Consider a regression model with the following causal structure:

The variable `x1` affects `y` directly and also indirectly via `x2`. The following R code implements the model and simulates a corresponding data set.

```
set.seed(1)
n = 10000
beta1 = 1; beta2=1
x1 = rnorm(n,0,1)
x2 = x1+rnorm(n,0,1)
y = beta1*x1 + beta2*x2 + rnorm(n,0,1)
```

Assume we want to consistently estimate the direct linear effect `beta1` from `x1` on `y`. To do so, we can simply estimate a multiple linear regression where we add `x2` as a control variable:

```
coef(lm(y~x1+x2))
```

```
## (Intercept)          x1          x2
## 0.007553765 0.998331323 1.000934100
```

But what does it intuitively mean to add `x2` as control variable? The Frisch-Waugh-Lovell Theorem implies that we get the same estimator for `beta1` as in the multiple regression above by the following procedure:

```
# y.tilde is residual of regression
# y on x2
y.tilde = resid(lm(y~x2))

# x1.tilde is residual of regression
# x1 on x2
x1.tilde = resid(lm(x1~x2))

# Regression y.tilde on x1.tilde
# we get the same estimate for beta1
# as in the multiple regression with x1 and x2
coef(lm(y.tilde ~ x1.tilde))
```

```
##   (Intercept)      x1.tilde
## -5.104062e-17  9.983313e-01
```

Hence, controlling for `x2` means that we essentially regress the residual variations of `y` and `x1` that cannot be linearly explained by `x2` on each other. So far this seems intuitive.

The interesting thing is that one gets the same estimate for `beta1` also with *one* of the following two regressions below (but only the regression above also yields correct standard errors):

```
# Approach A
lm(y.tilde ~ x1)
# Approach B
lm(y ~ x1.tilde)
```

Approach A regresses the residual variation of `y` that cannot be linearly predicted by `x2` on `x1`. Approach B regresses `y` on the residual variation of `x1` that cannot be linearly predicted by `x2`.

Only one approach yields a consistent estimate of `beta1`. Make a guess which one…

Let's check:

```
# A: Inconsistent
coef(lm(y.tilde ~ x1))[2]
```

```
##        x1
## 0.4860465
```

```
# B: Consistent
coef(lm(y ~ x1.tilde))[2]
```

```
##  x1.tilde
## 0.9983313
```

So only approach B works. Angrist and Pischke (2009) refer to it as *regression anatomy*. For me that result was a bit puzzling for a long time because my intuitive interpretation of what it means to control for `x2` was more in line with approach A. I first want to shed light on that intuition and explain why approach A does not work. Afterward I want to give some intuition for the working approach B.

An intuition for control variables and why approach A fails

I have different intuitions what controlling for `x2` means in the linear regression:

```
y = beta0 + beta1*x1 + beta2*x2 + eps
```

One of my intuitions is the following:

> "By controlling for `x2`, we essentially subtract the variation that can be linearly explained by `x2` from `y`, i.e. up to an estimation error we subtract `beta2*x2`."

This interpretation suggests that approach A should work, but that approach fails to get a consistent estimate for `beta1`. So is the intuition above wrong? Not completely, but the qualification "up to an estimation error" causes trouble for approach A. Consider the following code.

```
# Modified approach A
y.tilde2 = y - beta2*x2
coef(lm(y.tilde2 ~x1))[2]
```

```
##        x1
## 0.9992699
```

It is a modified version of approach A. It computes the residual variation `y.tilde2` by directly subtracting `beta2*x2` from `y`. Now we get a consistent estimator of `beta1` when regressing `y.tilde2` on `x1`.

But approach A differs because we subtract `beta2.hat*x2` from `y` where `beta2.hat` is estimated in the first stage regression:

```
# Same result as original approach A
beta2.hat = coef(lm(y~x2))[2]
beta2.hat # inconsistent estimate of beta2
```

```
##       x2
## 1.510801
```

```
y.tilde = y-beta2.hat*x2
coef(lm(y.tilde ~x1))[2] # inconsistent estimate of beta1
```

```
##        x1
## 0.4860465
```

The problem with approach A is that we don't estimate `beta2.hat` consistently in the regression of `y` on `x2`. Instead, since `x1` and `x2` are correlated, `beta2.hat` also captures some of the direct effect of `x1` on `y`. This means in `y.tilde` we have already removed some of the effect from `x1` on `y` that we want to estimate. Therefore approach A yields an estimator for `beta1` that is biased towards 0.

Remark: In the original computation of approach A, we also subtract the estimated constant from the initial regression when computing `y.tilde`, but that has no effect on the slope coefficient in the second stage regression.

Interestingly, in some empirical papers an approach similar to approach A is performed, i.e. one first computes residuals of `y` from a first regression and then regresses those residuals on another set of explanatory variables. But the computation above shows that one should really be careful with this approach, since it only works if the first regression yields consistent estimates.

Let us consider an example where such an approach would work. Consider the following modified model:

We now have an additional variable `z` that affects `x2` but is uncorrelated with `x1`.

```
z = rnorm(n,0,1)
x2 = x1+z+rnorm(n,0,1)
y = beta1*x1 + beta2*x2 + rnorm(n,0,1)
```

We now conduct a variation of approach A where `y.tilde3` are the residuals of an instrumental variable regression of `y` on `x2` using `z` as instrument:

```
library(AER)
reg1 = ivreg(y~x2|z)
coef(reg1)[2] # consistent beta2.hat
```

```
##       x2
## 1.006464
```

```
y.tilde3 = resid(reg1)
coef(lm(y.tilde3 ~ x1))[2] # consistent beta1.hat
```

```
##        x1
## 0.9756005
```

We now see that regressing `y.tilde3` on `x1` yields a consistent estimator of `beta1`.

Why does approach B work

Let us now discuss why approach B works. Given our causal structure I find it more intuitive to first discuss why a similar approach works to consistently estimate `beta2`.

```
# x2.tilde is residual from regression
# of x2 on x1
x2.tilde = resid(lm(x2~x1))
# consistent estimate of beta2
coef(lm(y ~ x2.tilde))[2]
```

```
## x2.tilde
## 0.996786
```

Here I have the following intuition why it works. Intuitively, to consistently estimate the causal effect of `x2` on `y` we need to distill variation of `x2` that is uncorrelated with `x1`. If we regress `x2` on `x1`, the residuals `x2.tilde` of this regression are by construction uncorrelated with `x1`. They describe the variation of `x2` that cannot be linearly predicted by `x1`. That is exactly the variation of `x2` needed to consistently estimate `beta2`.

The equivalent procedure also works to estimate `beta1` consistently:

```
x1.tilde = resid(lm(x1~x2))
coef(lm(y ~ x1.tilde))[2] # consistent
```

```
## x1.tilde
##   0.98519
```

So even though `x2` does not influence `x1` we can similarly distill in `x1.tilde` the relevant variation in `x1` that is uncorrelated with `x2`. For the regression anatomy it is irrelevant which causal direction has generated the correlation between `x1` and `x2`.

Final remarks

I find it amazing that over many years I still often learn new intuitions for basic econometric concepts like multiple linear regression. Currently, I think introducing multiple regression via the Frisch-Waugh-Lovell theorem and the regression anatomy can be much more helpful to build intuition in an applied empirical course than covering the matrix algebra. (Of course, it is a different story if you want to prove econometric theorems.) For an example of such a course, you can check out the open online material (videos, quizzes, interactive R exercises) of my course Market Analysis with Econometrics and Machine Learning.

References

Angrist, Joshua D., and Jörn-Steffen Pischke. 2009. Mostly Harmless Econometrics: An Empiricist's Companion.