Multilinear regression as loss minimization.

Goals

- Derive the general form of the ordinary least squares (OLS) estimator in matrix notation
  - Introduce the multilinear regression problem as a generalization of simple linear regression
  - Review matrix notation in the context of multilinear regression
  - Derive the general OLS formula and show that the simple least squares is a special case

Reading

These notes are a supplement for the following readings:

- Freedman (2009) Section 4.1
- Ding (2024) Section 3.1
- Davidson, MacKinnon, et al. (2004) Sections 1.4–1.5

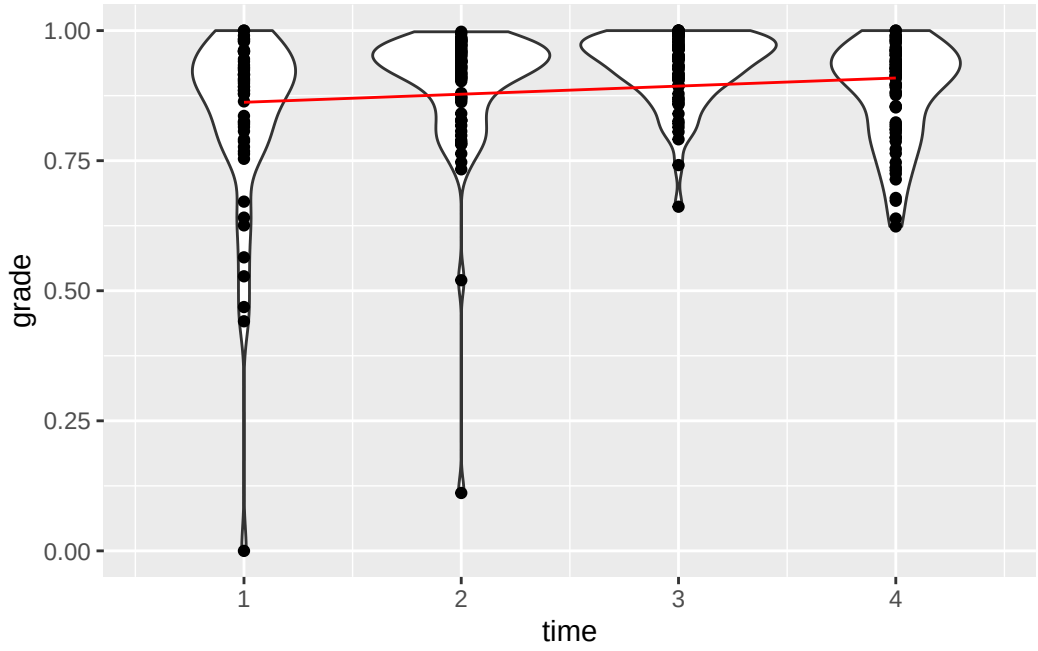
Have my grades been increasing over time?

Let's look again at the grades dataset, and consider the question: "have my grades been increasing over time?" I personally am interested in whether some aspect of my courses has been causing an upward trend, and maybe you are interested in extrapolating such a trend to the present semester.

```
grade_v_time <- lm(grade ~ time, all_grades_df)
print(summary(grade_v_time))
```

```
Call:
lm(formula = grade ~ time, data = all_grades_df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.86223 -0.03835  0.03343  0.07532  0.13777

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.846658   0.020112  42.098   <2e-16 ***
time        0.015571   0.007198   2.163   0.0315 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1234 on 252 degrees of freedom
Multiple R-squared:  0.01823,   Adjusted R-squared:  0.01434
F-statistic:  4.68 on 1 and 252 DF,  p-value: 0.03146
```

```r
grade_pred <- predict(grade_v_time, all_grades_df)
all_grades_df %>%
ggplot() +
  geom_violin(aes(x=time, y=grade, group=time)) +
  geom_point(aes(x=time, y=grade)) +
  geom_line(aes(x=time, y=grade_pred), color="red")
```



Here, we see a very weak but increasing trend, probably driven by some very low grades in the first two semesters.

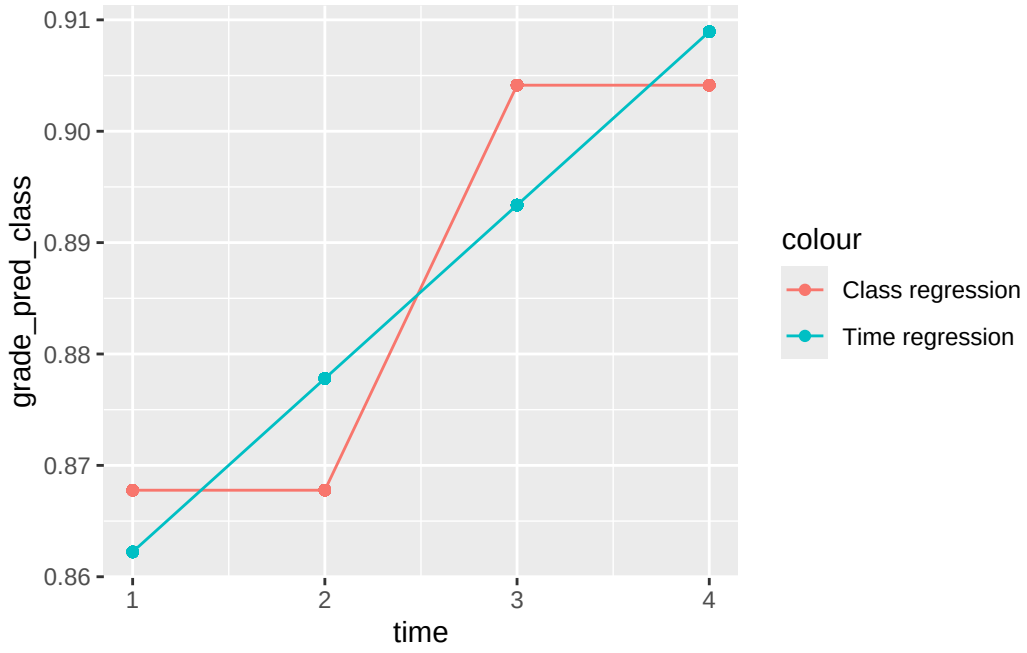
However, this comparison, even as it is, is complicated by the fact that time periods 1 and 2 were 151A, and time periods 3 and 4 were 154. In fact, if regress on the class label instead, I get a simliar result:

```
grade_v_class <- lm(grade ~ class, all_grades_df)
summary(grade_v_class)
```

```
Call:
lm(formula = grade ~ class, data = all_grades_df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.86777 -0.04168  0.03599  0.07514  0.13223

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.86777    0.01120  77.479   <2e-16 ***
class154     0.03636    0.01548   2.349   0.0196 *
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1232 on 252 degrees of freedom
Multiple R-squared:  0.02143,   Adjusted R-squared:  0.01755
F-statistic: 5.519 on 1 and 252 DF,  p-value: 0.01958
```

```r
grade_v_class <- lm(grade ~ class, all_grades_df)
grade_v_time <- lm(grade ~ time, all_grades_df)
grade_pred_class <- predict(grade_v_class, all_grades_df)
grade_pred_time <- predict(grade_v_time, all_grades_df)
all_grades_df %>%
  ggplot() +
  geom_line(aes(x=time, y=grade_pred_class, color="Class regression")) +
  geom_line(aes(x=time, y=grade_pred_time, color="Time regression")) +
  geom_point(aes(x=time, y=grade_pred_class, color="Class regression")) +
  geom_point(aes(x=time, y=grade_pred_time, color="Time regression"))
```



Here, we can see how the two fits differ. The prediction for regression on class is a step function for the change in which class is being taught, where the time trend is continuous.

> **Question:**
>
> Can we possibly disentangle the effect of time from which class is being taught? How?

One way to ask this formally is to run the regression on both:

```
lm(grade ~ time + class, all_grades_df) %>% summary()
```

```
Call:
lm(formula = grade ~ time + class, data = all_grades_df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.86615 -0.04175  0.03580  0.07452  0.13385

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.863349   0.027039  31.930   <2e-16 ***
time        0.002801   0.015584   0.180    0.858
class154    0.031012   0.033566   0.924    0.356
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1234 on 251 degrees of freedom
Multiple R-squared:  0.02156,   Adjusted R-squared:  0.01376
F-statistic: 2.765 on 2 and 251 DF,  p-value: 0.06488
```

Interestingly, we see that though each regression is statistically significant separately, in the combined regression neither are significant. **This is one way of saying, using regression, that we cannot disentangle the effect of class and time with this dataset.** In other words, without more data, the answer to our question is inconclusive — although there certainly doesn't seem to be strong evidence for marked trends.

This is no longer simple linear regression! In this unit, we will study versions of regression that include more than one covariate in this way.

Matrix notation

The simple linear regression formula came from combining the equations that set the univariate gradients equal to zero, and then recognizing a matrix equation. We can in fact do both at the same time! But first we need some notation

Here is a formal definition of the type of model that we will study for the vast majority of the semester:

$$y_n = \beta_1 x_{n1} + \beta_2 x_{n2} + ... + \beta_P x_{nP} + \varepsilon_n, \quad \text{For } n = 1, ..., N. \tag{1}$$

> **Notation**
>
> I will always use N for the number of observed data points, and P for the dimension of the regression vector.

Equation 1 is a general form of simpler cases. For example, if we take $x_{n1} \equiv 1$, $x_{n2} = x_n$ to be some scalar, and $P = 2$, then Equation 1 becomes **?@eq-lm-simple**:

$$y_n = \beta_1 + \beta_2 x_n + \varepsilon_n, \quad \text{For } n = 1, ..., N.$$

The residuals ε_n measure the "misfit" of the line. If you know $\beta_1, ..., \beta_P$, then you can compute

$$\varepsilon_n = y_n - (\beta_1 x_{n1} + \beta_2 x_{n2} + ... + x_{nP}).$$

But in general we only observe y_n and $x_{n1}, ..., x_{nP}$, and we choose $\beta_1, ..., \beta_P$ to make the residuals small. (How we do this precisely will be something we talk about at great length.)

The general form of Equation 1 can be written more compactly using matrix and vector notation. Specifically, if we let

$$x_n := \begin{pmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{nP} \end{pmatrix} \quad \text{and} \quad \beta := \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_P \end{pmatrix}$$

> **Notation**
>
> Bold lowercase variables are column vectors (unless otherwise specified).

Recall that the "transpose" operator $(\cdot)^\top$ flips the row and columns of a matrix. For example,

$$x_n^\top = \begin{pmatrix} x_{n1} & x_{n2} & ... & x_{nP} \end{pmatrix}.$$

By matrix multiplication rules,

$$x_n^\top \beta = \begin{pmatrix} x_{n1} & x_{n2} & ... & x_{nP} \end{pmatrix} \qquad \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_P \end{pmatrix} = \beta_1 x_{n1} + \beta_2 x_{n2} + ... + x_{nP}.$$

> **Notation**
>
> I have written $x_n^\top \beta$ for the "dot product" or "inner product" between x_n and β. Writing it in this way clarifies the relationship with matrix notation below.
> There are many other ways to denote inner products in the literature, including $x_n \cdot \beta$ and $< x_n, \beta >$.

Then we can compactly write

$$y_n = x_n^\top \beta + \varepsilon_n, \quad \text{For } n = 1, ..., N.$$

We can compactify it even further if we stack the n observations: %

$$\begin{aligned} y_1 &= x_1^\top \beta + \varepsilon_1 \\ y_2 &= x_2^\top \beta + \varepsilon_2 \\ &\vdots \\ y_N &= x_N^\top \beta + \varepsilon_N \end{aligned}$$

As before we can stack the responses and residuals:

$$Y := \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_P \end{pmatrix} \quad \text{and} \quad \varepsilon := \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_P \end{pmatrix}$$

We can also stack the regressors:

$$X := \begin{pmatrix} x_{11} & x_{12} & ... & x_{1P} \\ x_{21} & x_{22} & ... & x_{2P} \\ \vdots & & & \\ x_{n1} & x_{n2} & ... & x_{nP} \\ \vdots & & & \\ x_{N1} & x_{N2} & ... & x_{NP} \end{pmatrix} = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \\ \vdots \\ x_N^\top \end{pmatrix}$$

> **Notation**
>
> I will use upper case bold letters for multi-dimensional matrices like X. But I may also use upper case bold letters even when the quantity could also be a column vector, when I think it's more useful to think of the quantity as a matrix with a single column. Examples are Y above, or X when $P = 1$.

