# 3.1: Why Programming Is Hard to Learn

Programming a computer is a skill, just like playing a musical instrument or speaking a second language. And just like those skills, it takes a lot of work to get good at it — the only way to acquire a skill is through practice. There is nothing special or magical about people who are experts, other than the quality and quantity of their experience! However, not all practice is equally effective. A large amount of psychological research has shown that practice needs to be *deliberate*, meaning that it focuses on developing the specific skills that one needs to perform the skill, at a level that is always pushing one's ability.

If you have never programmed before, then it's going to seem hard, just as it would seem hard for a native English speaker to start speaking Mandarin. However, just as a beginning guitarist needs to learn to play their scales, we will teach you how to perform the basics of programming, which you can then use to do more powerful things.

One of the most important aspects of computer programming is that you can try things to your heart's content; the worst thing that can happen is that the program will crash. Trying new things and making mistakes is one of the keys to learning.

The hardest part of programming is figuring out why something didn't work, which we call *debugging*. In programming, things are going to go wrong in ways that are often confusing and opaque. Every programmer has a story about spending hours trying to figure out why something didn't work, only to realize that the problem was completely obvious. The more practice you get, the better you will get at figuring out how to fix these errors. But there are a few strategies that can be helpful.

## 3.1.1 Use the web

In particular, you should take advantage of the fact that there are millions of people programming in R around the world, so nearly any error message you see has already been seen by someone else. Whenever I experience an error that I don't understand, the first thing that I do is to copy and paste the error message into a search engine Often this will provide several pages discussing the problem and the ways that people have solved it.

## 3.1.2 Rubber duck debugging

The idea behind *rubber duck debugging* is to pretend that you are trying to explain what your code is doing to an inanimate object, like a rubber duck. Often, the process of explaning it aloud is enough to help you find the problem.