# Module 7: Data Wrangling with Pandas

### **CPE311 Computational Thinking with Python**

Submitted by: Belocora, John Rome A.

Performed on: 03/20/2024

Submitted on: 03/20/2024

Submitted to: Engr. Roman M. Richard

## Exercise 1

We want to look at data for the Facebook, Apple, Amazon, Netflix, and Google(FAANG) stocks, but we were given each as separate CSV file. Combine them into a single file and store the dataframe of the FAANG data as faang for the rest of the exercises:

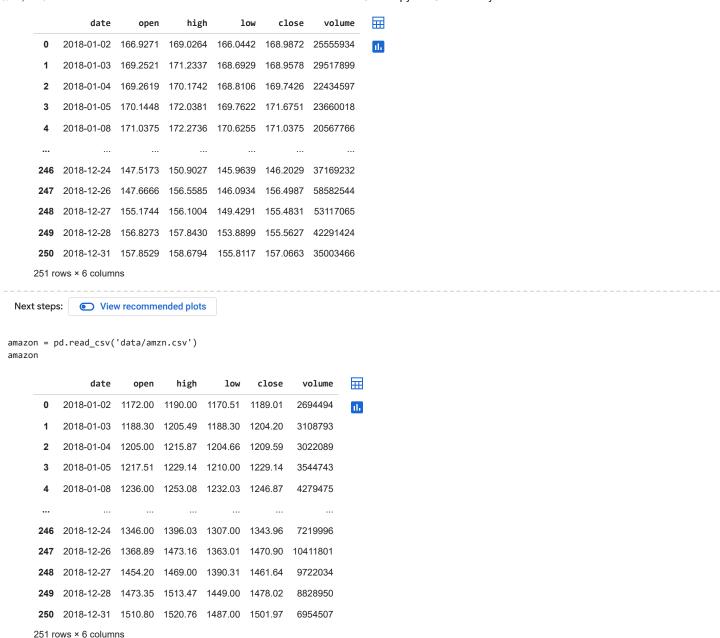
1. Read each file in.

```
import pandas as pd
# Reading each file for each dataset
facebook = pd.read_csv('data/fb.csv')
facebook
```

|                      | date       | open   | high   | low      | close  | volume   | #   |  |  |
|----------------------|------------|--------|--------|----------|--------|----------|-----|--|--|
| 0                    | 2018-01-02 | 177.68 | 181.58 | 177.5500 | 181.42 | 18151903 | ılı |  |  |
| 1                    | 2018-01-03 | 181.88 | 184.78 | 181.3300 | 184.67 | 16886563 |     |  |  |
| 2                    | 2018-01-04 | 184.90 | 186.21 | 184.0996 | 184.33 | 13880896 |     |  |  |
| 3                    | 2018-01-05 | 185.59 | 186.90 | 184.9300 | 186.85 | 13574535 |     |  |  |
| 4                    | 2018-01-08 | 187.20 | 188.90 | 186.3300 | 188.28 | 17994726 |     |  |  |
|                      |            |        |        |          |        |          |     |  |  |
| 246                  | 2018-12-24 | 123.10 | 129.74 | 123.0200 | 124.06 | 22066002 |     |  |  |
| 247                  | 2018-12-26 | 126.00 | 134.24 | 125.8900 | 134.18 | 39723370 |     |  |  |
| 248                  | 2018-12-27 | 132.44 | 134.99 | 129.6700 | 134.52 | 31202509 |     |  |  |
| 249                  | 2018-12-28 | 135.34 | 135.92 | 132.2000 | 133.20 | 22627569 |     |  |  |
| 250                  | 2018-12-31 | 134.45 | 134.64 | 129.9500 | 131.09 | 24625308 |     |  |  |
| 251 rows × 6 columns |            |        |        |          |        |          |     |  |  |



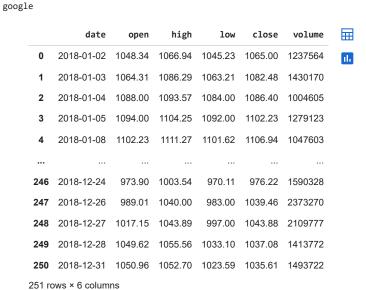
apple = pd.read\_csv('data/aapl.csv') apple



netflix = pd.read\_csv('data/nflx.csv')
netflix



google = pd.read\_csv('data/goog.csv')



Next steps: View recommended plots

2. Add a column to each dataframe, called ticker, indicating the ticker symbol it is for (Apple's is AAPL, for example). This is how you look up a stock. Each file's name is also the ticker symbol, so be sure to capitalize it.

```
# Adding a new column named "ticker" that consist the ticker's symbol
facebook['ticker'] = 'FB'
facebook
```

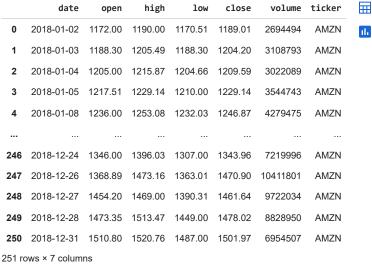


20110110 1 001011111

apple['ticker'] = 'AAPL'
apple

|                      | date       | open     | high     | low      | close    | volume   | ticker | $\blacksquare$ |  |  |
|----------------------|------------|----------|----------|----------|----------|----------|--------|----------------|--|--|
| 0                    | 2018-01-02 | 166.9271 | 169.0264 | 166.0442 | 168.9872 | 25555934 | AAPL   | ılı            |  |  |
| 1                    | 2018-01-03 | 169.2521 | 171.2337 | 168.6929 | 168.9578 | 29517899 | AAPL   |                |  |  |
| 2                    | 2018-01-04 | 169.2619 | 170.1742 | 168.8106 | 169.7426 | 22434597 | AAPL   |                |  |  |
| 3                    | 2018-01-05 | 170.1448 | 172.0381 | 169.7622 | 171.6751 | 23660018 | AAPL   |                |  |  |
| 4                    | 2018-01-08 | 171.0375 | 172.2736 | 170.6255 | 171.0375 | 20567766 | AAPL   |                |  |  |
|                      |            |          |          |          |          |          |        |                |  |  |
| 246                  | 2018-12-24 | 147.5173 | 150.9027 | 145.9639 | 146.2029 | 37169232 | AAPL   |                |  |  |
| 247                  | 2018-12-26 | 147.6666 | 156.5585 | 146.0934 | 156.4987 | 58582544 | AAPL   |                |  |  |
| 248                  | 2018-12-27 | 155.1744 | 156.1004 | 149.4291 | 155.4831 | 53117065 | AAPL   |                |  |  |
| 249                  | 2018-12-28 | 156.8273 | 157.8430 | 153.8899 | 155.5627 | 42291424 | AAPL   |                |  |  |
| 250                  | 2018-12-31 | 157.8529 | 158.6794 | 155.8117 | 157.0663 | 35003466 | AAPL   |                |  |  |
| 251 rows × 7 columns |            |          |          |          |          |          |        |                |  |  |

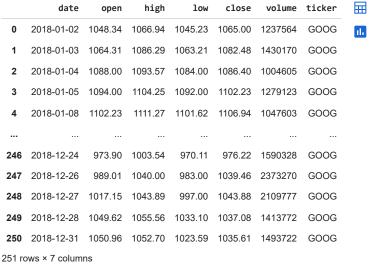
amazon['ticker'] = 'AMZN'
amazon



netflix['ticker'] = 'NFLX' netflix

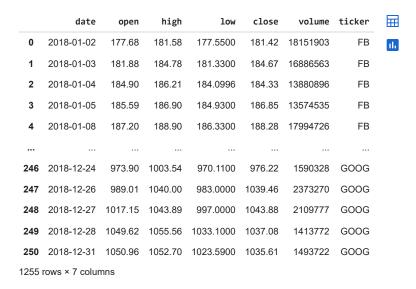
|                      | date       | open   | high     | low      | close   | volume   | ticker |  |  |
|----------------------|------------|--------|----------|----------|---------|----------|--------|--|--|
| 0                    | 2018-01-02 | 196.10 | 201.6500 | 195.4200 | 201.070 | 10966889 | NFLX   |  |  |
| 1                    | 2018-01-03 | 202.05 | 206.2100 | 201.5000 | 205.050 | 8591369  | NFLX   |  |  |
| 2                    | 2018-01-04 | 206.20 | 207.0500 | 204.0006 | 205.630 | 6029616  | NFLX   |  |  |
| 3                    | 2018-01-05 | 207.25 | 210.0200 | 205.5900 | 209.990 | 7033240  | NFLX   |  |  |
| 4                    | 2018-01-08 | 210.02 | 212.5000 | 208.4400 | 212.050 | 5580178  | NFLX   |  |  |
|                      |            |        |          |          |         |          |        |  |  |
| 246                  | 2018-12-24 | 242.00 | 250.6500 | 233.6800 | 233.880 | 9547616  | NFLX   |  |  |
| 247                  | 2018-12-26 | 233.92 | 254.5000 | 231.2300 | 253.670 | 14402735 | NFLX   |  |  |
| 248                  | 2018-12-27 | 250.11 | 255.5900 | 240.1000 | 255.565 | 12235217 | NFLX   |  |  |
| 249                  | 2018-12-28 | 257.94 | 261.9144 | 249.8000 | 256.080 | 10987286 | NFLX   |  |  |
| 250                  | 2018-12-31 | 260.16 | 270.1001 | 260.0000 | 267.660 | 13508920 | NFLX   |  |  |
| 251 rows × 7 columns |            |        |          |          |         |          |        |  |  |

google['ticker'] = 'GOOG' google



\_\_\_\_\_

# Using concat we can append all the dataframes together
faang = pd.concat([facebook, apple, amazon, netflix, google])
faang



