

# DayTrader Instructions and Reflections

Prepared for

Gove Allen, Professor  
Information Systems  
Brigham Young University

Prepared by

Kevin Cottrell

December 10, 2015

## Copyright

This project was done for Professor Gove Allen's IS 520 class during the Fall 2015 Semester at Brigham Young University. This model is not intended for personal use, especially for personal investing. Any such individual who uses this model for more than independent research on the potential with Excel is violating this copyright and is punishable by law.

## Table of Contents

|  |    |
|--|----|
| Copyright .....                                | ii |
| Executive Summary .....                        | iv |
| Introduction.....                              | 1  |
| Theory.....                                    | 2  |
| Instructions on Use .....                      | 2  |
| Initialize the Model .....                     | 2  |
| Import .....                                   | 2  |
| Score .....                                    | 3  |
| BuyAndSell Buttons.....                        | 5  |
| Now .....                                      | 5  |
| Timer .....                                    | 6  |
| Reflections .....                              | 7  |
| Update User-Defined Function: StockQuote ..... | 7  |
| Automatic Calculations .....                   | 7  |
| Deleted Purchased Stocks from List .....       | 7  |
| Additional Usages .....                        | 7  |
| Future Enhancements .....                      | 7  |
| Future Use .....                               | 7  |
| My Work .....                                  | 8  |
| Conclusions.....                               | 8  |

## Executive Summary

This model focuses on the theory that an individual may be able to find profitable stocks to trade on a daily basis. Many experts suggest that focuses on short-term performance and expecting large, short-term gains is not only unreasonable, but stupid. This model is a test for the contrary opinion.

This model enables a user to scan almost 7000 stocks and estimate which stocks would be beneficial for a timely gain. The user enables Excel to automatically check every five minutes and decide whether to buy or sell stocks.

This model is currently independent of a user and is designed to simply explore the possibility of short-term gains. Future enhancements must be made for this model to be usable for a human to interact with actual investing using advice generated from this model.

## Introduction

I have been interested in quite a while about the effectiveness of trying to quickly buy and sell stocks for a profit. Many experts would tell me that is a silly notion to take up. As an actuary, I deal with hedging stocks for company use with options. There are many sound models that are widely used that deal with using options to give the extra bang for your buck. Relying solely on buying and selling stocks is most likely not even in the future strategy of large firms.

This model is called DayTrader. It is designed to test the hypothesis that day trading can, in fact, be successful and lucrative.

Figure 1 shows DayTrader in use. This is the main tab showing key performance indicators as well as the stocks that are currently owned as well as the top 100 potential stocks of the day. This document is designed to be an instruction manual for the use of DayTrader with technical details on the VBA code used. After the technical explanation is given, information will be shared about my experience building it: strengths, difficulties, code comprises made due to ability or knowledge of VBA, and future enhancements.

The screenshot shows the DayTrader Excel spreadsheet with the following data:

| CurrentMoney         |  | \$31,562 | CurrentStockValue |  | \$97,661  |
|----------------------|--|----------|-------------------|--|-----------|
| PercentLimitPerStock |  | 25%      | TotalInvesting    |  | \$129,223 |
| PercentMinPerStock   |  | 10%      | Commission        |  | \$5       |

| HeldStocks  |        |              |              |               |                     |              |               |        |            | PotentialStocks |        |        |         |         |           |         |             |
|-------------|--------|--------------|--------------|---------------|---------------------|--------------|---------------|--------|------------|-----------------|--------|--------|---------|---------|-----------|---------|-------------|
| CurrentInfo |        |              |              |               | PurchaseInformation |              |               |        |            |                 |        |        |         |         |           |         |             |
| DaysHeld    | Ticker | CurrentPrice | CurrentValue | CurrentProfit | CurrentYield        | PurchaseDate | PurchasePrice | Stocks | Commission | PurchaseCost    | Ticker | Score  | Mean    | StDev   | LastClose | Now     | Probability |
| 6           | AVL    | \$0.08       | \$23,246.20  | -\$1,753.84   | -7%                 | 12/6/2015    | \$0.09        | 3E+05  | \$5.00     | \$25,000.04     | AMCO   | 97.474 | 0.02781 | 0.12874 | 0.32      | 0.32    | 0.5855074   |
| 1           | CUO    | \$14.90      | \$25,027.00  | \$21.92       | 0%                  | 12/6/2015    | \$14.88       | 1680   | \$5.00     | \$25,005.08     | BTN    | 87.369 | 0.0035  | 0.01719 | 4.69      | 4.6899  | 0.5812439   |
| 0           | BTG    | \$1.21       | \$24,187.74  | -\$809.76     | -3%                 | 12/6/2015    | \$1.25        | 19994  | \$5.00     | \$24,997.50     | WGA    | 95.171 | 0.01015 | 0.0521  | 0.52      | 0.5201  | 0.5758242   |
| 0           | MOC    | \$2.37       | \$25,199.95  | \$202.70      | 1%                  | 12/6/2015    | \$2.35        | 10655  | \$5.00     | \$24,997.25     | ACU    | 82.886 | 0.00191 | 0.01241 | 17.23     | 17.23   | 0.5612157   |
|             |        |              |              |               |                     |              |               |        |            |                 | CTO    | 89.837 | 0.00334 | 0.02182 | 55.259999 | 55.26   | 0.5607999   |
|             |        |              |              |               |                     |              |               |        |            |                 | BHV    | 78.086 | 0.00125 | 0.00956 | 17.549999 | 17.5471 | 0.5588085   |
|             |        |              |              |               |                     |              |               |        |            |                 | VMM    | 65.556 | 0.00079 | 0.00598 | 13.31     | 13.31   | 0.552452    |
|             |        |              |              |               |                     |              |               |        |            |                 | ACY    | 90.544 | 0.00293 | 0.02364 | 9.13      | 9.13    | 0.5493098   |
|             |        |              |              |               |                     |              |               |        |            |                 | AWX    | 95.917 | 0.00739 | 0.06472 | 2.1       | 2.1     | 0.5454361   |
|             |        |              |              |               |                     |              |               |        |            |                 | CPHI   | 96.932 | 0.0094  | 0.09625 | 0.24      | 0.2399  | 0.5405962   |
|             |        |              |              |               |                     |              |               |        |            |                 | BFY    | 66.518 | 0.00061 | 0.00616 | 14.47     | 14.47   | 0.539435    |
|             |        |              |              |               |                     |              |               |        |            |                 | IAF    | 81.01  | 0.00103 | 0.01112 | 5.56      | 5.56    | 0.5367637   |
|             |        |              |              |               |                     |              |               |        |            |                 | ASM    | 90.867 | 0.00195 | 0.02458 | 0.98      | 0.98    | 0.5316204   |
|             |        |              |              |               |                     |              |               |        |            |                 | ADGE   | 94.995 | 0.00337 | 0.0499  | 0.52      | 0.5199  | 0.5284253   |
|             |        |              |              |               |                     |              |               |        |            |                 | BPMX   | 95.073 | 0.00306 | 0.05089 | 1.18      | 1.18    | 0.523941    |
|             |        |              |              |               |                     |              |               |        |            |                 | CIX    | 89.362 | 0.00124 | 0.02075 | 11.17     | 11.17   | 0.5238479   |
|             |        |              |              |               |                     |              |               |        |            |                 | CANF   | 96.593 | 0.00442 | 0.08293 | 3.26      | 3.26    | 0.5212457   |
|             |        |              |              |               |                     |              |               |        |            |                 | DGSE   | 96.397 | 0.00276 | 0.07674 | 0.35      | 0.35    | 0.5143314   |
|             |        |              |              |               |                     |              |               |        |            |                 | ISL    | 71.447 | 0.00024 | 0.00726 | 17.690001 | 17.6901 | 0.512744    |
|             |        |              |              |               |                     |              |               |        |            |                 | VCF    | 82.63  | 0.00037 | 0.01222 | 14.62     | 14.62   | 0.5120541   |
|             |        |              |              |               |                     |              |               |        |            |                 | AMPE   | 92.048 | 0.00086 | 0.02877 | 3.42      | 3.42    | 0.511887    |
|             |        |              |              |               |                     |              |               |        |            |                 | BTI    | 77.773 | 0.00027 | 0.00942 | 115.33    | 115.33  | 0.5113423   |
|             |        |              |              |               |                     |              |               |        |            |                 | DHY    | 80.905 | 0.00016 | 0.01105 | 2.33      | 2.33    | 0.5057124   |
|             |        |              |              |               |                     |              |               |        |            |                 | BZC    | 93.514 | 0.0005  | 0.03645 | 19.74     | 19.74   | 0.5054194   |
|             |        |              |              |               |                     |              |               |        |            |                 | FCO    | 79.589 | 0.00014 | 0.0103  | 7.94      | 7.94    | 0.505406    |

Figure 1: A large look of the main page of DayTrader.

Enjoy learning about DayTrader. Feel free to contact me with any inquiries about the model. Noting the copyright listed above, the use of this model to actually buy and sell stocks is absolutely prohibited.

## Theory

The theory surrounding DayTrader is three parts: average rate increase over past 30 days, high volatility in stock return over past 30 days, and catching a stock on a low swing. Looking at each stock only 30 days at a time is a customizer feature; 30 days is an arbitrary number of days at this point in time, but the point is that it's not too long. Short-term gains rely only on short-term history.

## Instructions on Use

For a first time user, the first step is to declare how much cash you initially have. This goes into the main tab as the "CurrentMoney" input. Then, you are ready to declare which stocks you will analyze.

### Initialize the Model

Before this model is actively used, it must be initialized. The import feature only needs to be initialized when you want to change the stocks that are being considered. The score feature must change daily between close and open of stock market.

#### Import

This model starts with the current stock tickers in the NYSE, NASDAQ, and AMEX. CSV files are located in supplied folder. To import these stocks, click on the "Import" button in the designed "DayTrader" tab.



This will automatically import all stocks in the folder "Stocks." This folder should be located with the project file as included in the download.

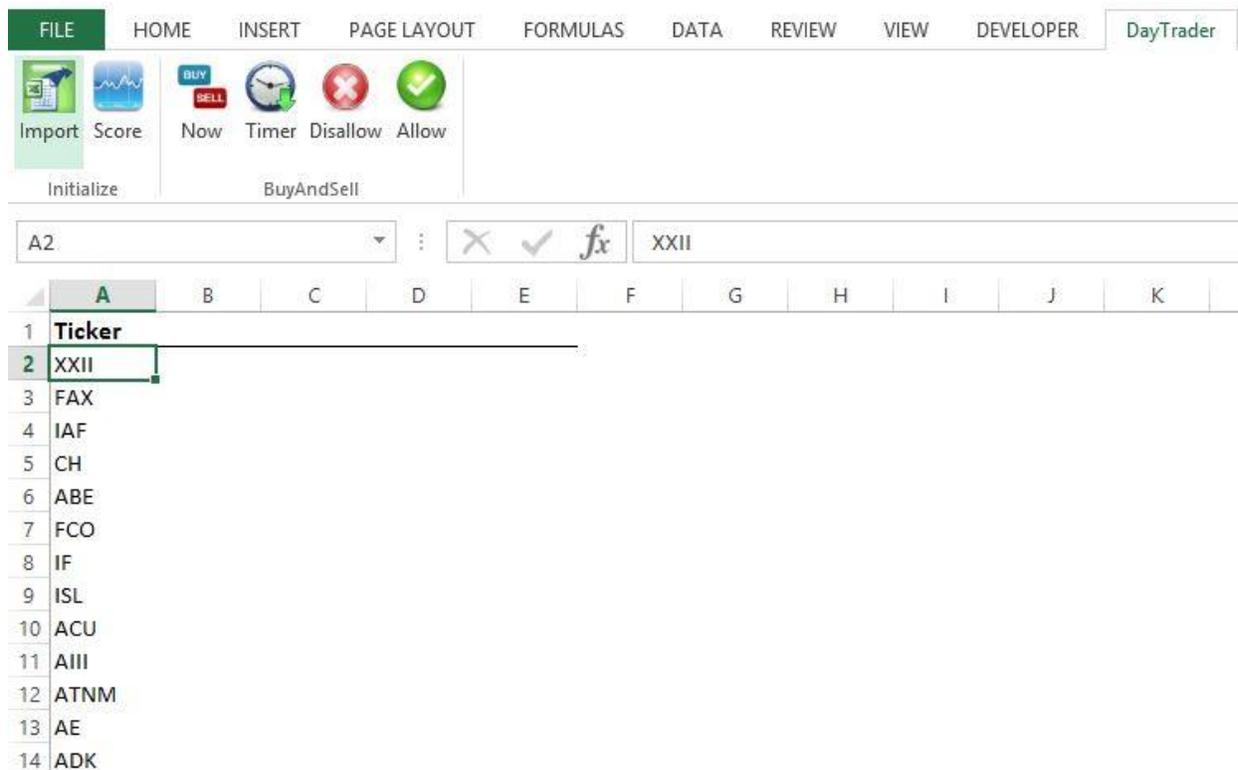


Figure 2: What the “StockScores” tab should look like after importing stock tickers.

Figure 2 shows what the “StockScores” tab should look like after the button is clicked. These CSV files have been downloaded from the NASDAQ site. Any CSV file saved in this folder and formatted the same way will import. Feel free to customize! This code only looks at CSV files in the folder, so any other file type will be ignored.

### Score

The “Score” button is the next button on the tab. This button is where lots of the magic happens.



This code is designed to be run between the close and open of stocks. It is not meant to be retrieved before the stocks open to take advantage of disparity between the close and open of a stock.

This code will look at each stock ticker you imported and look at its short-term history. The close rates are compared to get a mean and standard deviation (of rates from previous close, not closing stock price). Each score is then calculated according to a method owned by Kevin Cottrell which shall not be discussed except to say that the highest score is the best.

An array is used to collect all of this information in VBA without printing it onto the Excel Application. After all of this data is computed, the resulting array is then printed out onto the “StockScores” tab. This array is then automatically sorted by score with the best up top.

Currently, the top 100 potential stocks (best scores) are taken and placed on the home “HeldStocks” tab. Then stock prices are calculated for each of the 100 stocks. Lastly, the probability of gain is calculated based on the current price. The list is then sorted by probability, greatest listed first. These will be the stocks that are purchased. Probabilities can change every five minutes as the stock market is active and BuyAndSell buttons are active.

This process can take lots of time depending on internet speed. This sub queries the internet around 7000 times as is! That’s a lot. The status bar will report every 100 stocks that are scored as it computes. See Figure 3.



Figure 3: The progress is listed on the status bar. It records each 100 stocks that are scored.

Some computers take 10-20 minutes calculating this, while fast connections can take less. The computer will even state how long the program took at the end as seen in Figure 4.

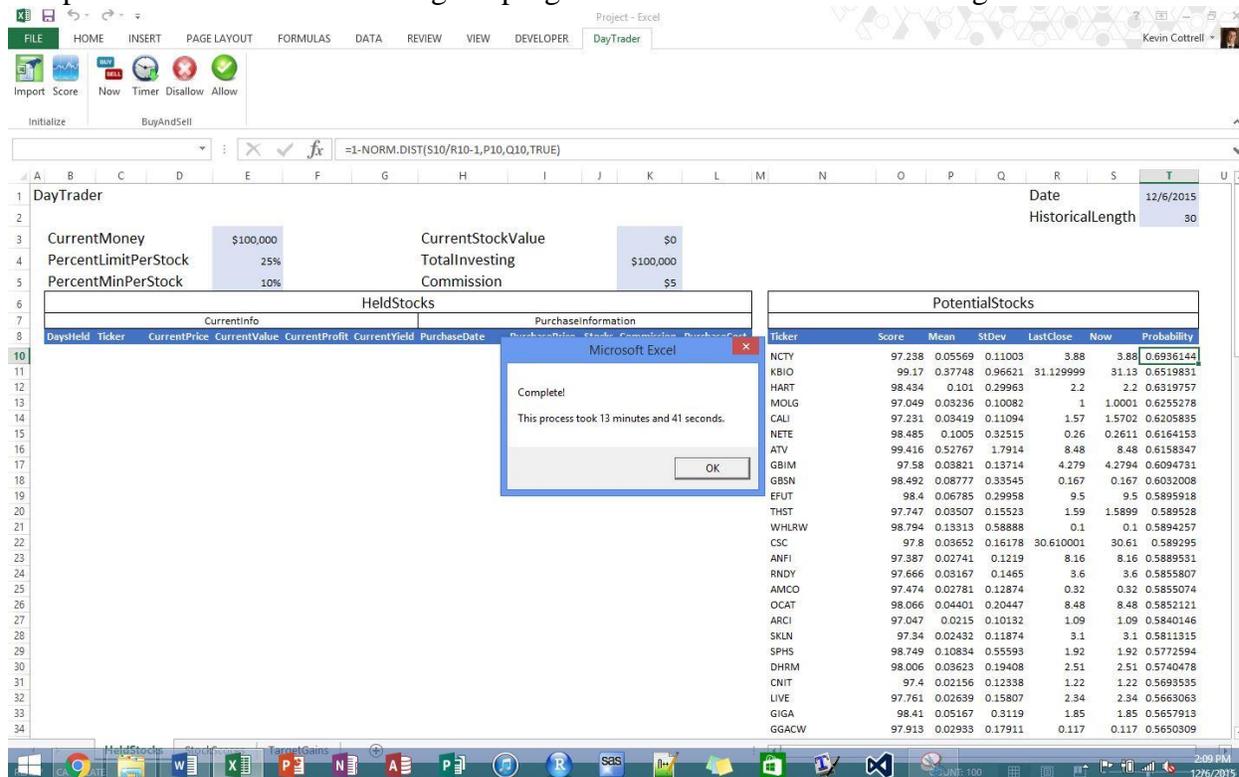


Figure 4: When the Score processes are completed, this message box will pop up.

The finished process will show a similar array as Figure 5 when complete.

| Ticker | Score  | Mean     | StDev    | LastClose |
|--------|--------|----------|----------|-----------|
| AEZS   | 98.309 | 0.03942  | 0.273999 | 7.94      |
| ABGB   | 97.728 | -0.02864 | 0.160881 | 2.25      |
| AMCO   | 97.474 | 0.027808 | 0.128741 | 0.32      |
| ERB    | 97.373 | -0.00803 | 0.123565 | 1.46      |
| STRP   | 97.366 | 0.004048 | 0.122099 | 12.14     |
| RVP    | 97.288 | 0.004092 | 0.116537 | 3.49      |
| CVSL   | 97.137 | -0.00519 | 0.107637 | 1.3       |
| MGT    | 97.108 | -0.00186 | 0.105821 | 0.26      |
| RBV    | 97.062 | -0.00376 | 0.10341  | 0.11      |
| GRH    | 97.031 | -0.02698 | 0.102997 | 0.16      |
| KIQ    | 97.004 | -0.00475 | 0.100447 | 0.96      |
| CPHI   | 96.932 | 0.009395 | 0.096255 | 0.24      |
| HH     | 96.853 | -0.01169 | 0.093646 | 0.12      |
| TAS    | 96.813 | -0.0035  | 0.091598 | 0.2       |
| YUMA   | 96.787 | -0.02963 | 0.091591 | 0.234     |
| AKER   | 96.751 | -0.03193 | 0.090193 | 1.5       |
| NTN    | 96.746 | -0.02135 | 0.089594 | 0.16      |
| SSY    | 96.729 | -0.02071 | 0.088895 | 1.07      |
| UAMY   | 96.727 | 0.010996 | 0.08757  | 0.38      |
| ENRJ   | 96.708 | -0.01179 | 0.087701 | 0.37      |
| AAMC   | 96.682 | -0.02597 | 0.087249 | 14.65     |
| DAKP   | 96.598 | -0.02494 | 0.084136 | 0.27      |
| CANF   | 96.593 | 0.004419 | 0.082931 | 3.26      |
| IGC    | 96.561 | 0.023704 | 0.081227 | 0.25      |
| NSPR   | 96.447 | -0.00058 | 0.07836  | 0.91      |
| SPP    | 96.431 | 0.028622 | 0.077002 | 14.25     |
| DGSE   | 96.397 | 0.002757 | 0.076742 | 0.35      |
| AGFS   | 96.347 | -0.00606 | 0.075565 | 6.02      |
| BGI    | 96.235 | -0.01008 | 0.072598 | 0.59      |
| HLTH   | 96.21  | -0.01679 | 0.072116 | 2.59      |
| ITPI   | 96.187 | -0.00178 | 0.071144 | 0.22      |

Figure 5: The results array will be on the “StockScores” tab sorted by Score.

## BuyAndSell Buttons

This model is designed for tracking the best stocks to buy and sell according to model theory. The current path is to initially click the “Allow” button first to enable ability to use the “Now” and “Timer” buttons. The model will then evaluate if it should sell and then buy any stocks every 5 minutes. If at any time you want to stop this process, click the “Disallow” button. When the stocks were supposed to refresh, instead they will not and the code will end.

Below is a commentary on the “Now” and “Timer” buttons and how they work.

### Now

The now buttons starts the process of selling and buying immediately. First, it will recalculate the workbook. Then it will decide if it should sell any owned stocks. The process is simple. If the stock earned enough, it will be sold. In the “TargetGains” tab, there is an array of goals. If the stock has surpassed that goal based on how many days the stock is owned, it will be automatically sold. The value of that stock (commission already taken out), is then added to cash and taken from the invested stock money. The stock is then removed from the CurrentStocks list.

After any selling has occurred, buying occurs. For both buying and selling, the status bar is updated as shown in Figure 6.

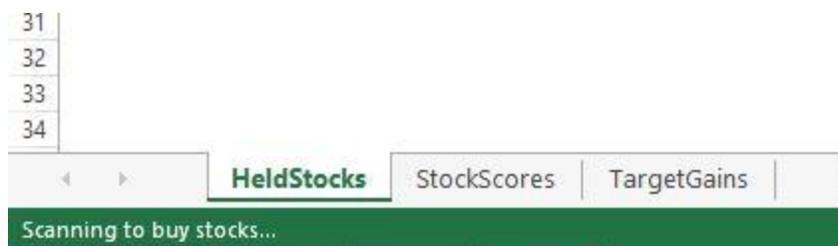


Figure 6: The status bar is updated to show the user it is running.

For buying, the stocks are updated and resorted based on what the probability of being a great stock is as of right now. The stocks will probability greater than the import on the “TargetGains” tab will be purchased. The stocks will be purchased with up to 25% of total money invested (sum of cash and currently invested). If there is not 25% as cash, it will buy with as much as it can down to 10%. If there is not 10%, it will not buy. It will also not buy if there are already some shares of that ticker as a protection. The stock’s information is then recorded as a purchased stock. The sheet is calculated again.

This process usually takes 30 seconds to a minute, but could take longer if many stocks are to be bought and sold.

This process is designed to repeat every five minutes. This is the target of this model: short-term buying and selling. The status bar will update with what time the next buying and selling will start.

If at any time you do not want it to calculate anymore, simply click on the “Disallow” button and the automatic sequence will stop. This is calculated in the code starting the next time it was supposed to run. This means that you cannot simply click on “Disallow” then “Allow,” but must wait until after it was supposed to calculate again before you click “Allow.”

#### Timer

This is the button that automatically buys and sells, but only during business hours as inputted in the “TargetGains” tab. The process is exactly that of the “Now” button, but simply starts when you declared as shown in Figure 7.



Figure 7: After clicking timer, the status bar will say when it will start to buy and sell.

This process will automatically start until the computer falls asleep. Therefore, either click this button before bedtime as long as your computer’s setting are such that it will never sleep, or click on it in the morning before the scheduled time.

This process also has an ending time to input so that you don’t have to click on the “Disallow” button.

## Reflections

This model was not easy to make. 15 to 20 hours have been spent on this model, and there are many plans to continue to improve it. There were many processes that were hard, but it proved very rewarding.

### Update User-Defined Function: StockQuote

This worksheet relies on getting stock quotes every 5 minutes. I made a function to get this stock quote using a combination of code from class and code online as well as my own intuition. However, having over 100 of these functions on the worksheet caused problems. These functions would not instantly, automatically refresh like basic functions in excel. Therefore, the entire spread had to be taken off automatic calculation. Every time a key action is performed in the code and information about the stock value is taken from the application, the whole spreadsheet had to be recalculated. The line “DoEvents” was common in the code for this purpose.

### Automatic Calculations

Automatic calculations were tough as I did not fully understand how it would work. Many google searches were made then eventually global Boolean variables were made to say what it currently can and cannot do. Some “GoTo” statements were added out of frustration. When they worked, they stayed.

### Deleted Purchased Stocks from List

For the life of me, I could not get stocks to delete once they were purchased. Or worse, the values would delete, but not the cells, causing run-time errors. I tried recalculating, “DoEvents,” and even creating a function that would force recalculation of everything. Nothing worked. Eventually I put it in its own function. It worked after that. It was frustrating

### Additional Usages

Additional items such as enumerations, status bar updates, array, web queries, and sorting were used with VBA. These items took time to implement, but seem to run very well. I tried to make sure code was fast by not interacting with the application more than I had to. For example, when scores are updated, the status bar only updates after every 100 scores are calculated using the “MOD” statement.

### Future Enhancements

I would very much like to use this model in the future. I will add an emailing features that indicates when a buy or sell should happen. A user form would then pop up declaring me to enter the suggested activity that can be modified based on what actually happens. All buys and sells will then be recorded in another tab for historical sake.

### Future Use

This model has not been proven. That’s the reason it has been made. This model will be run for a while to judge if there is merit in the process. If so, the model will be upgraded and used in a dry run with a mock buying and selling website. If that is successful, this model may be used solely by Kevin Cottrell for business use. Any ties to code, pictures, etc. will be removed so as not to infringe on anybody else’s work.

### My Work

This code and model is my work. Retrieving stock history code was loosely adapted on code presented in class on retrieving lots of stock information. The function for obtaining instant stock quotes is closely related and some help from multiple sources on the internet was used. Other than receiving guidance from these sources, the inspiration for this model and code came from me solely.

### Conclusions

This project was a lot of fun. I anticipate a long career in quantitative analytics, or at least as much as any actuary can hope for. I anticipate VBA will be my go to source until this is professionalized and another software seems better.