Quantitative Methods – Day 1: Regression Basics

Harm H. Schuett

06 December, 2017

Contents


```
# Imports
library(scatterplot3d)
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 3.4.3
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
library(tibble)
```

```
##
## Attaching package: 'tibble'
```

```
## The following object is masked from 'package:igraph':
##
##     as_data_frame
```

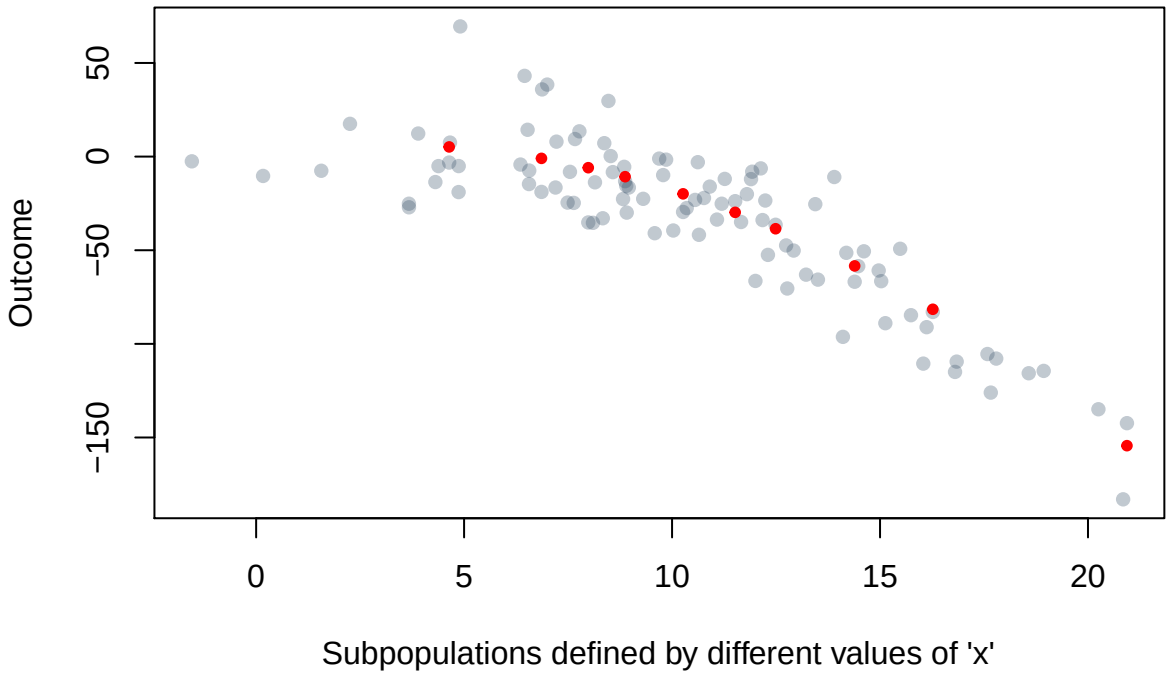
```
darkblue <- rgb(0.2, 0.3, 0.4, 0.3)  # define a color for later use
```

What is a Regression?

> Linear regression is a method that summarizes how the average values of a numerical *outcome* variable vary over subpopulations defined by linear functions of *predictors*. (Gelman.2007, p. 31)

This is a very nice description of what a linear regression is, but it might be a bit technical sounding at first. Visualizing this might help:

```
set.seed(123)  # fix the random number generator
x <- rnorm(100, 10, 5)  # draw 100 numbers randomly from a normal distribution
y <- 2 + 3 * x - 0.5*x^2 + rnorm(100, 0, 20)  # create a new variable y as linear additive function of
plot(x, y,  # plot x vs y (draws points by default)
     col=darkblue,  # color of the points
     pch=16,  # point character, 16 is a small dot
     xlab="Subpopulations defined by different values of 'x'",
     ylab="Outcome")
chosen_sub_population <- x[rank(x) %% 10 == 0]  # keep the 10th highest, 2th highest, etc. value
y_ticks <- 2 + 3 * chosen_sub_population - 0.5 * chosen_sub_population^2
points(chosen_sub_population, y_ticks, col="red", pch=20)  # adds the average y in red for evey ticks v
```

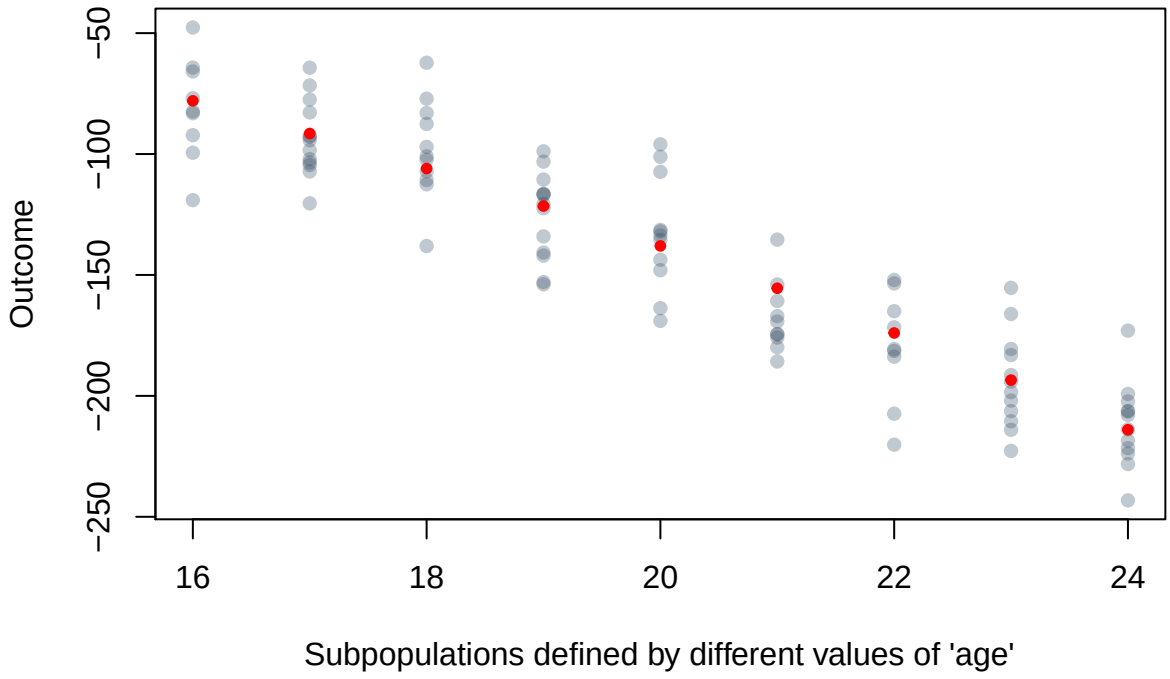


Think about the x dimension as a continuum of groups. People/firms/observations with a low value of x belong to a different group (subpopulation) than observations with a high value of x. The regression function describes how the outcome y changes "on average" as we move from one group to the next.

This becomes even more obvious with "categorical" x rather than contnuous x.

```
set.seed(123)  # fix the random number generator
age_range <- 16:24
age <- sample(x=age_range, size=100, replace=TRUE)  # draw 100 integers randomly
y <- 2 + 3 * age - 0.5*age^2 + rnorm(100, 0, 20)  # create a new variable y as linear additive function
plot(age, y,  # plot x vs y (draws points by default)
     col=darkblue,  # color of the points
     pch=16,  # point character, 16 is a small dot
```

```
     xlab="Subpopulations defined by different values of 'age'",
     ylab="Outcome")
y_ticks <- 2 + 3 * age_range - 0.5 * age_range^2
points(age_range, y_ticks, col="red", pch=20)  # adds the average y in red for evey ticks value
```



But even if x is continuous, the idea applies: a regression is way to describe how the average of an outcome changes going from one group of observations to another. It is a tool to reason about determinants of averages!

