

# ProbSet 4, February 6

Gideon Gordon      [gideongordon2029@u.northwestern.edu](mailto:gideongordon2029@u.northwestern.edu)

**Due:** February 6, 2026

**Submission:** <https://canvas.northwestern.edu/courses/245562/assignments/1676748>

---

## 0.1 Problem 1

**1a.** Define omitted variable bias in your own words.

Omitted variable bias is the change in the coefficient of one variable that results from removing another variable from a regression equation.

**1b.** Consider the linear models:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + u$$

$$y = \beta_0 + \beta_1 x_1 + u$$

Derive the formula for  $\beta_1$  in terms of  $\beta_0$ ,  $\beta_2$ , and the relationship between  $x_1$  and  $x_2$ . Show all steps.

We can start by defining  $x_1$  in terms of all the other variables in the full formula.  $(y_0 \beta_2 u) / \beta_1 =$   
 $x_1$

Then let's insert our formula for  $x_1$  into the other formula.  $y = \beta_0 + \beta_1[(y_0 - \beta_2 x_2 - u)/\beta_1] + u$

We want to get  $\beta_1$  on its own, so let's shuffle things around again.  $\beta_1(y_0 - u)/[(y_0 - \beta_2 x_2 - u)] = \beta_1$

**1c.** Interpret each term in your formula from 1b. Under what conditions does omitted variable bias occur? When is it zero?

$\beta_0$  is the intercept, with both  $x_1$  and  $x_2$  included.  $\beta_0$  is the intercept with only  $x_1$  included.  $x_1$  is the main variable of interest, whose coefficient will differ depending on what other variables we control for.  $\beta_1$  is the coefficient of  $x_1$  with a control for  $x_2$ .  $\beta_1$  is the coefficient of  $x_1$  without a control for  $x_2$ .  $x_2$  is a control variable on which  $x_1$  may be conditional.  $\beta_2$  is the coefficient of  $x_2$ .  $\hat{y}$  is the predicted value for a given  $x_1$  and  $x_2$ .  $u$  and  $u$  are the

Omitted variable bias will be the difference between  $\beta_1$  and  $\beta_1(\beta_1 - \beta_1)$ , which we can express as:

$$\beta_1(y_0 - u)/(y_0 - \beta_2 x_2 - u) = \beta_1 \beta_1(y_0 - u)/(y_0 - \beta_2 x_2 - u) = \beta_1$$

$$\beta_1(y_0 - u)/(y_0 - \beta_2 x_2 - u) - \beta_1 \beta_1(y_0 - u)/(y_0 - \beta_2 x_2 - u).$$

The omitted variable bias will only be zero if  $x_2$  has no covariance with  $x_1$  ( $cov(x_1, x_2) = 0$ ) or if  $\beta_2$  is 0. I know that. What I *don't* know is how to explain it based on the mess of formulae I just produced.

I would appreciate some help linking the official OVB formula to the version which is  $\beta_1 - \beta_1$ .

---

## 0.2 Problem 2

The lecture slides show a nonlinear CEF for GDP and turnout, with different linear approximations (BLPs) for different ranges of GDP. In your own words, explain:

2a. What does it mean for the BLP to be the “best linear approximation” of a nonlinear CEF?

It means basically that even though it's not the best approximation of the CEF, there is no

*linear* approximation that does a better job accounting for variation in  $y$  conditional on  $x$ . There are other non-linear approximations that do better though.

Even though the CEF is nonlinear, the BLP will still retain some handy properties: errors will average out to zero, and errors will be uncorrelated with the independent variable.

2b. How can the BLP coefficient change sign depending on which range of the data we focus on?

The BLP has to be linear. So, if the CEF is not linear, the BLP will average the slope over the area where most of the data is concentrated. For example, if the center of mass of a quadratic CEF is towards the right, the BLP will mostly approximate the section of the data where the quadratic is increasing as  $x$  increases (so it will have a positive coefficient). If the center of the distribution is towards the left, the BLP will mostly approximate the area where the value of  $y$  is decreasing as  $x$  increases (so it will have a negative coefficient).

2c. What are the implications for interpreting regression coefficients when the true CEF is nonlinear?

You have to pay careful attention to the distribution of the data, first of all. You have to recognize that the BLP is an approximation based on where data points exist in our sample. So, if our true CEF is quadratic and almost all the data we have is to the right of the quadratic's minimum value, we'll get something that looks like a positive BLP.

The BLP will generally be a better approximation of the CEF closer to the center of mass of the data, and farther from that center, the BLP will stop being so useful.

---

### 0.3 Problem 3

Return to your voter turnout analysis from Problem Set 3.

**3a.** Re-examine the difference between your bivariate model (turnout ~ income) and multivariate model (turnout ~ income + religiosity + age). Set up a version of the multivariate model that uses only income and religiosity and does not use age (turnout ~ income + religiosity). Calculate the omitted variable bias for income comparing this new multivariate model to the bivariate model using the formula from Problem 1.

```
library(poliscidata)
```

```
Registered S3 method overwritten by 'gdata':
```

```
method          from
reorder.factor  gplots
```

```
library(tidyverse)
```

```
Warning: package 'ggplot2' was built under R version 4.5.2
```

```
Warning: package 'tibble' was built under R version 4.5.2
```

```
Warning: package 'tidyr' was built under R version 4.5.2
```

```
Warning: package 'readr' was built under R version 4.5.2
```

```
Warning: package 'purrr' was built under R version 4.5.2
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
v dplyr      1.1.4      v readr      2.1.6
```

```
v forcats   1.0.1      v stringr    1.6.0
```

```
v ggplot2    4.0.1      v tibble     3.3.1
```

```
v lubridate  1.9.4      v tidyr      1.3.2
```

```
v purrr      1.2.1
```

```
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()
```

```
x dplyr::lag()     masks stats::lag()
```

i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to

```
library(ggplot2)

# Clean and prepare the data
states_data <- states %>%
  select(state, vep12_turnout, prcapinc, religiosity, over64) %>%
  filter(!is.na(vep12_turnout), !is.na(prcapinc)) %>%
  mutate(income_thousands = prcapinc / 1000)

# Your code here

turnout_bivariate <- lm(vep12_turnout ~ income_thousands, data = states_data)
summary(turnout_bivariate)
```

Call:

```
lm(formula = vep12_turnout ~ income_thousands, data = states_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-15.8558	-3.3015	0.0432	3.9256	13.5216

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	41.6116	6.2935	6.612	2.9e-08 ***
income_thousands	0.5735	0.1951	2.939	0.00505 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.076 on 48 degrees of freedom

Multiple R-squared: 0.1525, Adjusted R-squared: 0.1349

F-statistic: 8.639 on 1 and 48 DF, p-value: 0.005047

```
# Your code here
```

```
turnout_multivariate <- lm(vep12_turnout ~ income_thousands + religiosity + over64,  
                           data = states_data)  
summary(turnout_multivariate)
```

Call:

```
lm(formula = vep12_turnout ~ income_thousands + religiosity +  
    over64, data = states_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-16.0052	-3.7213	0.7419	3.9666	13.7749

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	44.77407	9.74654	4.594	3.39e-05 ***
income_thousands	0.34790	0.25516	1.363	0.179
religiosity	-0.02987	0.02144	-1.393	0.170
over64	0.09013	0.49958	0.180	0.858

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.077 on 46 degrees of freedom

Multiple R-squared: 0.1876, Adjusted R-squared: 0.1347

F-statistic: 3.542 on 3 and 46 DF, p-value: 0.02169

Then compute:  $OVB = 2 * Cov(x1, x2) / Var(x1)$

```
# Calculate the components needed for OVB formula
```

```
# You'll need:
```

```
# 1. 2 from multivariate model (coefficient for religiosity)
```

```
coef(turnout_multivariate)["religiosity"]
```

```
religiosity
```

```
-0.02986965
```

```
# 2. Covariance between income and religiosity
```

```
cov(states_data$income_thousands, states_data$religiosity)
```

```
[1] -150.9603
```

```
# 3. Variance of income
```

```
var(states_data$income_thousands)
```

```
[1] 19.79052
```

```
# Then compute:  $OVB = 2 * Cov(x1, x2) / Var(x1)$ 
```

```
OVB = coef(turnout_multivariate)["religiosity"] * cov(states_data$income_thousands, states_data$religiosity) / var(states_data$income_thousands)
```

