

15.5: Hypothesis Tests for Regression Models

So far we've talked about what a regression model is, how the coefficients of a regression model are estimated, and how we quantify the performance of the model (the last of these, incidentally, is basically our measure of effect size). The next thing we need to talk about is hypothesis tests. There are two different (but related) kinds of hypothesis tests that we need to talk about: those in which we test whether the regression model as a whole is performing significantly better than a null model; and those in which we test whether a particular regression coefficient is significantly different from zero.

At this point, you're probably groaning internally, thinking that I'm going to introduce a whole new collection of tests. You're probably sick of hypothesis tests by now, and don't want to learn any new ones. Me too. I'm so sick of hypothesis tests that I'm going to shamelessly reuse the F-test from Chapter 14 and the t-test from Chapter 13. In fact, all I'm going to do in this section is show you how those tests are imported wholesale into the regression framework.

15.5.1 Testing the model as a whole

Okay, suppose you've estimated your regression model. The first hypothesis test you might want to try is one in which the null hypothesis that there is *no relationship* between the predictors and the outcome, and the alternative hypothesis is that *the data are distributed in exactly the way that the regression model predicts*. Formally, our "null model" corresponds to the fairly trivial "regression" model in which we include 0 predictors, and only include the intercept term b_0

$$H_0: Y_i = b_0 + \epsilon_i$$

If our regression model has K predictors, the "alternative model" is described using the usual formula for a multiple regression model:

$$H_1: Y_i = \left(\sum_{k=1}^K b_k X_{ik} \right) + b_0 + \epsilon_i$$

How can we test these two hypotheses against each other? The trick is to understand that just like we did with ANOVA, it's possible to divide up the total variance SS_{tot} into the sum of the residual variance SS_{res} and the regression model variance SS_{mod} . I'll skip over the technicalities, since we covered most of them in the ANOVA chapter, and just note that:

$$SS_{\text{mod}} = SS_{\text{tot}} - SS_{\text{res}}$$

And, just like we did with the ANOVA, we can convert the sums of squares in to mean squares by dividing by the degrees of freedom.

$$MS_{\text{mod}} = \frac{SS_{\text{mod}}}{df_{\text{mod}}}$$

$$MS_{\text{res}} = \frac{SS_{\text{res}}}{df_{\text{res}}}$$

So, how many degrees of freedom do we have? As you might expect, the df associated with the model is closely tied to the number of predictors that we've included. In fact, it turns out that $df_{\text{mod}} = K$. For the residuals, the total degrees of freedom is $df_{\text{res}} = N - K - 1$.

$$F = \frac{MS_{\text{mod}}}{MS_{\text{res}}}$$

and the degrees of freedom associated with this are K and $N - K - 1$. This F statistic has exactly the same interpretation as the one we introduced in Chapter 14. Large F values indicate that the null hypothesis is performing poorly in comparison to the alternative hypothesis. And since we already did some tedious "do it the long way" calculations back then, I won't waste your time repeating them. In a moment I'll show you how to do the test in R the easy way, but first, let's have a look at the tests for the individual regression coefficients.

15.5.2 Tests for individual coefficients

The F -test that we've just introduced is useful for checking that the model as a whole is performing better than chance. This is important: if your regression model doesn't produce a significant result for the F -test then you probably don't have a very good regression model (or, quite possibly, you don't have very good data). However, while failing this test is a pretty strong indicator that the model has problems, *passing* the test (i.e., rejecting the null) doesn't imply that the model is good! Why is that, you might be wondering? The answer to that can be found by looking at the coefficients for the `regression.2` model:

```
print( regression.2 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep + baby.sleep, data = parenthood)
##
## Coefficients:
## (Intercept)    dan.sleep    baby.sleep
##    125.96557     -8.95025      0.01052
```

I can't help but notice that the estimated regression coefficient for the `baby.sleep` variable is tiny (0.01), relative to the value that we get for `dan.sleep` (-8.95). Given that these two variables are absolutely on the same scale (they're both measured in "hours slept"), I find this suspicious. In fact, I'm beginning to suspect that it's really only the amount of sleep that *I* get that matters in order to predict my grumpiness.

Once again, we can reuse a hypothesis test that we discussed earlier, this time the t-test. The test that we're interested has a null hypothesis that the true regression coefficient is zero ($b=0$), which is to be tested against the alternative hypothesis that it isn't ($b \neq 0$). That is:

$$H_0: b=0$$

$$H_1: b \neq 0$$

How can we test this? Well, if the central limit theorem is kind to us, we might be able to guess that the sampling distribution of \hat{b} , the estimated regression coefficient, is a normal distribution with mean centred on b . What that would mean is that if the null hypothesis were true, then the sampling distribution of \hat{b} has mean zero and unknown standard deviation. Assuming that we can come up with a good estimate for the standard error of the regression coefficient, $SE(\hat{b})$, then we're in luck. That's *exactly* the situation for which we introduced the one-sample t way back in Chapter 13. So let's define a t-statistic like this,

$$t = \frac{\hat{b}}{SE(\hat{b})}$$

I'll skip over the reasons why, but our degrees of freedom in this case are $df=N-K-1$. Irritatingly, the estimate of the standard error of the regression coefficient, $SE(\hat{b})$, is not as easy to calculate as the standard error of the mean that we used for the simpler t-tests in Chapter 13. In fact, the formula is somewhat ugly, and not terribly helpful to look at. For our purposes it's sufficient to point out that the standard error of the estimated regression coefficient depends on both the predictor and outcome variables, and is somewhat sensitive to violations of the homogeneity of variance assumption (discussed shortly).

In any case, this t-statistic can be interpreted in the same way as the t-statistics that we discussed in Chapter 13. Assuming that you have a two-sided alternative (i.e., you don't really care if $b>0$ or $b<0$), then it's the extreme values of t (i.e., a lot less than zero or a lot greater than zero) that suggest that you should reject the null hypothesis.

15.5.3 Running the hypothesis tests in R

To compute all of the quantities that we have talked about so far, all you need to do is ask for a `summary()` of your regression model. Since I've been using `regression.2` as my example, let's do that:

```
summary( regression.2 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep + baby.sleep, data = parenthood)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0345  -2.2198  -0.4016   2.6775  11.7496
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 125.96557    3.04095  41.423  <2e-16 ***
## dan.sleep   -8.95025    0.55346 -16.172  <2e-16 ***
## baby.sleep    0.01052    0.27106   0.039   0.969
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.354 on 97 degrees of freedom
## Multiple R-squared:  0.8161, Adjusted R-squared:  0.8123
## F-statistic: 215.2 on 2 and 97 DF,  p-value: < 2.2e-16
```

The output that this command produces is pretty dense, but we've already discussed everything of interest in it, so what I'll do is go through it line by line. The first line reminds us of what the actual regression model is:

```
Call:
lm(formula = dan.grump ~ dan.sleep + baby.sleep, data = parenthood)
```

You can see why this is handy, since it was a little while back when we actually created the `regression.2` model, and so it's nice to be reminded of what it was we were doing. The next part provides a quick summary of the residuals (i.e., the ϵ_i values),

```
Residuals:
      Min       1Q   Median       3Q      Max
-11.0345  -2.2198  -0.4016   2.6775  11.7496
```

which can be convenient as a quick and dirty check that the model is okay. Remember, we did assume that these residuals were normally distributed, with mean 0. In particular it's worth quickly checking to see if the median is close to zero, and to see if the first quartile is about the same size as the third quartile. If they look badly off, there's a good chance that the assumptions of regression are violated. These ones look pretty nice to me, so let's move on to the interesting stuff. The next part of the R output looks at the coefficients of the regression model:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 125.96557    3.04095  41.423  <2e-16 ***
dan.sleep   -8.95025    0.55346 -16.172  <2e-16 ***
baby.sleep    0.01052    0.27106   0.039   0.969
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Each row in this table refers to one of the coefficients in the regression model. The first row is the intercept term, and the later ones look at each of the predictors. The columns give you all of the relevant information. The first column is the actual estimate of b (e.g., 125.96 for the intercept, and -8.9 for the `dan.sleep` predictor). The second column is the standard error estimate $\hat{\sigma}_b$. The third column gives you the t-statistic, and it's worth noticing that in this table $t = \hat{b} / SE(\hat{b})$ every time. Finally, the fourth

column gives you the actual p value for each of these tests.²¹⁷ The only thing that the table itself doesn't list is the degrees of freedom used in the t-test, which is always $N-K-1$ and is listed immediately below, in this line:

```
Residual standard error: 4.354 on 97 degrees of freedom
```

The value of $df=97$ is equal to $N-K-1$, so that's what we use for our t-tests. In the final part of the output we have the F-test and the R^2 values which assess the performance of the model as a whole

```
Residual standard error: 4.354 on 97 degrees of freedom
Multiple R-squared: 0.8161, Adjusted R-squared: 0.8123
F-statistic: 215.2 on 2 and 97 DF, p-value: < 2.2e-16
```

So in this case, the model performs significantly better than you'd expect by chance ($F(2,97)=215.2$, $p<.001$), which isn't all that surprising: the $R^2=.812$ value indicate that the regression model accounts for 81.2% of the variability in the outcome measure. However, when we look back up at the t-tests for each of the individual coefficients, we have pretty strong evidence that the `baby.sleep` variable has no significant effect; all the work is being done by the `dan.sleep` variable. Taken together, these results suggest that `regression.2` is actually the wrong model for the data: you'd probably be better off dropping the `baby.sleep` predictor entirely. In other words, the `regression.1` model that we started with is the better model.

This page titled [15.5: Hypothesis Tests for Regression Models](#) 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.