

# Final\_Assignment\_Library

December 26, 2022

## 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 will 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

- Using yfinance to Extract Stock Info
- Using yfinance to Extract Historical Share Price Data
- Using yfinance to Extract Historical Dividends Data
- Exercise

Estimated Time Needed: 30 min

```
[1]: !pip install yfinance==0.1.93
      #!pip install pandas==1.3.3
```

Collecting yfinance==0.1.93

Downloading yfinance-0.1.93-py2.py3-none-any.whl (31 kB)

Requirement already satisfied: cryptography>=3.3.2 in

/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.93) (38.0.2)

Collecting appdirs>=1.4.4

Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)

Collecting multitasking>=0.0.7

Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)

Requirement already satisfied: lxml>=4.9.1 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
yfinance==0.1.93) (4.9.1)

Requirement already satisfied: numpy>=1.16.5 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
yfinance==0.1.93) (1.21.6)

Requirement already satisfied: pandas>=1.3.0 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
yfinance==0.1.93) (1.3.5)

Requirement already satisfied: requests>=2.26 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
yfinance==0.1.93) (2.28.1)

Requirement already satisfied: cffi>=1.12 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
cryptography>=3.3.2->yfinance==0.1.93) (1.15.1)

Requirement already satisfied: python-dateutil>=2.7.3 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
pandas>=1.3.0->yfinance==0.1.93) (2.8.2)

Requirement already satisfied: pytz>=2017.3 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
pandas>=1.3.0->yfinance==0.1.93) (2022.6)

Requirement already satisfied: charset-normalizer<3,>=2 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
requests>=2.26->yfinance==0.1.93) (2.1.1)

Requirement already satisfied: certifi>=2017.4.17 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
requests>=2.26->yfinance==0.1.93) (2022.9.24)

Requirement already satisfied: urllib3<1.27,>=1.21.1 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
requests>=2.26->yfinance==0.1.93) (1.26.13)

Requirement already satisfied: idna<4,>=2.5 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
requests>=2.26->yfinance==0.1.93) (3.4)

Requirement already satisfied: pycparser in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from  
cffi>=1.12->cryptography>=3.3.2->yfinance==0.1.93) (2.21)

Requirement already satisfied: six>=1.5 in  
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from python-  
dateutil>=2.7.3->pandas>=1.3.0->yfinance==0.1.93) (1.16.0)

Installing collected packages: multitasking, appdirs, yfinance  
Successfully installed appdirs-1.4.4 multitasking-0.0.11 yfinance-0.1.93

```
[3]: import yfinance as yf
import pandas as pd
```

## 0.1 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`.

```
[4]: 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>.

### 0.1.1 Stock Info

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

```
[5]: apple_info=apple.info  
apple_info
```

```
[5]: {'zip': '95014',  
      'sector': 'Technology',  
      'fullTimeEmployees': 164000,  
      'longBusinessSummary': 'Apple Inc. designs, manufactures, and markets  
smartphones, personal computers, tablets, wearables, and accessories worldwide.  
It also sells various related services. In addition, the company offers iPhone,  
a line of smartphones; Mac, a line of personal computers; iPad, a line of multi-  
purpose tablets; and wearables, home, and accessories comprising AirPods, Apple  
TV, Apple Watch, Beats products, and HomePod. Further, it provides AppleCare  
support and cloud services store services; and operates various platforms,  
including the App Store that allow customers to discover and download  
applications and digital content, such as books, music, video, games, and  
podcasts. Additionally, the company offers various services, such as Apple  
Arcade, a game subscription service; Apple Fitness+, a personalized fitness  
service; Apple Music, which offers users a curated listening experience with on-  
demand radio stations; Apple News+, a subscription news and magazine service;  
Apple TV+, which offers exclusive original content; Apple Card, a co-branded  
credit card; and Apple Pay, a cashless payment service, as well as licenses its  
intellectual property. The company serves consumers, and small and mid-sized  
businesses; and the education, enterprise, and government markets. It  
distributes third-party applications for its products through the App Store. The  
company also sells its products through its retail and online stores, and direct  
sales force; and third-party cellular network carriers, wholesalers, retailers,  
and resellers. Apple Inc. was incorporated in 1977 and is headquartered in  
Cupertino, California.',  
      'city': 'Cupertino',  
      'phone': '408 996 1010',  
      'state': 'CA',  
      'country': 'United States',  
      'companyOfficers': [],  
      'website': 'https://www.apple.com',
```

'maxAge': 1,  
 'address1': 'One Apple Park Way',  
 'industry': 'Consumer Electronics',  
 'ebitdaMargins': 0.33105,  
 'profitMargins': 0.2531,  
 'grossMargins': 0.43310001,  
 'operatingCashflow': 122151002112,  
 'revenueGrowth': 0.081,  
 'operatingMargins': 0.30289,  
 'ebitda': 130541002752,  
 'targetLowPrice': 122,  
 'recommendationKey': 'buy',  
 'grossProfits': 170782000000,  
 'freeCashflow': 90215251968,  
 'targetMedianPrice': 176,  
 'currentPrice': 131.86,  
 'earningsGrowth': 0.048,  
 'currentRatio': 0.879,  
 'returnOnAssets': 0.21214001,  
 'numberOfAnalystOpinions': 41,  
 'targetMeanPrice': 176.45,  
 'debtToEquity': 261.446,  
 'returnOnEquity': 1.75459,  
 'targetHighPrice': 214,  
 'totalCash': 48304001024,  
 'totalDebt': 132480000000,  
 'totalRevenue': 394328014848,  
 'totalCashPerShare': 3.036,  
 'financialCurrency': 'USD',  
 'revenuePerShare': 24.317,  
 'quickRatio': 0.709,  
 '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': 5.533,  
 'beta3Year': None,  
 'enterpriseToEbitda': 16.714,

'52WeekChange': -0.268785,  
 'morningStarRiskRating': None,  
 'forwardEps': 6.77,  
 'revenueQuarterlyGrowth': None,  
 'sharesOutstanding': 15908100096,  
 'fundInceptionDate': None,  
 'annualReportExpenseRatio': None,  
 'totalAssets': None,  
 'bookValue': 3.178,  
 'sharesShort': 125719918,  
 'sharesPercentSharesOut': 0.007900001,  
 'fundFamily': None,  
 'lastFiscalYearEnd': 1663977600,  
 'heldPercentInstitutions': 0.60025,  
 'netIncomeToCommon': 99802996736,  
 'trailingEps': 6.11,  
 'lastDividendValue': 0.23,  
 'SandP52WeekChange': -0.19752294,  
 'priceToBook': 41.491505,  
 'heldPercentInsiders': 0.00071999995,  
 'nextFiscalYearEnd': 1727136000,  
 'yield': None,  
 'mostRecentQuarter': 1663977600,  
 'shortRatio': 1.52,  
 'sharesShortPreviousMonthDate': 1667174400,  
 'floatShares': 15891096314,  
 'beta': 1.219468,  
 'enterpriseValue': 2181820448768,  
 'priceHint': 2,  
 'threeYearAverageReturn': None,  
 'lastSplitDate': 1598832000,  
 'lastSplitFactor': '4:1',  
 'legalType': None,  
 'lastDividendDate': 1667520000,  
 'morningStarOverallRating': None,  
 'earningsQuarterlyGrowth': 0.008,  
 'priceToSalesTrailing12Months': 5.319536,  
 'dateShortInterest': 1669766400,  
 'pegRatio': 2.44,  
 'ytdReturn': None,  
 'forwardPE': 19.477104,  
 'lastCapGain': None,  
 'shortPercentOfFloat': 0.007900001,  
 'sharesShortPriorMonth': 103178670,  
 'impliedSharesOutstanding': 0,  
 'category': None,  
 'fiveYearAverageReturn': None,

'previousClose': 132.23,  
 'regularMarketOpen': 130.92,  
 'twoHundredDayAverage': 152.1331,  
 'trailingAnnualDividendYield': 0.0068063224,  
 'payoutRatio': 0.14729999,  
 'volume24Hr': None,  
 'regularMarketDayHigh': 132.41,  
 'navPrice': None,  
 'averageDailyVolume10Day': 89022860,  
 'regularMarketPreviousClose': 132.23,  
 'fiftyDayAverage': 144.1318,  
 'trailingAnnualDividendRate': 0.9,  
 'open': 130.92,  
 'toCurrency': None,  
 'averageVolume10days': 89022860,  
 'expireDate': None,  
 'algorithm': None,  
 'dividendRate': 0.92,  
 'exDividendDate': 1667520000,  
 'circulatingSupply': None,  
 'startDate': None,  
 'regularMarketDayLow': 129.64,  
 'currency': 'USD',  
 'trailingPE': 21.581015,  
 'regularMarketVolume': 63814893,  
 'lastMarket': None,  
 'maxSupply': None,  
 'openInterest': None,  
 'marketCap': 2097642078208,  
 'volumeAllCurrencies': None,  
 'strikePrice': None,  
 'averageVolume': 86434942,  
 'dayLow': 129.64,  
 'ask': 131.7,  
 'askSize': 900,  
 'volume': 63814893,  
 'fiftyTwoWeekHigh': 182.94,  
 'fromCurrency': None,  
 'fiveYearAvgDividendYield': 0.99,  
 'fiftyTwoWeekLow': 129.04,  
 'bid': 131.66,  
 'tradeable': False,  
 'dividendYield': 0.0069999998,  
 'bidSize': 1100,  
 'dayHigh': 132.41,  
 'coinMarketCapLink': None,  
 'regularMarketPrice': 131.86,

```
'preMarketPrice': None,
'logo_url': 'https://logo.clearbit.com/apple.com',
'trailingPegRatio': 2.6404}
```

We can get the 'country' using the key country

```
[8]: apple_info['country']
```

```
[8]: 'United States'
```

### 0.1.2 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.

```
[9]: 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.

```
[10]: apple_share_price_data.head()
```

```
[10]:
```

	Open	High	Low	Close	Volume \
Date					
1980-12-12 00:00:00-05:00	0.099874	0.100308	0.099874	0.099874	469033600
1980-12-15 00:00:00-05:00	0.095098	0.095098	0.094663	0.094663	175884800
1980-12-16 00:00:00-05:00	0.088149	0.088149	0.087715	0.087715	105728000
1980-12-17 00:00:00-05:00	0.089886	0.090320	0.089886	0.089886	86441600
1980-12-18 00:00:00-05:00	0.092492	0.092927	0.092492	0.092492	73449600

	Dividends	Stock Splits
Date		
1980-12-12 00:00:00-05:00	0.0	0.0
1980-12-15 00:00:00-05:00	0.0	0.0
1980-12-16 00:00:00-05:00	0.0	0.0
1980-12-17 00:00:00-05:00	0.0	0.0
1980-12-18 00:00:00-05:00	0.0	0.0