(I must have gotten 1 *seriously* wrong, but I'm not sure how. **Please help.**)

**3b.** Does the OVB formula correctly predict the difference between the bivariate and multivariate income coefficients? Show your calculations.

```
OVB2 = coef(turnout_multivariate)["income_thousands"] - coef(turnout_bivariate)["income_thousands"]
```

For some reason the OVB I calculated directly just now (OVB2) was negative rather than

positive. It should be positive, since omitting the religiosity variable increases the coefficient of income. But I'm not sure where the error is coming from. There's also some slight difference at the thousandths place (0.226 versus 0.228). Broadly speaking, yes the OVB formula predicts the difference between bivariate and multivariate income coefficients, but there are probably some things going on under the hood that I'm missing.

**3c.** Based on the lecture slides' discussion of model specification: 1. Could adding more variables ever increase bias? Under what conditions? Adding a control variable will change the coefficient of the independent variable, as long as the control is correlated with both dependent and independent variables and *regardless of the reason for the correlation*.

So if you control for an important mediating variable, you might wind up reducing the apparent direct effect of our independent variable on our dependent variable, concealing a real effect.

If you control for a "collider", which is a variable resulting from both  $x_1$  and  $y$ , you create additional bias. An example: Let's say we find that disease outbreaks are made more likely when a civil war is occurring. If we were to add a variable for excess deaths, that would be correlated with the civil war (because of battle deaths) *and* with the disease outbreak (because of deaths from disease), even though the causation clearly runs from disease to excess death and not in reverse. Controlling for excess deaths would muddle the effect of civil war on disease outbreaks.

Only if you control for a confounding variable do things improve.

2. When might it be better to use a bivariate model even if you suspect omitted variables?

Bivariate models are generally easier to interpret and work with. They also require less data, because the more variables you throw in, the fewer degrees of freedom you have. So if you don't have a good theoretical description of how the omitted variable will work (for example, if you aren't certain which way causation ought to flow), or enough data to make a more

complicated regression safe, a bivariate model might be your best bet.

---

## 0.4 Problem 4

### Simulation Study of OVB

We'll study two scenarios of omitted variable bias through simulation.

#### Scenario A: Confounding (Both X1 and X2 cause Y)

```
set.seed(789)
n <- 1000
x1 <- rnorm(n)
x2 <- 0.7*x1 + rnorm(n) # x2 correlated with x1
y <- 2 + 1.5*x1 + 2*x2 + rnorm(n, sd = 0.5)

# Run regressions
model_bivariate <- lm(y ~ x1)
model_multivariate <- lm(y ~ x1 + x2)

summary(model_bivariate)
```

Call:

```
lm(formula = y ~ x1)
```

Residuals:

Min	1Q	Median	3Q	Max
-6.1525	-1.3460	-0.0749	1.3630	5.9756

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.96193	0.06260	31.34	<2e-16 ***
x1	2.92022	0.06245	46.76	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.98 on 998 degrees of freedom

Multiple R-squared: 0.6866, Adjusted R-squared: 0.6863

F-statistic: 2187 on 1 and 998 DF, p-value: < 2.2e-16

```
summary(model_multivariate)
```

Call:

```
lm(formula = y ~ x1 + x2)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.97057	-0.31425	0.01079	0.31582	1.52002

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.99251	0.01541	129.3	<2e-16 ***
x1	1.51282	0.01908	79.3	<2e-16 ***
x2	1.98904	0.01598	124.5	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4871 on 997 degrees of freedom

Multiple R-squared: 0.981, Adjusted R-squared: 0.981

F-statistic: 2.58e+04 on 2 and 997 DF, p-value: < 2.2e-16

**Questions for Scenario A:** 1. What is the true value of  $\beta_1$ ? The true value is the coefficient in the CEF, 1.5.

2. What is the estimated  $\beta_1$  in the bivariate model? How biased is it? The estimated  $\beta_1$  in the bivariate model is quite biased, almost double the real value (2.9 instead of 1.5). The bias is

```
coef(model_multivariate)["x1"] - coef(model_bivariate)["x1"]
```

```
x1  
-1.407403
```

3. Use the OVB formula to calculate the expected bias. Does it match the actual bias?

