程式設計與應用　期中作業

全局變數 & 套件包

ps:做完才發現有 ticker().download()可以用，以後要記得先看完文檔...

```
import os
import time
from datetime import datetime

import pandas as pd
import requests
from IPython.display import Markdown
from yfinance import Ticker

session = requests.Session()
session.headers['User-agent'] = (
    'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
ML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'
)


class StockData:
    def __init__(
        self,
        ticker: str,
        name: str,
        session: requests.Session,
        start: str,
        end: str,
    ):
        self.name = name
        self.getData(ticker, session, start, end)
        self.ax = None

    def getData(
        self, ticker: str, session: requests.Session, start: str, end:
 str
    ) -> None:
        printName = f'{ticker}-{start.date()}_{end.date()}'
        filename = f'data/{printName}.csv'

        if os.path.exists(filename):
            print(f'讀取 {printName} 資料: {filename}')
            self.data = pd.read_csv(filename, index_col=0, parse_dates
=True)
```

```
            self.data.index = pd.to_datetime(self.data.index, utc=Tru
e)
        else:
            print(f'下載 {printName} 資料...')
            self.data = Ticker(ticker, session=session).history(
                start=start, end=end
            )
            if self.data is None:
                print(f'找不到 {printName} 的資料')
            else:
                self.data.to_csv(filename)
                print(f'{printName} 存檔至 {filename}')

    def locateRange(self, start: str, end: str) -> pd.DataFrame:
        if self.data is None:
            raise ValueError(f'{self.name} 未初始化')
        return self.data.loc[start:end]
```

第一大題：取得數據

使用 yfinance 抓取特斯拉、福特和通用汽車的股票價格（2013-2023 年）。

```
start = datetime(2013, 1, 1)
end = datetime(2023, 1, 1)

tesla_first = StockData('TSLA', 'Tesla', session, start, end)
time.sleep(1)
ford_first = StockData('F', 'Ford', session, start, end)
time.sleep(1)
gm_first = StockData('GM', 'GM', session, start, end)
time.sleep(1)

# 設定顯示選項
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 10)
pd.set_option('display.width', 1000)
pd.set_option('display.float_format', '{:.2f}'.format)

# 列印數據
display(Markdown('### **Tesla**'))
display(tesla_first.data.head())
```

```
display(Markdown('### **Ford**'))
display(ford_first.data.head())

display(Markdown('### **GM**'))
display(gm_first.data.head())
```

output:

```
讀取 TSLA-2013-01-01_2023-01-01 資料: data/TSLA-2013-01-01_2023-01-01.csv
讀取 F-2013-01-01_2023-01-01 資料: data/F-2013-01-01_2023-01-01.csv
讀取 GM-2013-01-01_2023-01-01 資料: data/GM-2013-01-01_2023-01-01.csv
<IPython.core.display.Markdown object>
                           Open  High  Low  Close    Volume  Dividends  Stock Splits
Date
2013-01-02 05:00:00+00:00  2.33  2.36 2.31   2.36  17922000       0.00          0.00
2013-01-03 05:00:00+00:00  2.35  2.36 2.32   2.32  11130000       0.00          0.00
2013-01-04 05:00:00+00:00  2.32  2.32 2.26   2.29  10110000       0.00          0.00
2013-01-07 05:00:00+00:00  2.32  2.32 2.26   2.29   6630000       0.00          0.00
2013-01-08 05:00:00+00:00  2.30  2.30 2.21   2.25  19260000       0.00          0.00
<IPython.core.display.Markdown object>
                           Open  High  Low  Close     Volume  Dividends  Stock Splits
Date
2013-01-02 05:00:00+00:00  6.93  6.96 6.81   6.92   75274700       0.00          0.00
2013-01-03 05:00:00+00:00  6.94  7.18 6.84   7.05  121284700       0.00          0.00
2013-01-04 05:00:00+00:00  7.08  7.13 6.99   7.11   54669900       0.00          0.00
2013-01-07 05:00:00+00:00  7.08  7.12 6.99   7.04   43482400       0.00          0.00
2013-01-08 05:00:00+00:00  7.01  7.04 6.92   6.99   46336200       0.00          0.00
<IPython.core.display.Markdown object>
                           Open  High   Low  Close    Volume  Dividends  Stock Splits
Date
2013-01-02 05:00:00+00:00 21.98 22.04 21.39  21.77  15605900       0.00          0.00
2013-01-03 05:00:00+00:00 21.71 22.59 21.55  22.28  22903800       0.00          0.00
2013-01-04 05:00:00+00:00 22.22 22.31 21.84  22.31   8066600       0.00          0.00
2013-01-07 05:00:00+00:00 22.15 22.31 21.89  22.16   8385300       0.00          0.00
2013-01-08 05:00:00+00:00 22.16 22.30 21.71  21.95  10435200       0.00          0.00
```

第二大題：視覺化

取得特斯拉、福特和通用汽車的股票價格 (2012 年 1 月)。

並使用 matplotlib 視覺化(蠟燭圖)

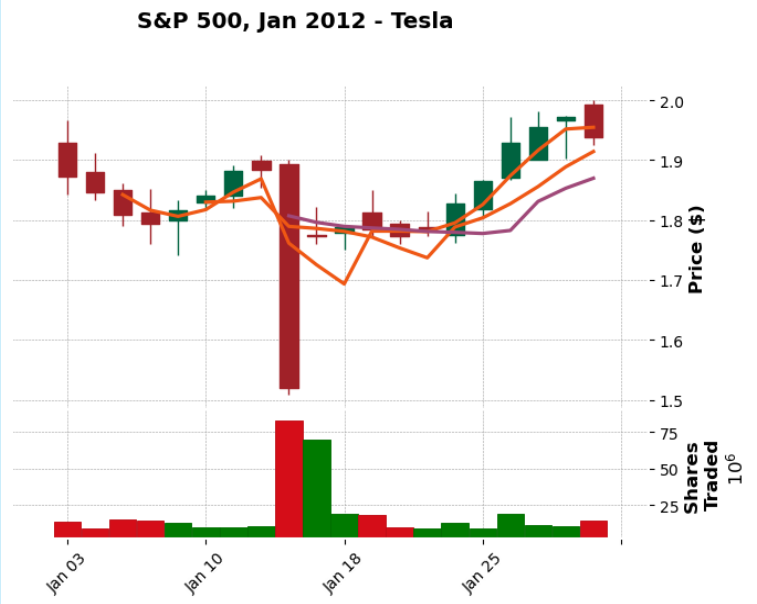
1. 每家各別視覺化
2. 三家合併到同一張圖中做比較

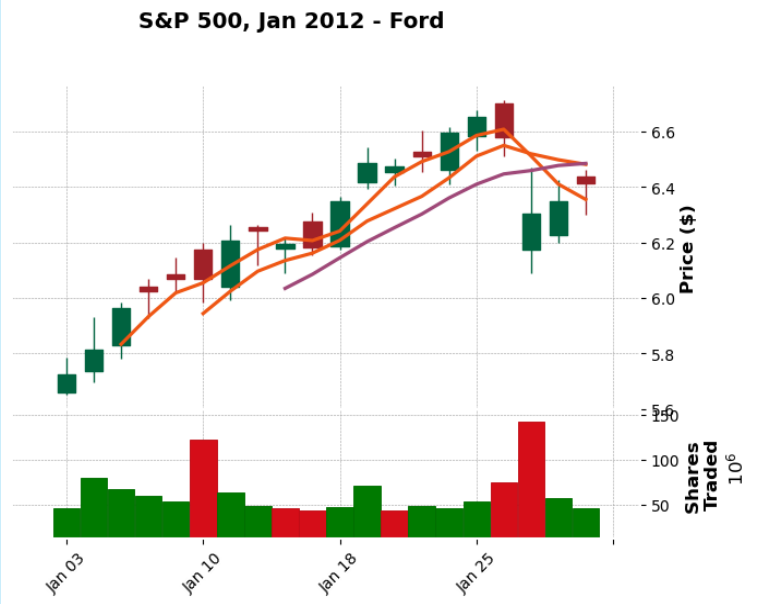
```
import matplotlib.pyplot as plt
import mplfinance as mpf
```

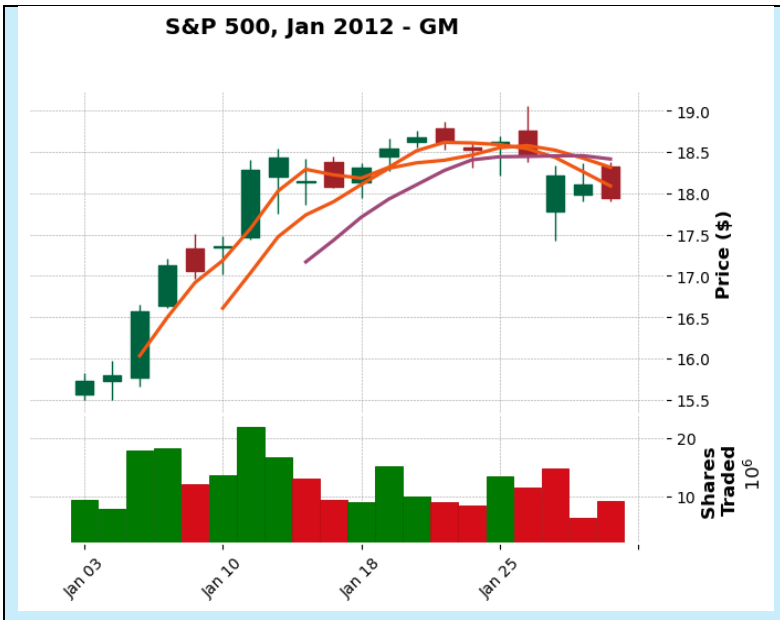
```python
start = datetime(2012, 1, 1)
end = datetime(2012, 12, 31)

tesla_second = StockData('TSLA', 'Tesla', session, start, end)
time.sleep(1)
ford_second = StockData('F', 'Ford', session, start, end)
time.sleep(1)
gm_second = StockData('GM', 'GM', session, start, end)
time.sleep(1)

# 第一小題
for stock in [tesla_second, ford_second, gm_second]:
    mpf.plot(
        stock.locateRange('2012-01', '2012-01'),
        type='candle',
        style='charles',
        title=f'S&P 500, Jan 2012 - {stock.name}',
        ylabel='Price ($)',
        ylabel_lower='Shares \nTraded',
        volume=True,
        mav=(3, 6, 9),
    )

# 第二小題
addplot = [
    mpf.make_addplot(
        ford_second.locateRange('2012-01', '2012-01'),
        type='candle',
        panel=1,
        ylabel='Ford',
    ),
    mpf.make_addplot(
        gm_second.locateRange('2012-01', '2012-01'),
        type='candle',
        panel=2,
        ylabel='GM',
    ),
]

mpf.plot(
    tesla_second.locateRange('2012-01', '2012-01'),
    type='candle',
    style='charles',
    addplot=addplot,
    panel_ratios=(1, 1, 1),
    title='Comparison: Tesla vs Ford vs GM',
    ylabel='Tesla',
)
```

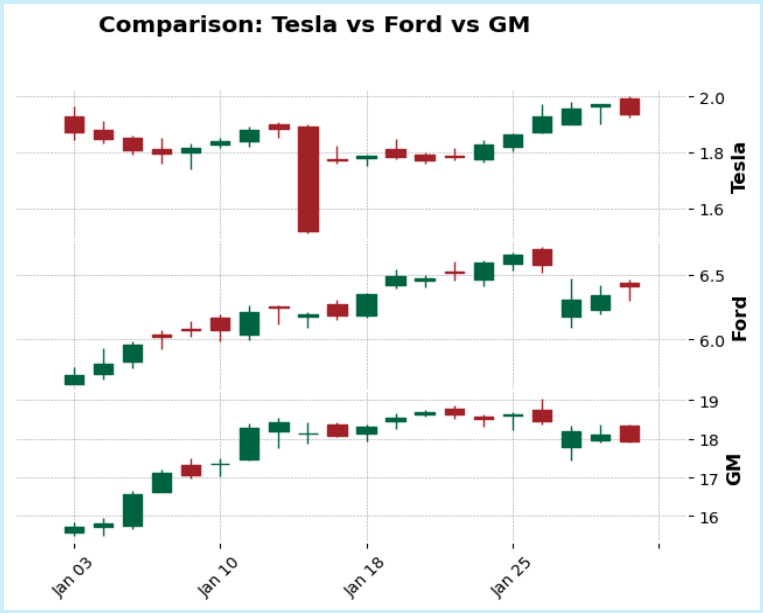
Output:

```
讀取 TSLA-2012-01-01_2012-12-31 資料: data/TSLA-2012-01-01_2012-12-31.csv
讀取 F-2012-01-01_2012-12-31 資料: data/F-2012-01-01_2012-12-31.csv
讀取 GM-2012-01-01_2012-12-31 資料: data/GM-2012-01-01_2012-12-31.csv
```






S&P 500, Jan 2012 - GM
19.0
18.5
18.0
17.5
17.0
16.5
16.0
15.5
Price ($)
20
10
Shares Traded
10^6
Jan 03
Jan 10
Jan 18
Jan 25


Comparison: Tesla vs Ford vs GM
2.0
1.8
1.6
Tesla
6.5
6.0
Ford
19
18
17
16
GM
Jan 03
Jan 10
Jan 18
Jan 25

第三大題:基本財務分析

1. 為每個資料框建立一個名為 `returns` 的新欄位。該列的值將根據收盤價列計算得出。有兩種方法可以實現：一種是使用 `.shift()` 方法，並按照上述公式進行簡單計算；另一種是使用 `pandas` 內建的 `pct_change` 方法。

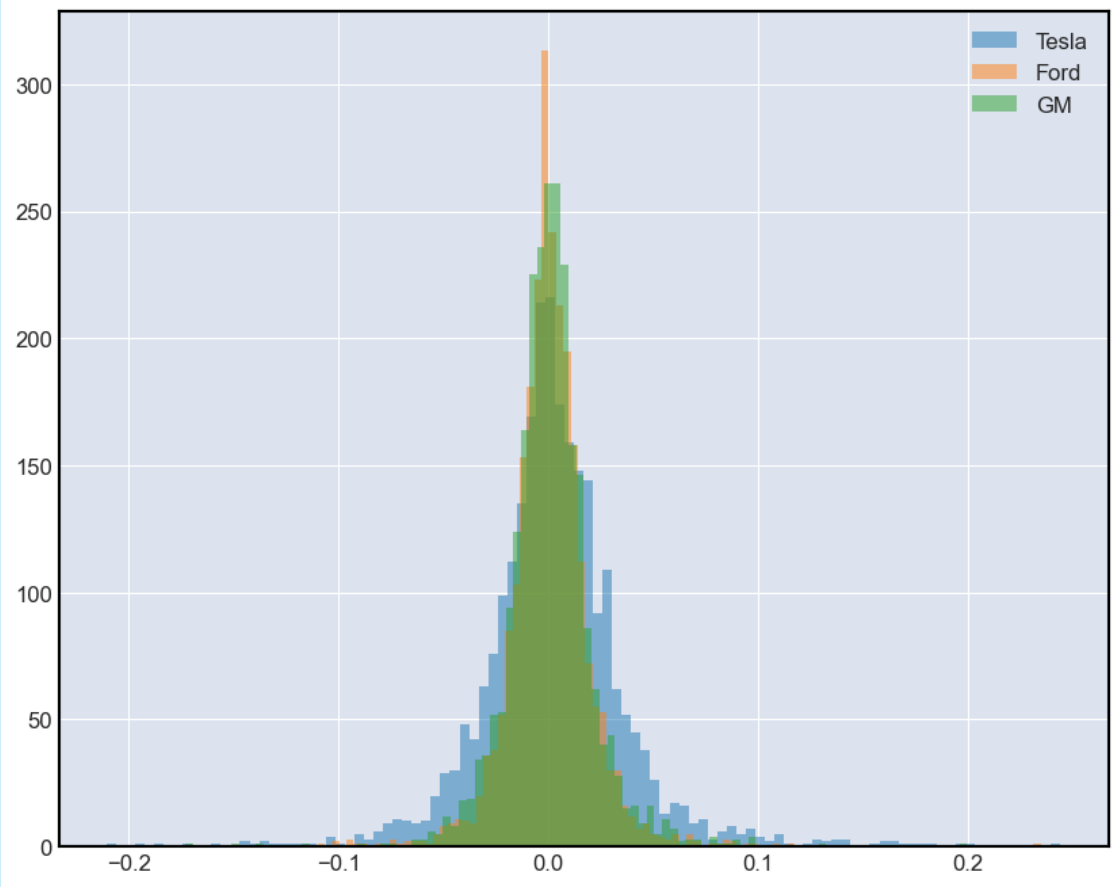
2. 繪製每家公司收益率的直方圖。可以分別繪製，也可以將它們疊在一起繪製。

```
start = datetime(2012, 1, 1)
end = datetime(2022, 1, 1)

tesla_third = StockData('TSLA', 'Tesla', session, start, end)
ford_third = StockData('F', 'Ford', session, start, end)
gm_third = StockData('GM', 'General Motors', session, start, end)


# 方法一
# tesla_third.data['returns'] = (tesla_third.data['Close'] / tesla_thi
rd.data['Close'].shift(1) ) - 1
# ford_third.data['returns'] = (ford_third.data['Close'] / ford_third.
data['Close'].shift(1) ) - 1
# gm_third.data['returns'] = (gm_third.data['Close'] / gm_third.data['
Close'].shift(1) ) - 1

# 方法二
tesla_third.data['returns'] = tesla_third.data['Close'].pct_change(1)
ford_third.data['returns'] = ford_third.data['Close'].pct_change(1)
gm_third.data['returns'] = gm_third.data['Close'].pct_change(1)

# 第一小題
display(Markdown('### ** Tesla **'))
display(tesla_third.data['returns'].head())

display(Markdown('### ** Ford **'))
display(ford_third.data['returns'].head())

display(Markdown('### ** GM **'))
display(gm_third.data['returns'].head())

# 第二小題
tesla_third.data['returns'].hist(
    bins=100, label='Tesla', figsize=(10, 8), alpha=0.5
)
ford_third.data['returns'].hist(
```

```
    bins=100, label='Ford', figsize=(10, 8), alpha=0.5
)
gm_third.data['returns'].hist(
    bins=100,
    label='GM',
    figsize=(10, 8),
    alpha=0.5,
)

plt.legend()
plt.show()
```

Output:

```
讀取 TSLA-2012-01-01_2022-01-01 資料: data/TSLA-2012-01-01_2022-01-01.csv
讀取 F-2012-01-01_2022-01-01 資料: data/F-2012-01-01_2022-01-01.csv
讀取 GM-2012-01-01_2022-01-01 資料: data/GM-2012-01-01_2022-01-01.csv
```



第四大題：每日累積收益率

每日收益率 = (每日收盤價 - 昨日收盤價) / 昨日收盤價

累積收益率 = 目前價格 / 買入價格

df[daily_cumulative_return] = (1 + df[pct_daily_return]).cumprod()

ps: pct_daily_return 就是上一題的函式。

1. (20%) Please Create a cumulative daily return column for each car company's dataframe, response the source code and results in your document.

2. (5%) Please plot three companies' Cumulative Return columns against the time series index in one figure. And answer which stock showed the highest return for a $1 invested? Which showed the lowest?

```
start = datetime(2025, 1, 1)
end = datetime(2025, 11, 26)

tesla_fourth = StockData('TSLA', 'tesla', session, start, end)
time.sleep(1)
ford_fourth = StockData('F', 'ford', session, start, end)
time.sleep(1)
gm_fourth = StockData('GM', 'gm', session, start, end)
time.sleep(1)

tesla_fourth.data['daily_cumulative_return'] = (
    1 + tesla_fourth.data['Close'].pct_change(1).fillna(0)
).cumprod()
ford_fourth.data['daily_cumulative_return'] = (
    1 + ford_fourth.data['Close'].pct_change(1).fillna(0)
).cumprod()
gm_fourth.data['daily_cumulative_return'] = (
    1 + gm_fourth.data['Close'].pct_change(1).fillna(0)
).cumprod()


# 第一小題
display(Markdown('### **Tesla**'))
display(tesla_fourth.data['daily_cumulative_return'])

display(Markdown('### **Ford**'))
```

```
display(ford_fourth.data['daily_cumulative_return'])

display(Markdown('### **GM**'))
display(gm_fourth.data['daily_cumulative_return'])

# 第二小題
tesla_fourth.data['daily_cumulative_return'].plot(label='Tesla')
ford_fourth.data['daily_cumulative_return'].plot(label='Ford')
gm_fourth.data['daily_cumulative_return'].plot(label='GM')

plt.legend()
plt.show()
```

Output:

```
讀取 TSLA-2025-01-01_2025-11-26 資料: data/TSLA-2025-01-01_2025-11-26.csv
讀取 F-2025-01-01_2025-11-26 資料: data/F-2025-01-01_2025-11-26.csv
讀取 GM-2025-01-01_2025-11-26 資料: data/GM-2025-01-01_2025-11-26.csv
<IPython.core.display.Markdown object>
Date
2025-01-02 05:00:00+00:00   1.00
2025-01-03 05:00:00+00:00   1.08
2025-01-06 05:00:00+00:00   1.08
2025-01-07 05:00:00+00:00   1.04
2025-01-08 05:00:00+00:00   1.04
                           ...
2025-11-19 05:00:00+00:00   1.07
2025-11-20 05:00:00+00:00   1.04
2025-11-21 05:00:00+00:00   1.03
2025-11-24 05:00:00+00:00   1.10
2025-11-25 05:00:00+00:00   1.11
Name: daily_cumulative_return, Length: 226, dtype: float64
<IPython.core.display.Markdown object>
Date
2025-01-02 05:00:00+00:00   1.00
2025-01-03 05:00:00+00:00   1.02
2025-01-06 05:00:00+00:00   1.03
2025-01-07 05:00:00+00:00   1.01
2025-01-08 05:00:00+00:00   1.01
                           ...
2025-11-19 05:00:00+00:00   1.44
2025-11-20 05:00:00+00:00   1.38
2025-11-21 05:00:00+00:00   1.43
2025-11-24 05:00:00+00:00   1.44
2025-11-25 05:00:00+00:00   1.47
Name: daily_cumulative_return, Length: 226, dtype: float64
<IPython.core.display.Markdown object>
Date
2025-01-02 05:00:00+00:00   1.00
2025-01-03 05:00:00+00:00   1.01
2025-01-06 05:00:00+00:00   1.04
2025-01-07 05:00:00+00:00   1.01
2025-01-08 05:00:00+00:00   0.99
                           ...
2025-11-19 05:00:00+00:00   1.34
2025-11-20 05:00:00+00:00   1.34
2025-11-21 05:00:00+00:00   1.38
2025-11-24 05:00:00+00:00   1.39
2025-11-25 05:00:00+00:00   1.43
Name: daily_cumulative_return, Length: 226, dtype: float64
```

第二小題: 根據圖表投資 Ford 回報最高(GM 也不錯)， Tesla 回報最低。

結論

學習筆記:

1. mamba 真的很好用，可以快速的建置環境，而且可以快速且乾淨的安裝套件，相信社群是對的。
2. python 各個套件的使用方法，以及如何使用套件來完成特定的任務。
3. 了解資料分析的粗略步驟，因為資料是已經被清整過的，感覺少了很多困難度。
4. 使用 Markdown 做標題，這是問 AI 的。
5. 學會怎麼使用 jupyter notebook 來呈現輸出(但是我也不知道是不是這樣用)。
6. 學會使用 jupyter notebook 匯出檔案。
7. Session 的設計有點多餘了，應該直接調用宣告的 session。
8. 永遠記得**套件包的文檔**一定要概略的看完再開始寫...

展望:

1. 需要學習財報的各種指標，才能跟這個課程整合出一個完整的分析。

提問:

1. 是不是可以在其他檔案包好物件，然後在 jupyter notebook 中引用，這樣應該會更乾淨?