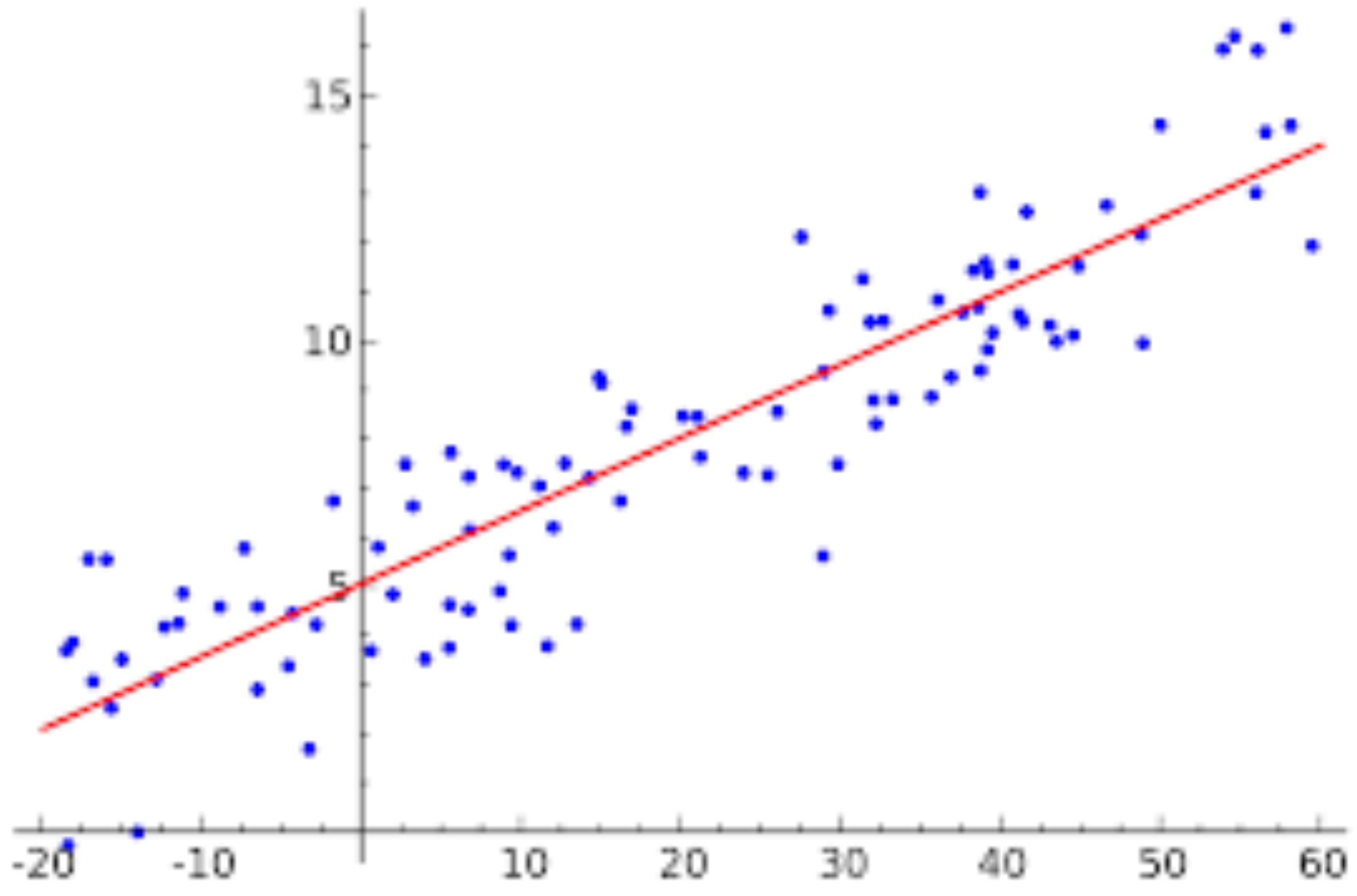
Business Experimentation and Causal Methods

Prof. Fradkin
Topic: Regression for Experiments



This Time

Using Regression for Experiments

1. Regression to measure the ATE.

2. Adding covariates to increase precision.

3. Avoiding ‘bad’ covariates.

Example Experiment: TutorGPT

- Suppose we’ve designed an app called TutorGPT, that helps students learn by using a chat bot.
- We give half the students the app, and the other half are in the control.
- We want to measure the effect on GPA.

