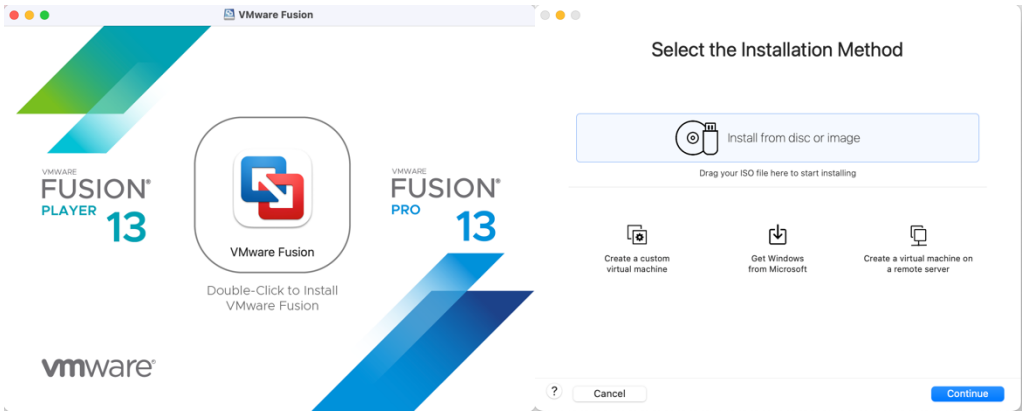
Hanlu Ma (USC ID: 1392-9443-71)

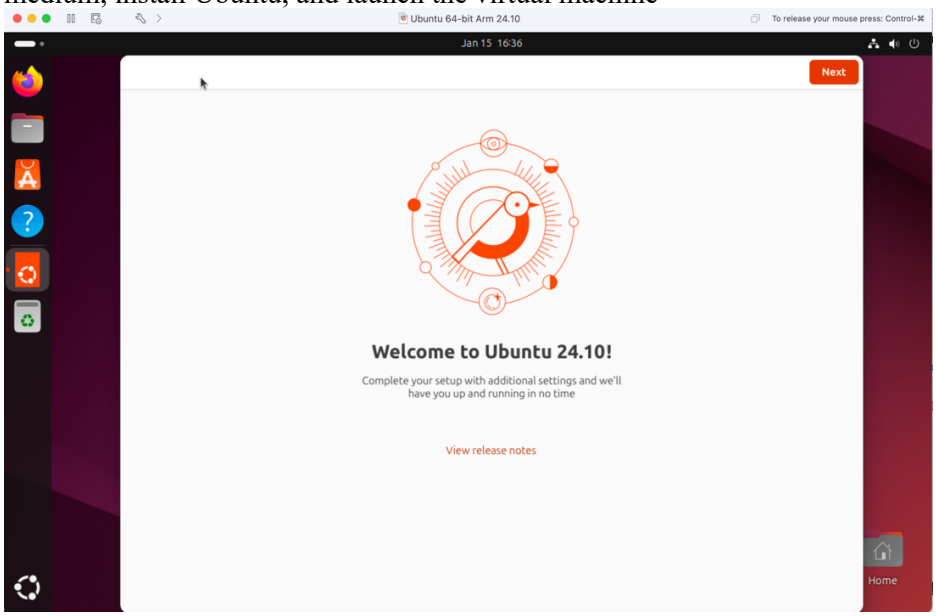
Data Science Professional Practicum (DSCI 560)

Laboratory Assignment 1

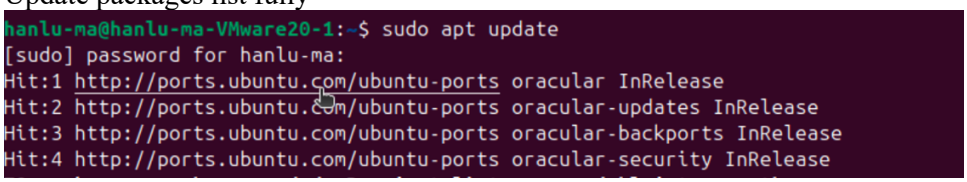
1. Installation and Setup
    a. Install VMware
    VMware is downloaded and is waiting to be launched fully through the steps in the next section.



    b. Download Ubuntu ISO Image
        i. Create a new virtual machine by using Ubuntu ISO image as the installation medium, install Ubuntu, and launch the virtual machine



    c. Install Python on Linux
        i. Update packages list fully



ii. Install python3 and confirm its installation

```
hanlu-ma@hanlu-ma-VMware20-1:~$ sudo apt install python3
python3 is already the newest version (3.12.6-0ubuntu1).
python3 set to manually installed.
The following packages were automatically installed and are no longer required:
  linux-headers-6.11.0-8          linux-modules-extra-6.11.0-8-generic
  linux-headers-6.11.0-8-generic  linux-tools-6.11.0-8
  linux-modules-6.11.0-8-generic  linux-tools-6.11.0-8-generic
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 70
hanlu-ma@hanlu-ma-VMware20-1:~$ python3 --version
Python 3.12.7
```

iii. Install pip and verify its installation

```
hanlu-ma@hanlu-ma-VMware20-1:~$ sudo apt install python3-pip
hanlu-ma@hanlu-ma-VMware20-1:~$ pip3 --version
pip 24.2 from /usr/lib/python3/dist-packages/pip (python 3.12)
```

2. Get Familiar with Linux and Python
   a. Playing around with Linux Terminal
      i. All the requested directories, subdirectories, and files are created and located as requested.

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop$ mkdir HanluMa_1392944371
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop$ cd HanluMa_1392944371
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371$ mkdir data
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371$ mkdir scripts
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371$ touch task_1.py
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371$ cd scripts
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ touch task_1.p
y
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ ls
task_1.py
```

   b. A basic Python Script
      i. Program the basic python script for ‘task_1.py’

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ nano task_1.p
y
```

```
hanlu-ma@hanlu-ma-VMware20-1: ~/Desktop/HanluMa_1392944371/scripts
  GNU nano 8.1                          task_1.py
name = input("Please enter your name:")
print(f"Hello, {name}!")
```

      ii. Run and test the python code, which shows the desired output.

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ python3 task_
1.py
Please enter your name:Hanlu
Hello, Hanlu!
```

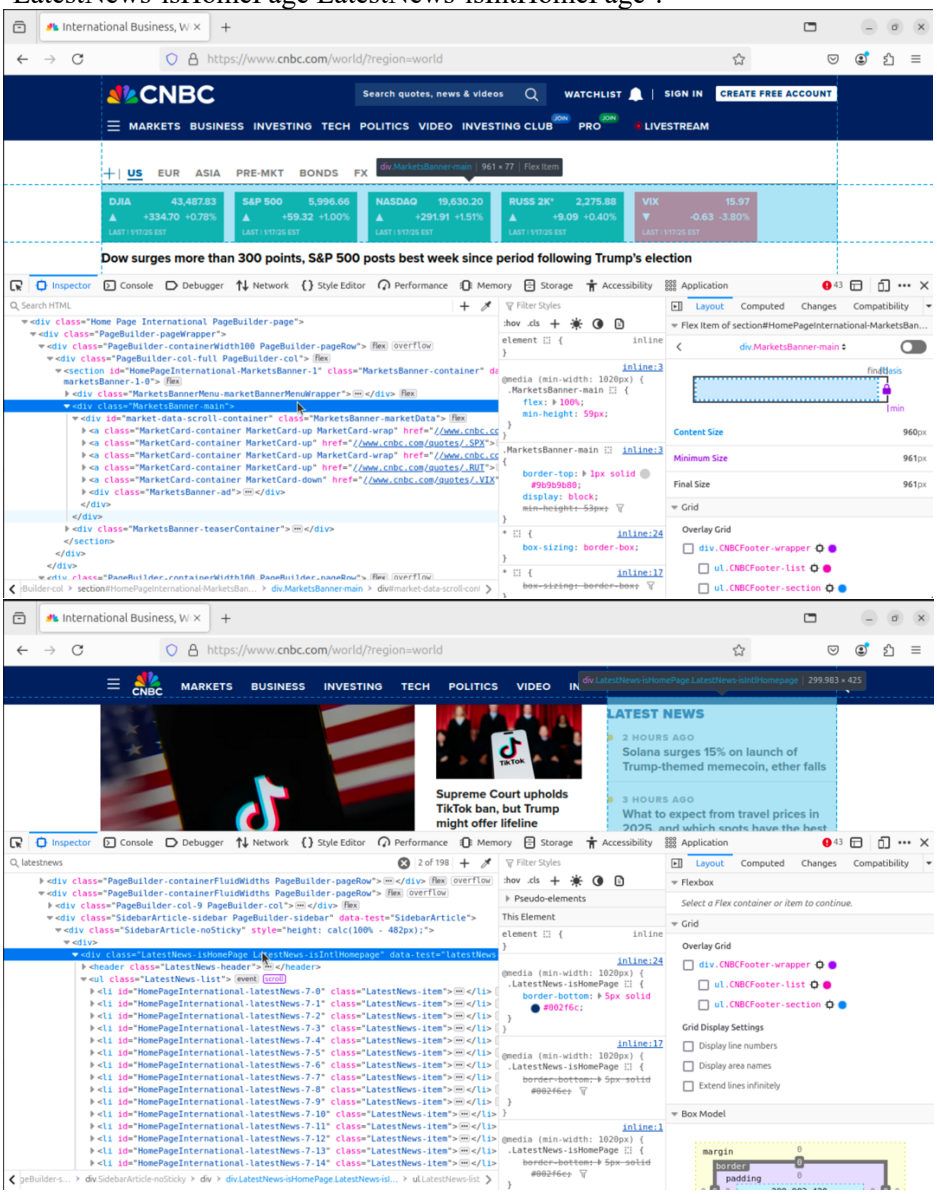
   c. Python Web-scraping Task
      i. Create ‘web_scraper.py’ file

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ touch web_scr
aper.py
```

      ii. Install libraries of requests, bs4, and selenium. I only installed requests and bs4 at first. However, when I start working on the data filtering script, I realized that the market data was not being parsed properly. After doing some research online, I learned that the market data was not parsed due to the highly dynamic nature. Therefore, I installed ‘selenium’ as this is the common tool.

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ sudo apt inst
all python3-requests
[sudo] password for hanlu-ma:
python3-requests is already the newest version (2.32.3+dfsg-1ubuntu1).
python3-requests set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 74
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ sudo apt inst
all python3-bs4
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ sudo apt inst
all python3-selenium
```

iii. Analyze the HTML structure and find the corresponding tags for the market banner and latest news. Specifically, the market banner and all the relevant details (i.e., symbol, stock position, change pct) are all shown within the div with the class of 'MarketBanner-main'. The latest news and all the revelant details (i.e., timestamp, title, link) are all shown within the div with the class of 'LatestNews-isHomePage LatestNews-isIntHomePage'.



iv. Create folders under data folder

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/data$ mkdir raw_data
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/data$ mkdir processed_
data
```

v. Write the 'web_scraper.py' script to scrap the webpage.

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ nano webscrap
er.py
```

Initially, I parsed the webpage and obtained the content through 'get' method in requests and BeautifulSoup.

hanlu-ma@hanlu-ma-VMware20-1: ~/Desktop/HanluMa_1392944371/scripts

GNU nano 8.1 web_scraper_requests.py

```
import requests
from bs4 import BeautifulSoup

url = "https://www.cnbc.com/world/?region=world"
response = requests.get(url)

if response.status_code == 200:
        web_data = BeautifulSoup(response.text, "html.parser")
        with open('../data/raw_data/web_data.html', 'w', encoding='utf-8') as file:
                file.write(web_data.prettify())
        with open('../data/raw_data/web_data.html', 'r', encoding='utf-8') as file:
                lines = file.readlines()
                print("\n".join(lines))
```

After realizing that the market data cannot be parsed, I start to utilize selenium to do web-scrapping. However, after trying out all the suggested methods in piazza, I still encountered obstacles in connecting to Firefox and Chrome after working around with numerous setups and parameters. Therefore, even though I did not reach my expectation to parse the webpage by using selenium, I still want to include screenshots of my code for web-scrapping by using Firefox and Chrome as the browser driver to showcase my efforts and tires.

'web_scraper_firefox.py':

hanlu-ma@hanlu-ma-VMware20-1: ~/Desktop/HanluMa_1392944371/scripts

GNU nano 8.1 web_scraper_firefox.py

```
from selenium import webdriver
from selenium.webdriver.firefox.service import Service
from selenium.webdriver.firefox.options import Options
from selenium.webdriver.firefox.firefox_profile import FirefoxProfile
import logging
from bs4 import BeautifulSoup
import time

logging.basicConfig(level=logging.DEBUG)
url = "https://www.cnbc.com/world/?region=world"
service = Service("/usr/local/bin/geckodriver")
service.log_path= '/usr/local/bin/geckodriver.log'
profile_path = '/home/hanlu-ma/snap/firefox/common/.cache/mozilla/firefox/4kdg8sho.selenium_profile'
profile = FirefoxProfile(profile_path)
profile.set_preference("browser.startup.page", 0)
options = Options()
options.headless = False
options.profile = profile
driver = webdriver.Firefox(service=service, options = options)
driver.get(url)
time.sleep(1000)
content = driver.page_source
web_data = BeautifulSoup(content, "html.parser")

with open('../data/raw_data/web_data.html', 'w', encoding='utf-8') as file:
        file.write(web_data.prettify())
with open('../data/raw_data/web_data.html', 'r', encoding='utf-8') as file:
        lines = file.readlines()
        print("\n".join(lines))

driver.close()
```

'web_scraper_chrome.py':

```
hanlu-ma@hanlu-ma-VMware20-1: ~/Desktop/HanluMa_1392944371/scripts
 GNU nano 8.1                                web_scraper_chrome.py
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
import logging
from bs4 import BeautifulSoup
import time

logging.basicConfig(level=logging.DEBUG)

url = "https://www.cnbc.com/world/?region=world"

service = Service("/usr/local/bin/chromedriver")

options = Options()
options.headless = False
options.add_argument("--disable-gpu")
options.add_argument("--no-sandbox")
options.add_argument("--disable-dev-shm-usage")
options.add_argument("--disable-extensions")
options.add_argument("--disable-notifications")
options.add_argument("--start-maximized")

driver = webdriver.Chrome(service=service, options = options)
driver.get(url)
time.sleep(10)
content = driver.page_source
web_data = BeautifulSoup(content, "html.parser")
with open('../data/raw_data/web_data.html', 'w', encoding='utf-8') as file:
        file.write(web_data.prettify())
with open('../data/raw_data/web_data.html', 'r', encoding='utf-8') as file:
        lines = file.readlines()
        print("\n".join(lines))

driver.close()
```

vi. Rationale of the scripts:
For 'requests' web-scrapping method, 'get' method in requests library prompt us to send HTTP requests to the website and receive the HTML content as the response. For 'selenium' web-scrapping method, the script retrieves the HTML content after JavaScript has executed and the content is fully loaded. Then, 'BeautifulSoup' library helps to parse the HTML content and extract the desired content. Eventually, the script saves these data into files.

vii. Outcome of the web-scrapping (i.e., use the requests library):

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/data/raw_data$ head -n 10 web_data.html
<!DOCTYPE html>
<html itemscope="" itemtype="https://schema.org/WebPage" lang="en" prefix="og=https://ogp.me/ns#">
 <head>
  <meta content="website" property="og:type"/>
  <meta content="International: Top News And Analysis" property="og:title"/>
  <meta content="CNBC International is the world leader for news on business, technology, China, trade, o
il prices, the Middle East and markets." property="og:description"/>
  <meta content="https://www.cnbc.com/world/" property="og:url"/>
  <meta content="CNBC" property="og:site_name"/>
  <meta content="max-image-preview:large" name="robots"/>
  <meta content="telephone=no" name="format-detection"/>
```

d. Data Filtering Task

i. Rationale of the script
On the high level, we read the content of the stored file previously through 'BeautifulSoup'. Then we locate and filter the data we want. Specifically, we zoom into the sections of HTML, find the parts (i.e., div, href, title, etc) with corresponding key words in class, and store relevant details within these parts into variables. Eventually, we iterate through every element in these variables and store them sequentially into csv.

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ touch data_filter.py
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ nano data_filter.py
```

hanlu-ma@hanlu-ma-VMware20-1: ~/Desktop/HanluMa_1392944371/scripts

GNU nano 8.1 data_filter.py *

```
from bs4 import BeautifulSoup
import csv

with open('../data/raw_data/web_data.html', 'r', encoding='utf-8') as file:
        web = file.read()
web_data = BeautifulSoup(web, 'html.parser')

print('Filtering fields for market data')
symbol = []
stock_position = []
changes_pct =[]

symbols = web_data.find_all(class_ = 'MarketCard-symbol')
stock_positions = web_data.find_all(class_ = 'MarketCard-stockPosition')
changes_pcts = web_data.find_all(class_ = 'MarketCard-changesPct')

print('Storing market data')
for i in symbols:
    symbol.append(i.get_text().strip())
for i in stock_positions:
    stock_position.append(i.get_text().strip())
for i in changes_pcts:
    changes_pct.append(i.get_text().strip())

market_data = zip(symbol, stock_position, changes_pct)

print('Create CSV for market data')
with open('../data/processed_data/market_data.csv', 'w', newline='', encoding='utf-8') as file:
        market_data_writer = csv.writer(file)
        market_data_writer.writerow(['Market Card Symbol', 'Market Card Stock Position', 'Market Card Changes PCT'])
        market_data_writer.writerows(market_data)

print('Filtering fields for latest news')
timestamp = []
title = []
link = []

timestamps = web_data.find_all(class_ = 'LatestNews-timestamp')
headlines = web_data.find_all(class_ = 'LatestNews-headline', title = True, href = True)

print('Storing news data')
for i in timestamps:
    timestamp.append(i.get_text().strip())
for headline in headlines:
    title.append(headline['title'])
    link.append(headline['href'])

news_data = zip(timestamp, title, link)

print('Create CSV for news data')
with open('../data/processed_data/news_data.csv', 'w', newline='', encoding='utf-8') as file:
        news_data_writer = csv.writer(file)
        news_data_writer.writerow(['Latest News Timestamp', 'Latest News Title', 'Latest News Link'])
        news_data_writer.writerows(news_data)
```

ii. Execution and output

Because requests library does not work well with the dynamic sections, we did not obtain the market data successfully. However, except this characteristic, the methods of parsing market data and news data are highly similar. We successfully obtained the news data, which also reflects the feasibility of the script.

```
hanlu-ma@hanlu-ma-VMware20-1:~/Desktop/HanluMa_1392944371/scripts$ python3 data_
filter.py
Filtering fields for market data
Storing market data
Create CSV for market data
Filtering fields for latest news
Storing news data
Storing market data
Create CSV for news data
```

Hanlu Ma (USC ID: 1392-9443-71)

parts of news data:

```
Latest News Timestamp,Latest News Title,Latest News Link
3 Hours Ago,TikTok says it will go dark on Sunday unless Biden intervenes,https://www.cnbc.com/2025/01/17/tiktok-says-it-will-go-dark-on-sunday-unless-biden-intervenes.html
5 Hours Ago,Cramer's Lightning Round: Adobe is a buy,https://www.cnbc.com/2025/01/17/cramers-lightning-round-adobe-is-a-buy.html
5 Hours Ago,"Jim Cramer parses big bank earnings, says stocks have 'quite a bit more upside'",https://www.cnbc.com/2025/01/17/jim-cramer-parses-bank-earnings-stocks-have-quite-a-bit-more-upside.html
5 Hours Ago,"Cramer's week ahead: Inauguration, Procter & Gamble, American Express earnings",https://www.cnbc.com/2025/01/17/cramers-week-ahead-inauguration-procter-gamble-american-express-earnings.html
8 Hours Ago,"DOJ sues Walgreens, alleging it filled millions of unlawful prescriptions",https://www.cnbc.com/2025/01/17/doj-sues-walgreens-prescriptions.html
8 Hours Ago,Returns on Treasury bonds haven't been this poor in the last 90 years,https://www.cnbc.com/2025/01/17/returns-on-treasury-bonds-havent-been-this-poor-in-the-last-90-years.html
8 Hours Ago,"These stocks offer dividend growth and have cash flow to back it up, Wolfe says",https://www.cnbc.com/2025/01/17/these-stocks-offer-high-dividend-growth-and-have-the-cash-flows-to-back-it-up-wolfe-says.html
8 Hours Ago,Here's what it will take for Apple to get out of its 2025 funk,https://www.cnbc.com/2025/01/17/heres-what-it-will-take-for-apple-to-get-out-of-its-2025-funk.html
```

market data:

```
Market Card Symbol,Market Card Stock Position,Market Card Changes PCT
```