Analyzing Tesla's Historical Stock and Revenue Data

```
!pip install yfinance==0.1.67
!mamba install bs4==4.10.0 -y
!pip install nbformat==5.8.0

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 the function make_graph
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_rangeslider_visible=True)

        fig.show()
```

# Use Yfinance to extract Tesla's stock data
# Create a ticker object, "tesla", that takes in the stock data using the key "TSLA"
tesla = yf.Ticker("TSLA")

# Extract stock information and save it in a dataframe named tesla_data. The period parameter is set to max so we get information for the maximum amount of time.
tesla_data = tesla.history(period="max")
tesla_data

Output:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2010-06-29	1.266667	1.666667	1.169333	1.592667	281494500	0	0.0
2010-06-30	1.719333	2.028000	1.553333	1.588667	257806500	0	0.0
2010-07-01	1.666667	1.728000	1.351333	1.464000	123282000	0	0.0
2010-07-02	1.533333	1.540000	1.247333	1.280000	77097000	0	0.0
2010-07-06	1.333333	1.333333	1.055333	1.074000	103003500	0	0.0
...	...	...	...	...	...	...	...
2023-02-28	210.589996	211.229996	203.750000	205.710007	153144900	0	0.0
2023-03-01	206.210007	207.199997	198.520004	202.770004	156852800	0	0.0
2023-03-02	186.740005	193.750000	186.009995	190.899994	181500700	0	0.0
2023-03-03	194.800003	200.479996	192.880005	197.789993	153800400	0	0.0
2023-03-06	198.539993	198.567993	192.309998	193.809998	127504923	0	0.0

3193 rows × 7 columns

# Reset the index and display the first 5 rows
tesla_data.reset_index(inplace=True)
Tesla_data

Output:

	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
...	...	...	...	...	...	...	...	...
3263	2023-06-15	248.399994	258.950012	247.289993	255.899994	160171200	0	0.0
3264	2023-06-16	258.920013	263.600006	257.209991	260.540009	167563700	0	0.0
3265	2023-06-20	261.500000	274.750000	261.119995	274.450012	165611200	0	0.0
3266	2023-06-21	275.130005	276.989990	257.779999	259.459991	211797100	0	0.0
3267	2023-06-22	250.770004	264.980011	248.250000	264.609985	166007038	0	0.0

Use web scraping to extract Tesla Revenue Data¶

Download the webpage https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.html and save the text as a variable, html_data

```
import requests
from bs4 import BeautifulSoup
url =
"https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm"
html_data  = requests.get(url).text
```

```
# Parse the html data using BeautifulSoup
soup = BeautifulSoup(html_data,'html.parser')
tag_object=soup.title
print("tag object:",tag_object)
```

```
# Output:
tag object: <title>Tesla Revenue 2010-2022 | TSLA | MacroTrends</title>
```

```
# Use BeautifulSoup to extract the table with Tesla Quarterly Revenue and store it into a dataframe named tesla_revenue, with columns Date and Revenue
tesla_revenue = pd.DataFrame(columns=["Date" , "Revenue"])
for table in soup.find_all('table'):
    if table.find('th').getText().startswith("Tesla Quarterly Revenue"):
        for row in table.find("tbody").find_all("tr"):
            col = row.find_all("td")
            date = col[0].text
            revenue = col[1].text
            tesla_revenue = tesla_revenue.append({"Date":date, "Revenue":revenue},
            ignore_index=True)
tesla_revenue.head()
```

```
# Output:
```

:

	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

# Check for duplicate values
duplicates = tesla_revenue.duplicated()
print(tesla_revenue[duplicates])

Output:

```
duplicates = tesla_revenue.duplicated()
print(tesla_revenue[duplicates])
```

```
Empty DataFrame
Columns: [Date, Revenue]
Index: []
```

Remove the comma and dollar sign from the Revenue column
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")

# Remove any null or empty strings in the Revenue column
tesla_revenue.dropna(inplace=True)
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
tesla_revenue.tail(5)

Output:

	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

# Check the data type of the ‘Revenue’ column
data_type = tesla_revenue['Revenue'].dtypes
print(data_type)

Output:
object

Check for outliers in the data

import numpy as np

Convert data type of 'Revenue' from object to int
tesla_revenue['Revenue'] = tesla_revenue['Revenue'].astype(int)

# Calculate the interquartile range (IQR)
Q1 = tesla_revenue['Revenue'].quantile(0.25)
Q3 = tesla_revenue['Revenue'].quantile(0.75)

IQR = Q3 - Q1

# Determine the lower and upper bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Check for outliers
outliers = tesla_revenue[(tesla_revenue['Revenue'] < lower_bound) | (tesla_revenue['Revenue'] > upper_bound)]

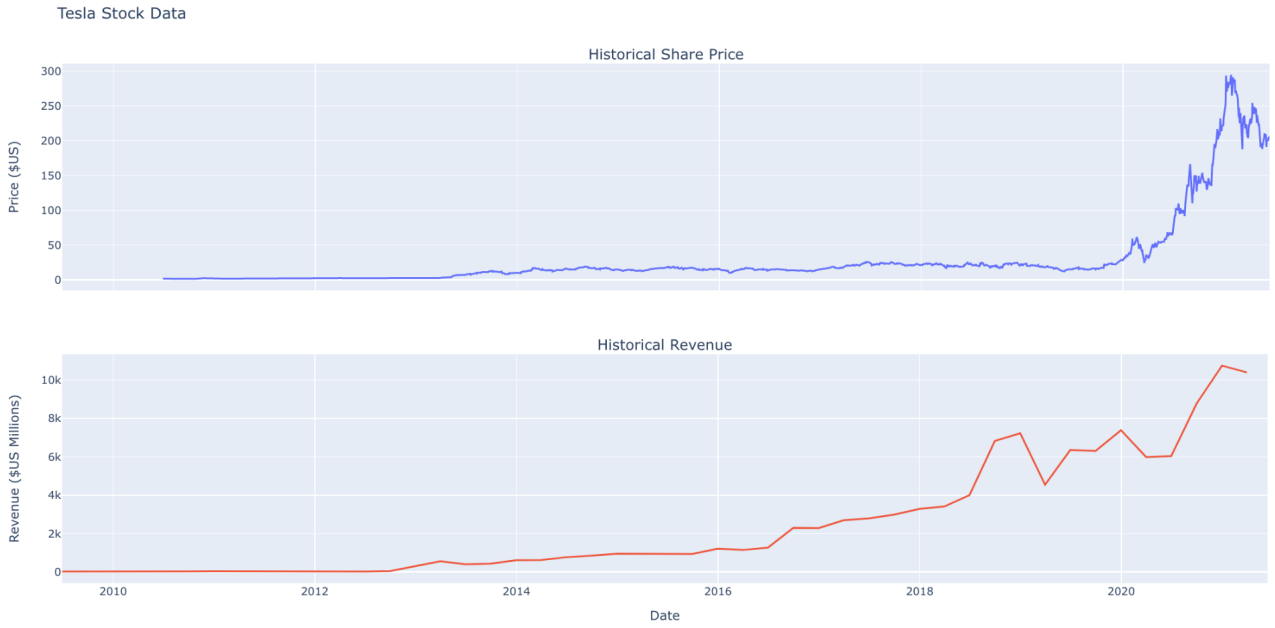
# Print the outliers
print(outliers)

Output:

	Date	Revenue
0	2022-09-30	21454
1	2022-06-30	16934
2	2022-03-31	18756
3	2021-12-31	17719

# Graph the Tesla Stock Data. Emit the data in 2022 to eliminate most of the outliers and only show data up from 2010 to June 2021
make_graph(tesla_data, tesla_revenue, 'Tesla Stock Data')

Output:



Analysis:

Tesla's Historical Share Price:

During the lockdown in the beginning of 2020, all stock prices decreased, including Teslas, since people had no desire to buy cars at the time. Shortly after the lockdown was lifted, people began to buy cars once again. However, production took a long time to recover, thereby increasing stock prices. In 2021, people started to realize that Tesla's stock price was exaggerated and overhyped, which likely caused the decline.

Tesla's Historical Revenue:

Tesla started to deliver the new Model 3 in 2017. However, they struggled to produce enough cars to meet the market demand. They began to overcome their production issues in 2018, allowing them to sell many more cars to cover the demand, which is likely the reason for the spike shown in late 2018. The government federal tax incentives ended during the 3rd and 4th quarter of 2019, which is likely the cause for the decline in revenue. However, Tesla continued to gain in popularity, which is likely the reason their revenue continued to increase following the decline of late 2019. The sharp revenue increase in mid-2020 is likely due to the end of lockdown, in which the demand came back in large, but slowed production during the lockdown caused a shortage, thereby causing prices to spike by tens of percentages.

Future focuses based on analysis:

In the future, Tesla should work on developing strong scenario planning systems and strategies to address potential production issues and slowdowns early on, such as the case when the new Model 3 came out in 2017 and Tesla struggled to produce enough cars to meet the market demand. Tesla should also prepare for potential future lockdowns or similar situations by optimizing inventory levels and production schedules, establishing a flexible and agile supply management system, and investing in collaborative, remote work, automation, and robotics.