Data

	gpa_last_year	treatment	gpa_this_year	treatment_factor
0	2.681864	1	2.280444	Treatment
1	3.323482	0	2.908015	Control
2	2.778256	1	2.744848	Treatment
3	3.199914	0	2.403374	Control
4	3.800534	1	3.362648	Treatment

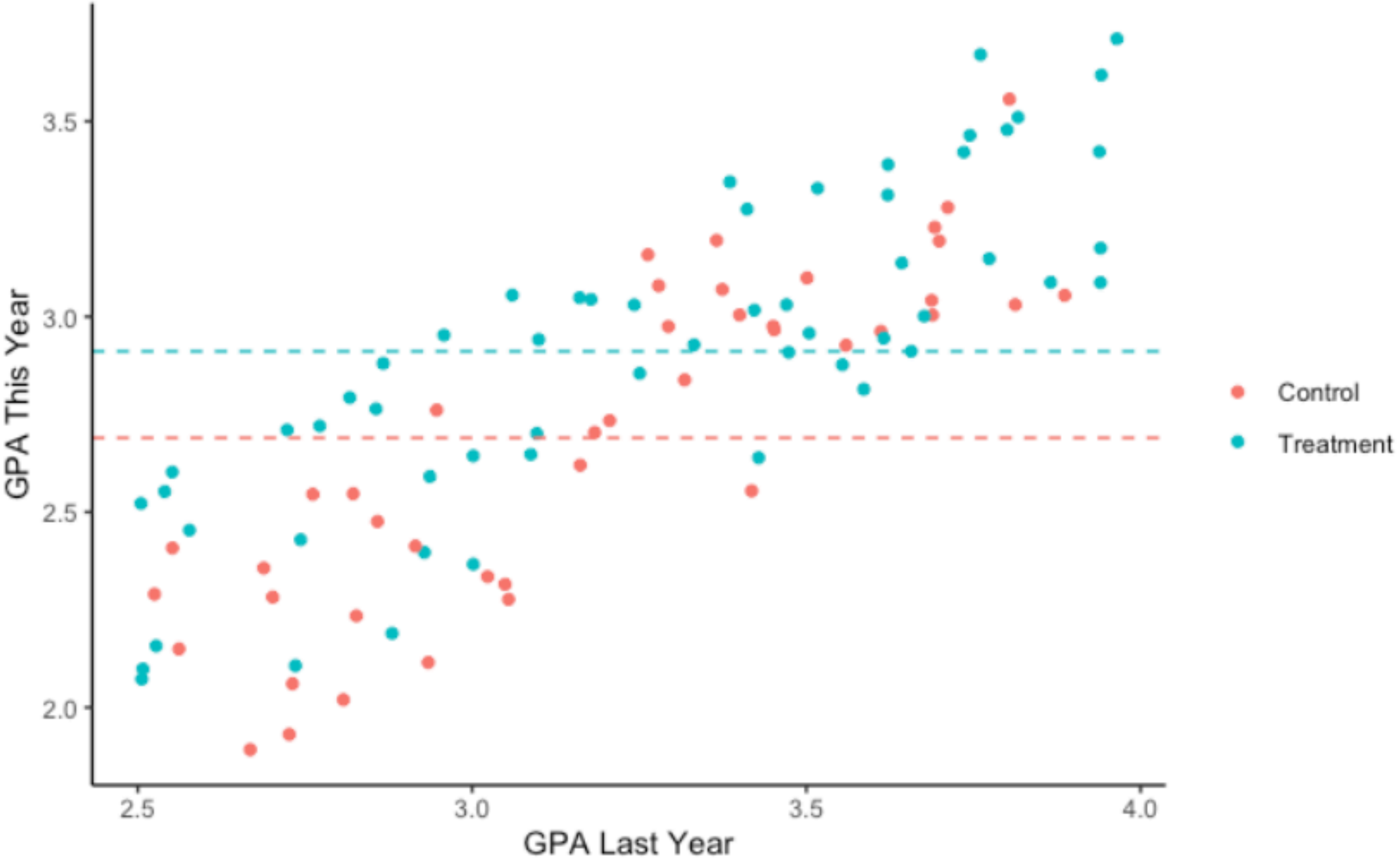
Estimate of the ATE

```
mean_ctr = data[data['treatment'] == 0]['gpa_this_year'].mean()
mean_trt = data[data['treatment'] == 1]['gpa_this_year'].mean()
print(mean_trt - mean_ctr)
```

