Regression Logic

EC 320: Introduction to Econometrics

Winter 2022

Prologue

Housekeeping

Exercise 3 due this Wednesday!

Problem Set 1 solution available.

Problem Sets due dates changed

- Extra two days
- Due Monday instead of Friday starting Problem Set 2K

Midterm 1 next week (Wednesday)

Midterm review on Monday

Last Time

1. Fundamental problem of econometrics

2. Selection bias

3. Randomized control trials

Regression Logic

Regression

Economists often rely on (linear) regression for statistical comparisons.

- *"Linear"* is more flexible than you think.

Regression analysis helps us make *other things equal* comparisons.

- We can model the effect of X on Y while **controlling** for potential confounders.
- Forces us to be explicit about the potential sources of selection bias.
- Failure to control for confounding variables leads to **omitted-variable bias**, a close cousin of selection bias

Returns to Private College

Research Question: Does going to a private college instead of a public college increase future earnings?

- **Outcome variable:** earnings
- **Treatment variable:** going to a private college (binary)

Q: How might a private school education increase earnings?

Q: Does a comparison of the average earnings of private college graduates with those of public school graduates isolate the economic returns to private college education? Why or why not?

Returns to Private College

How might we estimate the causal effect of private college on earnings?

Approach 1: Compare average earnings of private college graduates with those of public college graduates.

- Prone to selection bias.

Approach 2: Use a matching estimator that compares the earnings of individuals the same admissions profiles.

- Cleaner comparison than a simple difference-in-means.
- Somewhat difficult to implement.
- Throws away data (inefficient).

Approach 3: Estimate a regression that compares the earnings of individuals with the same admissions profiles.

The Regression Model

We can estimate the effect of X on Y by estimating a **regression model**:

$$Y_i = \beta_0 + \beta_1 X_i + u_i$$

- Y_i is the outcome variable.
- X_i is the treatment variable (continuous).
- u_i is an error term that includes all other (omitted) factors affecting Y_i.
- β_0 is the **intercept** parameter.
- β_1 is the **slope** parameter.

Running Regressions

The intercept and slope are population parameters.

Using an estimator with data on X_i and Y_i, we can estimate a **fitted regression line**:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$$

- $\hat{Y}_i$ is the **fitted value** of Y_i.
- $\hat{\beta}_0$ is the **estimated intercept**.
- $\hat{\beta}_1$ is the **estimated slope**.

The estimation procedure produces misses called **residuals**, defined as $Y_i - \hat{Y}_i$.

Running Regressions

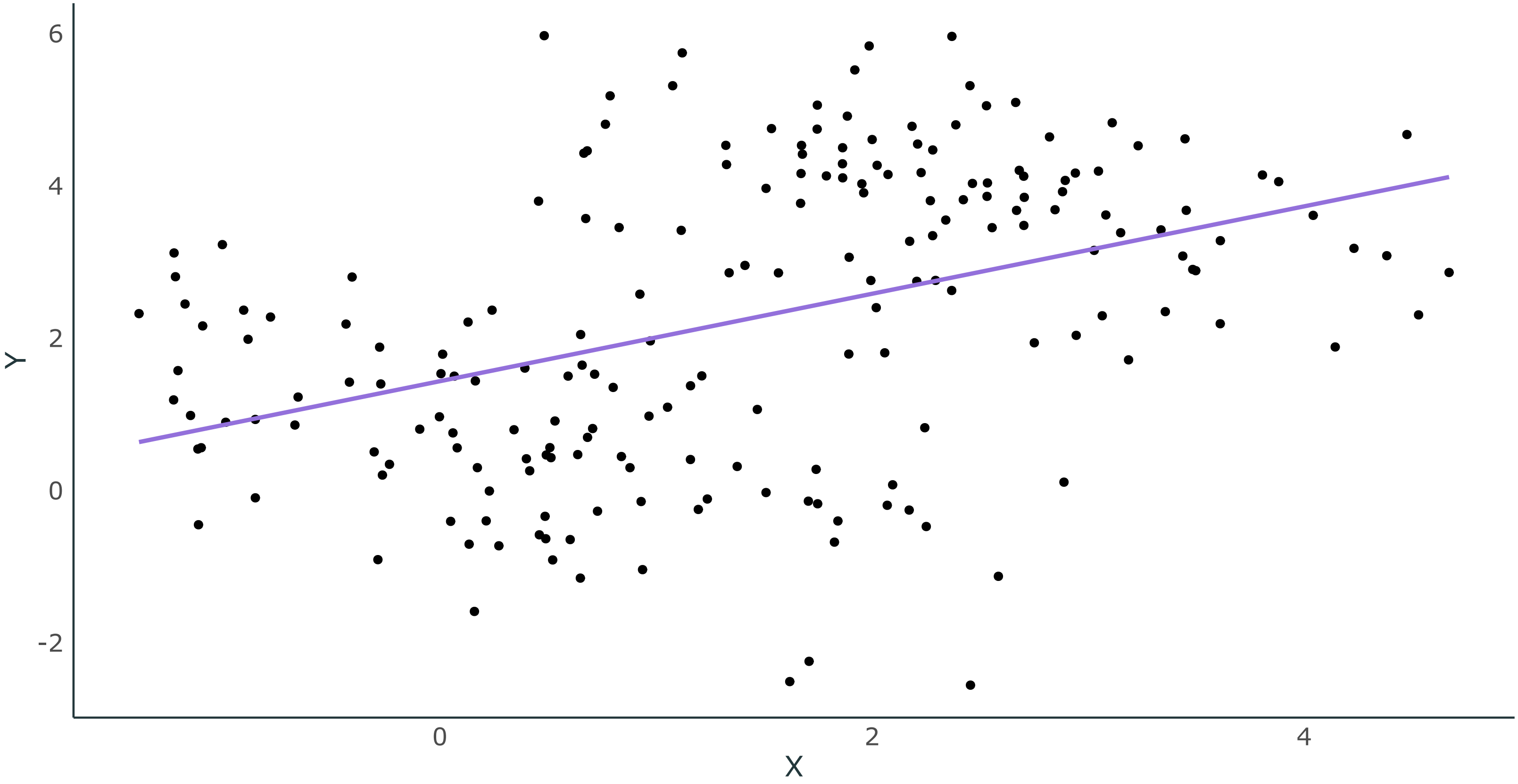
In practice, we estimate the regression coefficients using an estimator called **Ordinary Least Squares** (OLS).

- Picks estimates that make $\hat{Y}_i$ as close as possible to Y_i given the information we have on X and Y.

- We will dive into the weeds after the midterm.

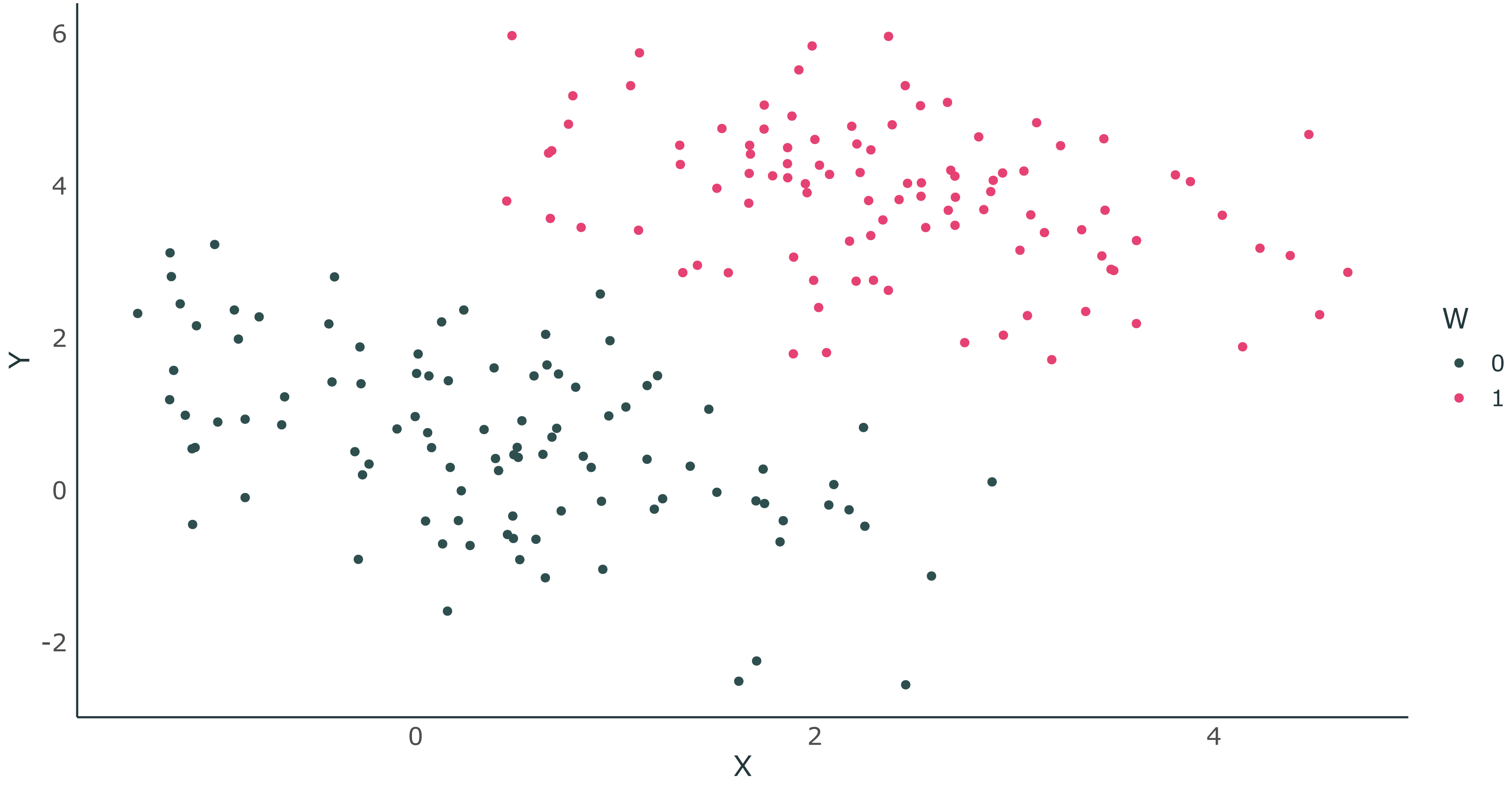
Running Regressions

OLS picks $\hat{\beta}_0$ and $\hat{\beta}_1$ that trace out the line of best fit. Ideally, we wound like to interpret the slope of the line as the causal effect of X on Y.



