WEEK 5 – MATCHING METHODS

Tutorial PSM with Python

2022-23925 SYALIANDA SALSABILA IZZATI

2022-25172 LEE SOOIN

O U T L I N E

Introduction: The Psychology of Growth

Data Description

Normal Linear Regression

Selection Bias

Propensity Score Estimation

Logistic Regression

Linear Regression using PSM

Easier way to implement C.I.

Result of using “CausalModel”

INTRO

What is Growth Mindset?

FIXED MINDSET VS. GROWTH MINDSET

Fixed Mindset

- Abilities are **given** at birth or in early childhood
- Should not **waste time** on areas where you don't excel.
- Wants to **prove** intelligence or talent
- Traits as stable and **unchangeable**

Growth mindset

- Intelligence can be **developed**
- Your failure is not as lack of tenacity, but lack of **practice**
- Want to **improve** intelligence or talent
- Traits as learnable and **changeable**

RESEARCH QUESTION

Is it that a growth mindset causes people to achieve more?

OR

Is simply the case that people who achieve more are prone to develop a growth mindset as a result of their success?

DATA DESCRIPTION

Treatment (T) : Students can choose to attend seminar about “Growth mindset”. (Intervention)

Dependent Variable (Y): How well they’ve performed academically (Achievement score)

Data Description

```
data = pd.read_csv("learning_mindset.csv")
```

```
data.shape
```

(10391, 13)

```
data.sample(5, random_state=5)
```

	schoolid	intervention	achievement_score	success_expect	ethnicity	gender	frst_in_family	school_urbanicity	school_mindset	school_achievement	s
259	73	1	1.480828	5	1	2	0	1	-0.462945	0.652608	
3435	76	0	-0.987277	5	13	1	1	4	0.334544	0.648586	
9963	4	0	-0.152340	5	2	2	1	0	-2.289636	0.190797	
4488	67	0	0.358336	6	14	1	0	4	-1.115337	1.053089	
2637	16	1	1.360920	6	4	1	0	1	-0.538975	1.433826	

NORMAL LINEAR REGRESSION

Given students who attended seminar, Intervention = 1, otherwise = 0 (Binary Variable)

```
smf.ols("achievement_score ~ intervention", data=data).fit().summary().tables[1]
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.1538	0.012	-13.201	0.000	-0.177	-0.131
intervention	0.4723	0.020	23.133	0.000	0.432	0.512

Effect of Intervention: 0.4723?

$E[Y|T=1] - E[Y|T=0]$ **=** Average Treatment Effect (ATE) + Selection Bias

SELECTION BIAS

Positive Bias

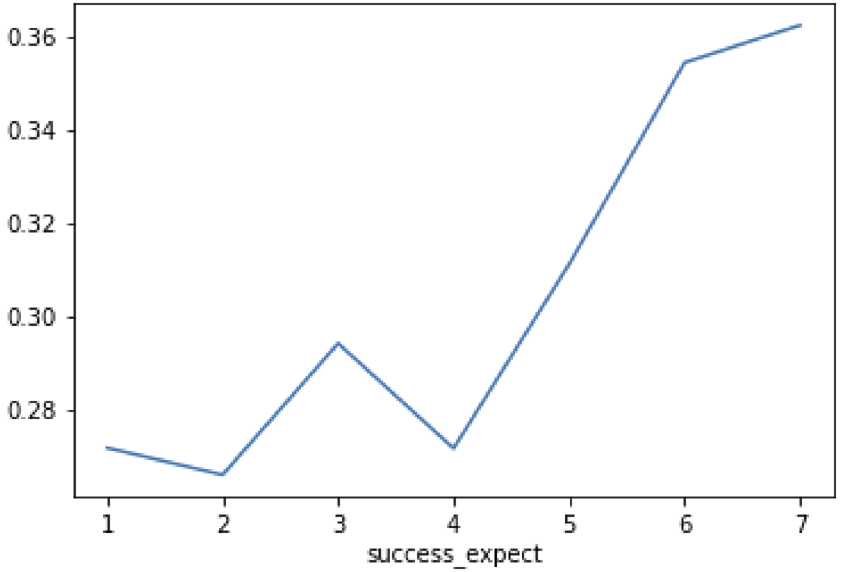
1. Did students who attended the seminar got higher achievement score? **(Treatment Effect)**

2. Or as more ambitious students are more willing to go to the seminar, even if they had not attended it they probably would have higher achievement score more. **(Selection Bias)**

Success expectation vs. Intervention