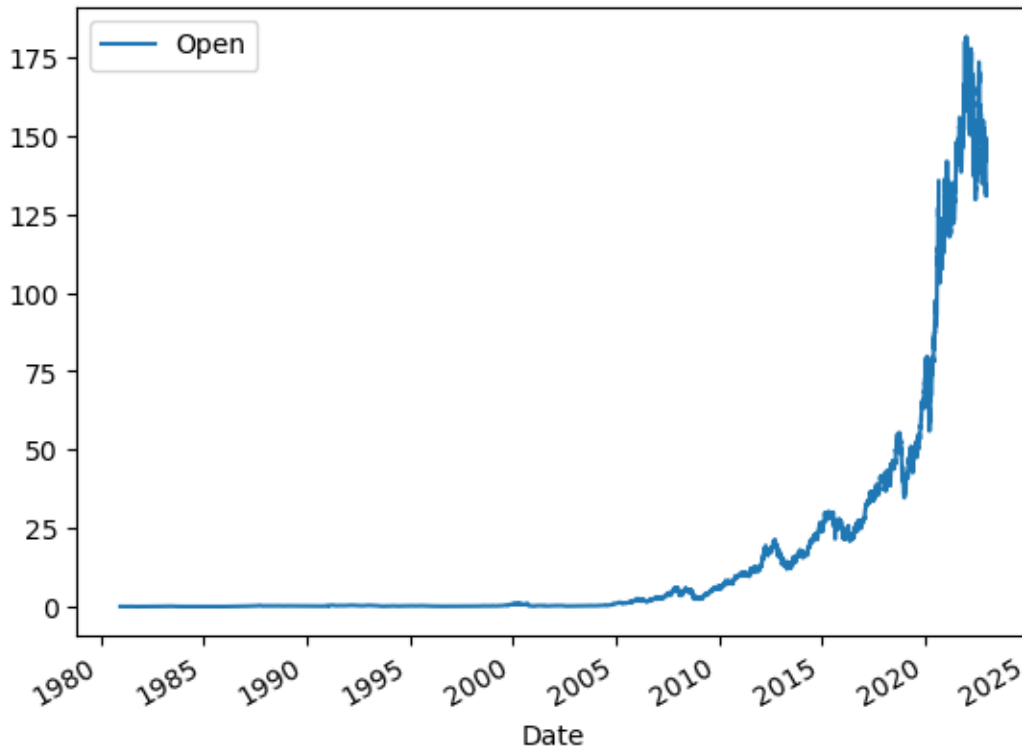
We can reset the index of the DataFrame with the `reset_index` function. We also set the `inplace` paramter to `True` so the change takes place to the DataFrame itself.

```
[11]: apple_share_price_data.reset_index(inplace=True)
```

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

```
[12]: apple_share_price_data.plot(x="Date", y="Open")
```

```
[12]: <AxesSubplot:xlabel='Date'>
```



### 0.1.3 Extracting Dividends

Dividends are the distribution of a company's 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.

```
[13]: apple.dividends
```

```
[13]: Date
1987-05-11 00:00:00-04:00    0.000536
1987-08-10 00:00:00-04:00    0.000536
1987-11-17 00:00:00-05:00    0.000714
1988-02-12 00:00:00-05:00    0.000714
1988-05-16 00:00:00-04:00    0.000714
...
2021-11-05 00:00:00-04:00    0.220000
2022-02-04 00:00:00-05:00    0.220000
2022-05-06 00:00:00-04:00    0.230000
```

