

Lecture 4 yfinance and pandas-datareader

Douglas Chung

National Chengchi University

8 October 2021

1 Downloading financial data

1.1 yfinance

This package downloads global stock market data from Yahoo! Finance.

```
[1]: # import packages
import yfinance as yf
import pandas as pd

msft = yf.Ticker("MSFT")
msft.info
```

```
[1]: {'zip': '98052-6399',
      'sector': 'Technology',
      'fullTimeEmployees': 181000,
      'longBusinessSummary': 'Microsoft Corporation develops, licenses, and supports software, services, devices, and solutions worldwide. Its Productivity and Business Processes segment offers Office, Exchange, SharePoint, Microsoft Teams, Office 365 Security and Compliance, and Skype for Business, as well as related Client Access Licenses (CAL); Skype, Outlook.com, OneDrive, and LinkedIn; and Dynamics 365, a set of cloud-based and on-premises business solutions for organizations and enterprise divisions. Its Intelligent Cloud segment licenses SQL, Windows Servers, Visual Studio, System Center, and related CALs; GitHub that provides a collaboration platform and code hosting service for developers; and Azure, a cloud platform. It also offers support services and Microsoft consulting services to assist customers in developing, deploying, and managing Microsoft server and desktop solutions; and training and certification on Microsoft products. Its More Personal Computing segment provides Windows original equipment manufacturer (OEM) licensing and other non-volume licensing of the Windows operating system; Windows Commercial, such as volume licensing of the Windows operating system, Windows cloud services, and other Windows commercial offerings; patent licensing; Windows Internet of Things; and MSN advertising. It also offers Surface, PC accessories, PCs, tablets, gaming and entertainment consoles, and other devices; Gaming, including Xbox hardware, and Xbox content and services; video games and third-party video game royalties; and Search, including Bing and Microsoft advertising. It sells its products through
```

OEMs, distributors, and resellers; and directly through digital marketplaces, online stores, and retail stores. It has collaborations with Dynatrace, Inc., Morgan Stanley, Micro Focus, WPP plc, ACI Worldwide, Inc., and iCIMS, Inc., as well as a strategic relationship with Avaya Holdings Corp. Microsoft Corporation was founded in 1975 and is headquartered in Redmond, Washington.',

```
'city': 'Redmond',
'phone': '425 882 8080',
'state': 'WA',
'country': 'United States',
'companyOfficers': [],
'website': 'http://www.microsoft.com',
'maxAge': 1,
'address1': 'One Microsoft Way',
'industry': 'Software-Infrastructure',
'ebitdaMargins': 0.48080003,
'profitMargins': 0.36451998,
'grossMargins': 0.68926,
'operatingCashflow': 76740001792,
'revenueGrowth': 0.213,
'operatingMargins': 0.41595,
'ebitda': 80815996928,
'targetLowPrice': 275,
'recommendationKey': 'buy',
'grossProfits': 115856000000,
'freeCashflow': 41337249792,
'targetMedianPrice': 330,
'currentPrice': 296.99,
'earningsGrowth': 0.486,
'currentRatio': 2.08,
'returnOnAssets': 0.13761,
'numberOfAnalystOpinions': 34,
'targetMeanPrice': 330.68,
'debtToEquity': 57.947,
'returnOnEquity': 0.47079,
'targetHighPrice': 411,
'totalCash': 130256003072,
'totalDebt': 82277998592,
'totalRevenue': 168087994368,
'totalCashPerShare': 17.333,
'financialCurrency': 'USD',
'revenuePerShare': 22.272,
'quickRatio': 1.898,
'recommendationMean': 1.6,
'exchange': 'NMS',
'shortName': 'Microsoft Corporation',
'longName': 'Microsoft Corporation',
'exchangeTimezoneName': 'America/New_York',
```

```
'exchangeTimezoneShortName': 'EDT',
'isEsgPopulated': False,
'gmtOffSetMilliseconds': '-14400000',
'quoteType': 'EQUITY',
'symbol': 'MSFT',
'messageBoardId': 'finmb_21835',
'market': 'us_market',
'annualHoldingsTurnover': None,
'enterpriseToRevenue': 12.935,
'beta3Year': None,
'enterpriseToEbitda': 26.904,
'52WeekChange': 0.41637135,
'morningStarRiskRating': None,
'forwardEps': 10.09,
'revenueQuarterlyGrowth': None,
'sharesOutstanding': 7514890240,
'fundInceptionDate': None,
'annualReportExpenseRatio': None,
'totalAssets': None,
'bookValue': 18.884,
'sharesShort': 45696881,
'sharesPercentSharesOut': 0.0061000003,
'fundFamily': None,
'lastFiscalYearEnd': 1625011200,
'heldPercentInstitutions': 0.71723,
'netIncomeToCommon': 61270999040,
'trailingEps': 8.05,
'lastDividendValue': None,
'SandP52WeekChange': 0.31088436,
'priceToBook': 15.727069,
'heldPercentInsiders': 0.00078,
'nextFiscalYearEnd': 1688083200,
'yield': None,
'mostRecentQuarter': 1625011200,
'shortRatio': 2.28,
'sharesShortPreviousMonthDate': 1627603200,
'floatShares': 7506925463,
'beta': 0.77893,
'enterpriseValue': 2174250516480,
'priceHint': 2,
'threeYearAverageReturn': None,
'lastSplitDate': None,
'lastSplitFactor': None,
'legalType': None,
'lastDividendDate': None,
'morningStarOverallRating': None,
'earningsQuarterlyGrowth': 0.469,
```

```
'priceToSalesTrailing12Months': 13.277849,  
'dateShortInterest': 1630368000,  
'pegRatio': 2.24,  
'ytdReturn': None,  
'forwardPE': 29.434092,  
'lastCapGain': None,  
'shortPercentOfFloat': 0.0061000003,  
'sharesShortPriorMonth': 47983963,  
'impliedSharesOutstanding': None,  
'category': None,  
'fiveYearAverageReturn': None,  
'previousClose': 295.71,  
'regularMarketOpen': 297.55,  
'twoHundredDayAverage': 263.5351,  
'trailingAnnualDividendYield': 0.0075749895,  
'payoutRatio': 0.272,  
'volume24Hr': None,  
'regularMarketDayHigh': 298.53,  
'navPrice': None,  
'averageDailyVolume10Day': 17943400,  
'regularMarketPreviousClose': 295.71,  
'fiftyDayAverage': 294.28235,  
'trailingAnnualDividendRate': 2.24,  
'open': 297.55,  
'toCurrency': None,  
'averageVolume10days': 17943400,  
'expireDate': None,  
'algorithm': None,  
'dividendRate': 2.24,  
'exDividendDate': 1629244800,  
'circulatingSupply': None,  
'startDate': None,  
'regularMarketDayLow': 294.08,  
'currency': 'USD',  
'trailingPE': 36.893166,  
'regularMarketVolume': 23652949,  
'lastMarket': None,  
'maxSupply': None,  
'openInterest': None,  
'marketCap': 2231847092224,  
'volumeAllCurrencies': None,  
'strikePrice': None,  
'averageVolume': 22127698,  
'dayLow': 294.08,  
'ask': 0,  
'askSize': 800,  
'volume': 23652949,
```

```
'fiftyTwoWeekHigh': 305.84,
'fromCurrency': None,
'fiveYearAvgDividendYield': 1.51,
'fiftyTwoWeekLow': 196.25,
'bid': 0,
'tradeable': False,
'dividendYield': 0.0076,
'bidSize': 800,
'dayHigh': 298.53,
'regularMarketPrice': 296.99,
'logo_url': 'https://logo.clearbit.com/microsoft.com'}
```

1.1.1 Download historical data for multiple stocks

```
[2]: # select tickers
tickers_list = ["AAPL", "GOOG"]

start_date = "2021-09-01"
end_date = "2021-10-01"

# download historical data from Yahoo! Finance
data = yf.download(tickers=tickers_list,
                    interval="1d", # 1d, 5d, 1mo, 3mo, 6mo, 1y, 2y, 5y, 10y, ytd, max #
                    start=start_date,
                    end=end_date)

data.head()
```

[*****100%*****] 2 of 2 completed

```
[2]:
```

	Adj Close		Close		High \
	AAPL	GOOG	AAPL	GOOG	AAPL
Date					
2021-08-31	151.830002	2909.239990	151.830002	2909.239990	152.800003
2021-09-01	152.509995	2916.840088	152.509995	2916.840088	154.979996
2021-09-02	153.649994	2884.379883	153.649994	2884.379883	154.720001
2021-09-03	154.300003	2895.500000	154.300003	2895.500000	154.630005
2021-09-07	156.690002	2910.379883	156.690002	2910.379883	157.259995

