Problem Set 3:
Expectations, Linear Projection, and OLS

Artschil Okropiridse
Econometrics I

February 16, 2024

Please let me know if you find any mistakes or typos and feel free to ask if you got any questions. You can email me at artschil.okropiridse@su.se and feel free to drop by my office A644 (on the 6th flow).

1 Problems

Solutions to the simulations and code are attached at the end.

1. *Law of iterated expectations.*

   (a) *Pure math.* Assume that $E|y| < \infty$. Prove that $E\big(E(y|x_1)|x_1, x_2\big)$ and $E\big(E(y|x_1, x_2)|x_1\big)$ both equal $E(y|x_1)$. Explicitly state within the proof where you use the assumption.

   Solution: First note that $\mathbb{E}[y|\, x_1]$ is a function of x_1 and nothing else. Using the conditioning theorem, we can just take it out of the outer expectations.

$$\begin{aligned}\mathbb{E}[\mathbb{E}[y|\, x_1]|\, x_1, x_2] &= \mathbb{E}[1 * \mathbb{E}[y|\, x_1]|\, x_1, x_2] \\ &= \mathbb{E}[y|\, x_1]\,\mathbb{E}[1|\, x_1, x_2] \\ &= \mathbb{E}[y|\, x_1]\end{aligned}$$

   To see what happens under the hood let's write $\mathbb{E}[y|\, x_1] = m(x_1)$ we can write this with integrals:

$$\begin{aligned}
\mathbb{E}\left[\mathbb{E}\left[y|\, x_1\right]|\, x_1, x_2\right] &= \int_{\mathbb{R}} m(x_1) f(1|x_1, x_2) d1 \\
\Big/ \text{controlling for } x_1 &\text{ there is no randomless left} \Big/ \\
&= m(x_1) \int_{\mathbb{R}} f(1|x_1, x_2) d1 \\
&= m(x_1) \\
&= \int_{\mathbb{R}} y f(y|x_1) dy \\
&= \mathbb{E}\left[y|\, x_1\right]
\end{aligned}$$

Question from the seminar There is a formal proof in Hanson section 2.33. But the intuition is that when controlling for x_1 in the outer expectation, $m(x_1)$ becomes a constant, therefore, we can take it out of the integral. So what's left is basically a constant (namely 1) which we take the integral over (which is just 1). Why do we not take the integral w.r.t. to y? Because $m(x_1)$ gets all its randomness from x_1. And why do we not integrate over x_1 then? Because by conditioning on it we turn it into a constant (inside the conditional expectation).

$\mathbb{E}\left[|y|\right] < \infty$ implies that $y f(y|x_1)$ is integrable. When we show that the expectation operator is basically an integral, the need for the assumption becomes visible.

For the second part:

$$\begin{aligned}
\mathbb{E}\left[\mathbb{E}\left[y|\, x_1, x_2\right]|\, x_1\right] &= \int_{\mathbb{R}^{k_2}} \mathbb{E}\left[y|\, x_1, x_2\right] f(x_2|x_1) dx_2 \\
&= \int_{\mathbb{R}^{k_2}} \left(\int_{\mathbb{R}} y f(y|x_1, x_2) dy \right) f(x_2|x_1) dx_2 \\
&= \int_{\mathbb{R}^{k_2}} \int_{\mathbb{R}} y f(y|x_1, x_2) f(x_2|x_1) dy dx_2 \\
\Big/ f(y|x_1, x_2) f(x_2|x_1) &= \frac{f(y, x_1, x_2)}{f(x_1, x_2)} \frac{f(x_1, x_2)}{f(x_1)} = f(y, x_2|x_1) \Big/ \\
&= \int_{\mathbb{R}^{k_2}} \int_{\mathbb{R}} y f(y, x_2|x_1) dy dx_2 \\
&= \int_{\mathbb{R}} y \int_{\mathbb{R}^{k_2}} f(y, x_2|x_1) dx_2 dy \\
&= \int_{\mathbb{R}} y f(y|x_1) dy \\
&= \mathbb{E}\left[y|\, x_1\right]
\end{aligned}$$

Again the need for the assumption becomes visible as soon as we show we are integrating over y.

(b) *OLS as conditional expectations.* An intuitive example to understand why $E\big(E(y|x_1, x_2)|x_1\big) = E(y|x_1)$. Simulate 500 observations according to the following data generating process (DGP):

- $u_0 \sim N(0, \sigma^2)$
- $u_1 \sim N(0, 1)$
- $u_2 \sim N(0, 1)$
- $x_1 = u_0 + u_1$
- $x_2 = u_0 + u_2$
- $\varepsilon \sim N(0, 1)$
- $y = x_1 + \beta_2 x_2 + \varepsilon$

i. Start with $\sigma^2 = 1$ and $\beta_2 = 5$. What is the correlation between x_1 and x_2?

ii. Assume that it is easy for researchers to get access to x_1, but that y and x_2 are difficult to get access to (they requires special permissions, or are confidential data, etc.). Ana has all three variables, while Björn only has x_1. For his project, Björn needs an individual-level prediction of y. Ana cannot provide the data, but is happy to share any estimates that Björn asks for. Predict y as a function of x_1 only, and save the fitted values, which we will call $\hat{y}^{(1)}$. This is your feasible prediction of y given only information on x_1: $E(y|x_1)$. What is the mean squared error of this prediction (which Björn cannot calculate but Ana can), which is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - \hat{y}_i^{(1)}\big)^2$$

iii. Given how important x_2 is for y, Björn is pretty sure he can do better if he can use that information somehow. Estimate:

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

Ana provides these estimates to Björn, and Björn calculates

$$\hat{y}^{(2)} \equiv= \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 \bar{x}_2$$

where $\bar{x}_2$ is the sample mean of x_2 (which Ana can provide). Calculate the MSE for $\hat{y}^{(2)}$.

iv. Björn realizes that he's just adding a constant, and that's not improving his estimate of y. But since x_1 and x_2 are correlated, knowing x_1 actually tells him quite a bit about x_2 at an individual-level, and he can use this information (which varies across people) to improve his estimate of y. He asks Ana to estimate:

$$x_2 = \gamma_0 + \gamma_1 x_1 + \nu$$

He then calculates the fitted values of y as:

$$\hat{y}^{(3)} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 \hat{x}_2$$

where $\hat{x}_2 \equiv \hat{\gamma}_0 + \hat{\gamma}_1 x_1$. Calculate the MSE for $\hat{y}^{(3)}$.

v. Calculate the correlation coefficients of $\hat{y}^{(1)}$, $\hat{y}^{(2)}$, and $\hat{y}^{(3)}$. Relate these correlations to the result from the Law of Iterated Expectations that you proved in part (a). Given that Björn only observes x_1, can you come up with a better estimate of $\hat{y}$ for him?

(c) **(This question is optional.)** *Implications for empirical work.* You are interested in how a grant awarded to municipalities affects wages. We'll simulate grant receipt that is correlated with average education in the community, and we'll simulate that in a way that leads to differences in average education across municipalities (since we already know that if education is iid across municipalities then the WLLN implies there will be no variation in municipality-level average education). Ultimately, wages (y) will be a function of grants (g), education (x_1), some other factor that we don't observe (x_2), and an idiosyncratic iid individual-level error term. Simulate data according to the following multilevel DGP, with 50 municipalities and 100 individuals in each municipality:

- Municipality-level stuff:
  - $\mu_{1,m} \sim N(0, \sigma_1^2)$. This will be education.
  - $\mu_{2,m} \sim N(0, \sigma_1^2)$. This will be some other thing we don't observe.
  - $g_m \sim binom(p_m)$ (i.e., $g_m \in \{0, 1\}$ with $Pr(g_m = 1) = p_m$), where $p_m = \frac{\mu_{1,m} - min(\mu_1)}{max(\mu_1) - min(\mu_1)}$. This ensures that the probability of g_m is linear in $\mu_{1,m}$.
- Individual-level stuff:
  - $x_{1,i,m} \sim N(\mu_{1,m}, 1)$. This is the education of individual i living in municipality m.
  - $x_{2,i,m} \sim N(\mu_{2,m}, 1)$. This is the other characteristic for individual i living in municipality m.
  - $e \sim N(0, \sigma_e^2)$.
  - $y_{i,m} = \beta_0 + \beta_g g_m + \beta_1 x_{1,i,m} + \beta_2 x_{2,i,m} + e_{i,m}$. This is the wage equation.

i. Simulate the data, starting with $\beta_0 = \beta_g = \beta_1 = \beta_2 = \sigma_e^2 = \sigma_1^2 = \sigma_2^2 = 1$. Regress wages on g, x_1, and x_2, and verify that $\hat{\beta}_g$ is close to 1. Note that we are assuming you cannot run this regression because you don't observe x_2.

ii. Regress wages on g_m only, and verify that $\hat{\beta}_g$ is biased.

iii. Regress wages on g_m and $x_{1,i,m}$. Verify that this $\hat{\beta}_g$ is closer to β_g than what you got in ii above.[1]

iv. Calculate $\bar{x}_{1,m} \equiv \frac{1}{n_m} \sum_{i=1}^{n_m} x_{1,i,m}$ as the average level of education in the municipality. Regress wages on g_m and $\bar{x}_{1,m}$. Compare the coefficient $\hat{\beta}_g$ to what you got in parts ii and iii. Note the R^2 from this regression. Clearly individual-level education is an important determinant of individual-level wages, but does controlling for municipality-level education only really under-perform controlling

[1] If you're curious, you could simulate g_m with $p_m = \frac{\mu_{1,m}^2 - min(\mu_1^2)}{max(\mu_1^2) - min(\mu_1^2)}$. Now, the probability of a grant is not simply a linear function of x_1, and you can verify that controlling for x_1 makes less of a dent in the bias in $\hat{\beta}_g$.

for the individual-level variable? It's often helpful to re-run the simulation a few times over and over to get an informal sense of the distribution of the estimates?

v. Play with the parameter values for $\beta_0, \beta_g, \beta_1, \beta_2, \sigma_e^2, \sigma_1^2, \sigma_2^2$. Change the values a bit, and for some set of values, run 50 iterations of the simulation under each value. Create one figure showing a result that you consider interesting.

vi. Set $\beta_g = 0$ and keep all other values as they were in i. Run the full simulation 100 times. How often is the estimated coefficient $\hat{\beta}_g$ statistically significant at the 5% level? Note that the true $\beta_g = 0$, so $\hat{\beta}_g$ "should" only be significant 5% of the time.[2]

vii. Now calculate averages wages and average education, and estimate

$$\bar{y}_m = \beta_0 + \beta_g g_m + \beta_1 \bar{x}_{1,m} + \nu_m$$

How often is the estimated coefficient $\hat{\beta}_g$ statistically significant at the 5% level?

2. *FWL, OV'B', and LATE's.* You are interested in the causal effect of parental income on children's outcomes. You will simulate the data, so you know the truth and can compare that with regression results. Simulate 500 observations according to the following data generating process (DGP):

- Earnings $\sim N(19, 1)$. Note that if I later refer to "labor market earnings," I am referring to this variable.
- Capital gains $\sim N(1, 1)$
- $u \sim N(0, 1)$
- $e \sim N(0, 1)$
- Occupational status = Earnings + u
- Child Outcomes = Earnings - Capital gains + e. Note that this means that labor market earnings are good for children (improve their outcomes), while capital gains are actually bad for them (perhaps because they are unearned and send a bad signal to them about the value of hard work.
- Income $\equiv$ Earnings + Capital gains

(a) On average, across all individuals in your simulated sample, what fraction of income comes from labor market earnings?

(b) Note that the average respondent will have income of 20. Given that you simulated the data, what would you say is the true causal effect on child outcomes of increasing the average person's income by 10%, from 20 to 22?

(c) Regress child outcomes on earnings and capital gains and verify that OLS recovers the correct coefficients. Verify that controlling for occupational status (which is not part of the DGP) does not affect these coefficients.

[2]This question was inspired by Bertrand, Marianne, Esther Duflo, and Sendhil Mullainathan. "How much should we trust differences-in-differences estimates?" *The Quarterly Journal of Economics* 119.1 (2004): 249-275.

(d) A researcher is not interested in the potential distinction between earnings and capital gains, and she pools both into income. Regress child outcomes on income, and compare the coefficient to your answer to (b) above. Is income endogenous?

Question from the seminar Credit to Allesandro who pointed this out. We can actually say things precisely here. Let's define the following equations:

$$\begin{aligned} Y &= E - C + e, && \text{true DGP} \\ Y &= \alpha + \beta I + \nu = \alpha + \beta E + \beta C + \nu, && \text{specified model} \end{aligned}$$

So we want to know if $Cov(I, \nu) = Cov(I, \nu) = 0$. To get that we just plug in what we know:

$$\begin{aligned} Cov(I, \nu) &= Cov(I, Y - \alpha - \beta I) \\ &= Cov(I, Y - \beta I) \\ &= Cov(I, Y) - Var(I)\beta \\ &= Cov(E + C, E - C) - Var(E + C)\beta \\ &= Var(E) - Cov(E, C) + Cov(C, E) \\ &-Var(C) - \beta(Var(E) + Var(C) + 2Cov(E, C)) \\ &= Var(E) - Var(C) - \beta(Var(E) + Var(C)) \\ &= 1 - 1 - \beta(1 + 1) = -2\beta \end{aligned}$$

So what's β? Using our friend the regression anatomy formula we get:

$$\begin{aligned} \beta &= \frac{Cov(I, Y)}{Var(I)} \\ &= \frac{Cov(E + C, E - C)}{Var(E + C)} \\ &= \frac{Var(E) - Cov(E, C) + Cov(C, E) - Var(C)}{Var(E + C)} \\ &= \frac{Var(E) - Var(C)}{Var(E + C)} \\ &= \frac{1 - 1}{Var(E + C)} = 0 \end{aligned}$$

So we have $Cov(I, \nu) = 0$. BUT this is only the case because variances are equal and E and C are not correlated. If this would be different things wouldn't cancel out so nicely and we would probably end up with endogeneity. Keep in mind that even in that case, the problem would not be OVB, but misspecification.

(e) Within your sample, what is the correlation between occupational status and income?

(f) The researcher is concerned that she should be controlling for occupational status: it is highly correlated with income and excluding it might cause omitted variable bias (OVB). In this univariate context, OVB is a function of three terms, and since you simulated the data, you know what all three terms are. Analytically (by hand, without a computer) calculate the OVB that results from excluding occupational status.

(g) Regress child outcomes on income, controlling for occupational status, and compare the coefficient to your answer to (f) above. Does the researcher conclude that there is OVB? Has she now recovered the causal effect of income on child outcomes? Compare your answer to your answer to (b) above, and discuss the role of endogeneity.

3. *Measurement error and indices.* Assume that x and y are two mean zero variables. Suppose that the true model is given by $y = \beta x + \varepsilon$ where $E(x'\varepsilon) = 0$. Let σ_x^2 and σ_ε^2 be the variance of x and ε, respectively.

   (a) You do not observe the true x. Instead, you observe x only with error. You observe $\tilde{x} = x + \nu$ where ν is a mean zero white noise error term[3] with variance σ_ν^2. You regress y on $\tilde{x}$. Write $plim\hat{\beta}$ as a function of β, σ_ν^2, σ_x^2, and σ_ε^2 (not all of those terms will show up in the expression). Interpret the result. Note: This is called classical measurement error, and the bias in $\hat{\beta}$ is called attenuation bias.

   Solution: To ease notation slightly, let's define the population model:

$$y = \delta\tilde{x} + e$$

   We know that the OLS (sample) estimate converges in probability to its population counterpart the linear projection (if this sentence does not make sense to you or you haven't covered this material yet it is explained in much detail in Hanson section 7.2). Using this and the regression anatomy formula (Hanson section 2.21 or with more intuition Mostly Harmless 3.1.2) we can write:

$$\hat{\delta} \to^p \delta = \frac{Cov(\tilde{x}, y)}{Var(\tilde{x})}$$

   Now we can take the right-hand side apart (note that $\perp$ means "independent of"):

$$\begin{aligned}
\frac{Cov(\tilde{x}, y)}{Var(\tilde{x})} &= \frac{Cov(x + v, y)}{Var(x + v)} \\
&= \frac{Cov(x, y)}{Var(x + v)}, && \text{bc. } v \perp y \\
&= \frac{Cov(x, y)}{Var(x) + Var(v)}, && \text{bc. LTV and } v \perp x \\
&= \frac{Cov(x, \beta x + e)}{\sigma_x^2 + \sigma_v^2}, && \text{Notation change and DGP} \\
&= \beta\frac{\sigma_x^2}{\sigma_x^2 + \sigma_v^2}, && \text{bc. a constant and, } \mathbb{E}\left[xe\right] = 0
\end{aligned}$$

   Now have a look at the nominator and the denominator: $\sigma_x^2 + \sigma_v^2 \geq \sigma_x^2$. This tells us that $\delta \leq \beta$. So the coefficient will be biased towards zero.

[3] "White noise error term" means it is iid and independent of all other variables in the model.

(b) You **do** observe the true x, but you do not observe the true y. You observe y only with error: You observe $\tilde{y} = y + e$ where e is a mean zero white noise error term with variance σ_e^2. You regress $\tilde{y}$ on x. Write $plim\hat{\beta}$ as a function of β, σ_ν^2, σ_x^2, and σ_ε^2 (not all of those terms will show up in the expression). Interpret the result.

Solution: Let's again define the model with:

$$\tilde{y} = \delta x + e$$

For the same reasons as above we can skip to:

$$\begin{aligned}\delta = \frac{Cov(\tilde{y}, x)}{Var(x)} &= \frac{Cov(y + e, x)}{\sigma_x^2} \\ &= \frac{Cov(y, x)}{\sigma_x^2}, \text{bc. } x \perp e \\ &= \beta\end{aligned}$$

So, despite the measurement error, we can consistently estimate the true β. Running a real-life regression, what will happen is that our estimates will be noisier. So we might need a larger sample for estimates to converge to population parameters.

(c) (**Note**: I do not currently know how much algebra this one is. If it's crazy, please give up.) You observe x only with error. You observe $\tilde{x} = x + \nu$ where ν is a mean zero and has a correlation of ρ with ε. You regress y on $\tilde{x}$. Write $plim\hat{\beta}$ as a function of β, σ_ν^2, σ_x^2, σ_ε^2, and ρ (not all of those terms will show up in the expression). Interpret the result. Note: This is called non-classical measurement error because the error in your measure of x is systematically correlated with the dependent variable.

Solution: As before

$$y = \delta\tilde{x} + e$$

Crucially, there is a big difference if $x \perp v$ or not. We don't have any information about that. So let's not assume it holds (the computations are of course easier when it does). Starting as before (I also switch notation now to keep this a bit cleaner, using $Cov(a, b) = \sigma_{ab}$):

$$
\begin{aligned}
\delta = \frac{Cov(\tilde{x}, y)}{Var(\tilde{x})} &= \frac{Cov(\tilde{x}+v, \beta x+\varepsilon)}{Var(x+v)} \\
&= \frac{Cov(x+v, \beta x+\varepsilon)}{\sigma_x^2+\sigma_v^2+2\sigma_{xv}} \\
&= \frac{\beta\sigma_x^2+\sigma_{x\varepsilon}+\beta\sigma_{vx}+\sigma_{v\varepsilon}}{\sigma_x^2+\sigma_v^2+2\sigma_{xv}} \\
&= \beta\frac{\sigma_x^2+\sigma_{vx}}{\sigma_x^2+\sigma_v^2+2\sigma_{xv}} + \frac{\sigma_{v\varepsilon}}{\sigma_x^2+\sigma_v^2+2\sigma_{xv}} \\
&= \beta\frac{\sigma_x^2+\sigma_{vx}}{\sigma_x^2+\sigma_v^2+2\sigma_{xv}} + \frac{\sigma_{v\varepsilon}}{\sigma_v\sigma_\varepsilon}\frac{\sigma_v\sigma_\varepsilon}{\sigma_x^2+\sigma_v^2+2\sigma_{xv}} \\
&= \beta\frac{\sigma_x^2+\sigma_{vx}}{\sigma_x^2+\sigma_v^2+2\sigma_{xv}} + \rho\frac{\sigma_v\sigma_\varepsilon}{\sigma_x^2+\sigma_v^2+2\sigma_{xv}}
\end{aligned}
$$

Without more information about the different parameters we cannot say in which direction the bias goes. Also note that if we impose $\mathbb{E}[xv]=0$ we get:

$$
\beta\frac{\sigma_x^2}{\sigma_x^2+\sigma_v^2} + \rho\frac{\sigma_v\sigma_\varepsilon}{\sigma_x^2+\sigma_v^2}
$$

So the first term becomes the same as in (a) above, and therefore, will be attenuated. The second term will take the sign of the correlation between v and e.

(d) Return to the setup of 3ai: Classical measurement error, in which only x is measured with error and that error is white noise. Suppose you observe some z, which is correlated with x but not correlated with ν.[4] You use z as an instrument for x. Let $\hat{\beta}_{IV}$ be the two-stage least squares estimate (from the second stage) of the coefficient on x. Write $plim\hat{\beta}_{IV}$ as a function of β, σ_ν^2, σ_x^2, σ_ε^2, and σ_z^2 (not all of those terms will show up in the expression).

Solution: We can use the condition given in the question to derive:

$$
\begin{aligned}
\mathbb{E}[z\varepsilon] &= 0 \\
\mathbb{E}[z(y-\beta x)] &= 0 \\
\mathbb{E}[zy] - \beta\mathbb{E}[zx] &= 0 \\
\mathbb{E}[zy] &= \beta\mathbb{E}[zx] \\
\beta &= \mathbb{E}[zx]^{-1}\mathbb{E}[zy]
\end{aligned}
$$

By replacing the population moments with sample moments we get the two-stage least squares estimator:

$$
\hat{\beta}_{IV} = (Z'X)^{-1}(Z'Y) \to^p \beta
$$

[4]Note that our assumption that $E(x'\varepsilon)=0$ and $E(x'z)\neq 0$ implies that $E(z'\varepsilon)=0$.

But we don't observe X. So what we end up estimating is (consistency of 2SLS is shown in Hanson 12.12, and 2SLS is a generalisation of IV):

$$\begin{aligned}
\mathbb{E}\left[z\tilde{x}\right]^{-1}\mathbb{E}\left[zy\right] &= \mathbb{E}\left[z(x+v)\right]^{-1}\mathbb{E}\left[zy\right] \\
&= \frac{\mathbb{E}\left[zy\right]}{\mathbb{E}\left[zx+zv\right]} \\
&= \frac{\mathbb{E}\left[zy\right]}{\mathbb{E}\left[zx\right]}, \qquad\qquad \text{bc. } \mathbb{E}\left[zv\right]=0 \\
&= \beta
\end{aligned}$$

So, the IV estimator is consistent, despite measurement error. Moreover we can show:

$$\begin{aligned}
\frac{\mathbb{E}\left[zy\right]}{\mathbb{E}\left[zx\right]} &= \frac{\mathbb{E}\left[zy\right]-\mathbb{E}\left[z\right]\mathbb{E}\left[y\right]}{\mathbb{E}\left[zx\right]-\mathbb{E}\left[z\right]\mathbb{E}\left[x\right]}, \quad \text{bc. } \mathbb{E}\left[y\right]=\mathbb{E}\left[x\right]0 \\
&= \frac{Cov(z,y)}{Cov(z,x)} \\
&= \frac{\frac{Cov(z,y)}{Var(z)}}{\frac{Cov(z,x)}{Var(z)}}
\end{aligned}$$

That is, the coefficient from regressing y on z (the first stage) over the coefficient from regressing x on z (the reduced form).

Question from the seminar A question came about the finite sample bias of 2SLS. This issue is explained in Hanson 12.14. The bottom line is that samples need to be sufficiently large for 2SLS to work (that is larger than it would be for OLS with no endogeneity issues).

