Assignment #4

March 3, 2022

[2]:
```
import pandas as pd
import numpy as np
from tiingo import TiingoClient
import datetime

_MY_SECRET_API_KEY = 'b6cb5b39baf0ee9f3f376a13d7c7707e0c0160b8'
config = {}
config['session'] = True
config['api_key'] = _MY_SECRET_API_KEY
client = TiingoClient(config)

NOW = datetime.datetime.now()

class Stock():
    def __init__(self, ticker: str):
        self._ticker = ticker
        fn = f'{ticker}.csv'
        self._df = None

    @property
    def ticker(self) -> str:
        return self._ticker

    def read_from_csv(self, filename: str = None):
        self._df = pd.read_csv(filename)

    def stock_rows(self) -> int:
        return len(self._df)

    def closes(self) -> pd.Series:
        return self._df['adjClose']

    def max_close(self) -> float:
        return self.closes().max()

    def max_close_np(self) -> float:
        return np.max(self.closes())
```

```
ticker_list = [str(x) for x in input('Enter Stock Tickers Separated by a Space:␣
 ↪').upper().split()]


while True:
    try:
        start_date = datetime.datetime.strptime(input('Enter a start date␣
 ↪(YYYY-MM-DD): '), '%Y-%m-%d')
        if start_date >= NOW:
            print('Date cannot be in the future')
            continue
        while True:
            try:
                end_date = datetime.datetime.strptime(input('Enter a end date␣
 ↪(YYYY-MM-DD): '), '%Y-%m-%d')
                if end_date <= start_date:
                    print('End Date must be later than Start Date')
                    continue
                elif end_date >= NOW:
                    print('Date cannot be in the future')
                    continue
            except ValueError:
                print('Wrong format')
                continue
            else:
                break
    except ValueError:
        print('Wrong format')
        continue
    else:
        break

for stock in ticker_list:
    chart_data = client.get_ticker_price(stock, fmt='csv',␣
 ↪startDate=start_date, endDate=end_date, frequency='daily')

    filename = f'{stock}.csv'
    with open(filename, 'w') as outfile:
        outfile.write(chart_data)


for stock in ticker_list:
    stock1 = Stock(stock)
    print(f'\nStock ticker {stock1.ticker}')
    stock1.read_from_csv(f'{stock}.csv')
```

```
    print(f'Read in {stock1.stock_rows()} records. ')
    print(f'The max close for {stock1.ticker} was ${stock1.max_close():,.2f}')
    print(f'The max close (using NumPy) was ${stock1.max_close_np():,.2f}')
```

```
Enter Stock Tickers Separated by a Space: goog ko tsla aapl
Enter a start date (YYYY-MM-DD): 20210101
Wrong format
Enter a start date (YYYY-MM-DD): 2021-01-01
Enter a end date (YYYY-MM-DD): 2020-12-31
End Date must be later than Start Date
Enter a end date (YYYY-MM-DD): 2022-12-31
Date cannot be in the future
Enter a end date (YYYY-MM-DD): 2021-12-31

Stock ticker GOOG
Read in 252 records.
The max close for GOOG was $3,014.18
The max close (using NumPy) was $3,014.18

Stock ticker KO
Read in 252 records.
The max close for KO was $59.21
The max close (using NumPy) was $59.21

Stock ticker TSLA
Read in 252 records.
The max close for TSLA was $1,229.91
The max close (using NumPy) was $1,229.91

Stock ticker AAPL
Read in 252 records.
The max close for AAPL was $180.10
The max close (using NumPy) was $180.10
```

[ ]: