```
Analyzing Historical Stock Revenue Data and Building a Dashboard
```

In [4]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\nitis\appdata\roaming\python\python39\site-packages (1.3.3)
Requirement already satisfied: numpy>=1.17.3 in c:\users\nitis\appdata\roaming\python\python39\site-packages (from p
andas) (1.22.0rc1)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\nitis\appdata\roaming\python\python39\site-package
s (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\nitis\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\nitis\appdata\roaming\python\python39\site-packages (from python
-dateutil>=2.7.3->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
pip install requests
```

```
Requirement already satisfied: requests in c:\users\nitis\appdata\roaming\python\python39\site-packages (2.26.0)
Requirement already satisfied: idna<4,>=2.5 in c:\users\nitis\appdata\roaming\python\python39\site-packages (from re
quests) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nitis\appdata\roaming\python\python39\site-packages (f
rom requests) (2021.10.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\nitis\appdata\roaming\python\python39\site-packages
(from requests) (1.26.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\nitis\appdata\roaming\python\python39\site-pack
ages (from requests) (2.0.8)
Note: you may need to restart the kernel to use updated packages.
```

In [6]:
```
pip install bs4
```

```
Collecting bs4
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
Requirement already satisfied: beautifulsoup4 in c:\users\nitis\anaconda3\lib\site-packages (from bs4) (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\nitis\appdata\roaming\python\python39\site-packages (from b
eautifulsoup4->bs4) (2.3.1)
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py): started
  Building wheel for bs4 (setup.py): finished with status 'done'
  Created wheel for bs4: filename=bs4-0.0.1-py3-none-any.whl size=1271 sha256=5f40690c4dd93a480be3db826d3ca5447813ab
09eb90894b0c6d0f99f3b8c8d8
Note: you may need to restart the kernel to use updated packages.  Stored in directory: c:\users\nitis\appdata\local
\pip\cache\wheels\73\2b\cb\099980278a0c9a3e57ff1a89875ec07bfa0b6fcbebb9a8cad3
Successfully built bs4
Installing collected packages: bs4
Successfully installed bs4-0.0.1
```

In [7]:
```
pip install plotly
```

```
Collecting plotly
  Downloading plotly-5.14.1-py2.py3-none-any.whl (15.3 MB)
Collecting tenacity>=6.2.0
  Downloading tenacity-8.2.2-py3-none-any.whl (24 kB)
Requirement already satisfied: packaging in c:\users\nitis\appdata\roaming\python\python39\site-packages (from plotl
y) (21.3)
Note: you may need to restart the kernel to use updated packages.Requirement already satisfied: pyparsing!=3.0.5,>=
2.0.2 in c:\users\nitis\appdata\roaming\python\python39\site-packages (from packaging->plotly) (3.0.6)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.14.1 tenacity-8.2.2
```

In [5]:
```
import yfinance as yf
import pandas as pd
import requests as r
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [9]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "Historical Rever
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_format=True), y=stock_data.Close.astype(
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_format=True), y=revenue_data.Revenue.a
    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 [10]:
```
tesla = yf.Ticker('TSLA')
```

In [11]:
```
tesla_data = tesla.history(period="max")
```

In [12]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head(5)
```

Out[12]:

	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 [13]:
```
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
html_data = requests.get(url).text
```

In [14]:
```
soup = BeautifulSoup(html_data, "html5lib")
print(soup.prettify())
```

```
<!DOCTYPE html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!-->
<html class="no-js">
 <!--<![endif]-->
 <head>
  <meta charset="utf-8"/>
  <meta content="IE=edge,chrome=1" http-equiv="X-UA-Compatible"/>
  <link href="https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue" rel="canonical"/>
  <title>
   Tesla Revenue 2010-2023 | TSLA | MacroTrends
  </title>
  <link href="/assets/images/icons/FAVICON/macrotrends-favicon.ico" rel="icon" type="image/x-icon"/>
  <meta content="Tesla annual/quarterly revenue history and growth rate from 2010 to 2023. Revenue can be defined
as the amount of money a company receives from its customers in exchange for the sales of goods or services.  Reve
nue is the top line item on an income statement from which all costs and expenses are subtracted to arrive at net
income.
```

In [15]:
```
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")
            if len(col) != 2: continue
            Date = col[0].text
            Revenue = col[1].text.replace("$","").replace(",","")

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

In [16]:
```
tesla_revenue.dropna(axis=0, how='all', subset=['Revenue']) #drop NaN values
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""] #drop empty string values
```

In [17]:
```
tesla_revenue.tail(5)
```

Out[17]:

	Date	Revenue
50	2010-09-30	31
51	2010-06-30	28
52	2010-03-31	21
54	2009-09-30	46
55	2009-06-30	27

Question 3: Use yfinance to Extract Stock Data

In [18]:
```
gme = yf.Ticker('GME')
```

In [19]:
```
gme_data = gme.history(period = "max")
```

In [20]:
```
gme_data.reset_index(inplace=True)
gme_data.head(5)
```

Out[20]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620128	1.693350	1.603296	1.691666	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	1.683250	1.687458	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	1.615921	1.662210	1.603296	1.662210	6892800	0.0	0.0

Question 4: Use Webscraping to Extract GME Revenue Data

In [6]:
```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
html_data = r.get(url).text
```

In [7]:
```
soup = BeautifulSoup(html_data, "html5lib")
```

In [8]:
```
gme_revenue = pd.DataFrame(columns = ["Date","Revenue"])

for table in soup.find_all('table'):
    if table.find('th').getText().startswith("GameStop Quarterly Revenue"):
        for row in table.find("tbody").find_all("tr"):
            col = row.find_all("td")
            if len(col) != 2: continue
            Date = col[0].text
            Revenue = col[1].text.replace("$","").replace(",","")

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

In [9]:
```
gme_revenue.tail(5)
```

Out[9]:

	Date	Revenue
52	2010-01-31	3524
53	2009-10-31	1835
54	2009-07-31	1739
55	2009-04-30	1981
56	2009-01-31	3492

Question 5: Plot Tesla Stock Graph

In [5]:
```
stock_data = yf.download("TSLA", start="2020-01-01", end="2021-09-30", progress=False)
revenue_data = yf.download("TSLA", start="2020-01-01", end="2021-09-30", progress=False)
stock_data.reset_index(inplace=True)
revenue_data.reset_index(inplace=True)

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.Volume.astype("float"),
                             name="Volume"), 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()

make_graph(stock_data, revenue_data, 'TESLA')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_17428/1194218839.py in <module>
----> 1 stock_data = yf.download("TSLA", start="2020-01-01", end="2021-09-30", progress=False)
      2 revenue_data = yf.download("TSLA", start="2020-01-01", end="2021-09-30", progress=False)
      3 stock_data.reset_index(inplace=True)
      4 revenue_data.reset_index(inplace=True)
      5

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

Question 6: Plot GameStop Stock Graph

In [3]:
```
mg = make_graph(gme_data, gme_revenue, 'GameStop')
print(mg)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_17428/3592544459.py in <module>
----> 1 mg = make_graph(gme_data, gme_revenue, 'GameStop')
      2 print(mg)

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

In [ ]:
```

```