Metrics 1 PS4

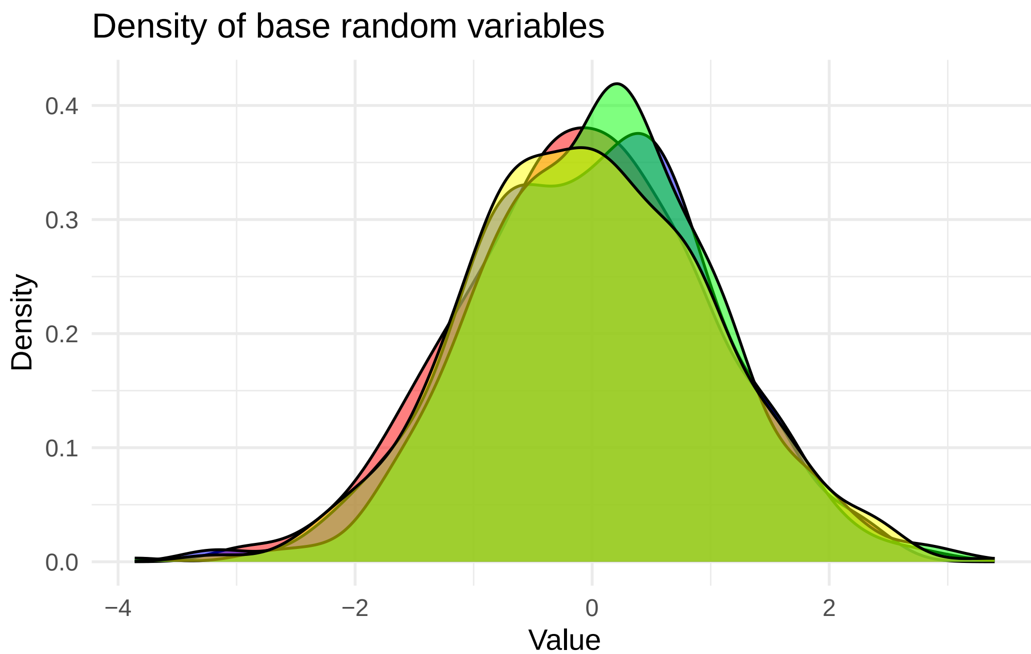
Artschil Okropiridse

1. Law of iterated expectations.

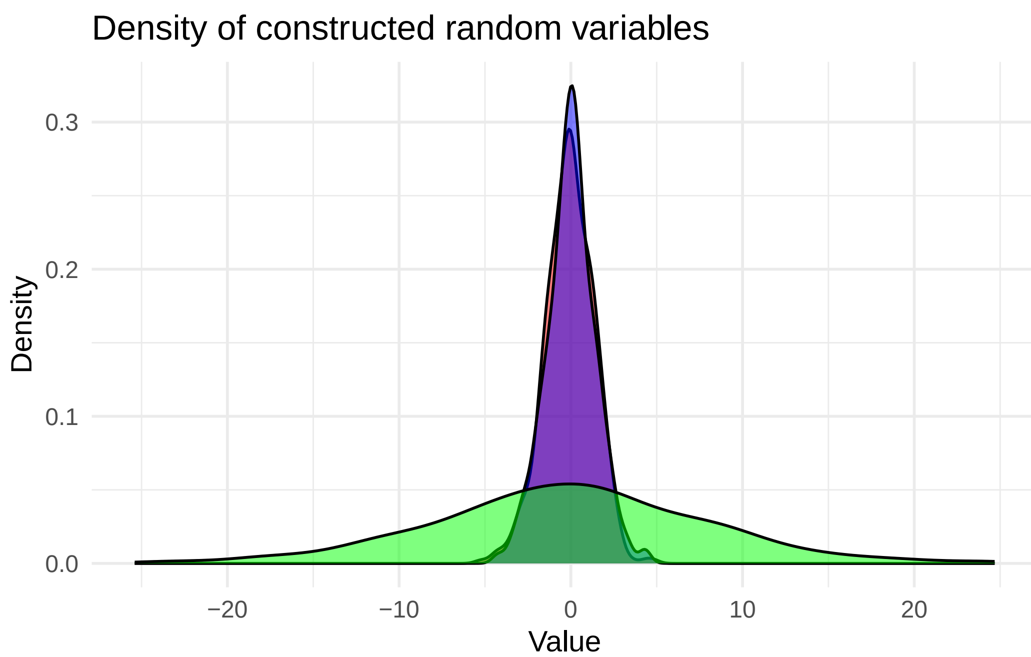
(b) OLS as conditional expectation

```
# Let's generate the data according to the given DGPs
n <- 500
set.seed(8)

## Setting parameters
sigma_sq <- 1
beta2 <- 5

## Creating base random variables
u0      <- rnorm(n, mean = 0, sd = sqrt(sigma_sq))
u1      <- rnorm(n, mean = 0, sd = 1) ## note that sqrt(1) = 1, so sd = var
u2      <- rnorm(n, mean = 0, sd = 1)
epsilon <- rnorm(n, mean = 0, sd = 1)

## Creating constructed random variables
x1   <- u0 + u1
x2   <- u0 + u2
y    <- x1 + beta2*x2 + epsilon

## Creating a tidy dataset with all the rvs
df <- tibble(u0, u1, u2, epsilon, x1, x2, y)

## Plotting all base rvs
ggplot() +
  geom_density(aes(x = u0), fill = "red", alpha = 0.5) +
  geom_density(aes(x = u1), fill = "blue", alpha = 0.5) +
```

```r
  geom_density(aes(x = u2), fill = "green", alpha = 0.5) +
  geom_density(aes(x = epsilon), fill = "yellow", alpha = 0.5) +
  labs(title = "Density of base random variables", x = "Value", y = "Density") +
  theme_minimal()
```




```r
## Plotting all constructed rvs
ggplot() +
  geom_density(aes(x = x1), fill = "red", alpha = 0.5) +
  geom_density(aes(x = x2), fill = "blue", alpha = 0.5) +
  geom_density(aes(x = y), fill = "green", alpha = 0.5) +
  labs(title = "Density of constructed random variables",
       x = "Value", y = "Density") +
  theme_minimal()
```



The plots are of course not necessary but help to build some visual understanding.

(i)

Start with $\sigma^2 = 1$ and $\beta_2 = 5$. What is the correlation between x_1 and x_2?

```r
## Using base R
cor(df$x1, df$x2)
```

```
[1] 0.513135
```

```r
## Using dplyr
df %>% summarise(cor(x1, x2))
```

```
# A tibble: 1 x 1
  `cor(x1, x2)`
          <dbl>
1         0.513
```

We can also derive the population (as in, not the sample) correlation algebraically:

$$\rho_{x_1,x_2} = \frac{Cov(x_1,x_2)}{\sqrt{Var(x_1)Var(x_2)}} = \frac{Var(u_0)}{\sqrt{Var(u_0)+Var(u_1)}\sqrt{Var(u_0)+Var(u_2)}} = \frac{1}{2}$$

(ii)

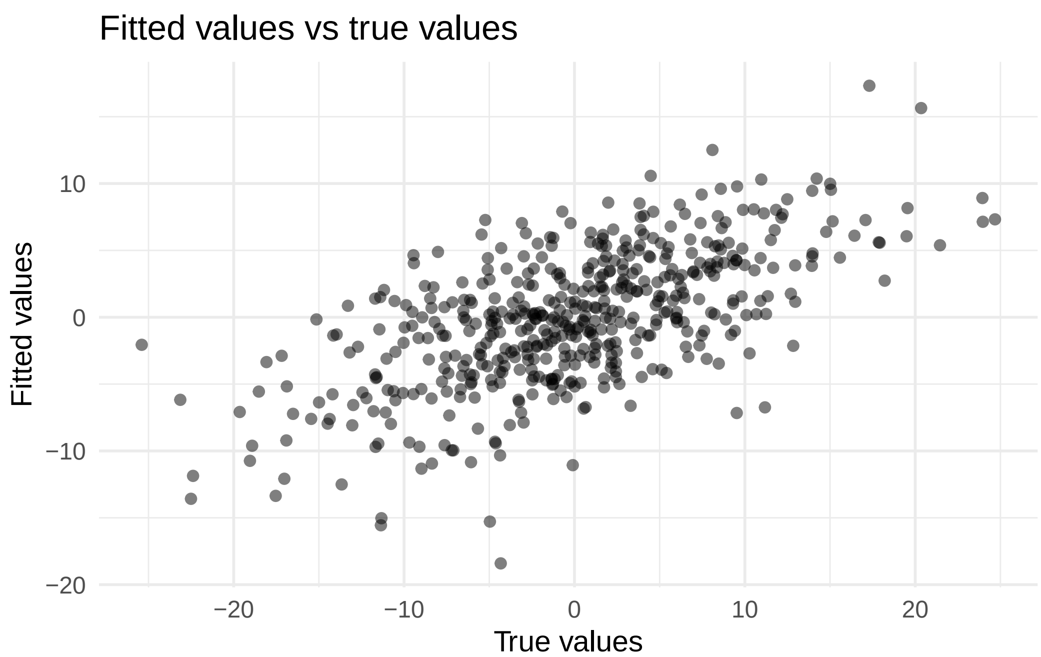
Let's pretend to be Björn. We ask Ana to provide us with an estimate of the short equation:

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

```r
## Let's not use the in-built functions but actually write this in terms of the
## projection matrix
Xs <- cbind(1, x1)
P_Xs <- Xs %*% solve(t(Xs) %*% Xs) %*% t(Xs)

## So our fitted values are:
y_hat1 <- P_Xs %*% y
MSE1 <- sum((y - y_hat1)^2)/n

## Let's get min, mean, max and so on for y_hat1
summary(y_hat1)
```

```
       V1
 Min.   :-18.41004
 1st Qu.: -3.20169
 Median : -0.03716
 Mean   : -0.09462
 3rd Qu.:  3.48602
 Max.   : 17.31519
```

```r
## Let's plot the fitted values against the true values
ggplot() +
  geom_point(aes(x = y, y = y_hat1), alpha = 0.5) +
  labs(title = "Fitted values vs true values",
       x = "True values", y = "Fitted values") +
  theme_minimal()
```




```
## The MSE is
MSE1
```

```
[1] 38.87233
```

(iii)

Now we ask Ana to provide us with an estimate of the long equation:

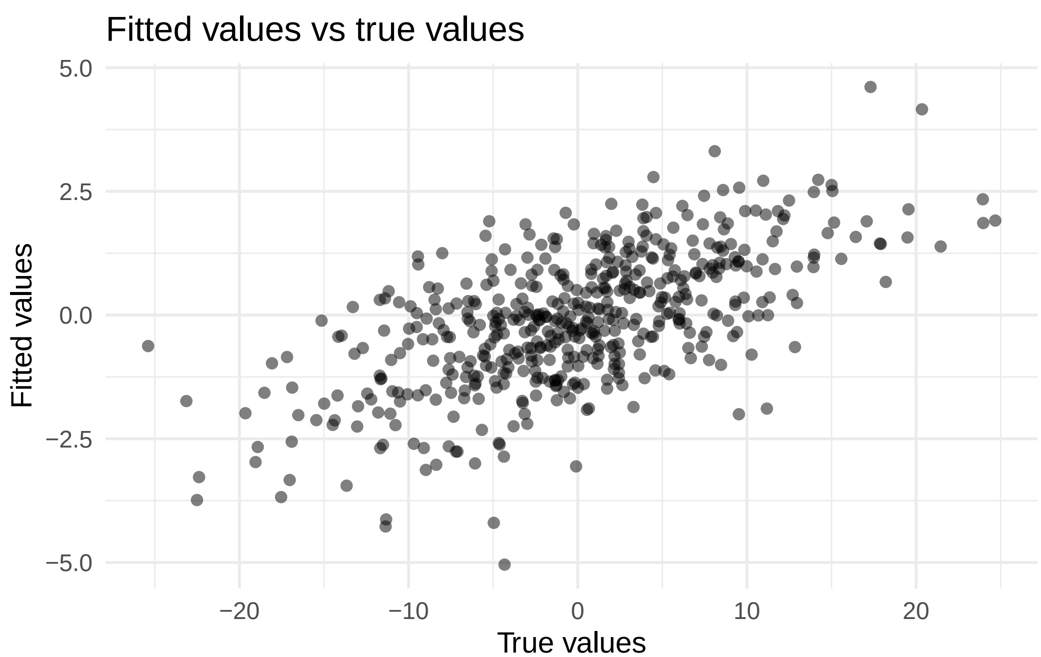
$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \varepsilon$$

```
## Let's now use the formula of the OLS estimator
Xl <- cbind(1, x1, x2)
beta_hatl <- solve(t(Xl) %*% Xl) %*% t(Xl) %*% y

## For Björn's fitted values we need to get the average of x_2
x2_bar <- mean(x2)
Xl_björn <- cbind(1, x1, x2_bar)

## So Björn's fitted values are:
y_hat2 <- Xl_björn %*% beta_hatl
```

```r
MSE2 <- sum((y - y_hat2)^2)/n

## Let's get min, mean, max and so on for y_hat2
summary(y_hat2)
```

```
       V1
 Min.   :-5.04370
 1st Qu.:-0.93419
 Median :-0.07909
 Mean   :-0.09462
 3rd Qu.: 0.87292
 Max.   : 4.60975
```

```r
## Let's plot the fitted values against the true values
ggplot() +
  geom_point(aes(x = y, y = y_hat2), alpha = 0.5) +
  labs(title = "Fitted values vs true values",
       x = "True values", y = "Fitted values") +
  theme_minimal()
```



```r
## The MSE is
MSE2
```

```
[1] 52.38313
```

