```
pip install yfinance beautifulsoup4 requests pandas matplotlib plotly
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.11/dist-packages (0.2.54)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (4.13.3)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (2.32.3)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: plotly in /usr/local/lib/python3.11/dist-packages (5.24.1)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.11/dist-packages (from yfinance) (1.26.4)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.11/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: platformdirs>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (4.3.6)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2025.1)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.11/dist-packages (from yfinance) (3.17.9)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4) (2.6)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests) (2025.1.31)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.1)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from plotly) (9.0.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

```
import yfinance as yf
import pandas as pd
```

```
tesla = yf.Ticker("TSLA")
tesla_data = tesla.history(period="max")
```

```
print(tesla_data.head())
```

```
                               Open      High       Low     Close     Volume  \
Date
2010-06-29 00:00:00-04:00  1.266667  1.666667  1.169333  1.592667  281494500
2010-06-30 00:00:00-04:00  1.719333  2.028000  1.553333  1.588667  257806500
2010-07-01 00:00:00-04:00  1.666667  1.728000  1.351333  1.464000  123282000
2010-07-02 00:00:00-04:00  1.533333  1.540000  1.247333  1.280000   77097000
2010-07-06 00:00:00-04:00  1.333333  1.333333  1.055333  1.074000  103003500

                           Dividends  Stock Splits
Date
2010-06-29 00:00:00-04:00        0.0           0.0
2010-06-30 00:00:00-04:00        0.0           0.0
2010-07-01 00:00:00-04:00        0.0           0.0
2010-07-02 00:00:00-04:00        0.0           0.0
2010-07-06 00:00:00-04:00        0.0           0.0
```

```
import requests
from bs4 import BeautifulSoup
```

```
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")
```

```
tables = soup.find_all("table")
for table in tables:
    if "Tesla Quarterly Revenue" in str(table):
        tesla_revenue = pd.read_html(str(table))[0]
        break
```

```
print(tesla_revenue.head())
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-13-7ba28e6a5527> in <cell line: 0>()
----> 1 print(tesla_revenue.head())

AttributeError: 'NoneType' object has no attribute 'head'
```

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")

tables = soup.find_all("table")
tesla_revenue = None  # Initialize tesla_revenue to None

for table in tables:
    if "Tesla Quarterly Revenue" in str(table):
        tesla_revenue = pd.read_html(str(table))[0]
        break

# Check if tesla_revenue was assigned a value
if tesla_revenue is not None:
    print(tesla_revenue.head())
else:
    print("Table with 'Tesla Quarterly Revenue' not found on the page.")
```

```
Table with 'Tesla Quarterly Revenue' not found on the page.
```

```
gamestop = yf.Ticker("GME")
gamestop_data = gamestop.history(period="max")
```

```
print(gamestop_data.head())
```

```
                                Open      High       Low     Close    Volume  \
Date
2002-02-13 00:00:00-05:00  1.620129  1.693350  1.603296  1.691667  76216000
2002-02-14 00:00:00-05:00  1.712707  1.716073  1.670626  1.683250  11021600
2002-02-15 00:00:00-05:00  1.683250  1.687458  1.658001  1.674834   8389600
2002-02-19 00:00:00-05:00  1.666418  1.666418  1.578047  1.607504   7410400
2002-02-20 00:00:00-05:00  1.615920  1.662210  1.603296  1.662210   6892800

                           Dividends  Stock Splits
Date
2002-02-13 00:00:00-05:00        0.0           0.0
2002-02-14 00:00:00-05:00        0.0           0.0
2002-02-15 00:00:00-05:00        0.0           0.0
2002-02-19 00:00:00-05:00        0.0           0.0
2002-02-20 00:00:00-05:00        0.0           0.0
```

```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")
```

```
tables = soup.find_all("table")
for table in tables:
    if "GameStop Quarterly Revenue" in str(table):
        gamestop_revenue = pd.read_html(str(table))[0]
        break
```

```
print(gamestop_revenue.head())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-18-79fecad2445e> in <cell line: 0>()
----> 1 print(gamestop_revenue.head())

NameError: name 'gamestop_revenue' is not defined
```

```
import matplotlib.pyplot as plt


plt.figure(figsize=(7,4))
plt.plot(tesla_data.index, tesla_data["Close"], label="Tesla Stock Price")
plt.xlabel("Date")
plt.ylabel("Stock Price")
plt.title("Tesla Stock Price Over Time")
plt.legend()
plt.show()
```


```
plt.figure(figsize=(10,5))
plt.plot(pd.to_datetime(tesla_revenue["Date"]), tesla_revenue["Revenue"], label="Tesla Revenue", color='red')
plt.xlabel("Date")
plt.ylabel("Revenue (Billion $)")
plt.title("Tesla Revenue Over Time")
plt.legend()
plt.show()
```

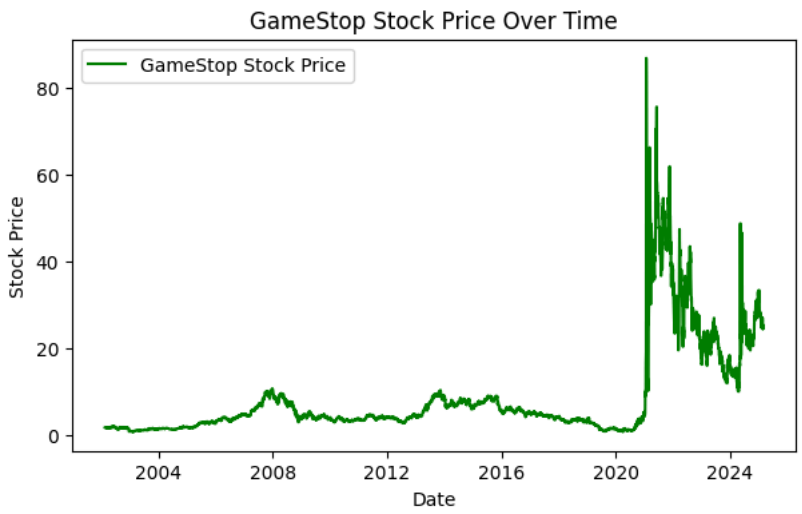
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-21-0d9d822d9009> in <cell line: 0>()
      1 plt.figure(figsize=(10,5))
----> 2 plt.plot(pd.to_datetime(tesla_revenue["Date"]), tesla_revenue["Revenue"], label="Tesla Revenue", color='red')
      3 plt.xlabel("Date")
      4 plt.ylabel("Revenue (Billion $)")
      5 plt.title("Tesla Revenue Over Time")

TypeError: 'NoneType' object is not subscriptable

<Figure size 1000x500 with 0 Axes>
```

```
print(tesla_revenue)
```

```
None
```

```
plt.figure(figsize=(7,4))
plt.plot(gamestop_data.index, gamestop_data["Close"], label="GameStop Stock Price", color='green')
plt.xlabel("Date")
plt.ylabel("Stock Price")
plt.title("GameStop Stock Price Over Time")
```

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




```
plt.figure(figsize=(10,5))
plt.plot(pd.to_datetime(gamestop_revenue["Date"]), gamestop_revenue["Revenue"], label="GameStop Revenue", color='purple')
plt.xlabel("Date")
plt.ylabel("Revenue (Million $)")
plt.title("GameStop Revenue Over Time")
plt.legend()
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-28-2ef3790634e2> in <cell line: 0>()
      1 plt.figure(figsize=(10,5))
----> 2 plt.plot(pd.to_datetime(gamestop_revenue["Date"]), gamestop_revenue["Revenue"], label="GameStop Revenue", color='purple')
      3 plt.xlabel("Date")
      4 plt.ylabel("Revenue (Million $)")
      5 plt.title("GameStop Revenue Over Time")

NameError: name 'gamestop_revenue' is not defined

<Figure size 1000x500 with 0 Axes>
```

```
tables = soup.find_all("table")
for table in tables:
    if "GameStop Quarterly Revenue" in str(table):
        gamestop_revenue = pd.read_html(str(table))[0]
        break
```

```
Start coding or generate with AI.
```

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")

tables = soup.find_all("table")
gamestop_revenue = None  # Initialize gamestop_revenue to None

for table in tables:
    if "GameStop Quarterly Revenue" in str(table):
        gamestop_revenue = pd.read_html(str(table))[0]
        break

# Check if gamestop_revenue was assigned a value
if gamestop_revenue is not None:
```

```
        plt.figure(figsize=(10, 5))
        plt.plot(pd.to_datetime(gamestop_revenue["Date"]), gamestop_revenue["Revenue"], label="GameStop Revenue", color='purple')
        plt.xlabel("Date")
        plt.ylabel("Revenue (Million $)")
        plt.title("GameStop Revenue Over Time")
        plt.legend()
        plt.show()
    else:
        print("Table with 'GameStop Quarterly Revenue' not found on the page.")
```

```
Table with 'GameStop Quarterly Revenue' not found on the page.
```

Start coding or generate with AI.