MAE 215 (Sections: 17781, 13003, 13004): Data Processing Lab – 1

How to become (an imaginary) millionaire*?

In this Data Processing lab, we will investigate the power of investing, the technicalities, and how to calculate investment returns from a simple investment strategy (and pay Uncle Sam to make sure we don't end up in jail). Since this is hard to do with pen and paper, we need our MATLAB's help!

In the process, we will learn how to read MATLAB documentation and figure out how to use various functions.

Previously:

You woke up on 1st January 2016 with a *strong feeling* that nothing bad at all was going to happen in the world in the coming years! As a New Year's resolution, you decide that we'll invest $3/day/stock when the NASDAQ is open in Microsoft, Apple and Google/Alphabet companies. They are referred to by their stock market tickers as MSFT, APPL, GOOG. We keep at it and steadily build a portfolio for the next 8 years!

Present day:

Today is 31st December 2024, and we want to sell the entire stock to buy a pet python (or not, depending on how much we made/lost!). We want to evaluate our stock value (to make sure the stockbroker doesn't cheat us) and estimate our taxes (to make sure we don't end up in jail).

Open up your computers, and our dear MATLAB program to conjure up a script. We will do the following:

Step-1: Download the stock data (10% Grade)

- Go to the NASDAQ website: https://www.nasdaq.com/
- Navigate to `Market Activity > Stocks > Symbol Search: MSFT`
- Go to `Historical Quotes` page, select `MAX` and hit `Download historical data`.
- You should download data going back 10 years in a CSV file. You can inspect this in Excel.
- The data includes the date, the closing price the previous day, the opening price on the day, the highest and lowest price transactions that happened on that day.
- Repeat this for the other stocks.

Step-2: Read and analyze the stock data (60% Grade)

- Import the data into MATLAB as a table with variable name: `DATA_TCKR`, where TCKR is replaced with appropriate ticker (e.g. `DATA_MSFT`).
- Sanitize the data:
    - Convert the Date data column (which is in a string format) to MATLAB date-time format. This can be done using the `datetime` function as, for e.g. `datetime('3/3/2025')`. Date-time format tells matlab that the string actually represents date and time (instead of text), and gives you special tools to handle it (for e.g. plots can be zoomed in, dates can be subtracted to know the duration, etc.)
    - We need to remove the '$' sign in the price columns and cover the string to numbers (Matlab doesn't know that these are numbers, yet). We do this by replacing the `'$'` text by an empty character (`''`). This is MATLAB speak for *removing*. Use the

`replace` function for this. Then you can convert the string (with only the numbers) to numbers using the `str2double` function.
    - Select data only between 1/1/2016 to 12/31/2024. You can remove all other dates. Remember, stock market doesn't trade on weekends and holidays. So you will have those dates missing.
- Let us visualize how the stocks have done during this time. In *figures 1-3*, plot the variation of price with date for each of the stocks (MSFT, APPL and GOOG respectively).
    - `figure(1)` command creates the figure with an ID of 1, so on and so forth.
    - For each plot, plot the opening price in black, with a line-width of 1.5.
    - On the same plot, plot the day-high and day-low price in blue lines, with a line-width of 0.5.
    - Have grid enabled for all plots, appropriate titles, labels for the lines and axes. Look up '`DisplayName`' option that you can provide in the `plot` command so that they appear on the legend.
    - Your stock was purchased somewhere between these two lines. But we will assume that it was purchased at the opening price. Hopefully, with the Lord's blessings, the stocks all went up (and high!).
- Stock *volatility* (i.e. how much the stock price can fluctuate) is a measure frequently used by analysts to gauge the risk associated with the investment. One of the measures is the Bollinger Bands, where, for short term purchase and sales, they use a moving average and moving standard deviation of opening stock price to visually represent the same.
    - For each table, create columns called `MovingAverage` and `MovingSTD` to represent these. You can use the `movmean` and `movstd` functions to create these. IMPORTANT: *Remember to make sure you do a 20-point trailing average (i.e. 19 days in the past, the current date, and 0 future dates)*. This is because, in reality, you don't have future date information. Read the documentation to do this correctly, and note the direction in which your dates are changing.
- Plot the Bollinger Bands as a function of date in figures 101-103 for each corresponding stock.
    - For each plot, plot the opening price in black, with a line-width of 0.5.
    - On the same plot, plot the moving average you calculated in black, with a line-width of 1.5.
    - On the same plot, plot 2*`MovingSTD` on either side (i.e. $\pm$) of the `MovingAverage` in red, with a line-width of 1. These are the Bollinger Bands, and indicate how much variation there was on the stock.
    - Format it similarly to figures 1-3.

Step-3: Analyze our returns (30% Grade)

- Remember, we invested $3/trading-day/stock starting 1/2/2016? Add a column, called `StockPurchased` that indicates the amount of stock purchased for that amount each day at the opening day price.
- Add a column, `StockCurrentValue`, that indicates the value of the purchased stock as of 12/31/2024.

MAE 215 (Sections: 17781, 13003, 13004): Data Processing Lab – 1

How to become (an imaginary) millionaire*?

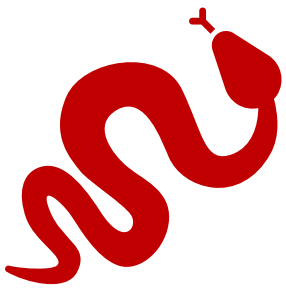
- Create variables called `PortfolioValue_TCKR` for each stock that indicates the total portfolio value (as of 12/31/2024) of all stocks purchased for each ticker.
- Create variables called `Returns_TCKR` that indicates the net returns on each stock. You will subtract the investments made for each stock (you know how many investments of 3$ you made for each stock) from the final portfolio value. **This is how much money you made on each stock, pre-tax!** 🥳 💰 🤑
- Assume that your portfolio was entirely liquidated (i.e. sold and converted into USD) on 12/31/2024. Create a variable `FinalReturns` representing this.
- Create a variable called PercentageFinalReturns, that tells you how much money you made on your invested money. You calculate it as follows:

  % Final Returns = [(∑ Final Returns)/∑ Investments] * 100
- Compute the long-term capital gains tax as per the 2024 tax rates, assuming this was your only income (and you are a single filer). Use a net tax rate of 15% on total earnings. This is Uncle Sam's cut! Call it `TaxOwed`. Technically, stocks purchased and sold in 2024 are considered short-term capital gains. But for simplicity, we will assume everything as long-term capital gains.
- **Finally, calculate the post-tax earnings from your investment and call it `PetPythonBudget`!**

For the curious in you, we made some approximations in this exercise. For example, we ignored the dividend payouts, inflation, time-value of money invested, tax brackets, etc. But the numbers still roughly indicate what you'd get if you invested in this way!

I hope that, by doing this, you learn:

1. How to use MATLAB to read and analyze data for a practical purpose.
2. How to read MATLAB documentation to figure out function syntax. **Remember: always use/try to find pre-written core MATLAB functions before deciding to write your own code/implementations/functions.**
3. A lot of debugging!!
4. The power of investing (hopefully in good stocks)! Remember that the economy went through hell and high-water in these 8 years, but you stayed true to your promise on 1/1/2016 (and it worked ok so far). Again, golden rule of stock market investing: evidence of past gains are no guarantee for future returns!



*no guarantees. Not a financial advice!

Spring 2025