(iv)

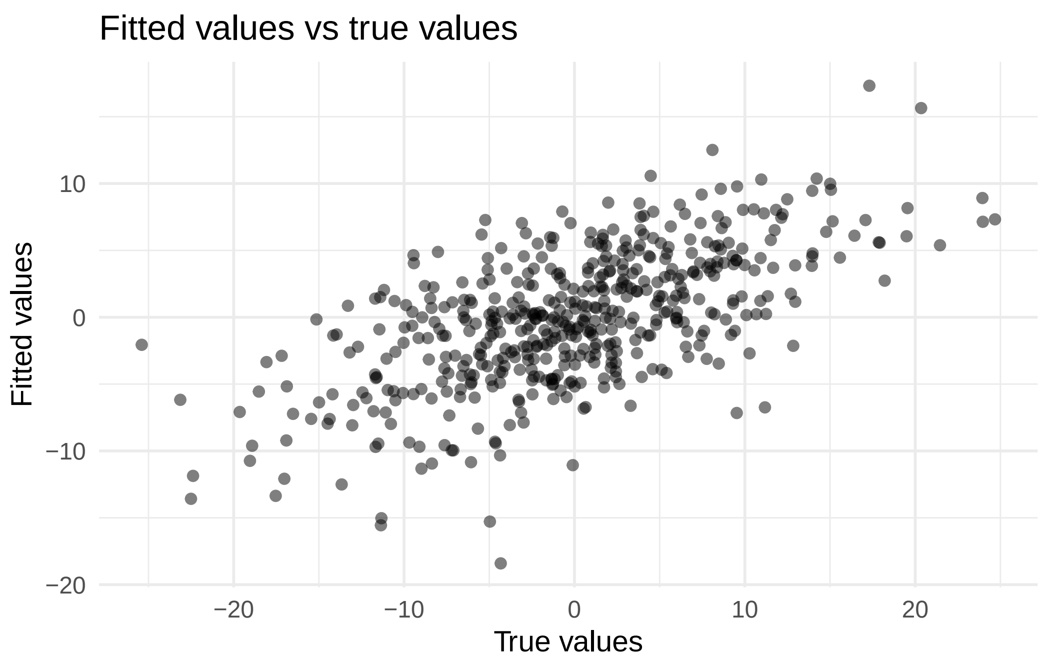
Lastly, we ask Ana to provide an estimate of a different equation namely:

$$x_2 = \gamma_0 + \gamma_1 x_1 + \nu$$

```r
## Let's first get fitted values for x_2 (they only contain information
## from x_1, which Björn has access to).
x2_hat <- P_Xs %*% x2

## So Björn's variables are
Xl_björn2 <- cbind(1, x1, x2_hat)

## OLS coefficients
y_hat3 <- Xl_björn2 %*% beta_hatl
MSE3 <- sum((y - y_hat3)^2)/n

## Let's get min, mean, max and so on for y_hat3
summary(y_hat3)
```

```
       V1
 Min.   :-18.41004
 1st Qu.: -3.20169
 Median : -0.03716
 Mean   : -0.09462
 3rd Qu.:  3.48602
 Max.   : 17.31519
```

```r
## Let's plot the fitted values against the true values
ggplot() +
  geom_point(aes(x = y, y = y_hat3), alpha = 0.5) +
  labs(title = "Fitted values vs true values",
       x = "True values", y = "Fitted values") +
  theme_minimal()
```




```r
## The MSE is
MSE3
```

```
[1] 38.87233
```

(v)

```r
## Let's get the correlation coefficients for all three fitted vectors and
## print them
cor(y_hat1, y_hat2)
```

```
     [,1]
[1,]    1
```

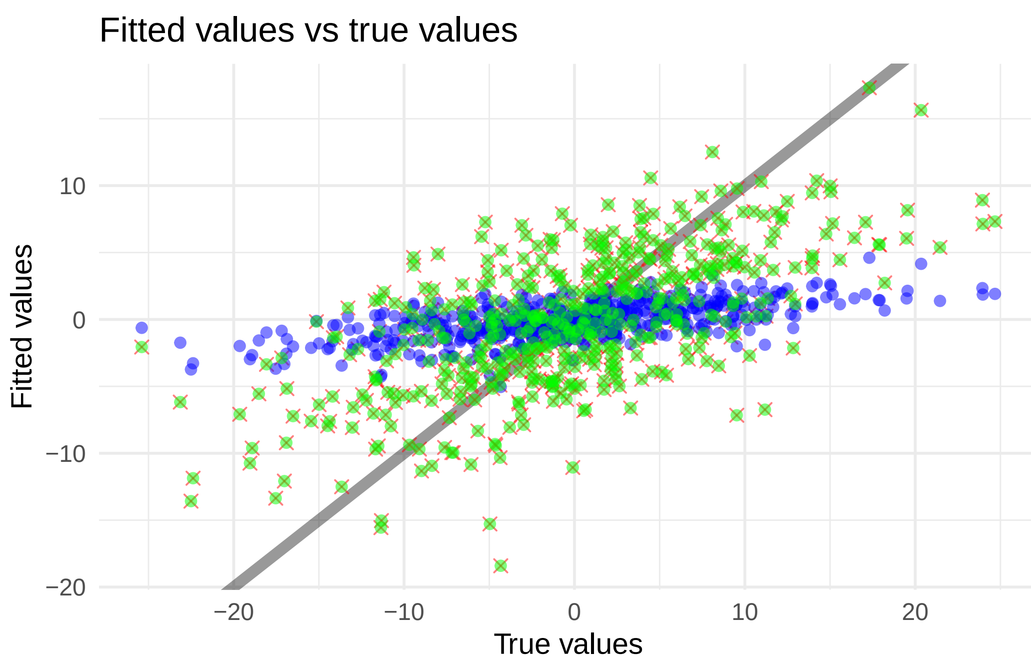
```r
cor(y_hat1, y_hat3)
```

```
     [,1]
[1,]    1
```

```r
cor(y_hat2, y_hat3)
```

```
     [,1]
[1,]    1
```

```r
## To see this let's plot all three fitted values against the true y
ggplot() +
  geom_abline(intercept = 0, slope = 1, color = "black", size = 2, alpha = .4) +
  geom_point(aes(x = y, y = y_hat1), alpha = 0.5, color = "red", shape = 4, size = 2) +
  geom_point(aes(x = y, y = y_hat2), alpha = 0.5, color = "blue") +
  geom_point(aes(x = y, y = y_hat3), alpha = 0.5, color = "green") +
  labs(title = "Fitted values vs true values",
       x = "True values", y = "Fitted values") +
  theme_minimal()
```



The point to understand here is that in the short regression, we are already using all information we have. No matter what transformation we "squeeze" x_1 through, we will not be able to get a better fit. Or in the projection interpretation, we only have the space spanned by x_1 to "place" a fitted value. But the linear projection already get's us as close to the true y as possible (in OLS we define close with the Euclidian distance, in principal nothing would stop us from defining it using the l_1 or l_3 or any other norm).

For $\hat{y}^{(2)}$, note that the coefficient for x_1 in the short equation is around 3.5 while it is around 1 in the long equation, that is why $\hat{y}^{(2)}$ has much less variance than $\hat{y}^{(1)}$ and $\hat{y}^{(3)}$. Since $\hat{\beta}_0 + \hat{\beta}_2 \bar{x}_2$ is a constant and by using a different $\hat{\beta}_1$ we are just scaling the values of x_1 (by roughly 3.5 in this example). The pearson correlation coefficient does not care about such transformations - i.e. denoting $\rho(.,.)$ as a correlation coefficient, $\rho(Y+1, X) = \rho(Y, X)$ and $\rho(aY, X) = \rho(Y, X)$. That's why the correlations are all 1 while the MSEs differ. This point is quite crucial for the rest of the problem set. So if you don't understand why these two equations hold, it might make sense to derive it yourself now.

In terms of the conditional expectation function (CEF), in each case we are just conditioning on x_1. So even if we take some $\mathbb{E}[y|x_1, x_2]$ we got from Ana, once we condition on Björn's information we just end up with $\mathbb{E}[y|x_1]$.

(c)

```
rm(list=ls())
set.seed(8)

## Since we will use the whole procedure a few times we will write a function
## with default values for it
run_simulation <- function(m, n_m, sig_sq1 = 1, sig_sq2 = 1, sig_sqe = 1,
                           beta0 = 1, betag = 1, beta1 = 1, beta2 = 1, power = 1){

  n <- m*n_m

  ## Setting municipality level variables
  mu_1m <- rnorm(m, mean = 0, sd = sqrt(sig_sq1))
  mu_2m <- rnorm(m, mean = 0, sd = sqrt(sig_sq2))

  p_m <- (mu_1m^power - min(mu_1m^power))/(max(mu_1m^power) - min(mu_1m^power))
  g_m <- rbinom(m, prob = p_m, size = 1)
  cluster <- seq(1:m)

  ## Setting individual level variables dependent on municipality
  cluster_i <- rep(cluster, each = n_m)

  g <- rep(g_m, each = n_m)


  test <- rnorm(n, mean = mu_1m[cluster_i], sd = 0)
  test
```

```r
  x_1 <- rnorm(n, mean = mu_1m[cluster_i], sd = 1)
  x_2 <- rnorm(n, mean = mu_2m[cluster_i], sd = 1)
  e <- rnorm(n, mean = 0, sd = sqrt(sig_sqe))

  y <- beta0 + betag*g + beta1*x_1 + beta2*x_2 + e

  ## Let's have a tibble as the output of the function
  return(tibble(cluster_i, x_1, x_2, e, y, g))
}
```

(i)

```r
## Setting parameters
m <- 50
n_m <- 100

sig_sq1 <- 1
sig_sq2 <- 1
sig_sqe <- 1

beta0 <- 1
betag <- 1
beta1 <- 1
beta2 <- 1

power <- 1

df <- run_simulation(m, n_m, sig_sq1, sig_sq2, sig_sqe,
                     beta0, betag, beta1, beta2, power)

## Let's do use the inbuilt R functions this time
model_1 <- lm(y ~ x_1 + x_2 + g, data = df)

stargazer(model_1, type = "text")
```

