

Chapter 5 - Exercise 2: Giao dịch chứng khoán

Cho 3 file .csv sau:

- **stocks1.csv** : *date, symbol, open, high, low, close, volume* : chứa thông tin giao dịch chứng khoán các công ty khác nhau
- **stocks2.csv** : *date, symbol, open, high, low, close, volume* : chứa thông tin giao dịch chứng khoán các công ty khác nhau
- **companies.csv** : *name, employees, headquarters_city, headquarters_state* : chứa thông tin về trụ sở và số lượng nhân viên cho một công ty cụ thể

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: # Câu 1a: Đọc file stocks1.csv => đưa dữ liệu vào stocks1
stocks1 = pd.read_csv('stock_trading_data/stocks1.csv')
# Hiển thị 5 dòng dữ liệu đầu của stocks1
stocks1.head()
```

Out[2]:

	date	symbol	open	high	low	close	volume
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358
2	05-03-19	AMZN	1702.95	1707.80	1689.01	1692.43	3681522
3	06-03-19	AMZN	1695.97	NaN	NaN	1668.95	3996001
4	07-03-19	AMZN	1667.37	1669.75	1620.51	1625.95	4957017

```
In [3]: # Hiển thị 5 dòng dữ liệu cuối của stocks1
stocks1.tail()
```

Out[3]:

	date	symbol	open	high	low	close	volume
10	01-03-19	GOOG	1124.90	1142.97	1124.75	1140.99	1450316
11	04-03-19	GOOG	1146.99	1158.28	1130.69	1147.80	1446047
12	05-03-19	GOOG	1150.06	NaN	NaN	1162.03	1443174
13	06-03-19	GOOG	1162.49	1167.57	1155.49	1157.86	1099289
14	07-03-19	GOOG	1155.72	1156.76	1134.91	1143.30	1166559

```
In [4]: # Cho biết kiểu dữ liệu (dtype) của các cột của stocks1
stocks1.dtypes
```

```
Out[4]: date          object
symbol          object
open           float64
high           float64
low            float64
close          float64
volume         int64
dtype: object
```



```
In [5]: # Xem thông tin (info) của stocks1
stocks1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   date    15 non-null      object
 1   symbol  15 non-null      object
 2   open    15 non-null      float64
 3   high    13 non-null      float64
 4   low     13 non-null      float64
 5   close   15 non-null      float64
 6   volume  15 non-null      int64
dtypes: float64(4), int64(1), object(2)
memory usage: 968.0+ bytes
```

```
In [6]: # Câu 1b: Đọc file stocks2.csv => đưa dữ liệu vào stocks2
stocks2 = pd.read_csv('stock_trading_data/stocks2.csv')
# Hiển thị 5 dòng dữ liệu đầu của stocks2
stocks2.head()
```

Out[6]:

	date	symbol	open	high	low	close	volume
0	01-03-19	FB	162.60	163.132	161.69	162.28	11097770
1	04-03-19	FB	163.90	167.500	163.83	167.37	18894689
2	05-03-19	FB	167.37	171.880	166.55	171.26	28187890
3	06-03-19	FB	172.90	173.570	171.27	172.51	21531723
4	07-03-19	FB	171.50	171.740	167.61	169.13	18306504

```
In [7]: # Hiển thị 5 dòng dữ liệu cuối của stocks2
stocks2.tail()
```

Out[7]:

	date	symbol	open	high	low	close	volume
5	01-03-19	TSLA	306.94	307.1300	291.90	294.79	22911375
6	04-03-19	TSLA	298.12	299.0000	282.78	285.36	17096818
7	05-03-19	TSLA	282.00	284.0000	270.10	276.54	18764740
8	06-03-19	TSLA	276.48	281.5058	274.39	276.24	10335485
9	07-03-19	TSLA	278.84	284.7000	274.25	276.59	9442483

```
In [8]: # Cho biết kiểu dữ liệu (dtype) của các cột của stocks2
stocks2.dtypes
```

Out[8]:

date	object
symbol	object
open	float64
high	float64
low	float64
close	float64
volume	int64

dtype: object


```
In [9]: # Xem thông tin (info) của stocks2
stocks2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   date        10 non-null    object
1   symbol      10 non-null    object
2   open        10 non-null    float64
3   high        10 non-null    float64
4   low         10 non-null    float64
5   close       10 non-null    float64
6   volume      10 non-null    int64
dtypes: float64(4), int64(1), object(2)
memory usage: 688.0+ bytes
```

```
In [10]: # Câu 1c: Đọc file companies.csv => đưa dữ liệu vào companies
companies = pd.read_csv('stock_trading_data/companies.csv')
# Xem dữ liệu của companies
companies
```

Out[10]:

	name	employees	headquarters_city	headquarters_state
0	AMZN	613300	Seattle	WA
1	GOOG	98771	Mountain View	CA
2	AAPL	132000	Cupertino	CA
3	FB	48268	Menlo Park	CA
4	TSLA	48016	Palo Alto	CA

```
In [11]: # Cho biết kiểu dữ liệu (dtype) của các cột của companies
companies.dtypes
```

Out[11]:

name	object
employees	int64
headquarters_city	object
headquarters_state	object
dtype:	object

```
In [12]: # Xem thông tin (info) của companies
companies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
#   Column              Non-Null Count  Dtype
---  -
0   name                5 non-null     object
1   employees            5 non-null     int64
2   headquarters_city    5 non-null     object
3   headquarters_state   5 non-null     object
dtypes: int64(1), object(3)
memory usage: 288.0+ bytes
```



```
In [13]: # Câu 2: Cho biết trong stocks1 có dữ liệu Null hay không?
stocks1.isnull().any()
```

```
Out[13]: date      False
symbol    False
open      False
high      True
low       True
close     False
volume    False
dtype: bool
```

```
In [14]: stocks1.isnull().sum()
```

```
Out[14]: date      0
symbol    0
open      0
high      2
low       2
close     0
volume    0
dtype: int64
```

```
In [15]: stocks1[stocks1['high'].isnull()]
```

```
Out[15]:
```

	date	symbol	open	high	low	close	volume
3	06-03-19	AMZN	1695.97	NaN	NaN	1668.95	3996001
12	05-03-19	GOOG	1150.06	NaN	NaN	1162.03	1443174

```
In [16]: # Nếu có, hãy thay thế với quy tắc sau:
# Nếu Null cột 'high' thì thay bằng giá trị max trên cột 'high' của mã chứng khoán đó
# Nếu Null cột 'low' thì thay bằng giá trị min trên cột 'low' của mã chứng khoán đó
stocks1['high'].fillna(stocks1.groupby(['symbol'])['high'].transform(max), inplace=True)
stocks1['low'].fillna(stocks1.groupby(['symbol'])['low'].transform(min), inplace=True)
```

```
In [17]: stocks1.iloc[[3,12]]
```

```
Out[17]:
```

	date	symbol	open	high	low	close	volume
3	06-03-19	AMZN	1695.97	1709.43	1620.51	1668.95	3996001
12	05-03-19	GOOG	1150.06	1167.57	1124.75	1162.03	1443174

In [18]:

```
# Câu 3: Tạo dataframe stocks bằng cách gộp stocks1 và stocks2 theo dòng
stocks = pd.concat([stocks1, stocks2], ignore_index=True)
# Xem 15 dòng dữ liệu cuối của stocks
stocks.tail(15)
```

Out[18]:

	date	symbol	open	high	low	close	volume
10	01-03-19	GOOG	1124.90	1142.9700	1124.75	1140.99	1450316
11	04-03-19	GOOG	1146.99	1158.2800	1130.69	1147.80	1446047
12	05-03-19	GOOG	1150.06	1167.5700	1124.75	1162.03	1443174
13	06-03-19	GOOG	1162.49	1167.5700	1155.49	1157.86	1099289
14	07-03-19	GOOG	1155.72	1156.7600	1134.91	1143.30	1166559
15	01-03-19	FB	162.60	163.1320	161.69	162.28	11097770
16	04-03-19	FB	163.90	167.5000	163.83	167.37	18894689
17	05-03-19	FB	167.37	171.8800	166.55	171.26	28187890
18	06-03-19	FB	172.90	173.5700	171.27	172.51	21531723
19	07-03-19	FB	171.50	171.7400	167.61	169.13	18306504
20	01-03-19	TSLA	306.94	307.1300	291.90	294.79	22911375
21	04-03-19	TSLA	298.12	299.0000	282.78	285.36	17096818
22	05-03-19	TSLA	282.00	284.0000	270.10	276.54	18764740
23	06-03-19	TSLA	276.48	281.5058	274.39	276.24	10335485
24	07-03-19	TSLA	278.84	284.7000	274.25	276.59	9442483

In [19]:

```
# Câu 4: Tạo dataframe stocks_companies bằng cách gộp stocks và companies
stocks_companies = stocks.merge(companies, left_on='symbol', right_on='name', how='inner')
# Xem 5 dòng dữ liệu đầu của stocks_companies
stocks_companies.head()
```

Out[19]:

	date	symbol	open	high	low	close	volume	name	employees	headquarters_city	headq
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877	AMZN	613300	Seattle	
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358	AMZN	613300	Seattle	
2	05-03-19	AMZN	1702.95	1707.80	1689.01	1692.43	3681522	AMZN	613300	Seattle	
3	06-03-19	AMZN	1695.97	1709.43	1620.51	1668.95	3996001	AMZN	613300	Seattle	
4	07-03-19	AMZN	1667.37	1669.75	1620.51	1625.95	4957017	AMZN	613300	Seattle	


```
In [20]: # Câu 5: Cho biết giá (open, high, low, close) trung bình
# và volume trung bình của mỗi công ty
cols = ['symbol', 'open', 'high', 'low', 'close', 'volume']
stocks_companies[cols].groupby('symbol').mean()
```

Out[20]:

	open	high	low	close	volume
symbol					
AAPL	174.890	175.76600	173.472	174.674	23733309.4
AMZN	1681.284	1694.13400	1651.078	1671.046	4755355.0
FB	167.654	169.56440	166.190	168.510	19603715.2
GOOG	1148.032	1158.63000	1134.118	1150.396	1321077.0
TSLA	288.476	291.26716	278.684	281.904	15710180.2

```
In [21]: # Câu 6: Cho biết giá đóng cửa (close) trung bình, lớn nhất và nhỏ nhất ở mỗi công ty
stocks_companies.groupby('symbol').close.agg(['mean', 'min', 'max'])
```

Out[21]:

	mean	min	max
symbol			
AAPL	174.674	172.50	175.85
AMZN	1671.046	1625.95	1696.17
FB	168.510	162.28	172.51
GOOG	1150.396	1140.99	1162.03
TSLA	281.904	276.24	294.79


```
In [22]: # Câu 7: Tạo cột parsed_time trong stocks_companies
# bằng cách đổi thời gian sang định dạng DateTime
stocks_companies['parsed_time'] = pd.to_datetime(stocks_companies['date'])
# Cho biết kiểu dữ liệu của cột parsed_time
print(stocks_companies['parsed_time'].dtype)
# Hiển thị 5 dòng dữ liệu đầu của stocks_companies
stocks_companies.head()
```

datetime64[ns]

Out[22]:

	date	symbol	open	high	low	close	volume	name	employees	headquarters_city	headq
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877	AMZN	613300	Seattle	
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358	AMZN	613300	Seattle	
2	05-03-19	AMZN	1702.95	1707.80	1689.01	1692.43	3681522	AMZN	613300	Seattle	
3	06-03-19	AMZN	1695.97	1709.43	1620.51	1668.95	3996001	AMZN	613300	Seattle	
4	07-03-19	AMZN	1667.37	1669.75	1620.51	1625.95	4957017	AMZN	613300	Seattle	

```
In [23]: # Câu 8: Thêm cột result, nếu giá 'close' > 'open'
# thì cột result có giá trị 'up', ngược lại 'down'
stocks_companies.loc[stocks_companies['close'] > stocks_companies['open'], 'result'] =
stocks_companies.loc[stocks_companies['close'] < stocks_companies['open'], 'result'] =
stocks_companies.head()
```

Out[23]:

	date	symbol	open	high	low	close	volume	name	employees	headquarters_city	headq
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877	AMZN	613300	Seattle	
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358	AMZN	613300	Seattle	
2	05-03-19	AMZN	1702.95	1707.80	1689.01	1692.43	3681522	AMZN	613300	Seattle	
3	06-03-19	AMZN	1695.97	1709.43	1620.51	1668.95	3996001	AMZN	613300	Seattle	
4	07-03-19	AMZN	1667.37	1669.75	1620.51	1625.95	4957017	AMZN	613300	Seattle	

In []:

