Reading Stock Data using Pandas Datareader

Takao Oba

Assisted by quantpy

I will aim to import financial stock data through pandas_datareader

In [1]:
```
!pip install datetime
```

```
Collecting datetime
  Downloading DateTime-4.9-py2.py3-none-any.whl (52 kB)
     |████████████████████████████████| 52 kB 624 kB/s eta 0:00:01
Requirement already satisfied: pytz in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from datetime) (2021.3)
Requirement already satisfied: zope.interface in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from datetime) (5.4.0)
Requirement already satisfied: setuptools in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from zope.interface->dateti
me) (58.0.4)
Installing collected packages: datetime
Successfully installed datetime-4.9
```

In [2]:
```
!pip install pandas_datareader
```

```
Collecting pandas_datareader
  Downloading pandas_datareader-0.10.0-py3-none-any.whl (109 kB)
     |████████████████████████████████| 109 kB 733 kB/s eta 0:00:01
Requirement already satisfied: requests>=2.19.0 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from pandas_datareade
r) (2.26.0)
Requirement already satisfied: pandas>=0.23 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from pandas_datareader)
(1.3.4)
Requirement already satisfied: lxml in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from pandas_datareader) (4.6.3)
Requirement already satisfied: python-dateutil>=2.7.3 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from pandas>=0.
23->pandas_datareader) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from pandas>=0.23->pandas
_datareader) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from pandas>=0.23->panda
s_datareader) (1.20.3)
Requirement already satisfied: six>=1.5 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from python-dateutil>=2.7.3->
pandas>=0.23->pandas_datareader) (1.16.0)
Requirement already satisfied: idna<4,>=2.5 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from requests>=2.19.0->pa
ndas_datareader) (3.2)
Requirement already satisfied: certifi>=2017.4.17 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from requests>=2.1
9.0->pandas_datareader) (2021.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from request
```

```
s>=2.19.0->pandas_datareader) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from requests>=
2.19.0->pandas_datareader) (1.26.7)
Installing collected packages: pandas-datareader
Successfully installed pandas-datareader-0.10.0
```

In [3]:
```python
import datetime as dt
import pandas as pd
from pandas_datareader import data as pdr
```

In [4]:
```python
# Specifying date range that will be used for analysis
# Begin by creating start and end dates using pythons datetime module

# If we wanted to specify the start time:
end = dt.datetime.now()
start = dt.datetime(2000, 1, 1) # start from 1/1/2000
start,end

# If we wanted to retrieve start time that is given days away from end time do the following:
end = dt.datetime.now()
start = end - dt.timedelta(days = 7000) # end minus 5000 days
start,end
```

Out[4]:
```
(datetime.datetime(2003, 10, 31, 23, 34, 46, 769525),
 datetime.datetime(2022, 12, 30, 23, 34, 46, 769525))
```

In [5]:
```python
# Next, I will select stocks/tickers that I would like to analyze
stocks = ['AAPL', 'NFLX', 'AMZN', 'MSFT']

# if we are dealing with for example australian stocks, the following can be used:
# stocks = [i + '.AX' for i in stocklist]
stocks
```

Out[5]:
```
['AAPL', 'NFLX', 'AMZN', 'MSFT']
```

Calling Pandas_Datareader DataReader module:

Two ways of doing this:

In [6]:
```python
# pdr.DataReader(stocks,'yahoo', start, end) # option 1
# pdr.get_data_yahoo(stocks, start, end) # option 2
```

In [7]:
```python
!pip install yfinance
import yfinance as yfin
yfin.pdr_override()
```

```
Collecting yfinance
  Downloading yfinance-0.2.3-py2.py3-none-any.whl (50 kB)
     |████████████████████████████████| 50 kB 461 kB/s eta 0:00:01
Collecting frozendict>=2.3.4
  Downloading frozendict-2.3.4-cp39-cp39-macosx_10_9_x86_64.whl (33 kB)
Requirement already satisfied: html5lib>=1.1 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from yfinance) (1.1)
Requirement already satisfied: appdirs>=1.4.4 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from yfinance) (1.4.4)
Collecting multitasking>=0.0.7
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Requirement already satisfied: numpy>=1.16.5 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from yfinance) (1.20.3)
Collecting lxml>=4.9.1
  Downloading lxml-4.9.2-cp39-cp39-macosx_10_15_x86_64.whl (4.8 MB)
     |████████████████████████████████| 4.8 MB 27 kB/s eta 0:00:014     |████████████████████▉            | 2.9 MB 304 kB/s eta
0:00:06     |████████████████████▍          | 3.4 MB 507 kB/s eta 0:00:03
Requirement already satisfied: pandas>=1.3.0 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from yfinance) (1.3.4)
Collecting pytz>=2022.5
  Downloading pytz-2022.7-py2.py3-none-any.whl (499 kB)
     |████████████████████████████████| 499 kB 329 kB/s eta 0:00:01
Requirement already satisfied: requests>=2.26 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from yfinance) (2.26.0)
Requirement already satisfied: cryptography>=3.3.2 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from yfinance) (3.
4.8)
Collecting beautifulsoup4>=4.11.1
  Downloading beautifulsoup4-4.11.1-py3-none-any.whl (128 kB)
     |████████████████████████████████| 128 kB 442 kB/s eta 0:00:01
Requirement already satisfied: soupsieve>1.2 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from beautifulsoup4>=4.1
1.1->yfinance) (2.2.1)
Requirement already satisfied: cffi>=1.12 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from cryptography>=3.3.2->y
finance) (1.14.6)
Requirement already satisfied: pycparser in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from cffi>=1.12->cryptograph
y>=3.3.2->yfinance) (2.20)
Requirement already satisfied: six>=1.9 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from html5lib>=1.1->yfinance)
(1.16.0)
Requirement already satisfied: webencodings in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from html5lib>=1.1->yfina
nce) (0.5.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from pandas>=1.
3.0->yfinance) (2.8.2)
Requirement already satisfied: idna<4,>=2.5 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from requests>=2.26->yfin
ance) (3.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from requests>=
2.26->yfinance) (1.26.7)
Requirement already satisfied: certifi>=2017.4.17 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from requests>=2.26
->yfinance) (2021.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from request
s>=2.26->yfinance) (2.0.4)
Installing collected packages: pytz, multitasking, lxml, frozendict, beautifulsoup4, yfinance
  Attempting uninstall: pytz
    Found existing installation: pytz 2021.3
    Uninstalling pytz-2021.3:
```

```
    Successfully uninstalled pytz-2021.3
  Attempting uninstall: lxml
    Found existing installation: lxml 4.6.3
    Uninstalling lxml-4.6.3:
      Successfully uninstalled lxml-4.6.3
  Attempting uninstall: beautifulsoup4
    Found existing installation: beautifulsoup4 4.10.0
    Uninstalling beautifulsoup4-4.10.0:
      Successfully uninstalled beautifulsoup4-4.10.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is th
e source of the following dependency conflicts.
conda-repo-cli 1.0.4 requires pathlib, which is not installed.
Successfully installed beautifulsoup4-4.11.1 frozendict-2.3.4 lxml-4.9.2 multitasking-0.0.11 pytz-2022.7 yfinance-0.2.3
```

In [8]:
```
df = pdr.get_data_yahoo(stocks, start, end)
df.head()
```

```
[**********100%**********]  4 of 4 completed

1 Failed download:
- AMZN: OperationalError('database is locked')
```

Out[8]:

	Adj Close				Close				High		...	Low				O
	AAPL	AMZN	MSFT	NFLX	AAPL	AMZN	MSFT	NFLX	AAPL	AMZN	...	MSFT	NFLX	AAPL	AMZN	MSFT	N
Date																	
2003-11-03 00:00:00	0.352384	NaN	16.834877	4.175000	0.413393	NaN	26.68	4.175000	0.416071	NaN	...	26.290001	4.085714	0.407679	NaN	26.350000	4.099
2003-11-04 00:00:00	0.348730	NaN	16.449970	4.257143	0.409107	NaN	26.07	4.257143	0.412500	NaN	...	26.010000	4.131429	0.411964	NaN	26.590000	4.146
2003-11-05 00:00:00	0.350557	NaN	16.468901	4.194286	0.411250	NaN	26.10	4.194286	0.413036	NaN	...	26.000000	4.139286	0.407500	NaN	26.150000	4.232
2003-11-06 00:00:00	0.351927	NaN	16.550920	3.630714	0.412857	NaN	26.23	3.630714	0.413393	NaN	...	26.000000	3.573571	0.409107	NaN	26.260000	3.942
2003-11-07 00:00:00	0.342489	NaN	16.468901	3.348571	0.401786	NaN	26.10	3.348571	0.415000	NaN	...	26.030001	3.301429	0.414107	NaN	26.379999	3.654

5 rows × 24 columns

In [9]:
```
df.index
```

Out[9]:
```
Index(['2003-11-03 00:00:00', '2003-11-04 00:00:00', '2003-11-05 00:00:00',
       '2003-11-06 00:00:00', '2003-11-07 00:00:00', '2003-11-10 00:00:00',
       '2003-11-11 00:00:00', '2003-11-12 00:00:00', '2003-11-13 00:00:00',
       '2003-11-14 00:00:00',
       ...
       '2022-12-16 00:00:00', '2022-12-19 00:00:00', '2022-12-20 00:00:00',
       '2022-12-21 00:00:00', '2022-12-22 00:00:00', '2022-12-23 00:00:00',
       '2022-12-27 00:00:00', '2022-12-28 00:00:00', '2022-12-29 00:00:00',
       '2022-12-30 00:00:00'],
      dtype='object', name='Date', length=4824)
```

In [10]:
```
df.columns # notice there are multi-index which makes this data frame complicated
```

Out[10]:
```
MultiIndex([('Adj Close', 'AAPL'),
            ('Adj Close', 'AMZN'),
            ('Adj Close', 'MSFT'),
            ('Adj Close', 'NFLX'),
            (    'Close', 'AAPL'),
            (    'Close', 'AMZN'),
            (    'Close', 'MSFT'),
            (    'Close', 'NFLX'),
            (     'High', 'AAPL'),
            (     'High', 'AMZN'),
            (     'High', 'MSFT'),
            (     'High', 'NFLX'),
            (      'Low', 'AAPL'),
            (      'Low', 'AMZN'),
            (      'Low', 'MSFT'),
            (      'Low', 'NFLX'),
            (     'Open', 'AAPL'),
            (     'Open', 'AMZN'),
            (     'Open', 'MSFT'),
            (     'Open', 'NFLX'),
            (   'Volume', 'AAPL'),
            (   'Volume', 'AMZN'),
            (   'Volume', 'MSFT'),
            (   'Volume', 'NFLX')],
           )
```

Accessing attributes

In [11]:
```
# subsections of the data frame
Close = df.Close

Close.head()
```

Out[11]:

	AAPL	AMZN	MSFT	NFLX

Date				
2003-11-03 00:00:00	0.413393	NaN	26.68	4.175000
2003-11-04 00:00:00	0.409107	NaN	26.07	4.257143
2003-11-05 00:00:00	0.411250	NaN	26.10	4.194286
2003-11-06 00:00:00	0.412857	NaN	26.23	3.630714
2003-11-07 00:00:00	0.401786	NaN	26.10	3.348571

In [12]:
```
# Use built in describe to gain insights for data frame
```

In [13]:
```
Close.describe()
```

Out[13]:

	AAPL	AMZN	MSFT	NFLX
count	4824.000000	0.0	4824.000000	4824.000000
mean	35.843115	NaN	76.447357	128.952818
std	45.028114	NaN	80.421484	170.646501
min	0.351786	NaN	15.150000	1.290000
25%	5.008929	NaN	27.120001	4.585714
50%	19.374821	NaN	33.759998	36.720001
75%	42.771251	NaN	93.177498	200.365005
max	182.009995	NaN	343.109985	691.690002

In [14]:
```
# Change percentile
Close.describe(percentiles = [0.1, 0.5, 0.9])
```

Out[14]:

	AAPL	AMZN	MSFT	NFLX
count	4824.000000	0.0	4824.000000	4824.000000
mean	35.843115	NaN	76.447357	128.952818
std	45.028114	NaN	80.421484	170.646501
min	0.351786	NaN	15.150000	1.290000
10%	1.867035	NaN	25.043000	3.180429
50%	19.374821	NaN	33.759998	36.720001
90%	125.859250	NaN	227.354001	391.393994
max	182.009995	NaN	343.109985	691.690002

In [15]:
```
# Note how we inputted a 100 days, but count is 69
# This is because there are weekends and holidays and not all 100 days are trading days
Close[Close.index > end - dt.timedelta(days = 100)].describe(percentiles = [0.1, 0.5, 0.9])
```

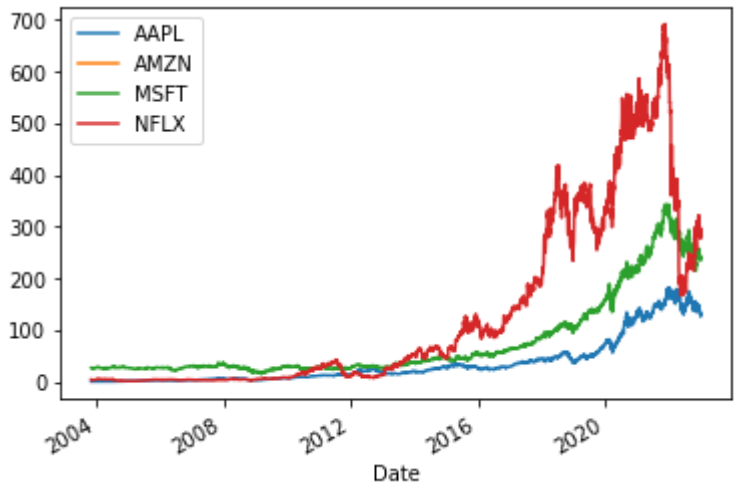
Out[15]:

	AAPL	AMZN	MSFT	NFLX
count	70.000000	0.0	70.000000	70.000000
mean	143.423142	NaN	239.761714	275.762286
std	6.728814	NaN	9.044087	30.135585
min	126.040001	NaN	214.250000	214.289993
10%	132.362996	NaN	227.757996	229.998000
50%	144.040001	NaN	240.994995	286.115005
90%	150.800005	NaN	250.245998	310.493008
max	155.740005	NaN	257.220001	320.410004

Plotting with matplotlib and plotly

In [16]:
```
Close.plot()
```

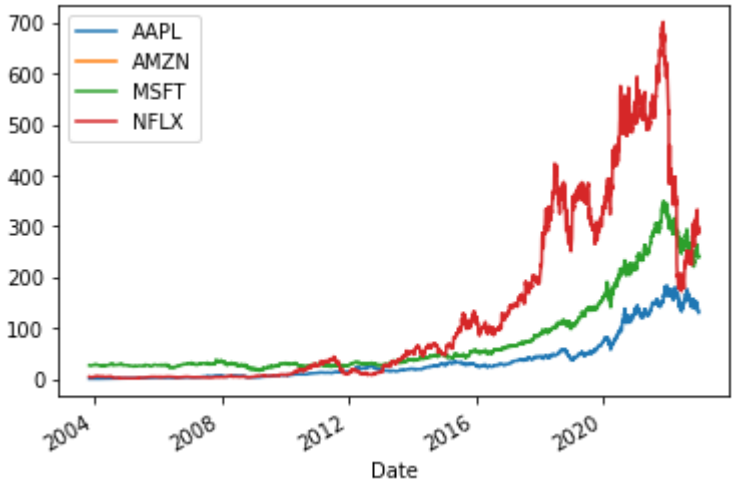
Out[16]:
```
<AxesSubplot:xlabel='Date'>
```



In [17]:
```
# for daily high values
high = df.High
```

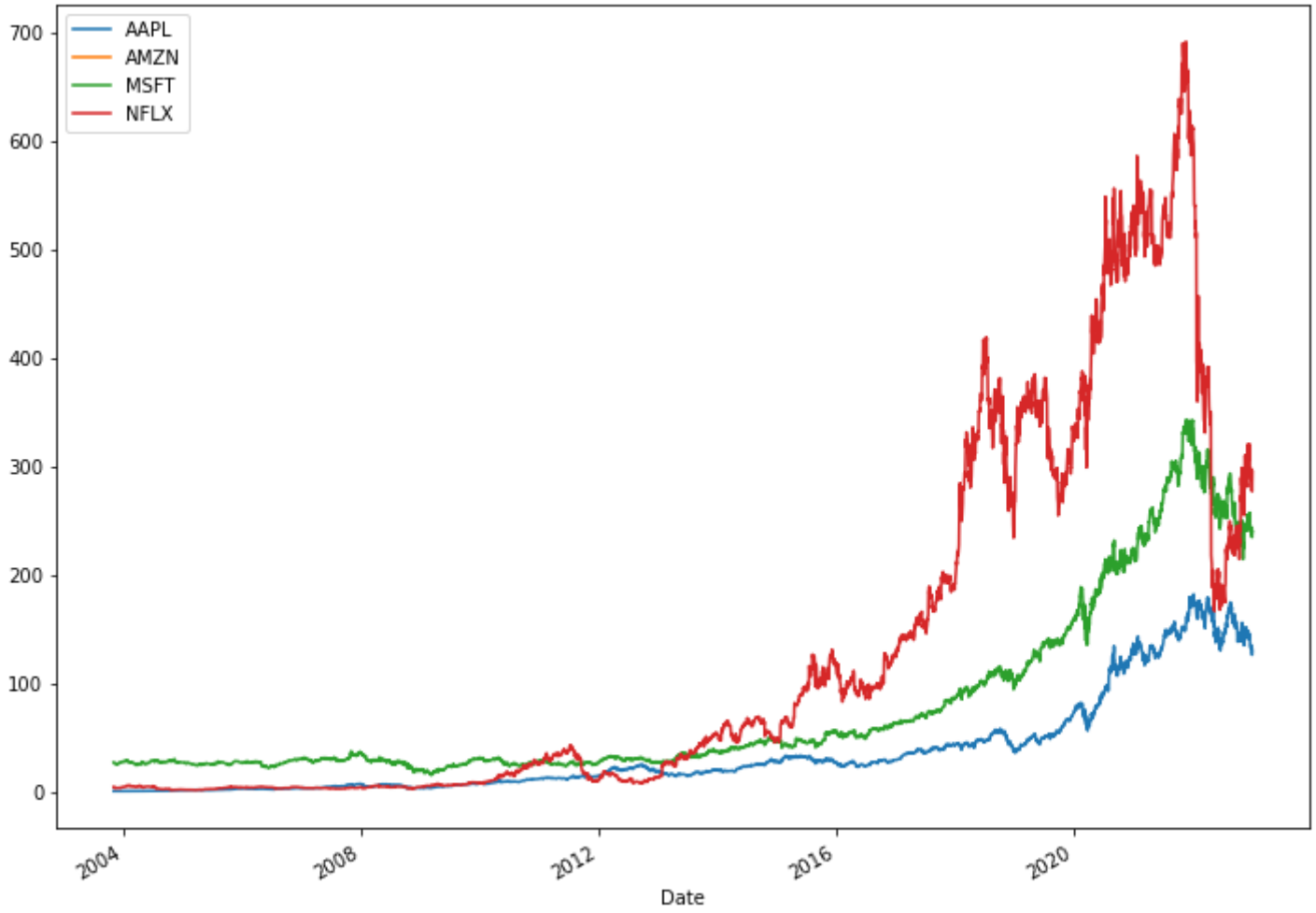
```
high.plot()
# the plot below is very similar to the plot above.
```