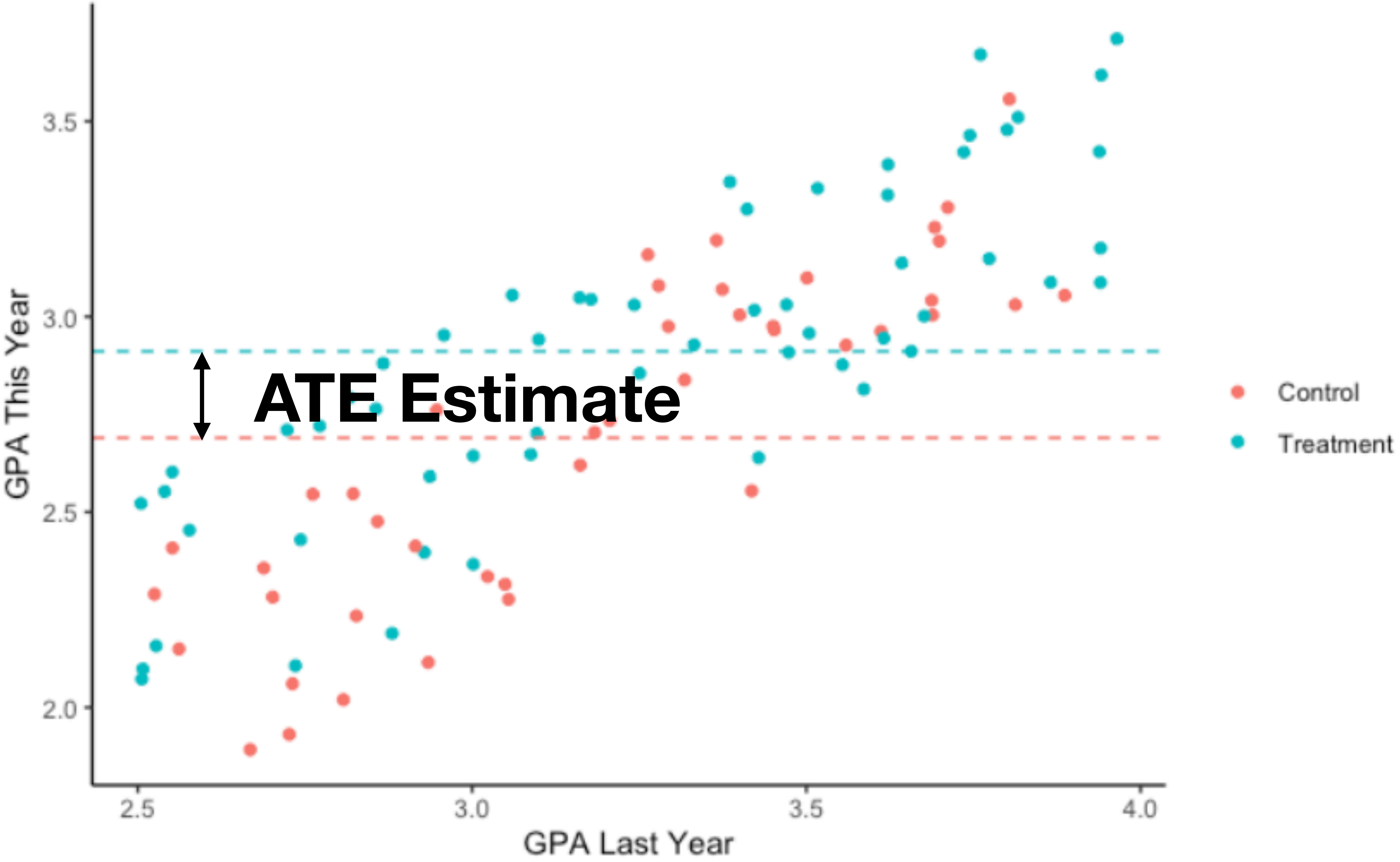
✓ 0.0s

```
0.14132693716381084
```

Each point is a student, blue points are treated.



Each point is a student, blue points are treated.



$$Outcome_i = a + bT_i + \epsilon_i$$

A regression finds the a and b which best fit the data.

Blue line is when T = 1 and red line when T = 0

The sm.OLS.from_formula function

- It take two parts: a formula and the data set to use.
- The formula separates the outcome variable from the explanatory variable with a ‘~’
- Make sure to specify ‘robust’ standard errors. From ‘cov_type’ = ‘HC1’.

Formula

Data to use

```
# Linear regression with statsmodels
import statsmodels.api as sm
from stargazer.stargazer import Stargazer

lm = sm.OLS.from_formula("gpa_this_year ~ treatment", data = data)
fit = lm.fit()

reg_robust = smf.ols('gpa_this_year ~ treatment', data=data).fit(cov_type='HC1')
```

Use the ‘Stargazer’ function to return regression estimates

```
Stargazer([fit, reg_robust])
```

✓ 0.0s

	Dependent variable: gpa_this_year	
	(1)	(2)
Intercept	2.724***	2.724***
	(0.053)	(0.054)
treatment	0.141	0.141*
	(0.085)	(0.084)
Observations	100	100
R^2	0.027	0.027
Adjusted R^2	0.017	0.017
Residual Std. Error	0.416 (df=98)	0.416 (df=98)
F Statistic	2.741 (df=1; 98)	2.807* (df=1; 98)
Note:	*p<0.1; **p<0.05; ***p<0.01	

Use the ‘Stargazer’ function to return regression estimates

```
Stargazer([fit, reg_robust])
```

✓ 0.0s

	Dependent variable: gpa_this_year	
	(1)	(2)
Intercept	2.724^{***}	2.724^{***}
	(0.053)	(0.054)
treatment	0.141	0.141^{*}
	(0.085)	(0.084)
Observations	100	100
R^2	0.027	0.027
Adjusted R^2	0.017	0.017
Residual Std. Error	0.416 (df=98)	0.416 (df=98)
F Statistic	2.741 (df=1; 98)	2.807^{*} (df=1; 98)
Note:		$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Coefficient

Use the 'Stargazer' function to return regression estimates

```
Stargazer([fit, reg_robust])
```

✓ 0.0s

	Dependent variable: gpa_this_year	
	(1)	(2)
Intercept	2.724***	2.724***
	(0.053)	(0.054)
treatment	0.141	0.141*
	(0.085)	(0.084)
Observations	100	100
R^2	0.027	0.027
Adjusted R^2	0.017	0.017
Residual Std. Error	0.416 (df=98)	0.416 (df=98)
F Statistic	2.741 (df=1; 98)	2.807* (df=1; 98)
Note:		*p<0.1; **p<0.05; ***p<0.01