Conditional Expectations

> A substantial portion of research in econometric methodology can be interpreted as finding ways to estimate conditional expectations in the numerous settings that arise in economic applications. ... we are interested in the e¤ect of a variable w on the expected value of y, holding fixed a vector of controls, c. The conditional expectation of interest is $E[y|w, c]$, which we will call a **structural conditional expectation.** (Wooldridge (2010), p. 13)

A conditional expectation is nothing more than the expected average y given what you know about x_i (or w_i in Wooldridge (2010)) and c_i and what know about the relation between the variables. This structural conditional expectation is essentially your assumed model of the links between y and x and c. It can be detailed – up to assuming a certain functional form and placing restrictions on the coefficients – or it can be rather vague. And as we will discuss in the latter parts of this course: That expectation can be uncertain to different degrees.

Let's look at the simple *linear* example. Of course, there are also non-linear versions and generalized linear models. We will talk about them later. For now, let's try to grasp the motivation behind the most common

form: the simple linear regression.

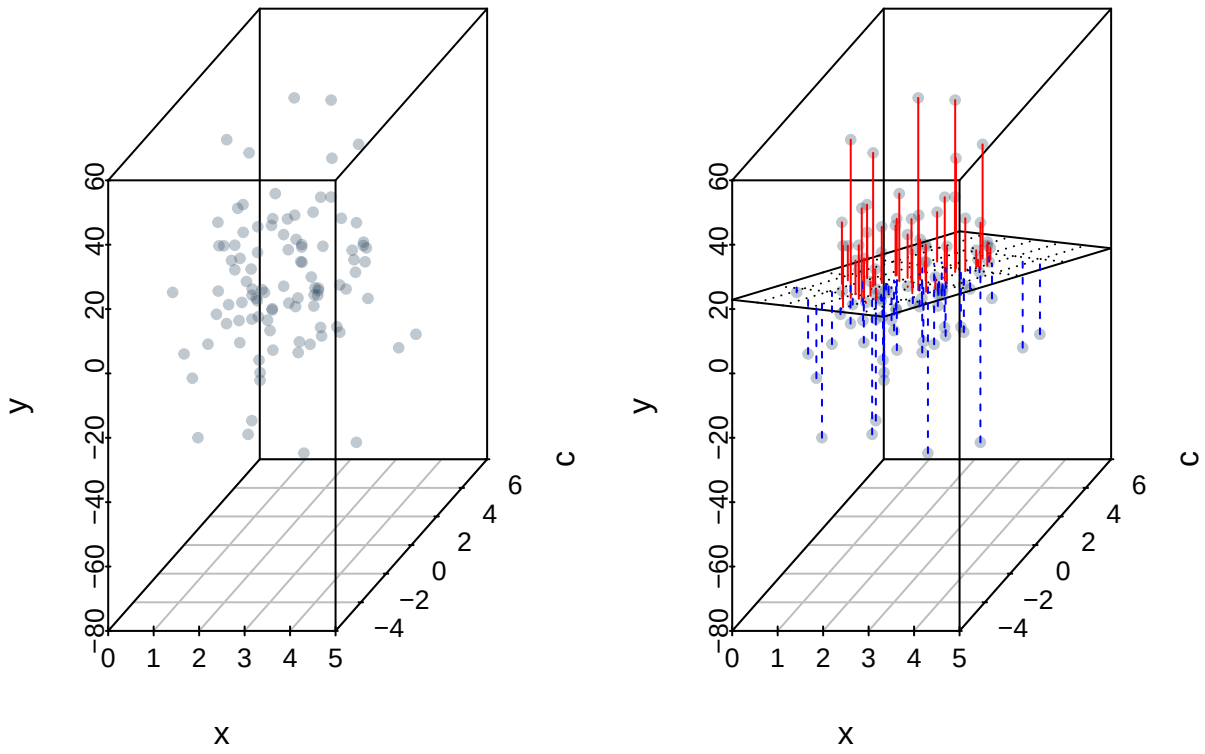
$$y_i = a_0 + a_1 \times x_i + a_2 \times c_i + u_i$$

The regression $y_i = a_0 + a_1 \times x_i + a_2 \times c_i + u_i$ in fact has a structural conditional expectation at its core:

$$E[y|x, c] = a_0 + a_1 \times x + a_2 \times c$$

Or in pictures:

```
set.seed(2345)  # fixing so that we always get the same numbers
x <- rnorm(100, 2, 1)
c <- rnorm(100, 3, 2)
u <- rnorm(100, 0, 20)
y <- 1 + 3 *x - 3 * c + u
par(mfrow=c(1,2))  # set graphics parameters. 1 row 2 columns
# 3d scatter plots. Drawing points
s3d <- scatterplot3d(x,c,y, pch=20, color=darkblue,
                     mar = c(3, 3, 2, 2), main="")
s3d <- scatterplot3d(x,c,y, pch=20, color=darkblue,
                     mar = c(3, 3, 2, 2), main="")
# adding the regression plance
fit1 <- lm(y ~ x+c)  # fitting a regression model
# adding the regression plane to the second plot
s3d$plane3d(fit1, lty.box="solid", lty="dotted")
# computing difference between y and fitted plane
orig     <- s3d$xyz.convert(x, c, y)
plane    <- s3d$xyz.convert(x, c,  fitted(fit1))
# labeling whether lines needs to be drawn red or blue
i.negpos <- 1 + (resid(fit1) > 0) # 1 for neg 2 for pos
# drawing the lines
segments(orig$x, orig$y, plane$x, plane$y,
         col=c("blue", "red")[i.negpos],  # color
         lty=(2:1)[i.negpos]  # line type
         )
```



As you can see the "conditional expectation" is a regression plane. The plane gives you the average value of y for a given combination of x and c. Mind you, this is only the case because we created the data y as a linear combination of x anc c. If say the true relationship is polynomial and we would fit a linear line through it, we would not be able to fit the average well. But, we would get the best linear predictor.

OLS and Conditional Expectation

Let's do some quick math to show that a regression indeed gives you the conditional expectation. And why we want to know that. The most common regressions are often also called the *minimum mean square linear predictor* or the *least squares linear predictor*. What can be shown quite easily is that the conditional expectation CE is the minimum mean square predictor of y. (See Goldberger (1991))

Assume we want to predict y with x. We need a good prediction function. Let's say we want to minimize our prediction errors, or more specifically, our squared prediction errors:

$$\underset{m \in M}{\text{minimize}} \; E\left[(y_i - m(x_i))^2\right]$$

m is a prediction function. We don't yet know, which one we want, other than that we want the one that minimizes the expected squared error from our predictions. M is the set of all possible functions. This is a difficult optimiziation problem. But we do not need to solve it, we just need to check whether our candidate, the $CE = E[y|x]$ is the solution. Let's expand the inner term with the CE:

$$\underset{m \in M}{\text{minimize}} \; E\left[([y_i - E[y|x]] + [E[y|x] - m(x_i)])^2\right]$$

The inner part is a binomial formula. So:

$$\underset{m \in M}{\text{minimize}}\ E\left[(y_i - E[y|x])^2 + 2\,(y_i - E[y|x])\,(E[y|x] - m(x_i)) + (E[y|x] - m(x_i))^2\right]$$

The first of the three terms doesn't depend on $m(x_i)$; so this term is irrelevant for the minimization. The second is zero. So this is equivalent to solving

$$\underset{m \in M}{\text{minimize}}\ E\left[(E[y|x] - m(x_i))^2\right]$$

Now, it is obvious that this term is minimized (set to zero) if $m(x_i) = E[y|x]$. So the conditional expectation function is what you should use if you want to minimize squared prediction errors. And the other way around as well. **If you are interested in the conditional expectation, minimizing mean squared prediction errors gets you there.**