Out[17]:
```
<AxesSubplot:xlabel='Date'>
```



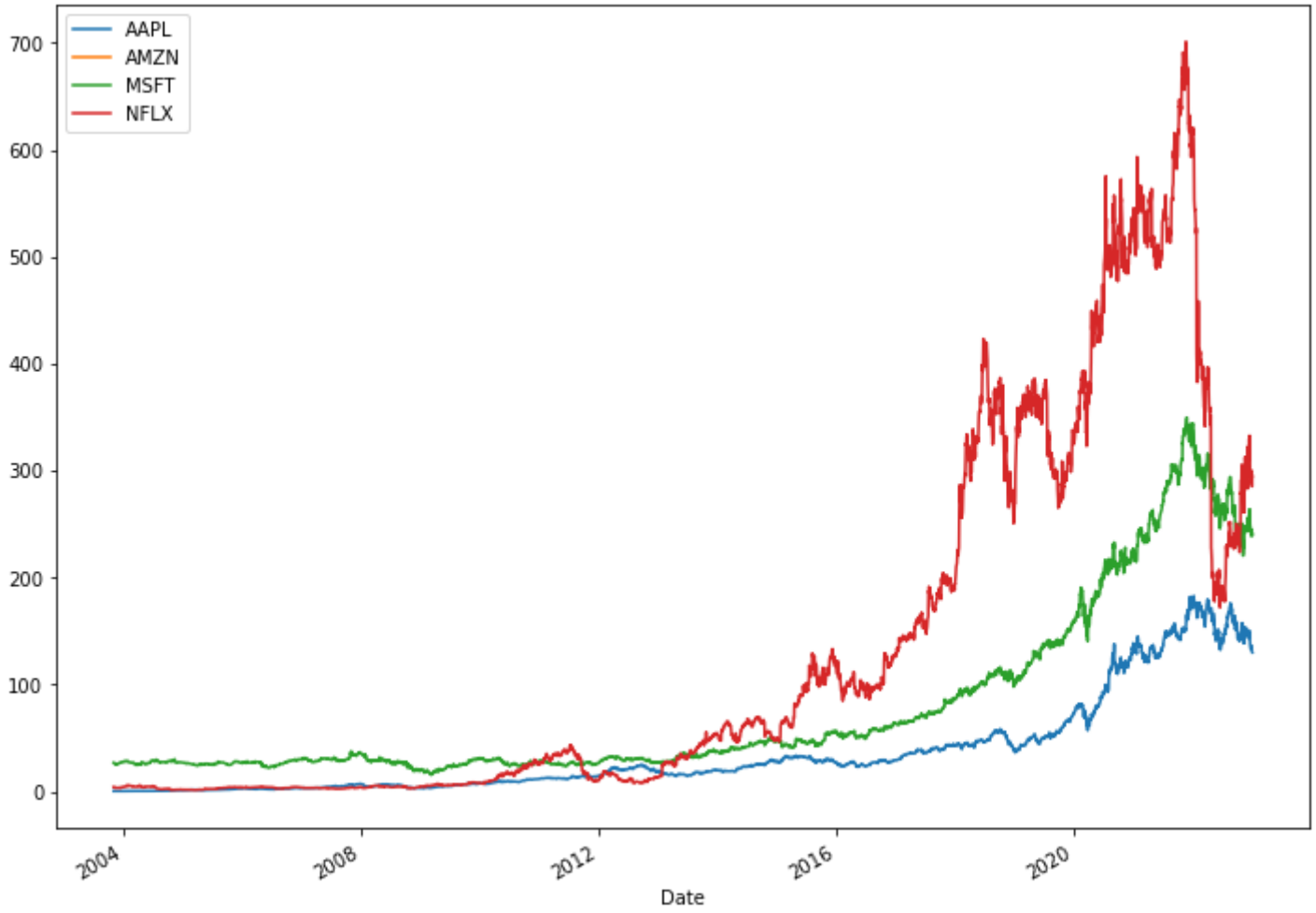
In [18]:
```
Close.plot(figsize = (12,9))
```

Out[18]:
```
<AxesSubplot:xlabel='Date'>
```



In [19]:
```
high.plot(figsize = (12,9))
```

Out[19]:
```
<AxesSubplot:xlabel='Date'>
```



Matplotlib can easily plot these time-series data

However, it is not the most practical and interactive

Use alternative of plotly

In [20]:
```
!pip install plotly
```

```
Collecting plotly
  Downloading plotly-5.11.0-py2.py3-none-any.whl (15.3 MB)
     |████████████████████████████████| 15.3 MB 1.5 MB/s eta 0:00:01
Collecting tenacity>=6.2.0
  Downloading tenacity-8.1.0-py3-none-any.whl (23 kB)
```

```
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.11.0 tenacity-8.1.0
```

In [21]:
```
jupyter labextension install jupyterlab-plotly
# !pip install --upgrade plotly
```

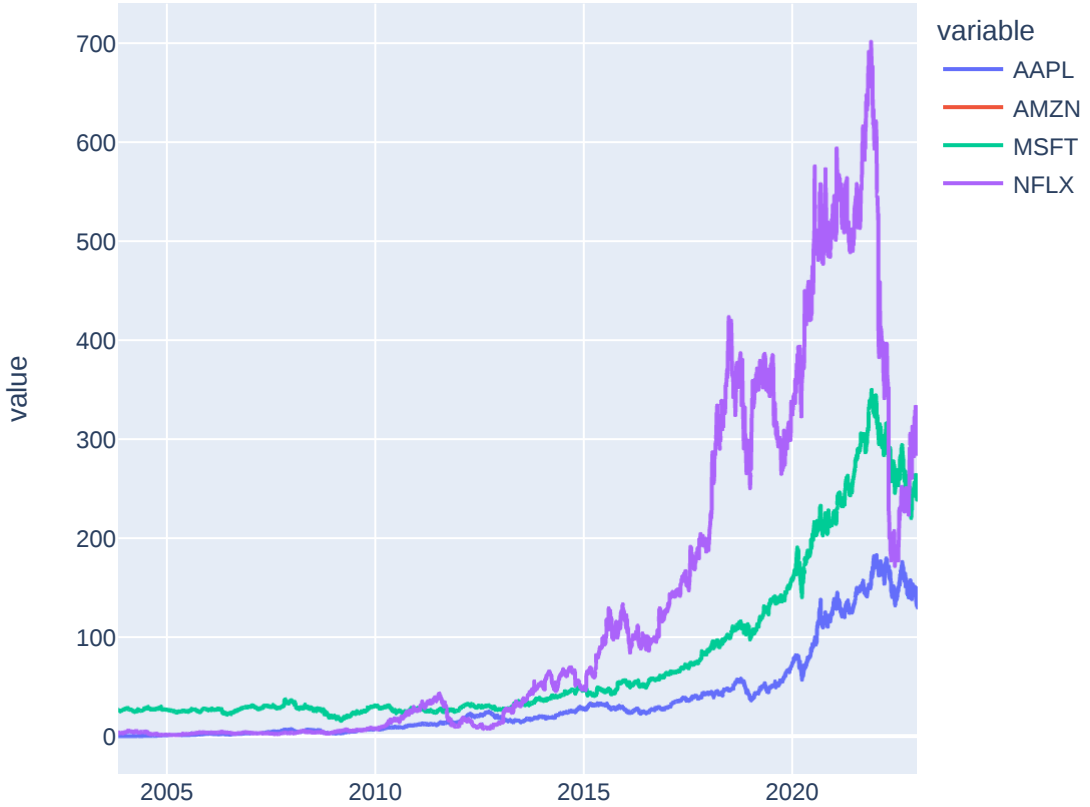
```
Requirement already satisfied: plotly in /Users/takaooba/anaconda3/lib/python3.9/site-packages (5.11.0)
Requirement already satisfied: tenacity>=6.2.0 in /Users/takaooba/anaconda3/lib/python3.9/site-packages (from plotly) (8.1.0)
```

In [22]:
```
import plotly.offline as pyo
```

In [23]:
```
pyo.init_notebook_mode(connected = True)
```

In [24]:
```
pd.options.plotting.backend = 'plotly' # alter from matplotlib to plotly
```

In [25]:
```
# From above, the default would be plotting through plotly
high.plot()
```

