

Lab-13

your name here

2024-04-25

Preface

The goal of this assignment is to help you gain familiarity with web scraping. As always, please come to office hours and reach out to your teaching staff if you have any questions.

Data

We will get data for this assignment from the web.

1. Let's warm up by building on our work to scrape stock prices from this week's example. Write code that scrapes Goldman Sach's historical daily share price from <https://finance.yahoo.com/quote/GS/history>. Clean the data, converting Date to a date and all other variables to doubles. Print the first 10 rows of the resulting data frame. What is the earliest date for which prices are available?

Note: If you don't know where to start, review example-13-solutions on Posit Cloud!

The earliest date for which prices are available is...

2. Now generalize your code from question 1 so that it returns historical daily prices for any ticker assigned to the name `symbol`. First assign a new ticker – “JPM” – to `symbol`. As before, clean the data and convert `Date` to a date and all other variables to doubles. Print the first 10 rows of the data frame.

Note: example-13-solutions could be helpful here, too!

3. Adapt your code from question 2 to create a variable `ticker` with the contents of the variable `symbol`. Use `relocate` to put `ticker` in the first column, and print the first 10 rows of the resulting data frame.

4. Tickers can be pretty hard to decipher. Modify your code to scrape company names from yahoo finance, and store it in a variable `company` in your data frame. Remove the trailing ticker in parentheses and any whitespace to get just the company's name. Assign the resulting data frame to the name `symbol_prices`, relocate the variables `ticker` and `company` to the front of the data frame, and print the first 10 rows.

5. Use the variable `Open` in the data frame from question 4 to compute the stock's cumulative returns over the period of data you scraped.

Note: Make sure you get the right dates when computing returns!

6. Now let's use the Download option to work with yahoo finance data directly rather than having to scrape it. Go back to the historical data for JPM on yahoo finance and retrieve the link to download the data. Use `read_csv` to read in the data and plot Open prices over time.

7. Now let's try to get lower-frequency price data over a longer time period. Go back to the historical data on yahoo finance and use the header to customize the time period to five years ("5Y") and the frequency to Monthly, then click Apply. Verify that the table updated. Now copy and paste your code from 6 and modify it to use this new url, plotting monthly Open prices over the past 5 years.

8. Now let's try to get the monthly price data over a custom date range: from January 1, 2000 to the present. Modify your code from 7 to depend on a variable `start_time` that encodes the date January 1, 2000 as a Unix time – the number of seconds since the start of January 1, 1970. The result should be a plot of share prices from January 1, 2000 through the present.