# Build Quick Start Challenge for R Tools 1.0 for Visual Studio 2017

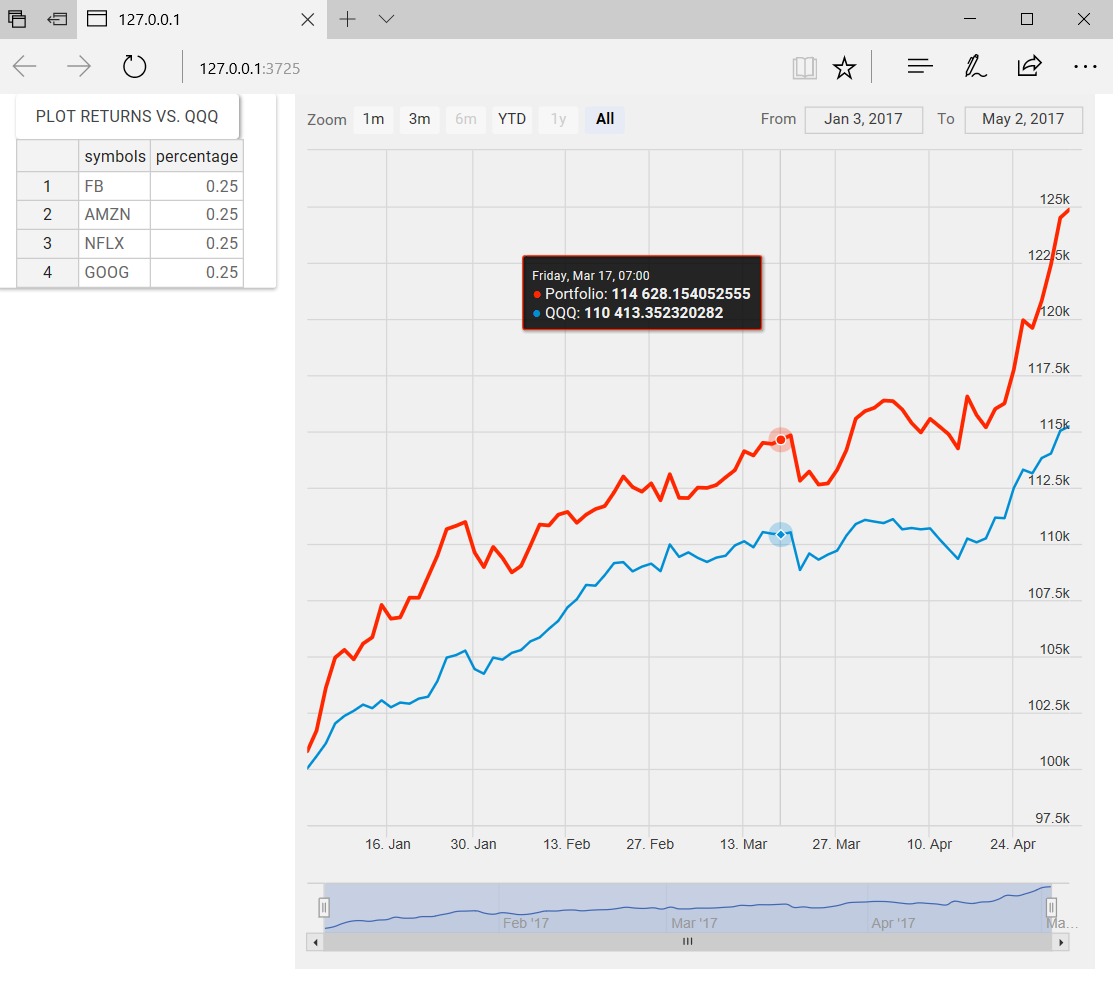
John Lam

## Your Challenge (If You Accept It!)

You will use RTVS to build a R Shiny application that lets you enter your own portfolio of stocks so that you can compare the performance of $100,000 invested in your portfolio vs. the NASDAQ 100 index that is tracked by the QQQ ETF. This is what the application looks like when you’re done. You can:

* Change the percentage of your investments (e.g, you can invest more in FB and less in NFLX)
* You can add/remove different stock ticker symbols
* You can interactively inspect the plot and zoom in on desired regions of interest

The entire application is built using R. No JavaScript or HTML required at all. Along the way, you will have a tour of some of the coolest features in RTVS in Visual Studio 2017. So let’s start!

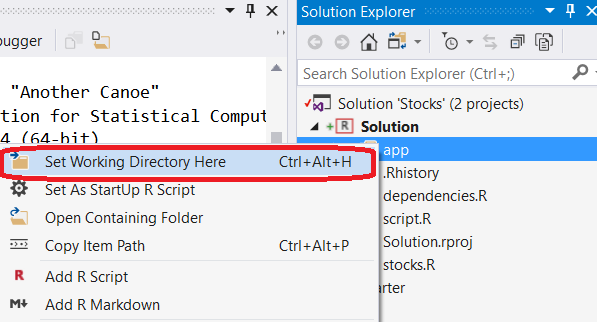


## Getting Started: Running the Solution

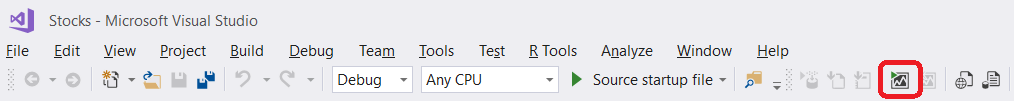
On your desktop, you’ll find a shortcut called Stocks. Double-click on that shortcut to open the lab solution in Visual Studio 2017. The solution is organized into two projects: **Solution** and **Starter**. The **Solution** contains the fully functional solution to the lab. If you get stuck, run out of time, or don’t believe the app actually works, you can always just run the project in the **Solution** directory.

Here is how you run a Shiny app:

1. Set the working directory to the **app** subdirectory of the **Solution** project by right-clicking on the **app** icon in the solution, and running the **Set Working Directory Here** command.



1. Click on the Run Shiny App toolbar button to start the stocks app. You should see the default web browser launch with the app up and running.



Next, try clicking on the PLOT RETURNS VS. QQQ button. You should see a plot that resembles the image at the top of this lab manual. Try editing some of the values, or changing the names of some of the stock symbols. Note that error handling is an exercise left to the reader!

You can also add additional columns to the grid by right clicking on the grid and inserting additional rows above or below the cursor. Try creating a new row, or removing a row. You’ll need to click on the PLOT RETURNS VS. QQQ button to see changes in the plot. See if you can beat the QQQ index with your own portfolio.

## Building your own app

The Stocks app uses several different R packages. These dependencies are all gathered together for you in the **dependencies.R** file. Open the file up in Visual Studio by double-clicking on it. You’ll see that we depend on four different packages:

* [shiny](http://shiny.rstudio.com/gallery/): package for building web applications in R
* [quantmod](http://www.quantmod.com/): package for working with financial time series data
* [highcharter](http://jkunst.com/highcharter/showcase.html): package for high quality HTML/CSS/JavaScript plots
* [rhandsontable](http://jrowen.github.io/rhandsontable/): package for editable, interactive HTML grid

The code in the **dependencies.R** file uses a helper function **install\_if\_not\_present** to check to see if a package was installed already before attempting to install it.

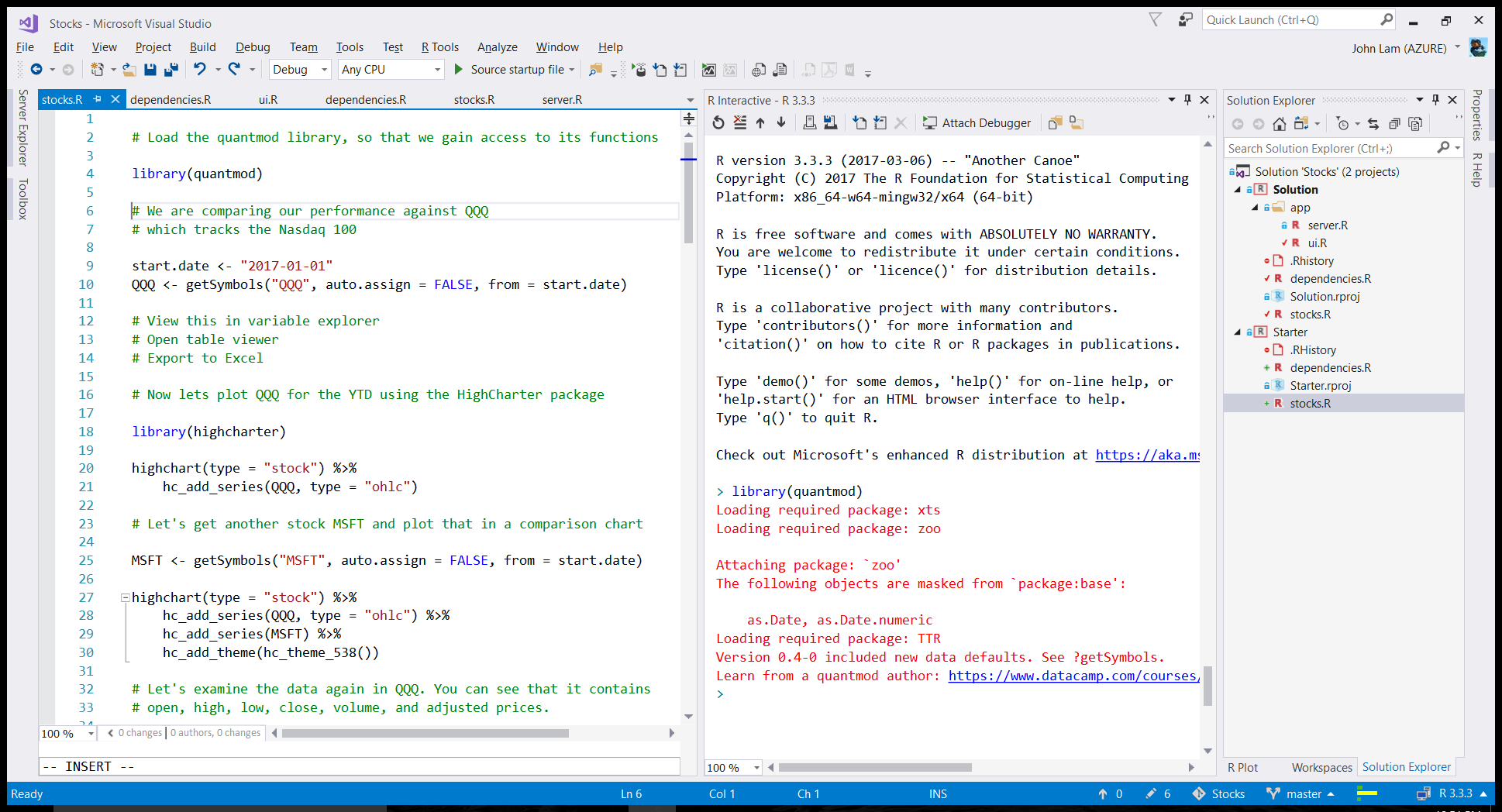
Right click on the file in Solution Explorer and **source** it. The source command instructs the R interpreter to execute the contents of the file.

## Interactive Execution

Open the **stocks.R** file where we’ll learn about the R programming language while interactively executing the code in the file. Below is a picture of the way that I like to arrange my windows on a 1920 x 1080 monitor. I have, on the left hand side, the **stocks.R** file opened in the editor. To the right of it, I have the R Interactive Window (sometimes called the REPL – Read-Eval-Print-Loop window) open.

I find this arrangement of windows especially useful for interactively executing code. While writing the lab, I would write the code that I want to write in the editor window on the left, and send the lines of code that I wanted to run to the REPL window on the right. There are two different ways that you can interactively execute some code in the REPL:

1. Select a block of text in the editor and press CTRL+ENTER to send the selection to the REPL window and executed.
2. Press CTRL+ENTER without a selection, and the current line under the caret will be sent to the REPL window and executed.



Place your cursor anywhere on the line that contains the **library(quantmod)** command and press CTRL+ENTER. You should see that command copied to the REPL window and run, producing the red lines of output that you see in the image above.

Follow along with the explanations in the **script.R** file to work through an example of how to download stock data from the Internet using functions in the **quantmod** library, and writing code to compute returns on a hypothetical $100,000 investment in a portfolio of stocks. You will get to see another key library at work as well, **highcharter**, which is a library that produces very high-quality charts in your default web browser.

If you’re impatient and just want to see what happens you could either:

1. Source the **stocks.R** file, which will run all of the code (and produce 4 plots)
2. Select all the lines of code in the file via CTRL+A, and then hitting CTRL+ENTER to execute all of those lines in the REPL window.

However, I strongly advise that you run the code line-by-line and read the comments to get the most out of your experience.