Tesla Stock Market

Amy Zhang
Asfia Khatoon
Carolina Connors

Introduction

An American electric and clean energy company founded in 2003 and launched in 2008. Its mission is to accelerate the world's transition to sustainable energy. The faster the world stops relying on fossil fuels and moves towards a zero emission future, the better.

It is one of leading company in the field of technology which offers many products which include:

- Electric cars
- Home batteries which serve as energy storage systems
- Solar Panels and Solar Roofing
- Financial services eg.vehicle loans and leasing

Motivation

We have chosen Tesla after a discussion about Elon Musk's most recent announcement of a humanoid robot.

Here in this project stock analysis is done by taking reference of 10 year stock trading record of Tesla Inc. until now.

<u>Question 1:</u> How did Tesla perform during COVID-19 and after?

What can be analyse from the data?

- What happened during COVID-19?
- What was the stock's price performance over the time?
- What was the change of the stock's volume?

Beginning the Process…

- Retrieved Tesla's historical stock data from Kaggle-contained data from 2010-2021
- Retrieved Tesla historical stock data from Yahoo Finance using Datareader library.

Data Cleaning and Exploration

```
#import libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import requests
from datetime import datetime
```

```
#install seaborn package

import seaborn as sns
```

```
#import libraries from vrious internet sources into a DataFrame

import pandas_datareader as pdr
import pandas_datareader.data as web
```

Install this library using pip install to convert the dates to strings

```
In [29]: file = "../TSLA (1).csv"
```

```
In [30]: df_tesla = pd.read_csv(file)
```

```
In [10]: df_tesla.head(20)
```

Out[10]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	01/07/2010	5.000	5.184	4.054	4.392	4.392	41094000
1	02/07/2010	4.600	4.620	3.742	3.840	3.840	25699000
2	06/07/2010	4.000	4.000	3.166	3.222	3.222	34334500
3	07/07/2010	3.280	3.326	2.996	3.160	3.160	34608500
4	08/07/2010	3.228	3.504	3.114	3.492	3.492	38557000
5	09/07/2010	3.516	3.580	3.310	3.480	3.480	20253000
6	12/07/2010	3.590	3.614	3.400	3.410	3.410	11012500

```
In [11]: #Count number

df_tesla.count()
```

```
Out[11]: Date         2843
Open         2843
High         2843
Low          2843
Close        2843
Adj Close    2843
Volume       2843
dtype: int64
```

```
In [12]: #change the index column

df = df_tesla.set_index("Date")
df.head()
```

Out[12]:

	Open	High	Low	Close	Adj Close	Volume
Date						
01/07/2010	5.000	5.184	4.054	4.392	4.392	41094000
02/07/2010	4.600	4.620	3.742	3.840	3.840	25699000
06/07/2010	4.000	4.000	3.166	3.222	3.222	34334500
07/07/2010	3.280	3.326	2.996	3.160	3.160	34608500
08/07/2010	3.228	3.504	3.114	3.492	3.492	38557000

```
In [13]: # Drop any na if exits

df_na = df_tesla.dropna(how='any')
df_na.head()
```

Out[13]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	01/07/2010	5.000	5.184	4.054	4.392	4.392	41094000
1	02/07/2010	4.600	4.620	3.742	3.840	3.840	25699000
2	06/07/2010	4.000	4.000	3.166	3.222	3.222	34334500
3	07/07/2010	3.280	3.326	2.996	3.160	3.160	34608500
4	08/07/2010	3.228	3.504	3.114	3.492	3.492	38557000

```
In [14]: #Check if the numbers has changed

df_na.count()
```

```
Out[14]: Date         2843
Open         2843
High         2843
Low          2843
```

Data Cleaning and Exploration

```
df_tesla.describe()
```

	Open	High	Low	Close	Adj Close	Volume
count	2843.000000	2843.000000	2843.000000	2843.000000	2843.000000	2.843000e+03
mean	105.868475	108.031370	103.555733	105.924597	105.924597	3.141524e+07
std	188.738974	192.483055	184.638617	188.836358	188.836358	2.841880e+07
min	3.228000	3.326000	2.996000	3.160000	3.160000	5.925000e+05
25%	10.698000	11.026000	10.420000	10.727000	10.727000	1.251050e+07
50%	45.874001	46.493999	45.102001	45.916000	45.916000	2.481500e+07
75%	65.021000	66.251999	64.015001	65.275002	65.275002	4.012025e+07
max	891.380005	900.400024	871.599976	883.090027	883.090027	3.046940e+08

We can also check some stats of the data, such as, how many rows are there, max value, mean value. By using a describe command, provided to us by Pandas.

```
data.tail()
```

	High	Low	Open	Close	Volume	Adj Close
Date						
2021-10-19	877.950012	862.510010	877.530029	864.270020	17381100.0	864.270020
2021-10-20	869.489990	857.380005	865.349976	865.799988	14032100.0	865.799988
2021-10-21	900.000000	855.500000	856.000000	894.000000	31481500.0	894.000000
2021-10-22	910.000000	890.960022	895.500000	909.679993	22836800.0	909.679993
2021-10-25	1045.020020	944.200012	950.530029	1024.859985	62689200.0	1024.859985

We can use Tail to check how old the data is.

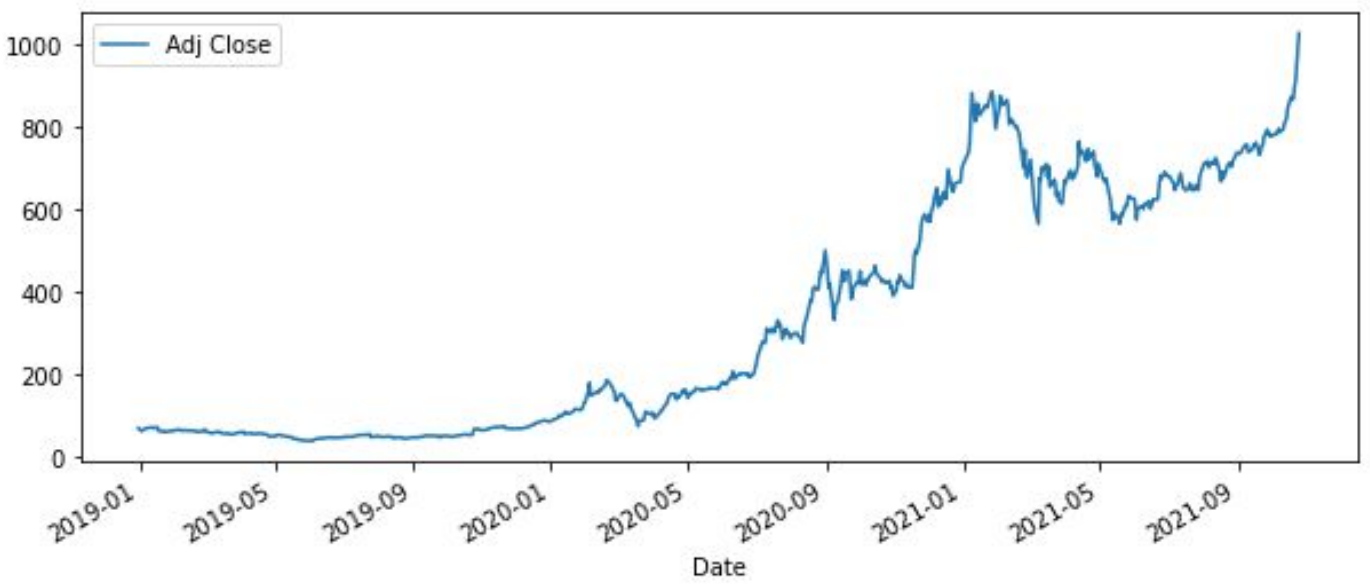
Data Analysis

Tesla Stock Market during COVID

```
#Data of 2019 during COVID
Ticker = "TSLA"
start = datetime(2019, 1, 1)

data2019 = pdr.get_data_yahoo(ticker, start)
data.head()
```

The easiest way to download the stock's historical data in Python is with yfinance package.

	High	Low	Open	Close	Volume	Adj Close
Date						
2020-10-26	425.760010	410.000000	411.630005	420.279999	28239200	420.279999
2020-10-27	430.500000	420.100006	423.760010	424.679993	22686500	424.679993
2020-10-28	418.600006	406.000000	416.480011	406.019989	25451400	406.019989
2020-10-29	418.059998	406.459991	409.959991	410.829987	22655300	410.829987
2020-10-30	407.589996	379.109985	406.899994	388.040009	42511300	388.040009



Tesla is about to reveal how badly coronavirus hit its profit plans

By Chris Isidore, CNN Business

Updated 1416 GMT (2216 HKT) April 29, 2020

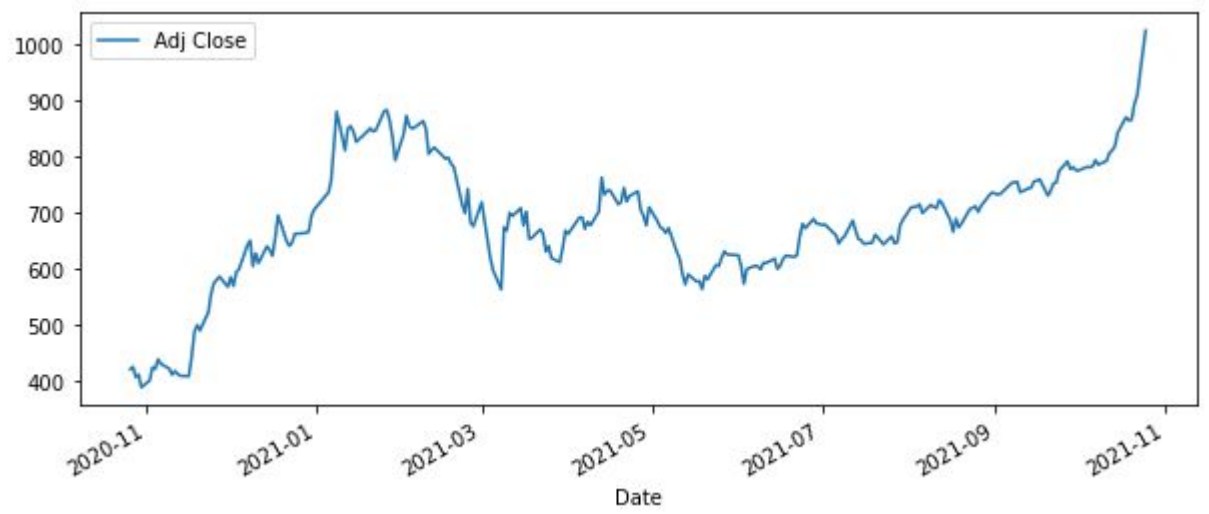
- To analyse the stock market performance during COVID I used 2019 data, defined the start datetime as 2019. The datetime library is very useful for when you want to we can use the datetime of the datetime module. The datetime() class requires three parameters to create a date: year, month, day.

Performance of Tesla Stock Market

Tesla stock traded over time

```
# Setting the Start and End date for Stock Market Analysis
Ticker = "TSLA"
start = datetime(end.year-1,end.month,end.day)
end = datetime.now()

data = pdr.get_data_yahoo(ticker, start)
data.head()
```

	High	Low	Open	Close	Volume	Adj Close
Date						
2020-10-26	425.760010	410.000000	411.630005	420.279999	28239200	420.279999
2020-10-27	430.500000	420.100006	423.760010	424.679993	22686500	424.679993
2020-10-28	418.600006	406.000000	416.480011	406.019989	25451400	406.019989
2020-10-29	418.059998	406.459991	409.959991	410.829987	22655300	410.829987
2020-10-30	407.589996	379.109985	406.899994	388.040009	42511300	388.040009

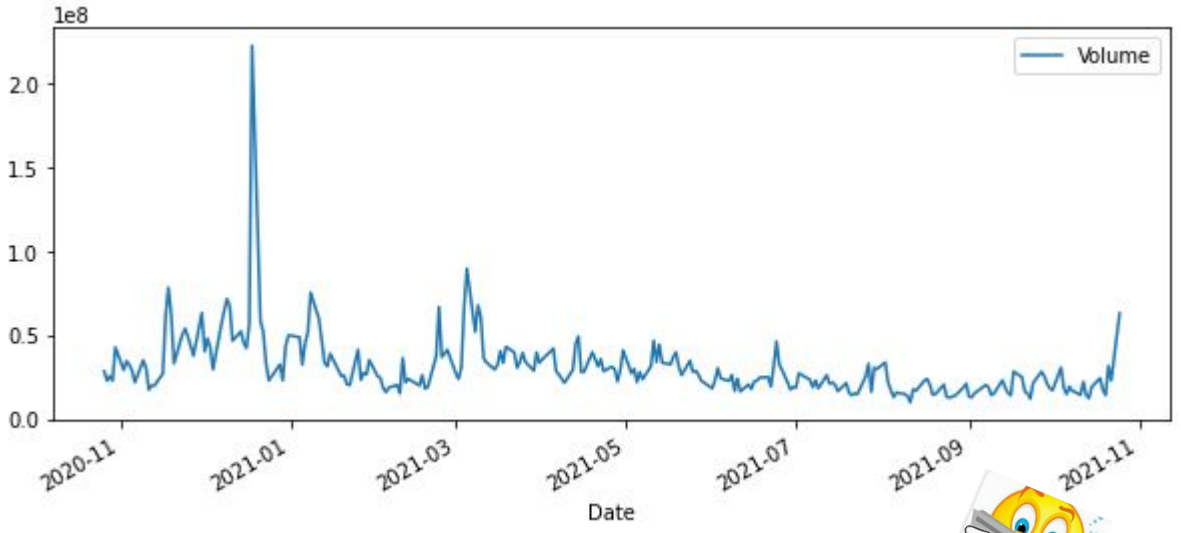


Total Volume per day

Total volume of stock being traded per day

```
# Plotting the total volume of stock being traded each day

data['Volume'].plot(legend=True,figsize=(10,4))
plt.show()

plt.savefig("current total volume of stock.png")
```

```
In [130]: volume_average = df_tesla["Volume"].mean()
          volume_average
Out[130]: 31415237.42525501

In [131]: total_volume = df_tesla["Volume"].max()
          total_volume
Out[131]: 304694000

In [132]: volume_deviation = df_tesla["Volume"].std()
          volume_deviation
Out[132]: 28418797.352951575
```



The Guardian

Tesla breaks $1tn valuation barrier after Hertz orders 100,000 vehicles

The carmaker's stock market value has soared during 2020 and 2021 as investors bet on accelerating sales of electric cars in the run-up to...

6 days ago

CNBC

Tesla posts record net income of $438 million, revenue surges by 74%

On an earnings call, CEO Elon Musk said the delayed new version of ... In January 2021 (during a fourth-quarter 2020 earnings update) Musk...

26 Apr 2021

Question 2: How Tesla performed against NIO?

What is NIO?

- Chinese manufacturer of electric vehicles and branched to autonomous driving
- Specialises in designing and developing premium electric cars that feature advanced technologies.
- Tesla's main competitor: "The Tesla of China"

Tesla or NIO?

- Both similar in nature: How do they perform relative to each other in the stock market?
- If I were to invest...Which should I invest in?
- Are there any relationships/correlation between the two?

Beginning the Process…

- Retrieved Tesla's historical stock data from Kaggle-contained data from 2010-2021
- Retrieved NIO's historical stock data from Yahoo Finance

Data Cleaning and Exploration

```
1]:
#Import libraries and dependencies
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as sts
import numpy as np
import datetime
from dateutil import parser
from pandas.plotting import scatter_matrix
```

Install this library using pip install to convert the dates to strings

```
2]:
#Specifying time frame
start=datetime.datetime(2019,1,1)
end=datetime.datetime(2022,1,1)
```

Founded in November 2014, started trading in late 2018

```
In [3]:
#Reading csv file: Tesla
tesla_file="TSLA (1).csv"
tesla_data=pd.read_csv(tesla_file)
tesla_data.head()
```

Out[3]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	01/07/2010	5.000	5.184	4.054	4.392	4.392	41094000
1	02/07/2010	4.600	4.620	3.742	3.840	3.840	25699000
2	06/07/2010	4.000	4.000	3.166	3.222	3.222	34334500
3	07/07/2010	3.280	3.326	2.996	3.160	3.160	34608500
4	08/07/2010	3.228	3.504	3.114	3.492	3.492	38557000

```
In [4]:
#Reading csv file: NIO
nio_file="NIO.csv"
nio_data=pd.read_csv(nio_file)
nio_data.head()
```

Out[4]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-11-20	7.65	7.91	7.15	7.67	7.67	26923200
1	2018-11-21	7.90	8.05	7.63	7.71	7.71	13993600
2	2018-11-23	7.68	7.70	7.41	7.46	7.46	8119700
3	2018-11-26	7.61	7.63	7.16	7.37	7.37	12164900
4	2018-11-27	7.30	7.70	7.23	7.48	7.48	9084900

```
In [5]:
#Date conversion for NIO
nio_data["Date"] =nio_data["Date"].apply(parser.parse)
nio_data
```

Out[5]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-11-20	7.650000	7.910000	7.150000	7.670000	7.670000	26923200
1	2018-11-21	7.900000	8.050000	7.630000	7.710000	7.710000	13993600
2	2018-11-23	7.680000	7.700000	7.410000	7.460000	7.460000	8119700
				0000	7.160000	7.370000	7.370000	12164900
				0000	7.230000	7.480000	7.480000	9084900
				...	...	...	...	...
				0000	39.160000	39.779999	39.779999	25120700
				9998	39.369999	39.970001	39.970001	25389100
				9999	38.540001	38.880001	38.880001	27962900
				9998	39.240002	41.270000	41.270000	46935600
				0001	40.470001	40.470001	40.470001	55798800

```
In [6]:
#Date conversion for Tesla
tesla_data["Date"]=tesla_data["Date"].apply(parser.parse)
tesla_data
```

Out[6]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-01-07	5.000000	5.184000	4.054000	4.392000	4.392000	41094000
1	2010-02-07	4.600000	4.620000	3.742000	3.840000	3.840000	25699000
2	2010-06-07	4.000000	4.000000	3.166000	3.222000	3.222000	34334500
3	2010-07-07	3.280000	3.326000	2.996000	3.160000	3.160000	34608500
4	2010-08-07	3.228000	3.504000	3.114000	3.492000	3.492000	38557000
...	...	...	...	...	...	...	...
2838	2021-08-10	796.210022	796.380005	780.909973	785.489990	785.489990	16711100
2839	2021-11-10	787.650024	801.239990	785.500000	791.940002	791.940002	14200300
2840	2021-12-10	800.929993	812.320007	796.570007	805.719971	805.719971	22020000
2841	2021-10-13	810.469971	815.409973	805.780029	811.080017	811.080017	14120100
2842	2021-10-14	815.489990	820.250000	813.349976	818.320007	818.320007	12203200

2843 rows × 7 columns

Before Date Conversion, tried to merge the Data Frames together and data was not showing.

