Student Name: Mano Velu

Batch: Nov-12-2019 to May-09-2020

Project: VBA-Challenge

Due Date: 12-07-2019

**Table of Contents**

[1. VBA-Challenge ReadMe 3](#_Toc26581253)

[2. Input files 5](#_Toc26581254)

[3. Design criteria and Assumptions 5](#_Toc26581255)

[4. Metrics Definition 5](#_Toc26581256)

[5. Pseudo Code 6](#_Toc26581257)

[6. VBA Scripts 6](#_Toc26581258)

[7. Code Validation & Testing 7](#_Toc26581259)

[8. Conclusion 7](#_Toc26581260)

[9. Screen shots: 8](#_Toc26581261)

# VBA-Challenge ReadMe

1. **# VBA Homework - The VBA of Wall Street**
2. **## Background**
3. You are well on your way to becoming a programmer and Excel master! In this homework assignment you will use VBA scripting to analyze real stock market data. Depending on your comfort level with VBA, you may choose to challenge yourself with a few of the challenge tasks.
4. **### Before You Begin**
5. 1. Create a new repository for this project called `VBA-challenge`. **\*\*Do not add this homework to an existing repository\*\***.
6. 2. Clone the new repository to your computer.
7. 3. Inside your local git repository, create a directory for both of the VBA Challenges. Use the folder name to correspond to the challenge: **\*\*VBAStocks\*\***.
8. 4. Inside the folder that you just created, add any VBA files. Theses will be the main scripts to run for each analysis.
9. 5. Push the above changes to GitHub or GitLab.
10. **### Files**
11. \* [Test Data](Resources/alphabetical\_testing.xlsx) - Use this while developing your scripts.
12. \* [Stock Data](Resources/Multiple\_year\_stock\_data.xlsx) - Run your scripts on this data to generate the final homework report.
13. **### Stock market analyst**
14. ![stock Market](Images/stockmarket.jpg)
15. **## Instructions**
16. \* Create a script that will loop through all the stocks for one year for each run and take the following information.
17. \* The ticker symbol.
18. \* Yearly change from opening price at the beginning of a given year to the closing price at the end of that year.
19. \* The percent change from opening price at the beginning of a given year to the closing price at the end of that year.
20. \* The total stock volume of the stock.
21. \* You should also have conditional formatting that will highlight positive change in green and negative change in red.
22. \* The result should look as follows.
23. ![moderate\_solution](Images/moderate\_solution.png)
24. **### CHALLENGES**
25. 1. Your solution will also be able to return the stock with the "Greatest % increase", "Greatest % Decrease" and "Greatest total volume". The solution will look as follows:
26. ![hard\_solution](Images/hard\_solution.png)
27. 2. Make the appropriate adjustments to your VBA script that will allow it to run on every worksheet, i.e., every year, just by running the VBA script once.
28. **### Other Considerations**
29. \* Use the sheet `alphabetical\_testing.xlsx` while developing your code. This data set is smaller and will allow you to test faster. Your code should run on this file in less than 3-5 minutes.
30. \* Make sure that the script acts the same on each sheet. The joy of VBA is to take the tediousness out of repetitive task and run over and over again with a click of the button.
31. **## Submission**
32. \* To submit please upload the following to Github:
33. \* A screen shot for each year of your results on the Multi Year Stock Data.
34. \* VBA Scripts as separate files.
35. \* After everything has been saved, create a sharable link and submit that to <https://bootcampspot-v2.com/>.
36. - - -
37. **### Copyright**
38. Trilogy Education Services © 2019. All Rights Reserved.

# 2. Input files

(i) alphabetical\_testing.xlsx: Contains sample data to develop the code

(ii) Multiple\_year\_stock\_data.xlsx: Contains the actual data for the project

# 3. Design criteria and Assumptions

The solution is aimed at creating a process with the following criteria.

1. Huge volume of data is assumed to be processed. Hence sorting of the data is avoided.
2. The stock trading date is not in chronological order.
3. Fist trading date for a ticker can be any day in the year – Not the first business day of the year
4. Last trading date for a ticker can be any day in the year – Not the last business day of the year
5. Data is to be read sequentially
6. Design to consider minimum number of passes / iterations in reading, to evaluate the information needed.
7. Design should be flexible to run on multiple worksheets without changes
8. Sample data is grouped by “Ticker Symbol” whereas the actual data is grouped by “Year”. Since the data is grouped by year on the “Multiple\_year\_stock\_data” no logic was added to check for the year. But, a logic to check for the opening price on the earlies date and the closing price for the last date is included in the design.

# 4. Metrics Definition

The definition of each of the metrics is calculated for each ticker by year and defined as follows:

1. Yearly Change = Percentage of (Opening price on the 1st day of data provided – Closing price on the last day of data provided)
2. Percentage Change each year = Percentage of (Opening price on the 1st day of data provided – Closing price on the last day of data provided) / Opening price on the 1st day of data provided
3. Total stock volume = Sum of stock transactions
4. Greatest % increase = The ticker with the maximum % increase for a given year
5. Greatest % decrease = The ticker with the maximum % decrease for a given year
6. Greatest Total = The ticker with the maximum number of stock transactions for a given year

# 5. Pseudo Code

1. Clear columns I through P on all work sheets: This is to erase all the data from the previous run
2. Get distinct of “Ticker Symbols on each sheet
3. Repeat the following steps on each sheet – “For Each WS”
   1. Filter the sheet for the first ticker symbol
   2. Get the EARLIEST and LAST transaction date
   3. Get the Opening price for the earliest date
   4. Get the Closing price for the Last date
   5. Calculate the metrics as defined in the “Metrics Definition” section
   6. Write the metrics to the Excel Sheet
   7. Loop for all Ticker Symbols
   8. Find the ADDRESS – ROW number of the ticker symbol with the “greatest % Decrease”
   9. Find the ADDRESS – ROW number of the ticker symbol with the “greatest % Increase”
   10. Find the ADDRESS – ROW number of the ticker symbol with the “Highest Stock Volume”
   11. Write the metrics to the Excel Sheet
4. Create a Form that shows the following
   1. Name of the Sheet that is being worked on
   2. Progress of the work in the form of a “Bar”

# 6. VBA Scripts

The following files are attached in zipped format:

1. VBA script as text file - VBA\_Challenge code\_v4
2. User Form - UserForm1.frm
3. User Form Controls - UserForm1.frx
4. Class File - ThisWorkbook.cls

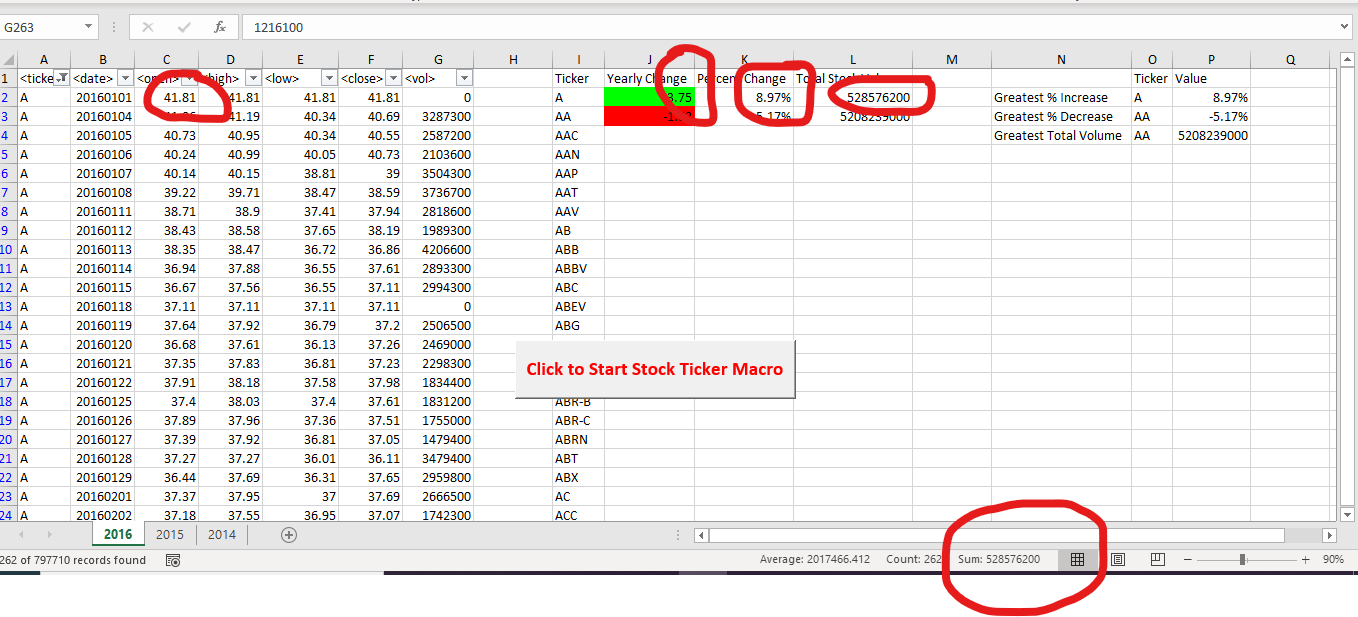


VBA\_code:



# 7. Code Validation & Testing

The code was tested by manually filtering for Ticker symbol -A for the years 2016 &2015. It matches with the values calculated through the program.



# 8. Conclusion

Following information is gathered:

**Moderate Solution:**

\* Column-I: The distinct ticker symbols gathered.

\* Column-J: Yearly change from opening price at the beginning of a given year to the closing price at the end of that year.

\* Column-K: The percent change from opening price at the beginning of a given year to the closing price at the end of that year.

\* Column-L: The total stock volume of the stock.

\* Formatted on Column-J: You should also have conditional formatting that will highlight positive change in green and negative change in red.

