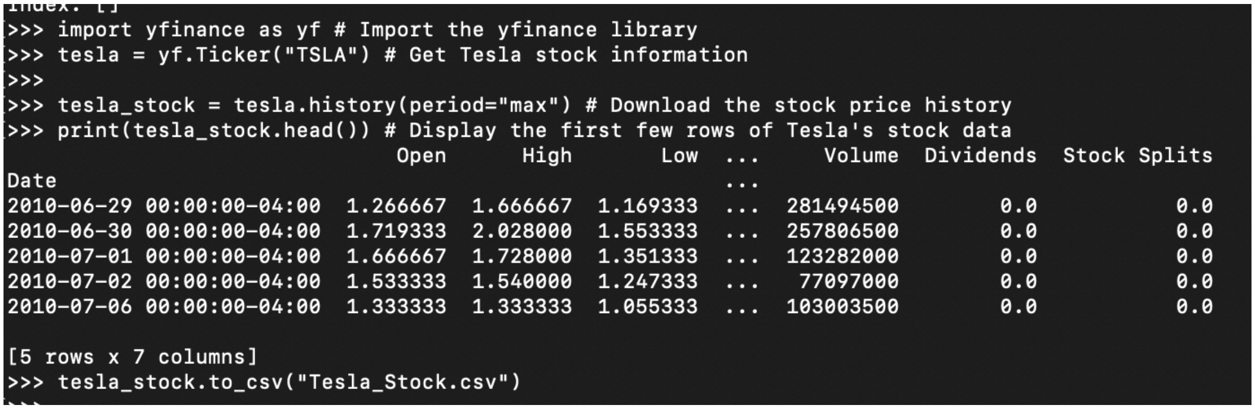
Peer-graded Assignment: Analyzing Historical Stock/Revenue Data and Building a Dashboard

Question 1 - Extracting Tesla Stock Data Using yfinance - 2 Points

I used the Terminal window on my Mac to execute the Python commands for extracting Tesla stock data using the yfinance library. The screenshot shows the commands I ran and the first few rows of the Tesla stock data retrieved using tesla_stock.head().

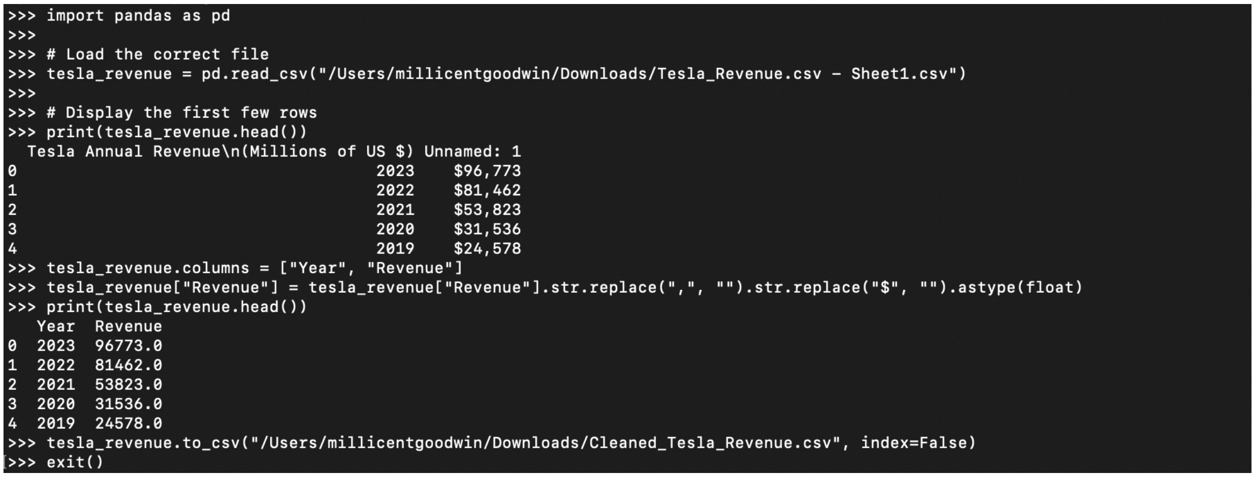


```
Index: []
>>> import yfinance as yf # Import the yfinance library
>>> tesla = yf.Ticker("TSLA") # Get Tesla stock information
>>>
>>> tesla_stock = tesla.history(period="max") # Download the stock price history
>>> print(tesla_stock.head()) # Display the first few rows of Tesla's stock data
                               Open      High       Low  ...     Volume  Dividends  Stock Splits
Date                                                     ...
2010-06-29 00:00:00-04:00  1.266667  1.666667  1.169333  ...  281494500        0.0           0.0
2010-06-30 00:00:00-04:00  1.719333  2.028000  1.553333  ...  257806500        0.0           0.0
2010-07-01 00:00:00-04:00  1.666667  1.728000  1.351333  ...  123282000        0.0           0.0
2010-07-02 00:00:00-04:00  1.533333  1.540000  1.247333  ...   77097000        0.0           0.0
2010-07-06 00:00:00-04:00  1.333333  1.333333  1.055333  ...  103003500        0.0           0.0

[5 rows x 7 columns]
>>> tesla_stock.to_csv("Tesla_Stock.csv")
>>>
```

Question 2 - Extracting Tesla Revenue Data Using Webscraping - 1 Points

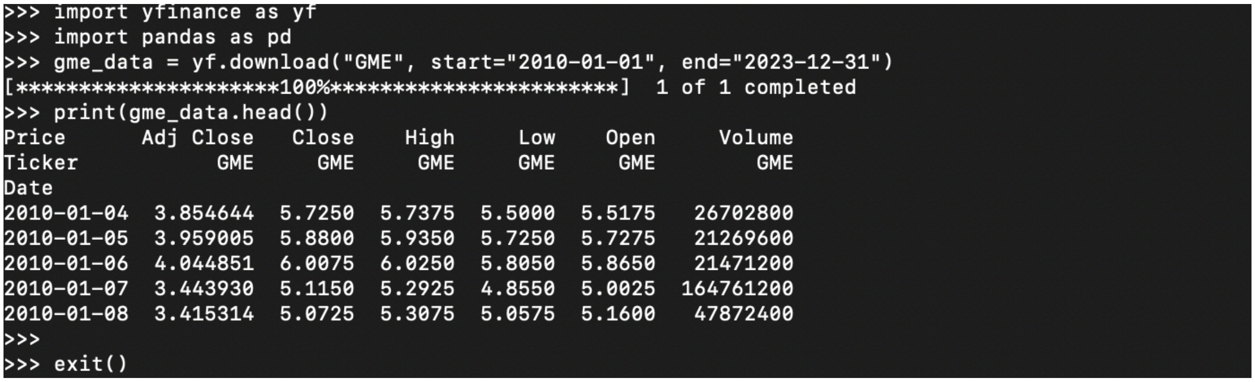
I extracted Tesla's revenue data from Macrotrends. Due to the dynamic nature of the website, I manually downloaded the table as a CSV file, then loaded it into Python for cleaning and analysis. Below is the cleaned data.



```
>>> import pandas as pd
>>>
>>> # Load the correct file
>>> tesla_revenue = pd.read_csv("/Users/millicentgoodwin/Downloads/Tesla_Revenue.csv - Sheet1.csv")
>>>
>>> # Display the first few rows
>>> print(tesla_revenue.head())
  Tesla Annual Revenue\n(Millions of US $) Unnamed: 1
0                                     2023    $96,773
1                                     2022    $81,462
2                                     2021    $53,823
3                                     2020    $31,536
4                                     2019    $24,578
>>> tesla_revenue.columns = ["Year", "Revenue"]
>>> tesla_revenue["Revenue"] = tesla_revenue["Revenue"].str.replace(",", "").str.replace("$", "").astype(float)
>>> print(tesla_revenue.head())
   Year  Revenue
0  2023  96773.0
1  2022  81462.0
2  2021  53823.0
3  2020  31536.0
4  2019  24578.0
>>> tesla_revenue.to_csv("/Users/millicentgoodwin/Downloads/Cleaned_Tesla_Revenue.csv", index=False)
>>> exit()
```

Question 3 - Extracting GameStop Stock Data Using yfinance - 2 Points

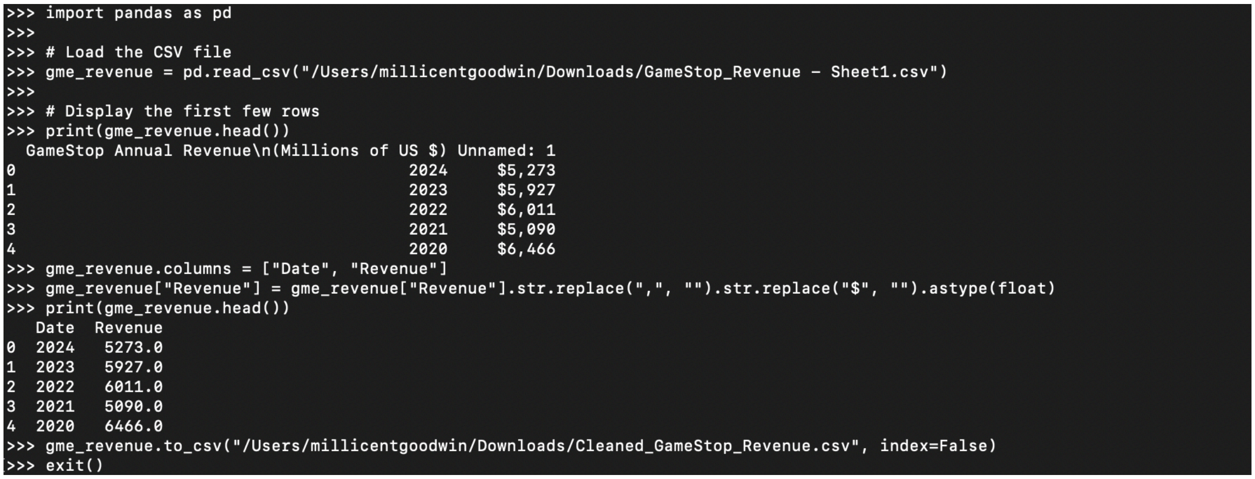
I used the `yfinance` library to download historical GameStop (GME) stock data from 2010 to 2023. Below is a sample of the extracted data.



```
>>> import yfinance as yf
>>> import pandas as pd
>>> gme_data = yf.download("GME", start="2010-01-01", end="2023-12-31")
[*********************100%***********************]  1 of 1 completed
>>> print(gme_data.head())
Price      Adj Close   Close    High     Low    Open     Volume
Ticker           GME     GME     GME     GME     GME        GME
Date
2010-01-04  3.854644  5.7250  5.7375  5.5000  5.5175   26702800
2010-01-05  3.959005  5.8800  5.9350  5.7250  5.7275   21269600
2010-01-06  4.044851  6.0075  6.0250  5.8050  5.8650   21471200
2010-01-07  3.443930  5.1150  5.2925  4.8550  5.0025  164761200
2010-01-08  3.415314  5.0725  5.3075  5.0575  5.1600   47872400
>>>
>>> exit()
```

Question 4 - Extracting GameStop Revenue Data Using Webscraping - 1 Points

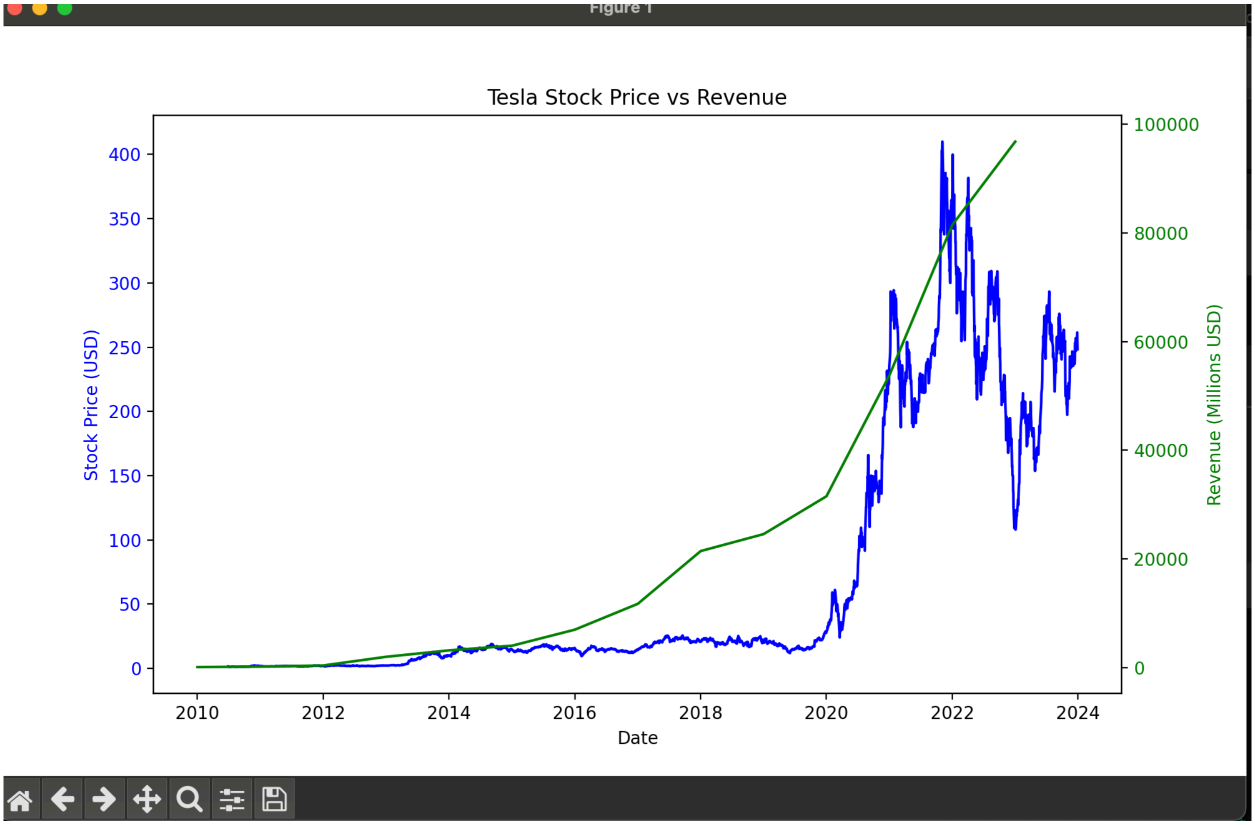
I manually downloaded GameStop’s annual revenue data from Macrotrends, saved it as a CSV, and loaded it into Python for cleaning and analysis. Below is the cleaned data.



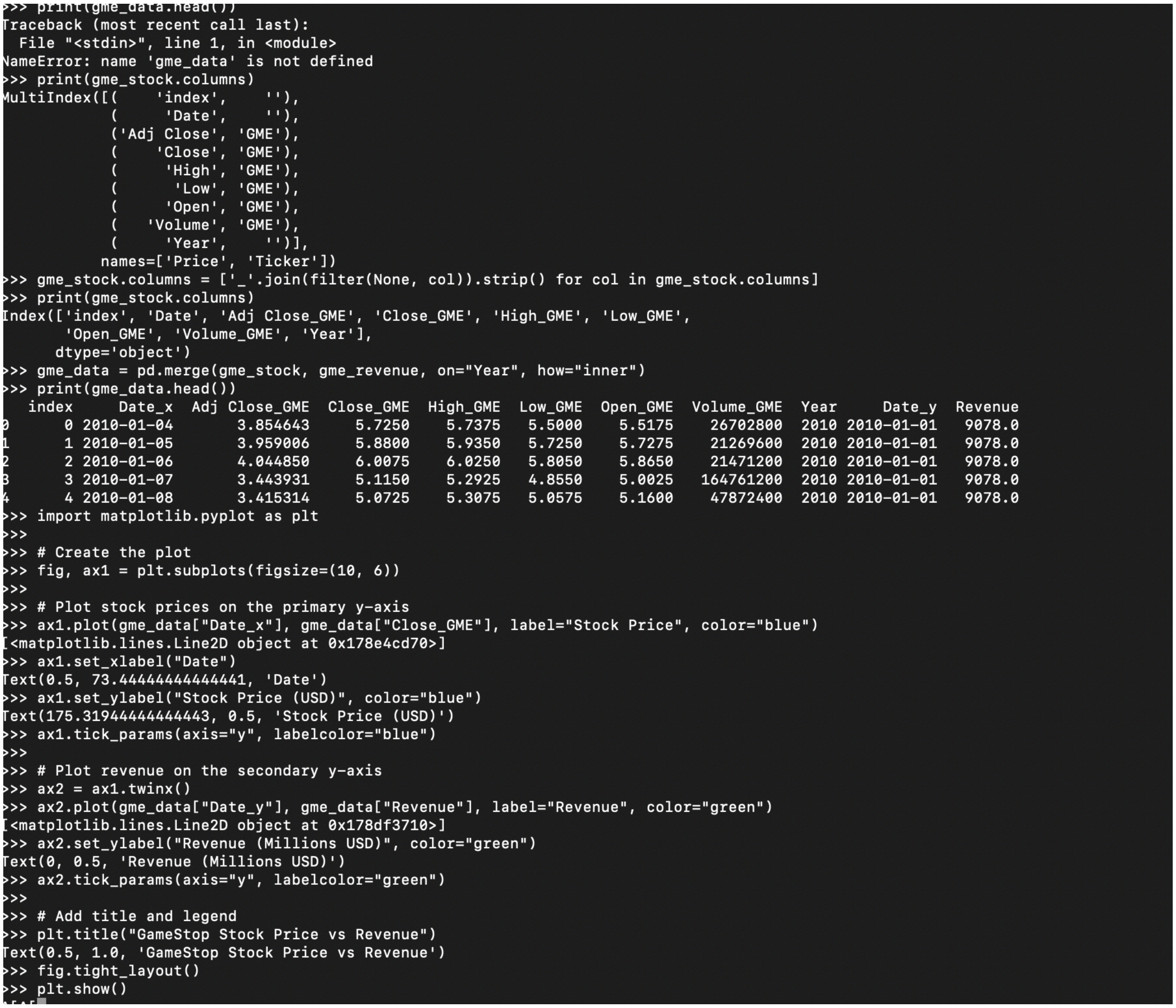
```
>>> import pandas as pd
>>>
>>> # Load the CSV file
>>> gme_revenue = pd.read_csv("/Users/millicentgoodwin/Downloads/GameStop_Revenue - Sheet1.csv")
>>>
>>> # Display the first few rows
>>> print(gme_revenue.head())
  GameStop Annual Revenue\n(Millions of US $) Unnamed: 1
0                                        2024     $5,273
1                                        2023     $5,927
2                                        2022     $6,011
3                                        2021     $5,090
4                                        2020     $6,466
>>> gme_revenue.columns = ["Date", "Revenue"]
>>> gme_revenue["Revenue"] = gme_revenue["Revenue"].str.replace(",", "").str.replace("$", "").astype(float)
>>> print(gme_revenue.head())
   Date  Revenue
0  2024   5273.0
1  2023   5927.0
2  2022   6011.0
3  2021   5090.0
4  2020   6466.0
>>> gme_revenue.to_csv("/Users/millicentgoodwin/Downloads/Cleaned_GameStop_Revenue.csv", index=False)
>>> exit()
```

Question 5 - Tesla Stock and Revenue Dashboard - 2 Points

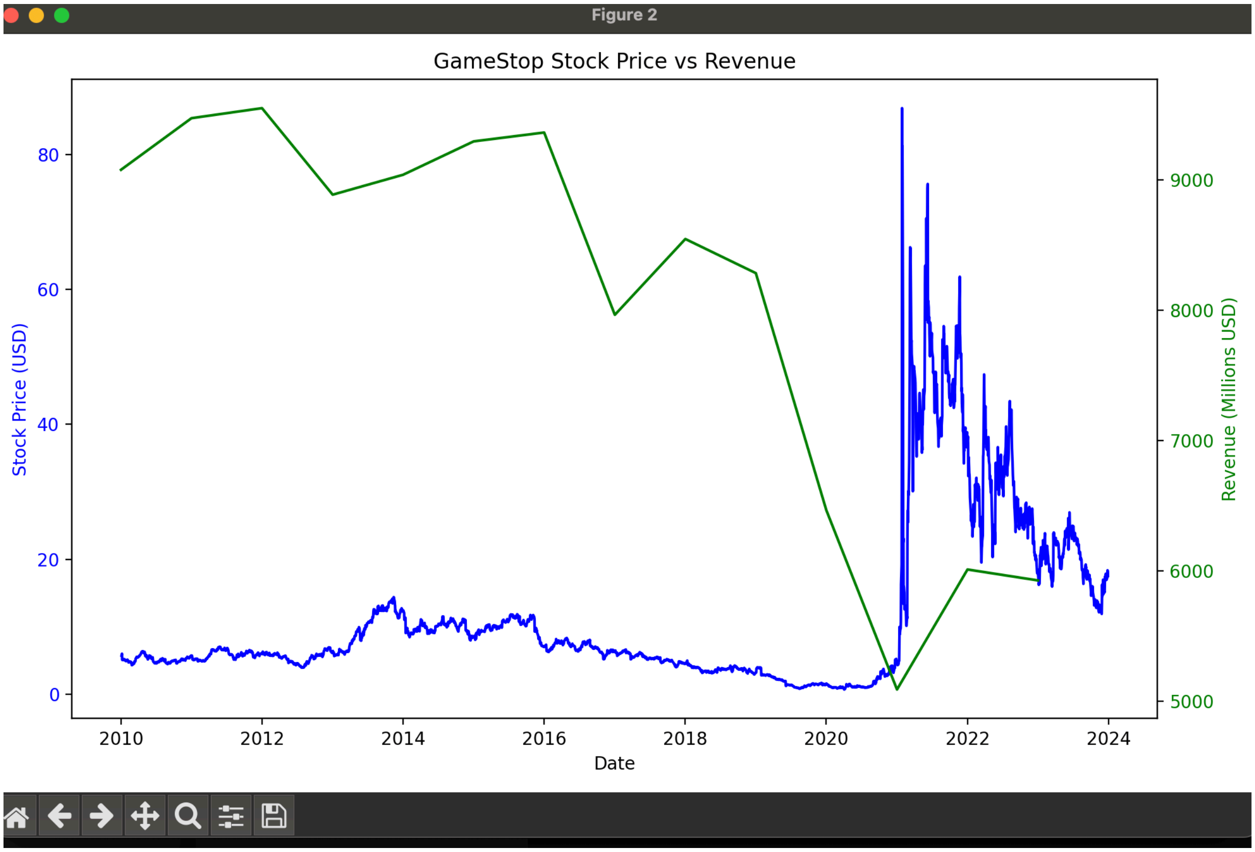
The plot highlights a strong correlation between Tesla’s revenue growth and its stock price increases, especially after 2020. As revenue surged, reflecting Tesla's expansion and market success, investor confidence grew, driving up stock prices. Before 2020, both revenue and stock prices were stable, but post-2020, Tesla's stock became more volatile while continuing to follow the upward revenue trend. This demonstrates Tesla’s financial growth directly influencing its market valuation.



Question 6 - GameStop Stock and Revenue Dashboard- 2 Points
The chart shows GameStop's stock price (blue) and revenue (green) from 2010 to 2024. Stock price surged in 2021 due to market activity, while revenue steadily declined after 2017, reflecting challenges in GameStop's business model.



```
>>> print(gme_data.head())
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'gme_data' is not defined
>>> print(gme_stock.columns)
MultiIndex([(    'index',    ''),
            (     'Date',    ''),
            ('Adj Close', 'GME'),
            (    'Close', 'GME'),
            (     'High', 'GME'),
            (      'Low', 'GME'),
            (     'Open', 'GME'),
            (   'Volume', 'GME'),
            (     'Year',    '')],
           names=['Price', 'Ticker'])
>>> gme_stock.columns = ['_'.join(filter(None, col)).strip() for col in gme_stock.columns]
>>> print(gme_stock.columns)
Index(['index', 'Date', 'Adj Close_GME', 'Close_GME', 'High_GME', 'Low_GME',
       'Open_GME', 'Volume_GME', 'Year'],
      dtype='object')
>>> gme_data = pd.merge(gme_stock, gme_revenue, on="Year", how="inner")
>>> print(gme_data.head())
   index     Date_x  Adj Close_GME  Close_GME  High_GME  Low_GME  Open_GME  Volume_GME  Year     Date_y  Revenue
0      0 2010-01-04       3.854643     5.7250    5.7375   5.5000    5.5175    26702800  2010 2010-01-01   9078.0
1      1 2010-01-05       3.959006     5.8800    5.9350   5.7250    5.7275    21269600  2010 2010-01-01   9078.0
2      2 2010-01-06       4.044850     6.0075    6.0250   5.8050    5.8650    21471200  2010 2010-01-01   9078.0
3      3 2010-01-07       3.443931     5.1150    5.2925   4.8550    5.0025   164761200  2010 2010-01-01   9078.0
4      4 2010-01-08       3.415314     5.0725    5.3075   5.0575    5.1600    47872400  2010 2010-01-01   9078.0
>>> import matplotlib.pyplot as plt
>>>
>>> # Create the plot
>>> fig, ax1 = plt.subplots(figsize=(10, 6))
>>>
>>> # Plot stock prices on the primary y-axis
>>> ax1.plot(gme_data["Date_x"], gme_data["Close_GME"], label="Stock Price", color="blue")
[<matplotlib.lines.Line2D object at 0x178e4cd70>]
>>> ax1.set_xlabel("Date")
Text(0.5, 73.44444444444441, 'Date')
>>> ax1.set_ylabel("Stock Price (USD)", color="blue")
Text(175.31944444444443, 0.5, 'Stock Price (USD)')
>>> ax1.tick_params(axis="y", labelcolor="blue")
>>>
>>> # Plot revenue on the secondary y-axis
>>> ax2 = ax1.twinx()
>>> ax2.plot(gme_data["Date_y"], gme_data["Revenue"], label="Revenue", color="green")
[<matplotlib.lines.Line2D object at 0x178df3710>]
>>> ax2.set_ylabel("Revenue (Millions USD)", color="green")
Text(0, 0.5, 'Revenue (Millions USD)')
>>> ax2.tick_params(axis="y", labelcolor="green")
>>>
>>> # Add title and legend
>>> plt.title("GameStop Stock Price vs Revenue")
Text(0.5, 1.0, 'GameStop Stock Price vs Revenue')
>>> fig.tight_layout()
>>> plt.show()
^[^[
```



Question 7 - Sharing your Assignment Notebook - 2 Points