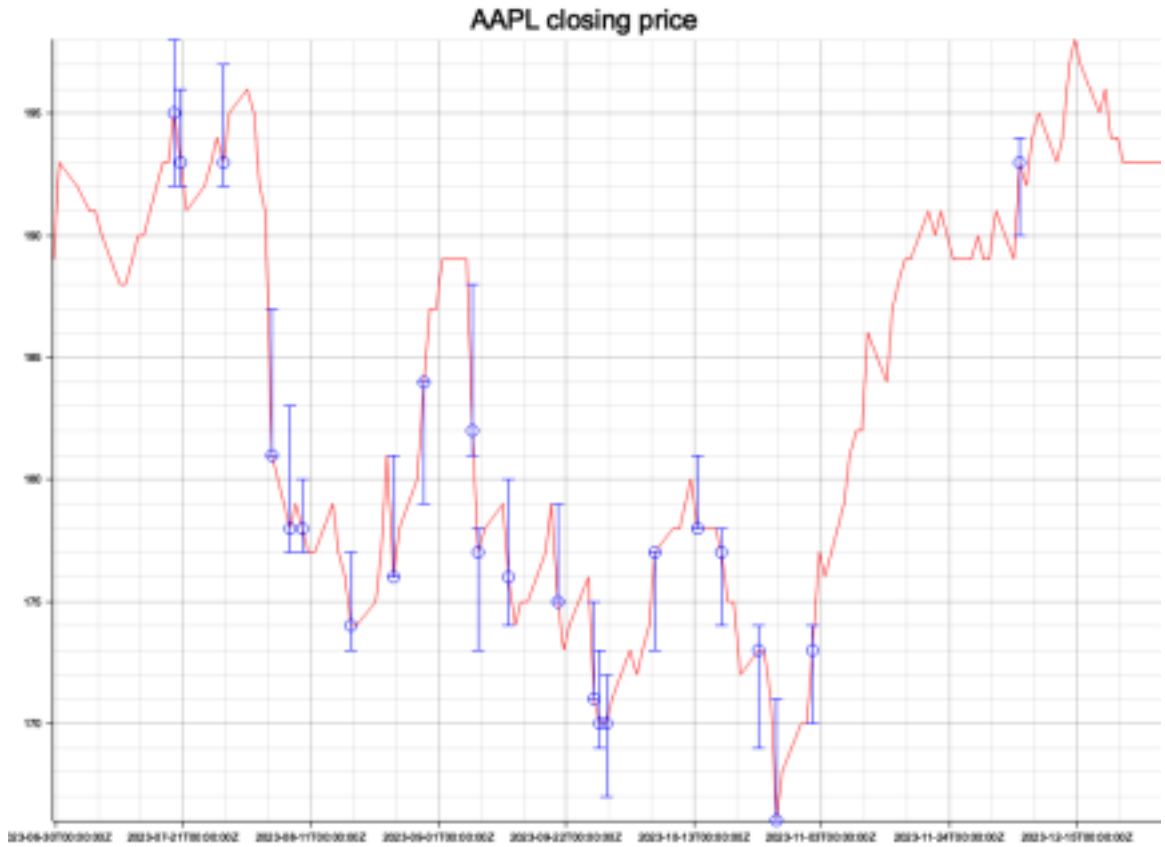
ECE 421 Project 1: Stock Market Monitor

Please note all projects are undertaken as GROUPS! Groups are normally of size 3. All members of the group will receive the same grade.

Write a simple stock monitoring program that takes a stock symbol as input and outputs a chart showing the daily closing price for the last six months. In addition to the daily closing price, the chart should highlight "volatile" days, where the stock price varied by more than 2% of the total price (as measured by the difference between the intra-day high and low). Finally, the program will print the minimum and maximum closing price for the interval, and the dates on which these values occurred.

As an example, the following chart was generated when the program was run with "AAPL" as input:



The error bars in blue show the intraday low, high, and close on volatile days.

Requirements

1. Your program must be written in Rust, using a functional programming style where appropriate. For example, to set the y-axis range the maximum value might be determined from the input stream as:
    - `let ymax = data.iter().map(|x| x.close).max().unwrap();`
2. Your program should read the stock ticker from the command line. You don't have to use the clap crate, but you do need to provide "--help" if asked.
3. You can use any method you want to get the stock quotes, but the crates that access Yahoo Finance (e.g. yahoo_finance) are free and easy to use.
4. Your program cannot crash if you are given a bad stock symbol.
5. You can use any method you want to generate the chart, but the plotters crate is popular and powerful. At minimum, you should generate an image file, but the plotters crate supports many back-ends, including WASM.
    - If you can generate a fancier plot than just an image file, your team will earn an extra 5%.
6. You are allowed to plot analysis other than volatile days, for example slow/fast moving averages or Bollinger bands. Just be aware that other indicators such as RSI or MACD require a second chart because they can't be superimposed over the closing price graph.

Deliverables

The program must be completed by **Friday February 16 @ 11:59 p.m.**

Write a proper README.md file in your rust project directory including:

1. The crates you used (if you used any) and why.
2. The financial analysis algorithm you used.
3. Charting setup, e.g. candlesticks vs scatter plot and autoscaling or axis features.
4. Project setup.
5. Usage instructions.

Please hand in your entire rust project by submitting via eclass to the group account. The project can be submitted as a tarball or zip file.

You will also be required to demo the project during the labs on **Monday 26th February** or **Tuesday 27th February** depending on your lab section.