Weekly Practice 2 Solutions

Omitted Variable Bias

The WP for this week is an example of what happens when we have omitted variable bias.

Part A

Load the appropriate libraries needed to be able to appropriately estimate a linear model in R, and then graph the coefficients.

```
library(estimatr)
library(tidyverse)
```

Part B

Build a simulated dataset. Set the random seed for reproducibility to 42.

Imagine your dataset has 1000 observations. It should contain the following:

D: a treatment vector that can take on either the values of 0 or 1. X: a covariate that affects Y that is normally distributed with a mean of 100 and a standard deviation of 50 Y_short: a variable that is the result of adding the treatment assignment and a random standard normal draw. Y_long: a variable that is the result of adding the treatment assignment, X, and a random standard normal draw.

```
set.seed(42)
data <- tibble(
  D = sample(0:1, 1000, replace = T),
  X = rnorm(1000, 100, 50),
  Y_short = D + rnorm(1000),
  Y_long = D + X + rnorm(1000)
)
```

Part C

Run the result of the regression of Y_short on D. Get the output as a data frame and save the result to the variable m1. Add a column named model to m1 that is defined for every variable as your formula call.

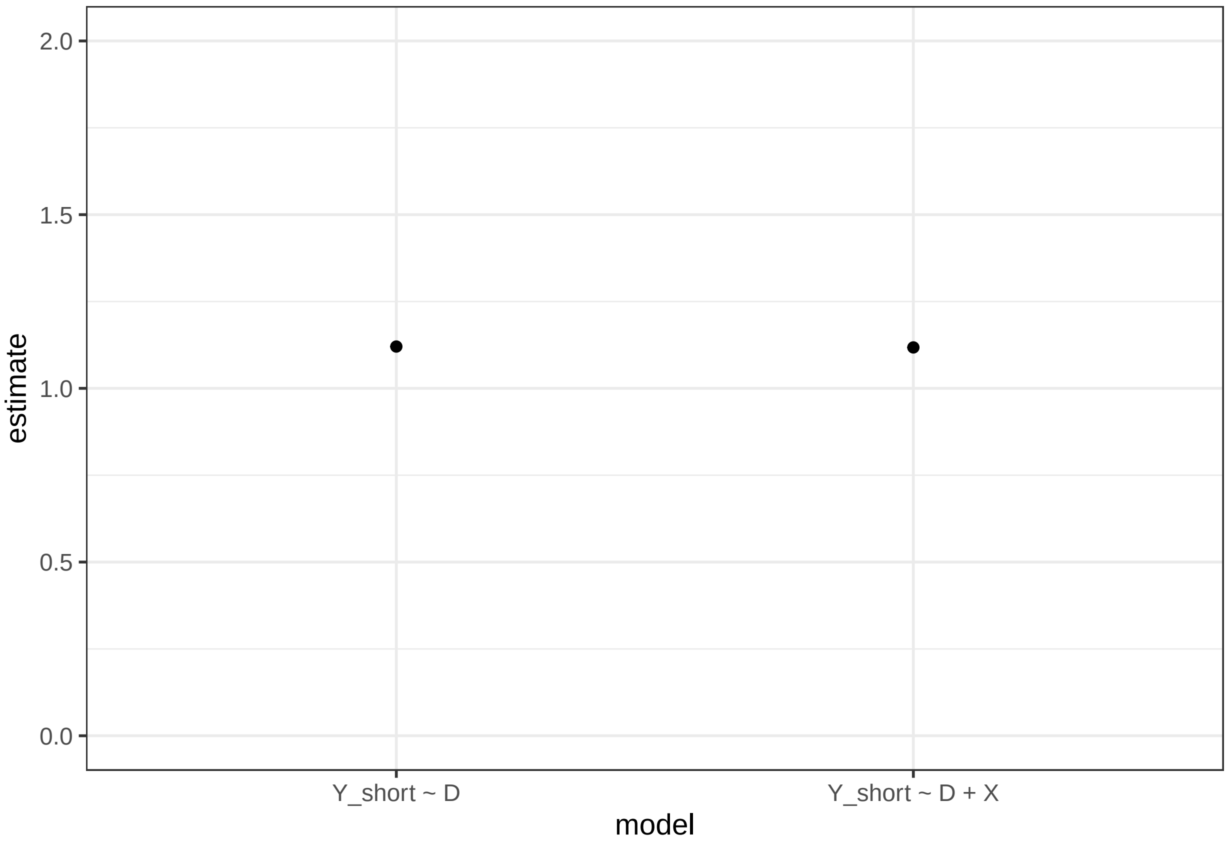
```
m1 <- lm_robust(Y_short ~ D, data = data, se_type = "stata")%>%
  tidy()%>%
  mutate(model = "Y_short ~ D")
```

Run the result of the regression of Y_short on D and X. Get the output as a data frame and save the result to the variable m2.Add a column named model to m2 that is defined for every variable as your formula call.

```
m2 <- lm_robust(Y_short ~ D + X, data = data, se_type = "stata")%>%
  tidy()%>%
  mutate(model = "Y_short ~ D + X")
```

Join the two data frames together. Call this data frame result1. Consult the `bind_rows()` function from dplyr for a way to do this. Graph just the D terms for each model on the same graph.

```
bind_rows(m1, m2)%>%
  filter(term == "D")%>%
  ggplot(aes(x = model, y = estimate))+
  geom_point()+
  theme_bw()+
  ylim(0,2)
```



Part D

Now repeat the same exercise as Part C, but using Y_long as the outcome variable.

