■ 3.2 Tests and Intervals

Recall from last time that in order to do hypothesis tests and compute confidence intervals, we need to know our test statistic, its standard error, and its distribution. We'll look at the standard errors for the most important quantities and their interpretation. Any statistical analysis software can compute these quantities automatically, so we'll focus on interpreting and understanding what comes out.

Warning: All the statistical tests here crucially depend on the assumption that the observed data actually comes from the probabilistic model defined in Equation (3.1)!

■ 3.2.1 Slope

For the slope β_1, our test statistic is

$$t_{\beta_1} = \frac{\hat{\beta}_1 - \beta_1}{s_{\beta_1}}, \tag{3.7}$$

which has a Student's t distribution with $n - 2$ degrees of freedom. The standard error of the slope s_{β_1} is

$$s_{\beta_1} = \frac{\hat{\sigma}}{\sqrt{\underbrace{\sum_{i=1}^{n}(x_i - \bar{x})^2}_{\text{how close together } x \text{ values are}}}} \tag{3.8}$$

and the mean squared error $\hat{\sigma}^2$ is

$$\hat{\sigma}^2 = \frac{\overbrace{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}^{\text{how large the errors are}}}{n - 2} \tag{3.9}$$

These terms make intuitive sense: if the x-values are all really close together, it's harder to fit a line. This will also make our standard error s_{β_1} larger, so we'll be less confident about our slope. The standard error also gets larger as the errors grow, as we should expect it to: larger errors should indicate a worse fit.

■ 3.2.2 Intercept

For the intercept β_0, our test statistic is

$$t_{\beta_0} = \frac{\hat{\beta}_0 - \beta_0}{s_{\beta_0}}, \tag{3.10}$$

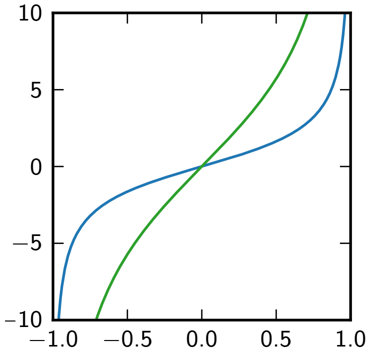


Figure 3.4: The test statistic for the correlation coefficient r for $n = 10$ (blue) and $n = 100$ (green).

which is also t-distributed with $n - 2$ degrees of freedom. The standard error is

$$s_{\beta_0} = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_i (x_i - \bar{x})^2}}, \tag{3.11}$$

and $\hat{\sigma}$ is given by Equation (3.9).

■ 3.2.3 Correlation

For the correlation coefficient r, our test statistic is the standardized correlation

$$t_r = r\sqrt{\frac{n-2}{1-r^2}}, \tag{3.12}$$

which is t-distributed with $n - 2$ degrees of freedom. Figure 3.4 plots t_r against r.

■ 3.2.4 Prediction

Let's look at the prediction at a particular value x^*, which we'll call $\hat{y}(x^*)$. In particular:

$$\hat{y}(x^*) = \hat{\beta}_0 + \hat{\beta}_1 x^*.$$

We can do this even if x^* wasn't in our original dataset.

Let's introduce some notation that will help us distinguish between predicting the line versus predicting a particular point generated from the model. From the probabilistic model given by Equation (3.1), we can similarly write how y is generated for the new point x^*:

$$y(x^*) = \underbrace{\beta_0 + \beta_1 x^*}_{\text{defined as } \mu(x^*)} + \varepsilon, \tag{3.13}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

Then it turns out that the standard error $s_{\hat{\mu}}$ for estimating $\mu(x^*)$ (i.e., the mean of the line at point x^*) using $\hat{y}(x^*)$ is:

$$s_{\hat{\mu}} = \hat{\sigma}\sqrt{\frac{1}{n} + \underbrace{\frac{(x^* - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2}}_{\text{distance from ``comfortable prediction region''}}}.$$

