Homework 4

Due: Thursday Nov 2, at 11:59pm via Blackboard

Q1. Import the necessary panda libraries (1 point)

In [1]:
```
import numpy as np # for mathematical caluclations
import pandas as pd
import datetime  # to access datetime

# for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px # for interactive plotting
import plotly.graph_objects as go # for interactive plotting
import random
plt.style.use('ggplot')
plt.rcParams["figure.figsize"] = (15,5)
```

Q2A. Using Yahoo Finance import the stock data for Meta and Tesla from 2021-1-1 to 2023-1-1. Show your code (1 point)

In [2]:
```
import yfinance as yf

meta = yf.download("META", start='2021-01-01', end='2023-01-01')
tesla = yf.download("TSLA", start='2021-01-01', end='2023-01-01')

# meta = yf.Ticker("META").history(period = 'max')
# tesla = yf.Ticker("TSLA").history(period = 'max')

# meta.index = pd.to_datetime(meta.index.date)
# meta.index.name = 'Date'

# tesla.index = pd.to_datetime(tesla.index.date)
# tesla.index.name = 'Date'
```

```
[**********100%%**********]  1 of 1 completed
[**********100%%**********]  1 of 1 completed
```

Q2B. Create the data frames for the Meta and Tesla date, parsing the Date variable as dates and changing the index column to Date. (1 point)

In [3]:
```
# using download method from yahoo finance, the index is automatically in tl
meta.index
```

Out[3]:
```
DatetimeIndex(['2021-01-04', '2021-01-05', '2021-01-06', '2021-01-07',
               '2021-01-08', '2021-01-11', '2021-01-12', '2021-01-13',
               '2021-01-14', '2021-01-15',
               ...
               '2022-12-16', '2022-12-19', '2022-12-20', '2022-12-21',
               '2022-12-22', '2022-12-23', '2022-12-27', '2022-12-28',
               '2022-12-29', '2022-12-30'],
              dtype='datetime64[ns]', name='Date', length=503, freq=None)
```

In [4]:
```
meta.head()
```

Out[4]:

Date	Open	High	Low	Close	Adj Close	Volume
2021-01-04	274.779999	275.000000	265.200012	268.940002	268.940002	15106100
2021-01-05	268.290009	272.399994	268.209991	270.970001	270.970001	9871600
2021-01-06	262.000000	267.750000	260.010010	263.309998	263.309998	24354100
2021-01-07	265.899994	271.609985	264.779999	268.739990	268.739990	15789800
2021-01-08	268.309998	268.950012	263.179993	267.570007	267.570007	18528300

In [5]:
```
# using download method from yahoo finance, the index is automatically in t
tesla.index
```

Out[5]:
```
DatetimeIndex(['2021-01-04', '2021-01-05', '2021-01-06', '2021-01-07',
               '2021-01-08', '2021-01-11', '2021-01-12', '2021-01-13',
               '2021-01-14', '2021-01-15',
               ...
               '2022-12-16', '2022-12-19', '2022-12-20', '2022-12-21',
               '2022-12-22', '2022-12-23', '2022-12-27', '2022-12-28',
               '2022-12-29', '2022-12-30'],
              dtype='datetime64[ns]', name='Date', length=503, freq=None)
```

In [6]:
```
tesla.head()
```

Out[6]:

Date	Open	High	Low	Close	Adj Close	Volume
2021-01-04	239.820007	248.163330	239.063339	243.256668	243.256668	145914600
2021-01-05	241.220001	246.946671	239.733337	245.036667	245.036667	96735600
2021-01-06	252.830002	258.000000	249.699997	251.993332	251.993332	134100000
2021-01-07	259.209991	272.329987	258.399994	272.013336	272.013336	154496700
2021-01-08	285.333344	294.829987	279.463318	293.339996	293.339996	225166500

Q3. Merge the stock data for Meta and Tesla stock data. Include the necessary prefixes (2 points)

In [9]:
```
stocks_comb=meta.merge(tesla,on='Date',suffixes=("_meta","_tsla"))
stocks_comb.head()
```

