The general linear model – Lecture 5
Multiple predictors

Jan Vanhove
`https://janhove.github.io`

Ghent, 12–14 July 2023

As we've seen in the previous lectures, the general linear model can accommodate multiple predictor variables at once, yielding **multiple regression models** as opposed to simple regression models. Mathematically and computationally, nothing really changes when adding more predictor variables to the model. Conceptually, however, multiple regression models often poses researchers a number of challenges. These mainly relate to the decision whether to include several predictor variables in the model and, relatedly, to how to interpret the estimated regression coefficients. Here, I don't use the verb 'interpret' to refer to subject-matter interpretations, but to more basic statistical interpretations: What do all these numbers literally mean? Clearly, before we can lend subject-matter interpretations to the output of statistical models, we need to understand what they mean literally.

In this lecture, we'll write simulations to illustrate the consequences of including several predictors in a handful of key scenarios.

> **Causal relationships.** The parameter estimates that the general linear model produces first and foremost describe associations in the data. Often, however, researchers want to lend a causal interpretation to these associations. For instance, we're usually not content with finding out that the experimental group outperforms the control group on average – we want to know whether the experimental group outperforms the control group on average *because* they're the experimental group.
>
> **Statistical tools cannot prove claims about causality.** This goes even for so-called causal models. But what we can do is make assumptions about the causal connections between our variables and reason about whether and how we can model these statistically. Whether these causal assumptions are reasonable is a subject-matter issue, not a statistical one.

A useful tool when discussing assumptions about causal connections are **directed acyclic graphs** (DAGs). For an introduction to DAGs, see Rohrer (2018) and McElreath (2020). We will use DAGs throughout this lecture to represent causal connections between variables ($x, y, z, \dots$). Throughout, we're interested in estimating the causal influence that x exerts on y. Our guiding questions are, first, whether this is at all possible, and, second, if so, which variables we should include in the model.

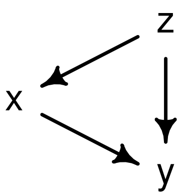

Figure 1: In this scenario, z influences both x and y. As a result, z acts as a confounding variable if we're interested in the causal influence of x on y.

1 Taking into account confounding variables

First, we consider the scenario shown in Figure 1. We're interested in the causal influence of x on y. However, it is possible that a third variable, z, influences both x and y. If you want to, you can replace these abstract variable names by something more specific (e.g., $x =$ participation in a content- and language-integrated programme, $y =$ proficiency in the target language, $z =$ the parents' socio-economic status). But I think it's ultimately more useful to get used to the abstraction, as this makes it clearer that the lessons drawn from this scenario are valid more generally.

Nevertheless, we'll make this scenario a bit more concrete by fleshing out what the causal connections between the three variables are. To keep this simple, we'll assume that the z variable was drawn from a normal distribution with mean 0 and standard deviation 1; but this isn't too important:

$$z_i \sim \text{Normal}(0, 1^2).$$

Next, we assume that a one-unit increase in the z variable causes an increase of 1.2 units in the x variable. There is, however, some variability in the x variable that is unrelated to z. We express this additional variability in an error term τ, which we assume is also drawn from a normal distribution with mean 0 and standard deviation 1:

$$\begin{aligned} x_i &= 0 + 1.2 \cdot z_i + \tau_i, \qquad (1) \\ \tau_i &\sim \text{Normal}(0, 1^2). \end{aligned}$$

The numbers 1.2 in the first line and 1 in the second line were chosen arbitrarily – you can play around with these numbers at home. The 0 in the first line merely means that the mean of the x variable is 0, but little hinges on this.

Additionally, we assume that the y variable is described by the equation below. The influence of x on y is such that a one-unit increase in x leads to a 0.6-unit increase in y. The z variable, however, has a negative causal effect on y, but we don't want to estimate this effect. Again, the numbers 5.2, 0.6 and -1.3 were chosen arbitrarily.

$$\begin{aligned} y_i &= 5.2 + 0.6 \cdot x_i - 1.3 \cdot z_i + \varepsilon_i, \qquad (2) \\ \varepsilon_i &\sim \text{Normal}(0, 1^2). \end{aligned}$$

Let's now simulate a dataset with 100 observations of these three variables. If we don't specify any further parameters, the `rnorm(n)` call generates n observations from a normal distribution with mean 0 and standard deviation 1:

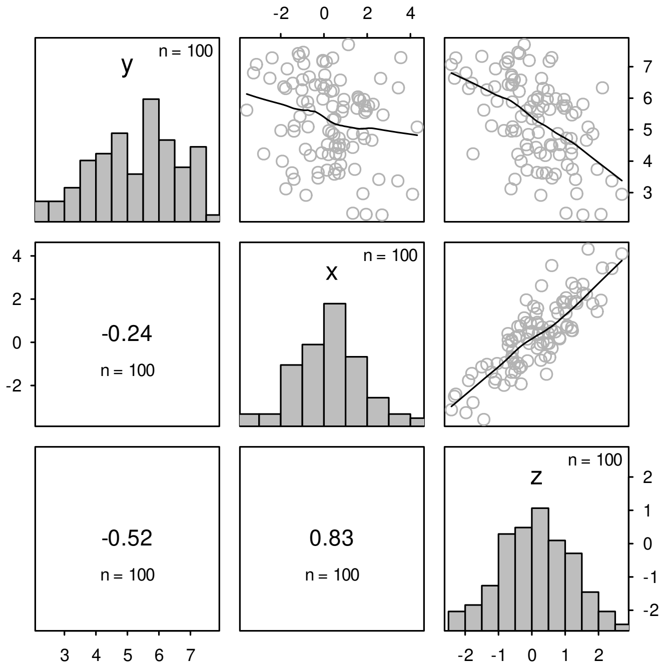

Figure 2: A scatterplot matrix of the three simulated variables. The numbers in the lower triangle are Pearson correlation coefficients and the number of data pairs they are based on. The trend lines in the scatterplots in the upper triangle are scatterplot smoothers. For more information, see `https://janhove.github.io/reporting/2019/11/28/scatterplot-matrix`.

```
> library(tidyverse)
> library(here)
>
> n <- 100
> z <- rnorm(n)
> x <- 0 + 1.2*z + rnorm(n)
> y <- 5.2 + 0.6*x - 1.3*z + rnorm(n)
> d <- tibble(y, x, z)
```

In order to visualise the pairwise associations between several continuous variables at once, we can draw a scatterplot matrix. The file `scatterplot_matrix.R` in the `functions` directory defines a custom-made function for plotting scatterplot matrices. Let's read it in and plot the simulated data; see Figure 2.

```
> source(here("functions", "scatterplot_matrix.R"))
> scatterplot_matrix(d)
```

We will now compare two strategies for analysing these simulated data. If we were analysing real data, we wouldn't do this – we'd only run the analysis that makes most sense. But in this lecture, we want to use simulated data in order to find out what the most sensible strategy is.

In the first model, we ignore the z variable:

```
> dag1.lm1 <- lm(y ~ x, data = d)
> summary(dag1.lm1)$coefficients

             Estimate Std. Error t value   Pr(>|t|)
(Intercept)  5.34505   0.133487 40.0418 1.5104e-62
x           -0.19641   0.081777 -2.4018 1.8200e-02
```

The resulting parameter estimates are to be interpreted in exactly the same way as explained in Lecture 2:

- If we were to take a large number of observations for which all x values were 0, then, according to the model, we'd expect the mean of their y values to be 5.35 ± 0.13.
- If we were to take a large number of observations for which all x values were 1, then, according to the model, their mean y value is expected to be 0.20 ± 0.08 lower than that of the $x = 0$ group.

This interpretation is absolutely fine! Note, however, that this interpretation does not involve any causal claims.

In the second model, we include z as a predictor:

```
> dag1.lm2 <- lm(y ~ x + z, data = d)
> summary(dag1.lm2)$coefficients

             Estimate Std. Error t value   Pr(>|t|)
(Intercept)  5.29515    0.10815 48.9614 3.3341e-70
x            0.52293    0.11897  4.3956 2.8281e-05
z           -1.30970    0.18007 -7.2733 9.0728e-11
```

These parameter estimates, too, are to be interpreted as outlined in Lecture 2:

- If we were to take a large number of observations for which all x *and* z values were 0, then, according to the model, we'd expect the mean of their y values to be 5.30 ± 0.11.
- If we were to take a large number of observations for which all x values were 1 and all z values were 0, then, according to the model, their mean y value is expected to be 0.52 ± 0.12 higher than that of the $x = 0, z = 0$ group.
- If we were to take a large number of observations for which all x values were 0 and all z values were 1, then, according to the model, their mean y value is expected to be 1.3 ± 0.2 lower than that of the $x = 0, z = 0$ group.

The second point does *not* contradict the interpretation of the parameter estimates of the first model: Just because parameters with the same name occur in both models (`(Intercept)`, `x`) doesn't mean that these have the same same interpretation. Specifically, the parameter estimates in the second model can only be interpreted correctly when taking into account the other variable in the model, z. Similarly, to interpret the parameter estimates in the first model, you must not implicitly assume a fixed value for the z variable that was not included in the model.