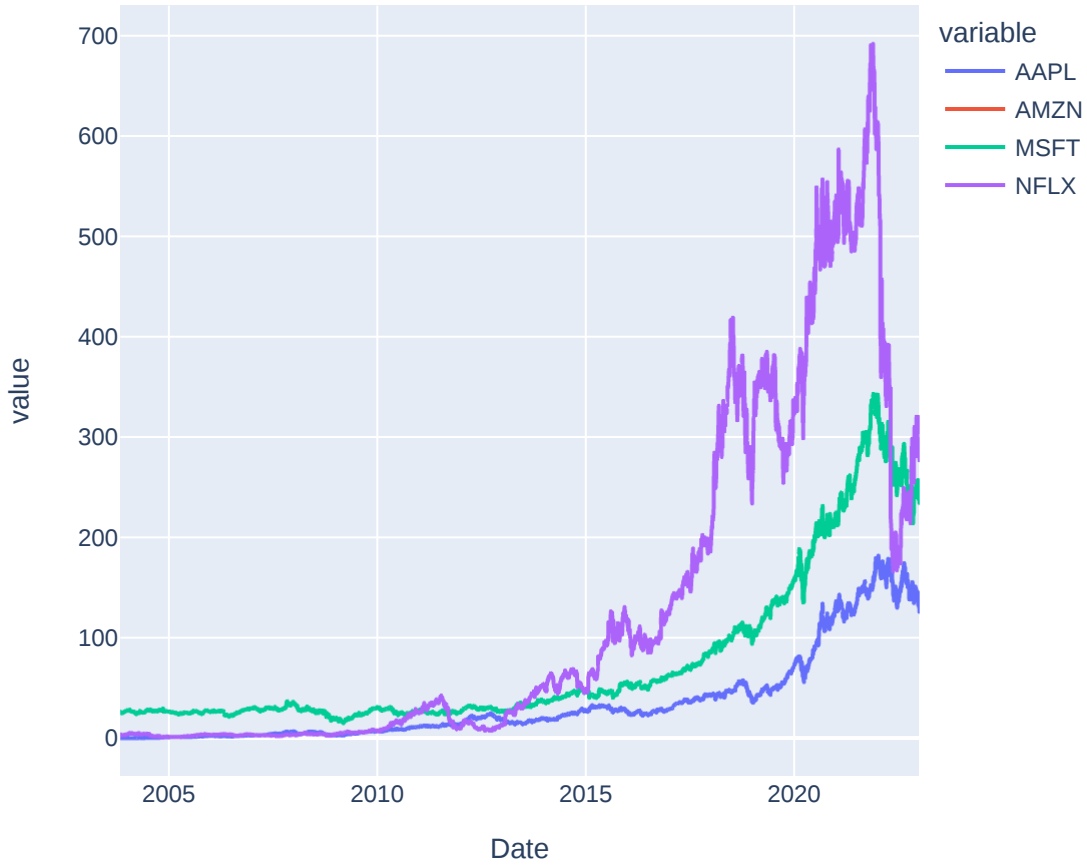


Date

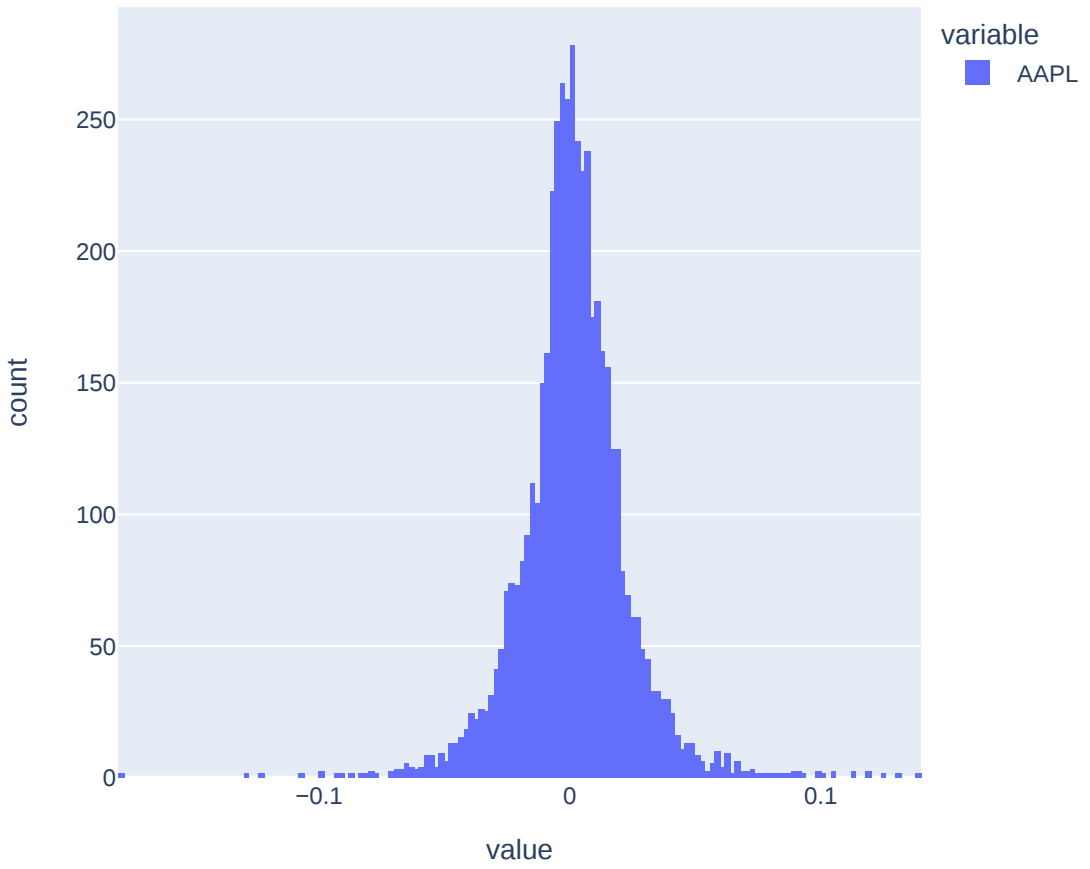
Notice how the dashboard became interactive.

I can select multiple stocks to be on the graph and read the values at a given time through hovering over the line.

In [26]:
```
Close.plot()
```



In [28]:
```
Close['AAPL'].pct_change().plot(kind = 'hist') # plotting the histogram of the percent change
```



In conclusion, I successfully was able to read current and past stock data through pandas datareader. I have looked in to ways to alter the start and end time to retrieve the time-frame of interest. Further, I plotted the price of the stock through matplotlib and plotly. By generating the graphs via plotly, I was able to make it more visually appealing and interactive with the users.