Run the result of the regression of Y_long on D. Get the output as a data frame and save the result to the variable m3. Add a column named model to m3 that is defined for every variable as your formula call.

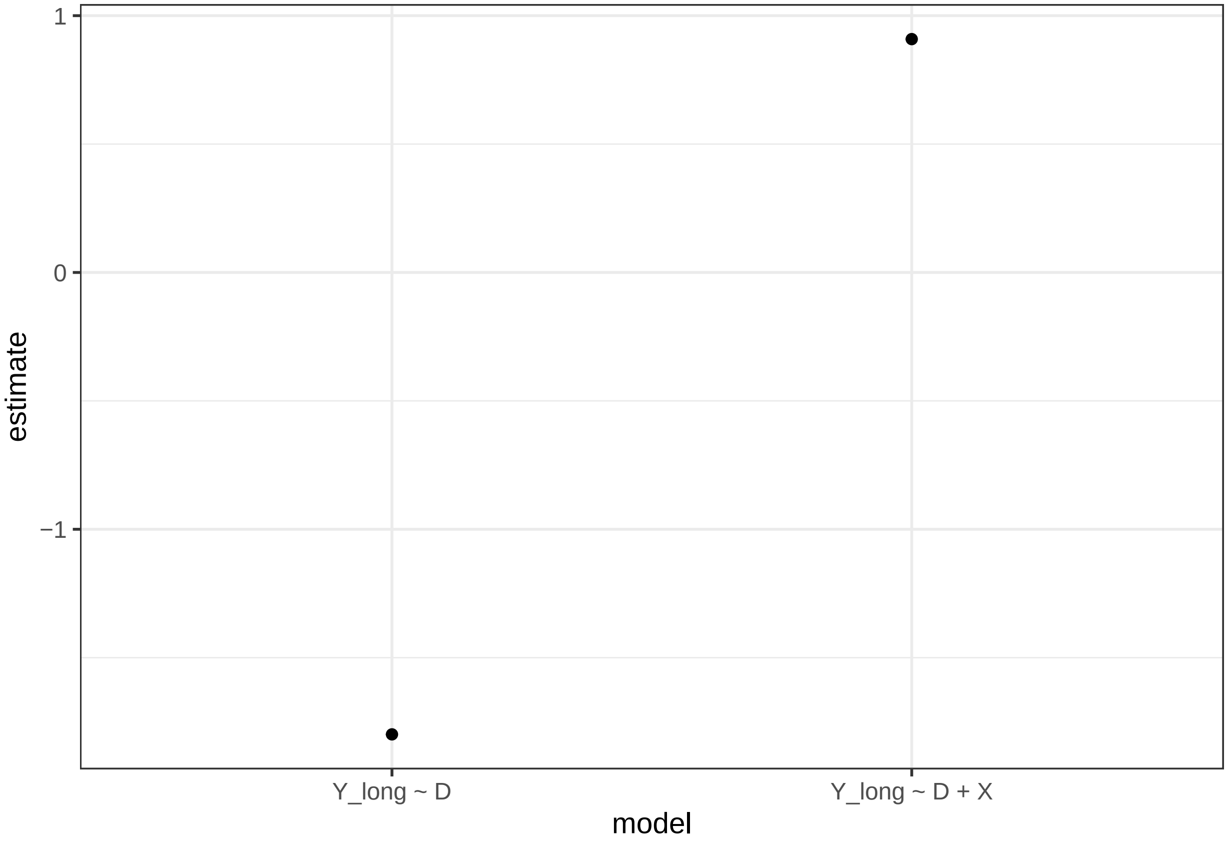
```
m3 <- lm_robust(Y_long ~ D, data = data, se_type = "stata")%>%
  tidy()%>%
  mutate(model = "Y_long ~ D")
```

Run the result of the regression of Y_long on D and X. Get the output as a data frame and save the result to the variable m4. Add a column named model to m4 that is defined for every variable as your formula call.

```
m4 <- lm_robust(Y_long ~ D + X, data = data, se_type = "stata")%>%
  tidy()%>%
  mutate(model = "Y_long ~ D + X")
```

Join the two data frames together. Call this data frame result2. Consult the `bind_rows()` function from dplyr for a way to do this. Graph just the D terms for each model on the same graph.

```
bind_rows(m3, m4)%>%
  filter(term == "D")%>%
  ggplot(aes(x = model, y = estimate))+
  geom_point()+
  theme_bw()
```



Part E

Based on the two graphs you made, what is a conclusion about the effect of omitted variables?

A conclusion among others is that omitted variables bias matters when X affects the data generating process. In the first regression, because X doesn't affect Y, ignoring it poses no problems. In the second case, ignoring X flips both the sign and magnitude of D.