Logistic Regression Basics

Binary response

When the response variable is binary, i.e., taking only 0 or 1 as values, then our regression will try to capture the *probability* that it equals 1. But probabilities have to be between 0 and 1 so they don't work well as linear functions of predictor variables. The solution is to change the linear function, which can take any real number value, into a probability using the **logistic function**:

$$\text{probability} = p = \frac{e^{\text{linear}}}{1 + e^{\text{linear}}}.$$

What to minimize in logistic regression

Suppose that a random number can either be one or zero, with probabilities p and $1 - p$, respectively. We begin with what's called a *probability mass function*, which is a function that gives p when you plug in $y = 1$ and $1 - p$ when you plug in $y = 0$:

$$f(y) = p^y (1-p)^{1-y} = (1-p) \left(\frac{p}{1-p} \right)^y.$$

Next, substitute the expression for p above to get

$$f(y) = \frac{1}{1 + e^{\text{linear}}} \left(e^{\text{linear}} \right)^y.$$

Our goal is to find the maximum value of this expression as a function of "linear". Mathematically, it's easier (and equivalent) to maximize the logarithm of the expression, which is the same as minimizing the negative logarithm:

$$-\log f(y) = \log(1 + e^{\text{linear}}) - y \times \text{linear}.$$

This is why the code we want to minimize, after summing over all the observations, looks like this:

```
np.log(1 + np.exp(linear)) - y * linear
```