Source: http://www.mit.edu/~6.s085/notes/lecture3.pdf

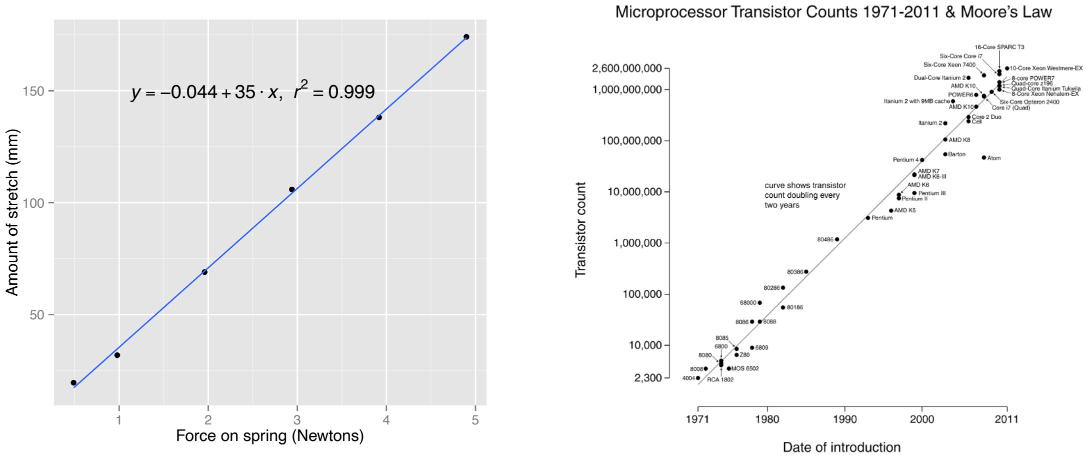
Chapter 3

Linear Regression

Once we've acquired data with multiple variables, one very important question is how the variables are related. For example, we could ask for the relationship between people's weights and heights, or study time and test scores, or two animal populations. **Regression** is a set of techniques for estimating relationships, and we'll focus on them for the next two chapters.

In this chapter, we'll focus on finding one of the simplest type of relationship: linear. This process is unsurprisingly called **linear regression**, and it has many applications. For example, we can relate the force for stretching a spring and the distance that the spring stretches (Hooke's law, shown in Figure 3.1a), or explain how many transistors the semiconductor industry can pack into a circuit over time (Moore's law, shown in Figure 3.1b).

Despite its simplicity, linear regression is an incredibly powerful tool for analyzing data. While we'll focus on the basics in this chapter, the next chapter will show how just a few small tweaks and extensions can enable more complex analyses.



(a) In classical mechanics, one could empirically verify Hooke's law by dangling a mass with a spring and seeing how much the spring is stretched.

(b) In the semiconductor industry, Moore's law is an observation that the number of transistors on an integrated circuit doubles roughly every two years.

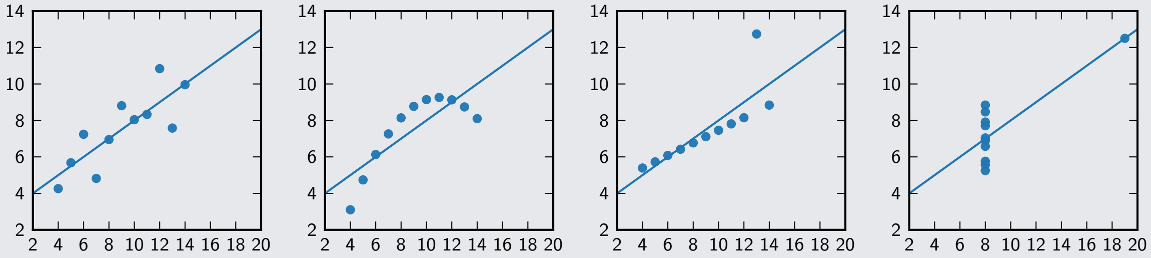
Figure 3.1: Examples of where a line fit explains physical phenomena and engineering feats.[1]

[1] The Moore's law image is by Wgsimon (own work) [CC-BY-SA-3.0 or GFDL], via Wikimedia Commons.

But just because fitting a line is easy doesn't mean that it always makes sense. Let's take another look at Anscombe's quartet to underscore this point.

Example: Anscombe's Quartet Revisited

Recall Anscombe's Quartet: 4 datasets with very similar statistical properties under a simple quantitative analysis, but that look very different. Here they are again, but this time with linear regression lines fitted to each one:



For all 4 of them, the slope of the regression line is 0.500 (to three decimal places) and the intercept is 3.00 (to two decimal places). This just goes to show: visualizing data can often reveal patterns that are hidden by pure numeric analysis!