Now compare the parameter estimates of the second models with those in Equation 2 on page 2. The estimated parameters are quite close to the true parameters we used to generate the simulated data. In fact, the discrepancies between the estimates and the true values are purely due to chance. We can verify this by simulating lots of datasets using the same parameters as used in Equations 1 and 2 and analysing them in the same way as we did here. Below, we define the function `generate_dag1()`, which by default generates 10,000 such datasets, containing 100 observations each. Each dataset is analysed twice: Once without taking the *z* variable into account, and once including this variable in the model. For each simulated dataset and each model, the estimated `x` parameter is extracted.

```
> generate_dag1 <- function(
+   n = 100,           # number of observations per dataset
+   sims = 10000,      # number of datasets
+   z_x = 1.2,         # influence z -> x
+   x_y = 0.6,         # influence x -> y
+   z_y = -1.3,        # influence z -> y
+   baseline_y = 5.2   # baseline y
+ ) {
+   est_lm1 <- vector(length = sims)
+   est_lm2 <- vector(length = sims)

+   for (i in 1:sims) {
+     z <- rnorm(n)
+     x <- z_x*z + rnorm(n)
+     y <- baseline_y + x_y*x + z_y*z + rnorm(n)
+     mod.lm1 <- lm(y ~ x)
+     mod.lm2 <- lm(y ~ x + z)
+     est_lm1[[i]] <- coef(mod.lm1)[[2]]
+     est_lm2[[i]] <- coef(mod.lm2)[[2]]
+   }

+   return(tibble(`Model 1` = est_lm1,
+                 `Model 2` = est_lm2))
+ }
```

The code below runs this function using the default settings and then plots the estimated parameters obtained by both models; Figure 3.

```
> est_dag1 <- generate_dag1()
>
> est_dag1 |>
+   pivot_longer(cols = everything(),
+                names_to = "Model",
+                values_to = "Estimate") |>
+   ggplot(aes(x = Estimate)) +
+   geom_histogram(bins = 50,
+                  fill = "grey", colour = "black") +
+   geom_vline(xintercept = 0.6, linetype = "dashed") +
+   facet_grid(cols = vars(Model)) +
+   xlab("Estimate x → y") +
```

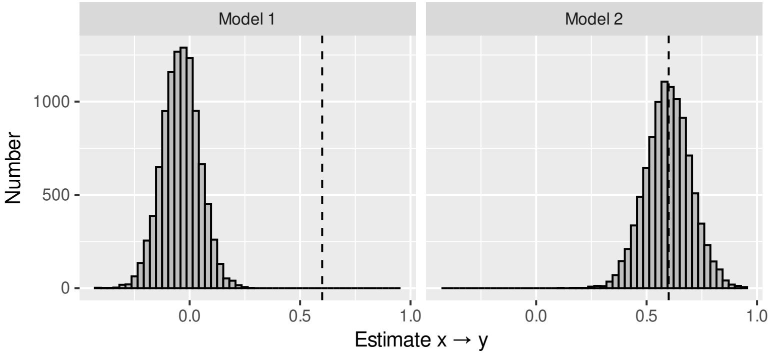

Figure 3: Model 1 does not estimate the causal effect of x on y in an unbiased way. Depending on the parameter values chosen in Equations 1 and 2, the bias could be an overestimate or, like here, an underestimate. Model 2, by contrast, does provide an unbiased estimate of the effect of x on y: On average, the estimated parameter values correspond exactly to the true parameter value.

```
+   ylab("Number")
```

This simulation confirms that the second model yields an unbiased estimate of the causal effect of x on y (0.6), but the first doesn't:

```
> apply(est_dag1, 2, mean)

  Model 1   Model 2
-0.040036  0.597803
```

The model without the confounding variable (z) isn't wrong. If you want to estimate the average difference in the y variable between groups differing in x, this is the model you need. But if we assume the causal structure shown in Figure 1, we cannot interpret its parameter estimates causally.

It seems clear what conclusion we ought to draw from the considerations above: If confounding variables are at play and you want to make causal claims, you need to include these confounders in the analysis. In practice, however, things aren't so simple.

- We may not know all confounders or we may not have been able to measure all of them.
- Even if we did measure the confouders, we probably didn't measure them perfectly.
- It is possible that the confounders exert some nonlinear influence, whereas we only considered linear effects.

In the exercises below, you explore the first two complicating factors. To anticipate the take-home message:

Don't pin your hopes on statistical tools to neutralise the effect of confounding variables. This would require you to have perfectly measured all confounders and to have

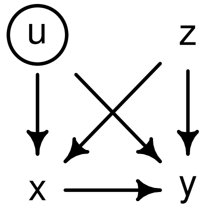

Figure 4: The confounder z was measured; the confounder u wasn't.

specified the functional form of their causal effects correctly. **Statistical tools are no substitute for a solid research design that neutralises confounders.**

Exercise 1: Unknown confounders. In Figure 4, one confounder, u, was added to our DAG, but for some reason, it wasn't measured. We assume that the following causal relationships are at play:

$$\begin{aligned}
z_i &\sim \text{Normal}(0, 1^2), \\
u_i &\sim \text{Normal}(0, 1^2), \\
x_i &= 0 + 1.2 \cdot z_i + 0.9 \cdot u_i + \tau_i, \\
y_i &= 5.2 + 0.6 \cdot x_i - 1.3 \cdot z_i + 2.5 \cdot u_i + \varepsilon_i, \\
\tau_i &\sim \text{Normal}(0, 1^2), \\
\varepsilon_i &\sim \text{Normal}(0, 1^2).
\end{aligned}$$

We can generate a dataset governed by these equations that contains 50 observations as follows. Note while we simulate the u variable, we don't add it to the dataset, since it wasn't measured.

```
> n <- 50
> z <- rnorm(n)
> u <- rnorm(n)
> x <- 0 + 1.2*z + 0.9*u + rnorm(n)
> y <- 5.2 + 0.6*x - 1.3*z + 2.5*u + rnorm(n)
> d <- tibble(y, x, z)
```

While we cannot include u in the model, we can include z:

```
> exercise1.lm <- lm(y ~ x + z, data = d)
> summary(exercise1.lm)$coefficients

            Estimate Std. Error t value   Pr(>|t|)
(Intercept)   5.6079    0.27599 20.3191 6.4343e-25
x             1.6582    0.18338  9.0423 7.4386e-12
z            -2.5881    0.32252 -8.0246 2.3408e-10
```

1. Explain what the parameter estimate for `x` literally means.
2. Adapt the `generate_dag1()` function and show that the `exercise1.lm` model does not

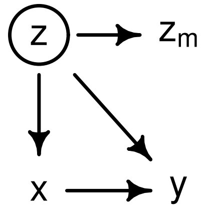

Figure 5: The z variable confounds the causal relationship between x and y, but it wasn't measured directly. Instead, we need to make do with a proxy variable z_m that captures z imperfectly.

provide an unbiased estimate of the causal effect of x on y, even though it includes the measured confounder z.

3. Would more data help solve this problem? Justify your answer by means of a simulation in which each dataset features 200 instead of 50 observations.

Exercise 2: Measurement error on the confounder. Figure 5 depicts another scenario where z confounds the causal relationship between x and y. This time, however, we didn't measure z itself. Instead, we obtained an **indicator** z_m of z. This indicator represents the **construct** z imperfectly. This is quite usual: constructs such as working memory capacity, L2 writing skills, L1 vocabulary knowledge, intelligence, socioeconomic status etc., cannot be observed directly and have to be inferred from test results, questionnaire responses etc. It is therefore crucial to understand the effect of measurement error on statistical control.

We assume that the same causal links are at play as earlier. The only difference is that we include z_m instead of z in the dataset. To construct z_m, we take the values of z and add some Gaussian noise to it (from a normal distribution with mean 0 and standard deviation 0.3).

$$\begin{aligned}
z_i &\sim \text{Normal}(0, 1^2), \\
z_{m,i} &= z_i + \psi_i, \\
x_i &= 0 + 1.2 \cdot z_i + \tau_i, \\
y_i &= 5.2 + 0.6 \cdot x_i - 1.3 \cdot z_i + \varepsilon_i, \\
\psi_i &\sim \text{Normal}(0, 0.3^2), \\
\tau_i &\sim \text{Normal}(0, 1^2), \\
\varepsilon_i &\sim \text{Normal}(0, 1^2).
\end{aligned} \tag{3}$$

Let's simulate the data.

```
> n <- 50
> z <- rnorm(n)
> z_m <- z + rnorm(n, sd = 0.3)
> x <- 0 + 1.2*z + rnorm(n)
> y <- 5.2 + 0.6*x - 1.3*z + rnorm(n)
> d <- tibble(y, x, z_m)
```

This time, we include z_m in the analysis:

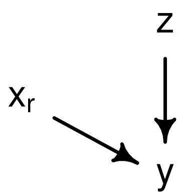

Figure 6: In this scenario, the values of x were assigned randomly and independenty of z. This is typical of experiments in which the participants are randomly assigned to the conditions.

```
> exercise2.lm <- lm(y ~ x + z_m, data = d)
> summary(exercise2.lm)$coefficients

            Estimate Std. Error t value   Pr(>|t|)
(Intercept)  5.24215    0.14902 35.1769 2.0466e-35
x            0.59012    0.15853  3.7224 5.2697e-04
z_m         -1.33034    0.21562 -6.1697 1.4891e-07
```

1. Explain what the parameter estimate for `x` literally means.
2. Adapt the `generate_dag1()` function and show that the `exercise2.lm` model does not provide an unbiased estimate of the causal effect of x on y, even though it includes an indicator of the confounder (i.e., z_m).
3. Would more data help solve this problem? Justify your answer by means of a simulation in which each dataset features 200 instead of 50 observations.
4. Would it be better not to take into account the confounder at all? That is, should we just fit the model without the indicator of z?
5. What would happen if we had a more reliable indicator for z? To answer this question, rerun your simulation but use a standard deviation of 0.1 instead of 0.3 for ψ in Equation 3.

