

# Lab-14

your name here

May 06, 2022

## **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 Twitter's historical daily share price from <https://finance.yahoo.com/quote/TWTR/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 on this lab, review example-14-solutions on RStudio 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, and print the first 10 rows of the data frame.

*Note: example-14-solutions could be helpful here, too!*

3. Adapt your function from question 2 to create a variable `ticker` with the ticker for that firm. Then use the new 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 `bind_rows()` to combine the list of data frames into a single data frame. How many rows are in the data frame?

There are...

4. Tickers can be pretty hard to decipher. Modify your function to scrape company names from yahoo finance, and store it in a variable `company` in the data frame your function returns. Try to remove the trailing ticker in parentheses to get just the company's name. Print the first 10 rows of the data frame.

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

*Hint: It can be difficult to pin down selectors on some sites. If you have trouble, just manually enter `h1` in the SelectorGadget text field, hit enter/return, and see if that gets the information you want.*

**5. Use the variable `Open` in the data frame from question 3 to compute cumulative returns for each company. Plot the amounts in a graph with the best performing stock first and the worst performing stock last.**

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

6. 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 time period to five years ("5Y") and the frequency to Monthly, then click Apply. 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 Open prices for each ticker. Plot the prices over time for each company in facets using the option `scales = "free_y"`.

7. What if we want to focus on a different time period? Customize your function from question 6 to take three arguments: the ticker, start date, and end date (stored as dates). Use the function to make a data frame of monthly prices during the time you have been at Cornell, and make a plot analogous to the one from question 6.

*Hint: The yahoo finance url takes Unix times, which are the number of seconds since the start of January 1, 1970.*



8. **OPTIONAL:** If you want some extra practice, you could extend the function from question 7 to take additional arguments for the frequency of price data and/or the price series (e.g., Open, High, etc.) to plot. Or you can do something else of your choosing. Or just stop here!