Analyzing Historical Stock/Revenue Data and Building a Dashboard

In [1]:
```
!pip install yfinance
#!pip install pandas
#!pip install requests
!pip install bs4
#!pip install plotly
```

```
Collecting yfinance
  Downloading yfinance-0.2.31-py2.py3-none-any.whl (65 kB)
Collecting lxml>=4.9.1
  Using cached lxml-4.9.3-cp39-cp39-win_amd64.whl (3.9 MB)
Collecting multitasking>=0.0.7
  Using cached multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Collecting frozendict>=2.3.4
  Using cached frozendict-2.3.8-cp39-cp39-win_amd64.whl (35 kB)
Collecting pytz>=2022.5
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
Requirement already satisfied: pandas>=1.3.0 in c:\users\nitish singh\anaconda3\lib\site-packages (from yfinanc
e) (1.4.2)
Collecting html5lib>=1.1
  Using cached html5lib-1.1-py2.py3-none-any.whl (112 kB)
Requirement already satisfied: appdirs>=1.4.4 in c:\users\nitish singh\anaconda3\lib\site-packages (from yfinan
ce) (1.4.4)
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.
conda-repo-cli 1.0.4 requires pathlib, which is not installed.
anaconda-project 0.10.2 requires ruamel-yaml, which is not installed.
```

```
Collecting peewee>=3.16.2
  Downloading peewee-3.16.3.tar.gz (928 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
    Preparing wheel metadata: started
    Preparing wheel metadata: finished with status 'done'
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\users\nitish singh\anaconda3\lib\site-packages (fro
m yfinance) (4.11.1)
Requirement already satisfied: numpy>=1.16.5 in c:\users\nitish singh\anaconda3\lib\site-packages (from yfinanc
e) (1.21.5)
Collecting requests>=2.31
  Using cached requests-2.31.0-py3-none-any.whl (62 kB)
Requirement already satisfied: soupsieve>1.2 in c:\users\nitish singh\anaconda3\lib\site-packages (from beautif
ulsoup4>=4.11.1->yfinance) (2.3.1)
Requirement already satisfied: six>=1.9 in c:\users\nitish singh\anaconda3\lib\site-packages (from html5lib>=1.
1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in c:\users\nitish singh\anaconda3\lib\site-packages (from html5lib
>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\nitish singh\anaconda3\lib\site-packages (fro
m pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nitish singh\anaconda3\lib\site-packages (from re
quests>=2.31->yfinance) (2021.10.8)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\nitish singh\anaconda3\lib\site-packages (from re
quests>=2.31->yfinance) (1.26.9)
Requirement already satisfied: idna<4,>=2.5 in c:\users\nitish singh\anaconda3\lib\site-packages (from requests
>=2.31->yfinance) (3.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\nitish singh\anaconda3\lib\site-packages (f
rom requests>=2.31->yfinance) (2.0.4)
Building wheels for collected packages: peewee
  Building wheel for peewee (PEP 517): started
  Building wheel for peewee (PEP 517): finished with status 'done'
  Created wheel for peewee: filename=peewee-3.16.3-py3-none-any.whl size=135547 sha256=4ddc1b4dedef02e756b8d7c1
392837bdcb8e727012e5a19ec5436e94451bae6a
  Stored in directory: c:\users\nitish singh\appdata\local\pip\cache\wheels\46\c8\76\da8e4e28595d45fac70db74239
0350ab43211d5ae4b0eb376c
Successfully built peewee
Installing collected packages: pytz, requests, peewee, multitasking, lxml, html5lib, frozendict, yfinance
  Attempting uninstall: pytz
    Found existing installation: pytz 2021.3
    Uninstalling pytz-2021.3:
      Successfully uninstalled pytz-2021.3
  Attempting uninstall: requests
    Found existing installation: requests 2.27.1
    Uninstalling requests-2.27.1:
      Successfully uninstalled requests-2.27.1
  Attempting uninstall: lxml
    Found existing installation: lxml 4.8.0
    Uninstalling lxml-4.8.0:
      Successfully uninstalled lxml-4.8.0
Successfully installed frozendict-2.3.8 html5lib-1.1 lxml-4.9.3 multitasking-0.0.11 peewee-3.16.3 pytz-2023.3.p
ost1 requests-2.31.0 yfinance-0.2.31
Collecting bs4
  Using cached bs4-0.0.1-py3-none-any.whl
Requirement already satisfied: beautifulsoup4 in c:\users\nitish singh\anaconda3\lib\site-packages (from bs4) (
4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\nitish singh\anaconda3\lib\site-packages (from beautif
ulsoup4->bs4) (2.3.1)
Installing collected packages: bs4
Successfully installed bs4-0.0.1
```