Standard Error

Use the ‘Stargazer’ function to return regression estimates

```
Stargazer([fit, reg_robust])
```

✓ 0.0s

	Dependent variable: gpa_this_year	
	(1)	(2)
Intercept	2.724***	2.724***
	(0.053)	(0.054)
treatment	0.141	0.141*
	(0.085)	(0.084)
Observations	100	100
R^2	0.027	0.027
Adjusted R^2	0.017	0.017
Residual Std. Error	0.416 (df=98)	0.416 (df=98)
F Statistic	2.741 (df=1; 98)	2.807* (df=1; 98)
Note:		*p<0.1; **p<0.05; ***p<0.01

Significance Stars

Use the 'Stargazer' function to return regression estimates

```
Stargazer([fit, reg_robust])
```

✓ 0.0s

	Dependent variable: gpa_this_year	
	(1)	(2)
Intercept	2.724***	2.724***
	(0.053)	(0.054)
treatment	0.141	0.141*
	(0.085)	(0.084)
Observations	100	100
R^2	0.027	0.027
Adjusted R^2	0.017	0.017
Residual Std. Error	0.416 (df=98)	0.416 (df=98)
F Statistic	2.741 (df=1; 98)	2.807* (df=1; 98)
Note:		*p<0.1; **p<0.05; ***p<0.01

R-squared

Less nice way to output results, without Stargazer

```
fit.summary()
```

[22] ✓ 0.0s

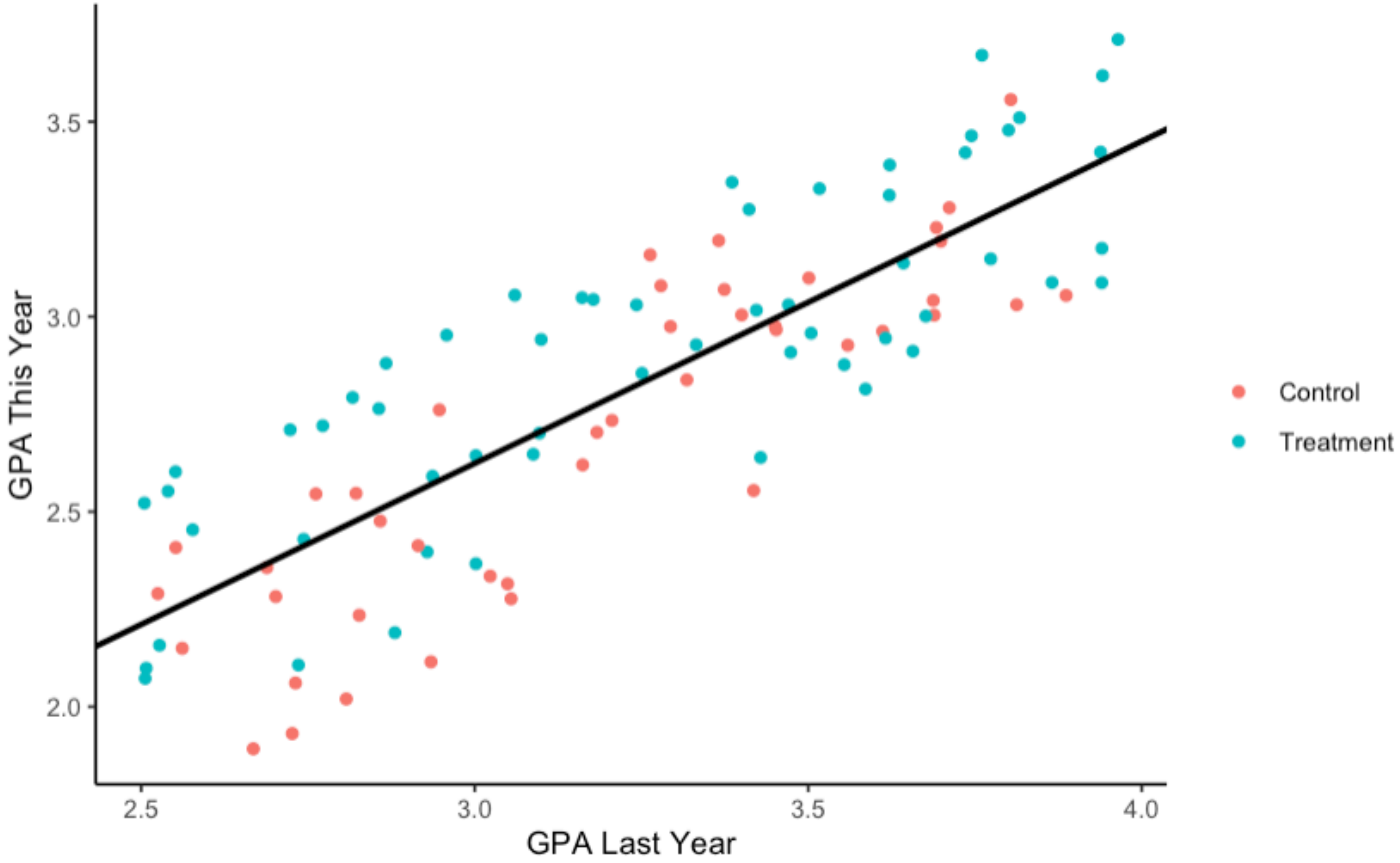
OLS Regression Results

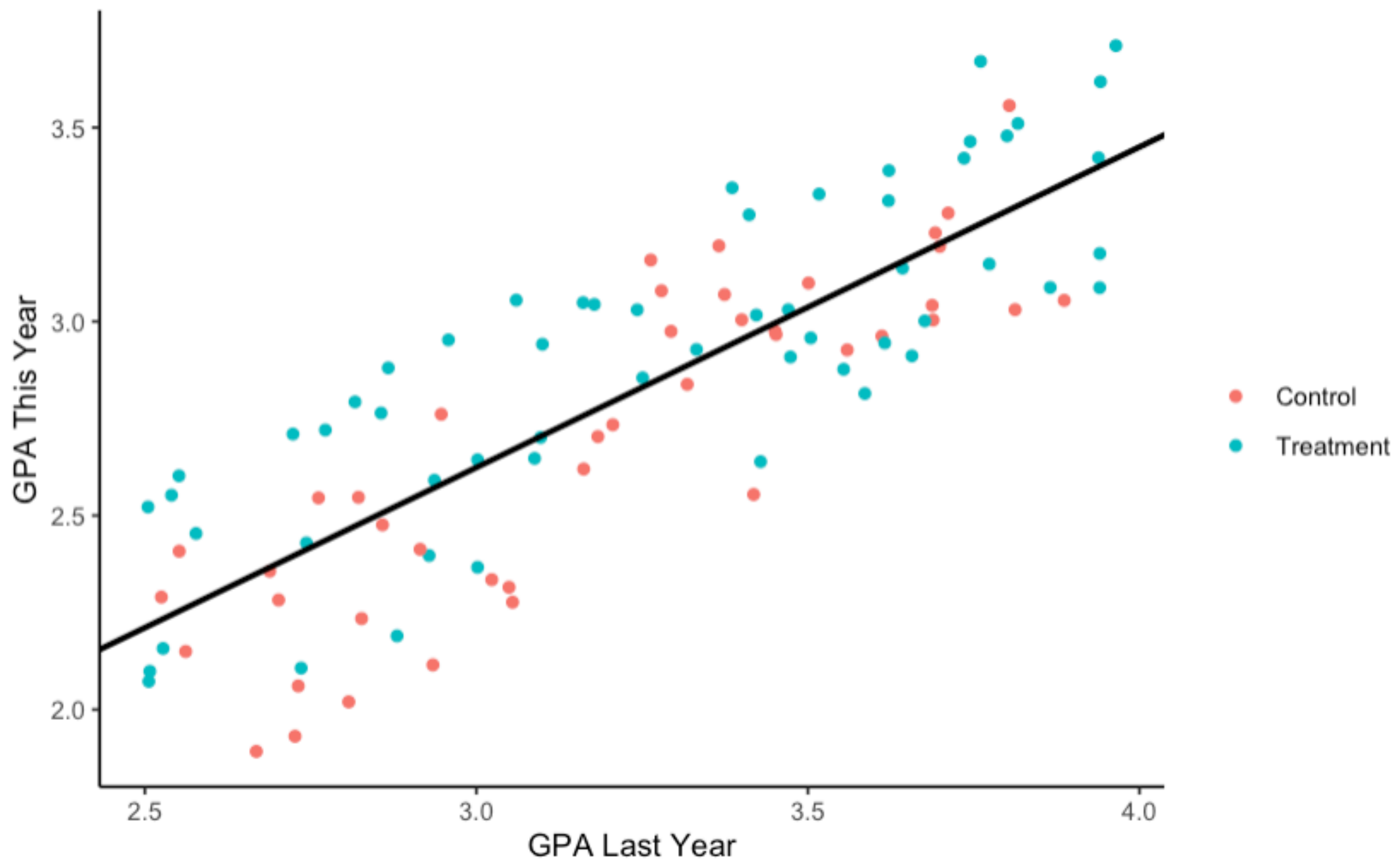
Dep. Variable:	gpa_this_year	R-squared:	0.027
Model:	OLS	Adj. R-squared:	0.017
Method:	Least Squares	F-statistic:	2.741
Date:	Thu, 08 Feb 2024	Prob (F-statistic):	0.101
Time:	10:06:25	Log-Likelihood:	-53.264
No. Observations:	100	AIC:	110.5
Df Residuals:	98	BIC:	115.7
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.7243	0.053	51.104	0.000	2.619	2.830
treatment	0.1413	0.085	1.656	0.101	-0.028	0.311

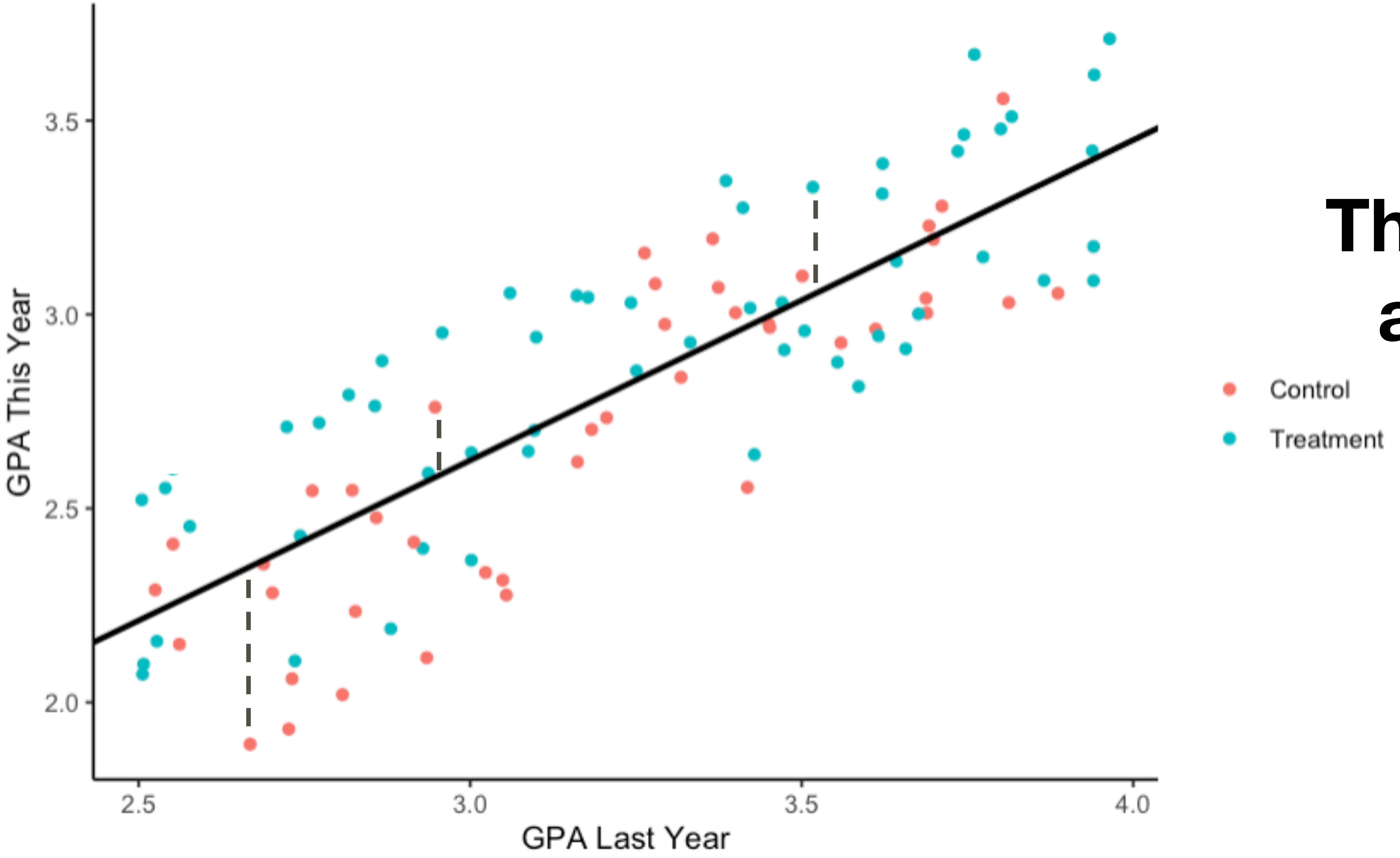
Omnibus:	7.212	Durbin-Watson:	2.122
Prob(Omnibus):	0.027	Jarque-Bera (JB):	3.324
Skew:	0.163	Prob(JB):	0.190
Kurtosis:	2.169	Cond. No.	2.44

Regress GPA this year on GPA last year





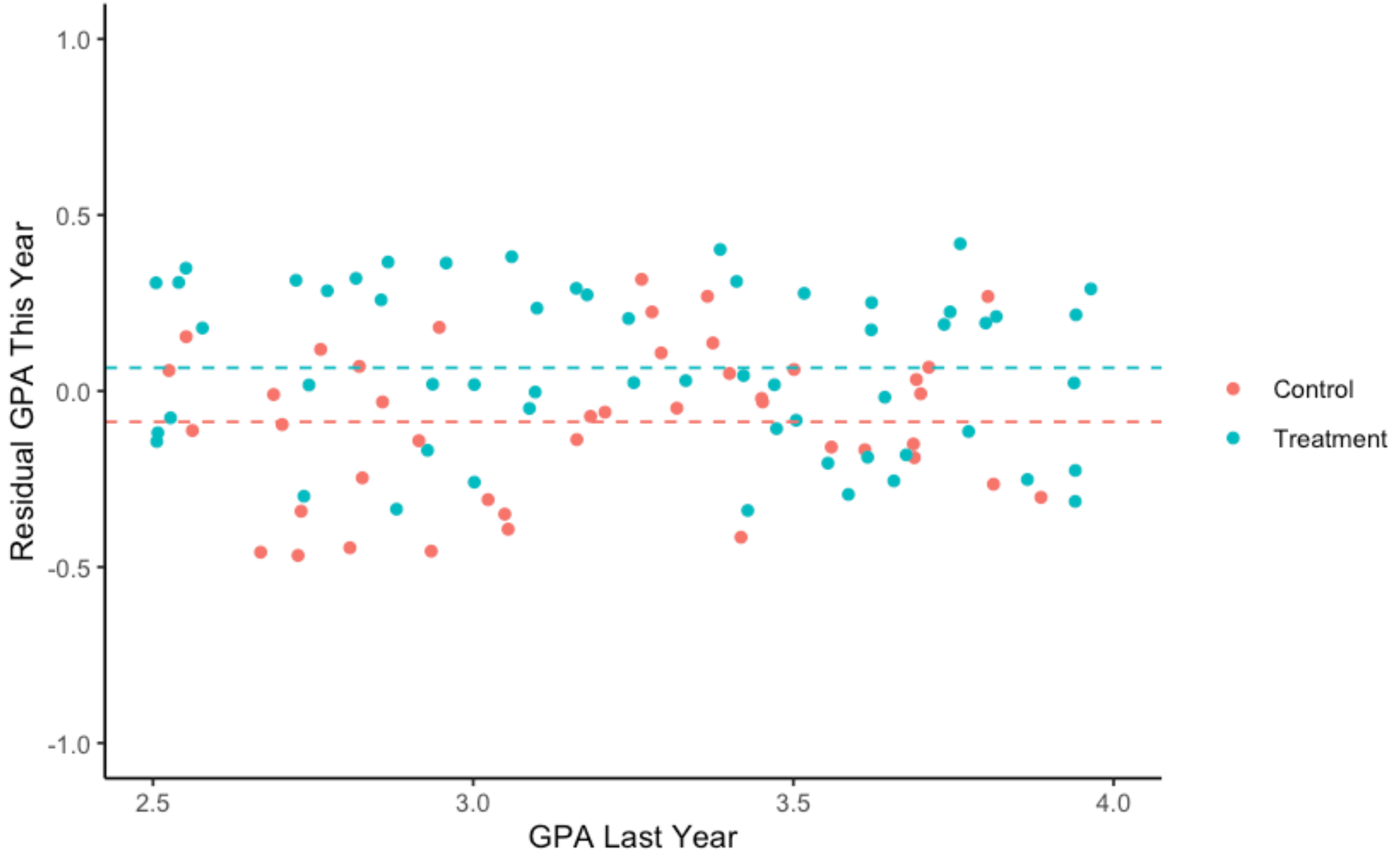
$$GPA\_This\_Year_i = \hat{a} + \hat{b}GPA\_Last\_Year_i + \hat{\epsilon}_i$$



The dashed vertical are the residuals.

$$GPA\_This\_Year_i = \hat{a} + \hat{b}GPA\_Last\_Year_i + \hat{\epsilon}_i$$

Plot residuals against last year's GPA



Much less variation in outcome!

Adjusting for a covariate in a regression

- We can add gpa last year as a covariate to a regression.

- This is also called ‘controlling’ for gpa last year.

- It is different from the ‘control group’.

Controlling for a covariate in a regression

We can add gpa last year as a covariate to a regression.

This is also called ‘controlling’ for gpa last year.

It is different than the ‘control group’.

No Covariates: $Outcome_i = a + bT_i + \epsilon_i$

One Covariate: $Outcome_i = a + bT_i + cX_i + \epsilon_i$

```python
reg_covariate_treat = smf.ols('gpa_this_year ~ treatment + gpa_last_year', data=data).fit(cov_type='HC1')
Stargazer([reg_robust, reg_covariate_treat])
```

✓ 0.0s

	Dependent variable: gpa_this_year	
	(1)	(2)
Intercept	2.724***	0.053
	(0.054)	(0.162)
gpa_last_year		0.831***
		(0.049)
treatment	0.141*	0.254***
	(0.084)	(0.044)
Observations	100	100
R^2	0.027	0.742
Adjusted R^2	0.017	0.737
Residual Std. Error	0.416 (df=98)	0.216 (df=97)
F Statistic	2.807* (df=1; 98)	145.837*** (df=2; 97)
Note:		*p<0.1; **p<0.05; ***p<0.01

Add last year's gpa as covariate.

```python
reg_covariate_treat = smf.ols('gpa_this_year ~ treatment + gpa_last_year', data=data).fit(cov_type='HC1')
Stargazer([reg_robust, reg_covariate_treat])
```

✓ 0.0s

	Dependent variable: gpa_this_year	
	(1)	(2)
Intercept	2.724***	0.053
	(0.054)	(0.162)
gpa_last_year		0.831***
		(0.049)
treatment	0.141*	0.254***
	(0.084)	(0.044)
Observations	100	100
R^2	0.027	0.742
Adjusted R^2	0.017	0.737
Residual Std. Error	0.416 (df=98)	0.216 (df=97)
F Statistic	2.807* (df=1; 98)	145.837*** (df=2; 97)
Note:		*p<0.1; **p<0.05; ***p<0.01

Reduces standard error

Controlling for a covariate in a regression

- It should not change our estimate of the treatment effect by much.
- **It can reduce our standard errors and p-values.**

Bad Covariates!

Some controls make the regression invalid for learning the treatment effect.

Example of a bad covariate

- Example: Suppose we randomly increased the price of cigarettes and we were interested in the effect on health.

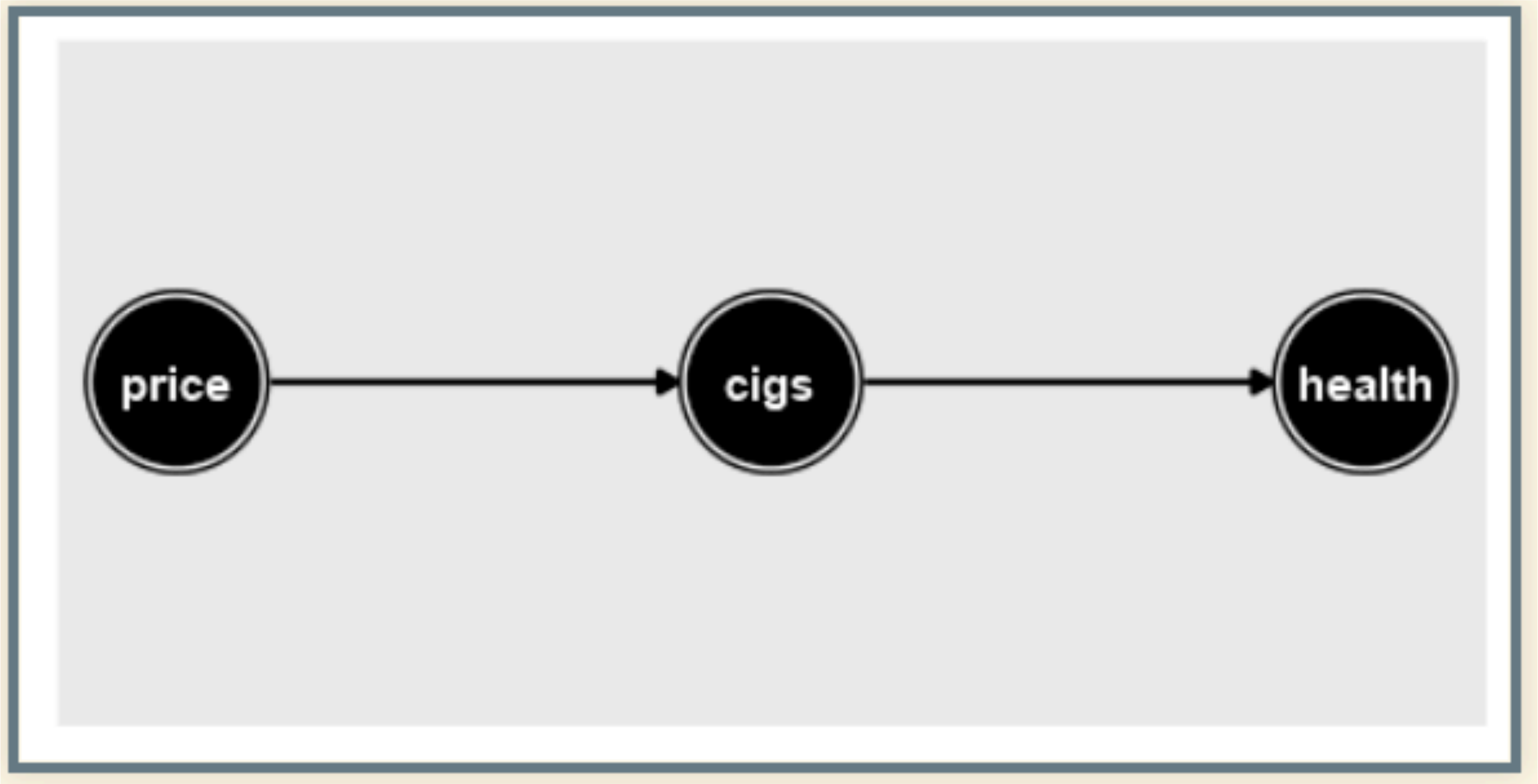
- The number of cigarettes smoked decreases and this improves our health.

- Adjusting for number of cigarettes smoked is bad, because the effect of the price increase on health is **caused** by the number of cigarettes smoked.

- So once we control for the number of cigarettes smoked, then a regression would say that the treatment had no effect. But this is obviously false, since it caused us to smoke fewer cigarettes.

Helpful to draw a diagram

Cigarettes block health. Bad covariate.



Controls for the learning app.

- Prior year's GPA: Good, since it is not affected by the treatment.
- How much you use the App: Bad since the treatment affects how much you use the app, and how much you use the app is correlated with this year's GPA.
- Student's SAT: Good, since it is not affected by the treatment.
- How many times you asked the professor a question: Bad, since it may be affected by having access to the app.

Summary: Regression

- More precision gain if covariate is predictive.
- Don’t include bad covariates (bad controls). These are measures that occur after the treatment.