Optional: Gamestop stock vs Tesla

Determining the price of a stock is complex; it depends on the number of outstanding shares, the size of the company's future profits, and much more. An essential factor is the company's profit and growth of profits; if the company's profit is increasing, the stock price should increase. If you suspect the company's profit increases, you should buy the stock as the stock should increase, But what happens if you think the stock price will decrease.

Short selling is how you make money if the stock decreases. An investor borrows a stock, sells the stock, and then repurchases it to return it to the lender. Typically stocks fall faster than they rise, so you can make a profit more quickly. Usually, experienced investors such as hedge funds partake in short selling. One problem is if the stock price increases, the investor can lose money.

Sometimes short sellers get it wrong; for example, Tesla. A few years ago, many short sellers targeted Tesla. Then Tesla started becoming profitable, and profits were increasing; thus, the company stock went up. This was based on the companies performance, so the stock should continue to rise, and the short seller should sell the stock. Recently shorted stocks can increase for reasons that are not based on fundamentals; this is less sustainable.

Individual investors using the forum on the Reddit online community named WallStreetBets, started buying into shares of GameStop, a video and computer-game retailer losing money. The influx of demand caused GameStop shares to

Individual investors using the forum on the Reddit online community named WallStreetBets, started buying into shares of GameStop, a video and computer-game retailer losing money. The influx of demand caused GameStop shares to soar. All this produced billions of dollars in losses for hedge funds who had sold the stock short. [1] GameStop's share price should fall eventually, so the Hedge funds should hold on to the short positions. As a data scientist working for a hedge fund, you will extract the profit data for Tesla and GameStop and build a dashboard to compare the price of the stock vs the profit for the hedge fund.

Final Assignment.ipynb

Markdown | git | Run as Pipeline | Python

Note:- If you are working in IBM Cloud Watson Studio, please replace the command for installing nbformat from `!pip install nbformat==4.2.0` to simply `!pip install nbformat`

[1]:
```
!pip install yfinance==0.1.67
!mamba install bs4==4.10.0 -y
!pip install nbformat==4.2.0
```

```
Collecting yfinance==0.1.67
  Downloading yfinance-0.1.67-py2.py3-none-any.whl (25 kB)
Requirement already satisfied: pandas>=0.24 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packa
ges (from yfinance==0.1.67) (1.3.5)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packag
es (from yfinance==0.1.67) (1.21.6)
Requirement already satisfied: requests>=2.20 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-pac
kages (from yfinance==0.1.67) (2.29.0)
Collecting multitasking>=0.0.7 (from yfinance==0.1.67)
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packag
es (from yfinance==0.1.67) (4.9.2)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/
site-packages (from pandas>=0.24->yfinance==0.1.67) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packa
ges (from pandas>=0.24->yfinance==0.1.67) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/jupyterlab/conda/envs/python/lib/python3.
7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packa
ges (from requests>=2.20->yfinance==0.1.67) (3.4)
```

Final Assignment.ipynb

Markdown | git | Run as Pipeline | Python

```
      Successfully uninstalled nbformat-5.8.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed.
This behaviour is the source of the following dependency conflicts.
jupyter-server 1.24.0 requires nbformat>=5.2.0, but you have nbformat 4.2.0 which is incompatible.
nbclient 0.7.4 requires nbformat>=5.1, but you have nbformat 4.2.0 which is incompatible.
nbconvert 7.4.0 requires nbformat>=5.1, but you have nbformat 4.2.0 which is incompatible.
Successfully installed nbformat-4.2.0
```

[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
```

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.

[ ]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "Histo
    stock_data_specific = stock_data[stock_data.Date <= '2021--06-14']
```

Final Assignment.ipynb

Markdown git Run as Pipeline Python

[ ]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "Histo
    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=stoc
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=re
    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_rangeslider_visible=True)
    fig.show()
```

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`.

Final Assignment.ipynb

Markdown git Run as Pipeline Python

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`.

[4]:
```
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=tesla.history(period='max')
```

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.

[8]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

[8]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits

Final Assignment.ipynb

Markdown git Run as Pipeline Python

[8]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29	1.266667	1.666667	1.169333	1.592667	281494500	0	0.0
1	2010-06-30	1.719333	2.028000	1.553333	1.588667	257806500	0	0.0
2	2010-07-01	1.666667	1.728000	1.351333	1.464000	123282000	0	0.0
3	2010-07-02	1.533333	1.540000	1.247333	1.280000	77097000	0	0.0
4	2010-07-06	1.333333	1.333333	1.055333	1.074000	103003500	0	0.0

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`.

[ ]:

Parse the html data using `beautiful_soup`.

[ ]:

Final Assignment.ipynb WebScraping_Review_Lab.ip

Markdown git Run as Pipeline Python

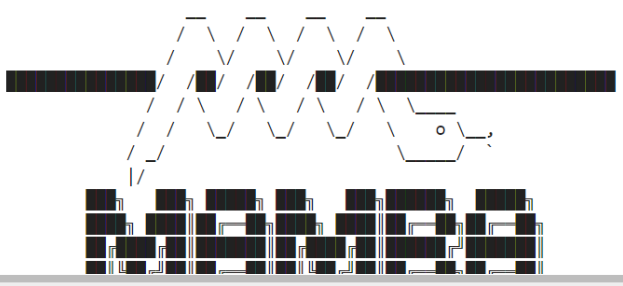
Question 2: Use Webscraping to Extract Tesla Revenue Data

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`.

[23]:

```
!pip install lxml==4.6.4
!mamba install html5lib==1.1 -y
url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-
html_data=requests.get(url).text
```

```
Requirement already satisfied: lxml==4.6.4 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packag
es (4.6.4)
```



Final Assignment.ipynb | WebScraping_Review_Lab.ip

Markdown | git | Run as Pipeline | Python

```
        mamba (1.4.2) supported by @QuantStack

        GitHub:  https://github.com/mamba-org/mamba
        Twitter: https://twitter.com/QuantStack

[redacted]


Looking for: ['html5lib==1.1']

pkgs/main/linux-64                                          Using cache
pkgs/main/noarch                                            Using cache
pkgs/r/linux-64                                             Using cache
pkgs/r/noarch                                               Using cache

Pinned packages:
  - python 3.7.*


Transaction

  Prefix: /home/jupyterlab/conda/envs/python

  All requested packages already installed
```

Parse the html data using `beautiful_soup`.

Final Assignment.ipynb | WebScraping_Review_Lab.ip

Markdown | git | Run as Pipeline | Python

[25]:
```
soup=BeautifulSoup(html_data,'html.parser')
```

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`.

▶ Click here if you need help locating the table

[36]:
```
tables=soup.find_all('table')
len(tables)

for index,table in enumerate(tables):
    if ("Tesla Quarterly Revenue" in str(table)):
        table_index = index
print(table_index)

tesla_revenue = pd.DataFrame(columns=["Date", "Revenue"])

for row in tables[table_index].tbody.find_all("tr"):
    col = row.find_all("td")
    if (col != []):
        date = col[0].text
        revenue = col[1].text
        tesla_revenue = tesla_revenue.append({"Date":date, "Revenue":revenue}, ignore_index=True)

tesla_revenue
```

1

Final Assignment.ipynb | WebScraping_Review_Lab.i

Markdown | git | Run as Pipeline | Python

1

[36]:

	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

Final Assignment.ipynb | WebScraping_Review_Lab.i

Markdown | git | Run as Pipeline | Python

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

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

Final Assignment.ipynb | WebScraping_Review_Lab.ip

Markdown | git | Run as Pipeline | Python

[37]:
```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: The default value of regex will change from True to False in a future version.
  """Entry point for launching an IPython kernel.
```

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

[38]:
```
tesla_revenue.dropna(inplace=True)

tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
```

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

[39]:
```
tesla_revenue.tail()
```

[39]:

	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

Final Assignment.ipynb

Markdown | git | Run as Pipeline | Python

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`.

[33]:
```
gamestop=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.

[34]:
```
gme_data=gamestop.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.

[35]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

[35]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits

Final Assignment.ipynb

Markdown git Run as Pipeline Python

[35]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	1.620128	1.693350	1.603296	1.691667	76216000	0.0	0.0
1	2002-02-14	1.712707	1.716073	1.670626	1.683250	11021600	0.0	0.0
2	2002-02-15	1.683251	1.687459	1.658002	1.674834	8389600	0.0	0.0
3	2002-02-19	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20	1.615920	1.662210	1.603296	1.662210	6892800	0.0	0.0

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`.

[36]:
```
url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-
html_data=requests.get(url).text
```

Parse the html data using `beautiful_soup`.

[37]:
```
soup=BeautifulSoup(html_data, 'html.parser')
```

Final Assignment.ipynb

Markdown git Run as Pipeline Python

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 using a method similar to what you did in Question 2.

▶ Click here if you need help locating the table

[38]:
```
tables=soup.find_all('table')

for index,table in enumerate(tables):
    if ("GameStop Quarterly Revenue" in str(table)):
        table_index = index
print(table_index)

GameStop_revenue = pd.DataFrame(columns=["Date", "Revenue"])

for row in tables[table_index].tbody.find_all("tr"):
    col = row.find_all("td")
    if (col != []):
        date = col[0].text
        revenue = col[1].text
        GameStop_revenue = GameStop_revenue.append({"Date":date, "Revenue":revenue}, ignore_index=True)

GameStop_revenue["Revenue"] = GameStop_revenue['Revenue'].str.replace(',|\$',"")
GameStop_revenue.dropna(inplace=True)
GameStop_revenue = GameStop_revenue[GameStop_revenue['Revenue'] != ""]
```

Final Assignment.ipynb

Markdown | git | Run as Pipeline | Python

```
GameStop_revenue
```

```
1
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launcher.py:17: FutureWarning:

The default value of regex will change from True to False in a future version.
```

[38]:

	Date	Revenue
0	2020-04-30	1021
1	2020-01-31	2194
2	2019-10-31	1439
3	2019-07-31	1286
4	2019-04-30	1548
...	...	...
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

Final Assignment.ipynb

Markdown | git | Run as Pipeline | Python

62 rows × 2 columns

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

[39]:
```
GameStop_revenue.tail()
```

[39]:

	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

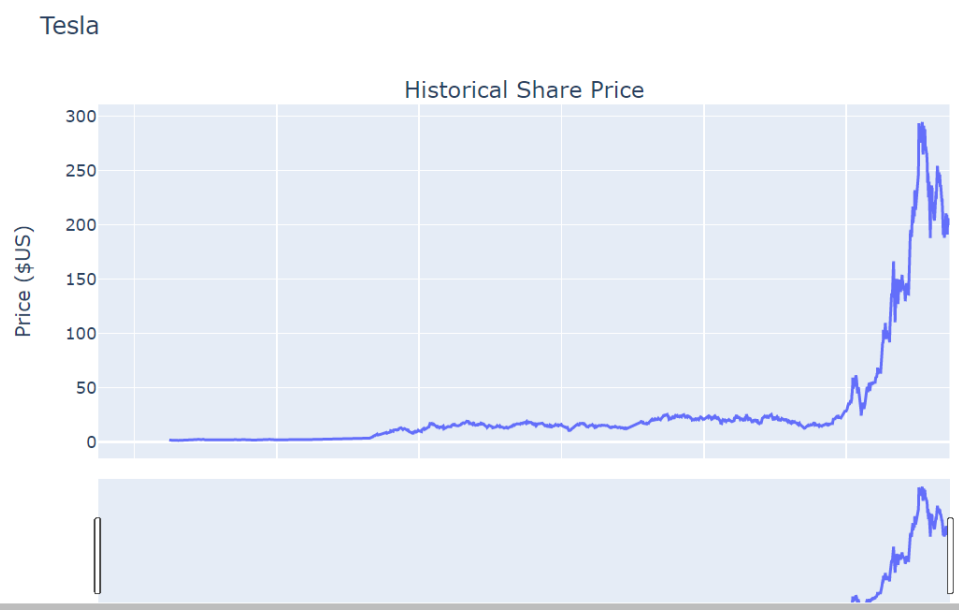
Question 5: Plot Tesla Stock Graph

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

Final Assignment.ipynb

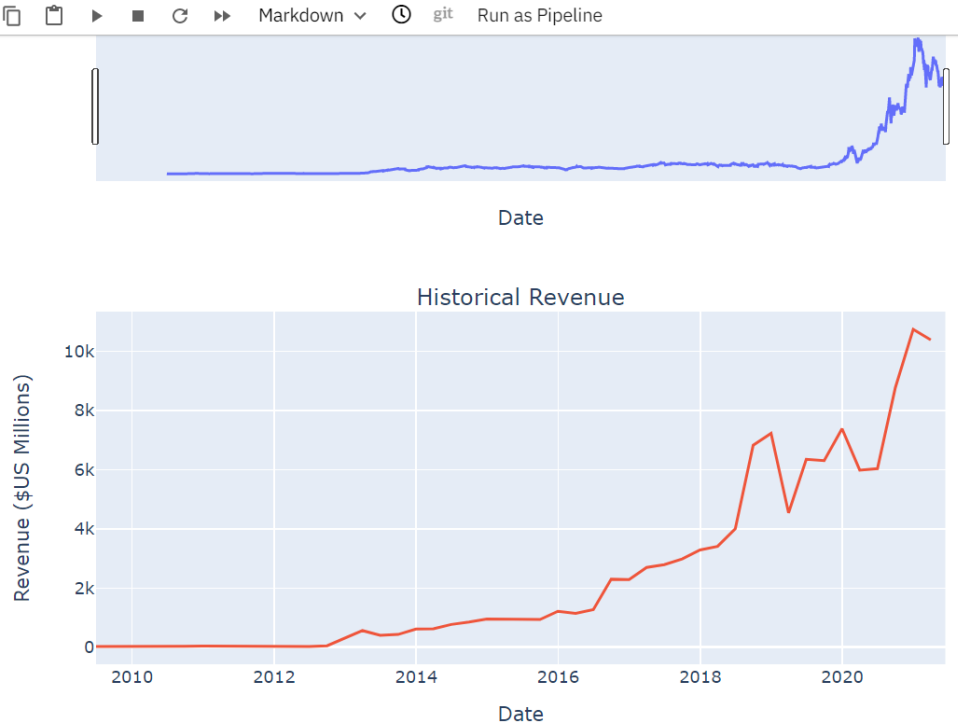
Markdown git Run as Pipeline Python

[28]:
```
make_graph(tesla_data,tesla_revenue,'Tesla')
```



Final Assignment.ipynb

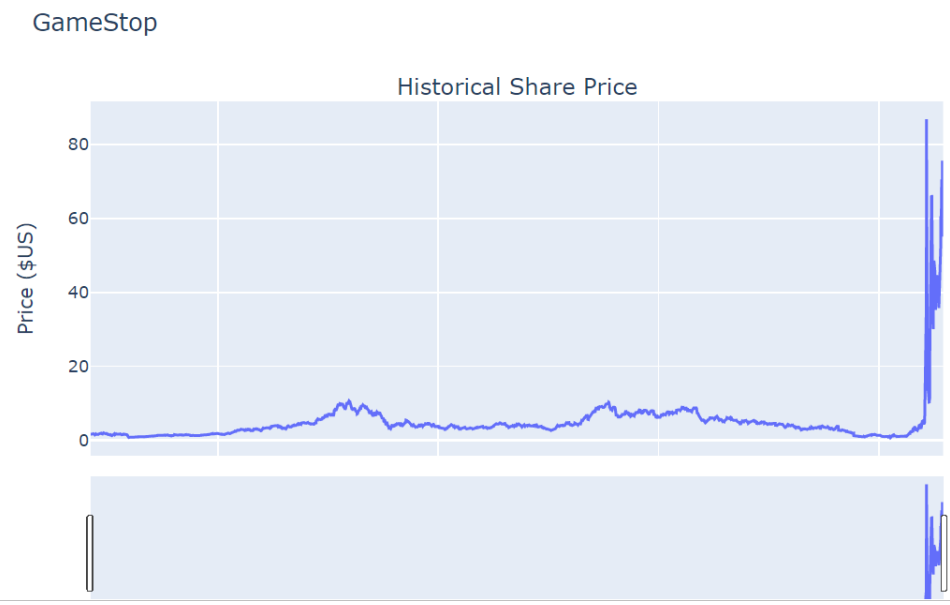
Markdown git Run as Pipeline Python



Final Assignment.ipynb

Markdown git Run as Pipeline Python

[40]:
```
make_graph(gme_data,GameStop_revenue,'GameStop')
```



Final Assignment.ipynb

Markdown git Run as Pipeline Python