		Low		Open	
	GOOG	AAPL	GOOG	AAPL	GOOG
Date					
2021-08-31	2922.239990	151.289993	2900.000000	152.660004	2917.689941
2021-09-01	2936.409912	152.339996	2912.290039	152.830002	2913.000000
2021-09-02	2926.500000	152.399994	2882.129883	153.869995	2918.989990
2021-09-03	2907.540039	153.089996	2870.100098	153.759995	2882.919922
2021-09-07	2916.479980	154.389999	2890.820068	154.970001	2894.989990

	Volume	
	AAPL	GOOG
Date		
2021-08-31	86453100	1337800
2021-09-01	80313700	791200
2021-09-02	71115500	1092200
2021-09-03	57808700	955200
2021-09-07	82278300	758500

1.1.2 Subsetting data

To compute stock returns, we need to use adjusted closing prices ('Adj Close'). Since there are two layers of indexes, we can select them using multiindex.

```
[3]: data.loc[:, [('Adj Close', 'AAPL'), ('Adj Close', 'GOOG')]].head()
```

```
[3]:
```

	Adj Close	
	AAPL	GOOG
Date		
2021-08-31	151.830002	2909.239990
2021-09-01	152.509995	2916.840088
2021-09-02	153.649994	2884.379883
2021-09-03	154.300003	2895.500000
2021-09-07	156.690002	2910.379883

We can also select the first layer directly.

```
[4]: data.loc[:, 'Adj Close']
```

```
[4]:
```

	AAPL	GOOG
Date		
2021-08-31	151.830002	2909.239990
2021-09-01	152.509995	2916.840088
2021-09-02	153.649994	2884.379883
2021-09-03	154.300003	2895.500000
2021-09-07	156.690002	2910.379883
2021-09-08	155.110001	2897.669922
2021-09-09	154.070007	2898.270020
2021-09-10	148.970001	2838.419922
2021-09-13	149.550003	2869.300049

1.1.3 Computing returns

When there are many stocks, we may have to manipulate the dataframe to get what we want. The following code demonstrates the “chaining” method in Pandas. At last, we use the “pct_change()” method to compute daily returns.

```
[5]: ret = (data
        .unstack()
        .xs('Adj Close', level=0)
        .unstack(level=0)
        .pct_change()
    )
ret.head()
```

```
[5]:           AAPL      GOOG
Date
2021-08-31      NaN      NaN
2021-09-01  0.004479  0.002612
2021-09-02  0.007475 -0.011129
2021-09-03  0.004230  0.003855
2021-09-07  0.015489  0.005139
```

1.1.4 Average returns

```
[6]: ret.mean()
```

```
[6]: AAPL    -0.001790
      GOOG    -0.001682
      dtype: float64
```

1.1.5 Standard derivations and variances

```
[7]: ret.std()
```

```
[7]: AAPL    0.014955
      GOOG    0.010094
      dtype: float64
```

```
[8]: ret.var()
```

```
[8]: AAPL    0.000224
      GOOG    0.000102
      dtype: float64
```

1.1.6 Covariance and correlation matrices

```
[9]: ret.cov()
```

```
[9]:           AAPL      GOOG
AAPL  0.000224  0.000110
GOOG  0.000110  0.000102
```

```
[10]: ret.corr()
```

```
[10]:          AAPL      GOOG
AAPL  1.000000  0.727636
GOOG  0.727636  1.000000
```

1.1.7 Skewness and kurtosis

```
[44]: ret.skew()
```

```
[44]: AAPL    -1.411303
      GOOG    -0.950605
      dtype: float64
```

```
[45]: ret.kurt()
```

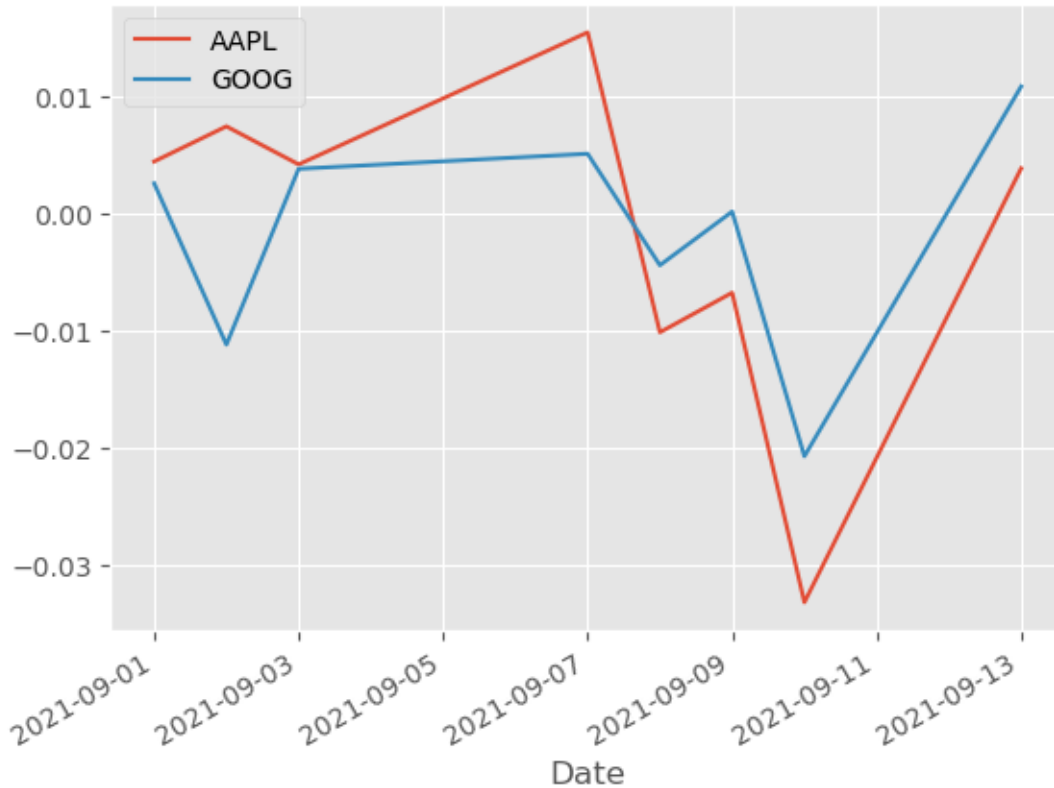
```
[45]: AAPL      2.412562
      GOOG      0.576685
      dtype: float64
```

1.1.8 Plotting

```
[43]: import matplotlib.pyplot as plt

      plt.style.use('ggplot')

      ret.plot()
      plt.legend(ret.columns, loc='best')
      plt.show()
```

1.2 Pandas datareader

This package financial and economic downloads data from multiple sources.

```
[36]: import numpy as np
import pandas as pd
import pandas_datareader as pdr

data_list = pdr.famafrench.get_available_datasets()
data_list = pd.DataFrame(data_list, columns=['Data'])
data_list.head()
```

```
[36]:
```

	Data
0	F-F_Research_Data_Factors
1	F-F_Research_Data_Factors_weekly
2	F-F_Research_Data_Factors_daily
3	F-F_Research_Data_5_Factors_2x3
4	F-F_Research_Data_5_Factors_2x3_daily

```
[37]: industries = pdr.get_data_famafrench('5_Industry_Portfolios', start='1-1-1926')
print(industries['DESCR'])
```

5 Industry Portfolios

This file was created by CMPT_IND_RETs using the 202107 CRSP database. It contains value- and equal-weighted returns for 5 industry portfolios. The portfolios are constructed at the end of June. The annual returns are from January to December. Missing data are indicated by -99.99 or -999. Copyright 2021 Kenneth R. French

```
0 : Average Value Weighted Returns -- Monthly (1141 rows x 5 cols)
1 : Average Equal Weighted Returns -- Monthly (1141 rows x 5 cols)
2 : Average Value Weighted Returns -- Annual (94 rows x 5 cols)
3 : Average Equal Weighted Returns -- Annual (94 rows x 5 cols)
4 : Number of Firms in Portfolios (1141 rows x 5 cols)
5 : Average Firm Size (1141 rows x 5 cols)
6 : Sum of BE / Sum of ME (96 rows x 5 cols)
7 : Value-Weighted Average of BE/ME (96 rows x 5 cols)
```

```
[38]: type(industries)
```

```
[38]: dict
```

```
[40]: industries[0].head()
```

```
[40]:
```

	Cnsmr	Manuf	HiTec	Hlth	Other
Date					
1926-07	5.43	2.73	1.83	1.77	2.13
1926-08	2.76	2.33	2.41	4.25	4.35
1926-09	2.16	-0.44	1.06	0.69	0.29
1926-10	-3.90	-2.42	-2.26	-0.57	-2.84
1926-11	3.70	2.50	3.07	5.42	2.11