```
2 * cov(x1, x2) / var(x1)
```

```
[1] 1.415157
```

Just about! It's a little bit off in the hundredths and thousandths place.

**Scenario B: Collider Bias (X2 is a common effect)**

```
set.seed(789)  
n <- 1000  
  
# Correct setup for collider scenario  
x1 <- rnorm(n)  
y <- 2 + 1.5*x1 + rnorm(n, sd = 0.5) # y depends only on x1
```

```

x2 <- 0.7*x1 - 1.5*y + rnorm(n, sd = 0.5) # x2 is a collider

# Run regressions
model_correct <- lm(y ~ x1) # Correct specification
model_collider <- lm(y ~ x1 + x2) # Including collider

summary(model_correct)

```

Call:

```
lm(formula = y ~ x1)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.53024	-0.32887	-0.01085	0.32344	1.51190

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.99231	0.01525	130.61	<2e-16 ***
x1	1.50379	0.01522	98.83	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4824 on 998 degrees of freedom

Multiple R-squared: 0.9073, Adjusted R-squared: 0.9072

F-statistic: 9768 on 1 and 998 DF, p-value: < 2.2e-16

```
summary(model_collider)
```

Call:

```
lm(formula = y ~ x1 + x2)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.77772	-0.17814	-0.00328	0.17006	0.93515

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.625167	0.029928	20.89	<2e-16 ***
x1	0.796156	0.017079	46.62	<2e-16 ***
x2	-0.456355	0.009585	-47.61	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2667 on 997 degrees of freedom

Multiple R-squared: 0.9717, Adjusted R-squared: 0.9716

F-statistic: 1.711e+04 on 2 and 997 DF, p-value: < 2.2e-16

**Questions for Scenario B:** 1. What is the true value of  $\beta_1$ ? The true value here is again 1.5, from the true CEF.

2. What happens when we include  $x_2$  in the regression? Why? Including  $x_2$  reduces the coefficient  $\beta_1$  from its actual value, because  $x_2$  captures some of the variation in  $x_1$  and  $y$  (even though it is downstream of both).

3. This demonstrates “bad control” or collider bias. Explain in your own words why including  $x_2$  creates bias even though  $x_2$  is correlated with both  $x_1$  and  $y$ . Because  $x_2$  is not actually a cause of  $y$ ; it is an effect of  $y$  and  $x_1$ . The causation is backwards, but a regression can’t tell the direction of causation.

**4c. Simulation Synthesis:** 1. Create a table comparing both scenarios.

I’ll first create a vector of values of  $\beta_1$ ,  $\beta_1^*$ ,  $\beta_2$  for each scenario.

```
scenarioA_coefs = c(coef(model_multivariate)["x1"], coef(model_bivariate)["x1"], coef(model_collider)["x1"])
scenarioB_coefs = c(coef(model_correct)["x1"], coef(model_collider)["x1"], coef(model_collider)["x2"])

beta1 = c(coef(model_multivariate)["x1"], coef(model_correct)["x1"])
beta1star = c(coef(model_bivariate)["x1"], coef(model_collider)["x1"])
beta2 = c(coef(model_multivariate)["x2"], coef(model_collider)["x2"])

scenariosA_B = data.frame(cbind(scenarioA_coefs, scenarioB_coefs))
rownames(scenariosA_B) <- c("Correct X1", "Biased X1", "Control (right or wrong)")
colnames(scenariosA_B) <- c("Scenario A", "Scenario B")

library(tinytable)
```

Attaching package: 'tinytable'

The following object is masked from 'package:ggplot2':