This makes sense because if we're trying to predict for a point that's far from the mean, then we should be less sure, and our prediction should have more variance. To compute the standard error for estimating a particular point $y(x^*)$ and not just its mean $\mu(x^*)$, we'd also need to factor in the extra noise term ε in Equation (3.13):

$$s_{\hat{y}} = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})}{\sum_i (x_i - \bar{x})^2} \underbrace{+1}_{\text{added}}}.$$

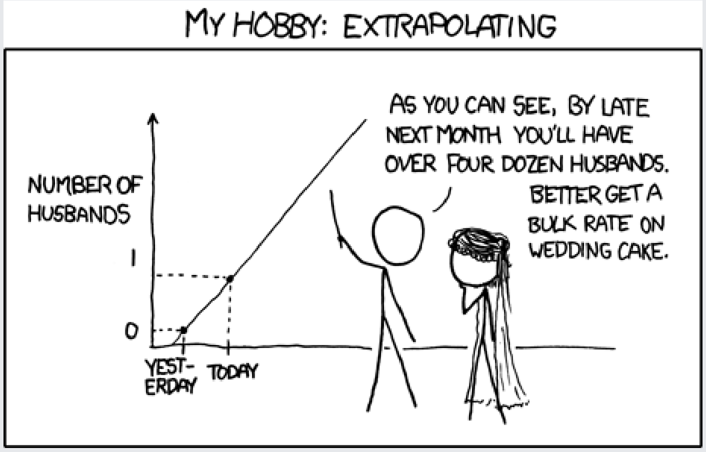
While both of these quantities have the same value when computed from the data, when analyzing them, we have to remember that they're different random variables: $\hat{y}$ has more variation because of the extra ε.

Interpolation vs. extrapolation

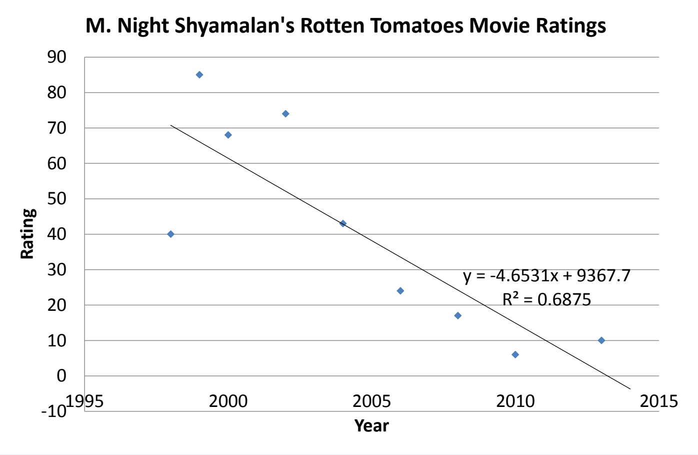
As a reminder, everything here crucially depends on the probabilistic model given by Equation (3.1) being true. In practice, when we do prediction for some value of x we haven't seen before, we need to be very careful. Predicting y for a value of x that is within the interval of points that we saw in the original data (the data that we fit our model with) is called **interpolation**. Predicting y for a value of x that's outside the range of values we actually saw for x in the original data is called **extrapolation**.

For real datasets, even if a linear fit seems appropriate, we need to be extremely careful about extrapolation, which can often lead to false predictions!

Example: The perils of extrapolation


MY HOBBY: EXTRAPOLATING
NUMBER OF HUSBANDS
1
0
YEST-ERDAY
TODAY
AS YOU CAN SEE, BY LATE NEXT MONTH YOU'LL HAVE OVER FOUR DOZEN HUSBANDS. BETTER GET A BULK RATE ON WEDDING CAKE.

By fitting a line to the Rotten Tomatoes ratings for movies that M. Night Shyamalan directed over time, one may erroneously be led to believe that in 2014 and onward, Shyamalan's movies will have negative ratings, which isn't even possible!



3.3 Multiple Linear Regression

Now, let's talk about the case when instead of just a single scalar value x, we have a vector $(x_1, \ldots, x_p)$ for every data point i. So, we have n data points (just like before), each with p different predictor variables or **features**. We'll then try to predict y for each data point as a linear function of the different x variables:

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p. \tag{3.14}$$

Even though it's still linear, this representation is very versatile; here are just a few of the things we can represent with it: