Model specification

Katie Schuler

2024-10-08

0.1 Correlation as model building

Correlation is actually a simple case of model building in which we use one value (x) to predict another (y). Specifically, we are fitting the linear model $y = ax + b$, where a and b are free parameters. Here, y is known as the **response variable** (the value we are trying to predict) and x is the **explanatory variable** (the one that we are attempting to *explain* the response variable with).

- After z-scoring our variables, the correlation between x and y is equal to the slope of the line that best predicts y from x.

0.2 Model building overview

Ideally we want to understand statistical modeling beyond the simple case of correlation. What if we have more than one explanatory variable? What if the relationship between variables is not linear? To address model building more broadly, it is helpful to think of building any model as a four-step process. We'll treat each of these separately over the coming weeks. The goal for today is to get a big picture overview of the model building process and the types of models we might encounter in our research.

- **Model specification** (this week): what is the form?
- **Model fitting** (this week); you have the form, how do you guess the free parameters?
- **Model accuracy** (after break): you've estimated the parameters, how well does that model describe your data?
- **Model reliability** (after break): when you estimate the parameters, there is some uncertainty on them

0.3 Types of models

Model specification involves deciding which type of model we'd like to apply. We will mostly apply linear models in this class, but it's useful to first have a conceptual overview of the types of models we *could* apply.

0.3.1 Supervised vs unsupervised

As a starting point, we can divide statistical models into two types of learning (so called because we are trying to "learn" something about the data):

- In **supervised learning**, we want to predict an output (or response) variable based on one or more input (or explanatory) variables. We call this supervised learning because both the input and output variables are known (sometimes this is called "labeled" data), and we are trying to learn the relationship between them. Linear regression is an example of a supervised learning model.
- In **unsupervised learning**, there is no specific output variable that we are trying to predict. Instead, the model's objective is to discover the underlying structure or patterns in the data. We call this unsupervised learning because only the input data is available (sometimes this is called "unlabeled" data); the model is trying to identify relationships in the data without being "supervised" by an outcome variable. PCA and cluster analysis are examples of unsupervised learning.
- There are other machine learning approaches beyond these, like semi-supervised learning (combining both labeled and unlabeled data) and reinforcement learning (learning through trial and error based on rewards or penalties). But in this course we will focus on supervised learning models.

0.3.2 Regression v classification

Regression and classification are both types of supervised learning models — using one or more input variables to predict an output variable. The only difference between them is in type of output variable:

- **Regression** is used when we want to predict a continuous output, meaning it is a number that can take on any value within a range (e.g. height, weight, response time)
- **Classification** is used when we want to predict a categorical output, meaning it falls into specific classes or categories (e.g. true/false, yes/no, male/female/nonbinary). We cover this during advanced model building.

0.3.3 Linear v nonlinear regression

There are many types of regression models, but we can simplify by dividing them into two main types of models:

- In **linear regression**, the relationship between the explanatory variable(s) and response variable is represented by a linear equation (a straight line graphed on a two-dimensional plane).
- **Nonlinear regression** is useful when the data does not follow a linear pattern, and the relationship between the variables is better captured by more complex functions (e.g. a curve or any other nonlinear shape). Nonlinear regression models can be further divided into two types:
  - We can **linearize** a nonlinear model by applying a mathematical transformation to make it look like a linear equation (e.g. log, square root, etc). We can fit a linearized model just like a linear model, but the prediction of the model is not linear with respect to x.
  - Sometimes, no matter what transformation you apply, you cannot acheive a linear form. These types of models are referred to as **nonlinearizable** nonlinear models and are beyond the scope of this class!

0.4 Model specification

Recall from last week that **model specification** is one aspect of the model building process. It involves selecting the functional form of the model (the type of model) and choosing which variables to include. When specifying a model, you'll need to make the following decisions:

1. **Response variable (y):** Choose the variable you want to predict or explain (output).
2. **Explanatory variables(x_n):** Choose the variables that may explain the variation in the response variable (inputs).
3. **Functional form:** Specify the functional relationship between the response and explanatory variables. For linear models, the relationship is linear, and we use the linear model equation as our functional form!
4. **Model terms:** Choose which model terms to include, which is another way of saying that you need to decide *how* to include your explanatory variables in the model (since they can be included in more than one way).

In addition to the decisions above, the following issues can also be considered part of the model specification process. But we will consider these in future weeks.

- **Model assumptions:** Check any assumptions underlying the model you selected (e.g. does the model assume the relationship is linear?).

- **Model complexity:** Simple models are easier to interpret but may not capture all complexities in the data. Complex models may suffer from overfitting the data or being difficult to interpret.

A well-specified model should be based on a clear understanding of the data, the underlying relationships, and the research question.

0.4.1 Response variable, y

Choosing the response variable is usually straightforward once you've clearly defined your research question: what is the thing you are trying to understand? You also need to make sure whatever you've selected is something you can measure (or has already been measured!)

- In the swim records example, we are trying to explain variation in record times, so we choose record time as our y. In the brain size example we are trying to explain variation in brain sizes, so we choose brain size as our y.
- Remember from last time that we can use regression for continuous response variables (numbers) but we need to use classification if the response variable is categorical (categories or levels).

0.4.2 Explanatory variables, x_n

Choosing which explanatory variables to include requires a bit more careful consideration. It's one part using your knowledge about the domain you are studying and one part exploratory data analysis!

- One extreme would be to include just one explanatory variable: the obvious one based on your research question. In the swim records example, we want to understand how swim records change over time, so we should definitely include `time` as an explanatory variable. But this model is **underspecified**. We need to consider other variables that have the *potential* to explain variation in our response variable, even if they are not of direct interest. For example, some variation in swim record times can likely be explained by the swimmer's gender, so we should include `gender` as an explanatory variable in our model.
- Importantly, we do not include every explanatory variable we can think of! We want to explain the variation in our response variable without building too complex a model or overfitting the data (**overspecifying**). We'll go into more detail about this in future lectures. For now, just remember you're Goldilocks: you want the explanatory variables in the model to explain just the right amount of variation.

0.4.3 Functional form

Last lecture we introduced the types of models that we could select when specifying a model. We also mentioned that we will focus on linear models in this class (which are a type of regression model, which are themselves types of supervised learning models!)

When specifying the functional form of a model, we're literally specifying the mathematical formula we're going to use to represent the relationship between our response and explanatory variables. **Linear models** are models in which the response variable (output) is a weighted sum of the explanatory variables (inputs). In other words, there is a linear relationship between the response variable and the explanatory variables. The **linear model equation** can be expressed in many ways (which can be confusing!). Here are four different ways of representing the formula for the linear model, to emphasize that *they are all the same thing*.

1. In **high school algebra**, the linear model equation is represented as the equation of a straight line. y is the response variable, x is the explanatory variable, a is the slope of the line (the relationship between x and y) and b is the y-intercept (the value of y when x is zero). You have (hopefully!) already encountered this equation.

- $y = ax + b$.

2. In **machine learning**, the linear model equation is usually represented as a weighted sum of input variables. Note that the only changes are that we refer to the free parameters as weights (w_n) instead of a and b (to emphasize these are the weights the model learns) and the ability to add more than one input ($x_1, x_2, ...x_n$ instead of just x):

- $y = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n$

3. In **statistics**, the linear model equation is also represented as a weighted sum of input variables, except we call the weights "regression coefficients" (β_n) and we add an error term to account for unexplained variability (ε):

- $y = \beta_0 + \beta_1x_1 + \beta_2x_2 + ... + \beta_nx_n + \varepsilon$

4. In **matrix** notation, the linear model equation is represented as a dot product of vectors. This is just a more compact representation of the statistics (or machine learning) way, often used in linear algebra and statistics. X is the matrix containing the values of the explanatory variables, β is the vector of regression coefficients, and ε is the vector of error terms.

- $y = X\beta + \varepsilon$

0.4.4 Model terms

We've specified our response and explanatory variables and the functional form of our model. Now we need to specify the model terms. Model terms describe *how* to include the explanatory variables in our model (they can be included in more than one way!) There are four kinds of terms: (1) **intercept**, (2) **main**, (3) **interaction**, and (4) **transformation**.

1. The **intercept** term, β_0, is a constant (not variable) capturing the typical value of the response variable when all explanatory variables are zero. It allows the model to have an offset from the origin, so it is also called the "offset" parameter in some fields. Unless it makes sense for our response variable to be zero when all other variables are zero (it rarely does!) we should include the intercept term.

- in R: `y ~ 1`
- in eq: $y = \beta_0 + \varepsilon$

2. **Main** terms (AKA **main effects**) represent the effect of each explanatory variable on the response variable directly. In other words, how does the response variable change as a result of changes in a given explanatory variable, when all other explanatory variables are zero? Each main term corresponds to one explanatory variable and is included in the model as a single term ($\beta_n x_n$). We can add as many explanatory variables as we like to the model:

- in R: `y ~ 1 + year + gender`
- in eq: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
- Note that can include **categorical explanatory variables** like gender, we just need to find a way to represent the same information numerically, since linear models require numerical inputs.

3. **Interaction** terms allow us to express that the effect of one explanatory variable on the response variable is different at different values of another explanatory variable. For example, in the swim records data, the effect of gender on record times changes over year (or said another way, the effect of year on record times is different for men and women). We are still describing how variation in the response variable is explained by one or more explanatory variables, we're just describing how two (or more) variables *combine* to influence the response. In the linear model equation, we add a term to the model in which we multiply the values of the interacting variables.

- in R: `y ~ 1 + year + gender + year:gender`
  - or the short way: `y ~ 1 + year * gender`
- in eq: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$

4. **Transformation** terms allow us to modify the explanatory variables to accommodate nonlinear relationships with the response variable. Some of the most common transformations are x^2, $\sqrt{x}$, and $log(x)$. Note that x must be a quantitative variable: we can't transform categorical variables. In the swim records example, squaring the year term (x_1^2) allowed our model to have a curve shape. But notice that this makes it seem like record times are slowing down after 1990. This is obviously not the case — records inherently only get faster! — but models lack common sense, and there is no easy math way to tell our model to "be curvy, but also never slope upward".

- in R: `y ~ sq(year) * gender`
- eq: $y = \beta_0 + \beta_1 x_1^2 + \beta_2 x_2 + \beta_3 x_1^2 x_2$

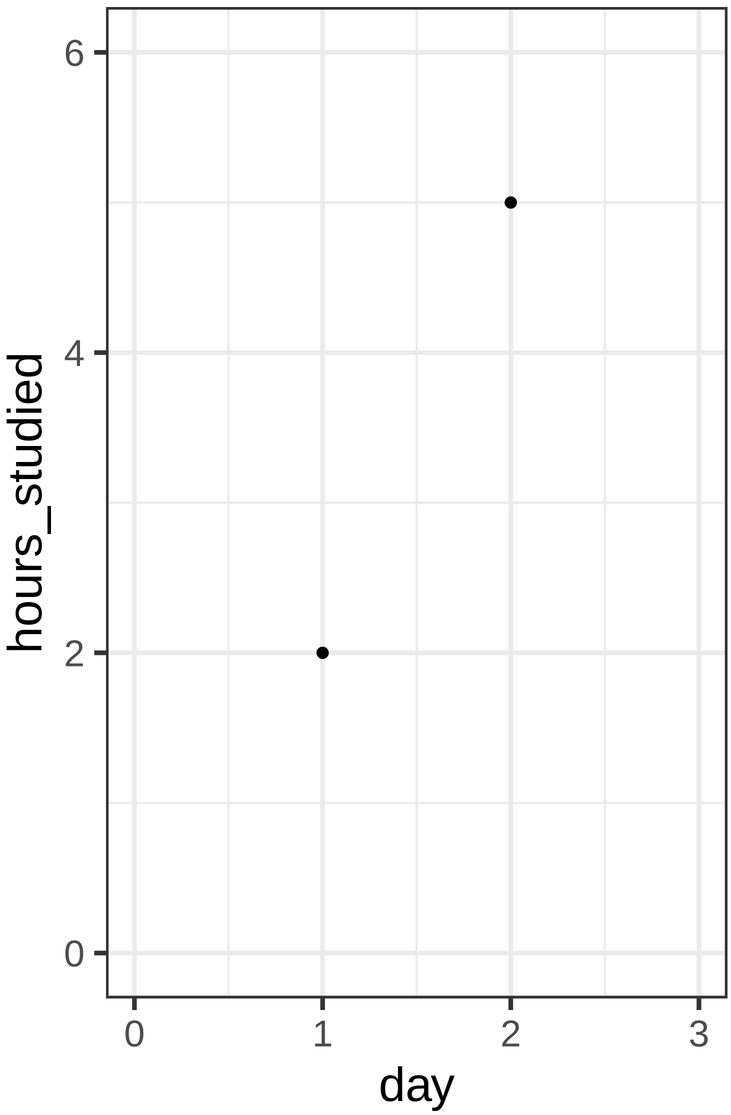
0.5 Model specification practice

```
library(tidyverse)
```

0.5.1 Simplest model

Let's start with the simplest possible example, a dataset with two data points. Suppose you record how many days you study over the next two days. On day 1, you study for 2 hours. On day 2, you study for 3 hours. Your dataset might look something like this.

0.6 Plot



0.7 Data

day	hours_studied
1	2
2	5

0.8 Code

```
# create the dataset
study_data <- tibble(
  day = c(1, 2),
  hours_studied = c(2, 5)
)

# make the plot
study_data %>%
  ggplot(aes(
    x = day,
    y = hours_studied
  )) +
  geom_point() +
  theme_bw(base_size = 25) +
  xlim(c(0,3)) +
  ylim(c(0,6))
```

0.9 Further reading

- Ch 6: Language of models in Statistical Modeling