**Read me for web scraping Financial data from SEC Edgar online database**

**for the US S&P 500 companies:**

1. **Locating, building & accessing content of each company’s filing link from SEC Edgar**

1. Extracting CIK from the SEC website only for each company TIKR recorded previously in the 300 companies’ data collection.

2. Wrote a function to set parameters that allow extracting the yearly filings on SEC Edgar based on our requirement (eg. 10K filing).

3. Create a user-agent for requesting access to the URL to avoid getting HTTP Error 403 forbidden from web-scraping the SEC website.

4. Build the database for filing type, filing date, filing number, partial link to the company’s document, clicking the interactive button to arrive at the link leading to tabular financial data.

5. Identify the content of interest from finding consistent labels in the HTML code of the webpage, XBRL or JSON file (eg. https://www.sec.gov/Archives/edgar/data/320193/000032019321000105/index.json).

6. Request ‘lxml’ content of the filing summary in the custom link built for a company.

1. **Target statement and html tag identification**

7. Within the dozens of filings look for filing statements of interest. While the names of statements vary greatly across companies and years, identified a workaround to only look for keywords in statement titles instead. After lowercasing the titles, looked for keywords or key phrases:  *"cover", "balance sheets", "of operations", "of earning", "cash flow", "income tax", "property, and equipment"*

8.  As going through each row of each financial information entered in the financial statements of a company can be computationally intensive, we further chop-down on the list of financial statement links we record. For context, inspect <https://www.sec.gov/Archives/edgar/data/1265107/000126510719000004/R2.htm> I could parse them if I knew what I was dealing with, a header, section or regular row.

9.  ON inspection, we see that header rows can be identified  because they have a 'th' tag i.e. table headers. Non-current liabilities is a section row because in front of it there is no real data - empty columns on looking it was observed that only a section row had a <strong> tag but the table header in addition to th can also have a <strong> tag inside it. So, the only way to distinguish section and header is that header has both th and <strong> tag but section header only has <strong> tag A normal row does not have either one - th or <strong> So, creating a logic statement with this observation if its a regular row and not a section or a table header

10. Drawing header 1 or header 2 or data based on the type of statement under consideration.

11. Replace using regex symbols like \, $, [, ( and replacing [(] with a minus sign to denote losses in the accounting statements as well as NaNs instead of empty cells.

12. Request access to the link to each financial statement and locating the name of the company to be concatenated to matching financials we are collecting.

1. **Selectively search and locate metrics of interest from different financial statements**

13.  If, ‘operations’ in the statement header 1, capture ‘Operating Costs’ by looking for any strings containing “ 'total operating cost', 'total operating expense' – these also vary company-wise.

14.  NetRevenue from rows containing 'net revenue', 'net sale'

15.  EmployeeWages from the operations financial statement containing 'selling, general and administrative', 'selling, general & administrative'

16. To make things trickier, Amgen and some other companies can have R&D expenses in operations too so looked for 'research and development', 'research & development', 'R&D', 'R & D'

17.  Compiled 12-15 by row combining only for the populated rows and dropping unpopulated rows as well as concatenating columns containing company name and CIK to attribute each value to a company from an year’s filings.

18.  If ‘ tax’ in statement header, TotalUStax is the row containing 'effective tax rate', 'effective rate'

19.  Use regex again to clean out the table content: '\$':'','â€‹':'','%':'' Remove any/all % and $ signs that take up separate columns or within a column preceding the value because we are only studying US$. Also, need to precede the dollar sign with a \ to be identified in regex – needs extensive cleaning and pruning along with transposition.

20.  Capture the dates of metric capture for each

21.  If ‘earning’ in statement title, capture R&D expenses for some other companies that file this information under a different statement and look for values of rows containing 'research and development','research & development', 'R&D','R & D'. Further use regex again for data content cleaning and capture dates

22.  Create a clean dataframe with the metric captured, date1, date2, date3 of value reported, company name and CIK string.

23. The number of columns of filing dates varies greatly over filing metric of interest and the company reporting making row concatenation harder.