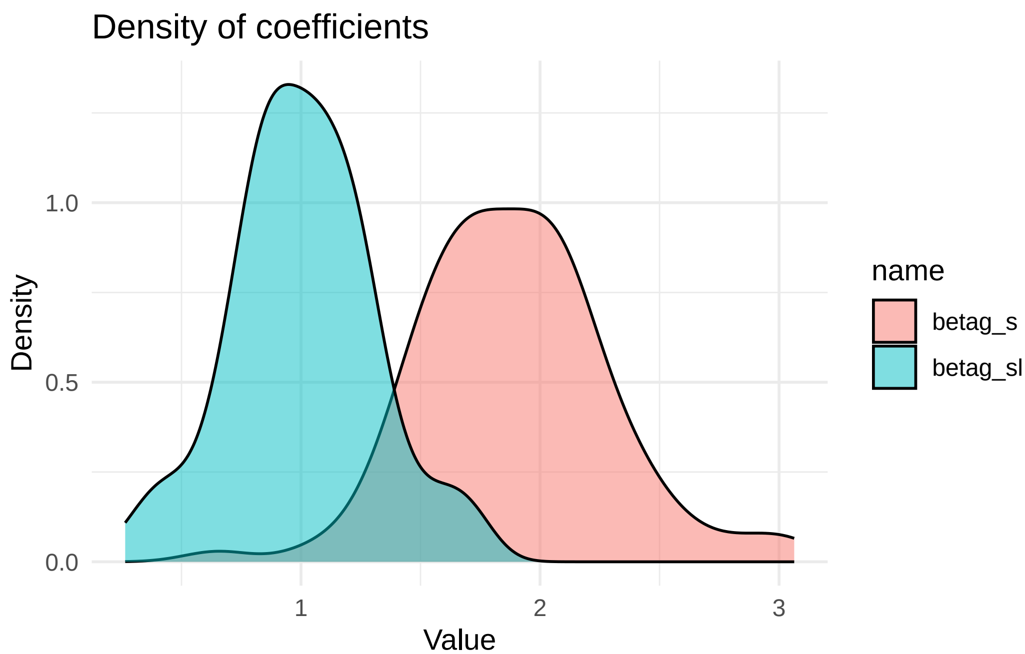
```
===============================================
                        Dependent variable:
                    ---------------------------
                                 y
```

```
-----------------------------------------------
x_1                          0.996***
                              (0.010)

x_2                          1.021***
                              (0.009)

g                            1.007***
                              (0.031)

Constant                     1.016***
                              (0.025)

-----------------------------------------------
Observations                    5,000
R2                              0.847
Adjusted R2                     0.847
Residual Std. Error      0.996 (df = 4996)
F Statistic         9,214.542*** (df = 3; 4996)
===============================================
Note:               *p<0.1; **p<0.05; ***p<0.01
```

(ii), (iii)

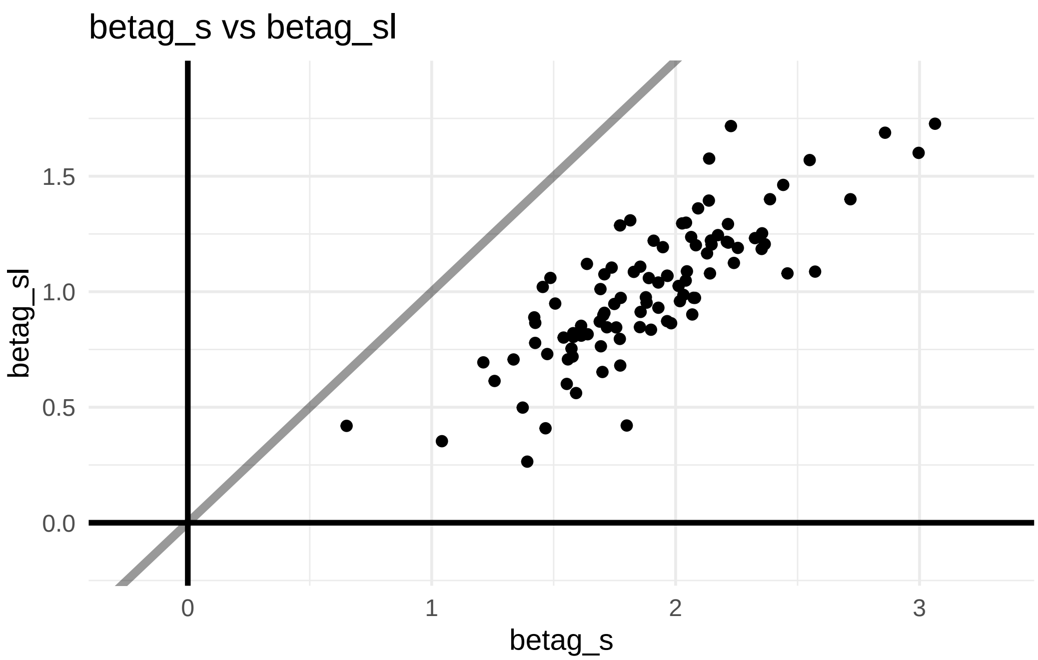
```
## Let's see how different the two models do, by running the simulation several
## times for each.
n_sims <- 100

coefs <- tibble(betag_s = numeric(), betag_sl = numeric())

for (i in 1:n_sims) {
  df <- run_simulation(m, n_m, sig_sq1, sig_sq2, sig_sqe,
                       beta0, betag, beta1, beta2, power)

  model_s  <- lm(y ~ g, data = df)
  model_sl <- lm(y ~ x_1 + g, data = df)

  coefs <- coefs |>
    add_row(
      betag_s = coef(model_s)[2],
      betag_sl = coef(model_sl)[3]
```

```
    )
}

## Let's plot the distributions
coefs |>
  pivot_longer(cols = everything()) |>
  ggplot() +
  geom_density(aes(x = value, fill = name), alpha = 0.5) +
  labs(title = "Density of coefficients", x = "Value", y = "Density") +
  theme_minimal()
```




```
## Or against each other, including the origin in the plot
coefs |>
  ggplot(aes(x = betag_s, y = betag_sl)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "black", size = 1.5, alpha = .4) +
  geom_vline(xintercept = 0, color = "black", size = 1) +  # For the y-axis
  geom_hline(yintercept = 0, color = "black", size = 1) +  # For the x-axis
  labs(title = "betag_s vs betag_sl",
       x = "betag_s", y = "betag_sl") +
  theme_minimal() +
```

```r
  scale_x_continuous(expand = c(.1, .1), limits = c(0, NA)) +
  scale_y_continuous(expand = c(.1, .1), limits = c(0, NA))
```



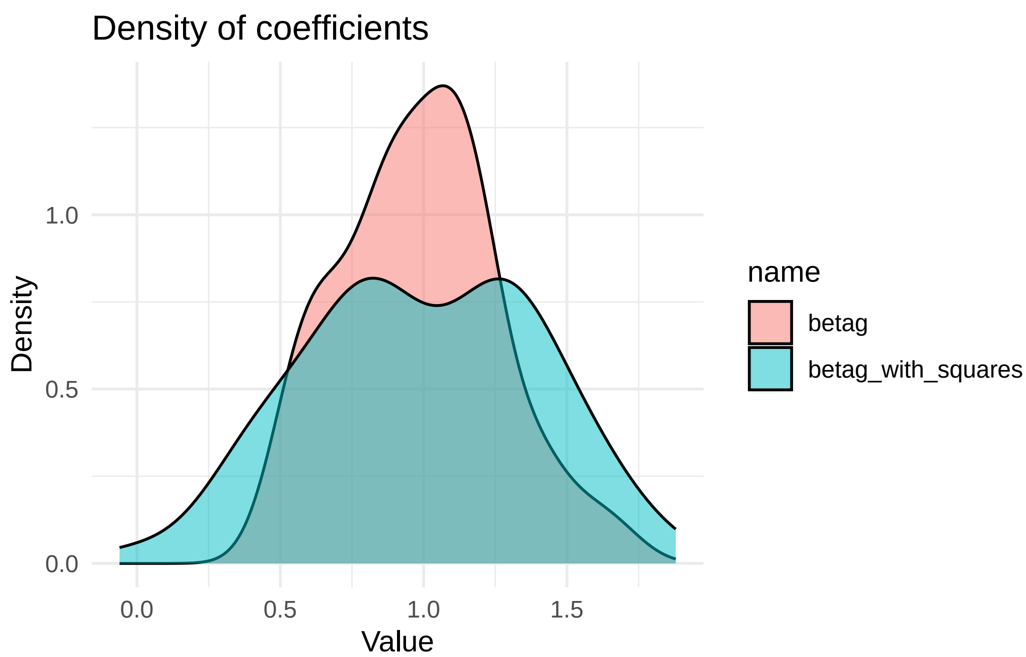
While the slightly longer regression gives us estimates very close to the true values, the short regression spits out a quite inflated $\hat{\beta}_g$. This can be seen in both graphs.

(iii)

extra

```r
## Let's run the simulation 50 times with and without the squares and store the
## coefficients so that we can compare the coefficients properly
n_sims <- 100
power_e <- 2

coefs <- tibble(betag_with_squares = numeric(),
                betag = numeric())

for (i in 1:n_sims) {
  ## Run simulation with squares
```

```r
  df_extra <- run_simulation(m, n_m, sig_sq1, sig_sq2, sig_sqe,
                             beta0, betag, beta1, beta2, power_e)
  model_l <- lm(y ~ g + x_1, data = df_extra)

  betag_with_squares <- coef(model_l)[2]


  ## Run it without squares
  df_extra <- run_simulation(m, n_m, sig_sq1, sig_sq2, sig_sqe,
                             beta0, betag, beta1, beta2, power)
  model_l <- lm(y ~ g + x_1, data = df_extra)

  betag_without_squares <- coef(model_l)[2]

  coefs <- coefs |>
    add_row(
      betag_with_squares = betag_with_squares,
      betag = betag_without_squares
    )
}

## Now let's plot both densities for both estimates
coefs |>
  pivot_longer(cols = everything()) |>
  ggplot() +
  geom_density(aes(x = value, fill = name), alpha = 0.5) +
  labs(title = "Density of coefficients", x = "Value", y = "Density") +
  theme_minimal()
```

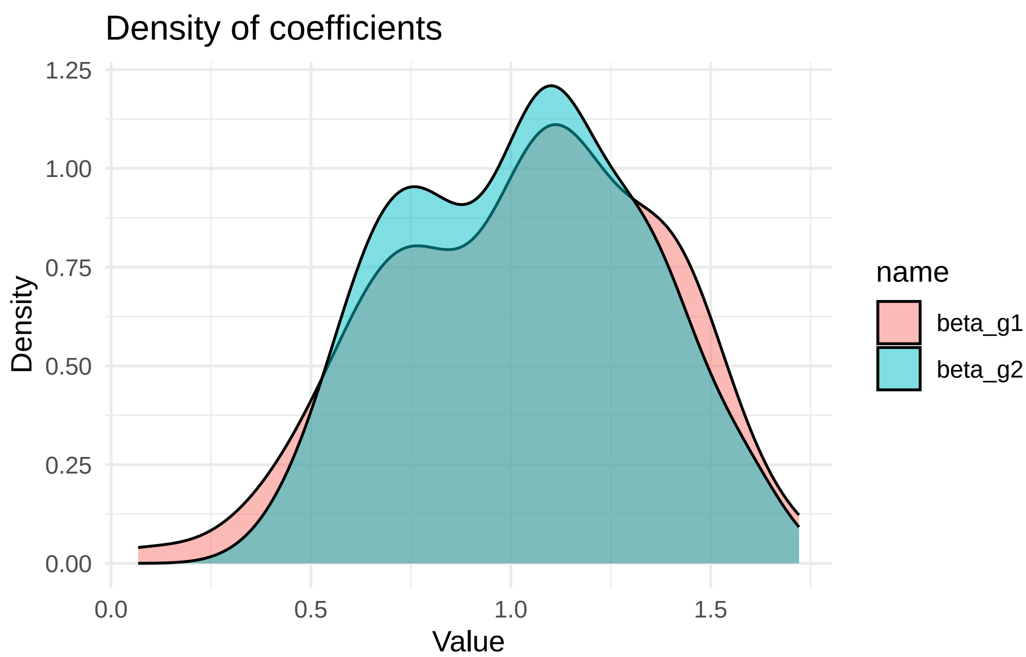


Note that the results are quite variable, every time one repeats the simulation coefficients turn out quite differently. So in case your estimates are not like mine, this might just be by chance.

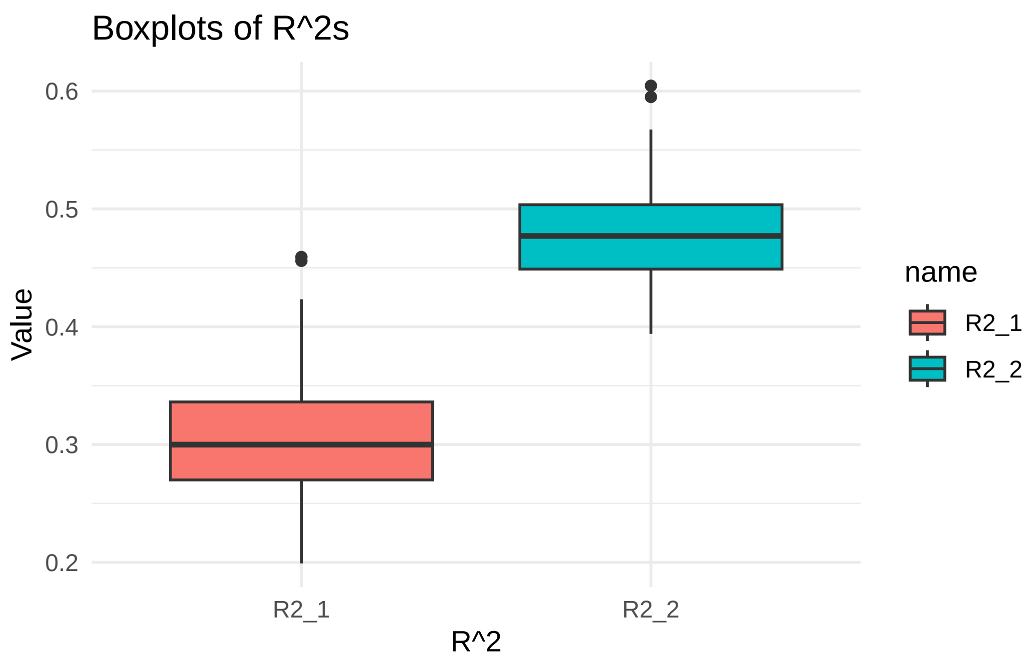
(iv)

```
## Let's run the reg a few times again
n_sims <- 100

coefs_R2 <- tibble(R2_1 = numeric(),
                   R2_2 = numeric(),
                beta_g1 = numeric(),
                beta_g2 = numeric())

for (i in 1:n_sims) {
  df_simuliv <- run_simulation(m, n_m, sig_sq1, sig_sq2, sig_sqe,
                               beta0, betag, beta1, beta2, power)

  ## Let's calculate the average x_1 per municipality m
  df_simuliv <- df_simuliv |>
    group_by(cluster_i) |>
```

```r
    mutate(x1_bar = mean(x_1)) |>
    ungroup()

  model_iv1 <- lm(y ~ x1_bar + g, data = df_simuliv)
  model_iv2 <- lm(y ~ x_1 + g, data = df_simuliv)


  ## Let's store R^2 and coeff estimate
  coefs_R2 <- coefs_R2 |>
    add_row(
      R2_1 = summary(model_iv1)$r.squared,
      beta_g1 = coef(model_iv1)[3],
      R2_2 = summary(model_iv2)$r.squared,
      beta_g2 = coef(model_iv2)[3]
    )
}

## Let's plot the coefficients
coefs_R2 |>
  pivot_longer(cols = c(beta_g1, beta_g2)) |>
  ggplot() +
  geom_density(aes(x = value, fill = name), alpha = 0.5) +
  labs(title = "Density of coefficients", x = "Value", y = "Density") +
  theme_minimal()
```




```r
## Now boxplots for the R^2s
coefs_R2 |>
  pivot_longer(cols = c(R2_1, R2_2)) |>
  ggplot() +
  geom_boxplot(aes(x = name, y = value, fill = name)) +
  labs(title = "Boxplots of R^2s", x = "R^2", y = "Value") +
  theme_minimal()
```

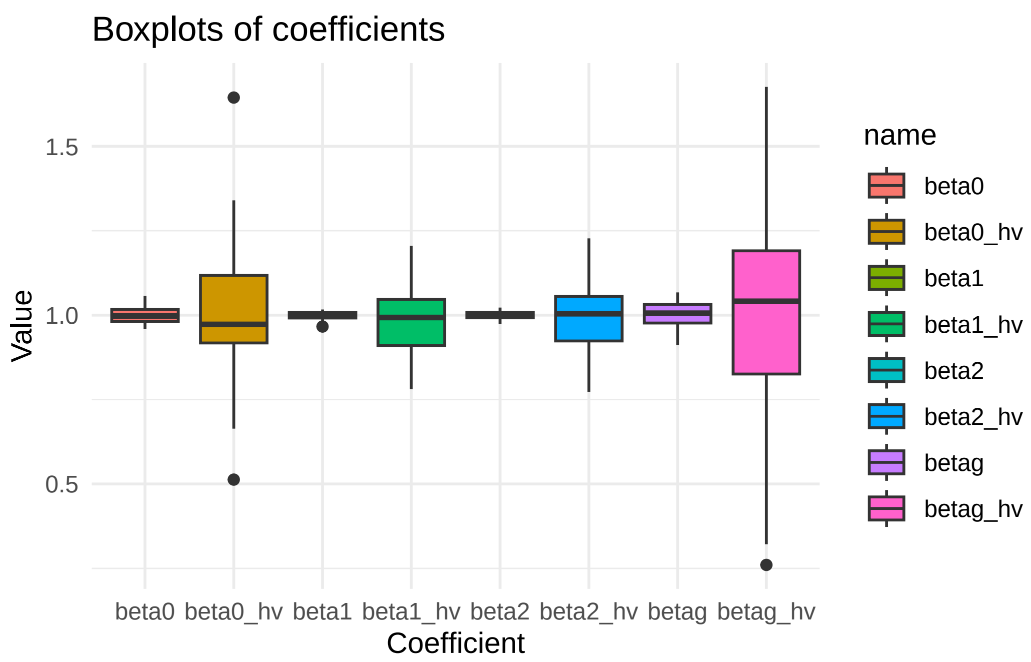


Again the results are quite variable. The R^2 is persistenly lower when using $\bar{x}_1$, while the coefficient estimate is neither always better nor always worse. This makes sense because treatment g varies on the municipality, not on the individual level.

(v)

```
## Let's run the simulation 50 times twice, and store the coefficients. The
## first time everything is as before. The second time variances have increased.
n_sims <- 50
coefs <- tibble(beta0 = numeric(), beta0_hv = numeric(),
                beta1 = numeric(), beta1_hv = numeric(),
                beta2 = numeric(), beta2_hv = numeric(),
                betag = numeric(), betag_hv = numeric())

for (i in 1:n_sims){
  df_simulv <- run_simulation(m, n_m)
  model_l <- lm(y ~ x_1 + x_2 + g, data = df_simulv)

  df_simulv2 <- run_simulation(m, n_m, sig_sqe  = 100)
  model_l2 <- lm(y ~ x_1 + x_2 + g, data = df_simulv2)
```

```r
  coefs <- coefs |>
    add_row(
      beta0 = coef(model_l)[1],
      beta1 = coef(model_l)[2],
      beta2 = coef(model_l)[3],
      betag = coef(model_l)[4],

      beta0_hv = coef(model_l2)[1],
      beta1_hv = coef(model_l2)[2],
      beta2_hv = coef(model_l2)[3],
      betag_hv = coef(model_l2)[4]
    )

}

## Let's plot the coefficients
coefs |>
  pivot_longer(cols = everything()) |>
  ggplot() +
  geom_boxplot(aes(x = name, y = value, fill = name)) +
  labs(title = "Boxplots of coefficients", x = "Coefficient", y = "Value") +
  theme_minimal()
```



By just adding a ton of noise to the error term, we end up with much more variant coefficients. Their means however, are all very close to the value we would expect.

(vi), (vii)

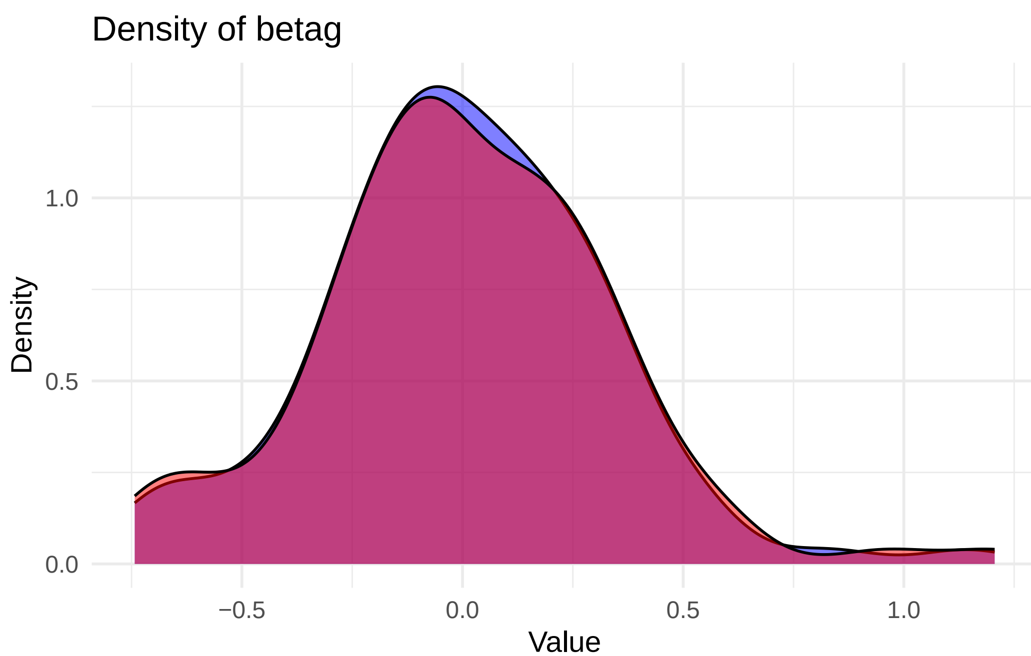
```r
n_sims <- 100

## Let's create an empty table to store the coefficients
coefs <- tibble(betag = numeric(), betag_pval = numeric(),
                betag_bar = numeric(), betag_pval_bar = numeric()
                )

## Let's run the simulation many times and store the coefficients
for (i in 1:n_sims){

  df_simulvi <- run_simulation(m, n_m, betag = 0, sig_sq1 = 10)
  model_l <- lm(y ~ g + x_1, data = df_simulvi)

  ## Let's collapse on y, x1 and g
  df_simulvii <- df_simulvi |>
  group_by(cluster_i) |>
  summarise(y_mbar = mean(y), x_1bar = mean(x_1), x_2bar = mean(x_2) , g = mean(g))

  model_l_bar <- lm(y_mbar ~ g + x_1bar, data = df_simulvii)

  coefs <- coefs |>
    add_row(
      betag = coef(model_l)[2],
      betag_pval = summary(model_l)$coefficients[2,4],

      betag_bar = coef(model_l_bar)[2],
      betag_pval_bar = summary(model_l_bar)$coefficients[2,4]
      )
}

## How often is betag significant, in each case?
coefs |>
  filter(betag_pval < 0.05) |>
  nrow()
```

```
[1] 69
```

```r
coefs |>
  filter(betag_pval_bar < 0.05) |>
  nrow()
```

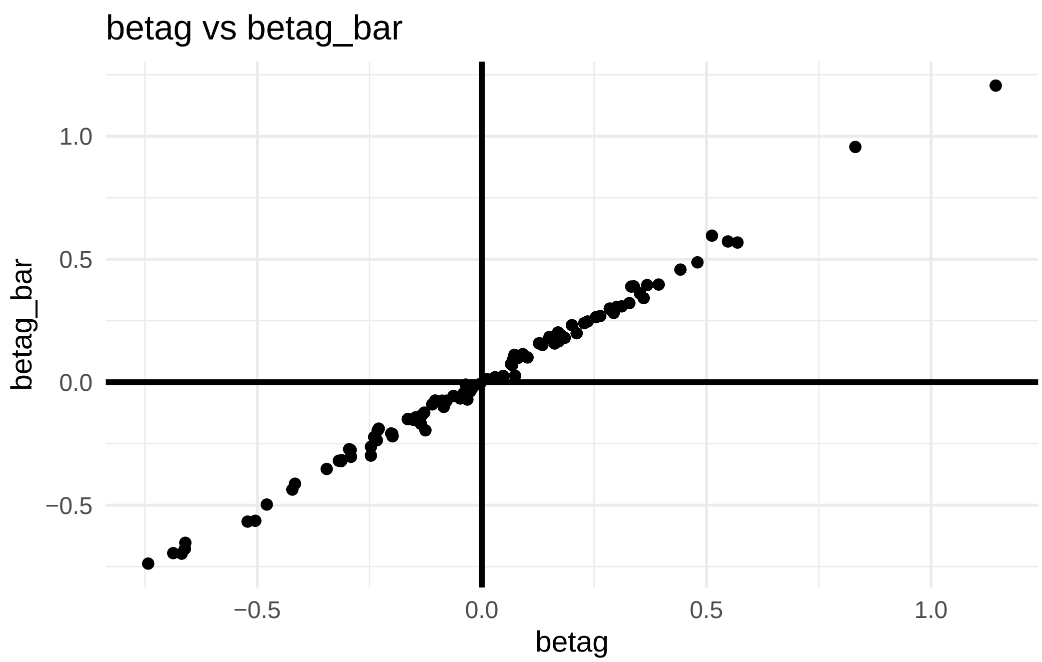
```
[1] 6
```

```r
## Let's plot the distribution of both coefficients with extended x-axis
coefs |>
  ggplot() +
  geom_density(aes(x = betag), fill = "blue", alpha = 0.5) +
  geom_density(aes(x = betag_bar), fill = "red", alpha = 0.5) +
  labs(title = "Density of betag", x = "Value", y = "Density") +
  theme_minimal()
```




```r
## Let's plot both coefficients against each other
coefs |>
  ggplot(aes(x = betag, y = betag_bar)) +
  geom_point() +
  geom_vline(xintercept = 0, color = "black", size = 1) +  # For the y-axis
  geom_hline(yintercept = 0, color = "black", size = 1) +  # For the x-axis
```

```
labs(title = "betag vs betag_bar",
     x = "betag", y = "betag_bar") +
theme_minimal()
```



The exact time $\hat{\beta}_g$ will be significant depends on chance. But it should be around 70% of the time. For (vi) and around 5% for (vii). Why is that the case? Why do they differ and why is it not 5% in the first case?

Looking at the coefficients (the last two plots), we can see that it is clearly not those that differ. I.e. aggregating on the treatment level does not affect the coefficients much. The difference therefore, must lie in the denominator of the t-stat (the test we use for constructing confidence intervals, more on that in the lectures to come). The t-stat in a test for a coefficient being equal to zero is: $t = \frac{\hat{\beta}_g - 0}{\sqrt{Var(\hat{\beta}_g)}}$. And when we don't aggregate we are calculating the variance in the denominator wrong, because our observations are not i.i.d., i.e. they are dependent across clusters. In our case we know that this dependence comes from the unobserved x_2. So if we could control for that, then we would no need to aggregate.

2. FWL, OV'B and LATE's

```
rm(list=ls())

## It will again be useful to write a function for the simulation
run_simulation <- function(n, earnings_mean = 19, c_gain_mean = 1,
                           sigma_sq_earnings = 1, sigma_sq_cgains = 1){
  ### The only thing that needs to be specified is n

  ## Let's create our rvs
  earnings <- rnorm(n, mean = earnings_mean, sd = sqrt(sigma_sq_earnings))
  c_gains  <- rnorm(n, mean = c_gain_mean, sd = sqrt(sigma_sq_cgains))
  u <- rnorm(n, mean = 0, sd = 1)
  e <- rnorm(n, mean = 0, sd = 1)

  ## Let's create our constructed rvs
  occ_st <- earnings + u
  child_oc <- earnings - c_gains + e
  income <- earnings + c_gains

  ## Let's put all our variables in a tibble
  return(tibble(earnings, c_gains, u, e, occ_st, child_oc, income))
}
```

(a)

```
## For this we can simply run the simulation once
df <- run_simulation(n = 500)

## So let's calculate the average
1/nrow(df) * sum(df$earnings/df$income)
```

```
[1] 0.9522471
```

```
## this is not the same as this:
sum(df$earnings)/sum(df$income)
```

```
[1] 0.9499237
```

While the two are not the same, in this particular example they are very close to each other.

(b)

Note that we can't know the effect of "income" as it is defined above, without knowing how much the constituents of income change. I.e. if the whole change is driven by an increase in Capital gains. Then we will have an average causal effect of -2. If the whole effects is driven by an increase in earnings, then we will have an average causal effect of 2. So there is no "right" answer. Also notice that OLS won't give more emphasis to the larger variable, because OLS only "cares" about variances (so if you increase earnings and capital gains both by 1, the effects will cancel out).

A second take-away from this exercise is that even knowing the DGP and having perfectly measured variables the true causal effect is a random variable and not a constant. Just as an example (this is not the "right" answer, it's just one arbitrary case), let's assume both factors increase by 10% (this way the larger variable increases by more, so its effect will dominate). Again let's start with calculations in "population-world". Let's denote the causal effect C and denote treatment t as 1 when an observation gets treated and 0 otherwise. We have

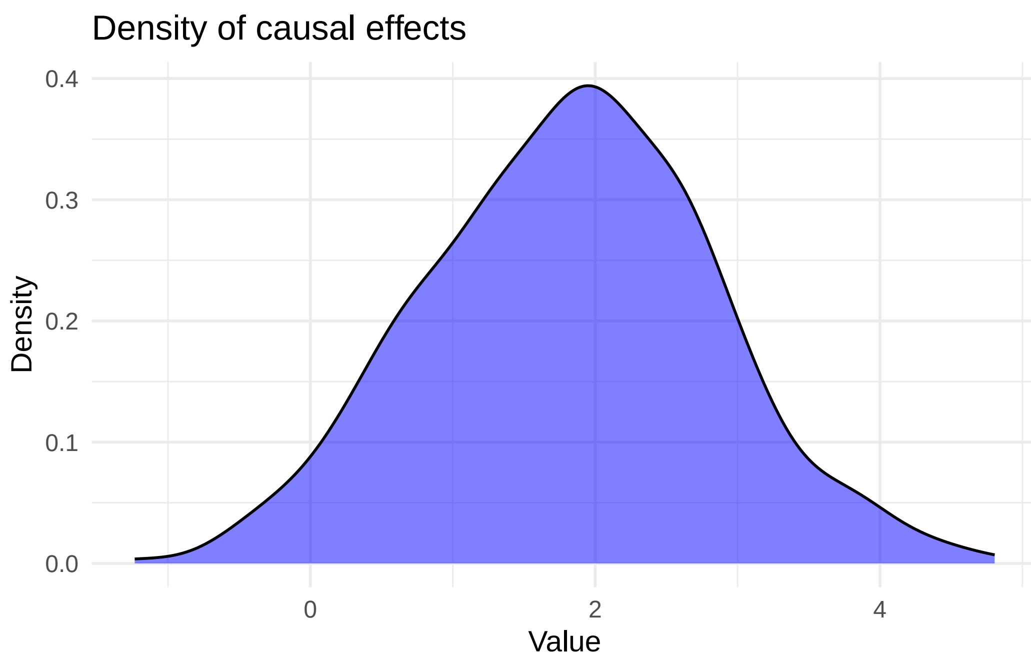
$$\begin{aligned} C_i &= child\ outcome_i(1) - child\ outcome_i(0) \\ &= (earnings_i - capital\ gains_i) * 1.1 - earnings_i - capital\ gains_i \\ &= 0.1 * (earnings_i - capital\ gains_i) \end{aligned}$$

If we want the average causal effect we can take expectations:

$$\mathbb{E}\left[C_i\right] = 0.1 * \mathbb{E}\left[Earnings_i - Capital\ gains_i\right] = 0.1 * (19 - 1) = 1.8$$

Note that for each child the exact value of the effect differs, depending on how large their specific parent's earnings and capital gains are. Now taking it to the data:

```r
## Let's calculate individual causal effects
df <- df |>
  mutate(t_earnings = 1.1 * earnings,
         t_c_gains  = 1.1 * c_gains,
         t_child_oc = t_earnings - t_c_gains,
         C_i = t_child_oc - child_oc)

## Let's plot the distribution of causal effects
df |> ggplot() +
  geom_density(aes(x = C_i), fill = "blue", alpha = 0.5) +
  labs(title = "Density of causal effects", x = "Value", y = "Density") +
  theme_minimal()
```



Note that the individual effects are quite variable. So even in a clean setting like this individuals can have very different causal effects.

(c)

Given the DGP the coefficient on earnings should be 1 and that on capital gains should be -1. Let's take it to the data:

```
## Let's run the two regressions
model1 <- lm(child_oc ~ earnings + c_gains, data = df)
model2 <- lm(child_oc ~ earnings + c_gains + occ_st, data = df)

## Let's compare the results
stargazer(model1, model2, type = "text")
```

```
=====================================================================
                                    Dependent variable:
                     ------------------------------------------------
                                         child_oc
                               (1)                        (2)
```

```
-----------------------------------------------------------------------
earnings                       1.047***                  0.990***
                               (0.046)                   (0.066)

c_gains                        -1.061***                 -1.062***
                               (0.045)                   (0.045)

occ_st                                                    0.054
                                                         (0.045)

Constant                        -0.859                    -0.798
                               (0.879)                   (0.881)

-----------------------------------------------------------------------
Observations                     500                       500
R2                              0.685                     0.685
Adjusted R2                     0.683                     0.684
Residual Std. Error       1.002 (df = 497)          1.001 (df = 496)
F Statistic         539.143*** (df = 2; 497) 360.216*** (df = 3; 496)
=======================================================================
Note:                                       *p<0.1; **p<0.05; ***p<0.01
```

The estimates are not precisely the same, but pretty close (earnings will change slightly more then capital gains, because it is correlated with occupational status).

(d)

```r
## Let's run the regression
model3 <- lm(child_oc ~ income, data = df)

## Let's compare the results
stargazer(model3, type = "text")
```

```
===============================================
                        Dependent variable:
                    ---------------------------
                             child_oc
-----------------------------------------------
income                        -0.042
```

```
                         (0.057)

Constant                 18.809***
                         (1.144)

-----------------------------------------------
Observations               500
R2                        0.001
Adjusted R2              -0.001
Residual Std. Error   1.780 (df = 498)
F Statistic         0.535 (df = 1; 498)
===============================================
Note:               *p<0.1; **p<0.05; ***p<0.01
```

```
rm(model1, model2, model3)
```

It's not super obvious how to interpret what we did here. Income is to a large extend earnings and only to a small extend capital gains, BUT both random variables have the same variance. And that is what OLS is picking up. To see why recall the regression anatomy formula (for the constant having a separate coefficient α):

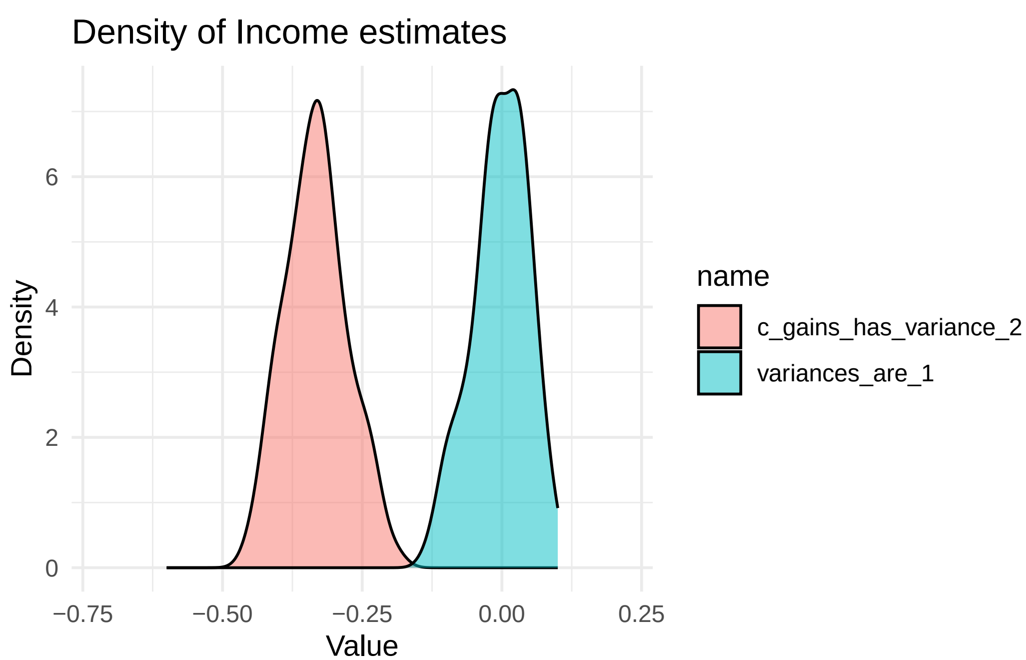
$$\beta = \left(Var\left[X\right]\right)^{-1} Cov\left[X, Y\right]$$

Essentially we are constraining the model such that earnings and capital gains have the same coefficient, and this results in the effect behaving weirdly. We can also have a look what happens if we play with the variances of each term (look at the code below, this is not necessary to answer the question but gives good intuition). We don't have endogeneity despite the misspecification. BUT that's only due to the variances being equal. See the digression further up in the document.

```
## Now let's repeat the same thing a few times and plot the coefficient of income
## from a regression like before and once with higher variance for capital gains
n_sims <- 100
coefs <- tibble(variances_are_1 = numeric(),
                c_gains_has_variance_2 = numeric())

for (i in 1:n_sims){
  df <- run_simulation(n = 500)
  model_3 <- lm(child_oc ~ income, data = df)
  coef_income <- coef(model_3)[2]
```

```
  df <- run_simulation(n = 500, sigma_sq_cgains = 2)
  model_3_high_variance <- lm(child_oc ~ income, data = df)
  coef_income_high_variance <- coef(model_3_high_variance)[2]


  coefs <- coefs |>
    add_row(variances_are_1 = coef(model_3)[2],
            c_gains_has_variance_2 = coef(model_3_high_variance)[2])
}

## plot the coefficients
coefs |>
  pivot_longer(cols = everything()) |>
  ggplot() +
  geom_density(aes(x = value, fill = name), alpha = 0.5) +
  labs(title = "Density of Income estimates", x = "Value", y = "Density") +
  theme_minimal() +
  scale_x_continuous(expand = c(.1, .1), limits = c(-0.6, .1))
```

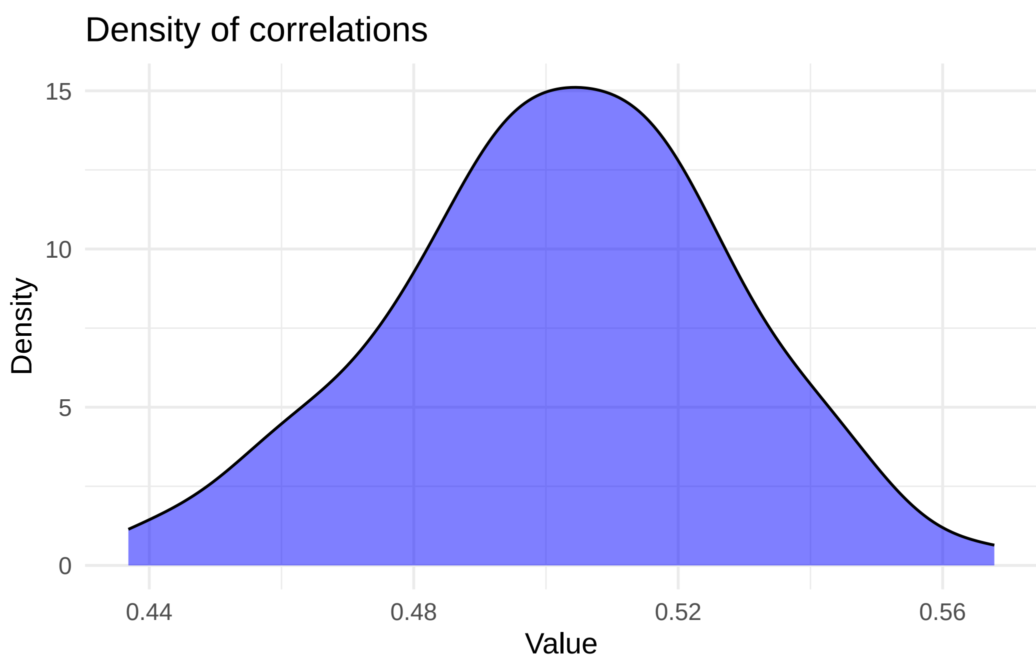
```
Warning: Removed 2 rows containing non-finite values (`stat_density()`).
```



```r
rm(coefs, df, model_3, model_3_high_variance)
```

(e)

```r
## Let's run the simulation a few times and store the correlation coefficients
n_sims <- 100
correlations <- numeric()

for (i in 1:n_sims){
  df <- run_simulation(500)
  correlations[i] <- cor(df$occ_st, df$income)
}

## Now let's plot their distributions
tibble(correlations) |>
  ggplot() +
  geom_density(aes(x = correlations), fill = "blue", alpha = 0.5) +
  labs(title = "Density of correlations", x = "Value", y = "Density") +
  theme_minimal()
```



This should be around $\frac{1}{2}$ as can be derived from the DGP.

(f)

The researcher thinks the world might look like this:

$$\textit{Child Outcomes} = \alpha + \beta_1 \ \textit{Income} + \beta_2 \textit{Occ. Status} + e$$

In which case we can derive the OVB from the short regression $\textit{Child Outcomes} = \gamma_0 + \gamma_1 \ \textit{Income} + \varepsilon$:

We can use the formula for a linear projection in the univariate case, to find γ_1, denoting Income I, Occ. status = O and Child Outcomes Y. So what the researcher expects might happen is:

$$\begin{aligned}
\gamma_1 &= \frac{\text{Cov}(I, Y)}{\text{Var}(I)} \\
&= \frac{\text{Cov}(I, \alpha + \beta_1 I + \beta_2 O + e)}{\text{Var}(I)} \\
&= \beta_1 + \beta_2 \frac{\text{Cov}(O, I)}{\text{Var}(I)}
\end{aligned}$$

Since the correlation between Income and Occ. Status is relatively large this would be a concern, as long as $\beta_2 \neq 0$.

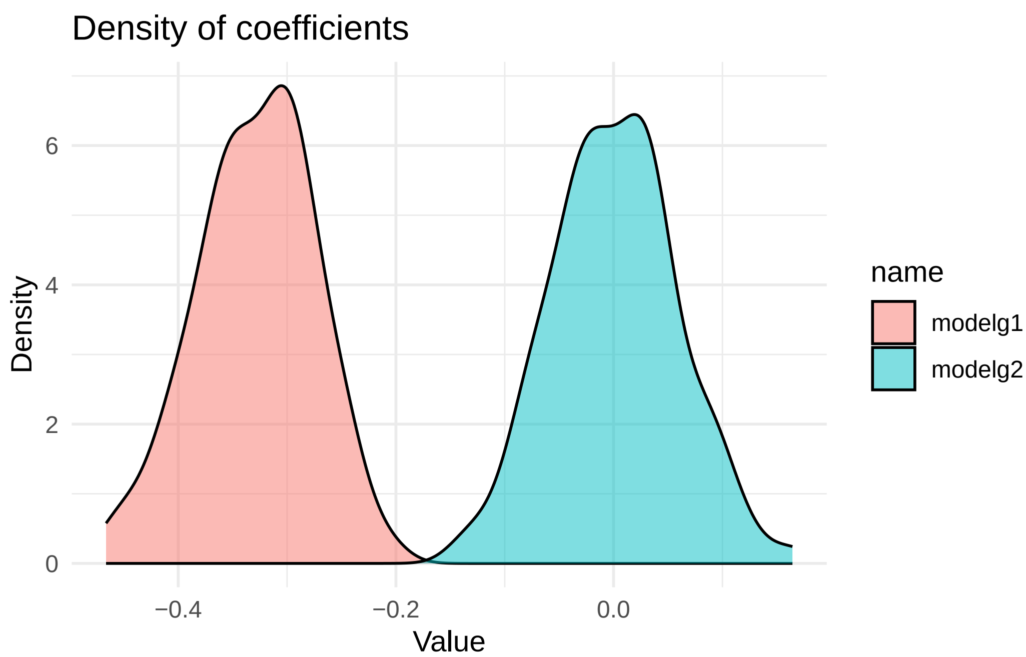
We however know more than the researcher. So we can show that (denoting earnings as E and capital gains as C):

$$\begin{aligned}
\gamma_1 = \frac{Cov(I, Y)}{Var(I)} &= \frac{Cov(I, E - C + e)}{Var(I)} \\
&= \frac{Cov(I, E)}{Var(I)} - \frac{Cov(I, C)}{Var(I)}
\end{aligned}$$

This means that γ_1 from the short regression is a sum of two covariances (scaled by the variance). Also note, that both nominators (and of course the denominator) are positive.

(g)

```r
## Let's run the simulation a few times and store the coefficients for both models
n_sims <- 100
coefs <- tibble(modelg1 = numeric(), modelg2 = numeric())

for (i in 1:n_sims){
  df <- run_simulation(500)
  modelg1 <- lm(child_oc ~ income + occ_st, data = df)
  modelg2 <- lm(child_oc ~ income, data = df)

  coefs <- coefs |>
    add_row(
      modelg1 = coef(modelg1)[2],
      modelg2 = coef(modelg2)[2]
    )
}

## Let's plot both

coefs |>
  pivot_longer(cols = everything()) |>
  ggplot() +
  geom_density(aes(x = value, fill = name), alpha = 0.5) +
  labs(title = "Density of coefficients", x = "Value", y = "Density") +
  theme_minimal()
```



Inclusion of Occupational status has a large effect. Where is this effect coming from? The researcher not knowing the true state of the world, will likely assume there is OVB and that the longer regression is "more correct" (conveniently also more significant). We, however, can show what is really going on. Keep in mind that our artificial DGP-world here is very simple. Let's have a look at what the coefficient on income really is.

Lets residualise I by regressing it on a constant and Occupational status (O) to obtain a new $\tilde{I}$ (let's denote the coefficients in this regression δ):

To clarify the different models let's write them all down again:

$$\begin{aligned}
Y &= E - C + \varepsilon, && \text{True DGP} \\
I &= \delta_0 + O\delta_1 + \tilde{I}, && \text{Residualisation} \\
Y &= \alpha + \beta_1 I + \beta_2 O + e, && \text{Model we estimate}
\end{aligned}$$

Now, let's use the regression anatomy formula (if that does not ring a bell, have a look at Mostly Harmless section 3.1.2):

$$
\begin{aligned}
\beta_1 &= \frac{Cov(\tilde{I}, Y)}{Var(\tilde{I})} \\
&= \frac{Cov(\tilde{I}, E - C)}{Var(\tilde{I})} \\
&= \frac{Cov(I - \delta_0 - O\delta_1, E - C)}{Var(\tilde{I})} \\
&= \frac{Cov(I, E - C)}{Var(\tilde{I})} - \delta_1 \frac{Cov(O, E - C)}{Var(\tilde{I})} \\
&= \frac{Cov(E + C, E - C)}{Var(\tilde{I})} - \delta_1 \frac{Cov(E + u, E - C)}{Var(\tilde{I})} \\
&= \frac{Var(E) - Var(C)}{Var(\tilde{I})} - \delta_1 \frac{Var(E)}{Var(\tilde{I})} \\
Var(\tilde{I}) &= Var(I) + \delta_1^2 Var(O) - 2\delta_1 Cov(I, O) \\
&= Var(E) + Var(C) + \delta_1^2 \left(Var(E) + Var(u)\right) \\
&\quad - 2\delta_1 Cov(E + C, E + u) \\
&= Var(E) + Var(C) + \delta_1^2 \left(Var(E) + Var(u)\right) - 2\delta_1 Var(E) \\
&= Var(E)(1 - \delta_1)^2 + Var(C) + Var(u)\delta_1^2 \\
&= \frac{Var(E)(1 - \delta_1) - Var(C)}{Var(E)(1 - \delta_1)^2 + Var(C) + Var(u)\delta_1^2} \\
&= \frac{1(1 - \delta_1) - 1}{1(1 - \delta_1)^2 + 1 + 1\delta_1^2} = -\frac{1}{3} \\
&\text{bc. } \delta_1 = \frac{1}{2}
\end{aligned}
$$

There is no need to do derive this yoursel. It just shows you how complicated the relationships can get from even a very simple DGP. In words what happened is that the positive correlation of Earnings with Occupational status has absorbed some of the effect of income. The coefficient on income is now more pulled towards the effect of Capital gains. So is it “wrong” to control for Occupational status? Not really. Similarly, to just running the regression with income, it just becomes very hard to interpret what is actually going on - i.e. in real life we won’t know what all these variances and relationships are. The potential outcomes framework can help clarifying some of this.

End