Extracting Stock Data Using a Python Library

A company's stock share is a piece of the company more precisely:

A stock (also known as equity) is a security that represents the ownership of a fraction of a corporation. This entitles the owner of the stock to a proportion of the corporation's assets and profits equal to how much stock they own. Units of stock are called "shares." [1]
An investor can buy a stock and sell it later. If the stock price increases, the investor profits, If it decreases,the investor with incur a loss. Determining the stock price is complex; it depends on the number of outstanding shares, the size of the company's future profits, and much more. People trade stocks throughout the day the stock ticker is a report of the price of a certain stock, updated continuously throughout the trading session by the various stock market exchanges.

You are a data scientist working for a hedge fund; it's your job to determine any suspicious stock activity. In this lab you will extract stock data using a Python library. We will use the yfinance library, it allows us to extract data for stocks returning data in a pandas dataframe. You will use the lab to extract.

Table of Contents

Estimated Time Needed: **30 min**

In [1]:

```
!pip install yfinance
#!pip install pandas
```

Collecting yfinance
  Downloading yfinance-0.1.66-py2.py3-none-any.whl (25 kB)
Requirement already satisfied: pandas>=0.24 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from yfinance) (1.3.3)
Requirement already satisfied: requests>=2.20 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packag
es (from yfinance) (2.26.0)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from yfinance) (4.6.3)
Collecting multitasking>=0.0.7
  Downloading multitasking-0.0.10.tar.gz (8.2 kB)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from yfinance) (1.21.2)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/sit
e-packages (from pandas>=0.24->yfinance) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from pandas>=0.24->yfinance) (2021.1)
Requirement already satisfied: six>=1.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (fr
om python-dateutil>=2.7.3->pandas>=0.24->yfinance) (1.16.0)
Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-pa
ckages (from requests>=2.20->yfinance) (2021.5.30)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site
-packages (from requests>=2.20->yfinance) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from requests>=2.20->yfinance) (3.1)
Requirement already satisfied: charset-normalizer~=2.0.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/
site-packages (from requests>=2.20->yfinance) (2.0.0)
Building wheels for collected packages: multitasking
  Building wheel for multitasking (setup.py) ... done
  Created wheel for multitasking: filename=multitasking-0.0.10-py3-none-any.whl size=8500 sha256=c753538c01cb
cd8eb850953eb053086aa5416bfb0ead7b415dd2c508b28bdd27
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/34/ba/79/c0260c6f1a03f420ec7673eff9981778f293b91079
74679e36
Successfully built multitasking
Installing collected packages: multitasking, yfinance
Successfully installed multitasking-0.0.10 yfinance-0.1.66

```
import yfinance as yf
import pandas as pd
```

Using the yfinance Library to Extract Stock Data

Using the `Ticker` module we can create an object that will allow us to access functions to extract data. To do this we need to provide the ticker symbol for the stock, here the company is Apple and the ticker symbol is `AAPL`.

```
apple = yf.Ticker("AAPL")
```

Now we can access functions and variables to extract the type of data we need. You can view them and what they represent here
https://aroussi.com/post/python-yahoo-finance (https://aroussi.com/post/python-yahoo-finance?
utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-
SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0220ENSkillsNetwork23455606-2021-01-01).

Stock Info

Using the attribute `info` we can extract information about the stock as a Python dictionary.

```
apple_info=apple.info
apple_info
```

```
Out[4]:

{'zip': '95014',
 'sector': 'Technology',
 'fullTimeEmployees': 154000,
 'longBusinessSummary': 'Apple Inc. designs, manufactures, and markets smartphones, personal computers, table
ts, wearables, and accessories worldwide. It also sells various related services. In addition, the company of
fers iPhone, a line of smartphones; Mac, a line of personal computers; iPad, a line of multi-purpose tablets;
AirPods Max, an over-ear wireless headphone; and wearables, home, and accessories comprising AirPods, Apple T
V, Apple Watch, Beats products, HomePod, and iPod touch. Further, it provides AppleCare support services; clo
ud services store services; and operates various platforms, including the App Store that allow customers to d
iscover and download applications and digital content, such as books, music, video, games, and podcasts. Addi
tionally, the company offers various services, such as Apple Arcade, a game subscription service; Apple Musi
c, which offers users a curated listening experience with on-demand radio stations; Apple News+, a subscripti
on news and magazine service; Apple TV+, which offers exclusive original content; Apple Card, a co-branded cr
edit card; and Apple Pay, a cashless payment service, as well as licenses its intellectual property. The comp
any serves consumers, and small and mid-sized businesses; and the education, enterprise, and government marke
ts. It distributes third-party applications for its products through the App Store. The company also sells it
s products through its retail and online stores, and direct sales force; and third-party cellular network car
riers, wholesalers, retailers, and resellers. Apple Inc. was incorporated in 1977 and is headquartered in Cup
ertino, California.',
 'city': 'Cupertino',
 'phone': '408 996 1010',
 'state': 'CA',
 'country': 'United States',
 'companyOfficers': [],
 'website': 'http://www.apple.com',
 'maxAge': 1,
 'address1': 'One Apple Park Way',
 'industry': 'Consumer Electronics',
 'ebitdaMargins': 0.32867,
 'profitMargins': 0.25882,
 'grossMargins': 0.41779,
 'operatingCashflow': 104037998592,
 'revenueGrowth': 0.288,
 'operatingMargins': 0.29782,
 'ebitda': 120233000960,
 'targetLowPrice': 128.01,
 'recommendationKey': 'buy',
 'grossProfits': 152836000000,
 'freeCashflow': 73295003648,
 'targetMedianPrice': 170,
 'currentPrice': 154.0184,
```

