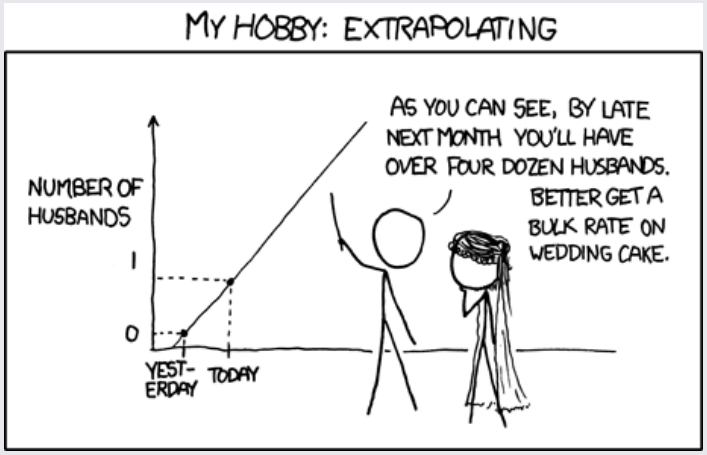
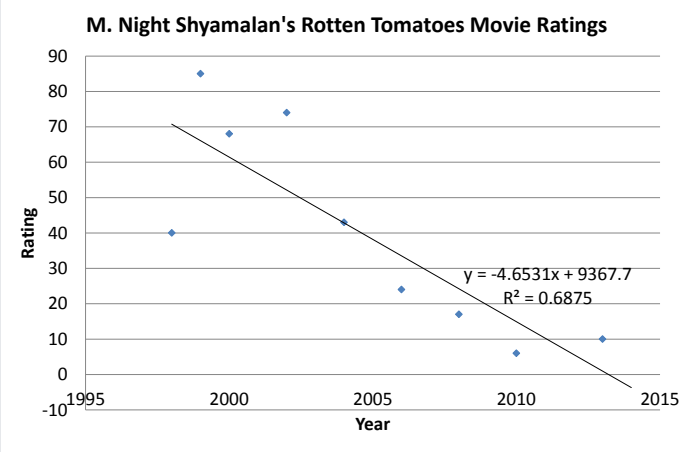
Example: The perils of extrapolation



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:

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

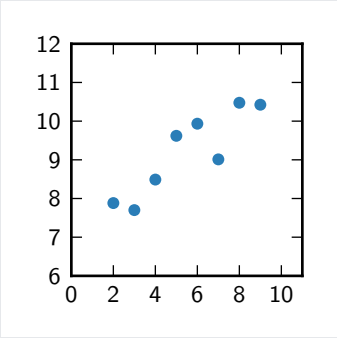
Even though it's still linear, this representation is very versatile; here are just a few of the things we can represent with it:

- Multiple dependent variables: for example, suppose we're trying to predict medical outcome as a function of several variables such as age, genetic susceptibility, and clinical diagnosis. Then we might say that for each patient, $x_1 = \text{age}, x_2 = \text{genetics}$, $x_3 = \text{diagnosis}$, and $y = \text{outcome}$.
- Nonlinearities: Suppose we want to predict a quadratic function $y = ax^2 + bx + c$: then for each data point we might say $x_1 = 1$, $x_2 = x$, and $x_3 = x^2$. This can easily be extended to any nonlinear function we want.

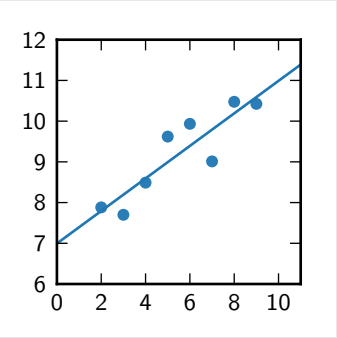
One may ask: why not just use multiple linear regression and fit an extremely high-degree polynomial to our data? While the model then would be much richer, one runs the risk of **overfitting**, where the model is so rich that it ends up fitting to the noise! We illustrate this with an example; it's also illustrated by a song[4].

Example: Overfitting

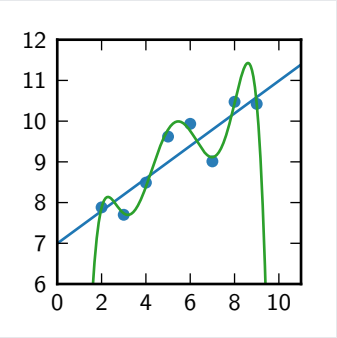
Using too many features or too complex of a model can often lead to overfitting. Suppose we want to fit a model to the points in Figure 3.3(a). If we fit a linear model, it might look like Figure 3.3(b). But, the fit isn't perfect. What if we use our newly acquired multiple regression powers to fit a 6th order polynomial to these points? The result is shown in Figure 3.3(c). While our errors are definitely smaller than they were with the linear model, the new model is far too complex, and will likely go wrong for values too far outside the range.



(a) A set of points with a simple linear relationship.



(b) The same set of points with a linear fit (blue).



(c) The same points with a 6th-order polynomial fit (green). As before, the linear fit is shown in blue.

We'll talk a little more about this in Chapters 4 and 5.

We'll represent our input data in matrix form as X, an $x \times p$ matrix where each row corresponds to a data point and each column corresponds to a feature. Since each output y_i is just a single number, we'll represent the collection as an n-element column vector y. Then our linear model can be expressed as

$$y = X\beta + \varepsilon \tag{3.15}$$

[4]Machine Learning A Cappella, Udacity. https://www.youtube.com/watch?v=DQWI1kvmwRg

where β is a p-element vector of coefficients, and ε is an n-element matrix where each element, like ε_i earlier, is normal with mean 0 and variance σ^2. Notice that in this version, we haven't explicitly written out a constant term like β_0 from before. We'll often add a column of 1s to the matrix X to accomplish this (try multiplying things out and making sure you understand why this solves the problem). The software you use might do this automatically, so it's something worth checking in the documentation.

This leads to the following optimization problem:

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - X_i\beta\right)^2, \tag{3.16}$$

where $\min_\beta$. just means "find values of β that minimize the following", and X_i refers to row i of the matrix X.

We can use some basic linear algebra to solve this problem and find the optimal estimates:

$$\hat{\beta} = (X^T X)^{-1} X^T y, \tag{3.17}$$

which most computer programs will do for you. Once we have this, what conclusions can we make with the help of statistics? We can obtain confidence intervals and/or hypothesis tests for each coefficient, which most statistical software will do for you. The test statistics are very similar to their counterparts for simple linear regression.

It's important not to blindly test whether all the coefficients are greater than zero: since this involves doing multiple comparisons, we'd need to correct appropriately using Bonferroni correction or FDR correction as described in the last chapter. But before even doing that, it's often smarter to measure whether the model even explains a significant amount of the variability in the data: if it doesn't, then it isn't even worth testing any of the coefficients individually. Typically, we'll use an **analysis of variance (ANOVA)** test to measure this. If the ANOVA test determines that the model explains a significant portion of the variability in the data, then we can consider testing each of the hypotheses and correcting for multiple comparisons.

We can also ask about which features have the most effect: if a feature's coefficient is 0 or close to 0, then that feature has little to no impact on the final result. We need to avoid the effect of scale: for example, if one feature is measured in feet and another in inches, even if they're the same, the coefficient for the feet feature will be twelve times larger. In order to avoid this problem, we'll usually look at the standardized coefficients $\frac{\hat{\beta}_k}{s_{\hat{\beta}_k}}$.

■ 3.4 Model Evaluation

How can we measure the performance of our model? Suppose for a moment that every point y_i was very close to the mean $\bar{y}$: this would mean that each y_i wouldn't depend on x_i, and that there wasn't much random error in the value either. Since we expect that this shouldn't be the case, we can try to understand how much the prediction from x_i and random error

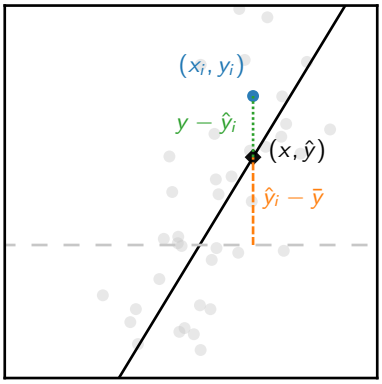


Figure 3.5: An illustration of the components contributing to the difference between the average y-value $\bar{y}$ and a particular point (x_i, y_i) (blue). Some of the difference, $\hat{y}_i - \bar{y}$, can be explained by the model (orange), and the remainder, $y_i - \hat{y}_i$, is known as the residual (green).

contribute to y_i. In particular, let's look at how far y_i is from the mean $\bar{y}$. We'll write this difference as:

$$y_i - \bar{y} = \underbrace{(\hat{y}_i - \bar{y})}_{\text{difference explained by model}} + \underbrace{(y_i - \hat{y}_i)}_{\text{difference not explained by model}} \tag{3.18}$$

In particular, the **residual** is defined to be $y_i - \hat{y}_i$: the distance from the original data point to the predicted value on the line. You can think of it as the error left over after the model has done its work. This difference is shown graphically in Figure 3.5. Note that the residual $y_i - \hat{y}$ isn't quite the same as the **noise** ε! We'll talk a little more about analyzing residuals (and why this distinction matters) in the next chapter.

If our model is doing a good job, then it should explain most of the difference from $\bar{y}$, and the first term should be bigger than the second term. If the second term is much bigger, then the model is probably not as useful.

If we square the quantity on the left, work through some algebra, and use some facts about linear regression, we'll find that

$$\underbrace{\sum_i (y_i - \bar{y})^2}_{\text{SS}_{\text{total}}} = \underbrace{\sum_i (\hat{y}_i - \bar{y})^2}_{\text{SS}_{\text{model}}} + \underbrace{\sum_i (y_i - \hat{y}_i)^2}_{\text{SS}_{\text{error}}}, \tag{3.19}$$

where "SS" stands for "sum of squares". These terms are often abbreviated as SST, SSM, and SSE respectively.

If we divide through by SST, we obtain

$$1 = \underbrace{\frac{\text{SSM}}{\text{SST}}}_{r^2} + \underbrace{\frac{\text{SSE}}{\text{SST}}}_{1-r^2},$$

where we note that r^2 is precisely the coefficient of determination mentioned earlier. Here, we see why r^2 can be interpreted as the fraction of variability in the data that is explained by the model.

One way we might evaluate a model's performance is to compare the ratio SSM/SSE. We'll do this with a slight tweak: we'll instead consider the mean values, $\text{MSM} = \text{SSM}/(p-1)$ and $\text{MSE} = \text{SSE}/(n-p)$, where the denominators correspond to the degrees of freedom. These new variables MSM and MSE have χ^2 distributions, and their ratio

$$f = \frac{\text{MSM}}{\text{MSE}} \tag{3.20}$$

has what's known as an F **distribution** with parameters $p-1$ and $n-p$. The widely used ANOVA test for categorical data, which we'll see in Chapter 6, is based on this F statistic: it's a way of measuring how much of the variability in the data is from the model and how much is from random error, and comparing the two.