Note that by matrix multiplication rules,

$$X = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \\ \vdots \\ x_N^\top \end{pmatrix} \qquad X\beta = \begin{pmatrix} x_1^\top \beta \\ x_2^\top \beta \\ \vdots \\ x_n^\top \beta \\ \vdots \\ x_N^\top \beta \end{pmatrix}$$

so we end up with the extremely tidy expression

$$y_n = \beta_1 x_{n1} + \beta_2 x_{n2} + ... + x_{nP} + \varepsilon_n, \quad \text{For } n = 1, ..., N$$

$$\text{is the same as} \tag{2}$$

$$Y = X\beta + \varepsilon.$$

In the case of simple least squares, we can write

$$X := \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix}, \tag{3}$$

and verify that the n–th row of Equation 2 is the same as **?@eq-lm-simple**.

Least squares in matrix notation

Using our tidy expression Equation 2, we can easily write out the sum of the squared errors as

$$\begin{aligned}\sum_{n=1}^{N} \varepsilon_n^2 = \varepsilon^\top \varepsilon &= (Y - X\beta)^\top (Y - X\beta) \\ &= Y^\top Y - \beta^\top X^\top Y - Y^\top X\beta + \beta^\top X^\top X\beta \\ &= Y^\top Y - 2Y^\top X\beta + \beta^\top X^\top X\beta.\end{aligned}$$

This is a quadratic function of the vector β. We wish to find the minimum of this quantity as a function of β. We might hope that the minimum occurs at a point where the gradient of this expression is zero.

Rather than compute the *univariate* derivative with respect to each component, we can compute the *multivariate gradient* with respect to the vector.

Let's recall some facts from vector calculus.

Notation

Take $z \in \mathbb{R}^P$ to be a P–vector. and let $f(z)$ a scalar–valued function of the vector z. We write

$$\frac{\partial f(z)}{\partial z} = \begin{pmatrix} \frac{\partial}{\partial z_1} f(z) \\ \vdots \\ \frac{\partial}{\partial z_P} f(z) \end{pmatrix}.$$

That is, the partial $\frac{\partial f(z)}{\partial z}$ is a P–vector of the stacked univariate dervatives.

Recall a couple rules from vector calculus. Let v denote a P–vector and A a symmetric matrix. Then

$$\frac{\partial v^\top z}{\partial z} = v \quad \text{and} \quad \frac{\partial z^\top A z}{\partial z} = 2Az.$$

Exercise

Prove these results above using univariate derivatives and our stacking convention.

Applying these two rules to our least squares objective,

$$\begin{aligned}\frac{\partial \varepsilon^\top \varepsilon}{\partial \beta} &= \frac{\partial}{\partial \beta} Y^\top Y - 2\frac{\partial}{\partial \beta} Y^\top X\beta + \frac{\partial}{\partial \beta} \beta^\top X^\top X\beta \\ &= 0 - 2X^\top Y + 2X^\top X\beta.\end{aligned}$$

Assuming our estimator $\hat{\beta}$ sets these partial derivatives are equal to zero, we then get

$$X^\top X\hat{\beta} = X^\top Y. \tag{4}$$

This is a set of P equations in P unknowns. If it is not degenerate, one can solve for $\hat{\beta}$. That is, if the matrix $X^\top X$ is invertible, then we can multiply both sides of Equation 4 by $(X^\top X)^{-1}$ to get

$$\hat{\beta} = (X^\top X)^{-1}X^\top Y \tag{5}$$

Notation

We're going to be talking again and again about the "ordinary least squares" ("OLS") problem

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\, \varepsilon^\top\varepsilon \quad \text{where} \quad Y = X\beta + \varepsilon \quad \text{where} \quad y_n = x_n^\top\beta + \varepsilon_n \text{ for all } n = 1, \ldots, N.$$

It will be nice to have some shorthand for this problem so I don't have to write this out every time. All of the following will be understood as shorthand for the preceding problem. In each, the fact that $\hat{\beta}$ minimizes the sum of squared residuals is implicit.

$Y \sim X\beta + \varepsilon$	Only the least squares criterion is implicit
$Y \sim X\beta$	ε implicit
$Y \sim X$	ε, β implicit
$y_n \sim x_n^\top\beta$	ε_n, N implicit
$y_n \sim x_n^\top\beta + \varepsilon_n$	N implicit
$y_n \sim x_n$	ε_n, β, N implicit
$y_n \sim x_{n1} + x_{n2} + \ldots + x_{nP}$	ε_n, β, N implicit

(The final shorthand is closest to the notation for the `R` `lm` function.) This is convenient because, for example, certain properties of the regression $y_n \sim x_n$ don't necessarily need to commit to which symbol we use for the coefficients. Symbols for the missing pieces will hopefuly be clear from context as necessary.

