HANDS-ON TUTORIALS

Three simple things about regression that every data scientist should know

Understanding these three things will improve how you go about linear and generalized linear modeling

Keith McNulty Jun 10, 2021 · 6 min read ★

I'm more of a mathematician than a data scientist. I can't bring myself to execute methods blindly, with no understanding of what's going on under the hood. I have to get deep into the math to trust the results. That's a good thing because it's very easy nowadays to just run models and go home.

A model is only as good as your understanding of it, and I worry that a lot of people are running models and just accepting the first thing that comes out of them. When it comes to regression modeling — one of the most common forms of modeling out there — you'll be a better data scientist if you can understand a few simple things about how these models work and why they are set up the way they are.

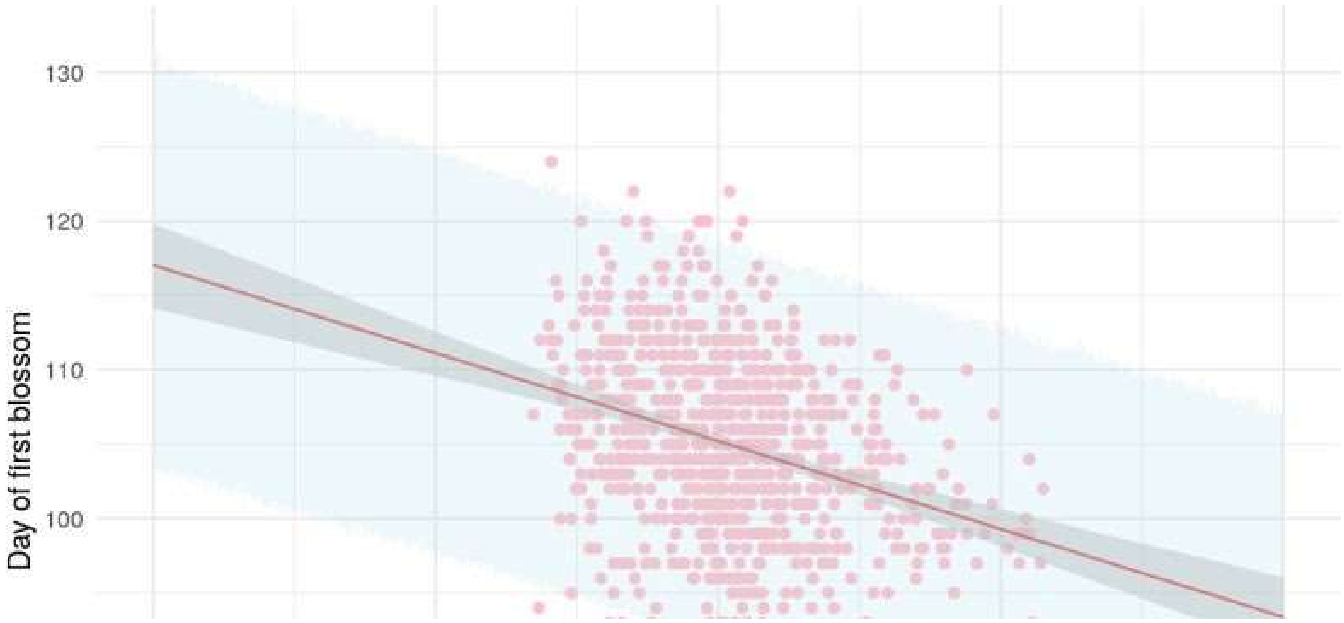
1. You are predicting an average — not an actual value

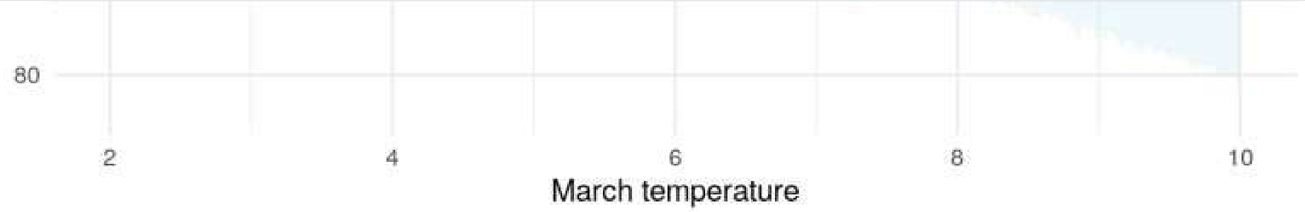
When you run a regression model, usually you are finding a relationship between the input variables and some sort of **mean** value related to the outcome. Let's look at linear

1. That the possible values of *y* for any given input variables are normally distributed around a mean. We expect a bell curve for the values of *y*.
2. That the **mean** of *y* has an *additive* relationship with the input variables. That is, to get the mean of *y* you **add up** some numbers that depend on each input variable.

When you use your model to make predictions, the predicted (or modeled) value of *y* for a given set of input values is an estimate of the **mean** of all the possible values *y* could take. Therefore, in communicating the results of your model, you should always be careful to ensure that this uncertainty is clear.

One way to do this is to use the *prediction interval* which takes into account the expected normal distribution of *y* around the modeled mean. Note that this is different from the *confidence interval* which is often produced by your model — that is simply an interval of uncertainty around your mean value, and so is usually much more narrow than the prediction interval. In the chart below, I show a fitted linear regression for the day of the first cherry blossom bloom in Japan, related to the average temperature in March. The red line represents the modeled mean, the darker grey area represents the confidence around that mean, but the light blue area represents the 95% prediction interval. See how wide that is? It's supposed to be, because it's trying to capture 95% of the possible values of *y*.





Link to R code for this image. All images in this article are author generated

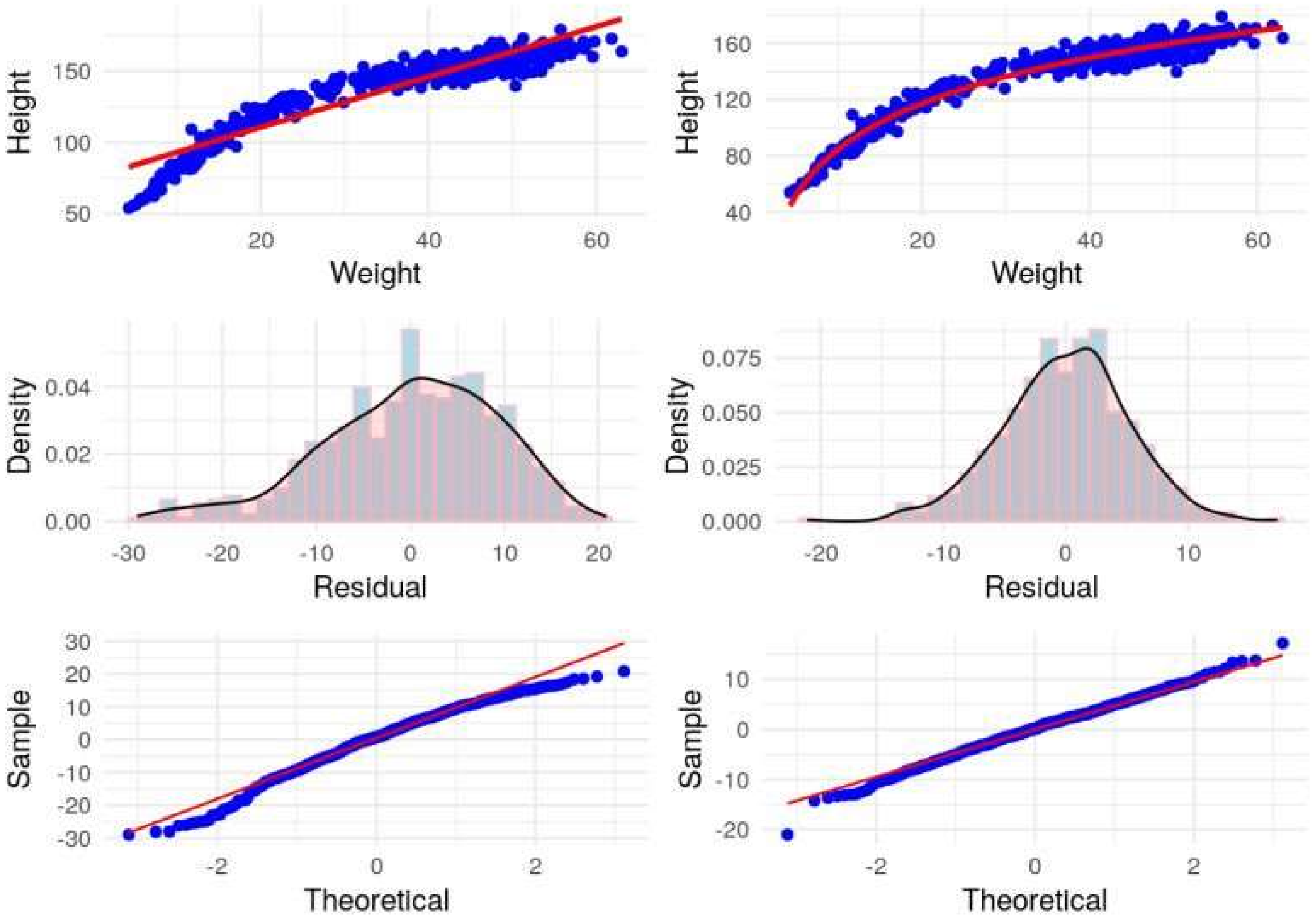
One way to think about the chart above is to consider a vertical 'slice' at any given value of March temperature. This slice can be considered a 'hidden bell curve'. The red dot at the centre represents the expected center of the bell curve, the darker grey area represents the 95% range of uncertainty around the centre of the bell curve, the light blue shaded area represents 95% of the entire bell curve.

2. There is an expectation of a normal distribution for the outcome

This was referenced in the previous point, but the outcome of the model is expected to take a normal distribution around the modeled mean. This means that if you take all the errors — or residuals — of your model, and plot them on a histogram, it should look like a bell curve.

You can use this expectation to actually evaluate how 'good' your model is. The more 'normal' your error distribution looks, the more confident you can be that the mean you have modeled is a 'good' mean. So after you have run a linear model, it's always a good idea to check the distribution of the residuals. You can use a simple histogram or density plot to do this. Or you can use a Quantile-Quantile plot (or QQ Plot). A QQ Plot compares the quantiles of your modeled outcome against the theoretical quantiles of a perfect normal distribution. The more perfect that line is, the more confident you can be that you modeled a 'good' mean.

In the example below, I model a relationship between the height and weight of a group of people, some of which are children. On the left I model a direct linear model between height and weight, and on the right I model a linear relationship between height and the *logarithm* of weight. The top row shows the model fit, the middle row shows the distribution of the residuals and the bottom row shows the corresponding QQ-plot. You



Link to R code for this image

3. When your process is multiplicative you have to make an important transformation to your outcome

The two examples above demonstrated what we typically expect from an *additive* process. What do I mean by this?

In linear modeling, we model our outcome by adding up a bunch of stuff — that 'stuff' is usually certain multiples or transformations of each input variable. Each of those input variables are assumed to be random in nature, and when you create linear combinations of random variables, you expect over a sufficiently repeated sample to see a normal distribution. To illustrate this, I summed 10 random numbers between 0 and 1 10,000 times and the chart below shows the histogram and density plot of the result. See what I mean?

Link to R code for this image

That’s all well and good for additive processes. But things change when your process becomes multiplicative.

When you model *probability*, you are modelling a fundamentally multiplicative process — input variables are believed to have a multiplicative effect on the odds or probability of the outcome.

Now what happens when I select ten random numbers between 1 and 2 and multiply them together? Over 10,000 tests I get this distribution:

Link to R code for this image

Um, Houston, we have a problem. This distribution is not normal, so all our linear regression methods are now out the window.

But wait. Is there a way we can transform our outcome variable to make it the result of an additive process? Remember from high school the rules of indices? Or alternatively stated, the rules of logarithms: log(ab) = log(a) + log(b). So *the logarithm of a multiplicative process is an additive process!*

Let's test this by looking at the distribution of the *logarithm* of the product in our previous chart:

Bingo! So we can model a multiplicative process like probability by predicting the log of our outcome. This means we can treat it just like linear regression except we have to remember to transform it back at the end by exponentiating everything. That's why in logistic regression we model the log-odds, and to get our odds ratios we exponentiate the coefficients, and that's why we often describe the distributions of multiplicative processes as log-normal distributions.

I hope you found these simple but important observations useful to your understanding of the foundations of regression. If you thought this was helpful, you might find my soon to be released book *Handbook of Regression Modeling in People Analytics* a useful resource.

Originally I was a Pure Mathematician, then I became a Psychometrician and a Data Scientist. I am passionate about applying the rigor of all those disciplines to complex people questions. I'm also a coding geek and a massive fan of Japanese RPGs. Find me on LinkedIn or on Twitter. Also check out my blog on drkeithmcnulty.com or my soon to be released textbook on People Analytics.

Data Science Python Data Marketing Hands On Tutorials