```
theme_void

tt(scenariosA_B, rownames = TRUE, digits = 3, escape = TRUE)
```

2. Under what circumstances does adding a control variable reduce bias? When might it increase bias?

rowname	Scenario A	Scenario B
Correct X1	1.51	1.504
Biased X1	2.92	0.796
Control (right or wrong)	1.99	-0.456

Adding a control variable, as I said, reduces bias as long as the control variable controls for a confounder (a cause of both  $x_1$  and  $y$ ). It will create new bias problems if the control variable controls for a collider (an effect of both  $x_1$  and  $y$ ), and we might consider controlling for a mediating variable (an effect of  $x_1$  and a cause of  $y$ ) problematic.

3. How can researchers decide which variables to include in a regression model?

My answer would be: You need to have a decent theory of what's going on before you start tinkering with the variables. In some cases, something is clearly a collider, mediator, or confounder. Other times, you should think carefully first and only add variables that, at minimum, you are sure aren't colliders.

You can't tell from the coefficients alone, since almost no matter what you do, adding variables will change the coefficients one way or another.

---

## 0.5 Problem 5

The lecture slides derive OLS using the plug-in principle and matrix algebra.

**5a. Plug-in Principle:** 1. Define the plug-in principle in your own words.

Plug-in principle basically means we plug in the sample version of an quantity into a formula in place of the population version. So we can estimate the population mean by plugging in the sample mean, for instance.

**5b. Matrix Derivation:** The OLS estimator in matrix form is:

$$= (X^T X)^{-1} X^T y$$

Using the turnout data with GDP and Temperature:

```
# Load and clean data
turnout <- read.csv("https://raw.githubusercontent.com/jnseawright/ps405/refs/heads/main/turnout.csv")
turnout_clean <- turnout[13:nrow(turnout), ] # Remove NA rows as in slides

# Create X matrix with intercept, GDP, Temperature
X <- as.matrix(cbind(1, turnout_clean$GDP, turnout_clean$Temperature))
colnames(X) <- c("Intercept", "GDP", "Temperature")

# Create y vector
y <- turnout_clean$Turnout

# Compare with lm() output
```

On the first try of this formula, I was warned about non-conformable arrays. It turns out for multiplying matrices you gotta use `%*%` to calm the thing down, and take two worked. But then on take three, I found that I had done everything wrong, because I'd used `^-1` instead of `solve()`.

Take three gave me an error saying that the “system is computationally singular”. On a Google search, it appears this means that my columns in `solve(t(X) %*% X)` are not linearly independent. I am... not sure how that's possible. My formula for take three was as follows:

```
OLS_Coefs = solve(t(X) %*% X) %*% t(X) %*% y
```

Aha! Looking at the guide and my class notes, it seems I need to set some tolerance in

`solve()`. This sets me up for takes four, five, and six, in which I tried to set tolerance and failed multiple times for unclear reasons. Ultimately I set the tolerance to zero and just saw what happened, and it worked, so I decided I was probably doing it backwards, and on take 8, set `tol=1e-40`.

```
# Calculate OLS coefficients using matrix algebra
OLS_Coefs = solve(t(X) %*% X, tol=1e-40) %*% t(X) %*% y
```

```
turnout_clean = as.data.frame(turnout_clean)
turnout = as.data.frame(turnout)
```

```
Turnout_lm = lm(Turnout ~ GDP + Temperature, data = turnout)
coef(Turnout_lm)
```

	(Intercept)	GDP	Temperature
	5.750768e-01	-3.524858e-09	9.408781e-04

```
OLS_Coefs
```

```
      [,1]
Intercept 5.750768e-01
GDP       -3.524858e-09
Temperature 9.408781e-04
```

After multiple tries made very embarrassing by the realization that the problem was, I hadn't capitalized Turnout in the variable name, we have some coefficients. And they are... not the same.

At least on take one, not the same at all.

1. Verify that your matrix calculation matches the `lm()` output.

Finally on take 8: yes, the coefficients are the same.