# Data Science for Economists

## Control variables

Kyle Coombs
Bates College | ECON/DCS 368

# Table of contents

# Prologue

# Prologue

- This week we are starting to think about causal inference
- Today, we're going to explore endogeneity a little bit
- We'll talk about how to solve it using *controls*
- As a warning: this approach is rarely the best approach to causal inference
- But it is a helpful starting point

# Attribution

These slides are adapted from slides by Nick Huntington-Klein on control variables, omitted variable bias, and endogeneity.

# Questions?

- Ask questions about course content, problem sets, etc.
- I am trying to build this step into future lectures

# Endogeneity and omitted variable bias

# Endogeneity vs. Exogeneity

- Last time I introduced **exogeneity** as a property of a variable in a model
- I suggested a new saying: Correlation plus **exogeneity** is causation.
- **Endogeneity** is the opposite of **exogeneity**
- We believe that our true model looks like this:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

- Where $\varepsilon$ is *everything that determines* $Y$ *other than* $X$
- If $X$ is related to some of those things, we have **endogeneity**
- Estimating the above model by OLS, it will mistake the effect of those *other* things for the effect of $X$, and our estimate of $\hat{\beta}_1$ won't represent the true $\beta_1$ no matter how many observations we have

# Endogeneity Recap

- For example, the model

$$IceCreamEating = \beta_0 + \beta_1 ShortsWearing + \varepsilon$$

- The true $\beta_1$ is probably $0$. But since $Temperature$ is in $\varepsilon$ and $Temperature$ is related to $ShortsWearing$, OLS will mistakenly assign the effect of $Temperature$ to the effect of $ShortsWearing$, making it look like there's a positive effect when there isn't one

- If $Temperature$ hangs around $ShortsWearing$, but OLS doesn't know about it, OLS will give $ShortsWearing$ all the credit for $Temperature$'s impact on $IceCreamEating$

- Here we're mistakenly finding a positive effect when the truth is $0$, but it could be anything - negative effect when truth is $0$, positive effect when the truth is a bigger/smaller positive effect, negative effect when truth is positive, etc. etc.

# Control variables

# To the Rescue

- One way we can solve this problem is through the use of *control variables*
- What if $Temperature$ *weren't* in $\varepsilon$? Then we'd be fine! OLS would know how to separate out its effect from the $ShortsWearing$ effect. How do we take it out? Just put it in the model directly!

$$IceCreamEating = \beta_0 + \beta_1 ShortsWearing + \beta_2 Temperature + \varepsilon$$

- Now we have a *multivariable* regression model. Our estimate $\hat{\beta}_1$ will *not* be biased by $Temperature$ because we've controlled for it

(probably more accurate to say "covariates" or "variables to adjust for" than "control variables" and "adjust for" rather than "control for" but hey what are you gonna do, "control" is standard)

# To the Rescue

- So the task of solving our endogeneity problems in estimating $\beta_1$ using $\hat{\beta}_1$ comes down to us *finding all the elements of $\varepsilon$ that are related to $X$ and adding them to the model*
- As we add them, they leave $\varepsilon$ and hopefully we end up with a version of $\varepsilon$ that is no longer related to $X$
- If $cov(X, \varepsilon) = 0$ then we have an unbiased estimate!
- (of course, we have no way of checking if that's true - it's based on what we think the data generating process looks like)

# How?

- Controlling for a variable works by *removing variation in $X$ and $Y$ that is explained by the control variable*
- So our estimate of $\hat{\beta}_1$ is based on *just the variation in $X$ and $Y$ that is unrelated to the control variable*
- Any "accidentally-assigning-the-value-of-Temperature-to-ShortsWearing" can't happen because we've removed the effect of $Temperature$ on $ShortsWearing$ as well as the effect of $Temperature$ on $IceCreamEating$
- We're asking at that point, *holding $Temperature$ constant*, i.e. *comparing two different days with the same $Temperature$* , how is $ShortsWearing$ related to $IceCreamEating$?
- We know we're comparing within the same $Temperature$ because we literally subtracted out all the $Temperature$ differences!

# Example: coin value

- Let's say we have several piles of coins from a collector with different amounts of quarters and dimes
- The piles are labeled with amounts of money and the amounts of coins, but we don't know the value of the coins
- We could use regression to find out
- One thing we do know is that the collector always had at least as many dimes as quarters
- My friend Szymon Sacher suggested this example

# Example: coin value

```
coins ← tibble(quarters = sample(0:10,10000,replace=TRUE), # sample generates a random amount of coins.
            pennies=sample(0:10, 10000, replace=TRUE),
            nickels=sample(0:10, 10000, replace= TRUE),
            error=rnorm(10000,0,.1)) %>%
  mutate(dimes = quarters + sample(0:10,10000,replace=TRUE)) %>%
  mutate(amount = 0.25*quarters + 0.10*dimes + 0.01*pennies + 0.05*nickels + error)

coins
```

```
## # A tibble: 10,000 × 6
##    quarters pennies nickels     error dimes amount
##       <int>   <int>   <int>     <dbl> <int>  <dbl>
##  1       10       2      10  0.0418      19   4.96
##  2        1       9       0 -0.0440       8   1.10
##  3        6       0       9  0.0242       6   2.57
##  4        1       7       2  0.0845       9   1.40
##  5        8       3       4  0.109       16   3.94
##  6        0       5       9 -0.000410     7   1.20
##  7        3      10       6 -0.0769       7   1.77
##  8        6       0       0  0.0193       6   2.12
##  9        4       4       7 -0.162        8   2.03
## 10        4       7       6 -0.0880       8   2.08
## # ℹ 9,990 more rows
```

# Straight-forward regression

```
allcoins ← feols(amount~ quarters + dimes + nickels+pennies,
  data = coins)
etable(allcoins,fitstat=~n,digits=2,se.below=TRUE) %>% kable(format="markdown")
```

| | **allcoins** |
|---|---|
| Dependent Var.: | amount |
| Constant | -0.0007 |
| | (0.003) |
| quarters | 0.25* |
| | (0.0004) |
| dimes | 0.10* |
| | (0.0003) |
| nickels | 0.05* |
| | (0.0003) |
| pennies | 0.010* |

# What if we remove quarters?

The coefficient on dimes changes a lot! Why?

```
noquarters ← feols(amount~ dimes + nickels+pennies, data = coins)
etable(allcoins,noquarters,fitstat=~n,digits=2,se.below=TRUE) %>% kable(format="markdown")
```

| | allcoins | noquart.. |
|---|---|---|
| Dependent Var.: | amount | amount |
| Constant | -0.0007 | 0.01 |
| | (0.003) | (0.02) |
| quarters | 0.25* | |
| | (0.0004) | |
| dimes | 0.10* | 0.23* |
| | (0.0003) | (0.001) |
| nickels | 0.05* | 0.05* |
| | (0.0003) | (0.002) |
| pennies | 0.010* | 0.009* |

# Endogeneity of quarters

- The number of dimes was a function of quarters
- When we dropped quarters, we omitted a variable that was related to dimes and the amount of money
- So the coefficient on dimes was biased

# Residualize variation from quarters

1. Average of `amount` and `dimes` by quarters, i.e. the part *explained by* `quarters`
2. Subtract from `amount` and `dimes`, to get the residual *not explained by* `quarters`

```
coins ← coins %>%
  group_by(quarters) %>%
  mutate(amount_mean = mean(amount), dimes_mean = mean(dimes),
    amount_res=amount-amount_mean,dimes_res=dimes-dimes_mean)
head(select(coins,quarters,matches('dimes|amount')))
```

```
## # A tibble: 6 × 7
## # Groups:   quarters [5]
##   quarters dimes amount amount_mean dimes_mean amount_res dimes_res
##      <int> <int>  <dbl>       <dbl>      <dbl>      <dbl>     <dbl>
## 1       10    19   4.96       4.31       15.1      0.650      3.89
## 2        1     8   1.10       1.16        6.05    -0.0688      1.95
## 3        6     6   2.57       2.91       11.1     -0.334     -5.06
## 4        1     9   1.40       1.16        6.05     0.240      2.95
## 5        8    16   3.94       3.61       13.1      0.328      2.86
## 6        0     7   1.20       0.793       4.95     0.407      2.05
```

# Residuals regression is unbiased

```
residuals ← feols(amount_res ~ dimes_res, data = coins)
etable(allcoins, noquarters,residuals,
  dict=c('dimes_res'='dimes'),fitstat=~n,digits=2,se.below=TRUE) %>% kable(format="markdown")
```

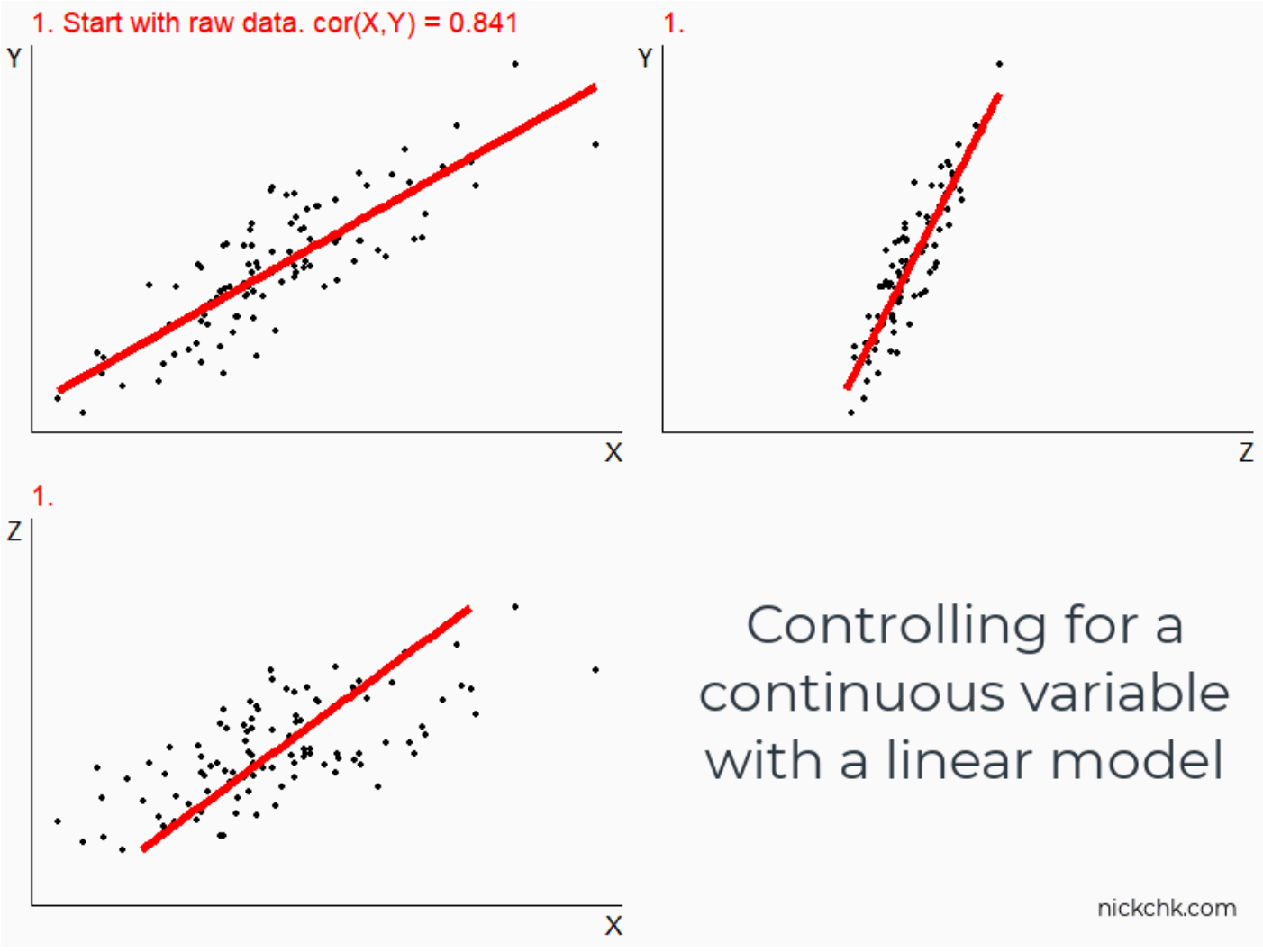
| | **allcoins** | **noquart..** | **residuals** |
|---|---|---|---|
| Dependent Var.: | amount | amount | amount_res |
| Constant | -0.0007 | 0.01 | 2.3e-17 |
| | (0.003) | (0.02) | (0.002) |
| quarters | 0.25* | | |
| | (0.0004) | | |
| dimes | 0.10* | 0.23* | 0.10* |
| | (0.0003) | (0.001) | (0.0006) |
| nickels | 0.05* | 0.05* | |
| | (0.0003) | (0.002) | |
| pennies | 0.010* | 0.009* | |

# Graphically with binary control $Z$

# Controlling

- We achieve all this just by adding the variable to the OLS equation!
- We can, of course, include more than one control, or controls that aren't binary
- Use OLS to predict $X$ using all the controls, then take the residual (the part not explained by the controls)
- Use OLS to predict $Y$ using all the controls, then take the residual (the part not explained by the controls)
- Now do OLS of just the $Y$ residuals on just the $X$ residuals

# A Continuous Control



# What do we get?

- We can remove some of the relationship between $X$ and $\varepsilon$
- Potentially all of it, making $\hat{\beta}_1$ us an *unbiased* (i.e. correct on average, but sampling variation doesn't go away!) estimate of $\beta_1$
- Maybe we can also get some estimates of $\beta_2$, $\beta_3$... but be careful, they're subject to the same identification and endogeneity problems!
- Often in econometrics we focus on getting *one* parameter, $\hat{\beta}_1$, exactly right and don't focus on parameters we haven't put much effort into identifying

# Summary

- We can remove endogeneity by adding omitted variables into our regression model if:

    1. We know/correctly assume what they are
    2. We can measure them

- This works by removing the part $X$ and $Y$ that is related to the omitted variable, $Z$
- This is fairly common, but often inadequate approach to causal inference
- Sometimes it is the best we can do though!