

15.11: Summary

- Basic ideas in linear regression and how regression models are estimated (Sections 15.1 and 15.2).
- Multiple linear regression (Section 15.3).
- Measuring the overall performance of a regression model using R^2 (Section 15.4)
- Hypothesis tests for regression models (Section 15.5)
- Calculating confidence intervals for regression coefficients, and standardised coefficients (Section 15.7)
- The assumptions of regression (Section 15.8) and how to check them (Section 15.9)
- Selecting a regression model (Section 15.10)

References

Fox, J., and S. Weisberg. 2011. *An R Companion to Applied Regression*. 2nd ed. Los Angeles: Sage.

Cook, R. D., and S. Weisberg. 1983. "Diagnostics for Heteroscedasticity in Regression." *Biometrika* 70: 1–10.

Long, J.S., and L.H. Ervin. 2000. "Using Heteroscedasticity Consistent Standard Errors in Thee Linear Regression Model." *The American Statistician* 54: 217–24.

Akaike, H. 1974. "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control* 19: 716–23.

214. The ϵ symbol is the Greek letter epsilon. It's traditional to use ϵ_i or e_i to denote a residual.
215. Or at least, I'm assuming that it doesn't help most people. But on the off chance that someone reading this is a proper kung fu master of linear algebra (and to be fair, I always have a few of these people in my intro stats class), it *will* help *you* to know that the solution to the estimation problem turns out to be $\hat{b} = (X^T X)^{-1} X^T y$, where \hat{b} is a vector containing the estimated regression coefficients, X is the "design matrix" that contains the predictor variables (plus an additional column containing all ones; strictly X is a matrix of the regressors, but I haven't discussed the distinction yet), and y is a vector containing the outcome variable. For everyone else, this isn't exactly helpful, and can be downright scary. However, since quite a few things in linear regression can be written in linear algebra terms, you'll see a bunch of footnotes like this one in this chapter. If you can follow the maths in them, great. If not, ignore it.
216. And by "sometimes" I mean "almost never". In practice everyone just calls it "R-squared".
217. Note that, although R has done multiple tests here, it hasn't done a Bonferroni correction or anything. These are standard one-sample t-tests with a two-sided alternative. If you want to make corrections for multiple tests, you need to do that yourself.
218. You can change the kind of correction it applies by specifying the `p.adjust.method` argument.
219. Strictly, you standardise all the *regressors*: that is, every "thing" that has a regression coefficient associated with it in the model. For the regression models that I've talked about so far, each predictor variable maps onto exactly one regressor, and vice versa. However, that's not actually true in general: we'll see some examples of this in Chapter 16. But for now, we don't need to care too much about this distinction.
220. Or have no hope, as the case may be.
221. Again, for the linear algebra fanatics: the "hat matrix" is defined to be that matrix H that converts the vector of observed values y into a vector of fitted values \hat{y} , such that $\hat{y} = Hy$. The name comes from the fact that this is the matrix that "puts a hat on y ". The *hat value* of the i -th observation is the i -th diagonal element of this matrix (so technically I should be writing it as h_{ii} rather than h_i). Oh, and in case you care, here's how it's calculated: $H = X(X^T X)^{-1} X^T$. Pretty, isn't it?
222. Though special mention should be made of the `influenceIndexPlot()` and `influencePlot()` functions in the `car` package. These produce somewhat more detailed pictures than the default plots that I've shown here. There's also an `outlierTest()` function that tests to see if any of the Studentised residuals are significantly larger than would be expected by chance.
223. An alternative is to run a "robust regression"; I'll discuss robust regression in a later version of this book.
224. And, if you take the time to check the `residualPlots()` for `regression.1`, it's pretty clear that this isn't some wacky distortion being caused by the fact that `baby.sleep` is a useless predictor variable. It's an actual nonlinearity in the relationship between `dan.sleep` and `dan.grump`.
225. Note that the underlying mechanics of the test aren't the same as the ones I've described for regressions; the goodness of fit is assessed using what's known as a score-test not an F-test, and the test statistic is (approximately) χ^2 distributed if there's no relationship

226. Again, a footnote that should be read only by the two readers of this book that love linear algebra (mmmm... I love the smell of matrix computations in the morning; smells like... nerd). In these estimators, the covariance matrix for \mathbf{b} is given by $(X^T X)^{-1} X^T \sum X (X^T X)^{-1}$. See, it's a "sandwich"? Assuming you think that $(X^T X)^{-1}$ = "bread" and $X^T \sum X$ = "filling", that is. Which of course everyone does, right? In any case, the usual estimator is what you get when you set $\sum = \hat{\sigma}^2 I$. The corrected version that I learned originally uses $\text{diag}(\epsilon_i^2)$ (White 1980). However, the version that Fox and Weisberg (2011)
227. Note, however, that the `step()` function computes the full version of AIC, including the irrelevant constants that I've dropped here. As a consequence this equation won't correctly describe the AIC values that you see in the outputs here. However, if you calculate the AIC values using my formula for two different regression models and take the difference between them, this will be the same as the differences between AIC values that `step()` reports. In practice, this is all you care about: the actual value of an AIC statistic isn't very informative, but the differences between two AIC values *are* useful, since these provide a measure of the extent to which one model outperforms another.
228. While I'm on this topic I should point out that there is also a function called `BIC()` which computes the Bayesian information criterion (BIC) for the models. So you could type `BIC(M0, M1)` and get a very similar output. In fact, while I'm not particularly impressed with either AIC or BIC as model selection methods, if you do find yourself using one of these two, the empirical evidence suggests that BIC is the better criterion of the two. In most simulation studies that I've seen, BIC does a much better job of selecting the correct model.
229. It's worth noting in passing that this same F statistic can be used to test a much broader range of hypotheses than those that I'm mentioning here. Very briefly: notice that the nested model M_0 corresponds to the full model M_1 when we constrain some of the regression coefficients to zero. It is sometimes useful to construct submodels by placing other kinds of constraints on the regression coefficients. For instance, maybe two different coefficients might have to sum to zero, or something like that. You can construct hypothesis tests for those kind of constraints too, but it is somewhat more complicated and the sampling distribution for F can end up being something known as the non-central F distribution, which is waaaaay beyond the scope of this book! All I want to do is alert you to this possibility.

This page titled [15.11: Summary](#) is shared under a [CC BY-SA 4.0](#) license and was authored, remixed, and/or curated by [Danielle Navarro](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.