We begin with **simple linear regression** in which there are only two variables of interest (e.g., weight and height, or force used and distance stretched). After developing intuition for this setting, we'll then turn our attention to **multiple linear regression**, where there are more variables.

Disclaimer: While some of the equations in this chapter might be a little intimidating, it's important to keep in mind that as a user of statistics, the most important thing is to understand their uses and limitations. Toward this end, make sure not to get bogged down in the details of the equations, but instead focus on understanding how they fit in to the big picture.

■ 3.1 Simple linear regression

We're going to fit a line $y = \beta_0 + \beta_1 x$ to our data. Here, x is called the **independent variable** or **predictor variable**, and y is called the **dependent variable** or **response variable**.

Before we talk about how to do the fit, let's take a closer look at the important quantities from the fit:

- β_1 is the slope of the line: this is one of the most important quantities in any linear regression analysis. A value very close to 0 indicates little to no relationship; large positive or negative values indicate large positive or negative relationships, respectively. For our Hooke's law example earlier, the slope is the spring constant[2].

[2] Since the spring constant k is defined as $F = -kx$ (where F is the force and x is the stretch), the slope in Figure 3.1a is actually the inverse of the spring constant.

- β_0 is the intercept of the line.

In order to actually fit a line, we'll start with a way to quantify how good a line is. We'll then use this to fit the "best" line we can.

One way to quantify a line's "goodness" is to propose a probabilistic model that generates data from lines. Then the "best" line is the one for which data generated from the line is "most likely". This is a commonly used technique in statistics: proposing a probabilistic model and using the probability of data to evaluate how good a particular model is. Let's make this more concrete.

A probabilistic model for linearly related data

We observe paired data points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where we assume that as a function of x_i, each y_i is generated by using some true underlying line $y = \beta_0 + \beta_1 x$ that we evaluate at x_i, and then adding some Gaussian noise. Formally,

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i. \tag{3.1}$$

Here, the noise ε_i represents the fact that our data won't fit the model perfectly. We'll model ε_i as being Gaussian: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Note that the intercept β_0, the slope β_1, and the noise variance σ^2 are all treated as fixed (i.e., deterministic) but unknown quantities.

Solving for the fit: least-squares regression

Assuming that this is actually how the data $(x_1, y_1), \ldots, (x_n, y_n)$ we observe are generated, then it turns out that we can find the line for which the probability of the data is highest by solving the following optimization problem[3]:

$$\min_{\beta_0, \beta_1} : \sum_{i=1}^{n} \left[y_i - (\beta_0 + \beta_1 x_i) \right]^2, \tag{3.2}$$

where $\min_{\beta_0,\beta_1}$ means "minimize over β_0 and β_1". This is known as the **least-squares linear regression problem**. Given a set of points, the solution is:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2} \tag{3.3}$$

$$= r \frac{s_y}{s_x}, \tag{3.4}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \tag{3.5}$$

[3] This is an important point: the assumption of Gaussian noise leads to squared error as our minimization criterion. We'll see more regression techniques later that use different distributions and therefore different cost functions.

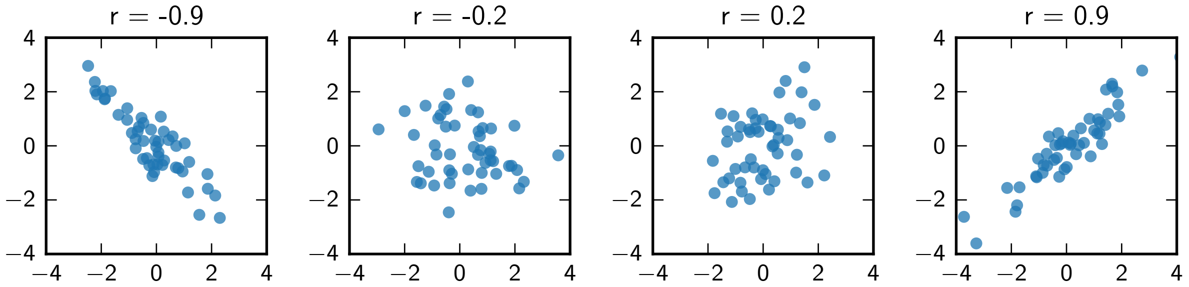


Figure 3.2: An illustration of correlation strength. Each plot shows data with a particular correlation coefficient r. Values farther than 0 (outside) indicate a stronger relationship than values closer to 0 (inside). Negative values (left) indicate an inverse relationship, while positive values (right) indicate a direct relationship.

where $\bar{x}$, $\bar{y}$, s_x and s_y are the sample means and standard deviations for x values and y values, respectively, and r is the **correlation coefficient**, defined as

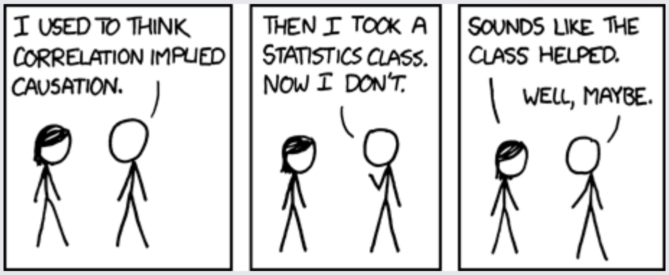
$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right). \tag{3.6}$$

By examining the second equation for the estimated slope $\hat{\beta}_1$, we see that since sample standard deviations s_x and s_y are positive quantities, the correlation coefficient r, which is always between -1 and 1, measures how much x is related to y and whether the trend is positive or negative. Figure 3.2 illustrates different correlation strengths.

The square of the correlation coefficient r^2 will always be positive and is called the **coefficient of determination**. As we'll see later, this also is equal to the proportion of the total variability that's explained by a linear model.

As an extremely crucial remark, correlation does not imply causation! We devote the entire next page to this point, which is one of the most common sources of error in interpreting statistics.

Example: Correlation and Causation


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Just because there's a strong correlation between two variables, there isn't necessarily a causal relationship between them. For example, drowning deaths and ice-cream sales are strongly correlated, but that's because both are affected by the season (summer vs. winter). In general, there are several possible cases, as illustrated below:

$$\begin{tikzcd} x \arrow[r] & y \\ x & y \arrow[l] \end{tikzcd}$$

(a) **Causal link**: Even if there is a causal link between x and y, correlation alone cannot tell us whether y causes x or x causes y.

$$\begin{tikzcd} & z \arrow[dl] \arrow[dr] & \\ x & & y \end{tikzcd}$$

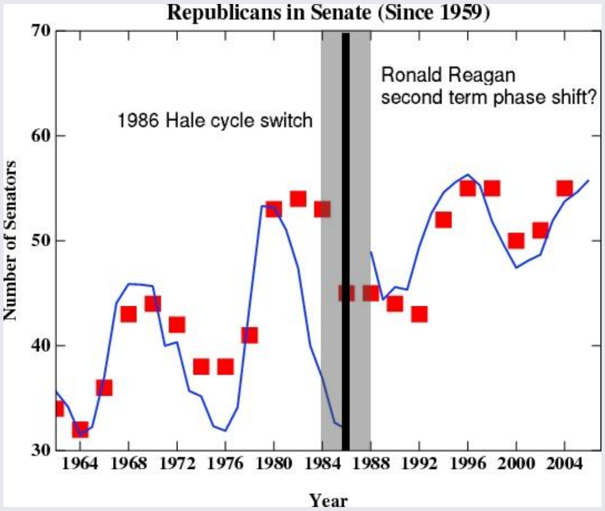
(b) **Hidden Cause**: A hidden variable z causes both x and y, creating the correlation.

$$\begin{tikzcd} & z \arrow[d] \\ x \arrow[r] & y \end{tikzcd}$$

(c) **Confounding Factor**: A hidden variable z and x both affect y, so the results also depend on the value of z.

$$x \quad y$$

(d) **Coincidence**: The correlation just happened by chance (e.g. the strong correlation between sun cycles and number of Republicans in Congress, as shown below).



(e) The number of Republican senators in congress (red) and the sunspot number (blue, before 1986)/inverted sunspot number (blue, after 1986). This figure comes from http://www.realclimate.org/index.php/archives/2007/05/fun-with-correlations/.

Figure 3.3: Different explanations for correlation between two variables. In this diagram, arrows represent causation.

■ 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

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

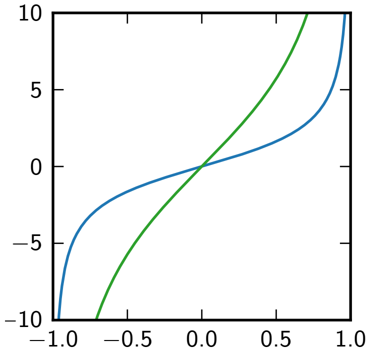


Figure 3.4: The test statistic for the correlation coefficient r for $n = 10$ (blue) and $n = 100$ (green).

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):

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

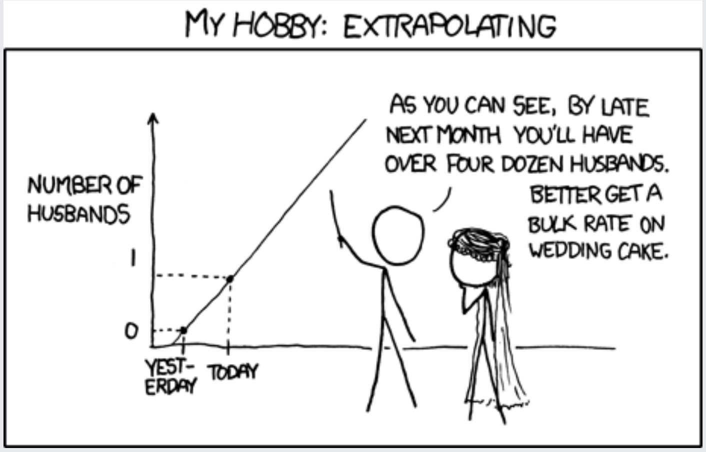
While both of these quantities have the same value when computed from the data, when analyzing them, we have to remember that they're different random variables: $\hat{y}$ has more variation because of the extra ε.

Interpolation vs. extrapolation

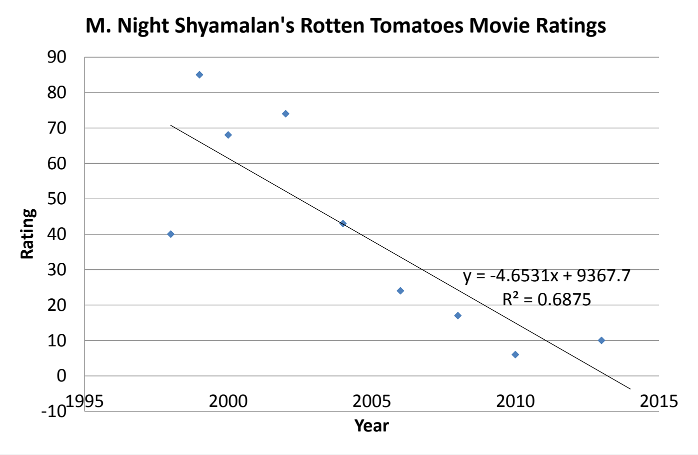
As a reminder, everything here crucially depends on the probabilistic model given by Equation (3.1) being true. In practice, when we do prediction for some value of x we haven't seen before, we need to be very careful. Predicting y for a value of x that is within the interval of points that we saw in 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:

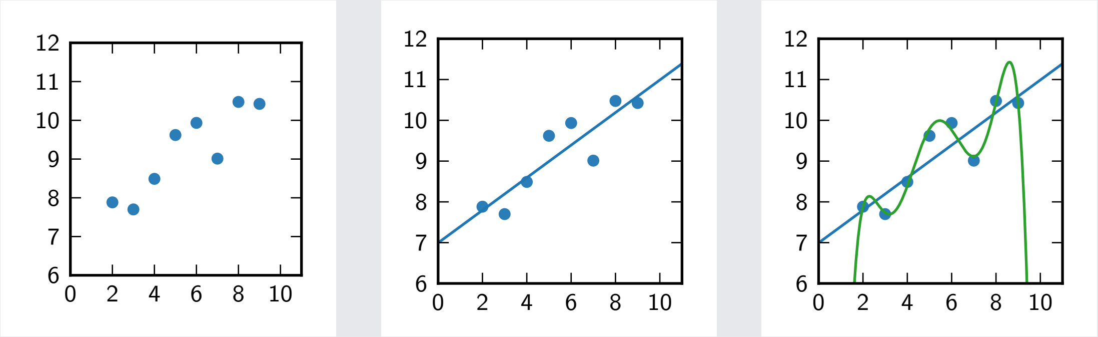
- Multiple dependent variables: for example, suppose we're trying to predict medical outcome as a function of several variables such as age, genetic susceptibility, and clinical diagnosis. Then we might say that for each patient, $x_1 = \text{age}, x_2 = \text{genetics}$, $x_3 = \text{diagnosis}$, and $y = \text{outcome}$.

- 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:

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

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

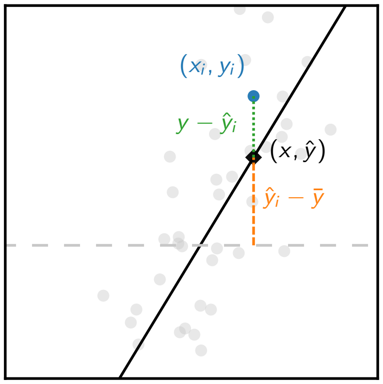


Figure 3.5: An illustration of the components contributing to the difference between the average y-value $\bar{y}$ and a particular point (x_i, y_i) (blue). Some of the difference, $\hat{y}_i - \bar{y}$, can be explained by the model (orange), and the remainder, $y_i - \hat{y}_i$, is known as the residual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