**Hard Solution:**

Row-N2: Shows Greatest % Increase

Row-N3: Shows Greatest % Decrease

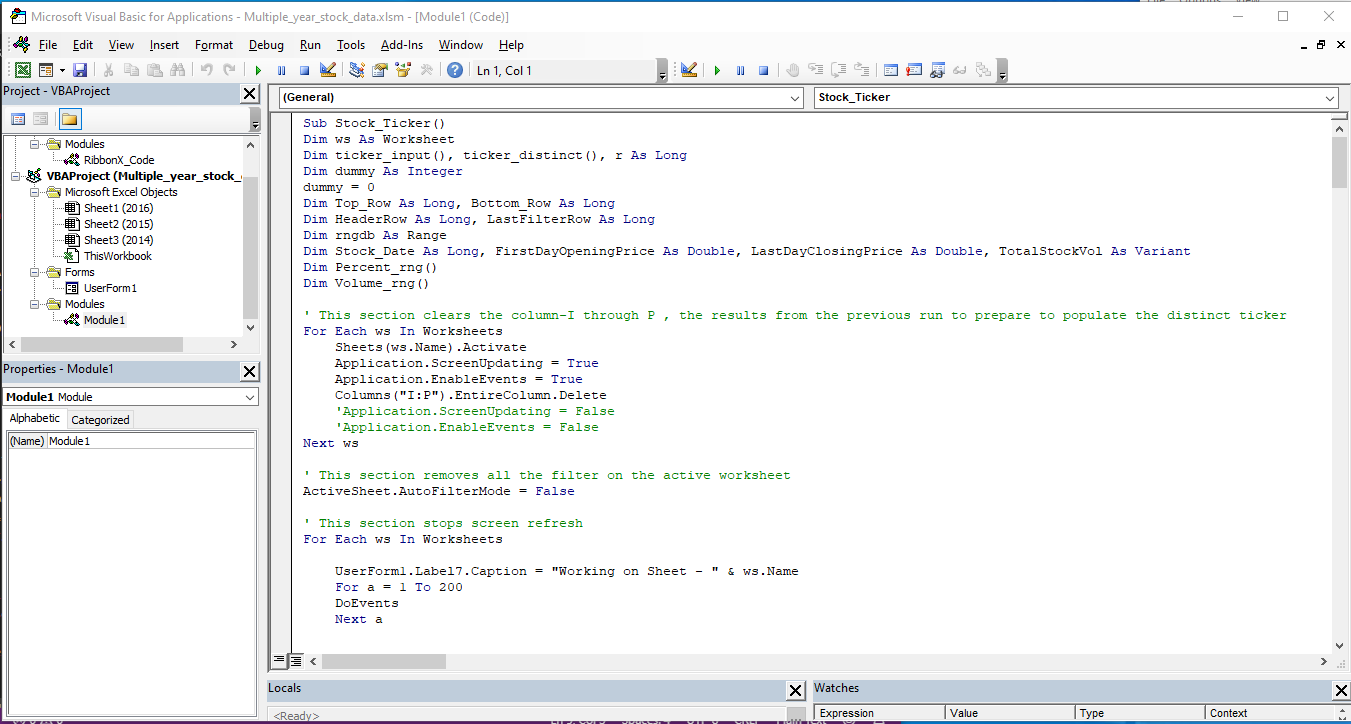
Row-N4: Shows Greatest Total Volume

\* A user form created to show the status of Macro run

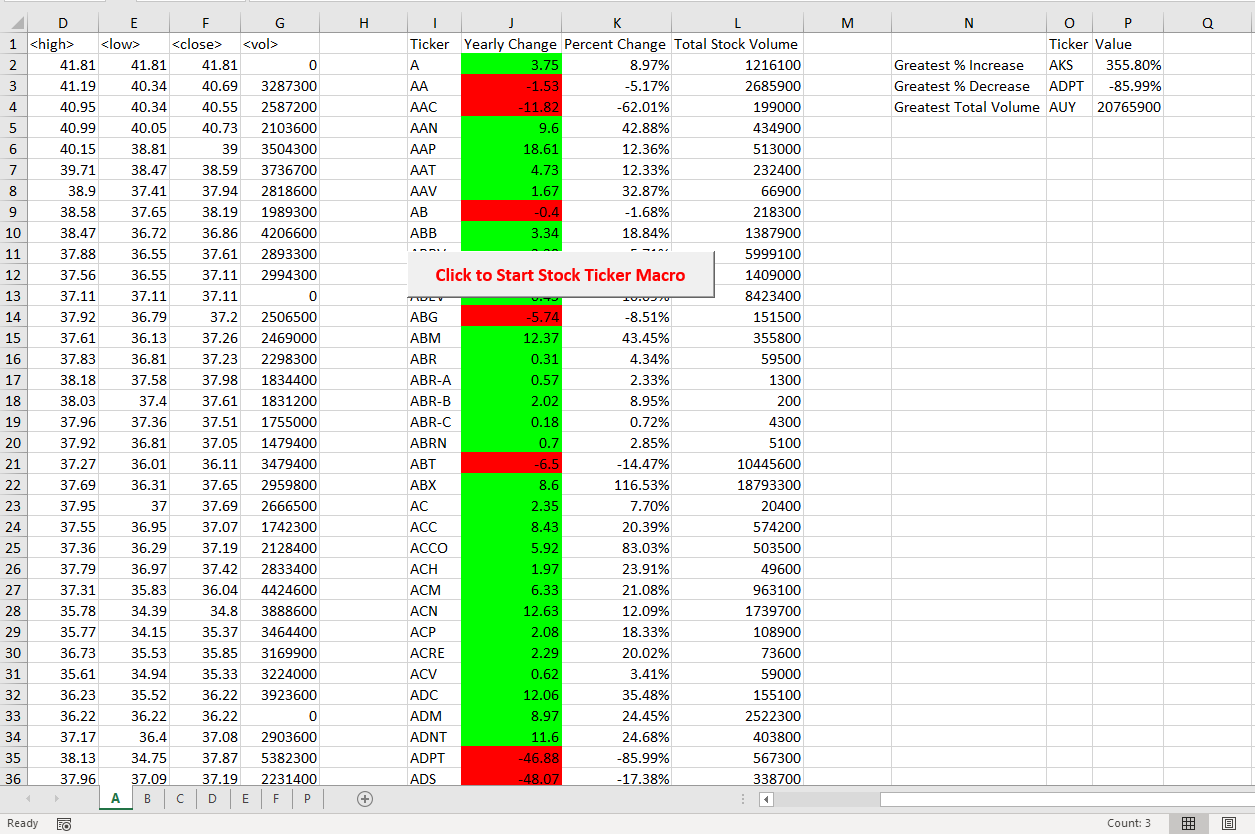
\* The results look as follows.

# 9. Screen shots:

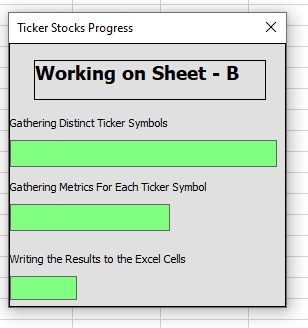
VBA-Code:



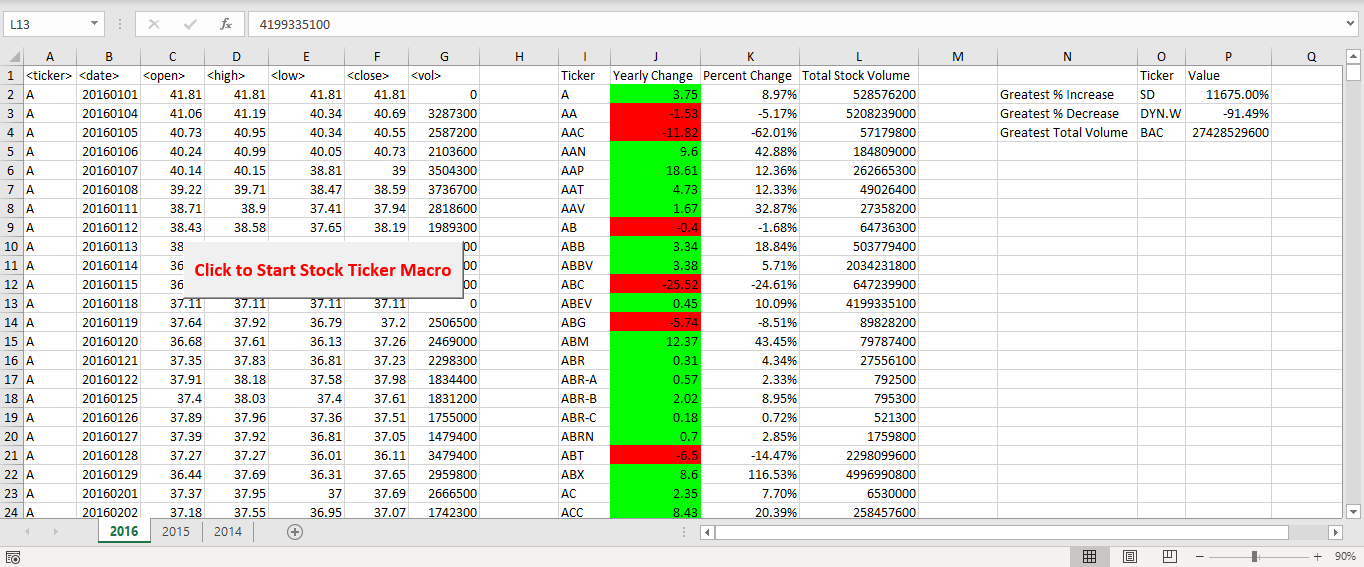
Testing with Sample set Alphabetical\_testing:



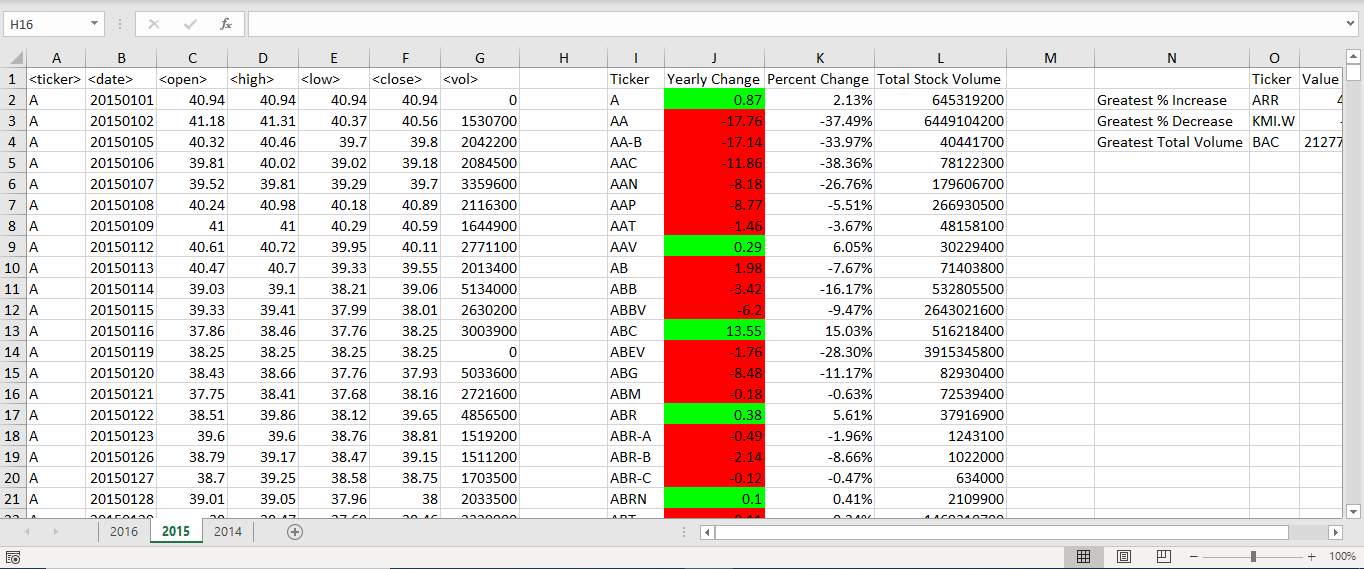
Progress Bar:



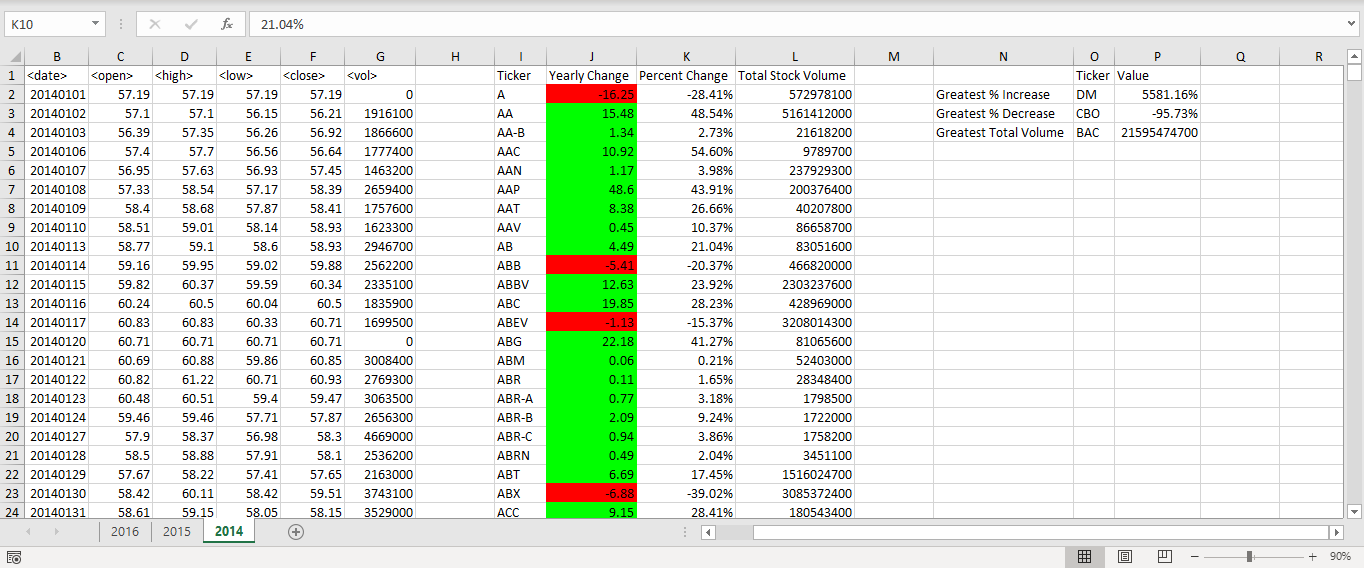
Year 2016:



Year 2015:



Year 2014:



Progress Bar – User form created:

