**Instructions for running the scripts:**

**Beautifulsoup Scrapper** –

1. Throttle – It adds delay in between the domain access
2. get\_soup – gets a soup object for a input link with optional certificate verification
3. get\_status\_code- checks for website with no response and returns error code
4. prepare\_zip - It is for issuu\_scraper , generates a zip for all the pages of PDF
5. issuu\_scraper – It inputs pdf\_url and pages as input , downloads all the pages and creates zip with prepare\_zip
6. extract\_date – It Is a date parser , returns date for any soup object
7. find\_internal\_url – takes url , depth and max\_depth as inputs ,looks for href links or path in the url and returns list of URLs
8. get\_scrap – gets all text from by parsing each of URLs

Instruction for running the script –

url = "" // Link for scraping , it works best when URL has href links , it returns empty if it doesn’t

Depth = 0 //

all\_urls = find\_internal\_url(url,0,2) // returns all the internal urls inside a url , depth signifies the recursive scraping urls inside of url

1. Replace the URL link , check for href before in the browser (with inspect element)
2. Depth 0 is recommended , sometimes some website contains a lot links, it might crash.
3. Set the csv name and encoding
4. For issuu scraper , set the PDF url and comment the other URL, find\_internal\_url and depth

**Beautifulsoup wrangler** –

1.It removes some special symbols or characters

2.converts string date to date time object

3.checks for duplicates

4. removes tags if any

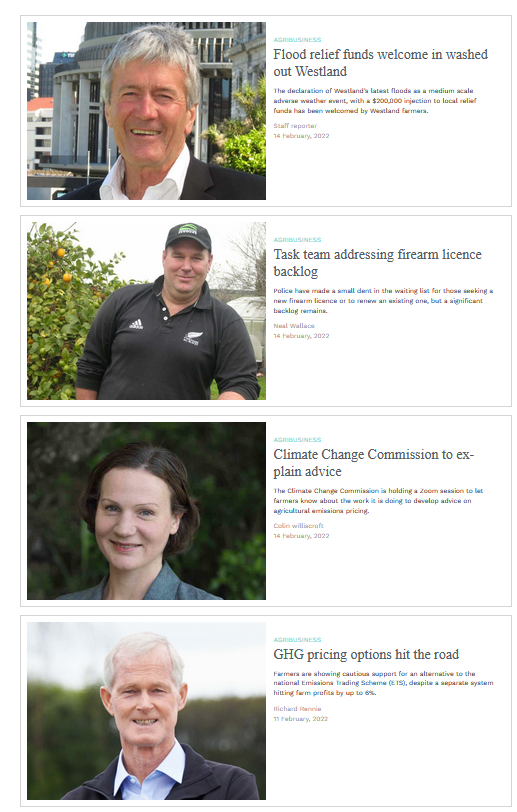
5.rename columns

Instruction for running the script -

Set the filename and run the script. Just make sure file csv files are in the current directory or directory can be set in the glob of the script.

**Selenium Scrapper** –

1. First download the ChromeDriver. Users need to download ChromeDriver for their respective Operating systems from [this link](https://chromedriver.chromium.org/downloads). Refer to the image to get a better sense of ChromeDriver downloads.
2. Now copy to path where selenium is installed, generally in python folder and set the path in environment variables of system properties.

Ad windows can block , just close while loading for scraping to continue.

Ideal link should contain list of stories with header. Example can be seen in the photo.

Each header story class will be different , to know the header class, right click on the header and inspect element.

The class header can be replaced in the get\_link\_headers function. Nested for loop if the header contains additional tags. Its function is to scrap the text of each header.

These 2 instance the BY.PARTIAL\_LINK\_TEXT can be replaced with By.LINKED\_TEXT if the header is a link or same if it’s a header.

element = ec.presence\_of\_element\_located(

(By.PARTIAL\_LINK\_TEXT, i))

element = driver.find\_element(By.PARTIAL\_LINK\_TEXT, i)

Adjusting the sleep time may depend on website.

At last, set the file name.

Selenium wrangler does the same as beautifulsoup wrangler , just make sure all the csvs in the current directory or set the directory in glob.

**Pre-processing** –

It reads all data[0-9] output files from the beautifulsoup and selenium wrangler and merge it.

It saves a file with name 'preprocessed.csv' after stemming , lematisation and punctuation removal.

**Feature Engineering** –

It read the pre-processed file and calculates the word count, sentence count with sentimental score.

It saves the dataframe with name 'fengineering.csv'.

**Modelling** –

It reads the file from feature engineering, generates the keywords , remove stop words , get all the occurrence dates from different article, and all the positive sentiment scores.

It contains list of custom stop words , adding more word that are not relevant will make a more refined list.

**Visualisation extract** –

Reads the csv file from modelling, it changes the column types , calculates the average sentiment score and the occurrence period.

There is category that to be set . default is global. It can be changed based on data source.

The resultant csv files from this script can be used in tableau dashboard.

Scrappers -> Wranglers -> Pre-processing -> Feature engineering -> Modelling -> Visualisation extract