Notation

Unlike many other regression texts (and the `lm` function), I will *not* necessarily assume that a constant is included in the regression. One can always take a generic regression $y_n \sim x_n$ to include a constant by assuming that one of the entries of x_n is one. At some points my convention of not including a constant by default will lead to formulas that

may be at odds with some textbooks. But these differences are superficial, and are, in my mind, more than made up for by the generality and simplicity of treating constants as just another regressor.

Some simple familiar examples in matrix form

The sample mean

Sample means are in fact a special case of multi linear regression. Seeing this will be helpful when we interpret categorical variables.

> Notation
>
> I will use 1 to denote a vector full of ones. Usually it will be an N–vector, but sometimes its dimension will just be implicit. Similarly, 0 is a vector of zeros.

We showed earlier that the sample mean is a special case of the regression $y_n \sim 1 \cdot \beta$. This can be expressed in matrix notation by taking $X = 1$ as a $N \times 1$ vector. We then have

$$X^\top X = 1^\top 1 = \sum_{n=1}^{N} 1 \cdot 1 = N,$$

so $X^\top X$ is invertible as long as $N > 0$ (i.e., if you have at least one datapoint), with $(X^\top X)^{-1} = 1/N$. We also have

$$X^\top Y = 1^\top Y = \sum_{n=1}^{N} 1 \cdot y_n = N\bar{y},$$

and so

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y = (1^\top 1)^{-1} 1^\top Y = \frac{N\bar{y}}{N} = \bar{y},$$

as expected.

A single regressor

For completness, let's also see what happens when we omit the constant from simple linear regression. Suppose that we regress $y_n \sim x_n$ where x_n is a scalar.

> **Warning**
>
> `R` adds a constant by default! To remove it, run something like `lm(y ~ x - 1, my_dataframe)`.

Let's suppose that $\mathbb{E}[x_n] = 0$ and $\text{Var}(x_n) = \sigma^2 > 0$. We have

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

so

$$X^\top X = \sum_{n=1}^{N} x_n^2.$$

Depending on the distribution of x_n, it may be possible for $X^\top X$ to be non–invertible!

> **Exercise**
>
> Produce a distribution for x_n where $X^\top X$ is non–invertible with positive probability for any N.

However, as $N \to \infty$, $\frac{1}{N} X^\top X \to \sigma^2$ by the LLN, and since $\sigma^2 > 0$, $\frac{1}{N} X^\top X$ will be invertible with probability approaching one as N goes to infinity.

Common quantities

Given a regression fit, there are a few key quantities that we'll use over and over again.

First of all, the fit is given by

$$\hat{Y} = X\hat{\beta} \quad \Leftrightarrow \quad \hat{y}_n = x_n^\top \hat{\beta}.$$

This is the regressions' "guess" at the response for a given value of x_n. Analogously, we can define the "error", "residual", or "fitted residual:"

$$\hat{\varepsilon} = Y - \hat{Y} \quad \Leftrightarrow \quad \hat{\varepsilon}_n = y_n - \hat{y}_n.$$

We can call

$$\begin{aligned}\hat{Y}^\top \hat{Y} &:= ESS = \text{"Explained sum of squares"} \\ Y^\top Y &:= TSS = \text{"Total sum of squares"} \\ \hat{\varepsilon}^\top \hat{\varepsilon} &:= RSS = \text{"Residual sum of squares"}.\end{aligned}$$

In your homework, you will show that $TSS = ESS + RSS$. It follows immediately that $0 \leq ESS \leq TSS$, and we can define

$$R^2 := \frac{ESS}{TSS} \in [0, 1].$$

In a sense, high R^2 means a "good fit" in the sense that the least squares fit has low error.

High R^2 is not necessarily a good fit

But high R^2 does not necessarily mean that the fit is accurate or useful! In particular, by increasing the number of regressors, you can only make R^2 increase, and there are clearly silly regressions with great R^2.

Here are two examples:

- $Y \sim Y$
- $Y \sim I$

Davidson, Russell, James G MacKinnon, et al. 2004. *Econometric Theory and Methods*. Vol. 5. Oxford University Press New York.
Ding, Peng. 2024. "Linear Model and Extensions." *arXiv Preprint arXiv:2401.00649*.
Freedman, David. 2009. *Statistical Models: Theory and Practice*. cambridge university press.