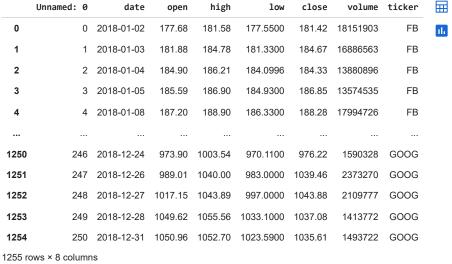
Next steps:



4. Save the result in a CSV file called faang.csv

```
# Converting the faang file into
faang.to_csv('faang.csv')
```

faang = pd.read\_csv('data/faang.csv')
faang



Next steps:

View recommended plots

### Exercise 2

· With faang, use type conversion to change the date column into a datetime and the volume column into integers. Then, sort by date and ticker.

# Converting the data type of date to datetime and volume to integer datatype date = pd.to datetime(faang.date), volume=faang.volume.astype('int'),

#### faang.dtypes

Unnamed: 0 int64 date datetime64[ns] open float64 high float64 float64 low close float64 volume int64 ticker object dtype: object

# Soring the date and ticker in the dataframe faang.sort\_values(by=['date','ticker'], inplace=True) faang

|      | Unnamed: 0 | date       | open      | high      | low       | close     | volume   | ticker |     |
|------|------------|------------|-----------|-----------|-----------|-----------|----------|--------|-----|
| 251  | 0          | 2018-01-02 | 166.9271  | 169.0264  | 166.0442  | 168.9872  | 25555934 | AAPL   | ıl. |
| 502  | 0          | 2018-01-02 | 1172.0000 | 1190.0000 | 1170.5100 | 1189.0100 | 2694494  | AMZN   |     |
| 0    | 0          | 2018-01-02 | 177.6800  | 181.5800  | 177.5500  | 181.4200  | 18151903 | FB     |     |
| 1004 | 0          | 2018-01-02 | 1048.3400 | 1066.9400 | 1045.2300 | 1065.0000 | 1237564  | GOOG   |     |
| 753  | 0          | 2018-01-02 | 196.1000  | 201.6500  | 195.4200  | 201.0700  | 10966889 | NFLX   |     |
|      |            |            |           |           |           |           |          |        |     |
| 501  | 250        | 2018-12-31 | 157.8529  | 158.6794  | 155.8117  | 157.0663  | 35003466 | AAPL   |     |
| 752  | 250        | 2018-12-31 | 1510.8000 | 1520.7600 | 1487.0000 | 1501.9700 | 6954507  | AMZN   |     |
| 250  | 250        | 2018-12-31 | 134.4500  | 134.6400  | 129.9500  | 131.0900  | 24625308 | FB     |     |
| 1254 | 250        | 2018-12-31 | 1050.9600 | 1052.7000 | 1023.5900 | 1035.6100 | 1493722  | GOOG   |     |
| 1003 | 250        | 2018-12-31 | 260.1600  | 270.1001  | 260.0000  | 267.6600  | 13508920 | NFLX   |     |
| 4055 |            |            |           |           |           |           |          |        |     |

1255 rows × 8 columns

• Find the seven rows with the highest value for volume.

# Locating the first 7 rows of volume with the highest value faang.sort\_values(by='volume', ascending=False).head(7)

|     | Unnamed: 0 | date       | open     | high     | low      | close    | volume    | ticker |     |
|-----|------------|------------|----------|----------|----------|----------|-----------|--------|-----|
| 142 | 142        | 2018-07-26 | 174.8900 | 180.1300 | 173.7500 | 176.2600 | 169803668 | FB     | ılı |
| 53  | 53         | 2018-03-20 | 167.4700 | 170.2000 | 161.9500 | 168.1500 | 129851768 | FB     |     |
| 57  | 57         | 2018-03-26 | 160.8200 | 161.1000 | 149.0200 | 160.0600 | 126116634 | FB     |     |
| 54  | 54         | 2018-03-21 | 164.8000 | 173.4000 | 163.3000 | 169.3900 | 106598834 | FB     |     |
| 433 | 182        | 2018-09-21 | 219.0727 | 219.6482 | 215.6097 | 215.9768 | 96246748  | AAPL   |     |
| 496 | 245        | 2018-12-21 | 156.1901 | 157.4845 | 148.9909 | 150.0862 | 95744384  | AAPL   |     |
| 463 | 212        | 2018-11-02 | 207.9295 | 211.9978 | 203.8414 | 205.8755 | 91328654  | AAPL   |     |

• Right now, the data is somewhere between long and wide format. Use melt() to make it completely long format. Hint: date and ticker are