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`.

In [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.

In [6]:
```
tesla_data = tesla.history(period = 'max')
tesla_data.head()
```

Out[6]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2010-06-29	3.800	5.000	3.508	4.778	93831500	0	0.0
2010-06-30	5.158	6.084	4.660	4.766	85935500	0	0.0
2010-07-01	5.000	5.184	4.054	4.392	41094000	0	0.0
2010-07-02	4.600	4.620	3.742	3.840	25699000	0	0.0
2010-07-06	4.000	4.000	3.166	3.222	34334500	0	0.0

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.

In [7]:
```
tesla_data.reset_index(inplace = True)
tesla_data.head()
```

Out[7]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29	3.800	5.000	3.508	4.778	93831500	0	0.0
1	2010-06-30	5.158	6.084	4.660	4.766	85935500	0	0.0
2	2010-07-01	5.000	5.184	4.054	4.392	41094000	0	0.0
3	2010-07-02	4.600	4.620	3.742	3.840	25699000	0	0.0
4	2010-07-06	4.000	4.000	3.166	3.222	34334500	0	0.0

Question 2: Use Webscraping to Extract Tesla Revenue Data

Use the `requests` library to download the webpage https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue. Save the text of the response as a variable named `html_data`.

In [57]:
```
html_data = requests.get(' https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue').text
```

Parse the html data using `beautiful_soup`.

In [58]:
```
soup = BeautifulSoup(html_data, 'html.parser')
```

Using beautiful soup extract the table with `Tesla Quarterly Revenue` and store it into a dataframe named `tesla_revenue`. The dataframe should have columns `Date` and `Revenue`. Make sure the comma and dollar sign is removed from the `Revenue` column.

In [59]:
```
tables = soup.find_all('table')

for index,table in enumerate(tables):
    if ("Tesla Quarterly Revenue" in str(table)):
        table_index = 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.strip()
        revenue = col[1].text.strip('$').replace(',','')

        tesla_revenue = tesla_revenue.append({"Date":date, "Revenue":revenue}, ignore_index=True)
```

Click here if you need help removing the dollar sign and comma

Remove the rows in the dataframe that are empty strings or are NaN in the Revenue column. Print the entire `tesla_revenue` DataFrame to see if you have any.

Remove the rows in the dataframe that are empty strings or are NaN in the Revenue column. Print the entire `tesla_revenue` DataFrame to see if you have any.

In [60]:
```
tesla_revenue.dropna(inplace=True)
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
tesla_revenue
```

Out[60]:

	Date	Revenue
0	2021-03-31	10389
1	2020-12-31	10744
2	2020-09-30	8771
3	2020-06-30	6036
4	2020-03-31	5985
5	2019-12-31	7384
6	2019-09-30	6303
7	2019-06-30	6350
8	2019-03-31	4541
9	2018-12-31	7226
10	2018-09-30	6824
11	2018-06-30	4002
12	2018-03-31	3409
13	2017-12-31	3288
14	2017-09-30	2985
15	2017-06-30	2790
16	2017-03-31	2696
17	2016-12-31	2285
18	2016-09-30	2298
19	2016-06-30	1270
20	2016-03-31	1147

20	2016-03-31	1147
21	2015-12-31	1214
22	2015-09-30	937
23	2015-06-30	955
24	2015-03-31	940
25	2014-12-31	957
26	2014-09-30	852
27	2014-06-30	769
28	2014-03-31	621
29	2013-12-31	615
30	2013-09-30	431
31	2013-06-30	405
32	2013-03-31	562
33	2012-12-31	306
34	2012-09-30	50
35	2012-06-30	27
36	2012-03-31	30
37	2011-12-31	39
38	2011-09-30	58
39	2011-06-30	58
40	2011-03-31	49
41	2010-12-31	36
42	2010-09-30	31
43	2010-06-30	28
44	2010-03-31	21
46	2009-09-30	46
47	2009-06-30	27

47	2009-06-30	27

Click here if you need help removing the Nan or empty strings

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

In [61]:
```
tesla_revenue.tail()
```

Out[61]:

	Date	Revenue
42	2010-09-30	31
43	2010-06-30	28
44	2010-03-31	21
46	2009-09-30	46
47	2009-06-30	27

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`.

In [62]:
```
game_stop = 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.

In [63]:
```
gme_data = game_stop.history(period = 'max')
gme_data
```

Out[63]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2002-02-13	6.480513	6.773399	6.413183	6.766666	19054000	0.0	0.0
2002-02-14	6.850831	6.864296	6.682506	6.733003	2755400	0.0	0.0
2002-02-15	6.733001	6.749833	6.632006	6.699336	2097400	0.0	0.0
2002-02-19	6.665671	6.665671	6.312189	6.430017	1852600	0.0	0.0
2002-02-20	6.463681	6.648838	6.413183	6.648838	1723200	0.0	0.0
...	...	...	...	...	...	...	...
2021-05-04	159.000000	161.490005	151.800003	160.729996	4007500	0.0	0.0
2021-05-05	161.830002	165.500000	158.330002	159.479996	1789200	0.0	0.0
2021-05-06	160.860001	164.720001	155.600006	161.009995	2942800	0.0	0.0
2021-05-07	160.110001	167.410004	157.500000	161.110001	2930200	0.0	0.0
2021-05-10	161.309998	161.610001	144.407806	145.300003	3597090	0.0	0.0

4843 rows × 7 columns

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.

In [64]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

Out[64]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	6.480513	6.773399	6.413183	6.766666	19054000	0.0	0.0
1	2002-02-14	6.850831	6.864296	6.682506	6.733003	2755400	0.0	0.0
2	2002-02-15	6.733001	6.749833	6.632006	6.699336	2097400	0.0	0.0
3	2002-02-19	6.665671	6.665671	6.312189	6.430017	1852600	0.0	0.0
4	2002-02-20	6.463681	6.648838	6.413183	6.648838	1723200	0.0	0.0

Question 4: Use Webscraping to Extract GME Revenue Data

Use the `requests` library to download the webpage https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue. Save the text of the response as a variable named `html_data`.

In [65]:
```
html_data = requests.get('https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue').text
```

Parse the html data using `beautiful_soup`.

In [66]:
```
soup = BeautifulSoup(html_data, 'html.parser')
```

Using beautiful soup extract the table with `GameStop Quarterly 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.

In [67]:
```
tables = soup.find_all('table')

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

gme_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.strip()
        revenue = col[1].text.strip('$').replace(',','')

        gme_revenue = gme_revenue.append({"Date":date, "Revenue":revenue}, ignore_index=True)
```

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

In [74]:
```
gme_revenue.tail()
```

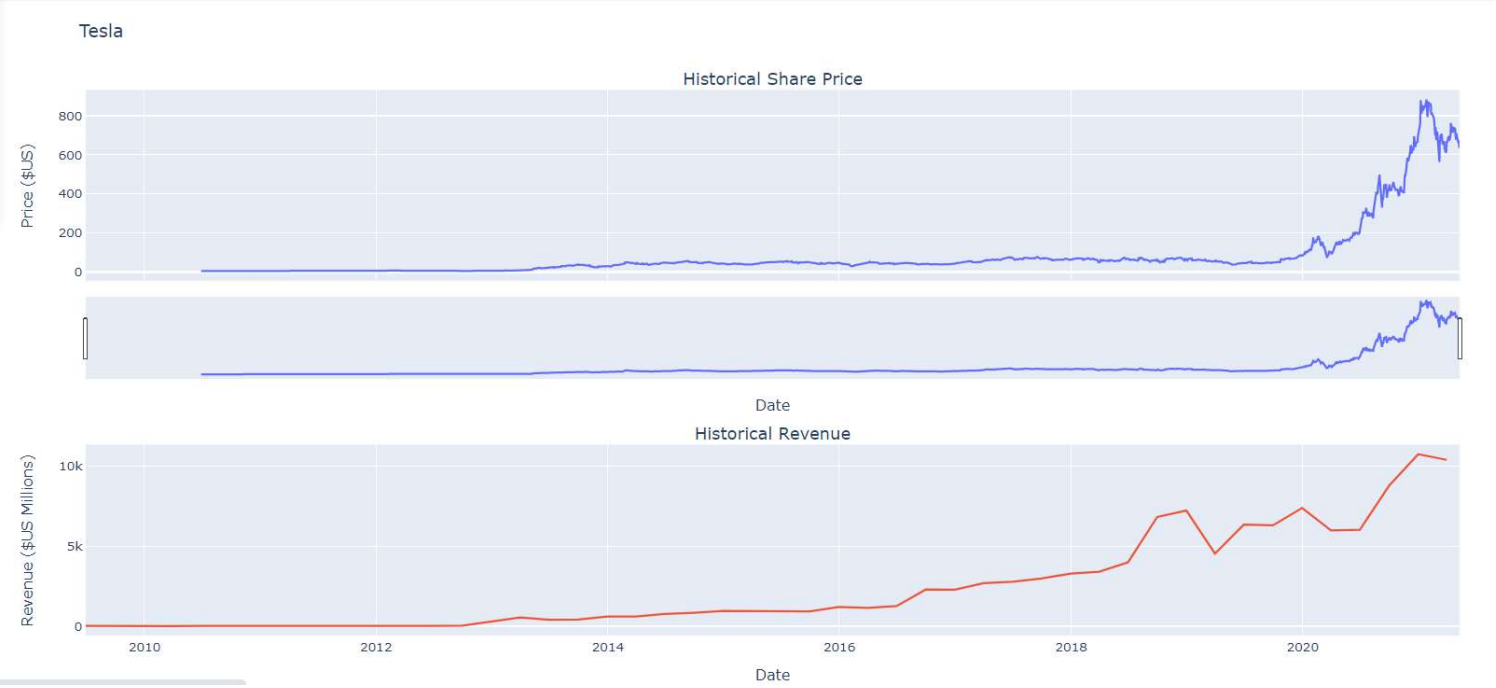
Out[74]:

	Date	Revenue
60	2006-01-31	1667
61	2005-10-31	534
62	2005-07-31	416
63	2005-04-30	475
64	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')`

In [101]:
```
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)

    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_format=True), y=stock_data.Close.astype("float"), name="Share Price"), row=1, col=1)

    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_format=True), y=revenue_data.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=700, title=stock, xaxis_rangeslider_visible=True)

    fig.show()

make_graph(tesla_data, tesla_revenue, 'Tesla')
```



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')`.

In [104]:

```
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)

    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_format=True), y=stock_data.Close.astype("float"), name="Share Price"), row=1, col=1)

    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_format=True), y=revenue_data.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=700, title=stock, xaxis_rangeslider_visible=True)

    fig.show()

make_graph(gme_data, gme_revenue, 'GameStop')
```