Phòng thí nghiệm thực hành: Phân tích dữ liệu cổ phiếu/doanh thu lịch sử và xây dựng bảng thông tin

Trích xuất dữ liệu cần thiết từ một tập dữ liệu và hiển thị nó là một phần cần thiết của khoa học dữ liệu; do đó, các cá nhân có thể đưa ra quyết định đúng đắn dựa trên dữ liệu. Trong bài tập này, bạn sẽ trích xuất một số dữ liệu chứng khoán, sau đó bạn sẽ hiển thị dữ liệu này trong biểu đồ.

Lưu ý: - Nếu bạn đang làm việc cục bộ bằng anaconda, vui lòng bỏ chú thích đoạn mã sau và thực thi nó. Sử dụng phiên bản theo phiên bản python của bạn.

```
!pip install yfinance
!pip install bs4
!pip install nbformat
!pip install --upgrade plotly
```

```
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

```
import plotly.io as pio
pio.renderers.default = "iframe"
```

Trong Python, bạn có thể bỏ qua cảnh báo bằng cách sử dụng mô-đun **warnings**. Bạn có thể sử dụng **hàm filterwarnings** để lọc hoặc bỏ qua các thông báo hoặc danh mục cảnh báo cụ thể.

```
import warnings
# Ignore all warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

1. Định nghĩa hàm đồ thị

Trong phần này, chúng ta định nghĩa hàm `make_graph`. **Bạn không cần biết hàm hoạt động như thế nào, bạn chỉ cần quan tâm đến các đầu vào. Hàm này lấy một khung dữ liệu có dữ liệu chứng khoán (khung dữ liệu phải chứa các cột Date và Close), một khung dữ liệu có dữ liệu doanh thu (khung dữ liệu phải chứa các cột Date và Revenue) và tên của chứng khoán.**

```
1  def make_graph(stock_data, revenue_data, stock):
2      fig = make_subplots(rows=2, cols=1, shared_xaxes=True,
       subplot_titles=("Historical Share Price", "Historical Revenue"),
       vertical_spacing = .3)
3      stock_data_specific = stock_data[stock_data.Date <= '2021-06-14']
       revenue_data_specific = revenue_data[revenue_data.Date <= '2021-04-
       30']
4      fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data_specific.Date,
       infer_datetime_format=True),
       y=stock_data_specific.Close.astype("float"), name="Share Price"), row=1,
       col=1)
5      fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date,
       infer_datetime_format=True),
       y=revenue_data_specific.Revenue.astype("float"), name="Revenue"),
       row=2, col=1)
6      fig.update_xaxes(title_text="Date", row=1, col=1)
7      fig.update_xaxes(title_text="Date", row=2, col=1)
8      fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
9      fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
10     fig.update_layout(showlegend=False,
11     height=900,
12     title=stock,
13     xaxis_rangeslider_visible=True)
14     fig.show()
15     from IPython.display import display, HTML
16     fig_html = fig.to_html()
17     display(HTML(fig_html))
```

Sử dụng hàm make_graph mà chúng ta đã định nghĩa. Bạn sẽ cần gọi hàm này trong câu hỏi 5 và 6 để hiển thị biểu đồ và tạo bảng điều khiển.

Lưu ý: Bạn không cần phải định nghĩa lại hàm để vẽ biểu đồ ở bất kỳ nơi nào khác trong sổ tay này; chỉ cần sử dụng hàm hiện có.

Bài tập

Câu hỏi 1: Sử dụng yfinance để trích xuất dữ liệu chứng khoán

Sử dụng hàm `Ticker` nhập ký hiệu ticker của cổ phiếu mà chúng ta muốn trích xuất dữ liệu để tạo đối tượng ticker. Cổ phiếu là Tesla và ký hiệu ticker của nó là `TSLA`.

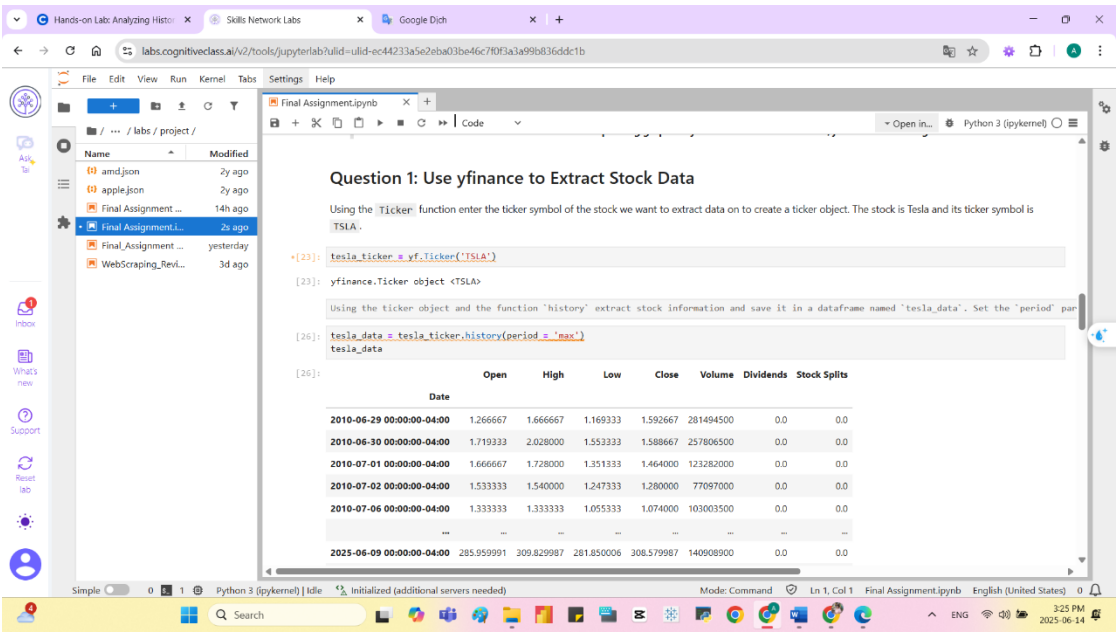
```
1  tesla_ticker = yf.Ticker('TSLA')
```

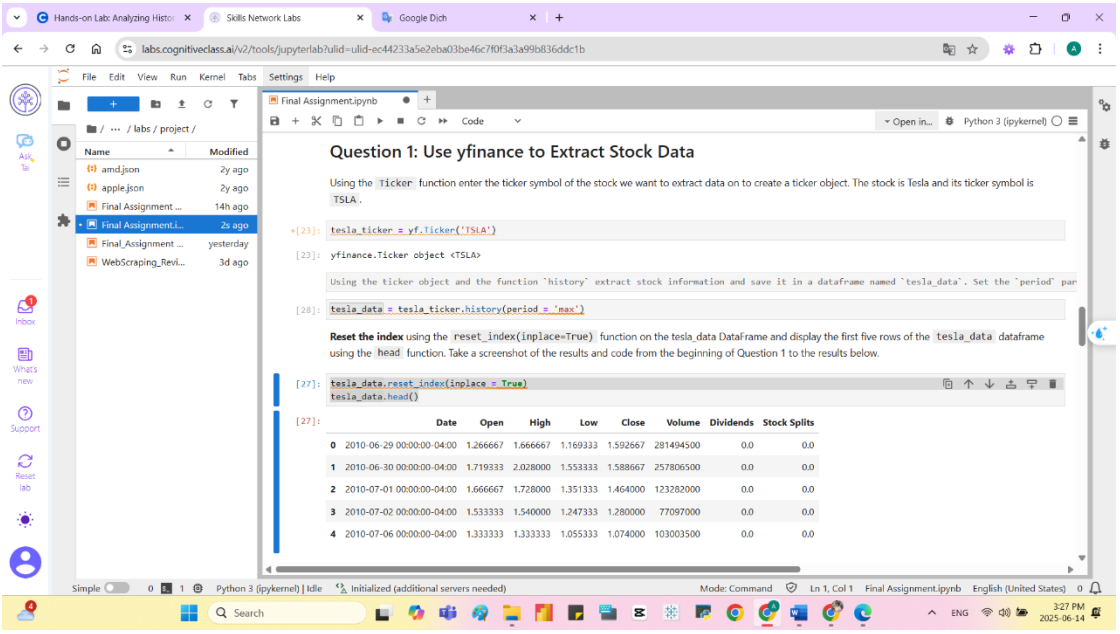
Sử dụng đối tượng ticker và hàm `history` để trích xuất thông tin chứng khoán và lưu vào một khung dữ liệu có tên là `tesla_data`. Đặt tham số `period` thành `"max"` để chúng ta có được thông tin trong khoảng thời gian tối đa.

```
1  tesla_data = tesla_ticker.history(period = 'max')
```

Đặt lại chỉ mục bằng hàm `reset_index(inplace=True)` trên DataFrame `tesla_data` và hiển thị năm hàng đầu tiên của dataframe tesla_data bằng hàm `head` (*phải dùng reset_index vì ban đầu cột Date là chỉ mục*). Chụp ảnh màn hình kết quả và mã từ đầu Câu hỏi 1 đến kết quả bên dưới.

```
1  tesla_data.reset_index(inplace = True)
2  tesla_data.head()
```





Câu hỏi 2: Sử dụng Webscraping để trích xuất dữ liệu doanh thu của Tesla

Sử dụng thư viện `requests` để tải xuống trang web https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm Lưu văn bản phản hồi dưới dạng biến có tên là `html_data`.

```
1  url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm'
2  html_data = requests.get(url).text
```

Phân tích dữ liệu html bằng `beautiful_soup` bằng trình phân tích cú pháp như `html5lib` hoặc `html.parser`.

```
beautiful_soup = BeautifulSoup(html_data, 'html.parser')
```

Sử dụng `BeautifulSoup` hoặc hàm `read_html` để trích xuất bảng có `Tesla Revenue` và lưu trữ vào một khung dữ liệu có tên là `tesla_revenue`. Khung dữ liệu này phải có các cột `Date` và `Revenue`.

Hướng dẫn từng bước

Sau đây là hướng dẫn từng bước:

1. Tạo một DataFrame trống
2. Tìm bảng có liên quan
3. Kiểm tra Bảng doanh thu quý của Tesla
4. Lặp qua các hàng trong phần thân bảng
5. Trích xuất dữ liệu từ các cột
6. Thêm dữ liệu vào DataFrame

Nhấp vào đây nếu bạn cần trợ giúp để định vị bảng

Dưới đây là mã để cô lập bảng, bây giờ bạn sẽ cần lặp qua các hàng và cột như trong phòng thí nghiệm trước

soup.find_all("tbody")[1]

Nếu bạn muốn sử dụng hàm read_html, bảng sẽ nằm ở chỉ mục 1

Chúng tôi đang tập trung vào doanh thu theo quý trong phòng thí nghiệm.

```
tesla_revenue = pd.DataFrame(columns = ['Date','Revenue'])
for row in beautiful_soup.find('tbody').find_all('tr'):
    col = row.find_all('td')
    date = col[0].text
    revenue = col[1].text

```

```
    tesla_revenue = pd.concat([tesla_revenue,pd.DataFrame({'Date':[date],
'Revenue':[revenue]})], ignore_index=True)
```

Thực hiện dòng sau để xóa dấu phẩy và dấu đô la khỏi cột Revenue.

```
tesla_revenue['Revenue'] = tesla_revenue['Revenue'].str.replace(",|\$","",
regex=True)
```

Giải thích:

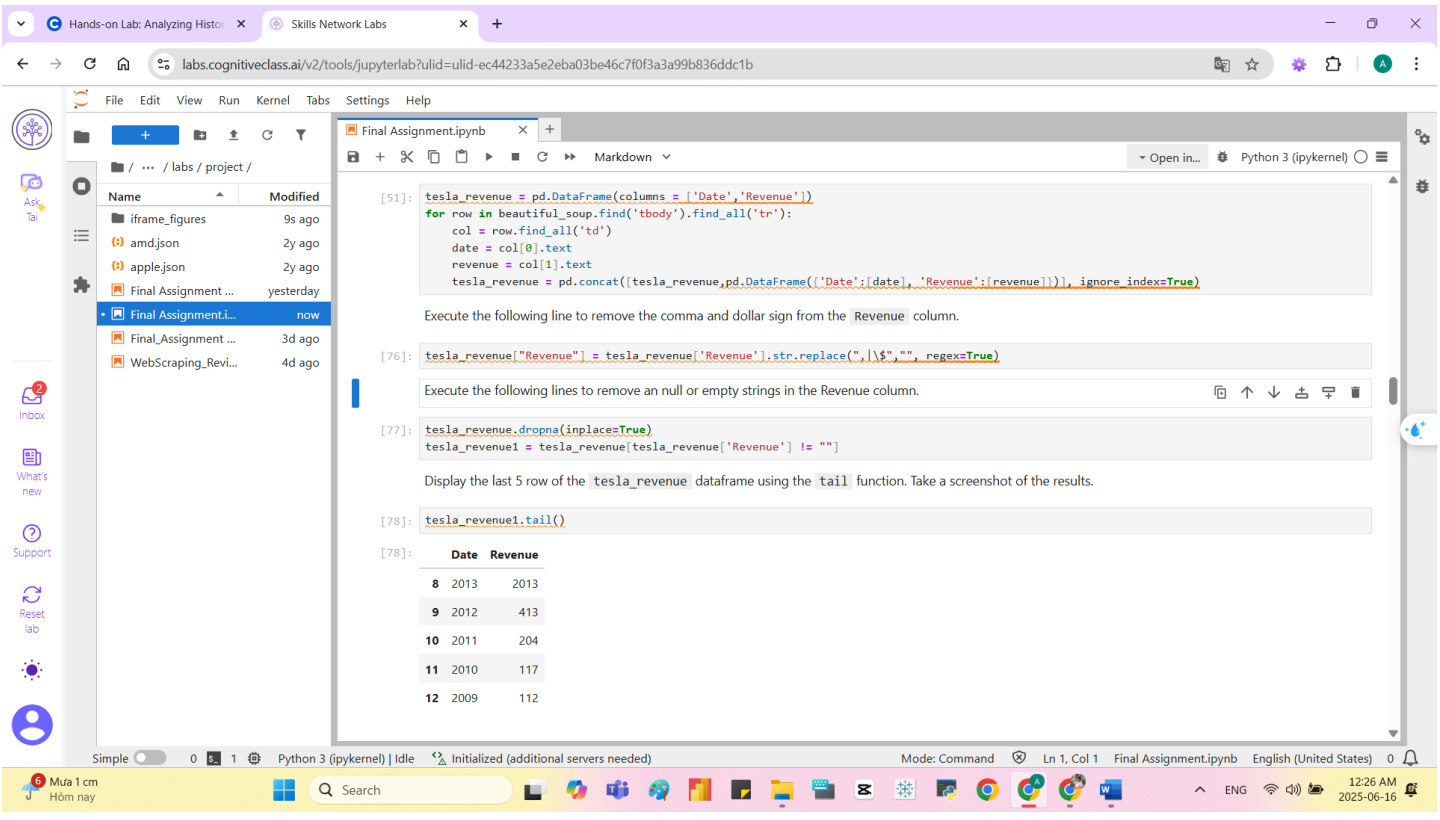
- regex=True: từ **pandas v1.4.0 trở lên**, str.replace() yêu cầu bạn phải ghi rõ regex=True nếu muốn dùng biểu thức chính quy (regular expression).

Thực hiện các dòng sau để xóa chuỗi null hoặc chuỗi rỗng trong cột Revenue.

```
tesla_revenue.dropna(inplace=True) # Xoá các dòng có giá trị NaN
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""] # Xoá dòng
có Revenue là chuỗi rỗng
```

Hiển thị 5 hàng cuối cùng của khung dữ liệu tesla_revenue bằng hàm tail. Chụp ảnh màn hình kết quả.

```
tesla_revenue.tail()
```



Câu hỏi 3: Sử dụng yfinance để trích xuất dữ liệu chứng khoán

Sử dụng hàm `Ticker` nhập ký hiệu ticker của cổ phiếu mà chúng ta muốn trích xuất dữ liệu để tạo đối tượng ticker. Cổ phiếu là GameStop và ký hiệu ticker của nó là `GME`.

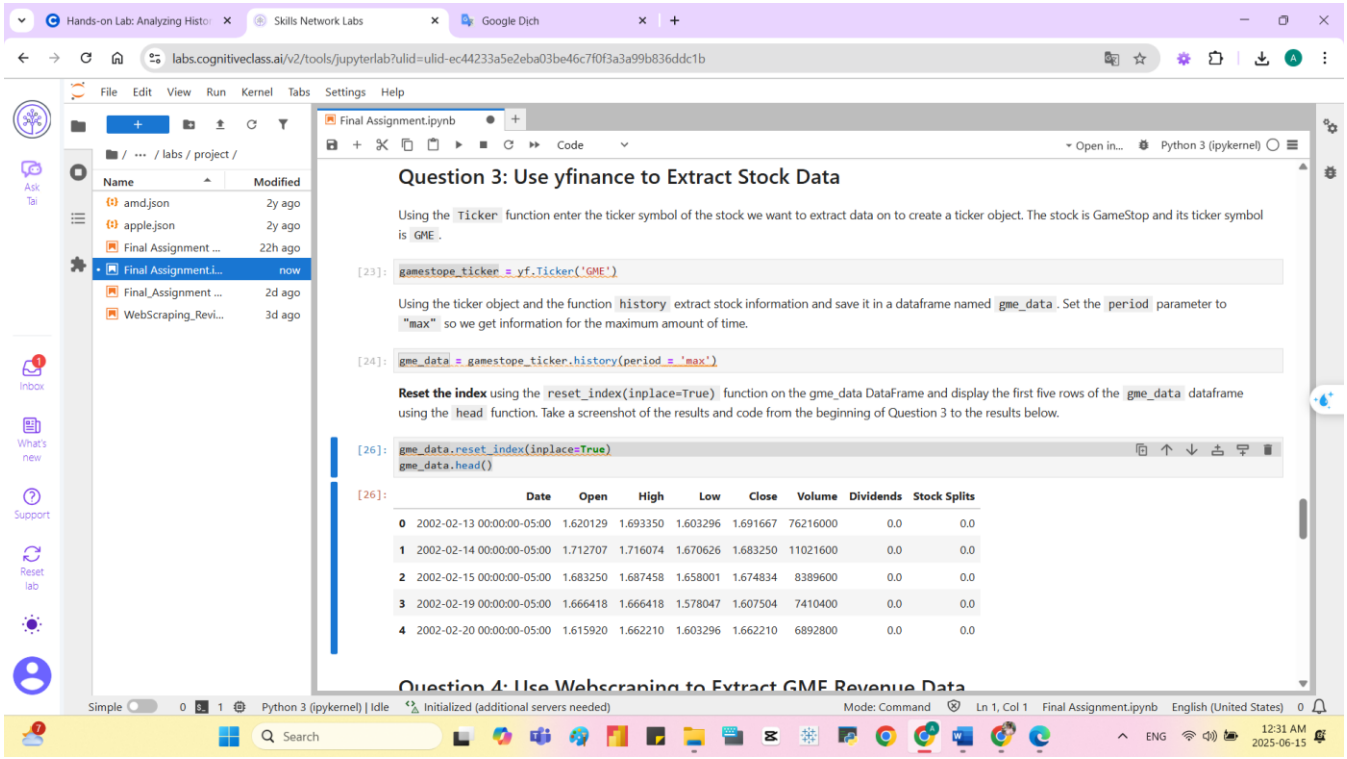
```
1  gamestope_ticker = yf.Ticker('GME')
```

Sử dụng đối tượng ticker và hàm `history` để trích xuất thông tin chứng khoán và lưu vào một khung dữ liệu có tên `gme_data`. Đặt tham số `period` thành `"max"` để chúng ta có được thông tin trong khoảng thời gian tối đa.

```
1  gme_data = gamestope_ticker.history(period = 'max')
```

Đặt lại chỉ mục bằng hàm `reset_index(inplace=True)` trên DataFrame `gme_data` và hiển thị năm hàng đầu tiên của dataframe `gme_data` bằng hàm `head`. Chụp ảnh màn hình kết quả và mã từ đầu Câu hỏi 3 đến kết quả bên dưới.

```
1  gme_data.reset_index(inplace=True)
2  gme_data.head()
```



Câu hỏi 4: Sử dụng Webscraping để trích xuất dữ liệu doanh thu GME

Sử dụng thư viện `requests` để tải xuống trang web https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html. Lưu văn bản phản hồi dưới dạng biến có tên `html_data_2`.

```
1  url1 = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html'
2  html_data_2 = requests.get(url1).text
```

Phân tích dữ liệu html bằng `beautiful_soup` bằng trình phân tích cú pháp như `html5lib` hoặc `html.parser`.

```
1  beautiful_soup2 = BeautifulSoup(html_data_2, 'html.parser')
```

Sử dụng `BeautifulSoup` hoặc hàm `read_html` để trích xuất bảng có `GameStop Revenue` và lưu trữ vào một khung dữ liệu có tên `gme_revenue`. Khung dữ liệu phải có các cột `Date` và `Revenue`. Đảm bảo dấu phẩy và dấu đô la đã được xóa khỏi cột `Revenue`.

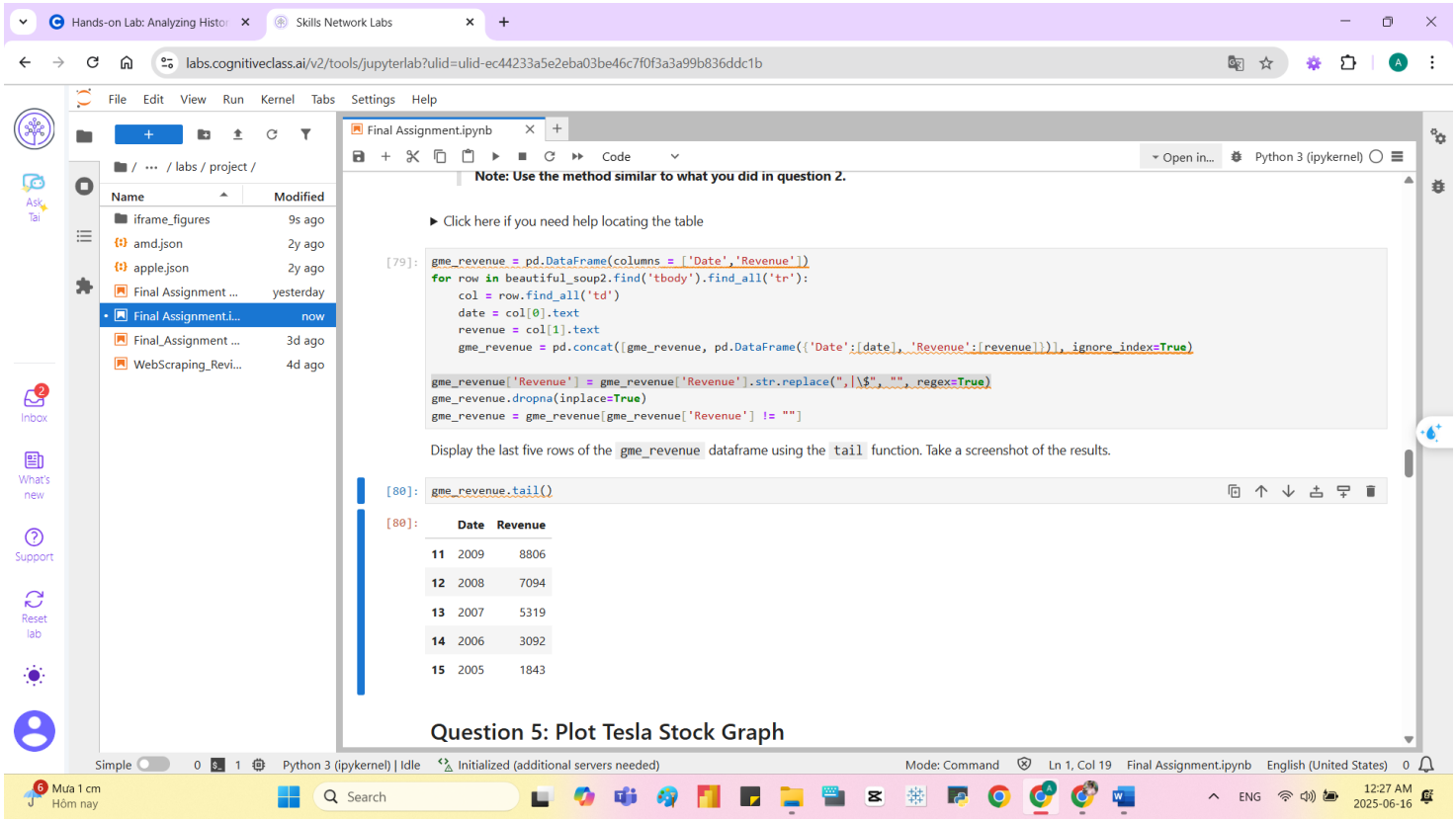
> **Lưu ý: Sử dụng phương pháp tương tự như những gì bạn đã làm ở câu hỏi 2.**

```
1  gme_revenue = pd.DataFrame(columns = ['Date','Revenue'])
2  for row in beautiful_soup2.find('tbody').find_all('tr'):
3      col = row.find_all('td')
4      date = col[0].text
5      revenue = col[1].text
6      gme_revenue = pd.concat([gme_revenue, pd.DataFrame({'Date':[date], 'Revenue':[revenue]})], ignore_index=True)
7
8
9  gme_revenue['Revenue'] = gme_revenue['Revenue'].str.replace(",|\$", "", regex=True)
10 gme_revenue.dropna(inplace=True)
```

```
11  gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
```

Hiển thị năm hàng cuối cùng của khung dữ liệu `gme_revenue` bằng hàm `tail`. Chụp ảnh màn hình kết quả.

```
1  gme_revenue.tail()
```



Câu hỏi 5: Vẽ đồ thị cổ phiếu Tesla

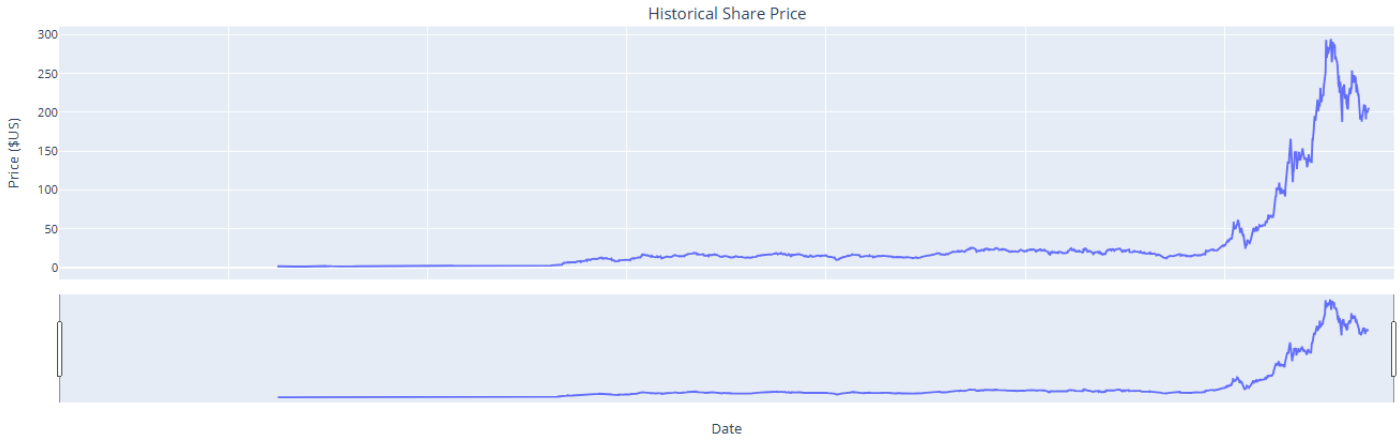
Sử dụng hàm `make_graph` để vẽ đồ thị Dữ liệu cổ phiếu Tesla, đồng thời cung cấp tiêu đề cho đồ thị. Lưu ý đồ thị sẽ chỉ hiển thị dữ liệu đến tháng 6 năm 2021.

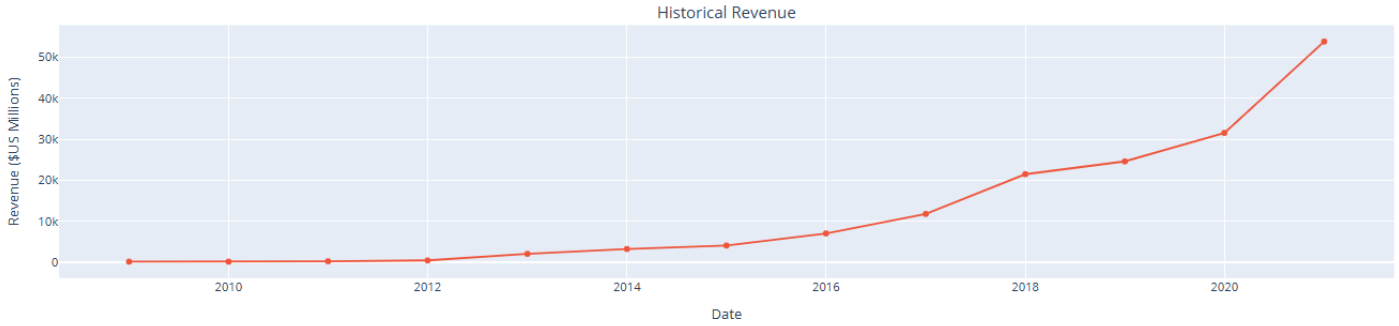
Gợi ý

Bạn chỉ cần gọi hàm make_graph với tham số cần thiết để in biểu đồ. Cấu trúc để gọi hàm `make_graph` là `make_graph(tesla_data, tesla_revenue, 'Tesla')`.

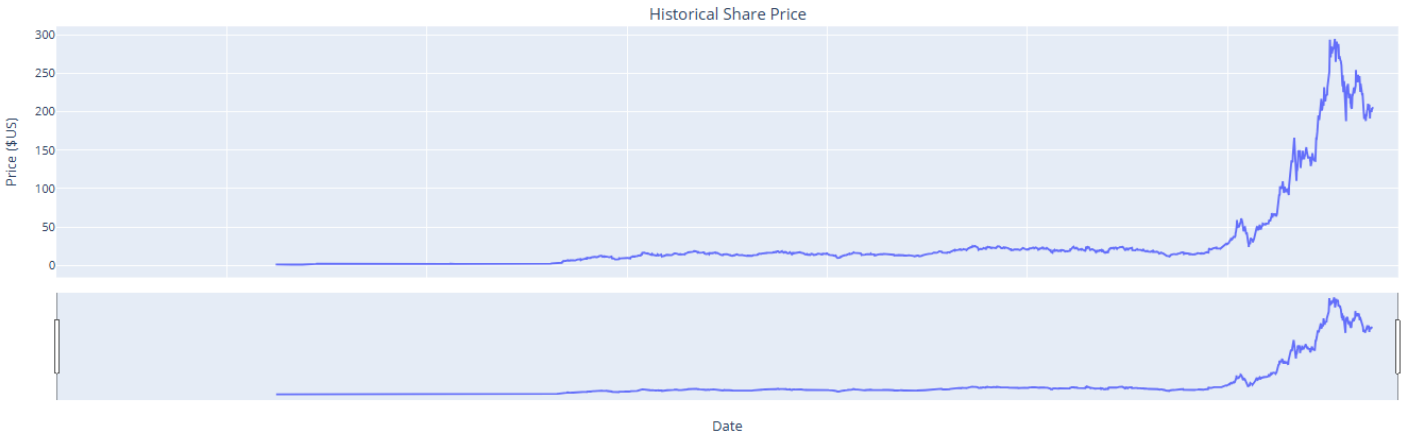
```
1  make_graph(tesla_data, tesla_revenue, 'Tesla')
```

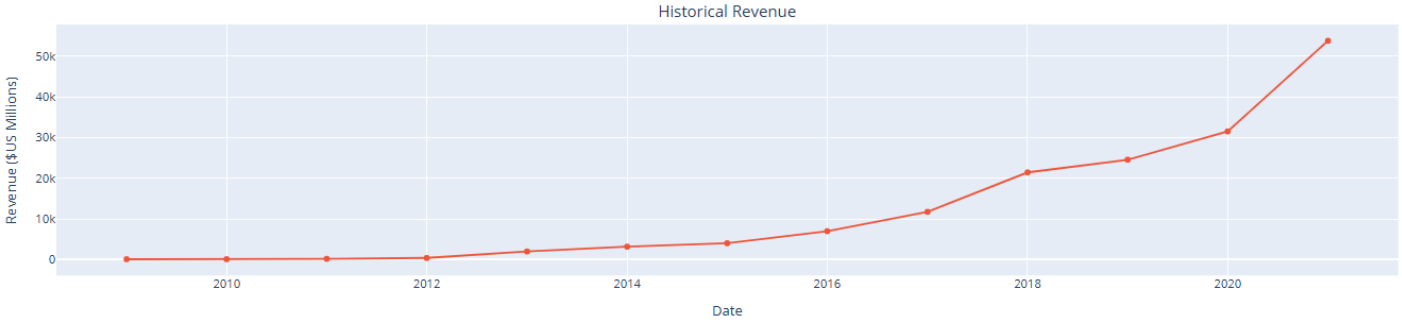
Tesla


Historical Share Price
Price ($US)
300
250
200
150
100
50
0
Date


Historical Revenue
Revenue ($US Millions)
50k
40k
30k
20k
10k
0
2010
2012
2014
2016
2018
2020
Date

Tesla


Historical Share Price
Price ($US)
300
250
200
150
100
50
0
Date


Historical Revenue
Revenue ($US Millions)
50k
40k
30k
20k
10k
0
2010
2012
2014
2016
2018
2020
Date

Câu hỏi 6: Vẽ đồ thị cổ phiếu GameStop

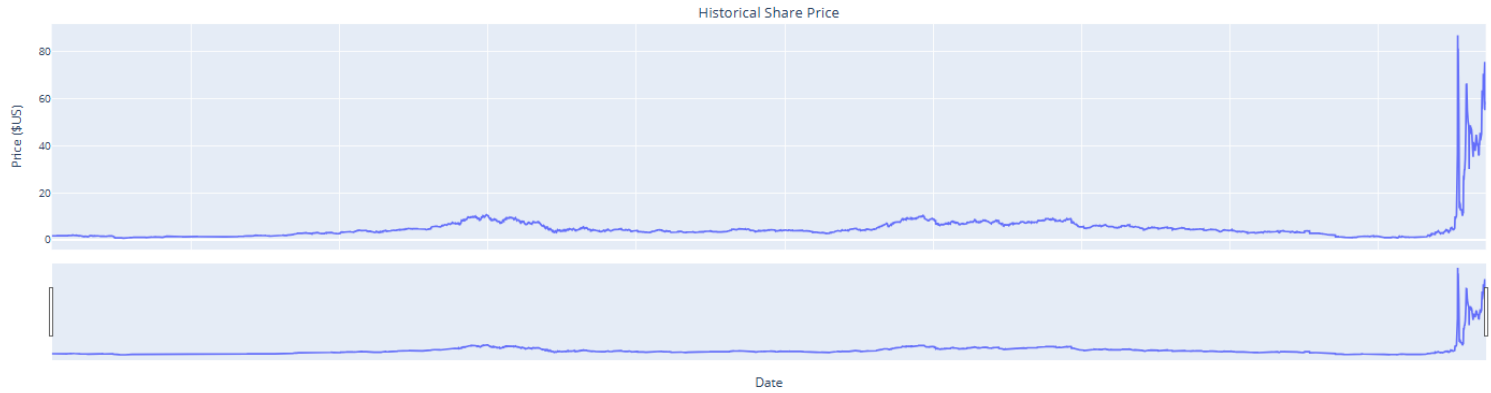
Sử dụng hàm make_graph để vẽ đồ thị Dữ liệu cổ phiếu GameStop, đồng thời cung cấp tiêu đề cho đồ thị. Cấu trúc để gọi hàm make_graph là make_graph(gme_data, gme_revenue, 'GameStop'). Lưu ý rằng đồ thị sẽ chỉ hiển thị dữ liệu đến tháng 6 năm 2021.

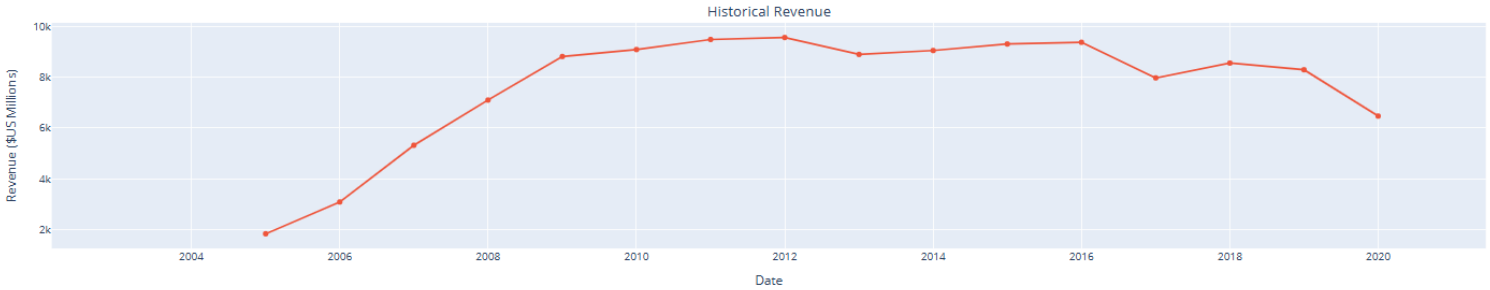
Gợi ý

Bạn chỉ cần gọi hàm make_graph với tham số bắt buộc để in biểu đồ. Cấu trúc để gọi hàm `make_graph` là `make_graph(gme_data, gme_revenue, 'GameStop')`

```
1  make_graph(gme_data, gme_revenue, 'GameStop')
```

GameStop


Historical Share Price
Price ($US)
80
60
40
20
0
Date


Historical Revenue
Revenue ($US Millions)
10k
8k
6k
4k
2k
2004
2006
2008
2010
2012
2014
2016
2018
2020
Date