

# Algorithmic Trading Program

---

Due: December 10th 2015

Questions or comments please contact:

William Burke

wburke2@u.rochester.edu

variables not be used: Volume; adj. Close

For this project, you will be given Apple's stock data for the previous 15 years. You are to write two different programs, one for which I will provide most of the parameters and algorithm and one that you will create yourself.

Stocks fluctuate significantly throughout the day and night. We will not attempt to explain the specific reasons why stock data fluctuates. However, **your project will attempt to predict future stock price based on past data.** You are provided the following variables: Open, High, Low, Close, Volume, and Adj Close. It is only possible to trade stock during the hours the market is open. The value of a given company's stock when the market opens is referred to as the Open. The High is the highest value that stock reaches that day. The Low is the lowest value the stock reaches that day. The Close is the value of the company's stock when the market closes. The Volume is the number of trades that occurred with that stock on a given day. The Adj Close is the closing price that is modified by analysts based on corporate actions.

## 1

The **first goal** for the project will be **to import the data**, which is saved as a csv file, into your program. First, inspect the data (you can inspect the data by opening it in an Excel or TextEdit) then ask yourself: Is the data formatted how you would like it? For example: The data provided to you is listed most recent data on top and earliest at the bottom. During the data preparation,

if you choose to modify the data before the analysis, you are allowed to go through all the data and modify it according to your liking (i.e. putting the earliest data is at the top and the most recent the bottom. This makes the program easier to iterate through). Your program should iterate through the days from the earliest date to the most recent, as if it was receiving a stream of data. It is VERY important that your analysis does not incorporate future data (i.e. including days that have not “occurred yet” in your analysis).

## 2

The first analysis you will perform will be a moving average of the stock data. For this analysis assign yourself 1,000 fake US dollars to use to invest as your initial principle amount. Your program should decide to buy stock, sell stock, or take no action based on the average of the stock prices for the previous twenty days.

Your program should not take any action during the first twenty days so that it can use them to calculate your first average. You will use this average to compare to the 21st day. Then your program should recalculate the average of the 20 days starting at 2 and ending at day 21. This pattern should be repeated until the last day of data.

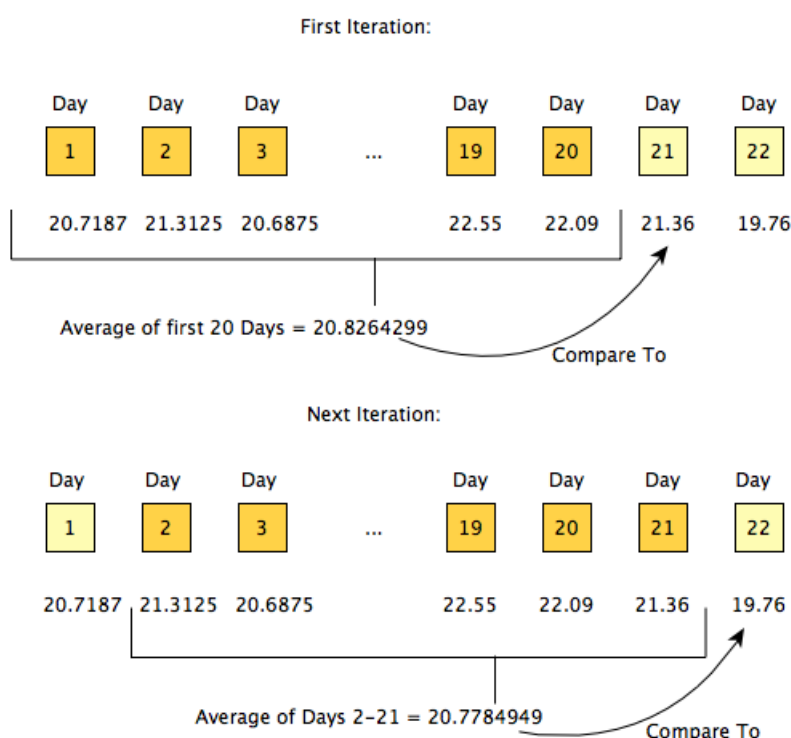


Figure 2.1:

For example: if you notice that the current day stock price is 20% lower than the average of the previous 20 days, you might decide that your program should choose to purchase some stock because you believe it will rise above that price soon. Then, if your algorithm observes that the current stock price is higher by a certain percent than the average, you might decide to have your algorithm sell. The time period of 20 days and other parameters are for you to decide and play with to maximize the amount of money your algorithm generates.

When your algorithm reaches the last day of data, have it sell all remaining stock, if you have any. Please have your program report the total amount of money you are left with.

Be sure to keep meticulous track of your finances. If your algorithm chooses to buy stock, keep track of both the number you purchased as well as the amount of money used to purchase the stock. The same logic should be applied to selling stocks.

In your comments at the beginning of your code, please explain thoroughly how your program works. For example, if your algorithm uses the previous 30 days Close stock values, you may choose to buy or sell based on the Open value. Subsequently, your algorithm might decide to take one action based on the Open value, and another on the High, Low, or Close value of that same day.

One important thing to remember is that it is okay if your program does not make money. Making money is not an easy task and this is a learning exercise.

How does your algorithm compare to purchasing as much stock as possible on the first day and selling on the last? Does the stock price grow over time?

### 3

The next task will be to come up with your own algorithm to analyze the data. This analysis has to be different from a moving average. An important part of being a computer scientist is to be able to bring your ideas from the drawing board to a working program.

You should try to be as creative as possible, for example you could sell stock on the even days of the month and buy on the odd. There are many different possible options. How does the algorithm you created compare to your moving average algorithm?

## 4 EXTRA CREDIT

There is a lot of possibilities to earn extra credit for this project. The first option is to implement your program using object oriented programming. Your program can be written without object oriented programming, but you can earn up to 10 % extra credit if it is implemented with object oriented programming.

There are many packages that can be imported when programming in Python. One useful visualization package is matplotlib, which can be used for making plots and graphs. If you import matplotlib and create several informative graphs regarding your analysis, along with an explanation regarding why they are meaningful, you can earn up to 5 % extra credit.

## 5 GRADING

- 10 % for properly importing the csv file into the program.
- 35 % for calculating and implementing a moving average in your first program.
- 35 % for having your programs correctly keep track of your finances and sell remaining stock on the last day.
- 20 % for generating an alternative algorithm.
- 10 % for implementing object oriented programming (will be covered later in semester).
- 5 % for importing and plotting figures using the package matplotlib.