y_i^1 \mid X_i{=}40, D_i{=}1]$
  - What would **non-trainees** have earned *with* training?
    → Estimated by observed earnings of **trainees** of the same age

Conditional Independence Assumption (CIA)

Violation Example

- CIA requires **all** confounders to be observable. If a confounder is **unobserved**, CIA fails.
- Suppose **motivation** affects both training enrollment and wage potential, but is **unobserved**:
  - Highly motivated individuals are more likely to enroll in training
  - These same individuals would earn more regardless of training
  - The two groups are **no longer comparable** even at the same age:

$$\mathrm{E}[Y_i^0 \mid X_i{=}40, D_i{=}1] > \mathrm{E}[Y_i^0 \mid X_i{=}40, D_i{=}0]$$

Conditional Independence Assumption (CIA)

Violation Example

- Consequences of CIA violation:
  - The treatment effect estimate will be **biased upward**
  - We would attribute higher earnings to the training effect, when part of the difference is actually due to motivation
  - In this case, our causal estimate still has **selection bias**

Key Takeaway

CIA fails when there are **unobservable confounders** —even after controlling for X_i, the estimate still suffers from **selection bias**.

Common Support Assumption

Common Support Assumption

$$0 < \Pr(D_i = 1|X_i) < 1$$

- For each value of covariates X_i, there is a positive probability of being both treated and untreated
- In other words, it is NOT possible to perfectly predict one's treatment status by using specific value of X_i
  - For example, this excludes:
    - All individuals with age 40 are in treatment group: $Pr(D_i = 1|X_i = 40) = 1$
    - All individuals with age 40 are in control group: $Pr(D_i = 1|X_i = 40) = 0$
- It ensures sufficient overlap in characteristics of treated and untreated units to find adequate matched sample

Identification Results for Matching

- Under CIA and Common Support, matching can identify causal effects. We proceed in three steps:

Step 1 Show that ODO at given X_i equals **CATT** (selection bias $=0$)

Step 2 Under CIA, CATT $=$ CATU $=$ **CATE**

Step 3 Apply LIE to average CATE over $X \Rightarrow$ **ATT, ATU, ATE**

Identification Results for Matching

Step 1: ODO = CATT

$$
\begin{aligned}
&\underbrace{\mathrm{E}[\mathrm{Y}_i|X_i, D_i = 1] - \mathrm{E}[\mathrm{Y}_i|X_i, D_i = 0]}_{\text{ODO at given } X_i} \\
&= \mathrm{E}[\mathrm{Y}_i^1|X_i, D_i = 1] - \mathrm{E}[\mathrm{Y}_i^0|X_i, D_i = 0] \\
&= \mathrm{E}[\mathrm{Y}_i^1|X_i, D_i = 1] - \mathrm{E}[\mathrm{Y}_i^0|X_i, D_i = 1] \\
&\quad + \mathrm{E}[\mathrm{Y}_i^0|X_i, D_i = 1] - \mathrm{E}[\mathrm{Y}_i^0|X_i, D_i = 0] \\
&= \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|X_i, D_i = 1]}_{\text{CATT}} + \underbrace{\mathrm{E}[\mathrm{Y}_i^0|X_i, D_i = 1] - \mathrm{E}[\mathrm{Y}_i^0|X_i, D_i = 0]}_{\text{Selection Bias=0 by CIA}} \\
&= \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|X_i, D_i = 1]}_{\text{CATT}}
\end{aligned}
$$

Identification Results for Matching

Step 2: CATT $=$ CATU $=$ CATE

$$\underbrace{\mathrm{E}[\mathrm{Y}_i|X_i, D_i = 1] - \mathrm{E}[\mathrm{Y}_i|X_i, D_i = 0]}_{\text{ODO at given } X_i}$$

$$= \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|X_i, D_i = 1]}_{\text{CATT}} \quad \text{(from Step 1)}$$

$$= \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|X_i, D_i = 0]}_{\text{CATU}} \quad \text{(by CIA)}$$

$$= \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|X_i]}_{\text{CATE}}$$

- Under CIA, matching identifies the causal effect for **any given subgroup** $X_i = x$

Review: The Law of Iterated Expectations (LIE)

The Law of Iterated Expectations (LIE)

$$\mathrm{E}[\mathrm{Y}_i] = \mathrm{E}[\mathrm{E}[\mathrm{Y}_i|\mathrm{X}_i]]$$

- Two equivalent ways to compute average earnings in Taiwan:

1 Direct average: 40% earn 1M, 40% earn 2M, 20% earn 3M

$$\mathrm{E}[\mathrm{Y}_i] = 1 \times 0.4 + 2 \times 0.4 + 3 \times 0.2 = 1.8\mathrm{M}$$

2 Average of subgroup averages: male avg = 2M, female avg = 1.6M, each 50%

$$\mathrm{E}[\mathrm{E}[\mathrm{Y}_i|\mathrm{X}_i]] = 2 \times 0.5 + 1.6 \times 0.5 = 1.8\mathrm{M} = \mathrm{E}[\mathrm{Y}_i]$$

- Key insight: **average of subgroup averages = overall average**

Identification Results for Matching

Step 3: From CATT to ATT

- From Step 1, ODO at given X_i identifies CATT:

$$\underbrace{\mathrm{E}[\mathrm{Y}_i|X_i, D_i = 1] - \mathrm{E}[\mathrm{Y}_i|X_i, D_i = 0]}_{\text{ODO at given } X_i} = \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|X_i, D_i = 1]}_{\text{CATT}}$$

- Applying LIE, average CATT over the **treatment group** distribution of X:

$$\mathrm{E}\Big[\underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|X_i, D_i = 1]}_{\text{CATT}}\Big|D_i = 1\Big] = \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|D_i = 1]}_{\text{ATT}}$$

Identification Results for Matching: ATT

A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300	15	25	23300
16	43	10700	16	24	9700	1	43	20900
17	28	11500	17	29	6200	8	28	8800
18	27	10700	18	35	30200	4	27	9300
19	28	16300	19	32	17800	8	28	8800
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:	28.5	13982

Identification Results for Matching: ATT

A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	age	earnings	unit	age	earnings	unit	age	earnings
1	28	17700	1	43	20900	8	28	8800
2	34	10200	2	50	31000	14	34	24200
3	29	14400	3	30	21000	17	29	6200
4	25	20800	4	27	9300	15	25	23300
5	29	6100	5	54	41100	17	29	6200
6	23	28600	6	48	29800	20	23	9500
7	33	21900	7	39	42000	10	33	15500
8	27	28800	8	28	8800	4	27	9300
9	31	20300	9	24	25500	12	31	26600
10	26	28100	10	33	15500	11,13	26	8450
11	25	9400	11	26	400	15	25	23300
12	27	14300	12	31	26600	4	27	9300
13	29	12500	13	26	16500	17	29	6200
14	24	19700	14	34	24200	9,16	24	17700
15	25	10100	15	25	23300	15	25	23300
16	43	10700	16	24	9700	1	43	20900
17	28	11500	17	29	6200	8	28	8800
18	27	10700	18	35	30200	4	27	9300
19	28	16300	19	32	17800	8	28	8800
			20	23	9500			
			21	32	25900			
Avg:	28.5	16426	Avg:	33	20724	Avg:	28.5	13982

Identification Results for Matching: ATT

A Numerical Example (1/2)

- CATT for age = 28 and age = 34:

$$
\begin{aligned}
&\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | X_i = 28, D_i = 1] \\
&= \mathrm{E}[\mathrm{Y}_i | X_i = 28, D_i = 1] - \mathrm{E}[\mathrm{Y}_i | X_i = 28, D_i = 0] \\
&= \frac{(17700 - 8800) + (11500 - 8800) + (16300 - 8800)}{3} \\
&= 15166.67 - 8800 \\
&= 6{,}366.67
\end{aligned}
$$

$$
\begin{aligned}
&\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | X_i = 34, D_i = 1] \\
&= \mathrm{E}[\mathrm{Y}_i | X_i = 34, D_i = 1] - \mathrm{E}[\mathrm{Y}_i | X_i = 34, D_i = 0] \\
&= \frac{10200 - 24200}{1} \\
&= -14{,}000 \quad \cdots
\end{aligned}
$$

Identification Results for Matching: ATT

A Numerical Example (2/2)

- ATT = weighted average of CATT across all ages (weights = share in treatment group):

$$
\begin{aligned}
\mathrm{ATT} &= \mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | X_i = 28, D_i = 1] \times \frac{3}{19} \\
&\quad + \mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | X_i = 34, D_i = 1] \times \frac{1}{19} + \cdots \\
&= \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | D_i = 1]}_{\text{ATT}}
\end{aligned}
$$

Identification Results for Matching

Step 3: ATU and ATE

- **ATU**: average CATU over the **control group** distribution of X:

$$\mathrm{E}\Big[\underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | X_i, D_i = 0]}_{\text{CATU}} \Big| D_i = 0\Big] = \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | D_i = 0]}_{\text{ATU}}$$

- **ATE**: average CATE over the **full population** distribution of X:

$$\mathrm{E}\Big[\underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | X_i]}_{\text{CATE}}\Big] = \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0]}_{\text{ATE}}$$

- **Summary**: Under CIA, matching can identify ATT, ATU, and ATE by averaging X-specific effects over the appropriate population

Estimation

Matching Estimator

- Suppose our sample is N individuals
- Treatment is job training and outcome is earning
  - N_1 individuals choose to join job training: treatment group
  - N_0 individuals choose not join it ($N_0 = N - N_1$): control group

Matching Estimator

Estimation for ATT

- Suppose we want to estimate ATT
  - Average treatment effect for treatment group
- In that case, a matching estimator of α_{ATT} can be constructed as:

$$\hat{\alpha}_{\text{ATT}} = \frac{1}{N_1} \sum_{D_i=1} (Y_i - Y_{j(i)})$$

  - We want to match **treated** individual *i*'s outcome Y_i
    - ★ We impute Y_i^0 using untreated units $Y_{j(i)}$ in control group
    - ★ $Y_{j(i)}$: the outcome of an untreated observation j such that $X_{j(i)}$ is the **same** or **closest** value to X_i among the untreated observations.