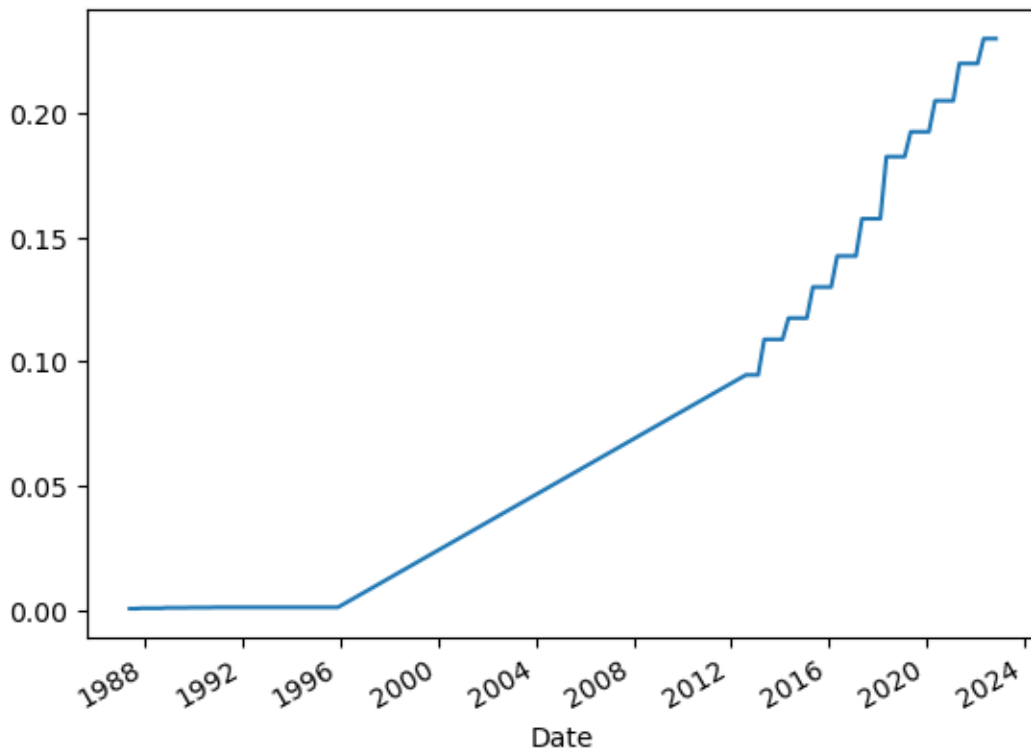


```
2022-08-05 00:00:00-04:00    0.230000
2022-11-04 00:00:00-04:00    0.230000
Name: Dividends, Length: 77, dtype: float64
```

We can plot the dividends overtime:

```
[14]: apple.dividends.plot()
```

```
[14]: <AxesSubplot:xlabel='Date'>
```



## 0.2 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`.

```
[26]: 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.

```
[27]: amd_info=amd.info
      amd_info
```

```
[27]: {'zip': '95054',
      'sector': 'Technology',
      'fullTimeEmployees': 15500,
      'longBusinessSummary': 'Advanced Micro Devices, Inc. operates as a semiconductor company worldwide. The company operates in two segments, Computing and Graphics; and Enterprise, Embedded and Semi-Custom. Its products include x86 microprocessors as an accelerated processing unit, chipsets, discrete and integrated graphics processing units (GPUs), data center and professional GPUs, and development services; and server and embedded processors, and semi-custom System-on-Chip (SoC) products, development services, and technology for game consoles. The company provides processors for desktop and notebook personal computers under the AMD Ryzen, AMD Ryzen PRO, Ryzen Threadripper, Ryzen Threadripper PRO, AMD Athlon, AMD Athlon PRO, AMD FX, AMD A-Series, and AMD PRO A-Series processors brands; discrete GPUs for desktop and notebook PCs under the AMD Radeon graphics, AMD Embedded Radeon graphics brands; and professional graphics products under the AMD Radeon Pro and AMD FirePro graphics brands. It also offers Radeon Instinct, Radeon PRO V-series, and AMD Instinct accelerators for servers; chipsets under the AMD trademark; microprocessors for servers under the AMD EPYC; embedded processor solutions under the AMD Athlon, AMD Geode, AMD Ryzen, AMD EPYC, AMD R-Series, and G-Series processors brands; and customer-specific solutions based on AMD CPU, GPU, and multi-media technologies, as well as semi-custom SoC products. It serves original equipment manufacturers, public cloud service providers, original design manufacturers, system integrators, independent distributors, online retailers, and add-in-board manufacturers through its direct sales force, independent distributors, and sales representatives. The company was incorporated in 1969 and is headquartered in Santa Clara, California.',
      'city': 'Santa Clara',
      'phone': '408 749 4000',
      'state': 'CA',
      'country': 'United States',
      'companyOfficers': [],
      'website': 'https://www.amd.com',
      'maxAge': 1,
      'address1': '2485 Augustine Drive',
      'industry': 'Semiconductors',
      'ebitdaMargins': 0.24299,
      'profitMargins': 0.09957,
      'grossMargins': 0.5095,
      'operatingCashflow': 3820000000,
      'revenueGrowth': 0.29,
      'operatingMargins': 0.13085,
      'ebitda': 5546999808,
      'targetLowPrice': 60,
      'recommendationKey': 'buy',
      'grossProfits': 7929000000,
      'freeCashflow': 4550750208,
```

'targetMedianPrice': 87.5,  
 'currentPrice': 64.52,  
 'earningsGrowth': -0.947,  
 'currentRatio': 2.155,  
 'returnOnAssets': 0.047280002,  
 'numberOfAnalystOpinions': 38,  
 'targetMeanPrice': 90.87,  
 'debtToEquity': 5.299,  
 'returnOnEquity': 0.073699996,  
 'targetHighPrice': 200,  
 'totalCash': 5591000064,  
 'totalDebt': 2889999872,  
 'totalRevenue': 22827999232,  
 'totalCashPerShare': 3.468,  
 'financialCurrency': 'USD',  
 'revenuePerShare': 15.646,  
 'quickRatio': 1.484,  
 'recommendationMean': 2,  
 'exchange': 'NMS',  
 'shortName': 'Advanced Micro Devices, Inc.',  
 'longName': 'Advanced Micro Devices, Inc.',  
 'exchangeTimezoneName': 'America/New\_York',  
 'exchangeTimezoneShortName': 'EST',  
 'isEsgPopulated': False,  
 'gmtOffsetMilliseconds': '-18000000',  
 'quoteType': 'EQUITY',  
 'symbol': 'AMD',  
 'messageBoardId': 'finmb\_168864',  
 'market': 'us\_market',  
 'annualHoldingsTurnover': None,  
 'enterpriseToRevenue': 4.439,  
 'beta3Year': None,  
 'enterpriseToEbitda': 18.267,  
 '52WeekChange': -0.5820161,  
 'morningStarRiskRating': None,  
 'forwardEps': 3.65,  
 'revenueQuarterlyGrowth': None,  
 'sharesOutstanding': 1612359936,  
 'fundInceptionDate': None,  
 'annualReportExpenseRatio': None,  
 'totalAssets': None,  
 'bookValue': 33.835,  
 'sharesShort': 30653701,  
 'sharesPercentSharesOut': 0.019,  
 'fundFamily': None,  
 'lastFiscalYearEnd': 1640390400,  
 'heldPercentInstitutions': 0.70497,

'netIncomeToCommon': 2272999936,  
'trailingEps': 1.65,  
'lastDividendValue': None,  
'SandP52WeekChange': -0.19752294,  
'priceToBook': 1.9069011,  
'heldPercentInsiders': 0.0034999999,  
'nextFiscalYearEnd': 1703462400,  
'yield': None,  
'mostRecentQuarter': 1663977600,  
'shortRatio': 0.38,  
'sharesShortPreviousMonthDate': 1667174400,  
'floatShares': 1602972249,  
'beta': 2.002155,  
'enterpriseValue': 101328216064,  
'priceHint': 2,  
'threeYearAverageReturn': None,  
'lastSplitDate': 966902400,  
'lastSplitFactor': '2:1',  
'legalType': None,  
'lastDividendDate': None,  
'morningStarOverallRating': None,  
'earningsQuarterlyGrowth': -0.928,  
'priceToSalesTrailing12Months': 4.557099,  
'dateShortInterest': 1669766400,  
'pegRatio': 1.24,  
'ytdReturn': None,  
'forwardPE': 17.67671,  
'lastCapGain': None,  
'shortPercentOfFloat': 0.0191,  
'sharesShortPriorMonth': 32943648,  
'impliedSharesOutstanding': 0,  
'category': None,  
'fiveYearAverageReturn': None,  
'previousClose': 63.86,  
'regularMarketOpen': 63.11,  
'twoHundredDayAverage': 84.34555,  
'trailingAnnualDividendYield': 0,  
'payoutRatio': 0,  
'volume24Hr': None,  
'regularMarketDayHigh': 64.66,  
'navPrice': None,  
'averageDailyVolume10Day': 58433270,  
'regularMarketPreviousClose': 63.86,  
'fiftyDayAverage': 66.8492,  
'trailingAnnualDividendRate': 0,  
'open': 63.11,  
'toCurrency': None,