In [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 [3]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "Historica
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_format=True), y=stock_data.Close.
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_format=True), y=revenue_data.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

In [4]:
```
#1
tesla = yf.Ticker("TSLA")

#2
tesla_data = tesla.history(period="max")

#3
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

Out[4]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
1	2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2	2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
3	2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
4	2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0

Question 2: Use Webscraping to Extract Tesla Revenue Data

In [5]:
```
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
html_data = requests.get(url).text
```

In [6]:
```
soup = BeautifulSoup(html_data, "html.parser")
soup.find_all('title')
```

Out[6]:
```
[<title>403 Forbidden</title>]
```

In [7]:
```
tsla = yf.Ticker("TSLA")

# Use the history() function to get historical stock data
historical_data = tsla.history(period="max")

# Print the historical data
print(historical_data)
tsla_data = pd.DataFrame(historical_data)
tsla_data.reset_index(inplace=True)
print(tsla_data.head())

tsla_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220
data = requests.get(tsla_url).text
soup = BeautifulSoup(data, 'xml')

tsla_dict = {}
tsla_dict["Date"]=[]
tsla_dict["Revenue"]=[]

table = soup.select("table")[1]
rows = table.select("tbody > tr")
for row in rows:
    col = row.find_all("td")
    if len(col) >= 2:
        tsla_date = col[0].text
        tsla_revenue_ = col[1].text
        tsla_dict["Date"].append(tsla_date)
        tsla_dict["Revenue"].append(tsla_revenue_)

#print(tsla_dict)
tsla_revenue = pd.DataFrame(tsla_dict)
print(tsla_revenue.head())
tsla_revenue["Revenue"] = tsla_revenue['Revenue'].str.replace(',|\$',"")
tsla_revenue.dropna(inplace=True)

tsla_revenue = tsla_revenue[tsla_revenue['Revenue'] != ""]
print(tsla_revenue.head())
```

```
                                 Open        High         Low       Close  \
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
...                               ...         ...         ...         ...
2023-10-04 00:00:00-04:00  248.139999  261.859985  247.600006  261.160004
2023-10-05 00:00:00-04:00  260.000000  263.600006  256.250000  260.049988
2023-10-06 00:00:00-04:00  253.979996  261.649994  250.649994  260.529999
2023-10-09 00:00:00-04:00  255.309998  261.359985  252.050003  259.670013
2023-10-10 00:00:00-04:00  257.750000  268.940002  257.649994  267.450012

                              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
...                              ...        ...           ...
2023-10-04 00:00:00-04:00  129721600        0.0           0.0
2023-10-05 00:00:00-04:00  119159200        0.0           0.0
2023-10-06 00:00:00-04:00  117947000        0.0           0.0
2023-10-09 00:00:00-04:00  101156100        0.0           0.0
2023-10-10 00:00:00-04:00   63699665        0.0           0.0

[3344 rows x 7 columns]
                       Date      Open      High       Low     Close  \
0 2010-06-29 00:00:00-04:00  1.266667  1.666667  1.169333  1.592667
1 2010-06-30 00:00:00-04:00  1.719333  2.028000  1.553333  1.588667
2 2010-07-01 00:00:00-04:00  1.666667  1.728000  1.351333  1.464000
3 2010-07-02 00:00:00-04:00  1.533333  1.540000  1.247333  1.280000
4 2010-07-06 00:00:00-04:00  1.333333  1.333333  1.055333  1.074000

      Volume  Dividends  Stock Splits
0  281494500        0.0           0.0
1  257806500        0.0           0.0
2  123282000        0.0           0.0
3   77097000        0.0           0.0
4  103003500        0.0           0.0
         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
         Date Revenue
0  2022-09-30   21454
1  2022-06-30   16934
2  2022-03-31   18756
3  2021-12-31   17719
4  2021-09-30   13757
C:\Users\NITISH SINGH\AppData\Local\Temp\ipykernel_19876\3721265835.py:33: FutureWarning: The default value of
regex will change from True to False in a future version.
  tsla_revenue["Revenue"] = tsla_revenue['Revenue'].str.replace(',|\$',"")
```

In [8]:
```
print(tsla_revenue.tail())
```

```
          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
```

Question 3: Use yfinance to Extract Stock Data

In [9]:
```
gamestop = yf.Ticker("GME")
```

In [10]:
```
gme_data=gamestop.history(period="max")
```

In [11]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

Out[11]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620129	1.693350	1.603296	1.691667	76216000	0.0	0.0
1	2002-02-14 00:00:00-05:00	1.712707	1.716074	1.670626	1.683250	11021600	0.0	0.0
2	2002-02-15 00:00:00-05:00	1.683251	1.687459	1.658002	1.674834	8389600	0.0	0.0
3	2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20 00:00:00-05:00	1.615920	1.662209	1.603296	1.662209	6892800	0.0	0.0

Question 4: Use Webscraping to Extract GME Revenue Data

In [12]:

```
gamestop = yf.Ticker("GME")

# Use the history() function to get historical stock data
historical_data_g = gamestop.history(period="max")

# Print the historical data
print(historical_data)
gme_data = pd.DataFrame(historical_data)
gme_data.reset_index(inplace=True)
print(gme_data.head())

gme_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220E
data_1 = requests.get(gme_url).text
soup_1 = BeautifulSoup(data_1, "html.parser")

gme_dict = {}
gme_dict["Date"] = []
gme_dict["Revenue"] = []

table_1 = soup_1.select("table")[1]
rows_1 = table_1.select("tbody > tr")

for row_1 in rows_1:
    col_1 = row_1.find_all("td")
    gme_date = col_1[0].text
    gme_revenue_ = col_1[1].text
    gme_dict["Date"].append(gme_date)
    gme_dict["Revenue"].append(gme_revenue_)

print(gme_dict)
gme_revenue = pd.DataFrame(gme_dict)
print(gme_revenue.head())
gme_revenue["Revenue"] = gme_revenue['Revenue'].str.replace(',|\$',"")
gme_revenue.dropna(inplace=True)

gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
print(gme_revenue.head())
```

```
                         Date        Open        High         Low  \
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
...                        ...         ...         ...         ...
3339 2023-10-04 00:00:00-04:00  248.139999  261.859985  247.600006
3340 2023-10-05 00:00:00-04:00  260.000000  263.600006  256.250000
3341 2023-10-06 00:00:00-04:00  253.979996  261.649994  250.649994
3342 2023-10-09 00:00:00-04:00  255.309998  261.359985  252.050003
3343 2023-10-10 00:00:00-04:00  257.750000  268.940002  257.649994

          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
...         ...        ...        ...           ...
3339  261.160004  129721600        0.0           0.0
3340  260.049988  119159200        0.0           0.0
3341  260.529999  117947000        0.0           0.0
3342  259.670013  101156100        0.0           0.0
3343  267.450012   63699665        0.0           0.0

[3344 rows x 8 columns]
   index                       Date      Open      High       Low     Close  \
0      0 2010-06-29 00:00:00-04:00  1.266667  1.666667  1.169333  1.592667
1      1 2010-06-30 00:00:00-04:00  1.719333  2.028000  1.553333  1.588667
2      2 2010-07-01 00:00:00-04:00  1.666667  1.728000  1.351333  1.464000
3      3 2010-07-02 00:00:00-04:00  1.533333  1.540000  1.247333  1.280000
4      4 2010-07-06 00:00:00-04:00  1.333333  1.333333  1.055333  1.074000

      Volume  Dividends  Stock Splits
0  281494500        0.0           0.0
1  257806500        0.0           0.0
2  123282000        0.0           0.0
3   77097000        0.0           0.0
4  103003500        0.0           0.0
{'Date': ['2020-04-30', '2020-01-31', '2019-10-31', '2019-07-31', '2019-04-30', '2019-01-31', '2018-10-31', '20
18-07-31', '2018-04-30', '2018-01-31', '2017-10-31', '2017-07-31', '2017-04-30', '2017-01-31', '2016-10-31', '2
016-07-31', '2016-04-30', '2016-01-31', '2015-10-31', '2015-07-31', '2015-04-30', '2015-01-31', '2014-10-31', '
2014-07-31', '2014-04-30', '2014-01-31', '2013-10-31', '2013-07-31', '2013-04-30', '2013-01-31', '2012-10-31',
'2012-07-31', '2012-04-30', '2012-01-31', '2011-10-31', '2011-07-31', '2011-04-30', '2011-01-31', '2010-10-31',
'2010-07-31', '2010-04-30', '2010-01-31', '2009-10-31', '2009-07-31', '2009-04-30', '2009-01-31', '2008-10-31',
'2008-07-31', '2008-04-30', '2008-01-31', '2007-10-31', '2007-07-31', '2007-04-30', '2007-01-31', '2006-10-31',
'2006-07-31', '2006-04-30', '2006-01-31', '2005-10-31', '2005-07-31', '2005-04-30', '2005-01-31'], 'Revenue': [
'$1,021', '$2,194', '$1,439', '$1,286', '$1,548', '$3,063', '$1,935', '$1,501', '$1,786', '$2,825', '$1,989', '
$1,688', '$2,046', '$2,403', '$1,959', '$1,632', '$1,972', '$3,525', '$2,016', '$1,762', '$2,061', '$3,476', '$
2,092', '$1,731', '$1,996', '$3,684', '$2,107', '$1,384', '$1,865', '$3,562', '$1,773', '$1,550', '$2,002', '$3
,579', '$1,947', '$1,744', '$2,281', '$3,693', '$1,899', '$1,799', '$2,083', '$3,524', '$1,835', '$1,739', '$1,
981', '$3,492', '$1,696', '$1,804', '$1,814', '$2,866', '$1,611', '$1,338', '$1,279', '$2,304', '$1,012', '$963
', '$1,040', '$1,667', '$534', '$416', '$475', '$709']}
         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
         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
C:\Users\NITISH SINGH\AppData\Local\Temp\ipykernel_19876\359356365.py:33: FutureWarning: The default value of r
egex will change from True to False in a future version.
  gme_revenue["Revenue"] = gme_revenue['Revenue'].str.replace(',|\$',"")
```

In [15]:
```
print(gme_revenue.tail())
```

```
          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
```

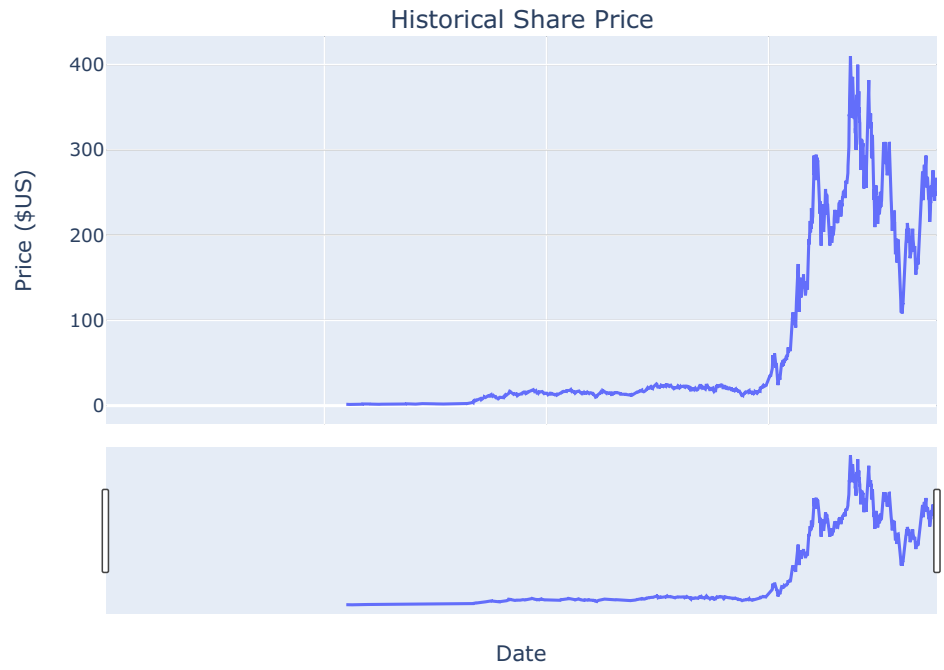
Question 5: Plot Stock Graphs

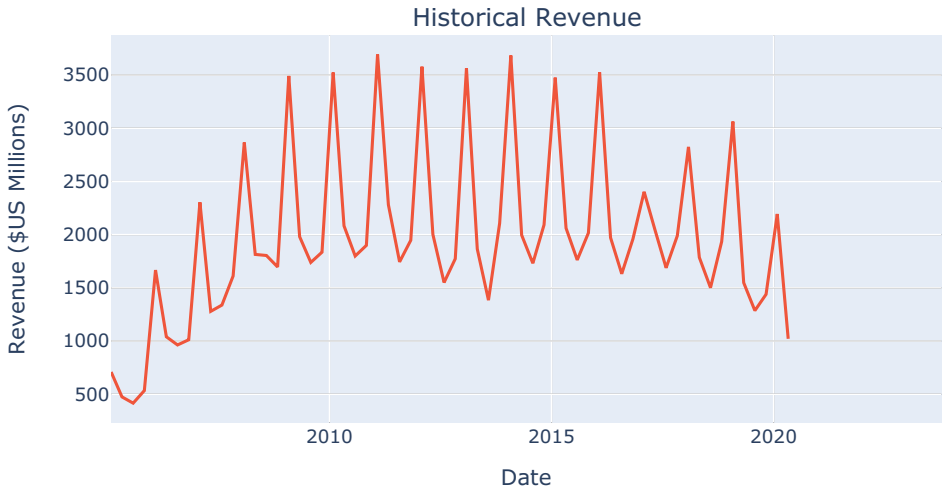
In [18]:
```
#1
make_graph(tesla_data, tsla_revenue, 'Tesla')
```

Question 6: Plot GameStop Stock Graph

In [17]:
```
make_graph(gme_data, gme_revenue, 'GameStop')
```

GameStop





In [ ]: