PEP Capstone Checkpoint 1 Documentation

Team 1

The data source we chose was from https://www.alphavantage.co/. It is a dataset of the changes in price of Amazon stock each day from 1999 to the present day.

To begin, we used the requests library in python to generate a request using the AMZN symbol, a full output size and my API key, after import requests and pandas.

```
response = rq.get("https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=AMZN&outputsize=full&apikey=KVUJT3KI90X21VGB")
✓ 0.3s
```

Moving along, we got the response.json() and then specifically selected the “Time Series (Daily)” section, which is our raw data. From that data, we created a dataframe, however it was incorrect so the .T attribute was applied to transpose the dataframe.

```
# Getting a dataframe from data
data = response.json()
time_series_data = data["Time Series (Daily)"]

df = pd.DataFrame(time_series_data).T
✓ 0.1s
```

After which, we then renamed all the columns to be clear and proper, have the date as the index of the dataframe.

```
# Renaming columns to be clearer
df.rename(columns={'1. open': 'open', '2. high': 'high', '3. low': 'low', '4. close': 'close', '5. volume': 'volume'}, inplace=True)
✓ 0.0s
```

On the next page is the full code for checkpoint 1, as well as a data sample.

```
[1]  import requests as rq
     import pandas as pd
     ✓ 0.5s
```

```
[2]  response = rq.get("https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=AMZN&outputsize=full&apikey=KVUJT3KI90X21VGB")
     ✓ 4.1s
```

```
[3]  # Getting a dataframe from data
     data = response.json()
     time_series_data = data["Time Series (Daily)"]

     df = pd.DataFrame(time_series_data).T
     ✓ 0.1s
```

```
[4]  # Renaming columns to be clearer
     df.rename(columns={'1. open': 'open', '2. high': 'high', '3. low': 'low', '4. close': 'close', '5. volume': 'volume'}, inplace=True)
     ✓ 0.0s
```

```
[5]  df
     ✓ 0.0s
```

	open	high	low	close	volume
2024-02-16	168.7400	170.4200	167.1700	169.5100	48107744
2024-02-15	170.5800	171.1700	167.5900	169.8000	49855196
2024-02-14	169.2100	171.2100	168.2800	170.9800	42815544
2024-02-13	167.7300	170.9500	165.7500	168.6400	56345122
2024-02-12	174.8000	175.3900	171.5400	172.3400	51050440
...	...	...	...	...	...
1999-11-05	64.7500	65.5000	62.2500	64.9400	11091400
1999-11-04	67.1900	67.1900	61.0000	63.0600	16759200
1999-11-03	68.1900	68.5000	65.0000	65.8100	10772100
1999-11-02	69.7500	70.0000	65.0600	66.4400	13243200
1999-11-01	68.0600	71.8800	66.3100	69.1300	12824100

6113 rows × 5 columns