Out[9]:

Date	Open_meta	High_meta	Low_meta	Close_meta	Adj Close_meta	Volume_meta	Ope
2021-01-04	274.779999	275.000000	265.200012	268.940002	268.940002	15106100	239.8
2021-01-05	268.290009	272.399994	268.209991	270.970001	270.970001	9871600	241.2
2021-01-06	262.000000	267.750000	260.010010	263.309998	263.309998	24354100	252.8
2021-01-07	265.899994	271.609985	264.779999	268.739990	268.739990	15789800	259.2
2021-01-08	268.309998	268.950012	263.179993	267.570007	267.570007	18528300	285.3

Q4a. Report the summary statistics for the Adjusted close for Tesla and Meta stocks prices. (1 point)

Summary:

- Seeing the META and TSLA stock have quite close mean value, it can be observed that META had more volatility level during the period since it has higher standard deviation compared TSLA

In [10]:
```
stocks_comb[['Adj Close_tsla', 'Adj Close_meta']].describe()
```

Out[10]:

	Adj Close_tsla	Adj Close_meta
count	503.000000	503.000000
mean	261.542545	250.817098
std	55.775860	84.762832
min	109.099998	88.910004
25%	223.201668	170.205002
50%	251.213333	265.739990
75%	296.856659	330.300003
max	409.970001	382.179993

Q4b: What are the Ranges and Interquartile Ranges for the Adjusted Close of Tesla and Meta Stock prices? (1 point)

In [11]:
```
# Meta
meta_adjclose_range = stocks_comb['Adj Close_meta'].max() - stocks_comb['Adj
meta_adjclose_iqr = stocks_comb['Adj Close_meta'].quantile(0.75) - stocks_co

print(f' Meta adjusted close price range: {meta_adjclose_range}')
print(f' Meta adjusted close interquantile range: {meta_adjclose_iqr}')
```

```
Meta adjusted close price range: 293.2699890136719
Meta adjusted close interquantile range: 160.09500122070312
```

In [12]:
```
# Tesla
tesla_adjclose_range = stocks_comb['Adj Close_tsla'].max() - stocks_comb['Ad
tesla_adjclose_iqr = stocks_comb['Adj Close_tsla'].quantile(0.75) - stocks_(

print(f' Tesla adjusted close price range: {tesla_adjclose_range}')
print(f' Tesla adjusted close interquantile range: {tesla_adjclose_iqr}')
```

```
Tesla adjusted close price range: 300.87000274658203
Tesla adjusted close interquantile range: 73.65499114990234
```

Q5. Plot line graphs for the daily volumes for Meta and Tesla Stock and include the title and labels. Which stock had greater volumn volatility? (2 points)