```
expect.plot()
```

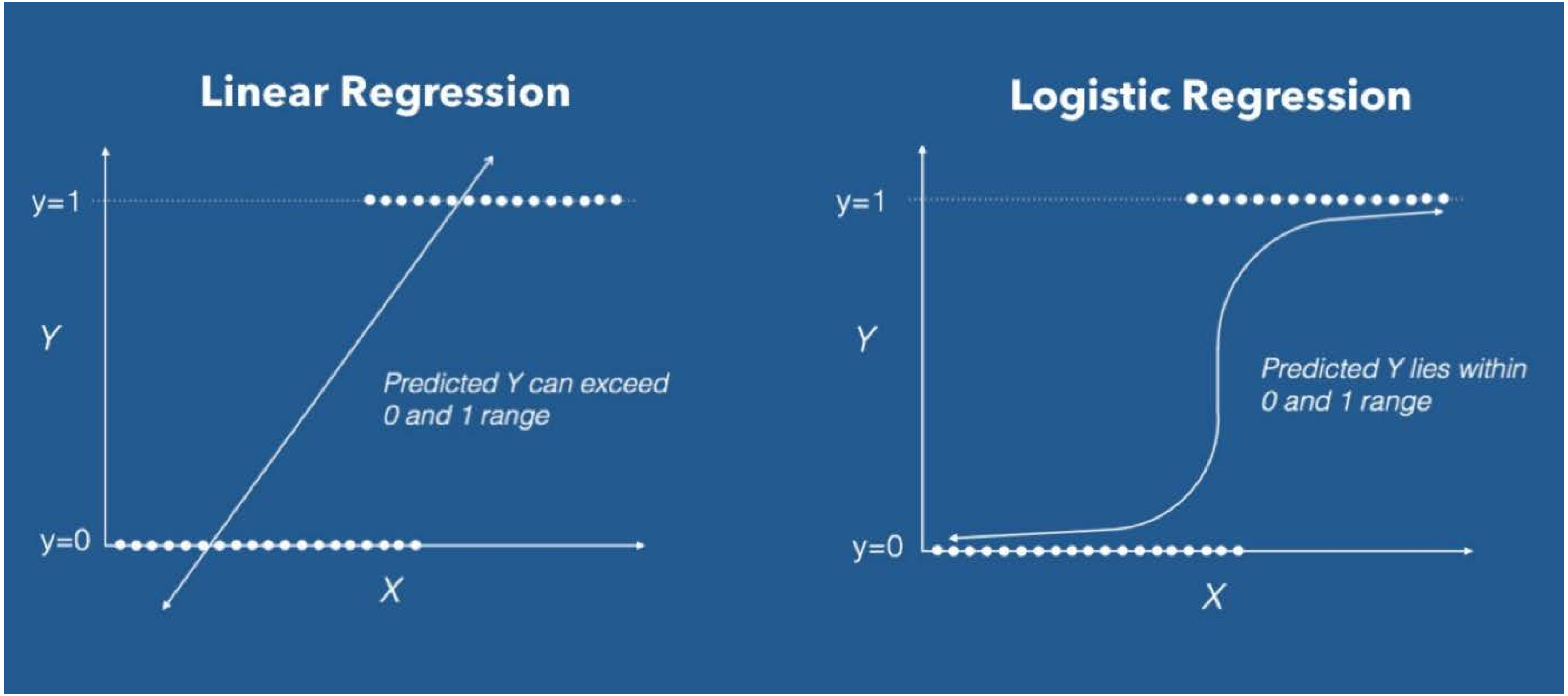
```
<AxesSubplot:xlabel='success_expect'>
```



PROPENSITY SCORE ESTIMATION

- Example of **covariate adjustment using the propensity score**
  - "The outcome variable is regressed on an indicator variable denoting **treatment status** and the **estimated propensity score**. "
  - $Y = \beta_0 + \beta_1 * T + \beta_2 * \hat{P}(x)$
- Even though, we have true propensity score $P(x)$, mechanism that assigns the treatment effect is **unknown** and we need to replace the true propensity score by an estimation score $\hat{\boldsymbol{P}}(\boldsymbol{x})$
- For Dichotomous (Binary) outcomes, we will be using **logistic regression** to estimate $\hat{P}(x)$
  - Range of propensity score: $0 \leq \hat{P}(x) \leq 1$, likelihood that subject will be given a specific treatment

LINEAR VS. LOGISTICS



PROCESS OF ESTIMATION

Changing into dummies

Changing categorical features into dummies

```
categ = ["ethnicity", "gender", "school_urbanicity"]
cont = ["school_mindset", "school_achievement", "school_ethnic_minority", "school_poverty", "school_size"]

data_with_categ = pd.concat([
    data.drop(columns=categ), # dataset without the categorical features
    pd.get_dummies(data[categ], columns=categ, drop_first=False)# categorical features converted to dummies
], axis=1)

print(data_with_categ.shape)
```

(10391, 32)

```
data_with_categ.sample(5, random_state=5)
```

ethnicity_13	ethnicity_14	ethnicity_15	gender_1	gender_2	school_urbanicity_0	school_urbanicity_1	school_urbanicity_2	school_urbanicity_3
0	0	0	0	1	0	1	0	0
1	0	0	1	0	0	0	0	0
0	0	0	0	1	1	0	0	0
0	1	0	1	0	0	0	0	0
0	0	0	1	0	0	1	0	0

Estimated propensity score

```
from sklearn.linear_model import LogisticRegression

T = 'intervention'
Y = 'achievement_score'
X = data_with_categ.columns.drop(['schoolid', T, Y])

ps_model = LogisticRegression(C=1e6).fit(data_with_categ[X], data_with_categ[T])

data_ps = data.assign(propensity_score=ps_model.predict_proba(data_with_categ[X])[:, 1])

data_ps[["intervention", "achievement_score", "propensity_score"]].head()
```

	intervention	achievement_score	propensity_score
0	1	0.277359	0.315490
1	1	-0.449646	0.263803
2	1	0.769703	0.344039
3	1	-0.121763	0.344039
4	1	1.526147	0.367797

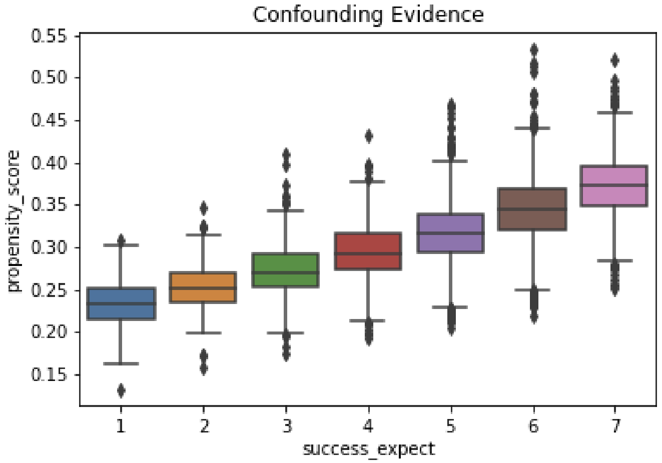
$$Log\left[\frac{Y}{1-Y}\right] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots + \beta_n X_n$$

Linear Regression using PSM

Success expect & Propensity score

```
sns.boxplot(x="success_expect", y="propensity_score", data=data_ps)

plt.title("Confounding Evidence");
```



Linear Regression with P.S

```
smf.ols("achievement_score ~ intervention + propensity_score", data=data_ps).fit().summary().tables[1]
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-3.0768	0.065	-47.055	0.000	-3.205	-2.949
intervention	0.3930	0.019	20.974	0.000	0.356	0.430
propensity_score	9.0547	0.200	45.308	0.000	8.663	9.446

Intervention Effect: 0.472 → 0.393

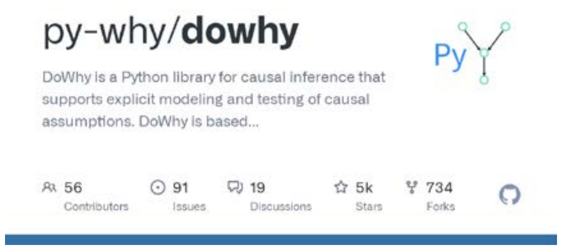
EASIER WAY TO IMPLEMENT C.I.

CAUSAL INFERENCE

- Python package provides various statistical methods
- Simple package used for basic causal analysis learning

CAUSALLIB

- Python package developed by IBM
- Provide causal analysis API unified with Scikit-Learn API



DOWHY

- Python package provides 4-step interface for causal inference
- Integration with EconML library

RESULT OF USING "CAUSALMODEL"

Actual ATE

```
confounders = data_with_categ.drop(columns=['achievement_score', 'intervention']).values
```

```
model = CausalModel(
    Y= data_ps["achievement_score"].values,
    D = data_ps["intervention"].values,
    X= confounders
)
```

```
model.est_via_matching(matches=1, bias_adj=True)

print(model.estimates)
```

```
Treatment Effect Estimates: Matching

                     Est.       S.e.          z       P>|z|      [95% Conf. int.]
-------------------------------------------------------------------------------
           ATE      0.382      0.023     16.327      0.000      0.336      0.428
           ATC      0.368      0.025     14.540      0.000      0.318      0.417
           ATT      0.412      0.025     16.206      0.000      0.362      0.462
```

Estimated ATE using PSM

```
model2 = CausalModel(
    Y= data_ps["achievement_score"].values,
    D = data_ps["intervention"].values,
    X= data_ps["propensity_score"].values
)
```

```
model2.est_via_matching(matches=1, bias_adj=True)

print(model2.estimates)
```

```
Treatment Effect Estimates: Matching

                     Est.       S.e.          z       P>|z|      [95% Conf. int.]
-------------------------------------------------------------------------------
           ATE      0.392      0.025     15.696      0.000      0.343      0.442
           ATC      0.385      0.027     14.031      0.000      0.331      0.439
           ATT      0.407      0.027     15.315      0.000      0.355      0.460
```

THANK YOU

https://matheusfacure.github.io/python-causality-handbook/11-Propensity-Score.html