'earningsGrowth': 0.662,
'currentRatio': 1.075,
'returnOnAssets': 0.20179,
'numberOfAnalystOpinions': 40,
'targetMeanPrice': 168.45,
'debtToEquity': 216.392,
'returnOnEquity': 1.47443,
'targetHighPrice': 190,
'totalCash': 62639001600,
'totalDebt': 136521998336,
'totalRevenue': 365817004032,
'totalCashPerShare': 3.818,
'financialCurrency': 'USD',
'revenuePerShare': 21.904,
'quickRatio': 0.91,
'recommendationMean': 1.9,
'exchange': 'NMS',
'shortName': 'Apple Inc.',
'longName': 'Apple Inc.',
'exchangeTimezoneName': 'America/New_York',
'exchangeTimezoneShortName': 'EST',
'isEsgPopulated': False,
'gmtOffSetMilliseconds': '-18000000',
'quoteType': 'EQUITY',
'symbol': 'AAPL',
'messageBoardId': 'finmb_24937',
'market': 'us_market',
'annualHoldingsTurnover': None,
'enterpriseToRevenue': 6.974,
'beta3Year': None,
'enterpriseToEbitda': 21.219,
'52WeekChange': 0.27933574,
'morningStarRiskRating': None,
'forwardEps': 6.15,
'revenueQuarterlyGrowth': None,
'sharesOutstanding': 16406400000,
'fundInceptionDate': None,
'annualReportExpenseRatio': None,
'totalAssets': None,
'bookValue': 3.841,
'sharesShort': 100495541,
'sharesPercentSharesOut': 0.0061000003,
'fundFamily': None,

'lastFiscalYearEnd': 1632528000,
'heldPercentInstitutions': 0.58763003,
'netIncomeToCommon': 94679998464,
'trailingEps': 5.61,
'lastDividendValue': 0.22,
'SandP52WeekChange': 0.3175944,
'priceToBook': 40.098515,
'heldPercentInsiders': 0.00071000005,
'nextFiscalYearEnd': 1695600000,
'yield': None,
'mostRecentQuarter': 1632528000,
'shortRatio': 1.34,
'sharesShortPreviousMonthDate': 1632960000,
'floatShares': 16389334347,
'beta': 1.205714,
'enterpriseValue': 2551248846848,
'priceHint': 2,
'threeYearAverageReturn': None,
'lastSplitDate': 1598832000,
'lastSplitFactor': '4:1',
'legalType': None,
'lastDividendDate': 1636070400,
'morningStarOverallRating': None,
'earningsQuarterlyGrowth': 0.622,
'priceToSalesTrailing12Months': 6.907518,
'dateShortInterest': 1635465600,
'pegRatio': 1.7,
'ytdReturn': None,
'forwardPE': 25.043642,
'lastCapGain': None,
'shortPercentOfFloat': 0.0061000003,
'sharesShortPriorMonth': 101107412,
'impliedSharesOutstanding': None,
'category': None,
'fiveYearAverageReturn': None,
'previousClose': 151,
'regularMarketOpen': 150.995,
'twoHundredDayAverage': 141.54402,
'trailingAnnualDividendYield': 0.005629139,
'payoutRatio': 0.1515,
'volume24Hr': None,
'regularMarketDayHigh': 155,
'navPrice': None,

'averageDailyVolume10Day': 57128442,
'regularMarketPreviousClose': 151,
'fiftyDayAverage': 146.77278,
'trailingAnnualDividendRate': 0.85,
'open': 150.995,
'toCurrency': None,
'averageVolume10days': 57128442,
'expireDate': None,
'algorithm': None,
'dividendRate': 0.88,
'exDividendDate': 1636070400,
'circulatingSupply': None,
'startDate': None,
'regularMarketDayLow': 150.995,
'currency': 'USD',
'trailingPE': 27.45426,
'regularMarketVolume': 48774609,
'lastMarket': None,
'maxSupply': None,
'openInterest': None,
'marketCap': 2526887542784,
'volumeAllCurrencies': None,
'strikePrice': None,
'averageVolume': 74787447,
'dayLow': 150.995,
'ask': 154.45,
'askSize': 900,
'volume': 48774609,
'fiftyTwoWeekHigh': 157.26,
'fromCurrency': None,
'fiveYearAvgDividendYield': 1.22,
'fiftyTwoWeekLow': 112.59,
'bid': 154.45,
'tradeable': False,
'dividendYield': 0.0058,
'bidSize': 900,
'dayHigh': 155,
'regularMarketPrice': 154.0184,
'preMarketPrice': 151,
'logo_url': 'https://logo.clearbit.com/apple.com'}

We can get the `'country'` using the key country

```
apple_info['country']
```

Out[5]:

```
'United States'
```

Extracting Share Price

A share is the single smallest part of a company's stock that you can buy, the prices of these shares fluctuate over time. Using the `history()` method we can get the share price of the stock over a certain period of time. Using the `period` parameter we can set how far back from the present to get data. The options for `period` are 1 day (1d), 5d, 1 month (1mo) , 3mo, 6mo, 1 year (1y), 2y, 5y, 10y, ytd, and max.

In [6]:

```
apple_share_price_data = apple.history(period="max")
```

The format that the data is returned in is a Pandas DataFrame. With the `Date` as the index the share `Open` , `High` , `Low` , `Close` , `Volume` , and `Stock Splits` are given for each day.

```
apple_share_price_data.head()
```

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
1980-12-12	0.100453	0.100890	0.100453	0.100453	469033600	0.0	0.0
1980-12-15	0.095649	0.095649	0.095213	0.095213	175884800	0.0	0.0
1980-12-16	0.088661	0.088661	0.088224	0.088224	105728000	0.0	0.0
1980-12-17	0.090408	0.090845	0.090408	0.090408	86441600	0.0	0.0
1980-12-18	0.093029	0.093466	0.093029	0.093029	73449600	0.0	0.0

We can reset the index of the DataFrame with the `reset_index` function. We also set the `inplace` paramter to `True` so the change takes place to the DataFrame itself.

```
apple_share_price_data.reset_index(inplace=True)
apple_share_price_data.head()
```

|	index	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	0	1980-12-12	0.100453	0.100890	0.100453	0.100453	469033600	0.0	0.0
1	1	1980-12-15	0.095649	0.095649	0.095213	0.095213	175884800	0.0	0.0
2	2	1980-12-16	0.088661	0.088661	0.088224	0.088224	105728000	0.0	0.0
3	3	1980-12-17	0.090408	0.090845	0.090408	0.090408	86441600	0.0	0.0
4	4	1980-12-18	0.093029	0.093466	0.093029	0.093029	73449600	0.0	0.0

We can plot the `Open` price against the `Date` :

In [10]:

```python
apple_share_price_data.plot(x="Date", y="Open")
```

Out[10]:

```
<AxesSubplot:xlabel='Date'>
```
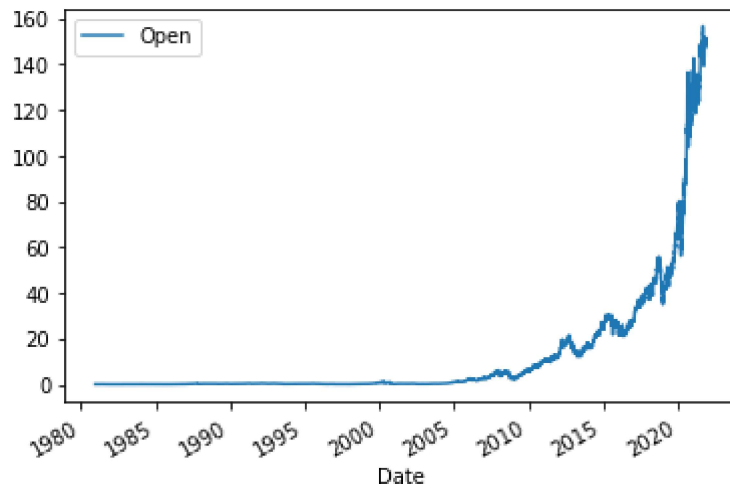

