

# Final Assignment

October 3, 2025

## Extracting and Visualizing Stock Data

### Description

Extracting essential data from a dataset and displaying it is a necessary part of data science; therefore individuals can make correct decisions based on the data. In this assignment, you will extract some stock data, you will then display this data in a graph.

### Table of Contents

- <li>Define a Function that Makes a Graph</li>
- <li>Question 1: Use yfinance to Extract Stock Data</li>
- <li>Question 2: Use Webscraping to Extract Tesla Revenue Data</li>
- <li>Question 3: Use yfinance to Extract Stock Data</li>
- <li>Question 4: Use Webscraping to Extract GME Revenue Data</li>
- <li>Question 5: Plot Tesla Stock Graph</li>
- <li>Question 6: Plot GameStop Stock Graph</li>

Estimated Time Needed: 30 min

**Note:-** If you are working Locally using anaconda, please uncomment the following code and execute it. Use the version as per your python version.

```
[1]: !pip install yfinance
      !pip install bs4
      !pip install nbformat
      !pip install --upgrade plotly
```

Requirement already satisfied: yfinance in /opt/conda/lib/python3.12/site-packages (0.2.66)

Requirement already satisfied: pandas>=1.3.0 in /opt/conda/lib/python3.12/site-packages (from yfinance) (2.3.3)

Requirement already satisfied: numpy>=1.16.5 in /opt/conda/lib/python3.12/site-packages (from yfinance) (2.3.3)

Requirement already satisfied: requests>=2.31 in /opt/conda/lib/python3.12/site-packages (from yfinance) (2.32.3)

Requirement already satisfied: multitasking>=0.0.7 in /opt/conda/lib/python3.12/site-packages (from yfinance) (0.0.12)

Requirement already satisfied: platformdirs>=2.0.0 in /opt/conda/lib/python3.12/site-packages (from yfinance) (4.3.6)

Requirement already satisfied: pytz>=2022.5 in /opt/conda/lib/python3.12/site-packages (from yfinance) (2024.2)

Requirement already satisfied: frozendict>=2.3.4 in /opt/conda/lib/python3.12/site-packages (from yfinance) (2.4.6)

Requirement already satisfied: peewee>=3.16.2 in /opt/conda/lib/python3.12/site-packages (from yfinance) (3.18.2)

Requirement already satisfied: beautifulsoup4>=4.11.1 in /opt/conda/lib/python3.12/site-packages (from yfinance) (4.12.3)

Requirement already satisfied: curl\_cffi>=0.7 in /opt/conda/lib/python3.12/site-packages (from yfinance) (0.13.0)

Requirement already satisfied: protobuf>=3.19.0 in /opt/conda/lib/python3.12/site-packages (from yfinance) (6.32.1)

Requirement already satisfied: websockets>=13.0 in /opt/conda/lib/python3.12/site-packages (from yfinance) (15.0.1)

Requirement already satisfied: soupsieve>1.2 in /opt/conda/lib/python3.12/site-packages (from beautifulsoup4>=4.11.1->yfinance) (2.5)

Requirement already satisfied: cffi>=1.12.0 in /opt/conda/lib/python3.12/site-packages (from curl\_cffi>=0.7->yfinance) (1.17.1)

Requirement already satisfied: certifi>=2024.2.2 in /opt/conda/lib/python3.12/site-packages (from curl\_cffi>=0.7->yfinance) (2024.12.14)

Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.12/site-packages (from pandas>=1.3.0->yfinance) (2.9.0.post0)

Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.12/site-packages (from pandas>=1.3.0->yfinance) (2025.2)

Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.12/site-packages (from requests>=2.31->yfinance) (3.4.1)

Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.12/site-packages (from requests>=2.31->yfinance) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.12/site-packages (from requests>=2.31->yfinance) (2.3.0)

Requirement already satisfied: pycparser in /opt/conda/lib/python3.12/site-packages (from cffi>=1.12.0->curl\_cffi>=0.7->yfinance) (2.22)

Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-packages (from python-dateutil>=2.8.2->pandas>=1.3.0->yfinance) (1.17.0)

Requirement already satisfied: bs4 in /opt/conda/lib/python3.12/site-packages (0.0.2)

Requirement already satisfied: beautifulsoup4 in /opt/conda/lib/python3.12/site-packages (from bs4) (4.12.3)

Requirement already satisfied: soupsieve>1.2 in /opt/conda/lib/python3.12/site-packages (from beautifulsoup4->bs4) (2.5)

Requirement already satisfied: nbformat in /opt/conda/lib/python3.12/site-packages (5.10.4)

Requirement already satisfied: fastjsonschema>=2.15 in /opt/conda/lib/python3.12/site-packages (from nbformat) (2.21.1)

Requirement already satisfied: jsonschema>=2.6 in /opt/conda/lib/python3.12/site-packages (from nbformat) (4.23.0)

Requirement already satisfied: jupyter-core!=5.0.\*,>=4.12 in /opt/conda/lib/python3.12/site-packages (from nbformat) (5.7.2)

Requirement already satisfied: traitlets>=5.1 in /opt/conda/lib/python3.12/site-packages (from nbformat) (5.14.3)

Requirement already satisfied: attrs>=22.2.0 in /opt/conda/lib/python3.12/site-packages (from jsonschema>=2.6->nbformat) (25.1.0)

Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /opt/conda/lib/python3.12/site-packages (from jsonschema>=2.6->nbformat) (2024.10.1)

Requirement already satisfied: referencing>=0.28.4 in /opt/conda/lib/python3.12/site-packages (from jsonschema>=2.6->nbformat) (0.36.2)

Requirement already satisfied: rpds-py>=0.7.1 in /opt/conda/lib/python3.12/site-packages (from jsonschema>=2.6->nbformat) (0.22.3)

Requirement already satisfied: platformdirs>=2.5 in /opt/conda/lib/python3.12/site-packages (from jupyter-core!=5.0.\*,>=4.12->nbformat) (4.3.6)

Requirement already satisfied: typing-extensions>=4.4.0 in /opt/conda/lib/python3.12/site-packages (from referencing>=0.28.4->jsonschema>=2.6->nbformat) (4.12.2)

Requirement already satisfied: plotly in /opt/conda/lib/python3.12/site-packages (6.3.1)

Requirement already satisfied: narwhals>=1.15.1 in /opt/conda/lib/python3.12/site-packages (from plotly) (2.6.0)

Requirement already satisfied: packaging in /opt/conda/lib/python3.12/site-packages (from plotly) (24.2)

```
[2]: 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
```

```
[3]: import plotly.io as pio
pio.renderers.default = "iframe"
```

In Python, you can ignore warnings using the warnings module. You can use the filterwarnings function to filter or ignore specific warning messages or categories.

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

## 0.1 Define Graphing Function

In this section, we define the function `make_graph`. You don't have to know how the function works, you should only care about the inputs. It takes a dataframe with stock data

(dataframe must contain Date and Close columns), a dataframe with revenue data (dataframe must contain Date and Revenue columns), and the name of the stock.

```
[5]: def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True,
    ↪subplot_titles=("Historical Share Price", "Historical Revenue"),
    ↪vertical_spacing = .3)
    stock_data_specific = stock_data[stock_data.Date <= '2021-06-14']
    revenue_data_specific = revenue_data[revenue_data.Date <= '2021-04-30']
    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)
    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)
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
    height=900,
    title=stock,
    xaxis_rangeflider_visible=True)
    fig.show()
    from IPython.display import display, HTML
    fig_html = fig.to_html()
    display(HTML(fig_html))
```

Use the `make_graph` function that we've already defined. You'll need to invoke it in questions 5 and 6 to display the graphs and create the dashboard. > **Note: You don't need to redefine the function for plotting graphs anywhere else in this notebook; just use the existing function.**

## 0.2 Question 1: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is Tesla and its ticker symbol is `TSLA`.

```
[6]: tesla = yf.Ticker('TSLA')
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `tesla_data`. Set the `period` parameter to `"max"` so we get information for the maximum amount of time.

```
[7]: tesla_data = pd.DataFrame(tesla.history(period="max"))
```

```
[8]: print(tesla_data.shape)
print(tesla_data.head)
```

```
(3841, 7)
<bound method NDFrame.head of
Low      Close \      Open      High
Date
2010-06-29 00:00:00-04:00    1.266667    1.666667    1.169333    1.592667
2010-06-30 00:00:00-04:00    1.719333    2.028000    1.553333    1.588667
2010-07-01 00:00:00-04:00    1.666667    1.728000    1.351333    1.464000
2010-07-02 00:00:00-04:00    1.533333    1.540000    1.247333    1.280000
2010-07-06 00:00:00-04:00    1.333333    1.333333    1.055333    1.074000
...
2025-09-29 00:00:00-04:00  444.350006  450.980011  439.500000  443.209991
2025-09-30 00:00:00-04:00  441.519989  445.000000  433.119995  444.720001
2025-10-01 00:00:00-04:00  443.799988  462.290009  440.750000  459.459991
2025-10-02 00:00:00-04:00  470.540009  470.750000  435.570007  436.000000
2025-10-03 00:00:00-04:00  443.285004  446.769989  416.575012  429.829987
```

```

Volume  Dividends  Stock Splits
Date
2010-06-29 00:00:00-04:00  281494500      0.0      0.0
2010-06-30 00:00:00-04:00  257806500      0.0      0.0
2010-07-01 00:00:00-04:00  123282000      0.0      0.0
2010-07-02 00:00:00-04:00   77097000      0.0      0.0
2010-07-06 00:00:00-04:00  103003500      0.0      0.0
...
2025-09-29 00:00:00-04:00   79491500      0.0      0.0
2025-09-30 00:00:00-04:00   74358000      0.0      0.0
2025-10-01 00:00:00-04:00   98122300      0.0      0.0
2025-10-02 00:00:00-04:00  137009000      0.0      0.0
2025-10-03 00:00:00-04:00  132426362      0.0      0.0
```

```
[3841 rows x 7 columns]>
```

**Reset the index** using the `reset_index(inplace=True)` function on the `tesla_data` DataFrame and display the first five rows of the `tesla_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 1 to the results below.

```
[9]: tesla_data.reset_index(inplace=True)
tesla_data.head
```

```
[9]: <bound method NDFrame.head of
High      Low \      Date      Open
0  2010-06-29 00:00:00-04:00    1.266667    1.666667    1.169333
1  2010-06-30 00:00:00-04:00    1.719333    2.028000    1.553333
2  2010-07-01 00:00:00-04:00    1.666667    1.728000    1.351333
3  2010-07-02 00:00:00-04:00    1.533333    1.540000    1.247333
4  2010-07-06 00:00:00-04:00    1.333333    1.333333    1.055333
...
3836 2025-09-29 00:00:00-04:00  444.350006  450.980011  439.500000
```

```

3837 2025-09-30 00:00:00-04:00 441.519989 445.000000 433.119995
3838 2025-10-01 00:00:00-04:00 443.799988 462.290009 440.750000
3839 2025-10-02 00:00:00-04:00 470.540009 470.750000 435.570007
3840 2025-10-03 00:00:00-04:00 443.285004 446.769989 416.575012

```

	Close	Volume	Dividends	Stock Splits
0	1.592667	281494500	0.0	0.0
1	1.588667	257806500	0.0	0.0
2	1.464000	123282000	0.0	0.0
3	1.280000	77097000	0.0	0.0
4	1.074000	103003500	0.0	0.0
...	...	...	...	...
3836	443.209991	79491500	0.0	0.0
3837	444.720001	74358000	0.0	0.0
3838	459.459991	98122300	0.0	0.0
3839	436.000000	137009000	0.0	0.0
3840	429.829987	132426362	0.0	0.0

```
[3841 rows x 8 columns]>
```

### 0.3 Question 2: Use Webscraping to Extract Tesla Revenue Data

Use the `requests` library to download the webpage <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm> Save the text of the response as a variable named `html_data`.

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

Parse the html data using `beautiful_soup` using parser i.e `html5lib` or `html.parser`.

```
[11]: html_data_parsed = BeautifulSoup(html_data, 'html.parser')
```

```
[12]: ## getting tables
tables = html_data_parsed.find_all('table')
#print(tables)
print(len(tables))
#iterating through tables to find quarterly revenue table
#it is table 1
for i, table in enumerate(tables):
    print(f'Table : {i}')
    print(table.prettify())
```

```

6
Table : 0
<table class="historical_data_table table">
  <thead>

```

```

<tr>
  <th colspan="2" style="text-align:center">
    Tesla Annual Revenue
  <br/>
  <span style="font-size:14px;">
    (Millions of US $)
  </span>
</th>
</tr>
</thead>
<tbody>
<tr>
  <td style="text-align:center">
    2021
  </td>
  <td style="text-align:center">
    $53,823
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2020
  </td>
  <td style="text-align:center">
    $31,536
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2019
  </td>
  <td style="text-align:center">
    $24,578
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2018
  </td>
  <td style="text-align:center">
    $21,461
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2017
  </td>
  <td style="text-align:center">

```

\$11,759	
2016	\$7,000
2015	\$4,046
2014	\$3,198
2013	\$2,013
2012	\$413
2011	



\$204
2010
\$117
2009
\$112

Table : 1

Tesla Quarterly Revenue (Millions of US \$)	
2022-09-30	\$21,454
2022-06-30	

```

    <td style="text-align:center">
      $16,934
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2022-03-31
    </td>
    <td style="text-align:center">
      $18,756
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2021-12-31
    </td>
    <td style="text-align:center">
      $17,719
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2021-09-30
    </td>
    <td style="text-align:center">
      $13,757
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2021-06-30
    </td>
    <td style="text-align:center">
      $11,958
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2021-03-31
    </td>
    <td style="text-align:center">
      $10,389
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2020-12-31
    </td>

```

```

    <td style="text-align:center">
      $10,744
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2020-09-30
    </td>
    <td style="text-align:center">
      $8,771
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2020-06-30
    </td>
    <td style="text-align:center">
      $6,036
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2020-03-31
    </td>
    <td style="text-align:center">
      $5,985
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019-12-31
    </td>
    <td style="text-align:center">
      $7,384
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019-09-30
    </td>
    <td style="text-align:center">
      $6,303
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019-06-30
    </td>

```

```

    <td style="text-align:center">
      $6,350
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019-03-31
    </td>
    <td style="text-align:center">
      $4,541
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018-12-31
    </td>
    <td style="text-align:center">
      $7,226
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018-09-30
    </td>
    <td style="text-align:center">
      $6,824
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018-06-30
    </td>
    <td style="text-align:center">
      $4,002
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018-03-31
    </td>
    <td style="text-align:center">
      $3,409
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2017-12-31
    </td>

```

```

    <td style="text-align:center">
      $3,288
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2017-09-30
    </td>
    <td style="text-align:center">
      $2,985
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2017-06-30
    </td>
    <td style="text-align:center">
      $2,790
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2017-03-31
    </td>
    <td style="text-align:center">
      $2,696
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2016-12-31
    </td>
    <td style="text-align:center">
      $2,285
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2016-09-30
    </td>
    <td style="text-align:center">
      $2,298
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2016-06-30
    </td>

```

```

    <td style="text-align:center">
      $1,270
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2016-03-31
    </td>
    <td style="text-align:center">
      $1,147
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2015-12-31
    </td>
    <td style="text-align:center">
      $1,214
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2015-09-30
    </td>
    <td style="text-align:center">
      $937
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2015-06-30
    </td>
    <td style="text-align:center">
      $955
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2015-03-31
    </td>
    <td style="text-align:center">
      $940
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2014-12-31
    </td>

```

```

    <td style="text-align:center">
      $957
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2014-09-30
    </td>
    <td style="text-align:center">
      $852
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2014-06-30
    </td>
    <td style="text-align:center">
      $769
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2014-03-31
    </td>
    <td style="text-align:center">
      $621
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2013-12-31
    </td>
    <td style="text-align:center">
      $615
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2013-09-30
    </td>
    <td style="text-align:center">
      $431
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2013-06-30
    </td>

```

```

    <td style="text-align:center">
      $405
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2013-03-31
    </td>
    <td style="text-align:center">
      $562
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2012-12-31
    </td>
    <td style="text-align:center">
      $306
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2012-09-30
    </td>
    <td style="text-align:center">
      $50
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2012-06-30
    </td>
    <td style="text-align:center">
      $27
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2012-03-31
    </td>
    <td style="text-align:center">
      $30
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2011-12-31
    </td>

```



```

    <td style="text-align:center">
      $39
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2011-09-30
    </td>
    <td style="text-align:center">
      $58
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2011-06-30
    </td>
    <td style="text-align:center">
      $58
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2011-03-31
    </td>
    <td style="text-align:center">
      $49
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2010-12-31
    </td>
    <td style="text-align:center">
      $36
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2010-09-30
    </td>
    <td style="text-align:center">
      $31
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2010-06-30
    </td>

```

```
  |
```

Table : 2

```

<table class="historical_data_table table">
<thead>
<tr>
<th style="text-align:center">
    Sector
</th>
<th style="text-align:center">
    Industry
</th>

```

Market Cap			
Revenue			
<a href="https://www.macrotrends.net/stocks/sector/5/auto-tires-trucks">Auto/Tires/Trucks</a>			
<a href="https://www.macrotrends.net/stocks/industry/7/">Auto Manufacturers - Domestic</a>			
\$549.575B			
\$53.823B			
<span>Tesla is the market leader in battery-powered electric car sales in the United States, with roughly 70% market share. The company's flagship Model 3 is the best-selling EV model in the United States. Tesla, which has managed to garner the reputation of a gold standard over the years, is now a far bigger entity that what it started off since its IPO in 2010, with its market cap crossing \$1 trillion for the first time in October 2021.? The EV king's market capitalization is more than the combined value of legacy automakers including Toyota, Volkswagen, Daimler, General Motors and Ford.Over the years, Tesla has shifted from developing niche products for affluent buyers to making more affordable EVs for the masses. The firm's three-pronged business model approach of direct sales, servicing, and charging its EVs sets it apart from other carmakers. Tesla, which is touted as the clean energy revolutionary automaker, is much more than just a car manufacturer.</span>			

Table : 3

Stock Name	Country	Market Cap	PE Ratio
<a href="/stocks/charts/GM/general-motors/revenue">General Motors (GM)</a>	United States	\$53.930B	5.56
<a href="/stocks/charts/F/ford-motor/revenue">Ford Motor (F)</a>	United States	\$52.668B	

8.09		
<a href="/stocks/charts/HOG/harley-davidson/revenue">Harley-Davidson (HOG)</a>	United States	\$6.762B
		9.56
<a href="/stocks/charts/PII/polaris/revenue">Polaris (PII)</a>	United States	\$6.267B
		11.86
<a href="/stocks/charts/IAA/iaa/revenue">IAA (IAA)</a>	United States	\$5.134B

```
 16.40 || Fisker \(FSR\) | United States | $2.261B | 0.00 |
| Lion Electric \(LEV\) | Canada | $0.551B | 0.00 |
| Volta \(VLTA\) | United States | $0.071B |  |

```

```

    <td style="text-align:center">
      0.00
    </td>
  </tr>
  <tr>
    <td style="text-align:left">
      <a href="/stocks/charts/BRDS/bird-global/revenue">
        Bird Global (BRDS)
      </a>
    </td>
    <td style="text-align:center">
      United States
    </td>
    <td style="text-align:center">
      $0.054B
    </td>
    <td style="text-align:center">
      0.00
    </td>
  </tr>
  <tr>
    <td style="text-align:left">
      <a href="/stocks/charts/ZEV/lightning-emotors/revenue">
        Lightning EMotors (ZEV)
      </a>
    </td>
    <td style="text-align:center">
      United States
    </td>
    <td style="text-align:center">
      $0.043B
    </td>
    <td style="text-align:center">
      0.00
    </td>
  </tr>
</tbody>
</table>

```

Table : 4

```

<table class="table">
  <thead>
    <tr>
      <th>
        Link Preview
      </th>
      <th>
        HTML Code (Click to Copy)
      </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>
        <a href="/stocks/charts/BRDS/bird-global/revenue">
          Bird Global (BRDS)
        </a>
      </td>
      <td>
        United States
      </td>
      <td>
        $0.054B
      </td>
      <td>
        0.00
      </td>
    </tr>
    <tr>
      <td>
        <a href="/stocks/charts/ZEV/lightning-emotors/revenue">
          Lightning EMotors (ZEV)
        </a>
      </td>
      <td>
        United States
      </td>
      <td>
        $0.043B
      </td>
      <td>
        0.00
      </td>
    </tr>
  </tbody>
</table>

```

```

        </th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>
            <a>
                Tesla Revenue 2010-2022 | TSLA
            </a>
        </td>
        <td>
            <input class="modal_link" size="60" type="text" value="&lt;a
href='https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue'&gt;Tesla
Revenue 2010-2022 | TSLA&lt;/a&gt;"/>
        </td>
    </tr>
    <tr>
        <td>
            <a>
                Macrotrends
            </a>
        </td>
        <td>
            <input class="modal_link" size="60" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/TSLA/tesla/revenue'&gt;Macrotrends&lt;/a&gt;"/>
        </td>
    </tr>
    <tr>
        <td>
            <a>
                Source
            </a>
        </td>
        <td>
            <input class="modal_link" size="60" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/TSLA/tesla/revenue'&gt;Source&lt;/a&gt;"/>
        </td>
    </tr>
</tbody>
</table>

```

Table : 5

```

<table class="table">
    <thead>
        <tr>
            <th>
                Link Preview
            </th>

```



```

<th>
    HTML Code (Click to Copy)
</th>
</tr>
</thead>
<tbody>
<tr>
<td>
<a>
    Tesla Revenue 2010-2022 | TSLA
</a>
</td>
<td>
<input class="modal_link" size="50" type="text" value="&lt;a
href='https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue'&gt;Tesla
Revenue 2010-2022 | TSLA&lt;/a&gt;"/>
</td>
</tr>
<tr>
<td>
<a>
    Macrotrends
</a>
</td>
<td>
<input class="modal_link" size="50" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/TSLA/tesla/revenue'&gt;Macrotrends&lt;/a&gt;"/>
</td>
</tr>
<tr>
<td>
<a>
    Source
</a>
</td>
<td>
<input class="modal_link" size="50" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/TSLA/tesla/revenue'&gt;Source&lt;/a&gt;"/>
</td>
</tr>
</tbody>
</table>

```

```

[13]: qrevenue_table = tables[1]  # This is the quarterly revenue table
      ## DataFrame as asked
      tesla_revenue = pd.DataFrame(columns=['Date', 'Revenue'])

```

```

# print(qrevenue_table.prettify())
## Iterating through quarterly revenue table and populating the Data Frame
for row in qrevenue_table.tbody.find_all('tr'):
    col = row.find_all('td')
    if col:
        date = col[0].text.strip()
        revenue = col[1].text.strip()
        rowdf = pd.DataFrame([{'Date':date, 'Revenue':revenue}])
        tesla_revenue = pd.concat([tesla_revenue, rowdf], ignore_index=True)

tesla_revenue

```

```

[13]:

```

	Date	Revenue
0	2022-09-30	\$21,454
1	2022-06-30	\$16,934
2	2022-03-31	\$18,756
3	2021-12-31	\$17,719
4	2021-09-30	\$13,757
5	2021-06-30	\$11,958
6	2021-03-31	\$10,389
7	2020-12-31	\$10,744
8	2020-09-30	\$8,771
9	2020-06-30	\$6,036
10	2020-03-31	\$5,985
11	2019-12-31	\$7,384
12	2019-09-30	\$6,303
13	2019-06-30	\$6,350
14	2019-03-31	\$4,541
15	2018-12-31	\$7,226
16	2018-09-30	\$6,824
17	2018-06-30	\$4,002
18	2018-03-31	\$3,409
19	2017-12-31	\$3,288
20	2017-09-30	\$2,985
21	2017-06-30	\$2,790
22	2017-03-31	\$2,696
23	2016-12-31	\$2,285
24	2016-09-30	\$2,298
25	2016-06-30	\$1,270
26	2016-03-31	\$1,147
27	2015-12-31	\$1,214
28	2015-09-30	\$937
29	2015-06-30	\$955
30	2015-03-31	\$940
31	2014-12-31	\$957
32	2014-09-30	\$852
33	2014-06-30	\$769

34	2014-03-31	\$621
35	2013-12-31	\$615
36	2013-09-30	\$431
37	2013-06-30	\$405
38	2013-03-31	\$562
39	2012-12-31	\$306
40	2012-09-30	\$50
41	2012-06-30	\$27
42	2012-03-31	\$30
43	2011-12-31	\$39
44	2011-09-30	\$58
45	2011-06-30	\$58
46	2011-03-31	\$49
47	2010-12-31	\$36
48	2010-09-30	\$31
49	2010-06-30	\$28
50	2010-03-31	\$21
51	2009-12-31	
52	2009-09-30	\$46
53	2009-06-30	\$27

Using BeautifulSoup or the `read_html` function extract the table with **Tesla Revenue** and store it into a dataframe named `tesla_revenue`. The dataframe should have columns **Date** and **Revenue**.

Step-by-step instructions

Here are the step-by-step instructions:

1. Create an Empty DataFrame
2. Find the Relevant Table
3. Check for the Tesla Quarterly Revenue Table
4. Iterate Through Rows in the Table Body
5. Extract Data from Columns
6. Append Data to the DataFrame

[Click here](#) if you need help locating the table

Below is the code to isolate the table, you will now need to loop through the rows and columns

```
soup.find_all("tbody")[1]
```

If you want to use the `read_html` function the table is located at index 1

We are focusing on quarterly revenue in the lab.

Execute the following line to remove the comma and dollar sign from the **Revenue** column.

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

```
[15]: print(tesla_revenue.shape)  
      print(tesla_revenue.isna().sum())
```

```
(54, 2)  
Date      0  
Revenue   0  
dtype: int64
```

Execute the following lines to remove an null or empty strings in the Revenue column.

```
[16]: tesla_revenue.dropna(inplace=True)  
  
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]  
print(tesla_revenue.isna().sum())
```

```
Date      0  
Revenue   0  
dtype: int64
```

Display the last 5 row of the `tesla_revenue` dataframe using the `tail` function. Take a screenshot of the results.

```
[17]: tesla_revenue.tail(5)
```

```
[17]:      Date Revenue  
48  2010-09-30      31  
49  2010-06-30      28  
50  2010-03-31      21  
52  2009-09-30      46  
53  2009-06-30      27
```

#### 0.4 Question 3: Use `yfinance` to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is GameStop and its ticker symbol is `GME`.

```
[18]: gme = yf.Ticker('GME')
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `gme_data`. Set the `period` parameter to "max" so we get information for the maximum amount of time.

```
[19]: gme_data = gme.history(period='max')
```

**Reset the index** using the `reset_index(inplace=True)` function on the `gme_data` DataFrame and display the first five rows of the `gme_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 3 to the results below.

```
[20]: gme_data.reset_index(inplace=True)
      gme_data.head
```

```
[20]: <bound method NDFrame.head of
High          Low          Close \
0    2002-02-13 00:00:00-05:00  1.620129  1.693350  1.603296  1.691667
1    2002-02-14 00:00:00-05:00  1.712707  1.716074  1.670626  1.683250
2    2002-02-15 00:00:00-05:00  1.683251  1.687459  1.658002  1.674834
3    2002-02-19 00:00:00-05:00  1.666418  1.666418  1.578047  1.607504
4    2002-02-20 00:00:00-05:00  1.615921  1.662210  1.603296  1.662210
...
5944 2025-09-29 00:00:00-04:00  27.200001  27.209999  26.790001  27.209999
5945 2025-09-30 00:00:00-04:00  27.209999  27.340000  26.799999  27.280001
5946 2025-10-01 00:00:00-04:00  27.250000  27.790001  27.180000  27.690001
5947 2025-10-02 00:00:00-04:00  28.000000  28.100000  26.719999  27.219999
5948 2025-10-03 00:00:00-04:00  26.360001  26.990000  25.155001  25.379999
```

	Volume	Dividends	Stock Splits
0	76216000	0.0	0.0
1	11021600	0.0	0.0
2	8389600	0.0	0.0
3	7410400	0.0	0.0
4	6892800	0.0	0.0
...	...	...	...
5944	7979000	0.0	0.0
5945	5392800	0.0	0.0
5946	8187100	0.0	0.0
5947	14526000	0.0	0.0
5948	11525945	0.0	0.0

```
[5949 rows x 8 columns]>
```

## 0.5 Question 4: Use Webscraping to Extract GME Revenue Data

Use the `requests` library to download the webpage <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html>. Save the text of the response as a variable named `html_data_2`.

```
[21]: url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
      ↪IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html'
      html_data2 = requests.get(url).text
```

Parse the html data using `beautiful_soup` using parser i.e `html5lib` or `html.parser`.

```
[22]: html_data2_parsed = BeautifulSoup(html_data2, 'html.parser')
```

```
[23]: gmetables = html_data2_parsed.find_all('table')
      print(len(gmetables))
```

6

```
[24]: for i, table in enumerate(gmetables):
      print(f'Table:{i}')
      print(table.prettify())
```

Table:0

```
<table class="historical_data_table table">
  <thead>
    <tr>
      <th colspan="2" style="text-align:center">
        GameStop Annual Revenue
      <br/>
      <span style="font-size:14px;">
        (Millions of US $)
      </span>
    </th>
  </tr>
</thead>
<tbody>
  <tr>
    <td style="text-align:center">
      2020
    </td>
    <td style="text-align:center">
      $6,466
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019
    </td>
    <td style="text-align:center">
      $8,285
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018
    </td>
    <td style="text-align:center">
      $8,547
    </td>
  </tr>
  <tr>
```

2017	\$7,965
2016	\$9,364
2015	\$9,296
2014	\$9,040
2013	\$8,887
2012	\$9,551

2011	\$9,474
2010	\$9,078
2009	\$8,806
2008	\$7,094
2007	\$5,319
2006	\$3,092



```
  |
```

Table:1

```

<table class="historical_data_table table">
<thead>
<tr>
<th colspan="2" style="text-align:center">
    GameStop Quarterly Revenue
<br/>
    <span style="font-size:14px;">
        (Millions of US $)
    </span>
</th>
</tr>
</thead>
<tbody>
<tr>
  |
```

```

<tr>
  <td style="text-align:center">
    2019-07-31
  </td>
  <td style="text-align:center">
    $1,286
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2019-04-30
  </td>
  <td style="text-align:center">
    $1,548
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2019-01-31
  </td>
  <td style="text-align:center">
    $3,063
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2018-10-31
  </td>
  <td style="text-align:center">
    $1,935
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2018-07-31
  </td>
  <td style="text-align:center">
    $1,501
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2018-04-30
  </td>
  <td style="text-align:center">
    $1,786
  </td>
</tr>

```

```

<tr>
  <td style="text-align:center">
    2018-01-31
  </td>
  <td style="text-align:center">
    $2,825
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2017-10-31
  </td>
  <td style="text-align:center">
    $1,989
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2017-07-31
  </td>
  <td style="text-align:center">
    $1,688
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2017-04-30
  </td>
  <td style="text-align:center">
    $2,046
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2017-01-31
  </td>
  <td style="text-align:center">
    $2,403
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2016-10-31
  </td>
  <td style="text-align:center">
    $1,959
  </td>
</tr>

```

```

<tr>
  <td style="text-align:center">
    2016-07-31
  </td>
  <td style="text-align:center">
    $1,632
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2016-04-30
  </td>
  <td style="text-align:center">
    $1,972
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2016-01-31
  </td>
  <td style="text-align:center">
    $3,525
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2015-10-31
  </td>
  <td style="text-align:center">
    $2,016
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2015-07-31
  </td>
  <td style="text-align:center">
    $1,762
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2015-04-30
  </td>
  <td style="text-align:center">
    $2,061
  </td>
</tr>

```

2015-01-31	\$3,476
2014-10-31	\$2,092
2014-07-31	\$1,731
2014-04-30	\$1,996
2014-01-31	\$3,684
2013-10-31	\$2,107

2013-07-31	\$1,384
2013-04-30	\$1,865
2013-01-31	\$3,562
2012-10-31	\$1,773
2012-07-31	\$1,550
2012-04-30	\$2,002

```

<tr>
  <td style="text-align:center">
    2012-01-31
  </td>
  <td style="text-align:center">
    $3,579
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2011-10-31
  </td>
  <td style="text-align:center">
    $1,947
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2011-07-31
  </td>
  <td style="text-align:center">
    $1,744
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2011-04-30
  </td>
  <td style="text-align:center">
    $2,281
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2011-01-31
  </td>
  <td style="text-align:center">
    $3,693
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2010-10-31
  </td>
  <td style="text-align:center">
    $1,899
  </td>
</tr>

```

```

<tr>
  <td style="text-align:center">
    2010-07-31
  </td>
  <td style="text-align:center">
    $1,799
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2010-04-30
  </td>
  <td style="text-align:center">
    $2,083
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2010-01-31
  </td>
  <td style="text-align:center">
    $3,524
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2009-10-31
  </td>
  <td style="text-align:center">
    $1,835
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2009-07-31
  </td>
  <td style="text-align:center">
    $1,739
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2009-04-30
  </td>
  <td style="text-align:center">
    $1,981
  </td>
</tr>

```



```

<tr>
  <td style="text-align:center">
    2009-01-31
  </td>
  <td style="text-align:center">
    $3,492
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2008-10-31
  </td>
  <td style="text-align:center">
    $1,696
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2008-07-31
  </td>
  <td style="text-align:center">
    $1,804
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2008-04-30
  </td>
  <td style="text-align:center">
    $1,814
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2008-01-31
  </td>
  <td style="text-align:center">
    $2,866
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2007-10-31
  </td>
  <td style="text-align:center">
    $1,611
  </td>
</tr>

```

2007-07-31	\$1,338
2007-04-30	\$1,279
2007-01-31	\$2,304
2006-10-31	\$1,012
2006-07-31	\$963
2006-04-30	\$1,040

```

<tr>
  <td style="text-align:center">
    2006-01-31
  </td>
  <td style="text-align:center">
    $1,667
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2005-10-31
  </td>
  <td style="text-align:center">
    $534
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2005-07-31
  </td>
  <td style="text-align:center">
    $416
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2005-04-30
  </td>
  <td style="text-align:center">
    $475
  </td>
</tr>
<tr>
  <td style="text-align:center">
    2005-01-31
  </td>
  <td style="text-align:center">
    $709
  </td>
</tr>
</tbody>
</table>

```

Table:2

```

<table class="historical_data_table table">
  <thead>
    <tr>
      <th style="text-align:center">

```

Sector
<th style="text-align:center"> Industry
<th style="text-align:center"> Market Cap
<th style="text-align:center"> Revenue
<div> <div> <a href="https://web.archive.org/web/20200814131437/https://www.macrotrends.net/stocks/sector/3/retail-wholesale">https://web.archive.org/web/20200814131437/https://www.macrotrends.net/stocks/sector/3/retail-wholesale</a> </div> <div>Retail/Wholesale</div> </div>
<div> <div> <a href="https://web.archive.org/web/20200814131437/https://www.macrotrends.net/stocks/industry/156/">https://web.archive.org/web/20200814131437/https://www.macrotrends.net/stocks/industry/156/</a> </div> <div>Retail - Consumer Electronics</div> </div>
\$0.293B
\$6.466B
<div> <div> <div> <div> <div>GameStop Corp. is the world's largest video game and entertainment software retailer. The company operates 4,816 retail stores across the United States and in fifteen countries worldwide. The company also operates two e-commerce sites, GameStop.com and EBgames.com, and publishes Game Informer? magazine, a leading multi-platform video game publication. GameStop Corp. sells new and used video game software, hardware and accessories for next generation video game systems from Sony, Nintendo, and Microsoft. In addition, the company sells PC entertainment software, related accessories and other merchandise.</div> </div> </div> </div> </div>

```

</tbody>
</table>

```

Table:3

historical_data_table table			
Stock Name	Country	Market Cap	PE Ratio
<a href="https://web.archive.org/web/20200814131437/https://www.macrotrends.net/stocks/charts/BBY/best-buy/revenue">https://web.archive.org/web/20200814131437/https://www.macrotrends.net/stocks/charts/BBY/best-buy/revenue</a>			
Best Buy (BBY)	United States	\$27.033B	18.16
<a href="https://web.archive.org/web/20200814131437/https://www.macrotrends.net/stocks/charts/AAN/aarons,-/revenue">https://web.archive.org/web/20200814131437/https://www.macrotrends.net/stocks/charts/AAN/aarons,-/revenue</a>			
Aaron's, (AAN)	United States		

```

</td>
<td style="text-align:center">
    $3.975B
</td>
<td style="text-align:center">
    15.14
</td>
</tr>
<tr>
<td style="text-align:left">
    <a href="https://web.archive.org/web/20200814131437/https://www.macrotrends.
net/stocks/charts/GMELY/gome-retail-holdings/revenue">
        GOME Retail Holdings (GMELY)
    </a>
</td>
<td style="text-align:center">
    China
</td>
<td style="text-align:center">
    $1.684B
</td>
<td style="text-align:center">
    0.00
</td>
</tr>
<tr>
<td style="text-align:left">
    <a href="https://web.archive.org/web/20200814131437/https://www.macrotrends.
net/stocks/charts/SYX/systemax/revenue">
        Systemax (SYX)
    </a>
</td>
<td style="text-align:center">
    United States
</td>
<td style="text-align:center">
    $0.873B
</td>
<td style="text-align:center">
    18.34
</td>
</tr>
<tr>
<td style="text-align:left">
    <a href="https://web.archive.org/web/20200814131437/https://www.macrotrends.
net/stocks/charts/CONN/conns/revenue">
        Conn's (CONN)
    </a>

```

```

</td>
<td style="text-align:center">
    United States
</td>
<td style="text-align:center">
    $0.325B
</td>
<td style="text-align:center">
    0.00
</td>
</tr>
<tr>
<td style="text-align:left">
    <a href="https://web.archive.org/web/20200814131437/https://www.macrotrends.
net/stocks/charts/TAIT/taitron-components/revenue">
        Taitron Components (TAIT)
    </a>
</td>
<td style="text-align:center">
    United States
</td>
<td style="text-align:center">
    $0.016B
</td>
<td style="text-align:center">
    10.50
</td>
</tr>
</tbody>
</table>

```

Table:4

```

<table class="table">
<thead>
<tr>
<th>
    Link Preview
</th>
<th>
    HTML Code (Click to Copy)
</th>
</tr>
</thead>
<tbody>
<tr>
<td>
<a>
    GameStop Revenue 2006-2020 | GME

```

```

        </a>
    </td>
    <td>
        <input class="modal_link" size="60" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/GME/gamestop/revenue'&gt;GameStop Revenue
2006-2020 | GME&lt;/a&gt;" />
    </td>
</tr>
<tr>
    <td>
        <a>
            Macrotrends
        </a>
    </td>
    <td>
        <input class="modal_link" size="60" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/GME/gamestop/revenue'&gt;Macrotrends&lt;/a&gt;"
/>
    </td>
</tr>
<tr>
    <td>
        <a>
            Source
        </a>
    </td>
    <td>
        <input class="modal_link" size="60" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/GME/gamestop/revenue'&gt;Source&lt;/a&gt;" />
    </td>
</tr>
</tbody>
</table>

```

Table:5

```

<table class="table">
    <thead>
        <tr>
            <th>
                Link Preview
            </th>
            <th>
                HTML Code (Click to Copy)
            </th>
        </tr>
    </thead>
    <tbody>
        <tr>

```



```

<td>
  <a>
    GameStop Revenue 2006-2020 | GME
  </a>
</td>
<td>
  <input class="modal_link" size="50" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/GME/gamestop/revenue'&gt;GameStop Revenue
2006-2020 | GME&lt;/a&gt;" />
</td>
</tr>
<tr>
<td>
  <a>
    Macrotrends
  </a>
</td>
<td>
  <input class="modal_link" size="50" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/GME/gamestop/revenue'&gt;Macrotrends&lt;/a&gt;"
/>
</td>
</tr>
<tr>
<td>
  <a>
    Source
  </a>
</td>
<td>
  <input class="modal_link" size="50" type="text" value="&lt;a href='https://w
ww.macrotrends.net/stocks/charts/GME/gamestop/revenue'&gt;Source&lt;/a&gt;" />
</td>
</tr>
</tbody>
</table>

```

Using BeautifulSoup or the `read_html` function extract the table with GameStop Revenue and store it into a dataframe named `gme_revenue`. The dataframe should have columns Date and Revenue. Make sure the comma and dollar sign is removed from the Revenue column.

**Note:** Use the method similar to what you did in question 2.

```

[25]: #html_data2_parsed.find_all("tbody")[1]
      gme_revenue_table = gmetables[1]
      print(gme_revenue_table.prettify())

```

```

<table class="historical_data_table table">

```

```

<thead>
  <tr>
    <th colspan="2" style="text-align:center">
      GameStop Quarterly Revenue
    <br/>
    <span style="font-size:14px;">
      (Millions of US $)
    </span>
  </th>
</tr>
</thead>
<tbody>
  <tr>
    <td style="text-align:center">
      2020-04-30
    </td>
    <td style="text-align:center">
      $1,021
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2020-01-31
    </td>
    <td style="text-align:center">
      $2,194
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019-10-31
    </td>
    <td style="text-align:center">
      $1,439
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019-07-31
    </td>
    <td style="text-align:center">
      $1,286
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019-04-30
    </td>

```

```

    <td style="text-align:center">
      $1,548
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2019-01-31
    </td>
    <td style="text-align:center">
      $3,063
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018-10-31
    </td>
    <td style="text-align:center">
      $1,935
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018-07-31
    </td>
    <td style="text-align:center">
      $1,501
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018-04-30
    </td>
    <td style="text-align:center">
      $1,786
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2018-01-31
    </td>
    <td style="text-align:center">
      $2,825
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2017-10-31
    </td>

```

```

    <td style="text-align:center">
      $1,989
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2017-07-31
    </td>
    <td style="text-align:center">
      $1,688
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2017-04-30
    </td>
    <td style="text-align:center">
      $2,046
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2017-01-31
    </td>
    <td style="text-align:center">
      $2,403
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2016-10-31
    </td>
    <td style="text-align:center">
      $1,959
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2016-07-31
    </td>
    <td style="text-align:center">
      $1,632
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2016-04-30
    </td>

```

```

    <td style="text-align:center">
      $1,972
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2016-01-31
    </td>
    <td style="text-align:center">
      $3,525
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2015-10-31
    </td>
    <td style="text-align:center">
      $2,016
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2015-07-31
    </td>
    <td style="text-align:center">
      $1,762
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2015-04-30
    </td>
    <td style="text-align:center">
      $2,061
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2015-01-31
    </td>
    <td style="text-align:center">
      $3,476
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2014-10-31
    </td>

```

```

    <td style="text-align:center">
      $2,092
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2014-07-31
    </td>
    <td style="text-align:center">
      $1,731
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2014-04-30
    </td>
    <td style="text-align:center">
      $1,996
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2014-01-31
    </td>
    <td style="text-align:center">
      $3,684
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2013-10-31
    </td>
    <td style="text-align:center">
      $2,107
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2013-07-31
    </td>
    <td style="text-align:center">
      $1,384
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2013-04-30
    </td>

```

```

    <td style="text-align:center">
      $1,865
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2013-01-31
    </td>
    <td style="text-align:center">
      $3,562
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2012-10-31
    </td>
    <td style="text-align:center">
      $1,773
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2012-07-31
    </td>
    <td style="text-align:center">
      $1,550
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2012-04-30
    </td>
    <td style="text-align:center">
      $2,002
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2012-01-31
    </td>
    <td style="text-align:center">
      $3,579
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2011-10-31
    </td>

```

```

    <td style="text-align:center">
      $1,947
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2011-07-31
    </td>
    <td style="text-align:center">
      $1,744
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2011-04-30
    </td>
    <td style="text-align:center">
      $2,281
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2011-01-31
    </td>
    <td style="text-align:center">
      $3,693
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2010-10-31
    </td>
    <td style="text-align:center">
      $1,899
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2010-07-31
    </td>
    <td style="text-align:center">
      $1,799
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2010-04-30
    </td>

```



```

    <td style="text-align:center">
      $2,083
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2010-01-31
    </td>
    <td style="text-align:center">
      $3,524
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2009-10-31
    </td>
    <td style="text-align:center">
      $1,835
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2009-07-31
    </td>
    <td style="text-align:center">
      $1,739
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2009-04-30
    </td>
    <td style="text-align:center">
      $1,981
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2009-01-31
    </td>
    <td style="text-align:center">
      $3,492
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2008-10-31
    </td>

```

```

    <td style="text-align:center">
      $1,696
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2008-07-31
    </td>
    <td style="text-align:center">
      $1,804
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2008-04-30
    </td>
    <td style="text-align:center">
      $1,814
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2008-01-31
    </td>
    <td style="text-align:center">
      $2,866
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2007-10-31
    </td>
    <td style="text-align:center">
      $1,611
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2007-07-31
    </td>
    <td style="text-align:center">
      $1,338
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2007-04-30
    </td>

```

```

    <td style="text-align:center">
      $1,279
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2007-01-31
    </td>
    <td style="text-align:center">
      $2,304
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2006-10-31
    </td>
    <td style="text-align:center">
      $1,012
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2006-07-31
    </td>
    <td style="text-align:center">
      $963
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2006-04-30
    </td>
    <td style="text-align:center">
      $1,040
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2006-01-31
    </td>
    <td style="text-align:center">
      $1,667
    </td>
  </tr>
  <tr>
    <td style="text-align:center">
      2005-10-31
    </td>

```

```

        <td style="text-align:center">
            $534
        </td>
    </tr>
    <tr>
        <td style="text-align:center">
            2005-07-31
        </td>
        <td style="text-align:center">
            $416
        </td>
    </tr>
    <tr>
        <td style="text-align:center">
            2005-04-30
        </td>
        <td style="text-align:center">
            $475
        </td>
    </tr>
    <tr>
        <td style="text-align:center">
            2005-01-31
        </td>
        <td style="text-align:center">
            $709
        </td>
    </tr>
</tbody>
</table>

```

[Click here](#) if you need help locating the table

Below is the code to isolate the table, you will now need to loop through the rows and columns

```
soup.find_all("tbody")[1]
```

If you want to use the `read_html` function the table is located at index 1

```

[26]: gme_revenue=pd.DataFrame(columns=['Date', 'Revenue'])
      #extract rows and cols from gme_rvenue_table and populate the gme_revenue_
      ↪dataframe
      for row in gme_revenue_table.tbody.find_all('tr'):
          col = row.find_all('td')
          if col:

```

```

date = col[0].text.strip()
revenue = col[1].text.strip()
rowdf = pd.DataFrame([{'Date': date, 'Revenue': revenue}])
gme_revenue = pd.concat([gme_revenue, rowdf], ignore_index=True)

gme_revenue

```

```

[26]:
      Date Revenue
0  2020-04-30  $1,021
1  2020-01-31  $2,194
2  2019-10-31  $1,439
3  2019-07-31  $1,286
4  2019-04-30  $1,548
..      ...
57 2006-01-31  $1,667
58 2005-10-31   $534
59 2005-07-31   $416
60 2005-04-30   $475
61 2005-01-31   $709

```

[62 rows x 2 columns]

```

[27]: print(gme_revenue.isna().sum())

```

```

Date      0
Revenue    0
dtype: int64

```

```

[28]: gme_revenue["Revenue"] = gme_revenue['Revenue'].str.
      ↪replace(',', '\\$', "", regex=True)

```

```

[29]: print(gme_revenue.tail(5))

```

```

      Date Revenue
57 2006-01-31   1667
58 2005-10-31    534
59 2005-07-31    416
60 2005-04-30    475
61 2005-01-31    709

```

```

[30]: #tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
      #print(tesla_revenue.isna().sum())
      gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
      print(gme_revenue.isna().sum())
      print(gme_revenue.shape)

```

```

Date      0

```

```
Revenue      0
dtype: int64
(62, 2)
```

Display the last five rows of the `gme_revenue` dataframe using the `tail` function. Take a screenshot of the results.

```
[31]: gme_revenue.tail(5)
```

```
[31]:      Date Revenue
57  2006-01-31    1667
58  2005-10-31     534
59  2005-07-31     416
60  2005-04-30     475
61  2005-01-31     709
```

## 0.6 Question 5: Plot Tesla Stock Graph

Use the `make_graph` function to graph the Tesla Stock Data, also provide a title for the graph. Note the graph will only show data upto June 2021.

Hint

You just need to invoke the `make_graph` function with the required parameter to print the graph.

```
[32]: make_graph(tesla_data, tesla_revenue, 'Tesla')
```

```
/tmp/ipykernel_5559/109047474.py:5: UserWarning:
```

```
The argument 'infer_datetime_format' is deprecated and will be removed in a
future version. A strict version of it is now the default, see
https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You
can safely remove this argument.
```

```
/tmp/ipykernel_5559/109047474.py:6: UserWarning:
```

```
The argument 'infer_datetime_format' is deprecated and will be removed in a
future version. A strict version of it is now the default, see
https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You
can safely remove this argument.
```

```
<IPython.core.display.HTML object>
```

## 0.7 Question 6: Plot GameStop Stock Graph

Use the `make_graph` function to graph the GameStop Stock Data, also provide a title for the graph. The structure to call the `make_graph` function is `make_graph(gme_data, gme_revenue, 'GameStop')`. Note the graph will only show data upto June 2021.

Hint

You just need to invoke the `make_graph` function with the required parameter to print the graph.

```
[33]: make_graph(gme_data, gme_revenue, 'GameStop')
```

```
/tmp/ipykernel_5559/109047474.py:5: UserWarning:
```

```
The argument 'infer_datetime_format' is deprecated and will be removed in a
future version. A strict version of it is now the default, see
https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You
can safely remove this argument.
```

```
/tmp/ipykernel_5559/109047474.py:6: UserWarning:
```

```
The argument 'infer_datetime_format' is deprecated and will be removed in a
future version. A strict version of it is now the default, see
https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You
can safely remove this argument.
```

```
<IPython.core.display.HTML object>
```

About the Authors:

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Azim Hirjani

## 0.8 Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2022-02-28	1.2	Lakshmi Holla	Changed the URL of GameStop
2020-11-10	1.1	Malika Singla	Deleted the Optional part
2020-08-27	1.0	Malika Singla	Added lab to GitLab

##

© IBM Corporation 2020. All rights reserved.