Matching Estimator

Estimation for ATT

- We can also use the average:

$$\hat{\alpha}_{ATT} = \frac{1}{N_1} \sum_{D_i=1} \left\{ Y_i - \left(\frac{1}{M} \sum_{m=1}^{M} Y_{jm(i)} \right) \right\}$$

- Works well when we can find good matches for each treated unit, so M is usually small (typically, $M = 1$ or $M = 2$)

- Perfect matches are often not available

Matching Estimator

Estimation for ATU

- Suppose we want to estimate ATU
  - Average treatment effect for control group
- In that case, a matching estimator of α_{ATU} can be constructed as:

$$\hat{\alpha}_{\text{ATU}} = \frac{1}{N_0} \sum_{D_i=0} (Y_{j(i)} - Y_i)$$

  - We want to match **untreated** individual *i*'s outcome Y_i
    - ★ We impute Y_i^1 using treated units $Y_{j(i)}$ in treatment group
    - ★ $Y_{j(i)}$: the outcome of an treated observation j such that $X_{j(i)}$ is the **same** or **closest** value to X_i among the treated observations.

Matching

Estimation for ATE

- We can also use matching to estimate ATE
  - Average treatment effect for both treatment and control groups
- In that case, we match in both directions:
  1. If observation i is treated, we impute Y_i^0 using untreated units $Y_{j(i)}$ in control group
  2. If observation i is untreated, we impute Y_i^1 using treated units $Y_{j(i)}$ in treatment group
- The matching estimator for ATE is:

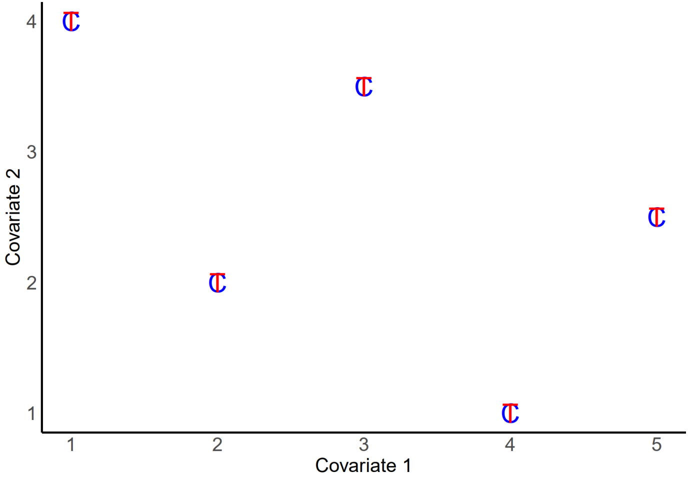
$$\hat{\alpha}_{\text{ATE}} = \frac{1}{N}\left\{\sum_{D_i=1}(Y_i - Y_{j(i)}) + \sum_{D_i=0}(Y_{j(i)} - Y_i)\right\}$$

Exact Matching

- Match each treated unit to a control unit that has exactly the same covariate values
- This is called **exact matching** and can be thought of as the gold standard for matching

Exact Matching

A Numerical Example



Source: Ben Elsner's slides

Exact Matching and the "Curse of Dimensionality"

- Matching becomes unfeasible with many covariates
- This is also true even if we divided each of covariates into coarse categories (subclassification)

Exact Matching and the "Curse of Dimensionality"

- Assume we have k covariates and divided each of them into 3 coarse categories
  - age could be "young", "middle age" or "old"
  - income could be "low", "medium" or "high"
- The number of subclassification cells is 3^k.
  - For $k = 10$, we obtain $3^{10} = 59049$
- Many cells may contain only treated or untreated observations
  - We may not be able to construct matched sample
  - Violate common support assumption

Approximate Matching

- In most cases, we just match similar units
- This is called **approximate matching**
- There are three main methods for approximate matching:
    1. **Distance Matching:** minimize distance in covariates X
    2. **Coarsened Exact Matching:** match within coarsened subgroups
    3. **Propensity Score Matching:** match on likelihood of being treated

Distance Matching

Measure Closeness

- We usually use more than one characteristics to construct a matched sample
- When the vector of matching covariates has more than one variables (h>1)

$$X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_H \end{pmatrix}$$

- We need to define a **distance metric** to measure "closeness" to construct a matched sample

Distance Matching

Measure Closeness

- The usual **Euclidean distance** is:

$$\begin{aligned} ||X_i - X_j|| &= \sqrt{(X_i - X_j)'(X_i - X_j)} \\ &= \sqrt{\sum_{h=1}^{H}(X_{hi} - X_{hj})^2}. \end{aligned}$$

  - Sum up the differences between treatment group and control group over *h* characteristics
    - ★ **Drawback:** The Euclidean distance is NOT invariant to changes in the scale of the *X*'s
    - ★ For this reason, we often use alternative distances that are invariant to changes in scale

Distance Matching

Measure Closeness

- A commonly used distance is the **normalized Euclidean distance**

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

where

$$\hat{V} = \begin{pmatrix} \hat{\sigma}_1^2 & 0 & \dots & 0 \\ 0 & \hat{\sigma}_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\sigma}_H^2 \end{pmatrix}.$$

- $\hat{\sigma}_h^2$ is the variance of variable h

Distance Matching

Measure Closeness

- Notice that, the normalized Euclidean distance is equal to:

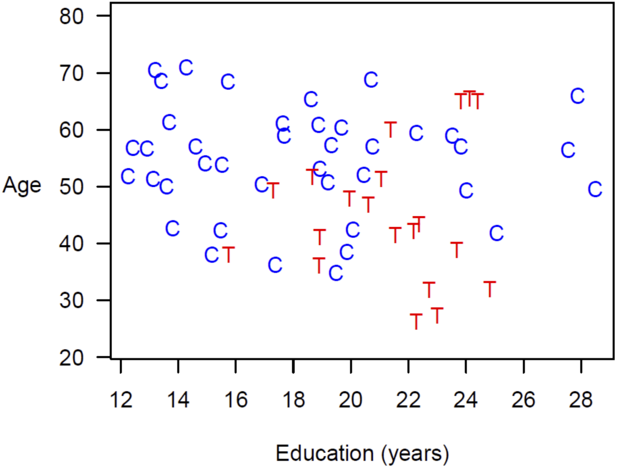
$$||X_i - X_j|| = \sqrt{\sum_{h=1}^{H} \frac{(X_{hi} - X_{hj})^2}{\hat{\sigma}_h^2}}.$$

⇒ Changes in the scale of X_{ki} affect also $\hat{\sigma}_k$, and the normalized Euclidean distance does not change

- k-Nearest-neighbor Matching
  - Match with the nearest neighbor or the k nearest neighbors in terms of normalized Euclidean distance
  - Drop the unmatched units
- Radius Matching
  - Match with all control units within a certain radius of the treated unit

Distance Matching

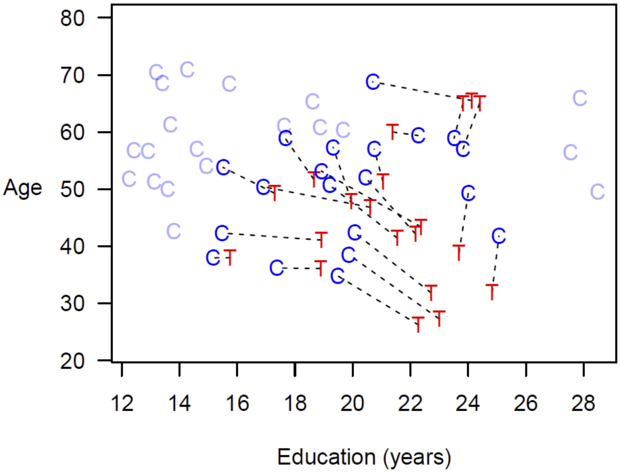
k-Nearest-neighbor Matching: k=1



Source: Ben Elsner's slides

Distance Matching

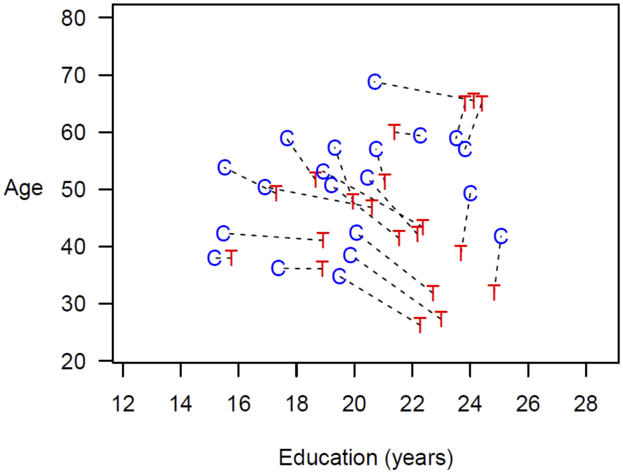
k-Nearest-neighbor Matching: k=1



Source: Ben Elsner's slides

Distance Matching

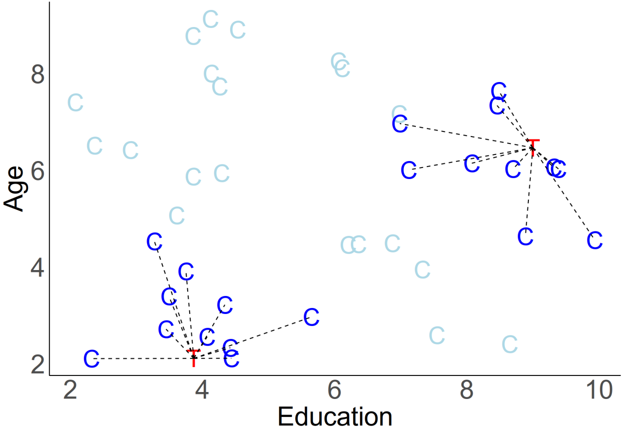
k-Nearest-neighbor Matching: k=1



Source: Ben Elsner's slides

Distance Matching

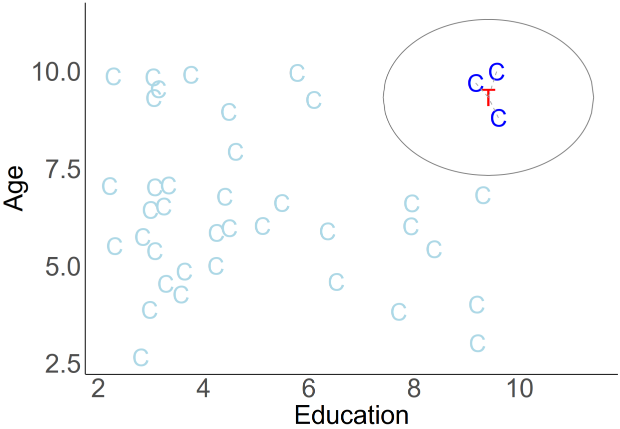
k-Nearest-neighbor Matching: k=10



Source: Ben Elsner's slides

Distance Matching

Radius Matching



Source: Ben Elsner's slides

Matching and the “Curse of Dimensionality”

- Matching discrepancies $||X_i - X_{j(i)}||$ tend to increase with number of covariates, the dimension of X
- It is difficult to find good matches in large dimensions: you need many observations if H is large

Propensity Score Matching
Main Idea

Propensity Score Matching

- Instead of matching over k dimensions, the method of **propensity score matching (PSM)** allows the matching problem to be reduced to a single dimension
  - The **propensity score** is defined as the treatment probability conditional on a set of observed variables X_i:

$$p(X_i) = E[D_i|X_i] = Pr(D_i = 1|X_i)$$

  - Intuitively, propensity score $p(X_i)$ summarized all information of a set of covariates X_i into a single value
  - Then, we can just control (match) $p(X_i)$ to eliminate selection bias

Propensity Score Matching

- Rosenbaum and Rubin (1983) proved that CIA (selection on observables) implies:

$$(Y_i^1, Y_i^0) \perp\!\!\!\perp D_i | p(X_i)$$

  - Conditioning on the propensity score $p(X_i)$ is enough to make treatment status be independent of the potential outcomes
  - Substantial dimension reduction in the matching variables!

Propensity Score Matching

Propensity Score Theorem

Suppose the CIA holds, such that $(\mathrm{Y}_i^1, \mathrm{Y}_i^0) \perp\!\!\!\perp D_i | X_i$. Then $(\mathrm{Y}_i^1, \mathrm{Y}_i^0) \perp\!\!\!\perp D_i | p(X_i)$

- If potential outcomes are independent of treatment status conditional on a set of covariates X_i
- Then, potential outcomes are independent of treatment status D_i conditional on the propensity score $p(X_i)$

Propensity Score Matching

- Goal of Proof:
  - Assume that $(\mathrm{Y}_i^1, \mathrm{Y}_i^0) \perp\!\!\!\perp D_i | X_i$. Then:

$\Rightarrow$ $Pr(D_i = 1 | \mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i)) = p(X_i) = Pr(D_i = 1 | p(X_i))$

$\Rightarrow$ $(\mathrm{Y}_i^1, \mathrm{Y}_i^0) \perp\!\!\!\perp D_i | p(X_i)$

Propensity Score Matching

Proof: Assume that $(\mathrm{Y}_i^1, \mathrm{Y}_i^0) \perp\!\!\!\perp D_i | X_i$. Then:

$$
\begin{aligned}
Pr(D_i = 1 | \mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i)) &= E[D_i | \mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i)] \\
&= E[E[D_i | \mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i), X_i] | \mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i)] \\
&= E[E[D_i | \mathrm{Y}_i^1, \mathrm{Y}_i^0, X_i] | \mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i)] \\
&= E[E[D_i | X_i] | \mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i)] \\
&= E[p(X_i) | \mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i)] \\
&= p(X_i)
\end{aligned}
$$

Propensity Score Matching

Using a similar argument, we obtain

$$\begin{aligned} Pr(D_i = 1|p(X_i)) &= E[D_i|p(X_i)] \\ &= E[E[D_i|p(X_i), X_i]|p(X_i)] \\ &= E[E[D_i|X_i]|p(X_i)] \\ &= E[p(X_i)|p(X_i)] \\ &= p(X_i) \end{aligned}$$

$\Rightarrow$ $Pr(D_i = 1|\mathrm{Y}_i^1, \mathrm{Y}_i^0, p(X_i)) = p(X_i) = Pr(D_i = 1|p(X_i))$

$\Rightarrow$ $(\mathrm{Y}_i^1, \mathrm{Y}_i^0) \perp\!\!\!\perp D_i|p(X_i)$

Propensity Score Matching

- From CIA, to get causal effect, we need only control for covariates that affect the probability of treatment
- The propensity score theorem says something more:
  - **The only covariate you really need to control for is the probability of treatment itself** $p(X_i) = Pr(D_i = 1|X_i)$

Identification Results for PSM

- PSM follows the same three identification steps as matching, but conditions on $p(X_i)$ instead of X_i:

Step 1 Show that ODO at given $p(X_i)$ equals **CATT** (selection bias $=0$)

Step 2 Under CIA, CATT $=$ CATU $=$ **CATE**

Step 3 Apply LIE to average CATE over $p(X)$ $\Rightarrow$ **ATT, ATU, ATE**

Identification Results for PSM

Steps 1 & 2: ODO = CATT = CATU = CATE

$$\underbrace{\mathrm{E}[\mathrm{Y}_i|p(X_i), D_i = 1] - \mathrm{E}[\mathrm{Y}_i|p(X_i), D_i = 0]}_{\text{ODO at given } p(X_i)}$$

$$= \mathrm{E}[\mathrm{Y}_i^1|p(X_i), D_i = 1] - \mathrm{E}[\mathrm{Y}_i^0|p(X_i), D_i = 0]$$

$$= \mathrm{E}[\mathrm{Y}_i^1|p(X_i), D_i = 1] - \mathrm{E}[\mathrm{Y}_i^0|p(X_i), D_i = 1]$$

$$+ \mathrm{E}[\mathrm{Y}_i^0|p(X_i), D_i = 1] - \mathrm{E}[\mathrm{Y}_i^0|p(X_i), D_i = 0]$$

$$= \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|p(X_i), D_i = 1]}_{\text{CATT}} + \underbrace{\mathrm{E}[\mathrm{Y}_i^0|p(X_i), D_i = 1] - \mathrm{E}[\mathrm{Y}_i^0|p(X_i), D_i = 0]}_{\text{Selection Bias=0 by CIA}}$$

$$= \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|p(X_i), D_i = 0]}_{\text{CATU}} \quad \text{(by CIA)} = \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0|p(X_i)]}_{\text{CATE}}$$

Identification Results for PSM

Step 3: ATT, ATU, and ATE

- **ATT**: average CATT over the **treatment group** distribution of $p(X)$:

$$\mathrm{E}\Big[\underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | p(X_i), D_i = 1]}_{\text{CATT}} \Big| D_i = 1\Big] = \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | D_i = 1]}_{\text{ATT}}$$

- **ATU**: average CATU over the **control group** distribution of $p(X)$:

$$\mathrm{E}\Big[\underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | p(X_i), D_i = 0]}_{\text{CATU}} \Big| D_i = 0\Big] = \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | D_i = 0]}_{\text{ATU}}$$

- **ATE**: average CATE over the **full population** distribution of $p(X)$:

$$\mathrm{E}\Big[\underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0 | p(X_i)]}_{\text{CATE}}\Big] = \underbrace{\mathrm{E}[\mathrm{Y}_i^1 - \mathrm{Y}_i^0]}_{\text{ATE}}$$

- **Summary**: Under CIA, PSM can identify ATT, ATU, and ATE by averaging $p(X)$-specific effects over the appropriate population

Propensity Score Matching
Estimation

Propensity Score Matching

Estimation

- There are two ways to estimate causal effect of treatment using PSM
  1. Nearest Neighbor:
     - ★ By matching each treated observation to the untreated observation with the same or similar values of the propensity score
  2. Weighting Approach
     - ★ Skip the cumbersome matching procedure and re-weight sample

Propensity Score Matching

Estimation: Nearest Neighbor

- There are two steps to estimate causal effect of treatment using PSM with nearest neighbor

  1. Estimate the propensity score: $\hat{p}(X) = \hat{P}r(D_i = 1|X_i)$ using logit or porbit regression

$$D_i = \beta_0 + \beta_1 X_i^1 + \beta_2 X_i^2 + ... + \beta_h X_i^h + \epsilon_i$$

  2. By matching each treated observation to the observation (control group) with the same or similar values of the propensity score $\hat{P}r(D_i = 1|X_i)$

Propensity Score Matching

A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	pro-score	earnings	unit	pro-score	earnings	unit	pro-score	earnings
1	0.28	17700	1	0.43	20900			
2	0.34	10200	2	0.50	31000			
3	0.29	14400	3	0.30	21000			
4	0.25	20800	4	0.27	9300			
5	0.29	6100	5	0.54	41100			
7	0.33	21900	7	0.39	42000			
8	0.27	28800	8	0.28	8800			
9	0.31	20300	9	0.24	25500			
10	0.26	28100	10	0.33	15500			
11	0.25	9400	11	0.26	400			
12	0.27	14300	12	0.31	26600			
13	0.29	12500	13	0.26	16500			
14	0.24	19700	14	0.34	24200			
15	0.25	10100	15	0.25	23300			
16	0.43	10700	16	0.24	9700			
17	0.28	11500	17	0.29	6200			
18	0.27	10700	18	0.35	30200			
19	0.28	16300	19	0.32	17800			
			20	23	9500			
			21	32	25900			
Avg:		16426	Avg:		20724	Avg:		

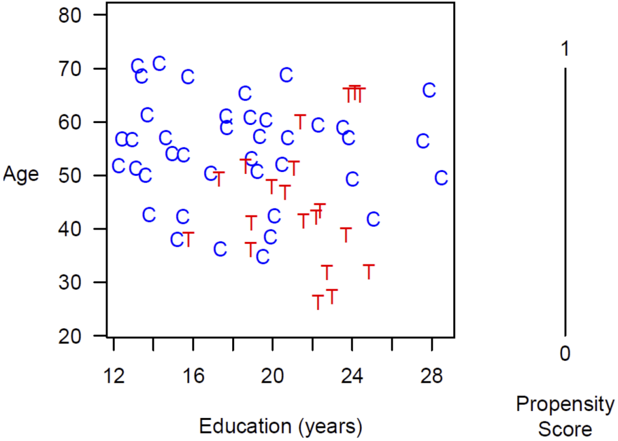
Propensity Score Matching

A Numerical Example

Trainees			Non-Trainees			Matched Sample		
unit	pro-score	earnings	unit	pro-score	earnings	unit	pro-score	earnings
1	0.28	17700	1	0.43	20900	8	0.28	8800
2	0.34	10200	2	0.50	31000	14	0.34	24200
3	0.29	14400	3	0.30	21000	17	0.29	6200
4	0.25	20800	4	0.27	9300	15	0.25	23300
5	0.29	6100	5	0.54	41100	17	0.29	6200
6	0.23	28600	6	0.48	29800	20	0.23	9500
7	0.33	21900	7	0.39	42000	10	0.33	15500
8	0.27	28800	8	0.28	8800	4	0.27	9300
9	0.31	20300	9	0.24	25500	12	0.31	26600
10	0.26	28100	10	0.33	15500	11,13	0.26	8450
11	0.25	9400	11	0.26	400	15	0.25	23300
12	0.27	14300	12	0.31	26600	4	0.27	9300
13	0.29	12500	13	0.26	16500	17	0.29	6200
14	0.24	19700	14	0.34	24200	9,16	0.24	17700
15	0.25	10100	15	0.25	23300	15	0.25	23300
16	0.43	10700	16	0.24	9700	1	0.43	20900
17	0.28	11500	17	0.29	6200	8	0.28	8800
18	0.27	10700	18	0.35	30200	4	0.27	9300
19	0.28	16300	19	0.32	17800	8	0.28	8800
			20	0.23	9500			
			21	0.32	25900			
Avg:		16426	Avg:		20724	Avg:		13982

Propensity Score Matching

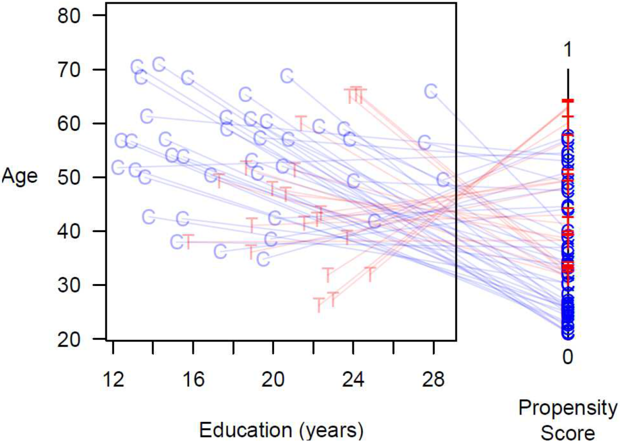
A Numerical Example



Source: Ben Elsner's slides

Propensity Score Matching

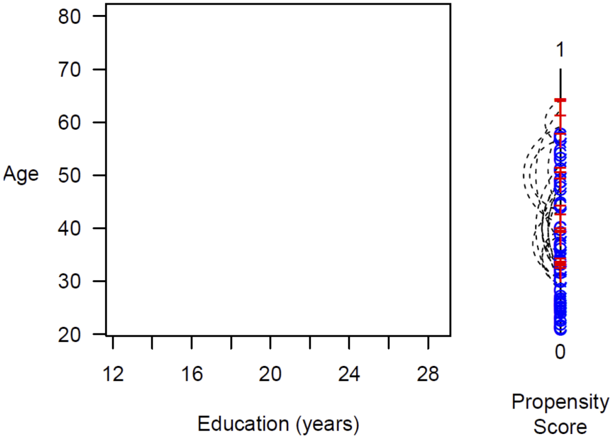
A Numerical Example



Source: Ben Elsner's slides

Propensity Score Matching

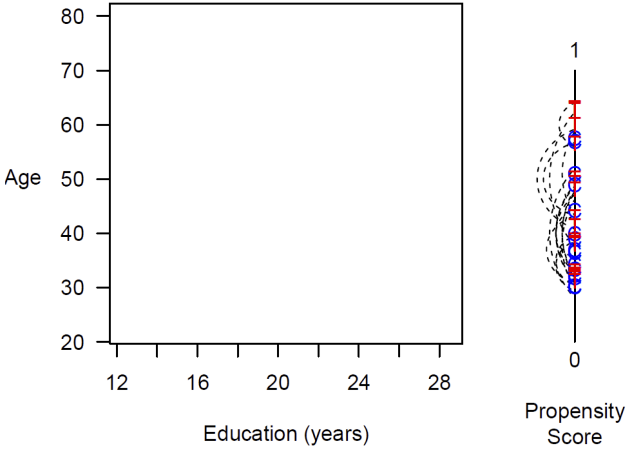
A Numerical Example



Source: Ben Elsner's slides

Propensity Score Matching

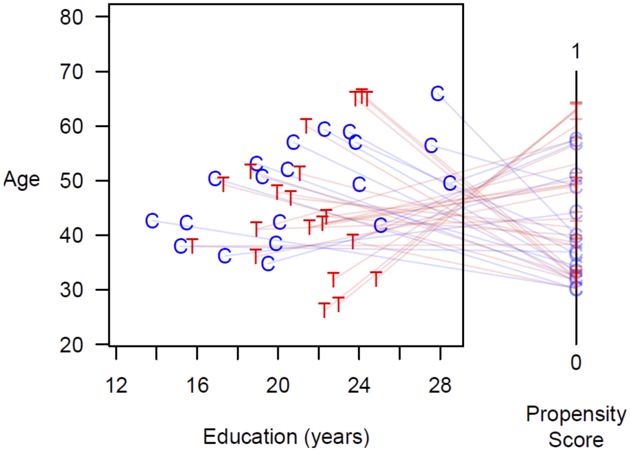
A Numerical Example



Source: Ben Elsner's slides

Propensity Score Matching

A Numerical Example



Source: Ben Elsner's slides

Propensity Score Matching
Statistical Inference

Propensity Score Matching

Statistical Inference

- A valid method to calculate standard errors when using estimated propensity scores was formally derived by Abadie and Imbens (2016)
- Abadie, Alberto, and Guido W. Imbens. "**Matching on the Estimated Propensity Score**." *Econometrica* 84.2 (2016): 781–807.
  - Need to account for the fact that propensity scores are **estimated**, not fixed and known constants
    - ★ For ATE: The adjustment is always **negative** —standard errors are smaller than if we treated $\hat{p}(X_i)$ as fixed constants
    - ★ For ATT: The adjustment can be **positive or negative** —standard errors can be either smaller or larger
  - Ignoring this adjustment can lead to **incorrect inference**

Propensity Score Matching

Drawback

- PSM is hugely popular method to estimate treatment effects even if it relies on **less convincing assumption**:
  - Selection on observables (CIA)

Empirical Analysis Workflow

Empirical Analysis Workflow

Step	Task	What to Do
1	**Organize project**	Set up folder structure, path setup block
2	**Read the codebook**	Understand variables, units, known data issues
3	**Clean data**	Label variables/values, handle missing values and outliers
4	**Examine data**	Examine distributions, means, and frequencies of key variables
5	**Pre-analysis balance check**	Compare treated vs. control group *before* any analysis
6	**Proceed to analysis**	PSM, regression, DiD, ...

Before You Start: Organize Your Project

- Set up a **clear folder structure** before touching any data:
  - `/rawdata` — original data, **never modify**
  - `/workdata` — cleaned/processed data
  - `/code` — all scripts (`.do`, `.R`, …)
  - `/output` — tables, figures, logs
- Use **relative paths** or a **path setup block** at the top of every script so the project runs on any machine
- **Never overwrite raw data** — all cleaning steps should be recorded in a script and saved to `/workdata`
- **Comment your code** — your future self (and collaborators) will thank you

Getting to Know Your Data

- **Read the codebook** before running anything:
  - What does each variable measure? What are the units?
  - How is the treatment defined? What is the outcome?
  - Are there known data issues (top-coding, imputation, ...)?
- **Label your variables and values** for clarity:
  - Give every variable a descriptive name and unit
  - Label categorical values (e.g., 0 = Control, 1 = Treated)
- **Explore the data** systematically:
  - Examine variable types, distributions, and frequencies
  - Check for **missing values**, **outliers**, and **implausible values**

How AI Agents Can Help

- **Agent-type AI** (e.g. Claude Code, Gemini CLI) can operate directly in your project folder — not just answer questions, but **read, write, and execute code** on your behalf

STATA Example

STATA Example

Dehejia et al. (1999)

Rajeev H. Dehejia; Sadek Wahba (1999)“**Causal Effects in Nonexperimental Studies: Reevaluating the Evaluation of Training Programs**”Journal of the American Statistical Association

- The authors wants to examine the effect of job training on workers' earnings
- We use this example to go through the procedure of implementing PSM

STATA Example

Dehejia et al. (1999)

- See **matching.do**
- Use lalonde.dta
- Install the following ado files:
  - psmatch2.ado

Path Setup

```
if "`c(username)'" == "ttyang" {
global do = "C:\nest\Dropbox\causal_data_course\code\
    Class_Data\do"
global rawdata = "C:\nest\Dropbox\causal_data_course\code\
    Class_Data\rawdata"
global workdata = "C:\nest\Dropbox\causal_data_course\code\
    Class_Data\workdata"
}
if "`c(username)'" == "nest" {
global do = "D:\nest\Dropbox\causal_data_course\code\
    Class_Data\do"
global rawdata = "D:\nest\Dropbox\causal_data_course\code\
    Class_Data\rawdata"
global workdata = "D:\nest\Dropbox\causal_data_course\code\
    Class_Data\workdata"
}
```

Install Package (ado file)

ssc install

```
ssc install psmatch2
```

- Install the psmatch2 package using the ssc install command

Read Data

import delimited: reads CSV files

```
import delimited "$rawdata/lalonde.csv", clear

export delimited using "$rawdata/lalonde.csv", replace
```

- **import delimited:** Standard command for reading CSV files in Stata
- **export delimited:** Writes data to CSV efficiently, with options for handling column names and delimiters

Read Data

use/save: Work with Stata files

```
use "$rawdata/lalonde.dta", clear

save "$rawdata/lalonde.dta", replace
```

- **use:** Basic command for reading Stata's ‘.dta‘ files
- **save:** Stores data in Stata's native ‘.dta‘ format with specified version compatibility

Examine Data

`codebook`: Display Summary Statistics

```
codebook age educ re78
```

- Display detailed information about variables: `age`, `educ`, and `re78`
  - **codebook**: Provides summary statistics, data type, range, and distribution details
  - Useful for initial data exploration and understanding variable characteristics

Examine Data

sum: Display Summary Statistics

```
sum re78, d
```

- Display detailed summary statistics for re78
  - **sum**: Computes summary statistics such as mean, standard deviation, and range
  - **d** option: Provides a more detailed summary, including percentiles and extreme values
  - Useful for understanding the distribution and variability of 1978 real earnings

Examine Data

tab: Produce a frequency table

```
tab treat
```

- Display frequency table for the treat variable
  - **tab**: Shows the count and percentage for each category of the treat variable
  - Useful for checking the balance between treatment and control groups in the study

Create Sample for Analysis

`gen`: Create New Variables

```
gen id=_n
```

- Generate a new variable `id` as a unique identifier for each observation
  - **gen**: Creates new variables in Stata
  - **_n**: Represents the observation number in the dataset
  - Useful for creating a unique ID for each row in the dataset

Examine Data

`duplicates`: Detecting Repeated Observations

```
duplicates report id
```

- Check for duplicate values in the `id` variable
  - **duplicates report**: Reports the number of observations and distinct values
  - Identifies any duplicate entries in the `id` variable
  - Crucial for ensuring each observation has a unique identifier

STATA Example

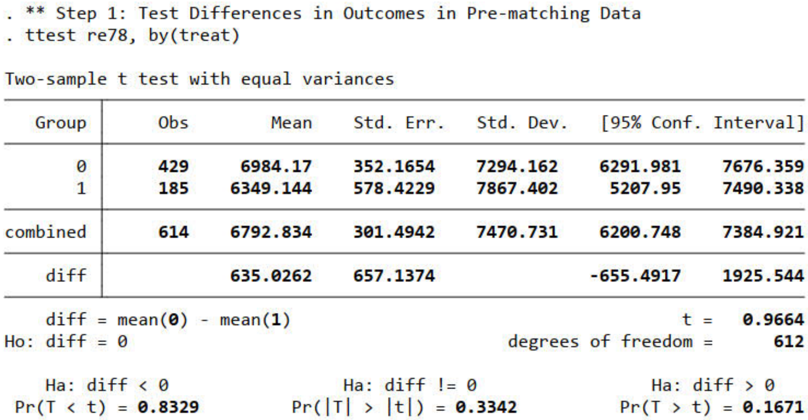
Step 1: Test Differences in Outcomes in Pre-matching Data

```
ttest re78, by(treat)
```

- Perform a two-sample t-test on the `re78` variable, grouped by the `treat` variable
    - **ttest**: Compares the mean 1978 real earnings (`re78`) between treatment and control groups
    - **by(treat)**: Specifies that the comparison is made between the two groups defined by `treat`
    - Tests whether the difference in means is statistically significant before matching

STATA Example

Step 1: Test Differences in Outcomes in Pre-matching Data



```
. ** Step 1: Test Differences in Outcomes in Pre-matching Data
. ttest re78, by(treat)

Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |     429     6984.17    352.1654    7294.162    6291.981    7676.359
       1 |     185    6349.144    578.4229    7867.402     5207.95    7490.338
---------+--------------------------------------------------------------------
combined |     614    6792.834    301.4942    7470.731    6200.748    7384.921
---------+--------------------------------------------------------------------
    diff |            635.0262    657.1374               -655.4917    1925.544
------------------------------------------------------------------------------
    diff = mean(0) - mean(1)                                      t =   0.9664
Ho: diff = 0                                     degrees of freedom =      612

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.8329         Pr(|T| > |t|) = 0.3342          Pr(T > t) = 0.1671
```

STATA Example

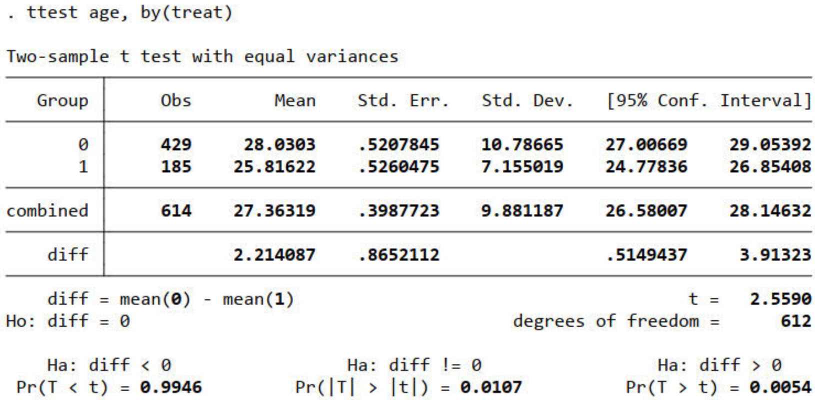
Step 2: Test Differences in Covariates in Pre-matching Data

```
ttest age, by(treat)
ttest educ, by(treat)
```

- Perform two-sample t-tests on the covariates age and educ, grouped by the `treat` variable
  - **ttest age, by(treat)**: Compares the mean age between treatment and control groups
  - **ttest educ, by(treat)**: Compares the mean education level between treatment and control groups
  - Assesses balance in key covariates before matching to detect pre-existing differences

STATA Example

Step 2: Test Differences in Covariates in Pre-matching Data



```
. ttest age, by(treat)

Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |     429     28.0303    .5207845    10.78665    27.00669    29.05392
       1 |     185    25.81622    .5260475    7.155019    24.77836    26.85408
---------+--------------------------------------------------------------------
combined |     614    27.36319    .3987723    9.881187    26.58007    28.14632
---------+--------------------------------------------------------------------
    diff |            2.214087    .8652112                .5149437     3.91323
------------------------------------------------------------------------------
    diff = mean(0) - mean(1)                                      t =   2.5590
Ho: diff = 0                                     degrees of freedom =      612

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.9946         Pr(|T| > |t|) = 0.0107          Pr(T > t) = 0.0054
```

STATA Example

Step 3: PSM Estimation – teffects psmatch

Syntax:

```
teffects psmatch (outcome) (treatment covariates, logit), nn
     (#) ate
```

- **nn(#)**: specify number of matches per observation; default is nn(1)
  - The number of variables generated may be more than nn(#) because of tied distances
- **logit**: use logit to predict propensity score (the default)
- **ate**: estimate average treatment effect in population (the default)
- **atet**: estimate average treatment effect on the treated

STATA Example

Step 3: PSM Estimation – teffects psmatch

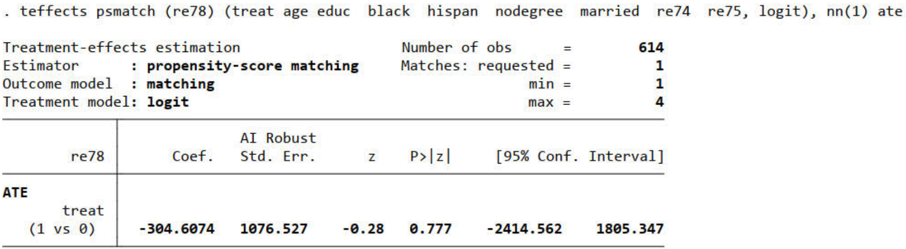
Example:

```
teffects psmatch (re78) (treat age educ  black  hispan
    nodegree  married  re74  re75, logit), nn(1) atet
teffects psmatch (re78) (treat age educ  black  hispan
    nodegree  married  re74  re75, logit), nn(1) ate
```

- Outcome: `re78` (earnings in 1978)
- Treatment: `treat` (get job training or not)

STATA Example

Step 3: PSM Estimation – teffects psmatch



```
. teffects psmatch (re78) (treat age educ  black  hispan  nodegree  married  re74  re75, logit), nn(1) ate

Treatment-effects estimation                   Number of obs      =        614
Estimator      : propensity-score matching     Matches: requested =          1
Outcome model  : matching                                     min =          1
Treatment model: logit                                        max =          4
----------------------------------------------------------------------------
             |              AI Robust
        re78 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+--------------------------------------------------------------
ATE          |
       treat |
   (1 vs 0)  |  -304.6074   1076.527    -0.28   0.777    -2414.562    1805.347
----------------------------------------------------------------------------
```

STATA Example

Step 3: PSM Estimation – teffects psmatch

```
. teffects psmatch (re78) (treat age educ  black  hispan  nodegree  married  re74  re75, logit), nn(1) atet

Treatment-effects estimation                   Number of obs      =        614
Estimator      : propensity-score matching     Matches: requested =          1
Outcome model  : matching                                     min =          1
Treatment model: logit                                        max =          4
------------------------------------------------------------------------------
             |              AI Robust
        re78 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ATET         |
       treat |
   (1 vs 0)  |     1968.8   1126.321     1.75   0.080    -238.7493    4176.349
------------------------------------------------------------------------------
```

STATA Example

Step 3: PSM Estimation – teffects psmatch

Understanding the matching process:

```
teffects psmatch (re78) (treat age educ  black  hispan
    nodegree  married  re74  re75), nn(1)  atet gen(matchnum
    )
```

- **gen(matchnum)**: specifies that the observation numbers of the nearest neighbors be stored in the new variables matchnum1, matchnum2, ...
- This option is required if you wish to perform postestimation based on the matching results

STATA Example

Step 3: PSM Estimation – teffects psmatch

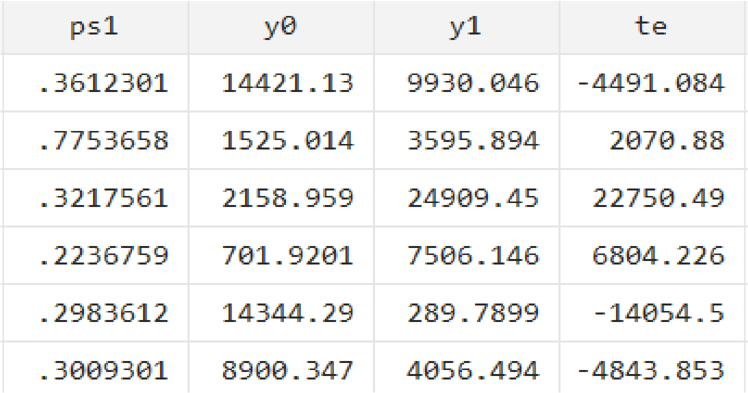
Understanding the matching process:

```
predict ps1, ps
predict y0 y1, po
predict te
```

- **predict ps1, ps**: predict propensity score (i.e. probability of getting treatment)
- **predict y0 y1, po**: generate the potential outcome with or without treatment
- **predict te**: get treatment effect for each observation

STATA Example

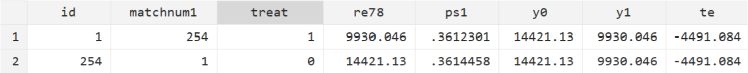
Step 3: PSM Estimation – teffects psmatch



ps1	y0	y1	te
.3612301	14421.13	9930.046	-4491.084
.7753658	1525.014	3595.894	2070.88
.3217561	2158.959	24909.45	22750.49
.2236759	701.9201	7506.146	6804.226
.2983612	14344.29	289.7899	-14054.5
.3009301	8900.347	4056.494	-4843.853

STATA Example

Step 3: PSM Estimation – teffects psmatch



	id	matchnum1	treat	re78	ps1	y0	y1	te
1	1	254	1	9930.046	.3612301	14421.13	9930.046	-4491.084
2	254	1	0	14421.13	.3614458	14421.13	9930.046	-4491.084

STATA Example

Step 3: PSM Estimation – psmatch2

Syntax:

```
psmatch2 treatment covariates, out(outcome) n(#) logit ate
```

- **n(#)**: specify number of matches per observation; default is nn(1)
  - The number of variables generated may be more than n(#) because of tied distances
- **out(var)**: specify an outcome variable
- **ate**: display ATT, ATU, ATE

STATA Example

Step 3: PSM Estimation – psmatch2

Example:

```
psmatch2 treat age educ  black  hispan  nodegree  married
    re74  re75, out(re78) logit n(1) ate
```

- The PSM estimate is similar to the one using teffects

STATA Example

Compare teffects psmatch and psmatch2

- The **teffects psmatch** command has one very important advantage over **psmatch2**
  - **teffects psmatch** takes into account the fact that propensity scores are estimated rather than known when calculating standard errors.
  - **teffects psmatch** calculates standard errors based on this paper:
    - ★ Abadie, Alberto, and Guido W. Imbens. "**Matching on the Estimated Propensity Score**." Econometrica 84.2 (2016): 781-807.

STATA Example

Compare teffects psmatch and psmatch2

- But **psmatch2** can allow matching without replacement, which is quite useful.

STATA Example

Step 3: PSM Estimation – psmatch2

Example:

```
psmatch2 treat age educ  black  hispan  nodegree  married
    re74  re75, out(re78) logit n(1) noreplace
```

- **noreplace**: STATA will perform PSM without replacement so that each untreated observation can be used only once.

STATA Example

Step 4: Post Matching Analysis – teffects psmatch

Example:

```
tebalance box re74
tebalance density educ
tebalance density
```

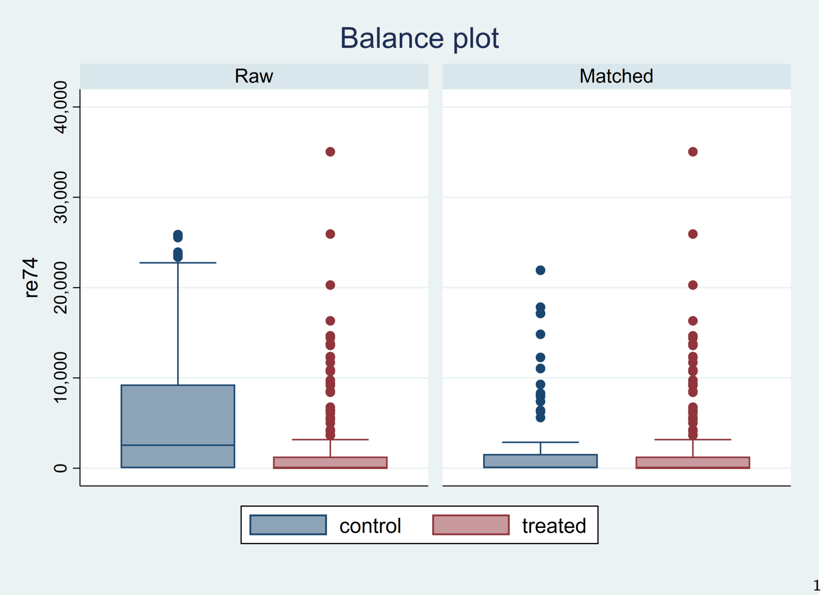
- **tebalance box**: Produces box plots that are used to check for balance in matched samples after **teffects**

- **tebalance density**: Produces density plots that are used to check for covariate balance after estimation by a **teffects**

- If you do not specify variable, it will plot the density of propensity score

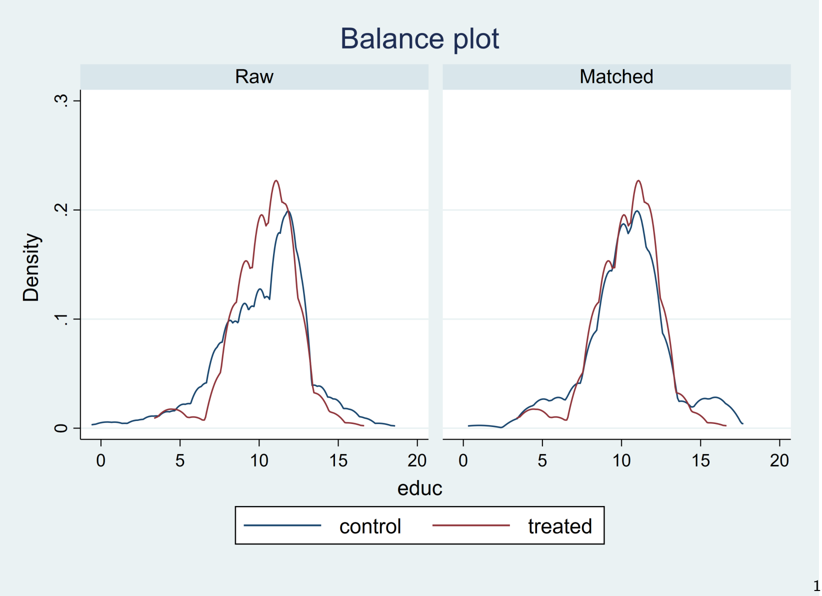
STATA Example

Step 4: Post Matching Analysis – teffects psmatch



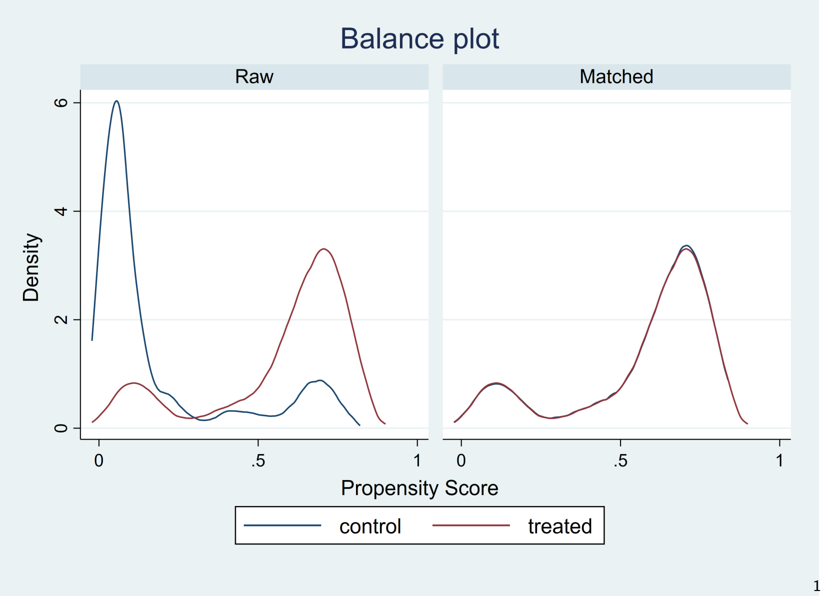
STATA Example

Step 4: Post Matching Analysis – teffects psmatch



STATA Example

Step 4: Post Matching Analysis – teffects psmatch



STATA Example

Step 4: Post Matching Analysis – psmatch2

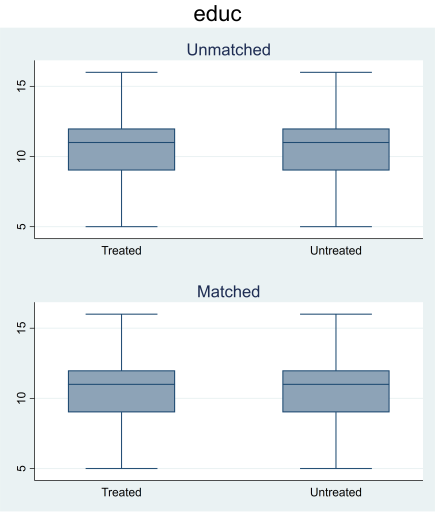
Example:

```
pstest age educ  black  hispan  nodegree  married  re74
    re75, both
pstest educ, box both
pstest _pscore, density both
```

- command **pstest**: calculates and optionally graphs several measures of the extent of balancing of the variables between two groups.
- option **both**: compares the extent of balancing between the two samples before and after having performed matching.
- option **box**: draw box plot to compare two groups
- option **density**: draw density plot to compare two groups

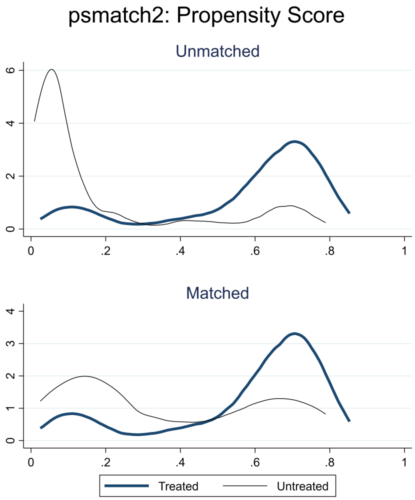
STATA Example

Step 4: Post Matching Analysis – psmatch2



STATA Example

Step 4: Post Matching Analysis – psmatch2



R Example

R Example

Dehejia et al. (1999)

- See **matching.R**
- Use lalonde.dta
- Install the following package:
  - MatchIt

Path Setup

```r
rm(list = ls())

# Set paths based on the username
username <- Sys.info()[["user"]]
if (username == "ttyang") {
rawdata <- "C:/nest/Dropbox/causal_data_course/code/Class_Data/rawdata"
} else if (username == "nest") {
 rawdata <- "D:/nest/Dropbox/causal_data_course/code/Class_Data/rawdata"
}
```

- **rm(list = ls()):** clears all objects from the environment
- **Sys.info()[[”user”]]:** retrieves current system username
- Conditional path setting allows for reproducible workflow across different machines
- Organizes project into logical directories (raw data, working data)

Install and Load Package

install.packages()

```
# Install necessary packages
install.packages('Matching') # for PSM
install.packages('haven') # for read_dta
install.packages('dplyr') # for mutate
install.packages('data.table') # for fread


# Load packages
library(Matching)
library(haven)
library(dplyr)
library(data.table)
```

- **install.packages()**: downloads and installs the package from CRAN
- **library()**: loads the installed package for use in the current R session

Read Data

fread(): reads CSV files

```
lalonde_csv <- fread(paste0(rawdata, "/lalonde.csv"), data.table = FALSE)

fwrite(lalonde, file = paste0(rawdata, "/lalonde.csv"))
```

- **fread:** Efficiently reads CSV files, significantly faster than ‘read.csv‘
- **fwrite:** Writes data to CSV efficiently, preferred over ‘write.csv‘ for large datasets

Read Data

read_dta(): Read the Stata files

```
lalonde <- read_dta(paste0(rawdata, "/lalonde.dta"))

write_dta(lalonde_csv, paste0(rawdata, "/lalonde.dta"),
    version = 14)
```

- **read_dta:** Reads Stata's '.dta' file into R as a data frame
- **write_dta:** Saves data to Stata's '.dta' format with specified version compatibility

Examine Data

summary(): Display Summary Statistics

```
summary(lalonde$re78)
```

- **summary():** Provides a concise statistical summary of `re78`
- Includes:
  - Minimum and maximum values
  - 1st quartile (25th percentile) and 3rd quartile (75th percentile)
  - Median (50th percentile) and mean
- Helps identify potential outliers or skewness in the dataset

Examine Data

table(): Display Frequency and Percentage Tables

```r
# Frequency table
table(lalonde$treat)

# Percentage table
prop.table(table(lalonde$treat)) * 100
```

- Display frequency and percentage tables for the treat variable
  - **table():** Shows the count for each category
  - **prop.table()** combined with **table()** calculates the percentage for each category

Create Sample for Analysis

`mutate()`: Create New Variables

```
library(dplyr)

lalonde <- lalonde %>%
mutate(id = row_number())
```

- **mutate():** Creates new variables in a data frame
- **id:** A unique identifier assigned to each observation
  - **row_number():** Generates a sequence of numbers based on the current row order
- **%>% (pipe operator):** Passes the left-hand side as the input to the right-hand function
  - Improves readability by avoiding nested function calls
  - Allows step-by-step transformations in a logical order
  - Commonly used in `dplyr` for chaining multiple operations

Examine Data

any(duplicated()): Check for Duplicates

```
# Check for any duplicates
any(duplicated(lalonde$id))

# Count total number of duplicates
sum(duplicated(lalonde$id))
```

- **Checking for Duplicates:** Ensures each observation has a unique identifier
    - **duplicated()**: Identifies duplicate entries in a vector or column
    - **any(duplicated())**: Returns `TRUE` if there are any duplicates, otherwise `FALSE`
    - **sum(duplicated())**: Counts the total number of duplicate entries
- Useful for detecting data entry errors and ensuring data integrity before analysis

R Example

Step 1: Test Differences in Outcomes in Pre-matching Data

```
t.test(re78 ~ treat, data = lalonde)
```

- Perform a two-sample t-test on the `re78` variable, grouped by the `treat` variable
  - **t.test()**: Compares the mean 1978 real earnings (`re78`) between treatment and control groups
  - Tests whether the difference in means is statistically significant before matching

R Example

Step 2: Test Differences in Covariates in Pre-matching Data

```
# T-test for age
t.test(age ~ treat, data = lalonde)

# T-test for education
t.test(educ ~ treat, data = lalonde)
```

- Perform two-sample t-tests on the covariates age and educ, grouped by the `treat` variable

R Example

Step 3: PSM Estimation via MatchIt

```
library(MatchIt)

# Estimate PS and perform nearest neighbor matching
m.out <- matchit(treat ~ age + educ + black + hispan +
                 nodegree + married + re74 + re75,
                 data     = lalonde,
                 method   = "nearest",
                 distance = "logit",
                 replace  = TRUE,
                 estimand = "ATT")
summary(m.out)
```

- **matchit()**: estimates PS via logistic regression and performs nearest neighbor matching in one step
- **distance = ”logit”**: uses logit model to estimate propensity scores
- **replace = TRUE**: matching with replacement

R Example

Step 3: Estimate ATT

```
# Extract matched data
m.data <- match.data(m.out)

# Estimate ATT on matched sample
fit <- lm(re78 ~ treat, data = m.data, weights = weights)
summary(fit)
```

- **match.data()**: extracts matched sample with `MatchIt` weights
- **lm()**: estimates ATT on the matched sample; `weights` is automatically created by `match.data()`
- **Advantage over `Match()`**: `MatchIt` provides a unified framework for PSM, CEM, and other methods with consistent syntax

Coarsened Exact Matching

Coarsened Exact Matching (CEM)

- PSM reduces matching to one dimension (the propensity score), but requires a correctly specified model
- **CEM** takes a more direct approach:
  - **Coarsen** each covariate into discrete bins (e.g., age 20–29, 30–39, …)
  - **Exact match** on the coarsened values: a treated and a control unit are matched only if they fall in the *same bin for every covariate simultaneously*
  - A treated and a control unit that cannot be matched across *all* covariates simultaneously are **discarded**
- No propensity score model needed — covariate balance is **guaranteed by construction**

How CEM Works

Steps 1 & 2: Coarsen and Match

1. **Coarsen**: bin each continuous covariate X_k into discrete intervals (automatic or user-defined cutpoints)
   - E.g., age: $[20, 30)$, $[30, 40)$, $[40, 50)$; earnings: $[0, 5K)$, $[5K, 15K)$, …
2. **Exact match** on coarsened values:
   - A treated and a control unit are matched only if they fall in the *same bin for every covariate simultaneously*
   - Each such group of matched units is called a **stratum**: a subgroup of "similar" units who fall in the same bin for every covariate
   - Units that cannot be matched across *all* covariates simultaneously are **discarded**

How CEM Works

Stratum Example

Each **stratum** = a unique combination of coarsened bins.
Units are matched *only within* the same stratum.

Stratum	**Age bin**	**Earnings bin**	**Treated** $(D = 1)$	**Control** $(D = 0)$
1	$[20, 30)$	$[0, 5K)$	2 units	3 units ✓
2	$[20, 30)$	$[5K, 15K)$	1 unit	0 units → discarded
3	$[30, 40)$	$[0, 5K)$	0 units	2 units → discarded
4	$[30, 40)$	$[5K, 15K)$	3 units	4 units ✓

- Stratam 1, 4: both treated and control units present ⇒ **matched**
- Stratum 2: no control units available ⇒ treated unit discarded
- Stratum 3: no treated units ⇒ control units discarded

How CEM Works

Step 3: Estimate ATT

3. For each **matched** treated unit i, impute the counterfactual using the **average outcome of control units in the same stratum** $s(i)$:

$$\hat{\alpha}_{\text{ATT}} = \frac{1}{N_1} \sum_{D_i=1} \left(Y_i - \bar{Y}_{0,s(i)} \right)$$

- N_1: number of **matched** treated units (unmatched treated units are discarded)
- $s(i)$: the stratum that treated unit i belongs to
- $\bar{Y}_{0,s(i)} = \frac{1}{N_{0,s(i)}} \sum_{D_j=0,\, j \in s(i)} Y_j$: average *observed* outcome of control units $(D_j = 0)$ in i's stratum

Advantages of CEM

- **No model dependence**: no propensity score to estimate, so no misspecification risk
- **Guaranteed balance**: treated and matched controls fall in the same covariate bins by construction
- **Transparent**: the coarsening is explicit and easy to interpret
- **Tradeoff**:
  - Finer bins ⇒ better balance but fewer matched units
  - Coarser bins ⇒ more matches but looser balance
  - With many covariates, most strata may be empty — **curse of dimensionality**

CEM: STATA Example

CEM: Stata Example

Install Package and Run CEM

```
use $rawdata\lalonde.dta, replace

* Install cem package
ssc install cem

* Automatic coarsening (Sturges' rule)
cem age educ black hispan nodegree married re74 re75, ///
    treatment(treat)
```

- **cem varlist, treatment()**: coarsens each variable, exact-matches, and creates `cem_weights` and `cem_matched`
- Without cutpoints, CEM uses automatic (Sturges’ rule) coarsening

CEM: Stata Example

Estimate ATT

```
* Estimate ATT on matched sample using CEM weights
reg re78 treat [iweight=cem_weights] if cem_matched==1, r

* Optional: manual coarsening (specify number of bins)
cem age (#4) educ (#3) re74 (#5) re75 (#5) ///
    black hispan nodegree married, treatment(treat)

reg re78 treat [iweight=cem_weights] if cem_matched==1, r
```

- **iweight=cem_weights**: applies CEM weights to balance strata
- **#k**: specifies *k* equal-width bins for that variable

CEM: R Example

CEM: R Example

Run CEM

```r
install.packages("MatchIt")
library(MatchIt)

# Run CEM
m.out <- matchit(treat ~ age + educ + black + hisp +
                 married + nodegree + re74 + re75,
                 data   = lalonde,
                 method = "cem")
summary(m.out)
```

- **matchit()**: coarsens covariates and creates strata; covariates in the formula are used for matching

CEM: R Example

Estimate ATT

```
# Extract matched data and estimate ATT
m.data <- match.data(m.out)
fit <- lm(re78 ~ treat, data = m.data, weights = weights)
summary(fit)
```

- **match.data()**: extracts the matched sample with CEM weights
- **lm()**: estimates ATT on the matched sample using CEM weights; `weights` is automatically created by `match.data()`