2 Control variables in randomised experiments

In Section 1, we discussed a scenario in which z causally affects both x and y. This scenario is typical for observational (or correlational) studies and quasi-experiments. In the present section, we turn our attention to randomised experiments, that is, experiments in which the values of x are manipulated by the researchers based on random assignment. To reflect this, the x variable is represented as x_r in Figure 6 (r for *randomised*). A typical example for the more abstract scenario we consider here are experiments in which participants are randomly assigned to the experiments' conditions. In this case, x would be a categorical variable. But without any loss of generality, we can restrict our discussion to a continuous x. This makes the simulations a bit easier; it doesn't affect the conclusions we will draw. The z variable would then be a variable that we're not really interested it but of which we suspect that it correlates with the outcome, y. The textbook case is a pretest/posttest experiment, where x represents the experimental condition, z the pretest performance, and y the posttest performance.

We assume that the causal links between the variables are described by the following equations:

$$\begin{aligned}
x_i &\sim \text{Normal}(0, 1^2), \\
z_i &\sim \text{Normal}(0, 1^2), \\
y_i &= 5.2 + 0.3 \cdot x_i + 0.9 \cdot z_i + \varepsilon_i, \qquad (4) \\
\varepsilon_i &\sim \text{Normal}(0, 1^2).
\end{aligned}$$

Let's simulate a dataset conforming to these equations:

```
> n <- 100
> x <- rnorm(n)
> z <- rnorm(n)
> y <- 5.2 + 0.3*x + 0.9*z + rnorm(n)
> d <- tibble(y, x, z)
```

```
> # Not shown in script
> scatterplot_matrix(d)
```

We again fit one model with and one model without z. Both models yield similar though different estimates for the `x` parameter: 0.35 ± 0.14 and 0.38 ± 0.10.

```
> dag2.lm1 <- lm(y ~ x, data = d)
> summary(dag2.lm1)$coefficients

            Estimate Std. Error t value   Pr(>|t|)
(Intercept)  5.40030    0.14198 38.0370 1.7153e-60
x            0.35473    0.14035  2.5275 1.3086e-02

> dag2.lm2 <- lm(y ~ x + z, data = d)
> summary(dag2.lm2)$coefficients

            Estimate Std. Error t value   Pr(>|t|)
(Intercept)  5.36358   0.097475 55.0254 5.9391e-75
x            0.38051   0.096327  3.9502 1.4813e-04
z            0.98832   0.093735 10.5438 9.0128e-18
```

The simulations below will confirm that *both* models yield unbiased estimates of the causal influence of x on y. By this standard, neither model is bad. That said, if we have the z variable at our disposal, the second model is to be preferred. The reason for this will become clearer once we've simulated a couple of thousand datasets. To this end, we could write a new function, `generate_dag2()`, but we can also just use `generate_dag1()` and set the parameter value of `z_x` to 0:

```
> est_dag2 <- generate_dag1(x_y = 0.3, z_y = 0.9, z_x = 0)
```

As shown in Figure 7, both models yield unbiased estimates of the causal influence of x on y in Equation 4. But these estimates vary less from sample to sample in model 2, that is, on average, they are closer to the true parameter value than the estimates that model 1 yields.

```
> est_dag2 |>
+   pivot_longer(cols = everything(),
+                names_to = "Model",
+                values_to = "Estimate") |>
```

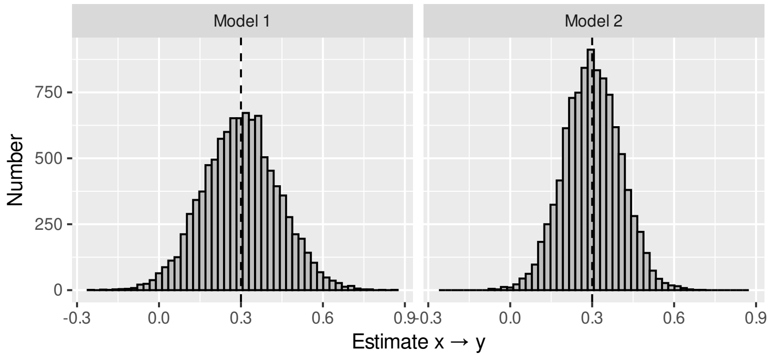

Figure 7: Both models yield unbiased estimates of the causal influence of x on y (0.3). But these estimates vary less from sample to sample in the second model than in the first model. In other words, on average, the second model yields more precise estimates.

```
+   ggplot(aes(x = Estimate)) +
+   geom_histogram(bins = 50,
+                  fill = "grey", colour = "black") +
+   geom_vline(xintercept = 0.3, linetype = "dashed") +
+   facet_grid(cols = vars(Model)) +
+   xlab("Estimate x → y") +
+   ylab("Number")
```

We can check this numerically. The means of the estimates in both models corresponds to the true value. But compared to the estimates by the first model, the standard deviation of the estimates by the second model is about 25% lower.

```
> apply(est_dag2, 2, mean)

Model 1 Model 2
0.30060 0.30018

> apply(est_dag2, 2, sd)

Model 1 Model 2
0.13736 0.10291
```

What's happened here is that by including the z variable in the model – even though it doesn't act as a confounder for the causal link between x and y – we have reduced the error variance and have hence increased the precision of our estimate. So even if you don't actually care about z, it can pay dividends to still collect it and include it in your analysis.

Don't go overboard with this, though. One or two additional variables that you strongly suspect to be correlated with the outcome may be useful, particularly if they're not too strongly related to each other. But don't collect umpteen additional variables on the off-chance that they might be related to the outcome and help you reduce the error variance. Moreover, the decision whether or not to include some additional variable in your analysis should be taken *before* you analyse your data – don't run multiple models and then report the one that works 'best'.

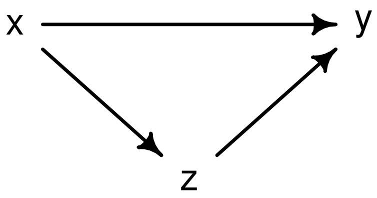

Figure 8

3 Posttreatment variables

We now briefly turn our attention to scenarios where z is a posttreatment variable. In the context of a randomised experiment with x as the predictor of interest, this means that z was collected *after* the randomisation. As a consequence, z may be influenced by x.

Mediators. First consider the DAG in Figure 8. Notice that there are two causal paths from x to y: one is direct ($x \to y$), one is **mediated** by z ($x \to z \to y$). A dataset conforming to this DAG can be simulated as follows:

```
> n <- 200
> x <- rnorm(n)
> z <- 0.4*x + rnorm(n)
> y <- 0.6*x + 1.2*z + rnorm(n)
> d <- tibble(x, y, z)
```

What's the causal influence of x on y? That is, if you increase x by one unit, what change in y does this cause?

The answer to this question is *not* 0.6 units. While the direct causal effect of x on y is indeed such that a one-unit increase results in a 0.6-unit increase in y, x also influences y via z. A one-unit increase in x results in a 0.4-unit increase in z; and a one-unit increase in z results in a 1.2-unit increase in y. So in addition to the 0.6-unit increase that a one-unit increase in x causes directly in y, it also results in a $0.4 \cdot 1.2 = 0.48$-unit increase via z, for a total causal effect of $0.6 + 0.48 = 1.08$ units increase in y per one-unit increase in x!

If we want to estimate the total causal influence of x on y, we shouldn't close any causal paths going from x to y by controlling for z, that is, we should fit a simple regression model that does *not* include z. If, however, we want to estimate the causal effect of x on y that is not mediated by z, then we *do* need to control for z. Which model you want to run depends entirely on what you want to estimate.

Colliders. Now consider the DAG in Figure 9. Here, both x and y causally affect z ($x \to z \leftarrow y$). A variable in which two or more causal variables clash together is known as a **collider**. In a sense, colliders are the opposite of confounders: As long as colliders are *not* controlled for, their presence does not bias the causal estimates of interest. But once they are controlled for, they may bias these causal estimates.

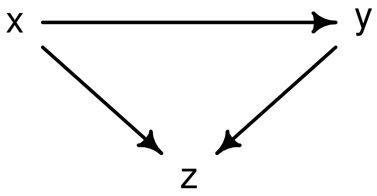

Figure 9

By way of example, let's assume the data are generated following these equations:

$$
\begin{aligned}
x_i &\sim \text{Normal}(0, 1^2), \\
y_i &= 0.4x_i + \varepsilon_i, \\
z_i &= 0.5x_i + 0.8y_i + \tau_i, \\
\varepsilon_i &\sim \text{Normal}(0, 1^2), \\
\tau_i &\sim \text{Normal}(0, 1^2).
\end{aligned}
$$

The function `generate_posttreatment()` is a slight adaptation of `generate_dag1()`.

```
> generate_posttreatment <- function(
+   n = 100,      # number of observations per dataset
+   sims = 10000, # number of datasets
+   x_y = 0.4,    # influence x -> y
+   x_z = 0.5,    # influence x -> z
+   y_z = 0.8     # influence y -> z
+ ) {
+   est_lm1 <- vector(length = sims)
+   est_lm2 <- vector(length = sims)

+   for (i in 1:sims) {
+     x <- rnorm(n)
+     y <- x_y*x + rnorm(n)
+     z <- x_z*x + y_z*y + rnorm(n)
+     mod.lm1 <- lm(y ~ x)
+     mod.lm2 <- lm(y ~ x + z)
+     est_lm1[[i]] <- coef(mod.lm1)[[2]]
+     est_lm2[[i]] <- coef(mod.lm2)[[2]]
+   }

+   return(tibble(`Model 1` = est_lm1,
+                 `Model 2` = est_lm2))
+ }
```

As this simulation shows, the model without the collider (z) is able to estimate the causal effect of x on y (0.4) in an unbiased way. The estimates for the model with the collider, by contrast, are biased.

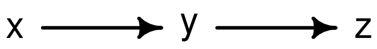

Figure 10

```
> est_collider <- generate_posttreatment()
> apply(est_collider, 2, mean)

   Model 1    Model 2
0.39975704 0.00059706
```

Note again, though, that the model with the collider is well-suited if you want to estimate the mean difference in the y variable between two groups that differ in the x variable but whose z values are all the same. As always, whether the model is justified depends on what you want to find out and on your assumptions.

It also bears pointing out that colliders are sometimes inadvertently controlled for during the design stage. See Rohrer (2018) for more on this important topic.

Miscellaneous. In the DAG in Figure 10, z is also a posttreatment variable. This time, however, it is only indirectly influenced by x. In order to check whether it would be best to include z as a variable in the model, let's assume the following equations describe the causal links between the variables:

$$\begin{aligned}
x_i &\sim \text{Normal}(0, 1^2), \\
y_i &= 0.4x_i + \varepsilon_i, \\
z_i &= y_i + \tau_i, \\
\varepsilon_i &\sim \text{Normal}(0, 1^2), \\
\tau_i &\sim \text{Normal}(0, 1^2).
\end{aligned}$$

We can reuse the `generate_posttreatment()` function and just specify the new parameter values. Note that, again, the model without the additional variable provides an unbiased estimate of the causal effect of x on y, whereas the model with this additional variable doesn't.

```
> est_proxy_y <- generate_posttreatment(x_y = 0.4, x_z = 0, y_z = 1)
> apply(est_proxy_y, 2, mean)

Model 1 Model 2
0.39863 0.19938
```

In the scenario depicted in Figure 10, including the additional variable will bias the estimate of interest towards zero. By spelling out what the literal meaning is of the estimated parameter for `x` in the second model, you should see why this is the case.

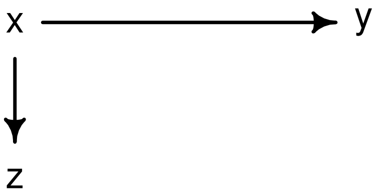

Figure 11

The situation is only slightly different in the scenario depicted in Figure 11. Here, z is directly affected by x, but not by y. We'll simulate datasets conforming to the following equations:

$$\begin{aligned}
x_i &\sim \text{Normal}(0, 1^2),\\
y_i &= 0.4x_i + \varepsilon_i,\\
z_i &= x_i + \tau_i,\\
\varepsilon_i &\sim \text{Normal}(0, 1^2),\\
\tau_i &\sim \text{Normal}(0, 1^2).
\end{aligned}$$

As the simulation results show, both models yield unbiased estimates of the causal effect of x on y. But notice that the estimates vary more from sample to sample for the second model than for the first. So a given estimate resulting from the model without the additional variable is more likely to be closer the actual parameter value than a given estimate resulting from the model with the additional variable. It's hard to imagine a research question where the second model would be preferred in this scenario.

```
> est_proxy_x <- generate_posttreatment(x_y = 0.4, x_z = 1, y_z = 0)
> apply(est_proxy_x, 2, mean)

Model 1 Model 2
0.39975 0.40048

> apply(est_proxy_x, 2, sd)

Model 1 Model 2
0.10225 0.14665
```

In sum, including posttreatment variables as predictors in the analysis is usually a bad idea. In randomised experiments, there luckily exists a simple trick for preventing that the predictors you include in the model are posttreatment variables: Just collect the predictors before carrying out the intervention!

Summary and recommendations

- A multiple regression model is not just multiple simple regressions run at once. The meaning of the parameter estimates changes if you add or remove predictors to or from the model. Also see Morrissey & Ruxton (2018) and Vanhove (2021).
- The decision which predictors to include in a model depends on what it is exactly you want to estimate and how you think different variables may be causally related. In

my experience, people's difficulties with regression models aren't so much statistical in nature as due to their not having worked out what they actually want to find out.

- Know what the literal meaning of the estimated model parameters is before you interpret them in terms of the subject-matter.
- As the exercises will show, plots of the actual data may help prevent both you and your readership from interpreting the model output incorrectly.

Further reading

On the limits of statistical control in observational studies, see Christenfeld et al. (2004), Huitema (2011, Part VII) and Westfall & Yarkoni (2016). On the utility of statistical control in randomised experiments, see Vanhove (2015) and references therein. For an accessible introduction to DAGs, confounders and colliders, see Rohrer (2018).

Exercises

1. Vanhove et al. (2019) had 1,000 short French texts written by children rated for their lexical diversity on a 9-point scale by between 2 and 18 raters each. For the purposes of this exercise, we're interested in modelling these human ratings in terms of the length of the text (the logarithmically transformed number of tokens (using base 2), `log2_ntokens`) and the type–token ratio (TTR), an easily computed metric of a text's lexical diversity.

```
> d <- read_csv(here("data", "text_ratings.csv"))
> lexdiv.lm <- lm(mean_rating ~ log2_ntokens + TTR, data = d)
> summary(lexdiv.lm)$coefficients

             Estimate Std. Error t value     Pr(>|t|)
(Intercept)   -4.0928   0.458028 -8.9357   1.9186e-18
log2_ntokens   1.2152   0.047348 25.6655 6.1198e-112
TTR            3.8364   0.330558 11.6059   2.6707e-29
```

   (a) (Paper and pencil.) For each claim, decide whether it is correct. Justify your answers.

   - The model output shows that human raters are sensitive to differences in the type–token ratio when ratings texts for their lexical diversity.
   - The model output shows that there is a positive linear relationship between the TTR values and the mean ratings: Texts with higher TTR values tend to receive higher ratings than texts with lower TTR values.

   (b) (With R.) Draw a scatterplot matrix of these variables and revise your answer to the previous question if needed.

   (c) (Paper and pencil.) Explain what each of the three estimated model parameters literally means.

   (d) (Paper and pencil.) According to this model, what mean rating would you expect for a text consisting of 64 tokens with a TTR of 0.7?

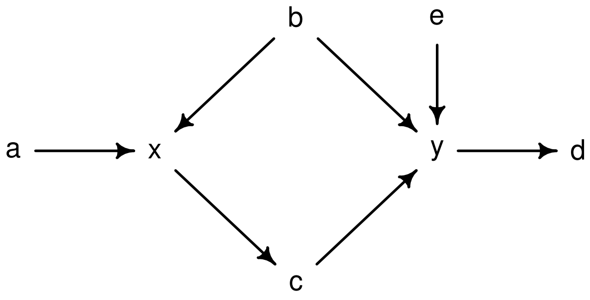

Figure 12

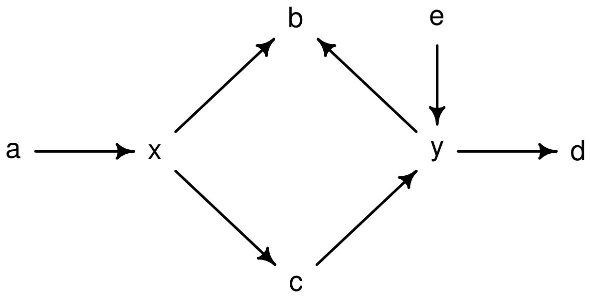

Figure 13

2. (Paper and pencil.) Consider Figure 12. Assume that you want to obtain an unbiased and maximally precise estimate the total causal effect of x on y and that all relationships shown are linear and additive. Which variables would you include in a general linear model as predictors? Justify your answer.

3. (Paper and pencil.) Same question for Figure 13.

4. (Paper and pencil.) Let's say that instead of merely observing and measuring all the variables in Figure 12, we devise a randomised experiments in which we randomly assign the participants to values of x.

   (a) Draw the updated DAG.

   (b) Which predictors would you in the general linear model now? Justify your answer.

References

Christenfeld, Nicholas J. S., Richard P. Sloan, Douglas Carroll & Sander Greenland. 2004. Risk factors, confounding, and the illusion of statistical control. *Psychosomatic Medicine* 66. 868–875. doi:10.1097/01.psy.0000140008.70959.41.

Huitema, Bradley E. 2011. *The analysis of covariance and alternatives: Statistical methods for experiments, quasi-experiments, and single-case studies*. Hoboken, NJ: Wiley.

McElreath, Richard. 2020. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Boca Raton, FL: CRC Press 2nd edn.

Morrissey, Michael B. & Greame D. Ruxton. 2018. Multiple regression is not multiple regressions: The meaning of multiple regression and the non-problem of collinearity. *Philosophy, Theory, and Practice in Biology* 10(3). doi:10.3998/ptpbio.16039257.0010.003.

Rohrer, Julia M. 2018. Thinking clearly about correlations and causation: Graphical causal models for observational data. *Advances in Methods and Practices in Psychological Science* 1(1). 27–42. doi:10.1177/2515245917745629.

Vanhove, Jan. 2015. Analyzing randomized controlled interventions: Three notes for applied linguists. *Studies in Second Language Learning and Teaching* 5. 135–152. doi:10.14746/ssllt.2015.5.1.7.

Vanhove, Jan. 2021. Collinearity isn't a disease that needs curing. *Meta-Psychology* 5. doi:10.15626/MP.2021.2548.

Vanhove, Jan, Audrey Bonvin, Amelia Lambelet & Raphael Berthele. 2019. Predicting perceptions of the lexical richness of short French, German, and Portuguese texts using text-based indices. *Journal of Writing Research* 10(3). 499–525. doi:10.17239/jowr-2019.10.03.04.

Westfall, Jacob & Tal Yarkoni. 2016. Statistically controlling for confounding constructs is harder than you think. *PLOS ONE* 11(3). e0152719. doi:10.1371/journal.pone.0152719.