```

'averageVolume10days': 58433270,
'expireDate': None,
'algorithm': None,
'dividendRate': None,
'exDividendDate': 798940800,
'circulatingSupply': None,
'startDate': None,
'regularMarketDayLow': 62.72,
'currency': 'USD',
'trailingPE': 39.103027,
'regularMarketVolume': 45923000,
'lastMarket': None,
'maxSupply': None,
'openInterest': None,
'marketCap': 104029454336,
'volumeAllCurrencies': None,
'strikePrice': None,
'averageVolume': 79203590,
'dayLow': 62.72,
'ask': 64.5,
'askSize': 1400,
'volume': 45923000,
'fiftyTwoWeekHigh': 156.73,
'fromCurrency': None,
'fiveYearAvgDividendYield': None,
'fiftyTwoWeekLow': 54.57,
'bid': 64.49,
'tradeable': False,
'dividendYield': None,
'bidSize': 800,
'dayHigh': 64.66,
'coinMarketCapLink': None,
'regularMarketPrice': 64.52,
'preMarketPrice': None,
'logo_url': 'https://logo.clearbit.com/amd.com',
'trailingPegRatio': 0.8442}

```

```
[28]: amd_info['country']
```

```
[28]: '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.

```
[30]: amd_info['sector']
```

```
[30]: '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).

```
[46]: amd_histstock_data = amd.history(period="max")
      amd_histstock_data
```

```
[46]:
```

		Open	High	Low	Close \
Date					
1980-03-17 00:00:00-05:00		0.000000	3.302083	3.125000	3.145833
1980-03-18 00:00:00-05:00		0.000000	3.125000	2.937500	3.031250
1980-03-19 00:00:00-05:00		0.000000	3.083333	3.020833	3.041667
1980-03-20 00:00:00-05:00		0.000000	3.062500	3.010417	3.010417
1980-03-21 00:00:00-05:00		0.000000	3.020833	2.906250	2.916667
...		...	...	...	...
2022-12-19 00:00:00-05:00		65.120003	65.589996	63.709999	64.589996
2022-12-20 00:00:00-05:00		63.759998	66.059998	63.520000	65.050003
2022-12-21 00:00:00-05:00		65.430000	67.849998	65.209999	67.680000
2022-12-22 00:00:00-05:00		66.080002	66.309998	62.259998	63.860001
2022-12-23 00:00:00-05:00		63.110001	64.660004	62.720001	64.519997

		Volume	Dividends	Stock Splits
Date				
1980-03-17 00:00:00-05:00		219600	0	0.0
1980-03-18 00:00:00-05:00		727200	0	0.0
1980-03-19 00:00:00-05:00		295200	0	0.0
1980-03-20 00:00:00-05:00		159600	0	0.0
1980-03-21 00:00:00-05:00		130800	0	0.0
...		...	...	...
2022-12-19 00:00:00-05:00		46246900	0	0.0
2022-12-20 00:00:00-05:00		52431300	0	0.0
2022-12-21 00:00:00-05:00		44167600	0	0.0
2022-12-22 00:00:00-05:00		73917700	0	0.0
2022-12-23 00:00:00-05:00		45898900	0	0.0

[10787 rows x 7 columns]

```
[47]: amd_histstock_data.head(1)
```

```
[47]:
```

		Open	High	Low	Close	Volume	Dividends \
Date							
1980-03-17 00:00:00-05:00		0.0	3.302083	3.125	3.145833	219600	0

		Stock Splits
Date		
1980-03-17 00:00:00-05:00		0.0

About the Authors:

Joseph Santarcangelo 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

### 0.3 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

##

© IBM Corporation 2020. All rights reserved.