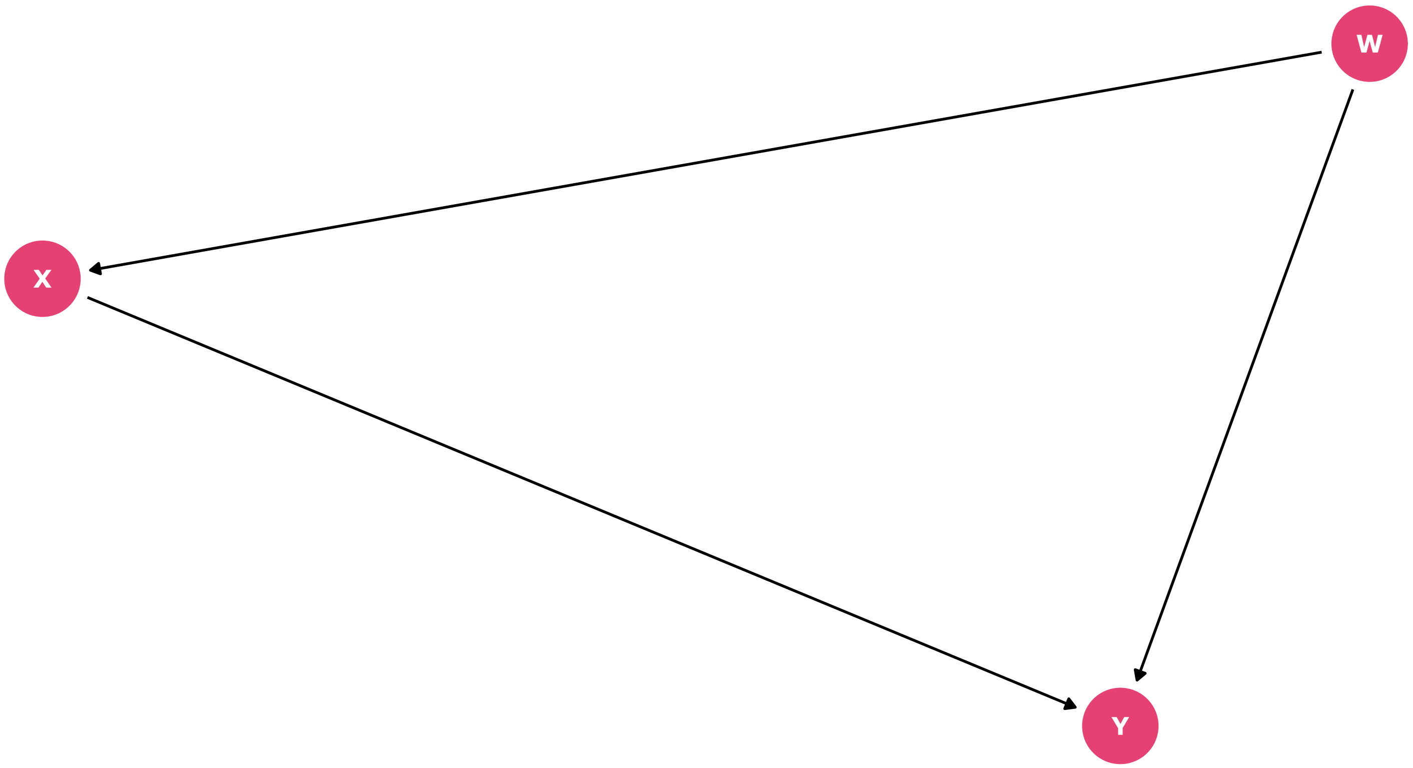
Confounders

However, the data are grouped by a third variable W. How would omitting W from the regression model affect the slope estimator?



Confounders

The problem with W is that it affects both Y and X. Without adjusting for W, we cannot isolate the causal effect of X on Y.



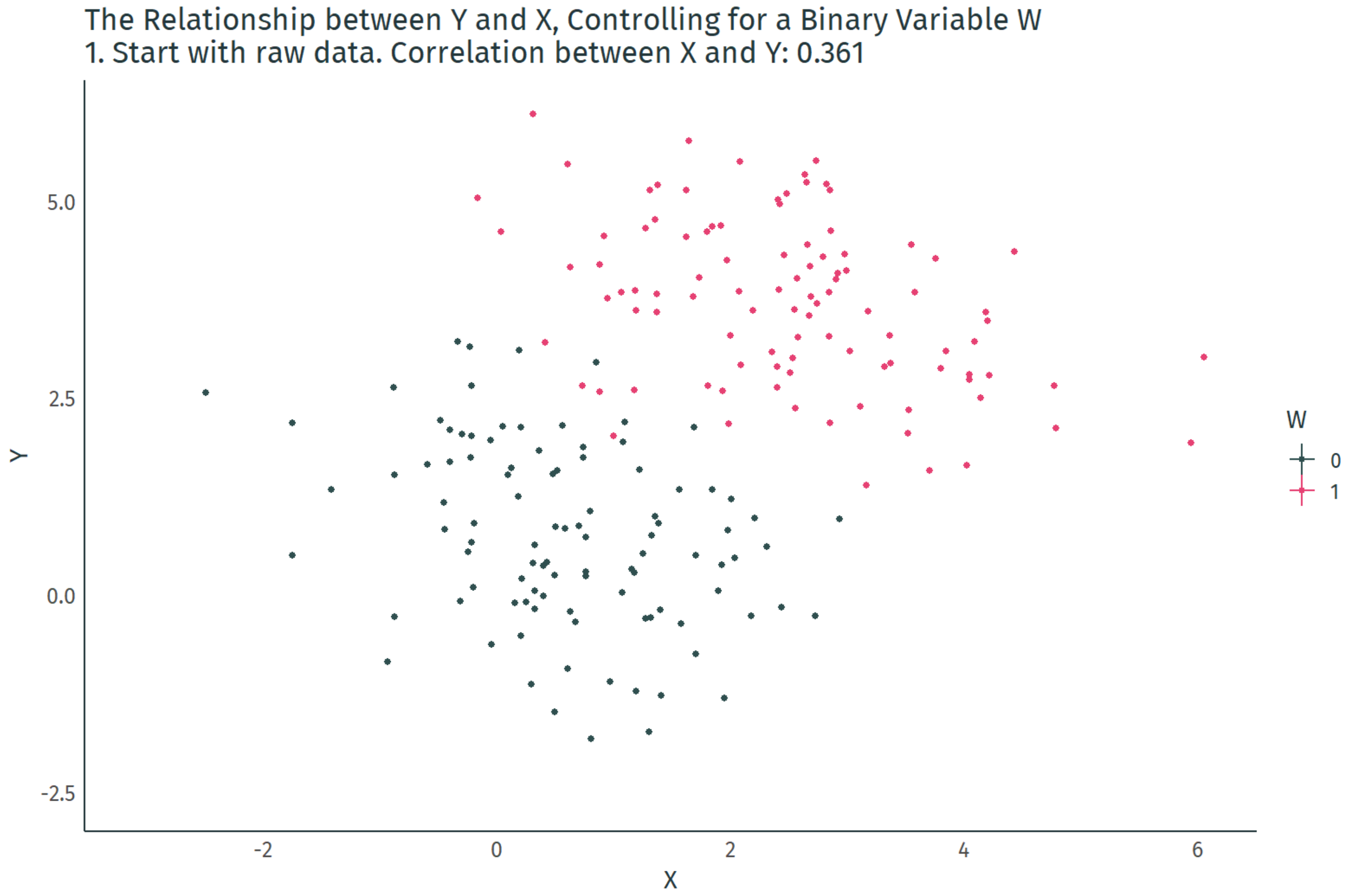
Controlling for Confounders

We can control for W by specifying it in the regression model:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 W_i + u_i$$

