



2024

# Stock Investment Dashboard: Analyzing Financial Data for Investment Decisions

A PROJECT FOR DATA SCIENCE [ANESU](#)

[DAVID PASIPANODYA](#)

## 1. Project Outline

Stock Investment Dashboard: Analysing Financial Data for Investment Decisions

In this project, I took on the role of a Data Scientist/Analyst for a startup investment firm, aiming to assist clients in making informed investment decisions. The primary objective was to create a dashboard that extracts, analyses, and visualises key financial data for Applw. The dashboard would help identify patterns and trends that could influence investment strategies. My functions were downloaded from <https://aroussi.com/post/python-yahoo-finance>.

To accomplish this, I used Python for data extraction and data visualization, incorporating web scraping and financial data libraries. I also leveraged tools such as Skills Network Labs and IBM Watson Studio, which provided a cloud-based environment to handle data and develop the project. The key focus was on displaying financial performance indicators clearly and interactively, using Plotly for rich visualization. The IDE I chose for use was the PyCharm community edition. <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>

Project outline created by:

[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

What is a DataFrame?

A dataframe is a data structure constructed with rows and columns, similar to a database or Excel spreadsheet. It consists of a dictionary of lists in which the list each have their own identifiers or keys, such as "last name" or "food group. <https://miamioh.edu/centers-institutes/center-foranalytics-data-science/students/coding-tutorials/python/pandas-dataframes.html#:~:text=A%20dataframe%20is%20a%20data,%E2%80%9D%20or%20%E2%80%9Cfood%20group.%E2%80%9D>

## Stock Info

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

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

Then I accessed functions and variables to extract the type of data we needed where they can be viewed here: <https://aroussi.com/post/python-yahoo-finance>.

Using the attribute info I extracted information about the stock as a Python dictionary.

```
import json
with open('apple.json') as json_file:
    apple_info = json.load(json_file)
    # Print the type of data variable
    #print("Type:", type(apple_info))
print(apple_info)
```

**Result:**

```
"C:\Data Analysis\Python project for data science\pythonProject1\venv\Scripts\python.exe"
"C:\Data Analysis\Python project for data science\Extracting Stock Data Using Python Library.py"
{'zip': '95014', 'sector': 'Technology', 'fullTimeEmployees': 100000, '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; AirPods Max, an over-ear wireless headphone; and wearables, home, and accessories
comprising AirPods, Apple TV, Apple Watch, Beats products, HomePod, and iPod touch. Further, it
provides AppleCare support services; 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 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': 0.33890998, 'profitMargins': 0.26579002, 'grossMargins':
0.43019, 'operatingCashflow': 112241000448, 'revenueGrowth': 0.112, 'operatingMargins': 0.309,
'ebitda': 128217997312, 'targetLowPrice': 160, 'recommendationKey': 'buy', 'grossProfits':
152836000000, 'freeCashflow': 80153247744, 'targetMedianPrice': 199.5, 'currentPrice': 177.77,
'earningsGrowth': 0.25, 'currentRatio': 1.038, 'returnOnAssets': 0.19875,
'numberOfAnalystOpinions': 44, 'targetMeanPrice': 193.53, 'debtToEquity': 170.714,
'returnOnEquity': 1.45567, 'targetHighPrice': 215, 'totalCash': 63913000960, 'totalDebt':
122797998080, 'totalRevenue': 378323009536, 'totalCashPerShare': 3.916, 'financialCurrency':
'USD', 'revenuePerShare': 22.838, 'quickRatio': 0.875, 'recommendationMean': 1.8, 'exchange':
'NMS', 'shortName': 'Apple Inc.', 'longName': 'Apple Inc.', 'exchangeTimezoneName':
'America/New_York', 'exchangeTimezoneShortName': 'EDT', 'isEsgPopulated': False,
'gmtOffSetMilliseconds': -14400000, 'quoteType': 'EQUITY', 'symbol': 'AAPL', 'messageBoardId':
'finmb_24937', 'market': 'us_market', 'annualHoldingsTurnover': None, 'enterpriseToRevenue':
7.824, 'beta3Year': None, 'enterpriseToEbitda': 23.086, '52WeekChange': 0.4549594,
'morningStarRiskRating': None, 'forwardEps': 6.56, 'revenueQuarterlyGrowth': None,
'sharesOutstanding': 16319399936, 'fundInceptionDate': None, 'annualReportExpenseRatio':
None, 'totalAssets': None, 'bookValue': 4.402, 'sharesShort': 111286790,
'sharesPercentSharesOut': 0.0068, 'fundFamily': None, 'lastFiscalYearEnd': 1632528000,
'heldPercentInstitutions': 0.59397, 'netIncomeToCommon': 100554997760, 'trailingEps': 6.015,
'lastDividendValue': 0.22, 'SandP52WeekChange': 0.15217662, 'priceToBook': 40.38392,
'heldPercentInsiders': 0.0007, 'nextFiscalYearEnd': 1695600000, 'yield': None,
'mostRecentQuarter': 1640390400, 'shortRatio': 1.21, 'sharesShortPreviousMonthDate':
1644883200, 'floatShares': 16302795170, 'beta': 1.185531, 'enterpriseValue': 2959991898112,
'priceHint': 2, 'threeYearAverageReturn': None, 'lastSplitDate': 1598832000, 'lastSplitFactor': '4:1',
'legalType': None, 'lastDividendDate': 1643932800, 'morningStarOverallRating': None,
'earningsQuarterlyGrowth': 0.204, 'priceToSalesTrailing12Months': 7.668314, 'dateShortInterest':
1647302400, 'pegRatio': 1.94, 'ytdReturn': None, 'forwardPE': 27.099087, 'lastCapGain': None,
'shortPercentOfFloat': 0.0068, 'sharesShortPriorMonth': 108944701, 'impliedSharesOutstanding':
0, 'category': None, 'fiveYearAverageReturn': None, 'previousClose': 178.96, 'regularMarketOpen':
```

```
178.55, 'twoHundredDayAverage': 156.03505, 'trailingAnnualDividendYield': 0.004833482,
'payoutRatio': 0.1434, 'volume24Hr': None, 'regularMarketDayHigh': 179.61, 'navPrice': None,
'averageDailyVolume10Day': 93823630, 'regularMarketPreviousClose': 178.96, 'fiftyDayAverage':
166.498, 'trailingAnnualDividendRate': 0.865, 'open': 178.55, 'toCurrency': None,
'averageVolume10days': 93823630, 'expireDate': None, 'algorithm': None, 'dividendRate': 0.88,
'exDividendDate': 1643932800, 'circulatingSupply': None, 'startDate': None,
'regularMarketDayLow': 176.7, 'currency': 'USD', 'trailingPE': 29.55445, 'regularMarketVolume':
92633154, 'lastMarket': None, 'maxSupply': None, 'openInterest': None, 'marketCap':
2901099675648, 'volumeAllCurrencies': None, 'strikePrice': None, 'averageVolume': 95342043,
'dayLow': 176.7, 'ask': 178.53, 'askSize': 800, 'volume': 92633154, 'fiftyTwoWeekHigh': 182.94,
'fromCurrency': None, 'fiveYearAvgDividendYield': 1.13, 'fiftyTwoWeekLow': 122.25, 'bid': 178.4,
'tradeable': False, 'dividendYield': 0.005, 'bidSize': 3200, 'dayHigh': 179.61, 'regularMarketPrice':
177.77, 'preMarketPrice': 178.38, 'logo_url': 'https://logo.clearbit.com/apple.com'}
```

- I got the country which the info was from using

```
aic = apple_info['country']
```

**Result**

**United States**

## Extracting Share Price

- 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.
- **Code**

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

```

                                Open      High  ...  Dividends  Stock Splits
Date
1980-12-12 00:00:00-05:00    0.098943    0.099373  ...         0.0           0.0
1980-12-15 00:00:00-05:00    0.094211    0.094211  ...         0.0           0.0
1980-12-16 00:00:00-05:00    0.087328    0.087328  ...         0.0           0.0
1980-12-17 00:00:00-05:00    0.089049    0.089479  ...         0.0           0.0
1980-12-18 00:00:00-05:00    0.091630    0.092061  ...         0.0           0.0
...
2024-10-28 00:00:00-04:00   233.320007   234.729996  ...         0.0           0.0
2024-10-29 00:00:00-04:00   233.100006   234.330002  ...         0.0           0.0
2024-10-30 00:00:00-04:00   232.610001   233.470001  ...         0.0           0.0
2024-10-31 00:00:00-04:00   229.339996   229.830002  ...         0.0           0.0
2024-11-01 00:00:00-04:00   220.970001   225.350006  ...         0.0           0.0

[11065 rows x 7 columns]

```

Here we see the dates go back to the 12<sup>th</sup> of December 1980, at 5am, opening at a share price of \$.098943, and stretches to the most recent date of the 1<sup>st</sup> of November at 220.97001 at the production of the code.

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.

#### Code

```
print(apple_share_price_data.head())
```

#### Output

```

                                Open      High  ...  Dividends  Stock Splits
Date
1980-12-12 00:00:00-05:00    0.098943    0.099373  ...         0.0           0.0
1980-12-15 00:00:00-05:00    0.094211    0.094211  ...         0.0           0.0
1980-12-16 00:00:00-05:00    0.087328    0.087328  ...         0.0           0.0
1980-12-17 00:00:00-05:00    0.089049    0.089479  ...         0.0           0.0
1980-12-18 00:00:00-05:00    0.091630    0.092061  ...         0.0           0.0

[5 rows x 7 columns]

```

- I also reset the index of the DataFrame with the `reset_index` function. In addition I set the `inplace` paramter to `True` so the change takes place to the DataFrame itself.

#### Code

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

## Output

None

- We can plot the Open price against the Date:

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

## Output

Axes(0.125,0.2;0.775x0.68)

## 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 I got a dataframe of the data. The period of the data is given by the period defined in the 'history' function.

### Code

```
print(apple.dividends)
```

### Output

```
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
...
2023-08-11 00:00:00-04:00    0.240000
2023-11-10 00:00:00-05:00    0.240000
2024-02-09 00:00:00-05:00    0.240000
2024-05-10 00:00:00-04:00    0.250000
2024-08-12 00:00:00-04:00    0.250000
Name: Dividends, Length: 84, dtype: float64
```

We can plot the dividends overtime:

### Code

```
print(apple.dividends) print(apple.dividends)
```

### Coordinates

Axes(0.125,0.2;0.775x0.68)

## Conclusion

Although this was merely a web scraping project, it is very interesting to note that the stock price apple has increased by 2 ½ thousand percent

United States		
	Open	High
Date		
1980-12-12 00:00:00-05:00	0.098834	0.099264
1980-12-15 00:00:00-05:00	0.094108	0.094108
1980-12-16 00:00:00-05:00	0.087232	0.087232
1980-12-17 00:00:00-05:00	0.088951	0.089381
1980-12-18 00:00:00-05:00	0.091530	0.091959
...	...	...
2024-11-04 00:00:00-05:00	220.747138	222.545148
2024-11-05 00:00:00-05:00	221.556241	223.703872
2024-11-06 00:00:00-05:00	222.365355	225.821559
2024-11-07 00:00:00-05:00	224.383137	227.629565
2024-11-08 00:00:00-05:00	227.169998	228.660004