Out[7]:

Date	Open_Tesla	High_Tesla	Low_Tesla	Close_Tesla	Adj Close_Tesla	Volume_Tesla	Open_NIO	High_NIO	Low_NIO	Close_NIO	Adj Close_NIO	Volume_NIO

Data Cleaning and Exploration

In [7]:
```
#Merging the dataframes
merged_data=pd.merge(tesla_data,nio_data,on="Date",suffixes=("_Tesla","_NIO"))
merged_data.head()
```

In [8]:
```
#Setting the date as index
merged_data['Date'] = pd.to_datetime(merged_data['Date'])

merged_data.set_index('Date', inplace = True)
```

In [9]:
```
#Rounding the figures
clean_df=merged_data.round(decimals=3)
clean_df.head()
```

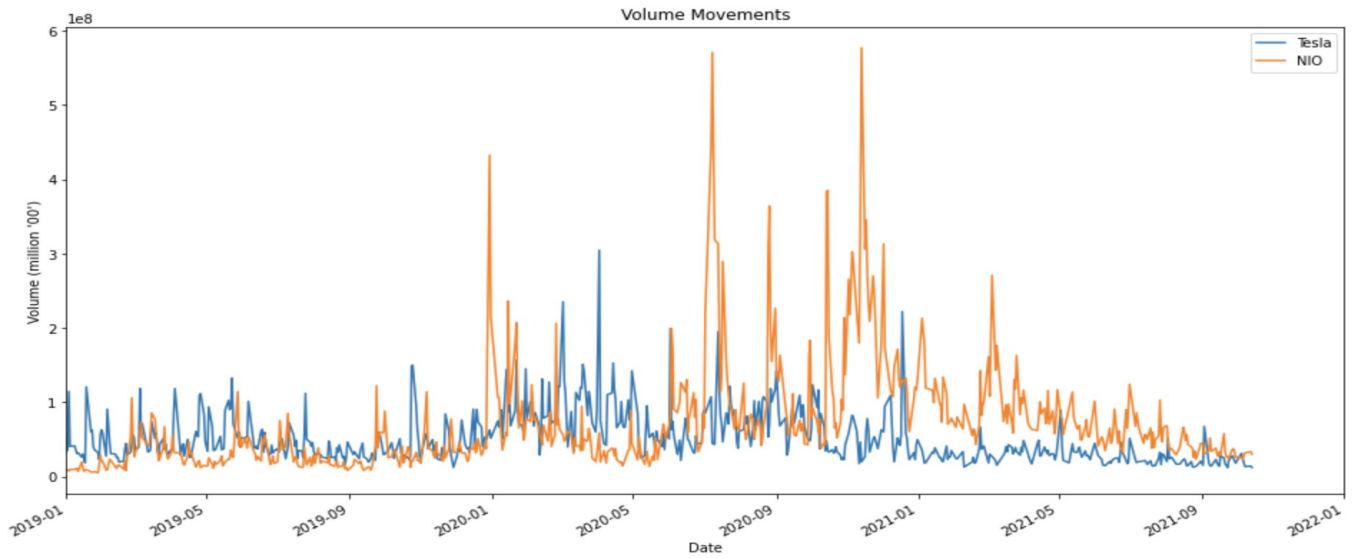
Out[9]:

Date	Open_Tesla	High_Tesla	Low_Tesla	Close_Tesla	Adj Close_Tesla	Volume_Tesla	Open_NIO	High_NIO	Low_NIO	Close_NIO	Adj Close_NIO	Volume_NIO
2018-12-03	65.722	69.442	65.300	69.102	69.102	41320000	8.10	8.13	7.50	7.58	7.58	22425400
2018-12-04	60.464	60.790	58.736	58.816	58.816	38044000	7.60	7.64	7.05	7.08	7.08	19624700
2018-12-06	68.940	70.994	67.600	68.554	68.554	111737000	6.86	7.37	6.72	7.37	7.37	16688100
2018-12-07	64.286	64.646	62.554	63.342	63.342	28606000	7.36	7.38	6.96	6.99	6.99	11576200
2018-12-10	52.200	52.398	50.402	51.756	51.756	36007000	7.03	7.12	6.74	7.03	7.03	9981000

Data Analysis

In [11]:

```
#Plotting based on Volume
clean_df["Volume_Tesla"].plot(label="Tesla",figsize=(16,8),title="Volume Movements")
clean_df["Volume_NIO"].plot(label="NIO")

plt.legend()
plt.xlabel("Date")
plt.ylabel("Volume (million)")

plt.xlim(("2019-01-01","2022-01-01"))

plt.show()

plt.savefig("Volume Movements")
```



Data Analysis

In [12]:
```
#Determine hgihest volume trade
telsa_vol=clean_df["Volume_Tesla"].idxmax()
tesla_max=clean_df["Volume_Tesla"].max()
nio_vol=clean_df["Volume_NIO"].idxmax()
nio_max=clean_df["Volume_NIO"].max()

print(f"Highest volume traded for Tesla of {tesla_max} on date: {telsa_vol}")
print(f"Highest volume traded for NIO of {nio_max} on date: {nio_vol}")
```

```
Highest volume traded for Tesla of 304694000 on date: 2020-04-02 00:00:00
Highest volume traded for NIO of 577185100 on date: 2020-11-13 00:00:00
```

NIO News and Event Update:

December 1, 2020 at 5:00 AM EST

NIO Inc. Provides November 2020 Delivery Update

Company Achieved Another New Record-High Monthly Deliveries

- *NIO delivered 5,291 vehicles in November 2020, increasing by 109.3% year-over-year*
- *NIO delivered 36,721 vehicles in 2020 in total, increasing by 111.1% year-over-year*
- *Cumulative deliveries of ES8, ES6 and EC6 as of November 30, 2020 reached 68,634*

CNBC Article:

MARKETS

Tesla shares end Tuesday 13% higher after surging as much as 23% earlier in the day

PUBLISHED TUE, FEB 4 2020•7:27 AM EST | UPDATED TUE, FEB 4 2020•4:42 PM EST

Michael Sheetz
@THESHEETZTWEETZ

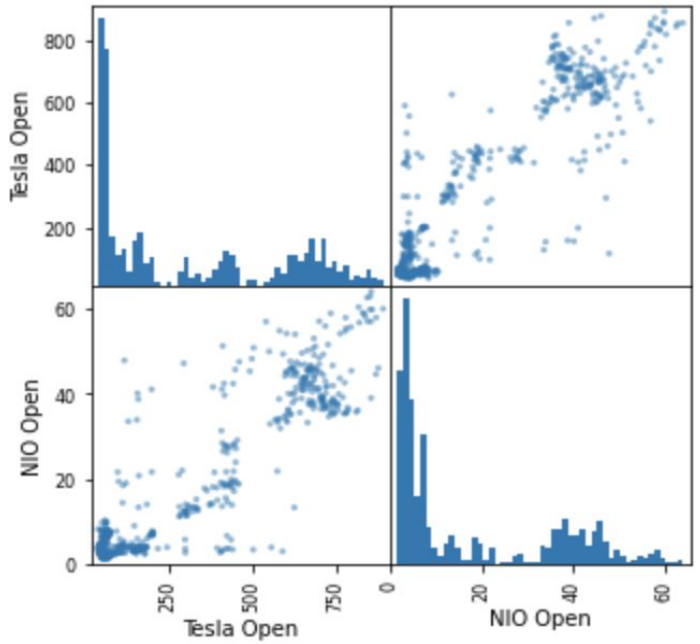
SHARE

Tesla stock surged again on Tuesday after major shareholder Ron Baron forecast the company will top $1 trillion in revenue in a decade and as investors who bet against the stock scrambled to catch up.

Data Analysis

In [13]:
```
#Determining a correlation
car_co=pd.concat([clean_df["Open_Tesla"],clean_df["Open_NIO"]],axis=1)
car_co.columns=["Tesla Open","NIO Open"]
```

In [14]:
```
scatter_matrix(car_co,figsize=(6,6),hist_kwds={"bins":50})
plt.savefig("scatter matrix")
```



In [15]:
```
correlation=sts.pearsonr(clean_df["Open_Tesla"],clean_df["Open_NIO"])
print(f"The correlation value between Tesla and NIO open prices is {correlation[0]}")
```

```
The correlation value between Tesla and NIO open prices is 0.9115463671593006
```

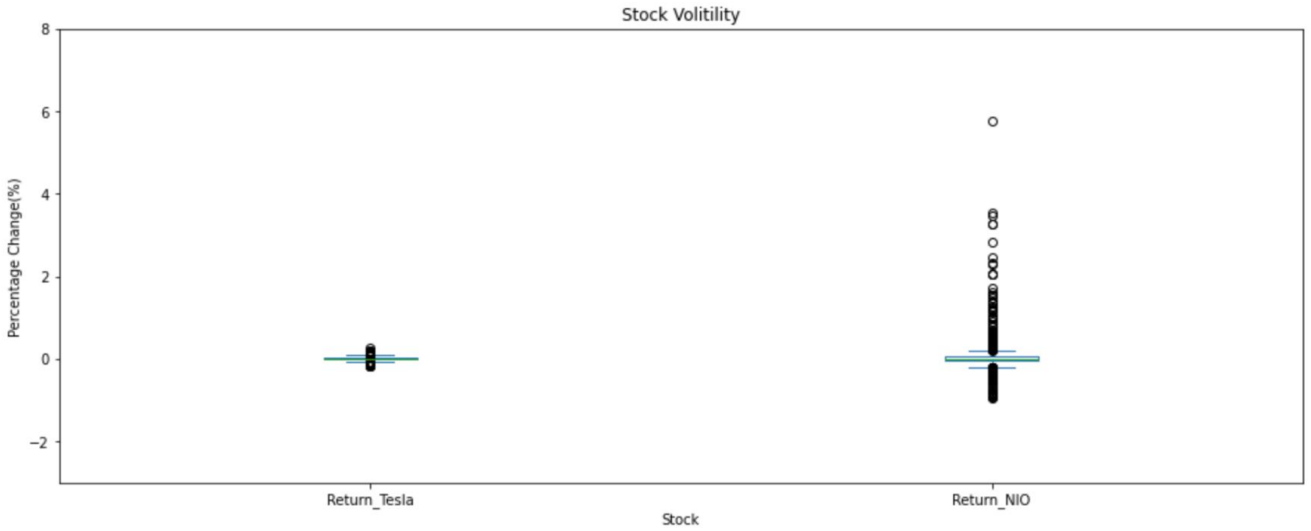
Data Analysis

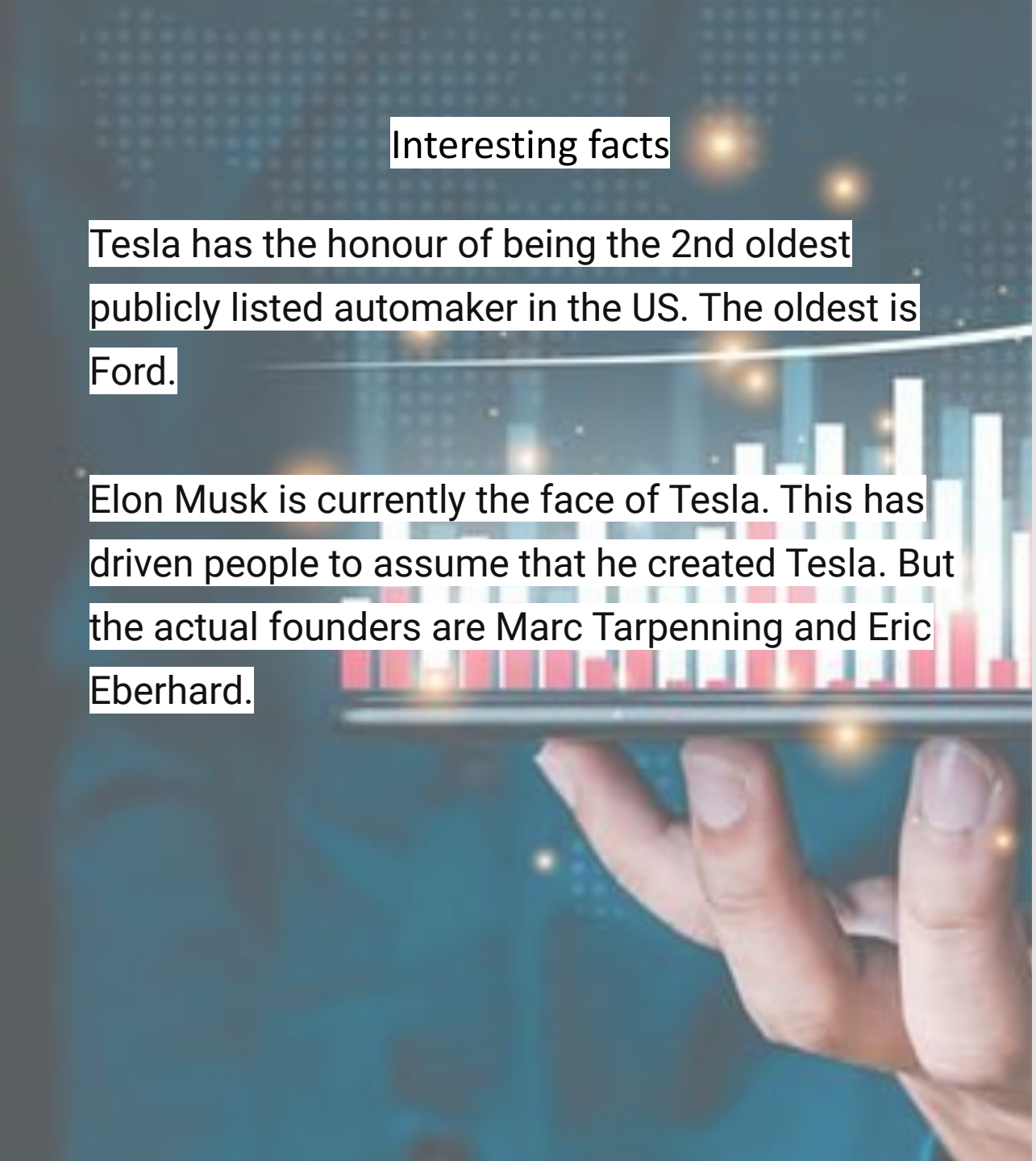
In [16]:
```
#Calculating daily Percentage Change

#Return(t)=(p(t)/p(t-1))-1

clean_df["Return_Tesla"]=(clean_df["Close_Tesla"]/clean_df["Close_Tesla"].shift(1))-1
clean_df["Return_NIO"]=(clean_df["Close_NIO"]/clean_df["Close_NIO"].shift(1))-1

clean_df[['Return_Tesla','Return_NIO']]
```

Out[16]:

Date	Return_Tesla	Return_NIO
2018-12-03	NaN	NaN
2018-12-04	-0.148852	-0.065963
2018-12-06	0.165567	0.040960
2018-12-07	-0.076028	-0.051560
2018-12-10	-0.182912	0.005722
...	...	...
2021-05-10	0.006589	-0.036486
2021-06-10	0.002767	0.245558
2021-08-10	0.003500	0.034144
2021-10-13	0.032578	-0.180461
2021-10-14	0.008926	0.001104

In [17]:
```
#Boxplots
boxplot_df=pd.concat([clean_df["Return_Tesla"],clean_df["Return_NIO"]],axis=1)
boxplot_df.plot(kind="box",figsize=(16,6))

plt.title("Stock Volitility")
plt.ylabel("Percentage Change(%)")
plt.xlabel("Stock")

plt.ylim(-3,8)

plt.savefig("Stock Volitility")
```





Interesting facts

Tesla has the honour of being the 2nd oldest publicly listed automaker in the US. The oldest is Ford.

Elon Musk is currently the face of Tesla. This has driven people to assume that he created Tesla. But the actual founders are Marc Tarpenning and Eric Eberhard.

Question 3: How did Tesla perform against Ford?

Data Source for Tesla and Ford -
Yahoo finance

```
In [2]: import pandas_datareader
        import datetime

In [3]: import pandas_datareader.data as web

In [20]: start = datetime.datetime(2012,1,1)
         end = datetime.datetime(2021,2,1)
         tesla=web.DataReader('TSLA','yahoo',start,end)

In [21]: tesla.head()
```

Out[21]:

	High	Low	Open	Close	Volume	Adj Close
Date						
2012-01-03	5.900	5.530	5.788	5.616	4640500.0	5.616
2012-01-04	5.734	5.500	5.642	5.542	3150500.0	5.542

```
[22]: ford=web.DataReader('FORD','yahoo',start,end)
      ford.head()
```

t[22]:

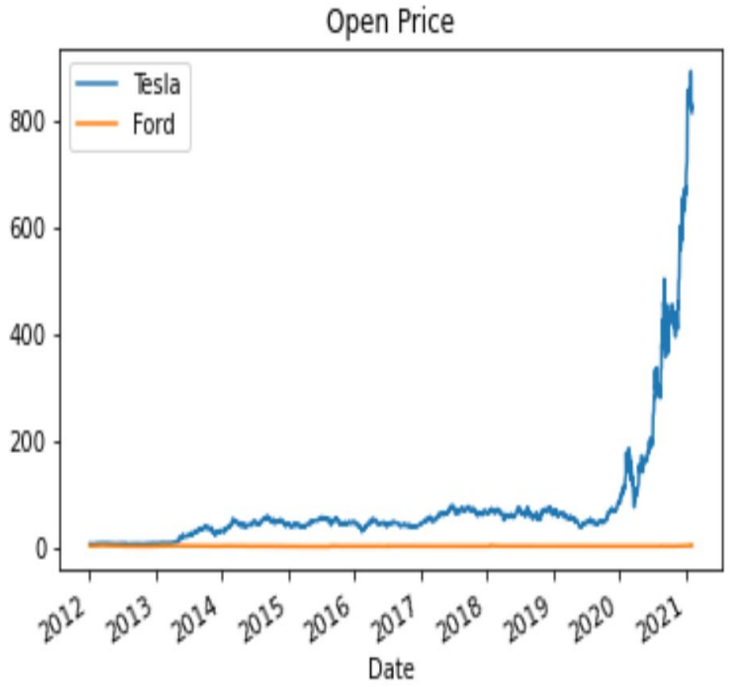
	High	Low	Open	Close	Volume	Adj Close
Date						
2012-01-03	1.70	1.65	1.69	1.67	16700	1.67
2012-01-04	1.71	1.65	1.68	1.66	5600	1.66

