2.1.1 Install Developer Tools

Steve is somewhat savvy in Excel, but he's asked you to help him with his analysis. This is a great opportunity to use VBA. But first, you'll need to install the tools necessary to access it in Excel.

One way to perform this data analysis would be to go through all of Steve's stock data manually and use Excel formulas for calculations. But with Visual Basic for Applications, which is typically referred to as "VBA," we can write code that will automate these analyses for us. Often used in the finance industry, VBA provides essentially infinite extensibility to Excel. Using code to automate tasks decreases the chance of errors and reduces the time needed to run analyses, especially if they need to be done repeatedly.

NOTE

> BASIC (short for Beginner's All-purpose Symbolic Instruction Code) was a programming language invented in the 1960s to help teach programming concepts. It soon gained traction and started to be used as a full-fledged programming language. In the 1990s, Microsoft

> created a version of BASIC with a visual form builder so that graphical desktop applications could be built, and Visual Basic was born! It lives on today in VBA and VB.NET.

By learning the ins and outs of Excel, you've dipped your toes into the waters of programming; with VBA, you're going to dive all the way in as you learn how to create your own functions and automate tasks in Excel. Adding VBA to Excel is like adding a superpower; but like any great power, it comes with great responsibility.

In developer parlance, automated tasks are called **macros**. Originally, macros were created by "recording" a task that you performed in Excel, and VBA would automatically generate code to repeat the task. A task could be something as simple as deleting a comma at the end of a cell's value and then moving to the next cell. The macro could be run repeatedly to quickly perform the task over and over again.

While recording a task seems like it might be a good way to create macros, the reality is that it doesn't work as well as one would hope, and the code is difficult to understand and edit. It's more efficient to write the code from scratch.

Almost all VBA code is written to create macros, which are sometimes called **subroutines**. However, VBA is powerful enough to connect to the internet and run applications in the operating system, which means it can be abused to write malicious code, like viruses and trojan horses. Because of this, VBA is included with Excel, but access to it is disabled by default. To enable VBA in Excel, we need to add the Developer tab to our ribbon.

How you enable VBA depends on whether you are a Windows or macOS user. Watch the video below for step-by-step instructions for your operating system.

---

macOS

Windows

GITHUB

Put your new GitHub skills to the test by creating a new repository for this project. Name the repository "stock-analysis." We'll use this repository to back

up our work. As you work through this module, remember to commit your work often! Reminders to "add, commit, push" are provided periodically at natural stopping points.