How does simple ordinary least squares fit in? Suppose we want to limit ourselfs to linear functions out of M:

$$\beta = \underset{b}{\text{argmin}}\ E\left[(y_i - x'b)^2\right]$$

This is a bit easier. We can compute the first derivative and set it to zero to get the first order condition for a minimum or maximum:

$$2E\left[x\,(y_i - x'\beta)\right] = 0$$

Reshuffling on both sides of the equal sign gives you the OLS projection (*not the OLS estimator per se*):

$$\beta = E\left[xx'\right]^{-1} E\left[xy\right]$$

So the OLS projection is the minimum mean squared error linear predictor of y. By applying the same expansion trick as we did above we can also show that OLS is MMSE predictor of the CE and also that both are equal if the CE is indeed linear.

$$\beta = \underset{b}{\text{argmin}}\ E\left[(y_i - x'b)^2\right] = E\left[(E[y|x] - x'b)^2\right]$$

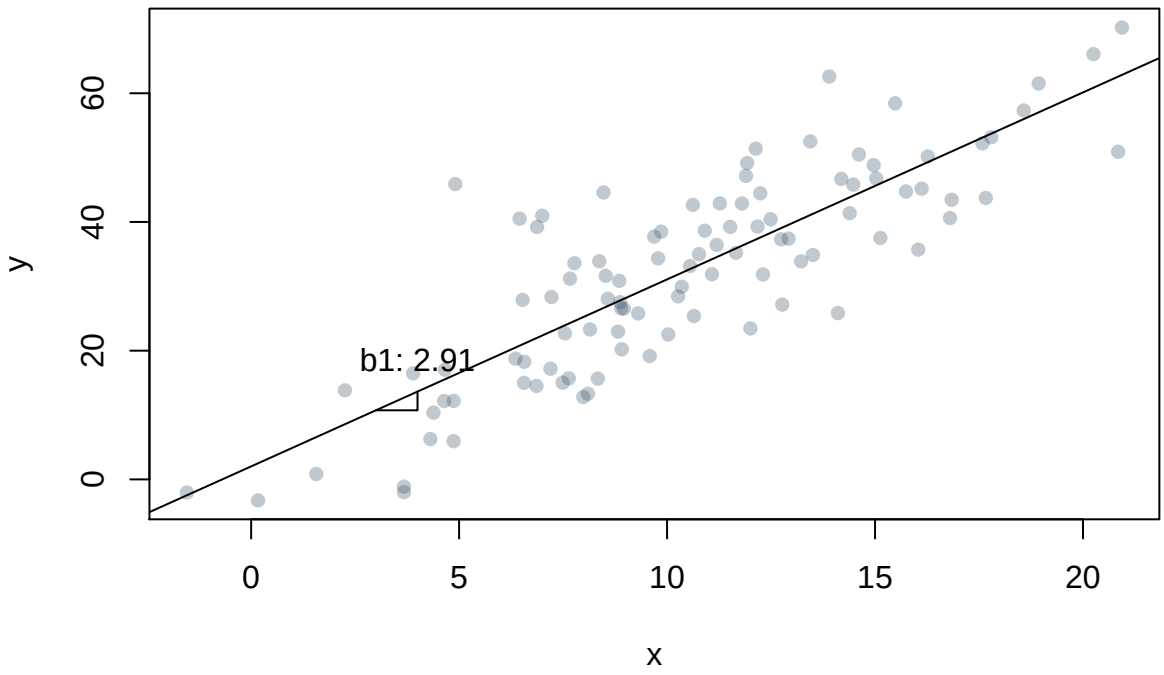
These are the reasons why OLS is so popular: It is the best linear approximation to an nearly always unknown CE. Why is that good? Because it is a "conservative" approximation if you don't know what the real CE is.

Interpreting Coefficients

What do we gain from approximating the CE? How are the coefficients to be interpreted?

```
set.seed(123)
x <- rnorm(100, 10, 5)
y <- 2 + 3 * x + rnorm(100, 0, 9)
plot(x, y, col=darkblue, pch=16)
uni_reg <- lm(y~x)
betas <- coef(uni_reg)
abline(uni_reg)
x_pos <- 4
text(x_pos, betas[1] + betas[2] * x_pos + 5, labels=paste("b1:", round(betas[2], 2)))
```

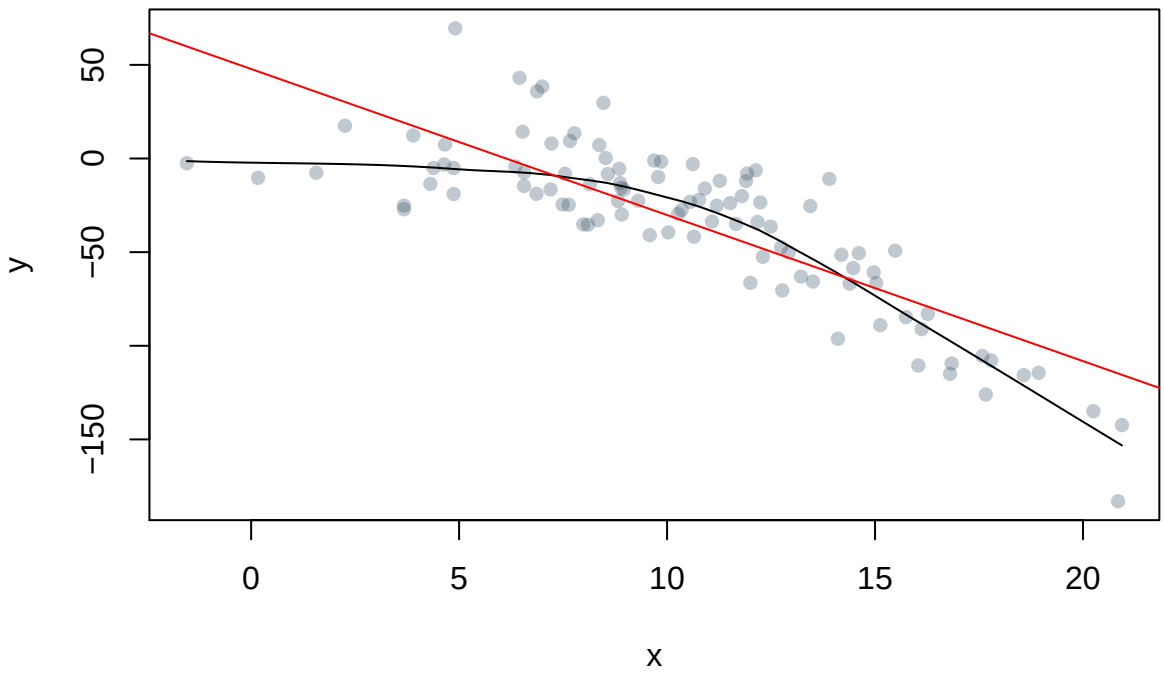
```r
triangle_x <- c(3, 4, 4)
triangle_y <- betas[1] + betas[2] * triangle_x
triangle_y[2] <- triangle_y[1]
lines(triangle_x, triangle_y)
```



For better visibility, here is the univariate case again. The coeeficients represent the slope of the (often many-dimensional) linar projection. It tells you how the expected value of y – the average y you should expect – changes in units of y with one additional unit of x.

But remember: A linear regression gives you the best linear predictor, which coincides with the conditional expecatation only if the true data generating process in indeed linear.

```r
set.seed(123)
x <- rnorm(100, 10, 5)
y <- 2 + 3 * x - 0.5*x^2 + rnorm(100, 0, 20)
scatter.smooth(x, y, col=darkblue, pch=16)
# plot(x, y, col=darkblue, pch=16)
abline(lm(y~x), col="red")
```



Here, we simulated a non-linear relationship (and it is not even obvious from the picture). The black line is the true CE, the red line is the best linear approximation of the black line.

The multivariate case

In the following cases we will use uppercase laters to denote Matrices, if a small cap letter does not have a subscript, consider it a vector

Consider the model:

$$y = X_1\beta_1 + X_2\beta_2 + u$$

One of the great things (and terribly confusing things) about multivariate Regressions is that the coefficients can be interpreted as the average change in y while holding other variables fixed. In the above equation β_1 is the avrage change in y for a 1-unit change in X_1, holding X_2 constant. The same goes for β_2.

Let's see this with an example:

```
set.seed(1234)
x1 <- rnorm(100, 3, 10)
x2 <- 0.3 * x1 + rnorm(100, 1, 3)
y = 1 + 2 * x1 - 2 * x2 + rnorm(100, 0, 1)
summary(lm(y ~ x1))
```

```
##
## Call:
```

```
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.0501  -3.8176  -0.4303   3.5982  17.3140
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.12648    0.62688  -1.797   0.0754 .
## x1           1.42347    0.06209  22.925   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.205 on 98 degrees of freedom
## Multiple R-squared:  0.8428, Adjusted R-squared:  0.8412
## F-statistic: 525.6 on 1 and 98 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1933 -0.6734  0.1060  0.5823  2.7650
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.10999    0.10348   10.73   <2e-16 ***
## x1           1.99922    0.01327  150.67   <2e-16 ***
## x2          -1.97056    0.03124  -63.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9624 on 97 degrees of freedom
## Multiple R-squared:  0.9963, Adjusted R-squared:  0.9962
## F-statistic: 1.291e+04 on 2 and 97 DF,  p-value: < 2.2e-16
```

The proof for why we we get this result is the **Frisch-Waugh-Lovell Theorem**.

For the theorem, we first need to introduce a certain way of presenting residuals, called the residual maker matrix $M = I - X(X'X)^{-1}X'$: It is called the residual maker because if you left-multiply it to a vector, you get the residual vector:

$$My = (I - X(X'X)^{-1}X')y = y - X(X'X)^{-1}X'y = y - X\beta = u$$

Multiplying M on the left with y yields the residual u. We need this matrix for the following proof:

Consider again: $y = X_1\beta_1 + X_2\beta_2 + u$. We are interested in the interpretation of β_2, which is the effect of X_2 on y hodling X_1 constant. In a way we could take out the relation of X_1 with both y and X_2 and regress what is left of y and X_2. We can use the residual maker matrix for that $\tilde{y} = M_1 y$ and $\tilde{X}_2 = M_1 X_2$ are the residuals after taking out the linear relation with X_1. Then regress $\tilde{y}$ on $\tilde{X}_2$

$$\tilde{y} = \tilde{X}_2 a_2 + u_2$$

Now, this is a regression of y, after taking out the relation with X_1, on X_2, after taking out its relation with X_1. The first part of the FWL Theoreom shows that $a_2 = \beta_2$, supporting the partialling out interpretation of β_2. The second part (which we won't look at) shows that you also get the same standard errors.

Let's look at a_2:

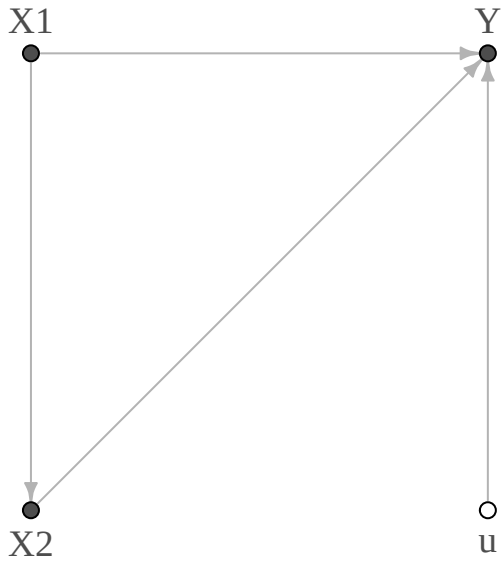
$$
\begin{aligned}
a_2 &= (\tilde{X}_2'\tilde{X}_2)^{-1}\tilde{X}_2'\tilde{y} && (1)\\
&= (X_2'M_1'M_1X_2)^{-1}X_2'M_1'M_1y && (2)\\
&= (X_2'M_1X_2)^{-1}X_2'M_1y && (3)\\
&= (X_2'M_1X_2)^{-1}X_2'M_1(X_1\beta_1 + x_2\beta_2 + \hat{u}) && (4)\\
&= (X_2'M_1X_2)^{-1}(X_2'M_1X_1\beta_1 + X_2'M_1X_2\beta_2 + X_2'M_1\hat{u}) && (5)
\end{aligned}
$$

There are a few things that help here, which I won't proof: $M_1'M_1 = M_1$, $M_1X = 0$, $M_1\hat{u} = \hat{u}$, and $X_2\hat{u} = 0$. That means, the above equation simplifies, because the first and last term in the bracket are zero:

$$a_2 = (X_2'M_1X_2)^{-1}(X_2'M_1X_2\beta_2) = \beta_2$$

So, a_2 from a short regression of $\tilde{y}$ on $\tilde{X}_2$ is the same as the β_2 of the long regression. Therefore, you can interpret β_2 as the relation between y and X_2 after X_1 has been controlled for. And "controlled for" means taking out the relation of X_1 out of *both* y and X_2. So, in the last simulated regression above, $y_i = 1 + 2x_1 - 2x_2 + u$ and $x_2 = 0.3x_1 + e$.

```
Nodes <- tribble(
  ~nodes, ~x, ~y,
  "Y",     1,  1,
  "X1",    0,  1,
  "X2",    0,  0,
  "u",     1,  0)
Edges <- tribble(
  ~from,  ~to,
  "X1",    "Y",
  "X1",    "X2",
  "X2",    "Y",
  "u",     "Y")
plot(graph_from_data_frame(vertices=Nodes, d=Edges, directed=TRUE),
     vertex.color=c(rep("gray30", nrow(Nodes)-1), "white"),
     vertex.size=7,
     label.font=2, vertex.size=30, vertex.label.cex=1.2,
     edge.arrow.size=0.5, edge.color="gray70",
     vertex.label.color="gray30", vertex.label.dist=2,
     vertex.label.degree=c(-pi/2, -pi/2, pi/2, pi/2))
```



So, x_1 and x_2 are related. If we simply regress $y = \hat{a}_0 + \hat{a}_1 x_1 + \hat{u}$, we get $a_1 = 1.4$. Why? Because x_1 has an additional effect on y via x_2; since x_2 has a negative effect on y, the overall effect is less than 2. But, if we regress $y = \hat{b}_0 + \hat{b}_1 x_1 + \hat{b}_2 x_2 + \hat{u}$, we get $b_1 = 2.00$ and $b_2 = -1.97$. Why, because if we take out the effect of x_2 on y, the additional channel of x_1 via x_2 is blocked. (We do not have to take out anything out of x_1, because x_2 does not affect x_1.)

As a word of caution:

> The problem is that regression is like a demon or djinn from folktale. It answers exactly that question that you asked it. Often you don't realize though that you asked the wrong question ~ paraphrasing Richard McElreath

We will talk about the pitfalls of interpreting coefficients tomorrow.

Excercises

1. What would happen in the last example (the one tied to the last arrow graph) if you regress y on x_2 only? Explain why? Then run the regression.
2. Use the idea of Frisch-Waugh-Lovell Theorem to simulate and see whether you get the same coefficients either from regressing residuals or multivariate regressions.

References

Goldberger, Arthur Stanley. 1991. *A Course in Econometrics*. Harvard University Press.

Wooldridge, Jeffrey M. 2010. *Econometric Analysis of Cross Section and Panel Data*. MIT Press.