Tesla vs Ford 2012 -2021- Basic evaluation using line plots

Tesla's stock performance has crushed Ford's. For example, here's the last ten years:

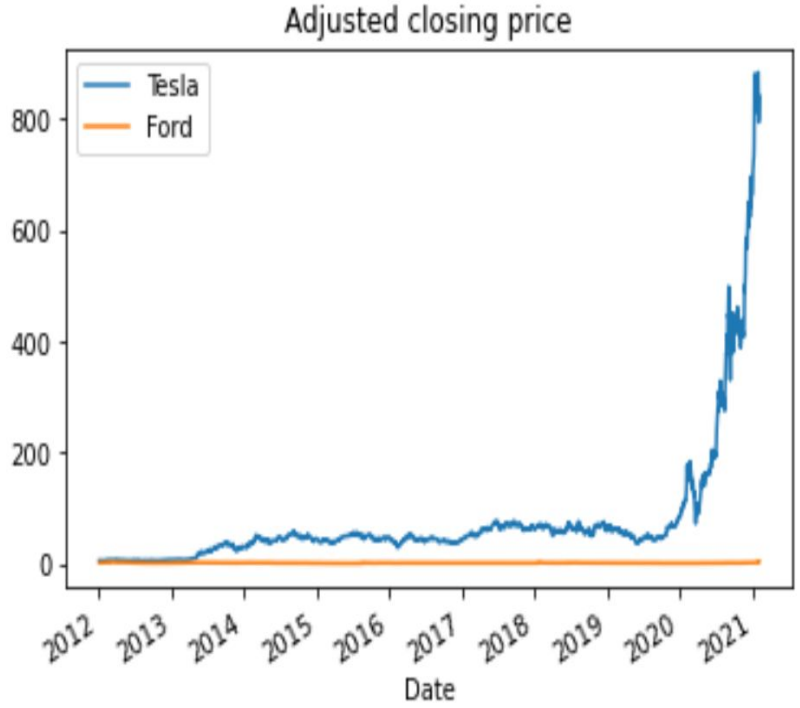
In [23]:
```
#plot based on opening prices
tesla['Open'].plot(label ='Tesla',title='Open Price')
ford['Open'].plot(label ='Ford')
plt.legend()
```

Out[23]: <matplotlib.legend.Legend at 0x1a08e660748>



In [24]:
```
#Plot based on adjusted closing prices
tesla['Adj Close'].plot(label ='Tesla',title='Adjusted closing price')
ford['Adj Close'].plot(label ='Ford')
plt.legend()
```

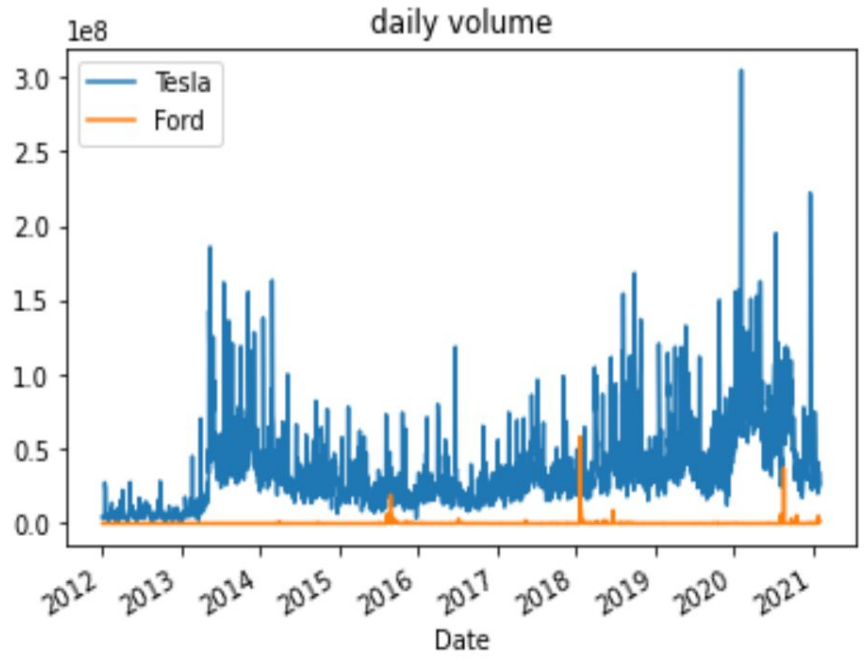
Out[24]: <matplotlib.legend.Legend at 0x1a09010eec8>



Distribution of daily volumes each Year

```
In [25]: #daily volume
         tesla['Volume'].plot(label ='Tesla',title='daily volume')
         ford['Volume'].plot(label ='Ford')
         plt.legend()
```

```
Out[25]: <matplotlib.legend.Legend at 0x1a0900bb6c8>
```




```
In [26]: #dates for maximimum trading
         tesla['Volume'].idxmax()
```

```
Out[26]: Timestamp('2020-02-04 00:00:00')
```

```
In [27]: ford['Volume'].idxmax()
```

```
Out[27]: Timestamp('2018-01-19 00:00:00')
```

Ford vs Tesla daily percentage change

In [46]:
```
#finding daily percentage change
tesla['returns'] = tesla['Close'].pct_change(1)
ford['returns'] = ford['Close'].pct_change(1)
tesla.head()
```

Out[46]:

	High	Low	Open	Close	Volume	Adj Close	Total Traded	returns
Date								
2012-01-03	5.900	5.530	5.788	5.616	4640500.0	5.616	2.685921e+07	NaN
2012-01-04	5.734	5.500	5.642	5.542	3150500.0	5.542	1.777512e+07	-0.013177
2012-01-05	5.586	5.370	5.552	5.424	5027500.0	5.424	2.791268e+07	-0.021292
2012-01-06	5.558	5.282	5.440	5.382	4931500.0	5.382	2.682736e+07	-0.007743
2012-01-09	5.498	5.224	5.400	5.450	4485000.0	5.450	2.421900e+07	0.012635

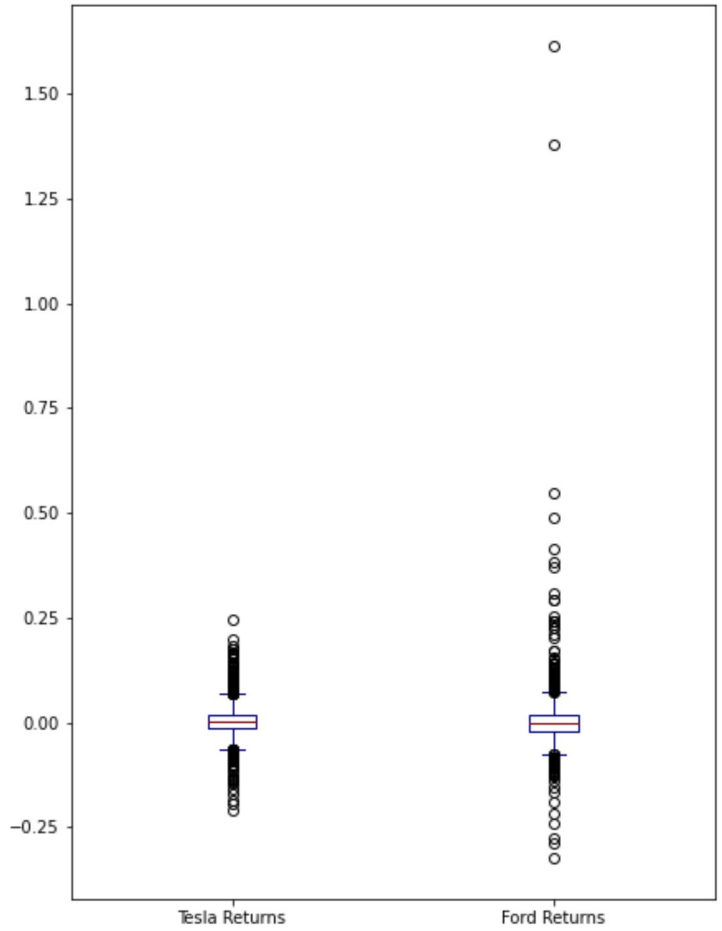
In [47]:
```
ford.head()
```

Out[47]:

	High	Low	Open	Close	Volume	Adj Close	Total Traded	returns
Date								
2012-01-03	1.70	1.65	1.69	1.67	16700	1.67	28223.000956	NaN
2012-01-04	1.71	1.65	1.68	1.66	5600	1.66	9407.999706	-0.005988
2012-01-05	1.69	1.65	1.65	1.67	9200	1.67	15179.999781	0.006024
2012-01-06	1.69	1.66	1.68	1.67	9800	1.67	16463.999486	0.000000
2012-01-09	1.75	1.68	1.72	1.75	34500	1.75	59340.000987	0.047904

In [48]:
```
#boxplots comparing returns
box_df = pd.concat([tesla['returns'],ford['returns']],axis=1)
box_df.columns = ['Tesla Returns','Ford Returns']
box_df.plot(kind='box',figsize=(7,10),colormap = 'jet')
```

Out[48]: <AxesSubplot:>



Elon Musk tweets on 18-12-2020

← **Tweet**

Elon Musk ✓
@elonmusk

Boring Company product launch on Dec 18. More than a tunnel opening. Will include modded but fully road legal autonomous transport cars & ground to tunnel car elevators.

8:18 AM · Dec 7, 2018 · Twitter for iPhone

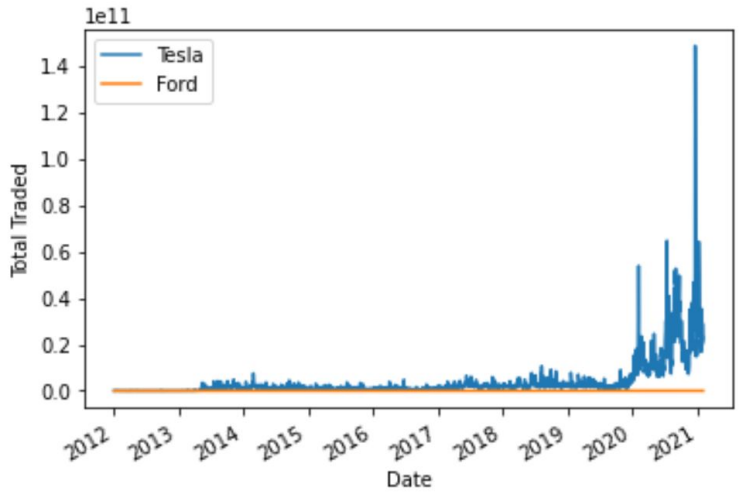
Finding the max trading day

Tesla make its much-anticipated debut into the benchmark S&P 500 index, after rising to a record high on Friday 18 Dec 2020 in a frantic day of trading.

"From the beginning of 2020 to the announcement date, Tesla's share price rose 400% from $83.67 to $408.09. Most of that runup occurred after the media began speculating in March about Tesla's likely addition to the index. From the announcement date through December 7, Tesla's share price rose another 49% to $608.32."

In [29]:
```
#finding total traded - open price * volume traded
tesla['Total Traded']= tesla['Open']*tesla['Volume']
ford['Total Traded']= ford['Open']*ford['Volume']

#plotting Total Traded against the time index'
tesla['Total Traded'].plot(label='Tesla')
ford['Total Traded'].plot(label='Ford')
plt.legend()
plt.ylabel('Total Traded')
```

Out[29]:
```
Text(0, 0.5, 'Total Traded')
```



In [32]:
```
#finding total traded with Elon Musk's tweets
#tesla Total traded surges
tesla['Total Traded'].idxmax()
```

Out[32]:
```
Timestamp('2020-12-18 00:00:00')
```