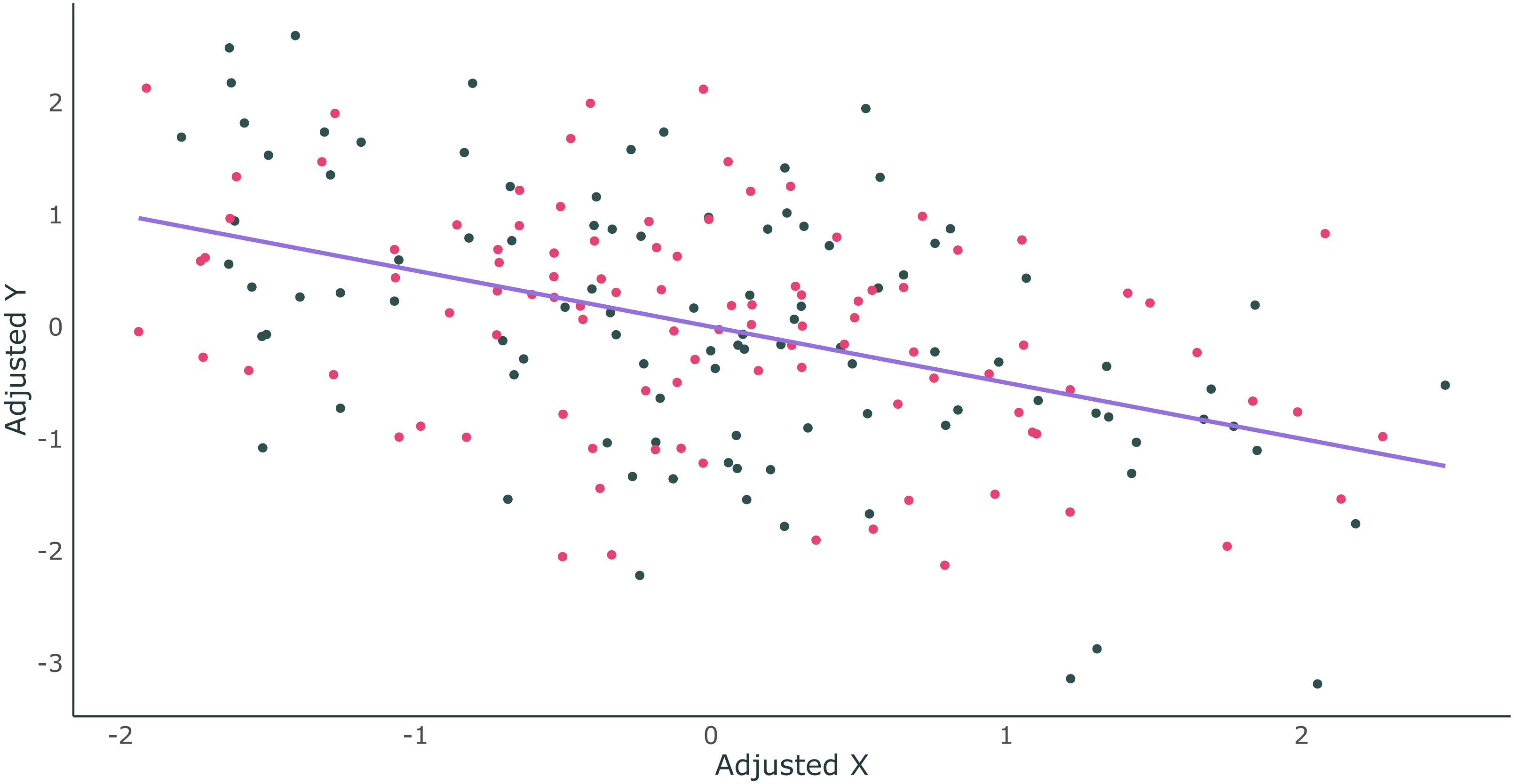
- W_i is a **control variable**.
- By including W_i in the regression, we can use OLS can difference out the confounding effect of W.
- **Note:** OLS doesn't care whether a right-hand side variable is a treatment or control variable, but we do.

Controlling for Confounders



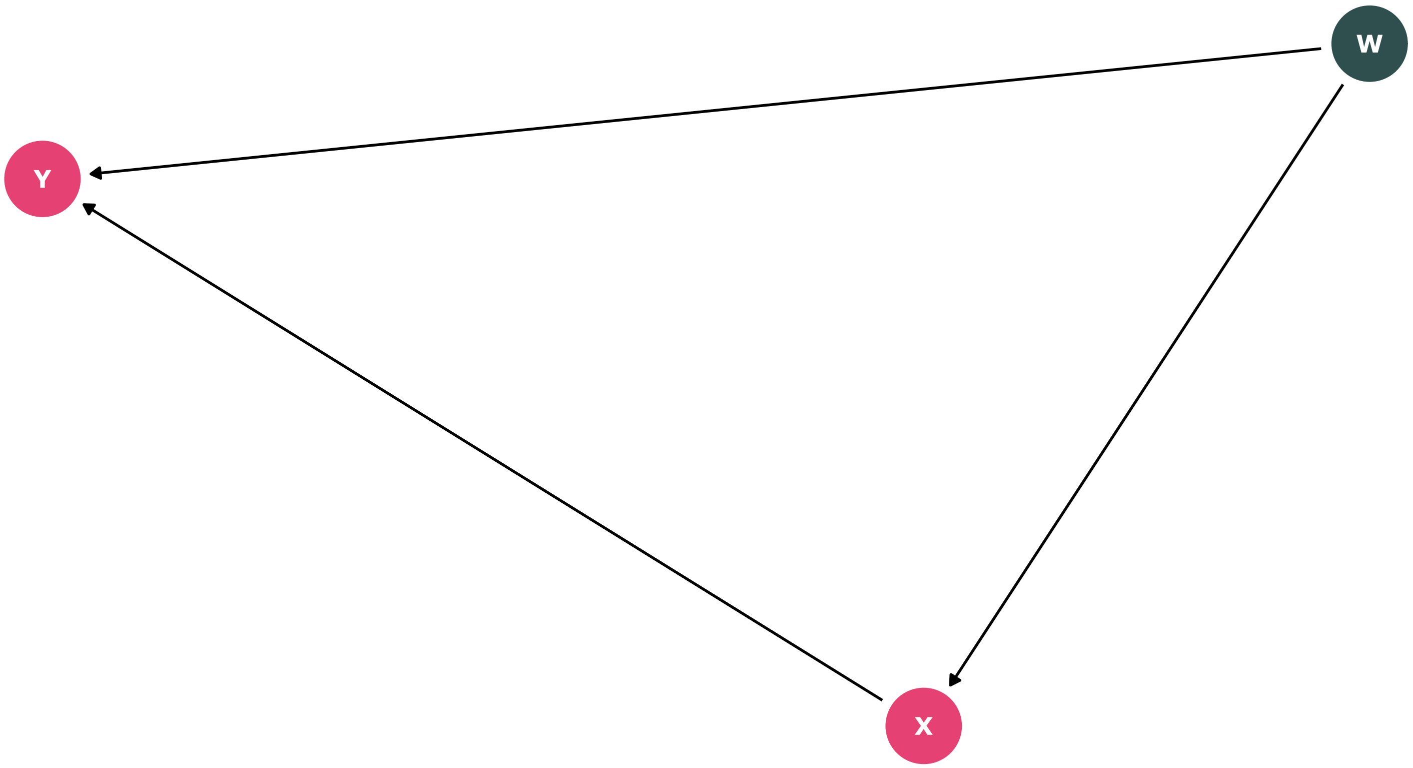
Controlling for Confounders

Controlling for W "adjusts" the data by **differencing out** the group-specific means of X and Y. **Slope of the estimated regression line changes!**



Controlling for Confounders

Can we interpret the estimated slope parameter as the causal effect of X on Y now that we've adjusted for W?



Controlling for Confounders

Example: Returns to schooling

Last class:

> **Q:** Could we simply compare the earnings those with more education to those with less?
> **A:** If we want to measure the causal effect, probably not.

What omitted variables should we worry about?

Controlling for Confounders

Example: Returns to schooling

Three regressions ***of*** wages ***on*** schooling.

Outcome variable: log(Wage)

Explanatory variable	**1**	**2**	**3**
Intercept	5.571	5.581	**5.695**
	(0.039)	(0.066)	**(0.068)**
Education	0.052	0.026	**0.027**
	(0.003)	(0.005)	**(0.005)**
IQ Score		0.004	**0.003**
		(0.001)	**(0.001)**
South			**-0.127**
			(0.019)

Omitted-Variable Bias

The presence of omitted-variable bias (OVB) precludes causal interpretation of our slope estimates.

We can back out the sign and magnitude of OVB by subtracting the slope estimate from a ***long*** regression from the slope estimate from a ***short*** regression:

$$\text{OVB} = \hat{\beta}_1^{\text{Short}} - \hat{\beta}_1^{\text{Long}}$$

Dealing with potential sources of OVB is one of the main objectives of econometric analysis!

Data and the `tidyverse`

Data

Experimental data

Data generated in controlled, laboratory settings.

Ideal for **causal identification**, but difficult to obtain in the social sciences.

- Intractable logistical problems
- Too expensive
- Morally repugnant

Experiments outside the lab: **randomized control trials** and **A/B testing**.

Data

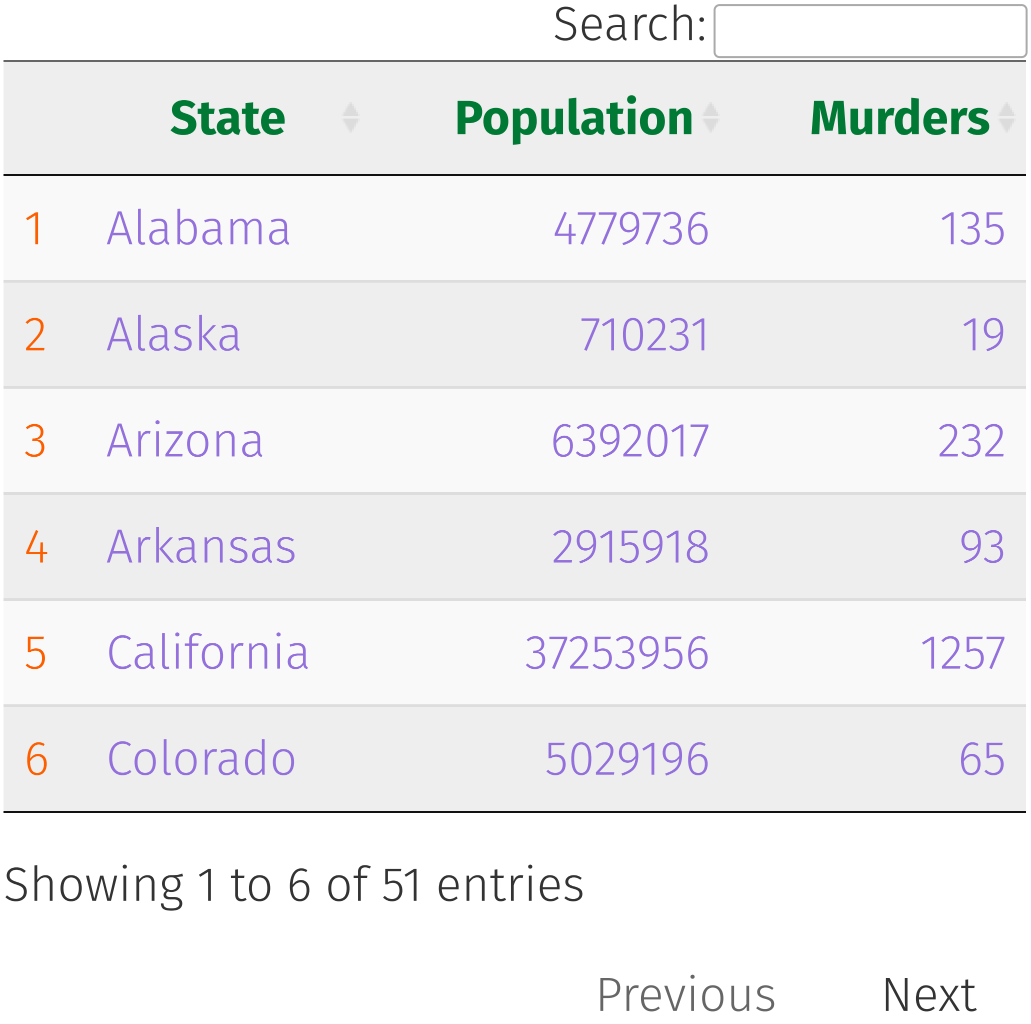
Observational data

Data generated in non-experimental settings.

- Surveys
- Censuses
- Administrative records
- Environmental data
- Financial and sales transactions
- Social media

Mainstay of economic research, but **poses challenges** to causal identification.

Tidy Data



Search:

	State	Population	Murders
1	Alabama	4779736	135
2	Alaska	710231	19
3	Arizona	6392017	232
4	Arkansas	2915918	93
5	California	37253956	1257
6	Colorado	5029196	65

Showing 1 to 6 of 51 entries

Rows represent **observations**.

Columns represent **variables**.

Each **value** is associated with an **observation** and a **variable**.

Cross Sectional Data

Sample of individuals from a population at a point in time.

Ideally, collected using **random sampling**.

- Random sampling + sufficient sample size = representative sample.
- Random sampling simplifies data analysis, but non-random samples are common (and difficult to work with).

Used extensively in applied microeconomics.*

Main focus of this course.

* Applied microeconomics = Labor, health, education, public finance, development, industrial organization, and urban economics.

Cross Sectional Data

Sample of US workers (Current Population Survey, 1976)

	Wage	Education	Tenure	Female?	Non-white?
1	3.1	11	0	1	0
2	3.24	12	2	1	0
3	3	11	0	0	0
4	6	8	28	0	0
5	5.3	12	2	0	0
6	8.75	16	8	0	0

Showing 1 to 6 of 526 entries

Previous 1 2 3 4 5 … 88 Next

Time Series Data

Observations of variables over time.

