Homework - Week 13

your name here

2025-04-23

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 into a function that takes a ticker as an input and returns historical daily prices. As before, clean the data and convert Date to a date and all other variables to doubles. Have the function return the data frame. Test the function with a ticker of your choice, so that it prints the first 10 rows of the data frame below.

3. Adapt your function from question 2 to create a variable `symbol` in the data frame that contains the value of the argument `ticker`. Use your revised function in conjunction with `map()` to scrape prices for several of the top employers for Dyson graduates: Bank of America, Barclays, Capital One, Citigroup, Goldman Sachs, J.P. Morgan, Lazard, and Morgan Stanley. Use the function `bind_rows()` to combine the list of data frames into a single data frame. How many rows are in the data frame?

There are … rows in the data frame.

4. Tickers can be pretty hard to decipher. Modify your function to scrape company names from yahoo finance using the selector `"title"`, and store it in a variable `company` in the data frame your function returns. Remove the trailing ticker in parentheses and any whitespace to get just the company's name. Use the revised function to create a data frame for the companies from question 3. How many columns are in the data frame?

Note: It's best practice to minimize requests to websites when scraping, so please only `read_html()` once within the function.

There are … columns in the data frame.

5. Use the variable `Open` in the data frame from question 4 to compute cumulative returns for each company over the period of data you scraped. Plot the amounts in a graph with the best performing stock first and worst performing stock last, labeled using company names (not tickers).

6. Use the variable `Open` in the data frame from question 4 to plot share prices over time for each company in facets using the option `scales = "free_y"`. Make sure the facets are labeled with company names (not tickers).

7. Now let's try to scrape data over a custom date range. Go back to the historical data on yahoo finance and use the header to customize the frequency to Monthly and time period to five years ("5Y"). Verify that the table updated. Now copy and paste the updated url into the code chunk below. Modify your code from question 4 to create a data frame of monthly prices for each ticker, and then plot the Open prices over time for each company in facets using the option `scales = "free_y"`.

8. What if we want to focus on a different time period? Customize your function from question 7 to take three arguments: the ticker, start date, and end date (in date format). Use the function to make a data frame of monthly prices for the ticker `GS` from January 1, 2000 through December 31, 2024, and make a plot of Open prices analogous to the one from question 7 (but for just Goldman Sachs).