

# Extracting and Visualizing Stock Data

```
In [2]: !pip install yfinance  
#!pip install pandas  
#!pip install requests  
!pip install bs4  
#!pip install plotly
```

Collecting yfinance  
  Downloading yfinance-0.1.63.tar.gz (26 kB)  
Requirement already satisfied: pandas>=0.24 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from yfinance) (1.2.4)  
Requirement already satisfied: numpy>=1.15 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from yfinance) (1.20.1)  
Requirement already satisfied: requests>=2.20 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from yfinance) (2.25.1)  
Collecting multitasking>=0.0.7  
  Downloading multitasking-0.0.9.tar.gz (8.1 kB)  
Requirement already satisfied: lxml>=4.5.1 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from yfinance) (4.6.3)  
Requirement already satisfied: python-dateutil>=2.7.3 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from yfinance) (2.8.1)  
Requirement already satisfied: pytz>=2017.3 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from pandas>=0.24->yfinance) (2021.1)  
Requirement already satisfied: six>=1.5 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance) (1.15.0)  
Requirement already satisfied: chardet<5,>=3.0.2 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from requests>=2.20->yfinance) (4.0.0)  
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from requests>=2.20->yfinance) (1.26.4)  
Requirement already satisfied: certifi>=2017.4.17 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from requests>=2.20->yfinance) (2020.12.5)  
Requirement already satisfied: idna<3,>=2.5 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from requests>=2.20->yfinance) (2.10)  
Building wheels for collected packages: yfinance, multitasking  
  Building wheel for yfinance (setup.py) ... done  
  Created wheel for yfinance: filename=yfinance-0.1.63-py2.py3-none-any.whl size=23909 sha256=43420288b5863ffaf614fd3d633d7d6705533452c840200ffeee2328e2eb1d31  
  Stored in directory: /Users/gennaro/Library/Caches/pip/wheels/ec/cc/c1/32da8ee853d742d5d7cbd11ee04421222eb354672020b57297  
  Building wheel for multitasking (setup.py) ... done  
  Created wheel for multitasking: filename=multitasking-0.0.9-py3-none-any.whl size=8368 sha256=c066589f04763658758edb2bae67e6a39fb8a6f3dc7fccca035257af9f34a91f9  
  Stored in directory: /Users/gennaro/Library/Caches/pip/wheels/57/6d/a3/a39b839cc75274d2acfb1c58bfead2f726c6577fe8c4723f13  
Successfully built yfinance multitasking  
Installing collected packages: multitasking, yfinance  
Successfully installed multitasking-0.0.9 yfinance-0.1.63  
Collecting bs4  
  Downloading bs4-0.0.1.tar.gz (1.1 kB)  
Requirement already satisfied: beautifulsoup4 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from bs4) (4.9.3)  
Requirement already satisfied: soupsieve>1.2 in /Users/gennaro/opt/anaconda3/lib/python3.8/site-packages (from beautifulsoup4->bs4) (2.2.1)  
Building wheels for collected packages: bs4  
  Building wheel for bs4 (setup.py) ... done  
  Created wheel for bs4: filename=bs4-0.0.1-py3-none-any.whl size=1273 sha256=73216188ef8557c455cb3bd8888023b6c8c007a880ee5de8c32753b9a3d6c6db  
  Stored in directory: /Users/gennaro/Library/Caches/pip/wheels/75/78/21/68b124549c9bdc94f822c02fb9aa3578a669843f9767776bca  
Successfully built bs4  
Installing collected packages: bs4  
Successfully installed bs4-0.0.1

```
In [4]:  
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
```

## Defining Graphing Function that will be used later

```
In [5]:  
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 ($US Millions)"))  
    stock_data_specific = stock_data[stock_data.Date <= '2021-06-14']  
    revenue_data_specific = revenue_data[revenue_data.Date <= '2021-04-30']  
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data_specific.Date, infer_datetime_format=True), y=stock_data_specific.Close, name="Stock Price"), row=1, col=1)  
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=revenue_data_specific.Revenue, name="Revenue ($US Millions)", line=dict(color="red")), 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 Stock Data

```
In [6]:  
tesla = yf.Ticker("TSLA")  
tesla_data = tesla.history(period = "max")  
tesla_data.reset_index(inplace = True)  
tesla_data.head()
```

```
Out[6]:  
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
```

## Webscraping to Extract Tesla Revenue Data

```
In [7]:  
html_data = requests.get('https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue').text #Use the requests module  
html_data_soup = BeautifulSoup(html_data, 'html5lib') #Parse the html data using 'beautiful_soup'  
  
#Using beautiful soup extract the table with `Tesla Quarterly Revenue` and store it into a dataframe named `tesla_revenue`  
tesla_revenue = pd.read_html('https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue', match = 'Tesla Quarterly Revenue')[0]  
tesla_revenue.columns = ['Date', 'Revenue']
```

```
In [9]:  
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',', '$').str.replace('$', '') #remove the comma and dollar sign from the Revenue column  
tesla_revenue.dropna(inplace=True) #remove all null or empty strings in the Revenue column  
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != '']
```

```
<ipython-input-9-9448d6a07a6e>:1: FutureWarning: The default value of regex will change from True to False in a future version.
```

```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',', '$').str.replace('$', '') #remove the comma and dollar sign from the Revenue column.
```

```
In [10]:  
tesla_revenue.head()
```

```
Out[10]:  
Date Revenue  
0 2021-06-30 11958  
1 2021-03-31 10389  
2 2020-12-31 10744  
3 2020-09-30 8771  
4 2020-06-30 6036
```

## Plot Tesla Stock Graph

```
In [11]:  
make_graph(tesla_data, tesla_revenue, 'Tesla Stock Graph')
```

