Module 2 VBA\_Challenge README

Author: Monica Holmes

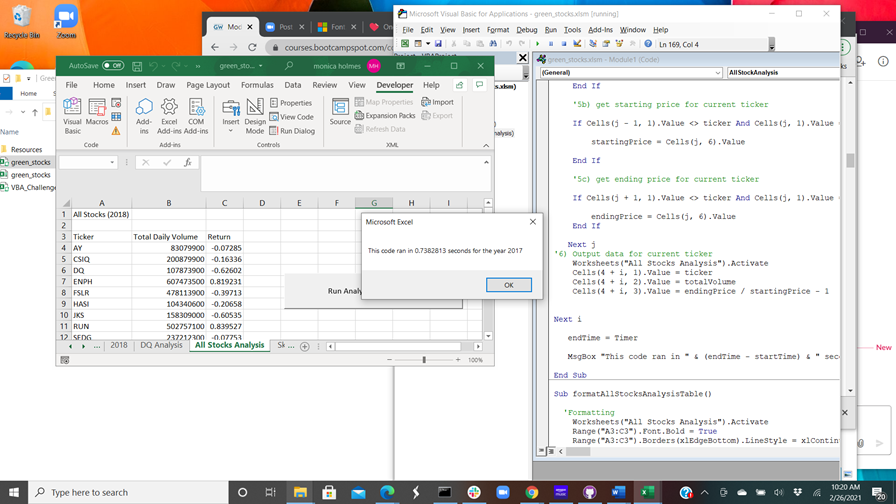
Date: March 1, 2021

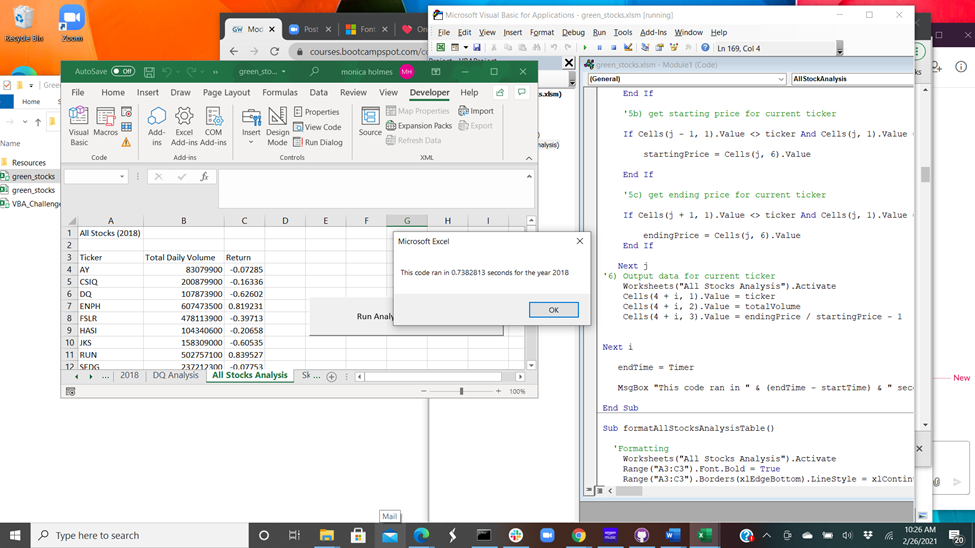
* **Overview of Project:**

The Purpose of this analysis was to utilize Visual Basic for Applications (VBA) to write and refactor macro code that will automate analyses of Steve’s stock data and then calculate and measure the codes performance. I wrote the code then utilized refactoring to write a faster code.

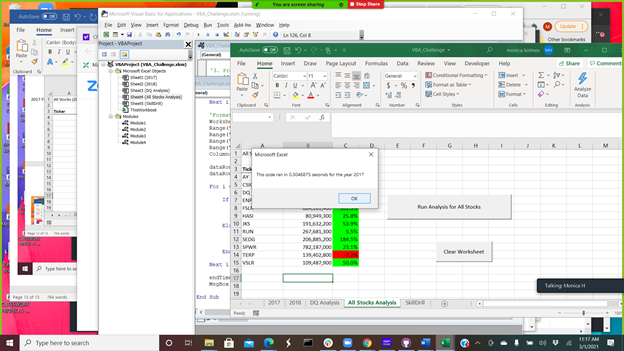
* **Results:**

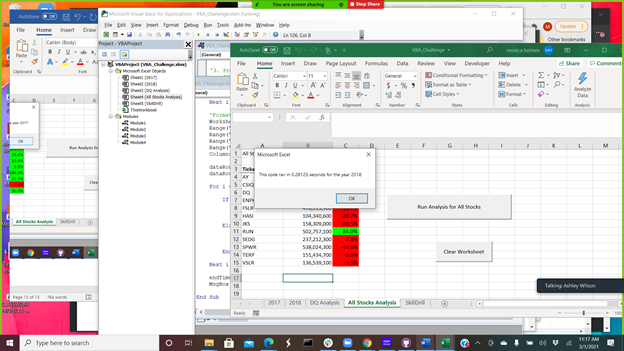
The results show that when I first ran the macro code (the original results of Green Stock data) it took the system 0.7382813 and 0.7382813 seconds to run 2017 and 2018 data, respectively (see screen prints below):





After refactoring the code in VBA Challenge, the run times for 2017 and 2018 were 0.3046875 and 0.28125, respectively (see screen prints below):





* **Summary:**

In summary, the advantages of refactoring code are, the run times tend to be faster, and you may copy and paste some of the code then edit it or simply craft a shorter code. The disadvantages of refactoring are, you may copy the wrong original code, paste the wrong code or to the wrong location, or receive an error because you need to edit the code or create an array versus a variable.

I experienced some of the above pros and cons when I applied the various methods of refactoring to the original VBA script. In summary, it was easy to mix up the data while going back and forth between the two modules. Also, when refactoring you needed to verify the results you want were identical or the necessary edits were made to the codes. Lastly, it was surprisingly refreshing to see how condensed and simplistic I was able to make the code, ultimately, speed up the codes run time.