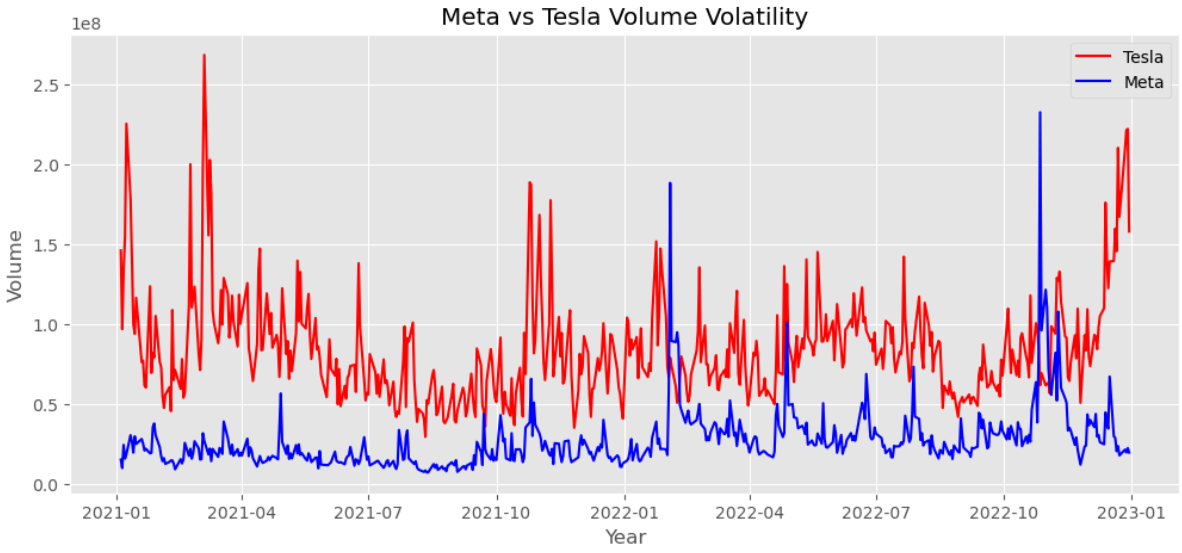
In [13]:
```
plt.figure(figsize = (12, 5))

sns.lineplot(data = stocks_comb,
             x = 'Date',
             y = 'Volume_tsla',
             color = 'red',label='Tesla')

sns.lineplot(data = stocks_comb,
             x = 'Date',
             y = 'Volume_meta',
             color = 'blue',label='Meta')

plt.ylabel('Volume')
plt.xlabel('Year')
plt.title('Meta vs Tesla Volume Volatility')
```

Out[13]:
```
Text(0.5, 1.0, 'Meta vs Tesla Volume Volatility')
```



Q6. Create a new variables for both Tesla and Meta stocks in the dataframe that measures the daily differential in price, defined as the difference between the High Price and Low Price for the day (1 point)

In [14]:
```
stocks_comb['PriceDff_tsla']=stocks_comb['High_tsla']-stocks_comb['Low_tsla
stocks_comb['PriceDff_meta']=stocks_comb['High_meta']-stocks_comb['Low_meta
stocks_comb.head(5)
```

Out[14]:

Date	Open_meta	High_meta	Low_meta	Close_meta	Adj Close_meta	Volume_meta	Ope
2021-01-04	274.779999	275.000000	265.200012	268.940002	268.940002	15106100	239.8
2021-01-05	268.290009	272.399994	268.209991	270.970001	270.970001	9871600	241.2
2021-01-06	262.000000	267.750000	260.010010	263.309998	263.309998	24354100	252.8
2021-01-07	265.899994	271.609985	264.779999	268.739990	268.739990	15789800	259.2
2021-01-08	268.309998	268.950012	263.179993	267.570007	267.570007	18528300	285.3

Q7. Resample the data to create the average monthly price diffferences between Tesla and Meta Stock. (2 points)

In [15]:
```
avg_month_price_diff = stocks_comb[['PriceDff_tsla','PriceDff_meta']].resamp
avg_month_price_diff.head()
```

Out[15]:

Date	PriceDff_tsla	PriceDff_meta
2021-01-31	11.362281	8.803686
2021-02-28	12.744733	6.808423
2021-03-31	14.384638	8.769563
2021-04-30	9.516985	6.107147
2021-05-31	8.813497	6.550000

Q8. Using Plotly, create line graphs for the monthly price differences for Tesla and Meta stocks. Include the range slider. Don't show the gridlines but include the title and labels (3 points)

In [16]:
```
fig = go.Figure()

fig.add_trace(go.Scatter(x=avg_month_price_diff.index, y=avg_month_price_di

fig.add_trace(go.Scatter(x=avg_month_price_diff.index, y=avg_month_price_di



fig.update_xaxes(rangeslider_visible = True)

fig.update_layout(xaxis=dict(showline=True,showgrid=False),
  yaxis=dict(
        showgrid=False,
        showline=False,
        showticklabels=False),
  legend=dict(title='Stocks'),)
```

```
fig.update_layout(title= 'Price Difference for Meta vs Tesla Stock',
                  xaxis_title='Month',
                  yaxis_title='Monthly Average Price Difference')


fig.show()
```