- Quarterly US GDP
- Annual US infant mortality rates
- Daily Amazon stock prices

Complication: Observations are not independent draws.

- GDP this quarter highly related to GDP last quarter.

Used extensively in empirical macroeconomics.

Requires more-advanced methods (EC 421 and EC 422).

Time Series Data

Number of US manufacturing strikes per month (Jan. 1968 to Dec. 1976)

	Period	Strikes	Output
1	1	5	0.01517
2	2	4	0.00997
3	3	6	0.0117
4	4	16	0.00473
5	5	5	0.01277
6	6	8	0.01138

Showing 1 to 6 of 108 entries

Previous 1 2 3 4 5 … 18 Next

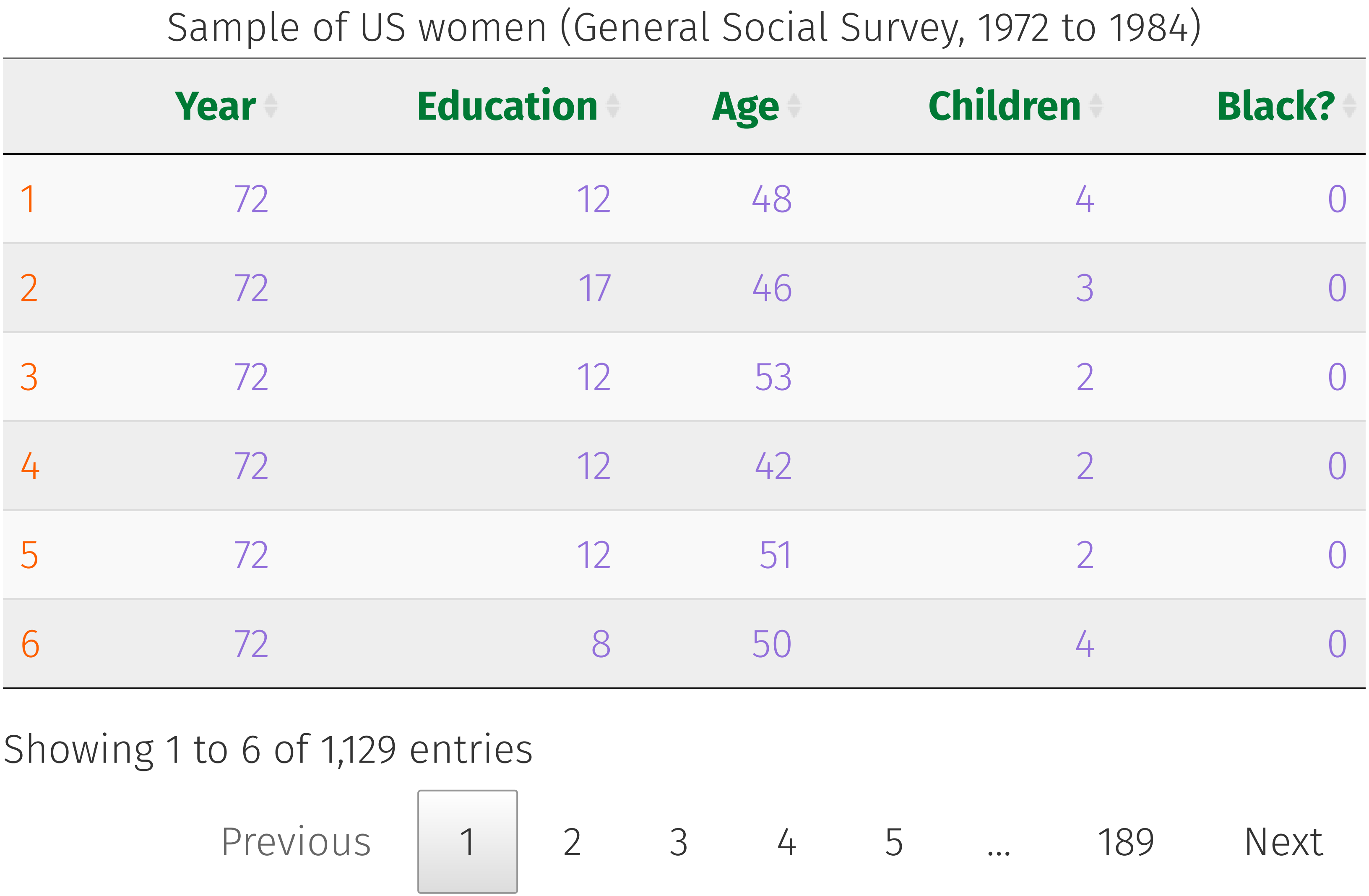
Pooled Cross Sectional Data

Cross sections from different points in time.

Useful for studying policy changes and relationship that change over time.

Requires more-advanced methods (EC 421 and many 400-level applied micro classes).

Pooled Cross Sectional Data



Sample of US women (General Social Survey, 1972 to 1984)

	Year	Education	Age	Children	Black?
1	72	12	48	4	0
2	72	17	46	3	0
3	72	12	53	2	0
4	72	12	42	2	0
5	72	12	51	2	0
6	72	8	50	4	0

Showing 1 to 6 of 1,129 entries

Panel or Longitudinal Data

Time series for each cross-sectional unit.

- Example: daily attendance data for a sample of students.

Difficult to collect, but useful for causal identification.

- Can control for *unobserved* characteristics.

Requires more-advanced methods (EC 421 and many 400-level applied micro classes).

Panel or Longitudinal Data

Panel of US workers (National Longitudinal Survey of Youth, 1980 to 1987)

	ID	Year	Experience	log(Wage)	Union
1	13	1980	1	1.2	no
2	13	1981	2	1.85	yes
3	13	1982	3	1.34	no
4	13	1983	4	1.43	no
5	13	1984	5	1.57	no
6	13	1985	6	1.7	no

Showing 1 to 6 of 4,360 entries

Previous 1 2 3 4 5 … 727 Next

Tidy Data?

	worker_id	year	variable	value
1	13	1980	educ	14
2	13	1981	educ	14
3	13	1982	educ	14
4	13	1983	educ	14
5	13	1984	educ	14
6	13	1985	educ	14

Showing 1 to 6 of 21,800 entries

Previous 1 2 3 4 5 … 3,634 Next

Messy Data

Analysis-ready datasets are rare. Most data are "messy."

The focus of this class is data analysis, but **data wrangling** is a non-trivial part of a data scientist/analyst's job.

R has a suite of packages that facilitate data wrangling.

- `readr`, `tidyr`, `dplyr`, `ggplot2` + others.
- Known collectively as the `tidyverse`.

tidyverse

The `tidyverse`: A package of packages

`readr`: Functions to import data.

`tidyr`: Functions to reshape messy data.

`dplyr`: Functions to work with data.

`ggplot2`: Functions to visualize data.

Workflow

Step 1: Load packages with `pacman`

```r
library(pacman)
p_load(tidyverse)
```

If the `tidyverse` hasn't already been installed, `p_load` will install it.

Loading the `tidyverse` automatically loads `readr`, `tidyr`, `dplyr`, `ggplot2`, and a few other packages.

Workflow

Step 2: Import data with `readr`

```r
workers <- read_csv("03-example_data.csv")
```

CSV files are a common non-proprietary format for storing tabular data.

The `read_csv` function imports CSV (comma-separated values) files.

- Converts the CSV file to a `tibble`, the `tidyverse` version of a `data.frame`.

Workflow

Step 3: Reshape data with `tidyr`

Variables are stored in rows instead of columns:

```
#> # A tibble: 21,800 × 4
#>    worker_id  year variable value
#>        <dbl> <dbl> <chr>    <dbl>
#>  1        13  1980 educ        14
#>  2        13  1981 educ        14
#>  3        13  1982 educ        14
#>  4        13  1983 educ        14
#>  5        13  1984 educ        14
#>  6        13  1985 educ        14
#>  7        13  1986 educ        14
#>  8        13  1987 educ        14
#>  9        17  1980 educ        13
#> 10        17  1981 educ        13
#> # … with 21,790 more rows
```

Workflow

Step 3: Reshape data with `tidyr`

Make the data tidy by using the `spread` function:

```r
workers <- workers %>%
  spread(key = variable, value = value)
```

Note the use of the **pipe operator**.

- `%>%` = *"and then."*
- Chains multiple commands together without having to define intermediate objects.

Workflow

Step 3: Reshape data with `tidyr`

The result:

```
#> # A tibble: 4,360 × 7
#>    worker_id  year black earnings  educ exper union
#>        <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>
#>  1        13  1980     0    8850.    14     1     0
#>  2        13  1981     0   14800.    14     2     1
#>  3        13  1982     0   11278.    14     3     0
#>  4        13  1983     0   12409.    14     4     0
#>  5        13  1984     0   14734.    14     5     0
#>  6        13  1985     0   15676.    14     6     0
#>  7        13  1986     0    1457.    14     7     0
#>  8        13  1987     0   14013.    14     8     0
#>  9        17  1980     0   13274.    13     4     0
#> 10        17  1981     0   12800.    13     5     0
#> # … with 4,350 more rows
```

Workflow

Step 4: Manipulate data with `dplyr`

Generate new variables with `mutate`:

```r
workers <- workers %>%
  mutate(union = ifelse(union == 1, "Yes", "No"))
```

Before, `union` was a binary variable equal to 1 if the worker is in a union or 0 if otherwise.

Now `union` is a character variable.

Workflow

Step 4: Manipulate data with `dplyr`

The result:

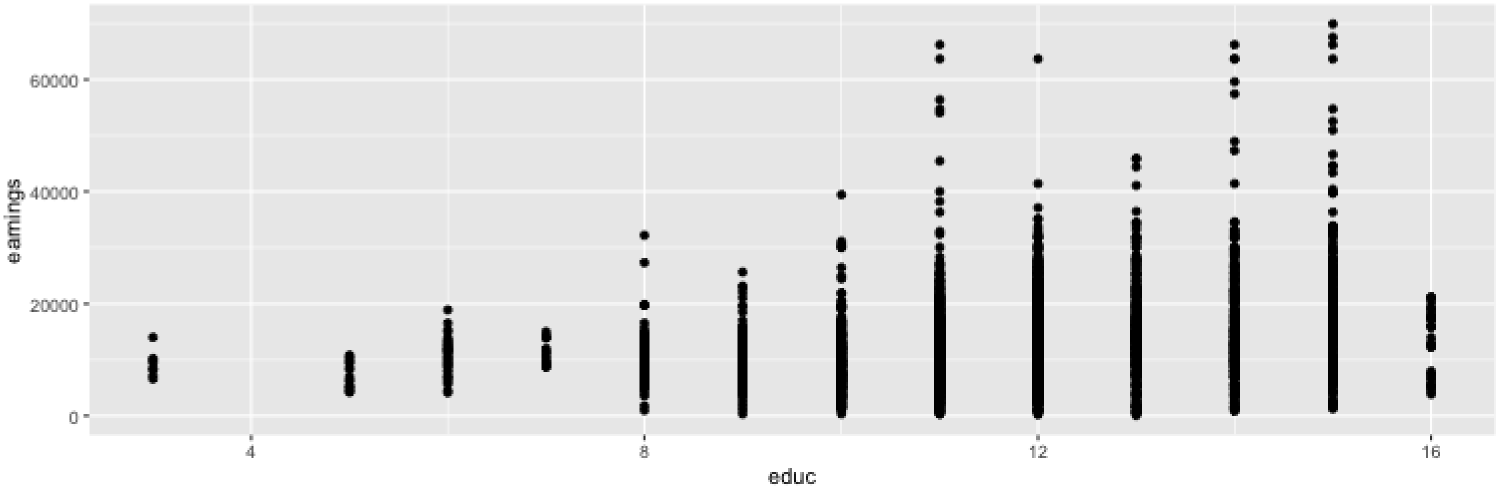
```
#> # A tibble: 4,360 × 7
#>    worker_id  year black earnings  educ exper union
#>        <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl> <chr>
#>  1        13  1980     0    8850.    14     1 No
#>  2        13  1981     0   14800.    14     2 Yes
#>  3        13  1982     0   11278.    14     3 No
#>  4        13  1983     0   12409.    14     4 No
#>  5        13  1984     0   14734.    14     5 No
#>  6        13  1985     0   15676.    14     6 No
#>  7        13  1986     0    1457.    14     7 No
#>  8        13  1987     0   14013.    14     8 No
#>  9        17  1980     0   13274.    13     4 No
#> 10        17  1981     0   12800.    13     5 No
#> # … with 4,350 more rows
```

Workflow

Step 6: Visualize and analyze data with `ggplot2`

How are education and earnings correlated?

```
workers %>%
  ggplot(aes(x = educ, y = earnings)) +
  geom_point()
```



Workflow

Step 6: Visualize and analyze data with `ggplot2`

How are education and earnings correlated?

Can also use the `cor` function from `base` R:

```
cor(workers$educ, workers$earnings)
```

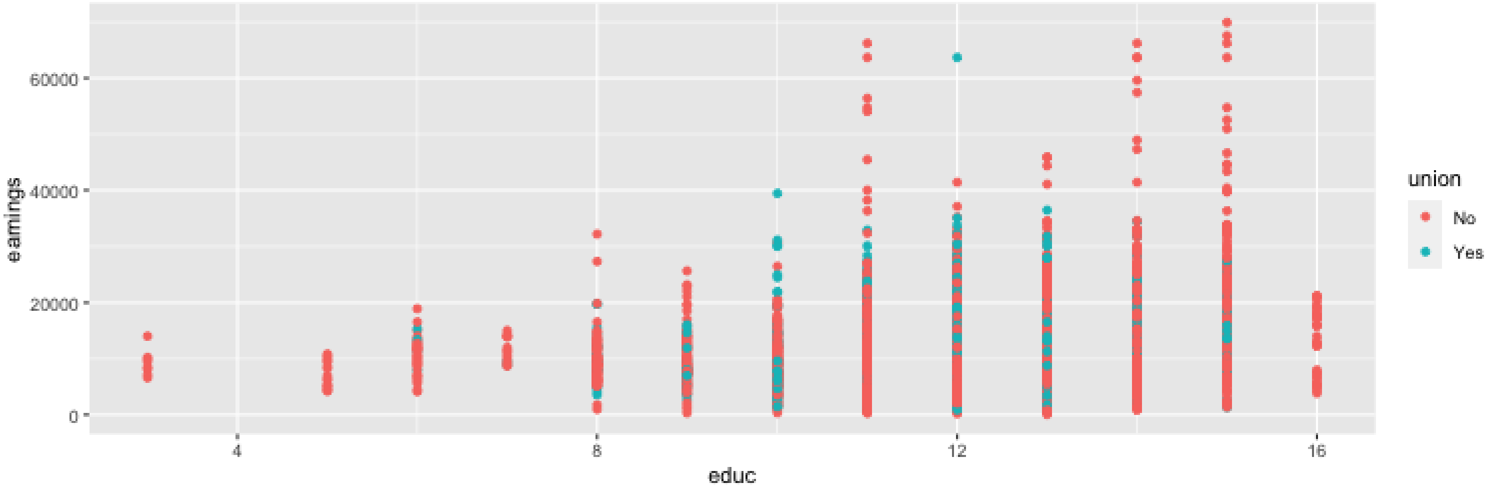
```
#> [1] 0.2685563
```

Workflow

Step 6: Visualize and analyze data with `ggplot2`

How are education and earnings correlated?

```
workers %>%
  ggplot(aes(x = educ, y = earnings, color = union)) +
  geom_point()
```

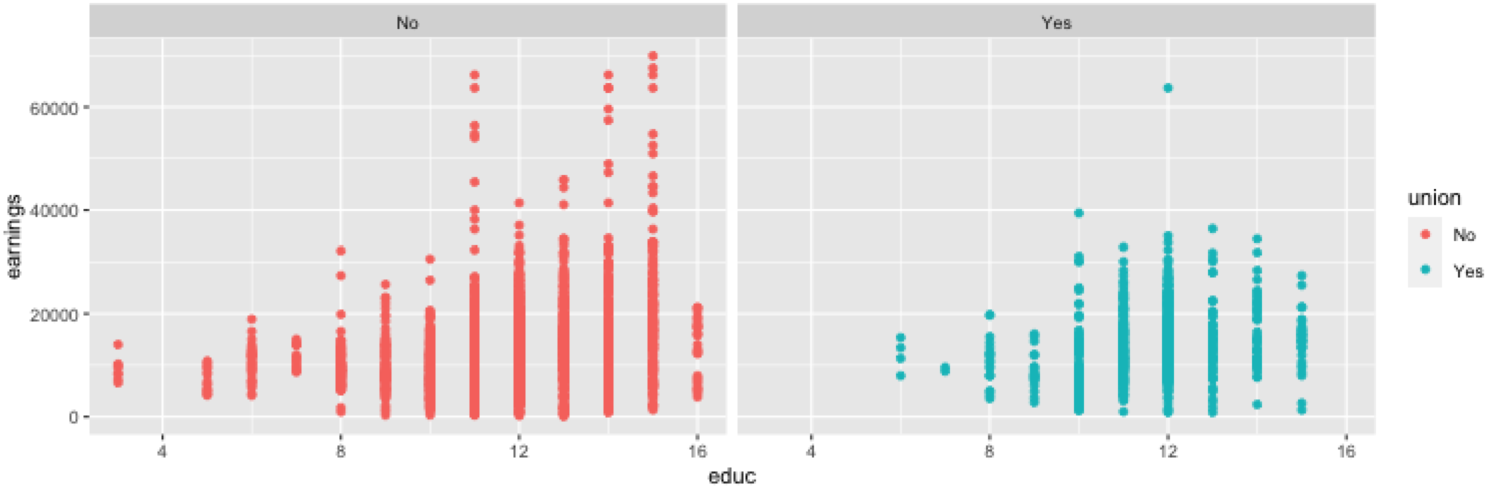


Workflow

Step 6: Visualize and analyze data with `ggplot2`

How are education and earnings correlated?

```
workers %>%
  ggplot(aes(x = educ, y = earnings, color = union)) +
  geom_point() +
  facet_grid(~union)
```



Workflow

Step 6: Visualize and analyze data with `ggplot2`

How are education and earnings correlated?

Can **subset** the data to get group-specific correlations:

```
workers_union <- workers %>%
  filter(union == "Yes")
cor(workers_union$educ, workers_union$earnings)
```

```
#> [1] 0.211482
```

```
workers_nounion <- workers %>%
  filter(union == "No")
cor(workers_nounion$educ, workers_nounion$earnings)
```

```
#> [1] 0.2809786
```

Why Bother?

Q: Why not just use **`MS Excel`** for data wrangling?

A: Reproducibility

- Easier to retrace your steps with R.

A: Portability

- Easy to re-purpose R code for new projects.

A: Scalability

- `Excel` chokes on big datasets.

A: R Saves time (eventually)

- Lower marginal costs in exchange for higher fixed costs.

Further Reading

1. Tidy Data by Hadley Wickham (creator of the `tidyverse`)

2. Cheatsheets