Extracting Dividends

Dividends are the distribution of a companys profits to shareholders. In this case they are defined as an amount of money returned per share an investor owns. Using the variable `dividends` we can get a dataframe of the data. The period of the data is given by the period defined in the 'history` function.

In [11]:

```
apple.dividends
```

Out[11]:

```
Date
1987-05-11    0.000536
1987-08-10    0.000536
1987-11-17    0.000714
1988-02-12    0.000714
1988-05-16    0.000714
                ...
2020-11-06    0.205000
2021-02-05    0.205000
2021-05-07    0.220000
2021-08-06    0.220000
2021-11-05    0.220000
Name: Dividends, Length: 73, dtype: float64
```
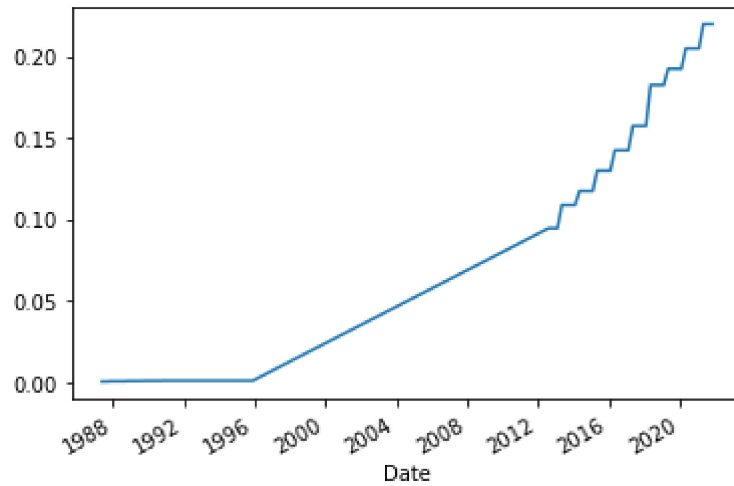
We can plot the dividends overtime:

```
apple.dividends.plot()
```

```
<AxesSubplot:xlabel='Date'>
```


Exercise

Now using the `Ticker` module create an object for AMD (Advanced Micro Devices) with the ticker symbol is `AMD` called; name the object `amd`.

In [13]:

```
amd = yf.Ticker("AMD")
```

Question 1 Use the key `'country'` to find the country the stock belongs to, remember it as it will be a quiz question.

In [15]:

```
amd_info=amd.info
amd_info

amd_info['country']
```

Out[15]:

```
'United States'
```

Question 2 Use the key `'sector'` to find the sector the stock belongs to, remember it as it will be a quiz question.

In [16]:

```
amd_info['sector']
```

Out[16]:

```
'Technology'
```

Question 3 Obtain stock data for AMD using the `history` function, set the `period` to max. Find the `Volume` traded on the first day (first row).

In [19]:

```python
amd_stock_data=amd.history(period="max")

amd_stock_data.reset_index(inplace=True)
amd_stock_data.head()
```

Out[19]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	1980-03-17	0.0	3.302083	3.125000	3.145833	219600	0	0.0
1	1980-03-18	0.0	3.125000	2.937500	3.031250	727200	0	0.0
2	1980-03-19	0.0	3.083333	3.020833	3.041667	295200	0	0.0
3	1980-03-20	0.0	3.062500	3.010417	3.010417	159600	0	0.0
4	1980-03-21	0.0	3.020833	2.906250	2.916667	130800	0	0.0

In [20]:

```python
amd_stock_data.iloc[0][5]
```

Out[20]:

219600

About the Authors:

[Joseph Santarcangelo (https://www.linkedin.com/in/joseph-s-50398b136/?
utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-
SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0220ENSkillsNetwork23455606-2021-01-01)](https://www.linkedin.com/in/joseph-s-50398b136/) has a PhD in Electrical Engineering, his research
focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working
for IBM since he completed his PhD.

Azim Hirjani

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-11-10	1.1	Malika Singla	Deleted the Optional part
2020-08-27	1.0	Malika Singla	Added lab to GitLab