US STOCK PRICES

PYTHON PROJECT

Ambika Sharma
Vatsal Pancholi

TABLE OF CONTENTS

INTRODUCTION

Financial exchange lists all over the planet are strong markers for worldwide and country-explicit economies. S&P 500, Dow Jones Industrial Average, and Nasdaq Composite are the three most extensively tracked lists by both media and financial backers in the United States. Notwithstanding these three records, there are around 5,000 others that make up the U.S. value market.[1]
A stock is a little piece of possession in organization. The trade cost of organization mirrors the remaining assessment of organization, it furthermore provides a slight understanding into its exhibition. These shares are exchanged on trades and their costs are continually changing because of their interest and supply on the lookout. Assuming a stock is popular and low in supply for example more individuals need to get it and less individuals will sell it then the cost for the stock will go up and comparably assuming the stock is in low interest and high inventory which means individuals more individuals are prepared to sell it however less individuals will get it then, at that point, its costs go low. The unexpected expansion in the interest for the stock can be because of different reasons with optimistic news about the organization or a declaration from the organization.
After a timeframe when the interest for the stock disappears its costs gradually creep down as the financial backer drops curiosity in it. These stock costs working all over is an iterative interaction and rehashed. This instability of stock makes financial backers apprehensive while putting resources into an organization.[2]

We would attempt to investigate only a glimpse of something larger for the stock marketplace investigation as specialized examination of the stock is a huge arena.

---

[1] Bloomberg. "There Are Now More Indexes Than Stocks" - https://web.archive.org/web/20170602050244/https://www.bloomberg.com/news/articles/2017-05-12/there-are-now-more-indexes-than-stocks

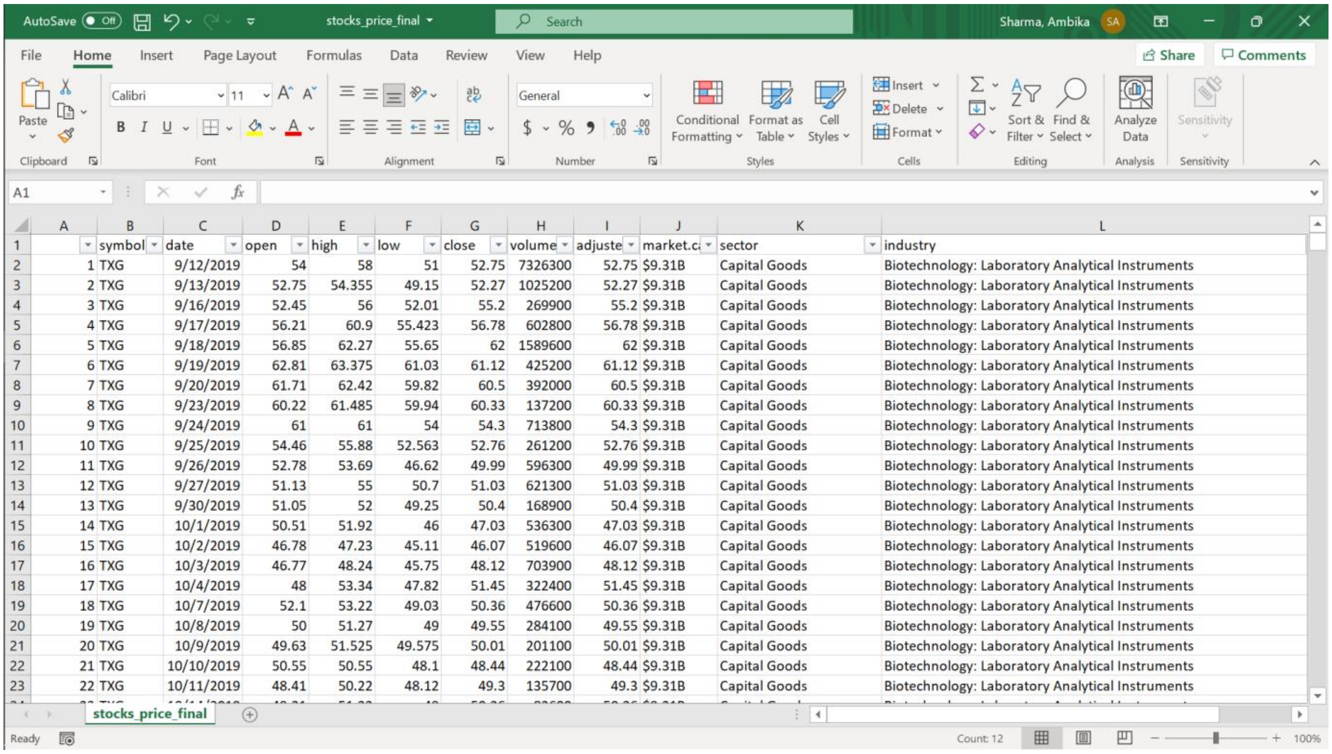
[2] Portfolio Project: Predicting Stock Prices Using Pandas and Scikit-learn - https://www.dataquest.io/blog/portfolio-project-predicting-stock-prices-using-pandas-and-scikit-learn/

DATA SET URL AND DATA SET DESCRIPTION

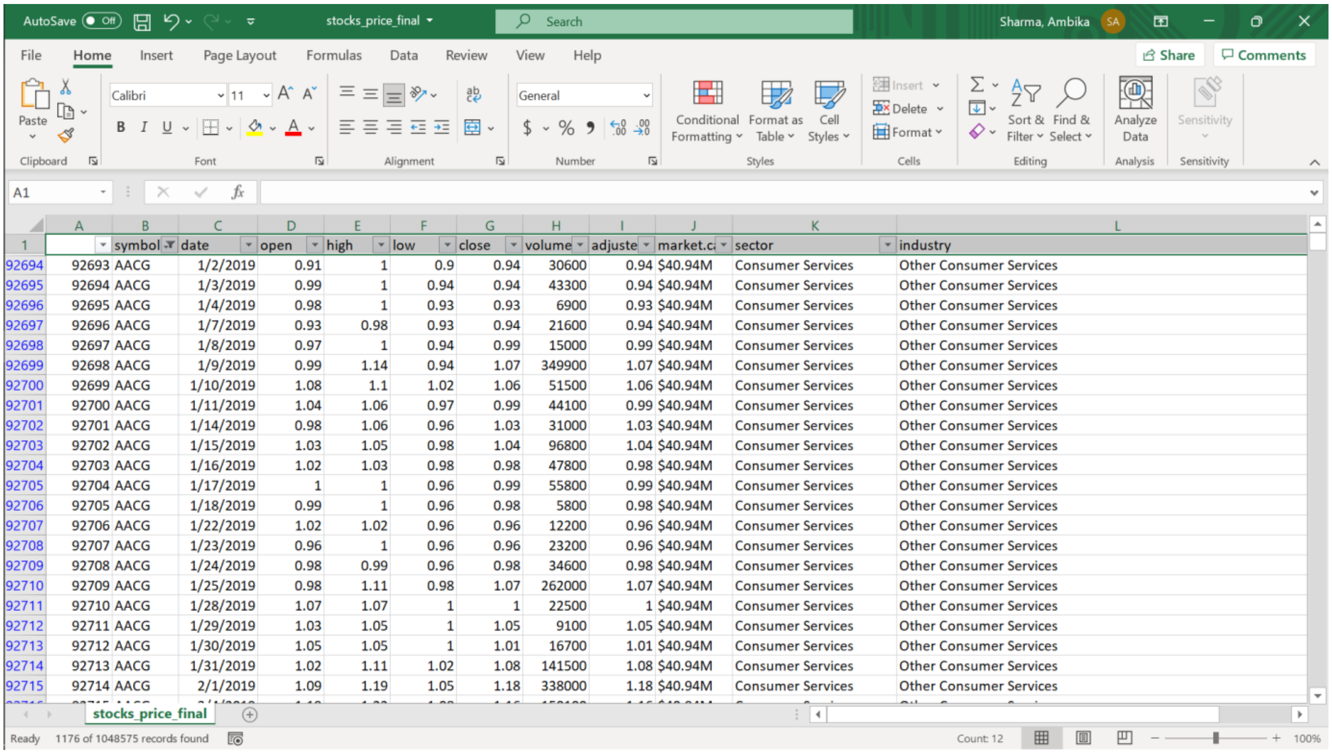
Data Set URL

https://www.kaggle.com/datasets/dinnymathew/usstockprices?select=stocks_price_final.csv

Number of columns – 13. Number of usable columns - 12



1		symbol	date	open	high	low	close	volume	adjuste	market.ca	sector	industry
2	1	TXG	9/12/2019	54	58	51	52.75	7326300	52.75	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
3	2	TXG	9/13/2019	52.75	54.355	49.15	52.27	1025200	52.27	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
4	3	TXG	9/16/2019	52.45	56	52.01	55.2	269900	55.2	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
5	4	TXG	9/17/2019	56.21	60.9	55.423	56.78	602800	56.78	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
6	5	TXG	9/18/2019	56.85	62.27	55.65	62	1589600	62	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
7	6	TXG	9/19/2019	62.81	63.375	61.03	61.12	425200	61.12	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
8	7	TXG	9/20/2019	61.71	62.42	59.82	60.5	392000	60.5	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
9	8	TXG	9/23/2019	60.22	61.485	59.94	60.33	137200	60.33	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
10	9	TXG	9/24/2019	61	61	54	54.3	713800	54.3	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
11	10	TXG	9/25/2019	54.46	55.88	52.563	52.76	261200	52.76	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
12	11	TXG	9/26/2019	52.78	53.69	46.62	49.99	596300	49.99	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
13	12	TXG	9/27/2019	51.13	55	50.7	51.03	621300	51.03	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
14	13	TXG	9/30/2019	51.05	52	49.25	50.4	168900	50.4	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
15	14	TXG	10/1/2019	50.51	51.92	46	47.03	536300	47.03	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
16	15	TXG	10/2/2019	46.78	47.23	45.11	46.07	519600	46.07	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
17	16	TXG	10/3/2019	46.77	48.24	45.75	48.12	703900	48.12	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
18	17	TXG	10/4/2019	48	53.34	47.82	51.45	322400	51.45	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
19	18	TXG	10/7/2019	52.1	53.22	49.03	50.36	476600	50.36	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
20	19	TXG	10/8/2019	50	51.27	49	49.55	284100	49.55	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
21	20	TXG	10/9/2019	49.63	51.525	49.575	50.01	201100	50.01	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
22	21	TXG	10/10/2019	50.55	50.55	48.1	48.44	222100	48.44	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments
23	22	TXG	10/11/2019	48.41	50.22	48.12	49.3	135700	49.3	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments



1		symbol	date	open	high	low	close	volume	adjuste	market.ca	sector	industry
92694	92693	AACG	1/2/2019	0.91	1	0.9	0.94	30600	0.94	$40.94M	Consumer Services	Other Consumer Services
92695	92694	AACG	1/3/2019	0.99	1	0.94	0.94	43300	0.94	$40.94M	Consumer Services	Other Consumer Services
92696	92695	AACG	1/4/2019	0.98	1	0.93	0.93	6900	0.93	$40.94M	Consumer Services	Other Consumer Services
92697	92696	AACG	1/7/2019	0.93	0.98	0.93	0.94	21600	0.94	$40.94M	Consumer Services	Other Consumer Services
92698	92697	AACG	1/8/2019	0.97	1	0.94	0.99	15000	0.99	$40.94M	Consumer Services	Other Consumer Services
92699	92698	AACG	1/9/2019	0.99	1.14	0.94	1.07	349900	1.07	$40.94M	Consumer Services	Other Consumer Services
92700	92699	AACG	1/10/2019	1.08	1.1	1.02	1.06	51500	1.06	$40.94M	Consumer Services	Other Consumer Services
92701	92700	AACG	1/11/2019	1.04	1.06	0.97	0.99	44100	0.99	$40.94M	Consumer Services	Other Consumer Services
92702	92701	AACG	1/14/2019	0.98	1.06	0.96	1.03	31000	1.03	$40.94M	Consumer Services	Other Consumer Services
92703	92702	AACG	1/15/2019	1.03	1.05	0.98	1.04	96800	1.04	$40.94M	Consumer Services	Other Consumer Services
92704	92703	AACG	1/16/2019	1.02	1.03	0.98	0.98	47800	0.98	$40.94M	Consumer Services	Other Consumer Services
92705	92704	AACG	1/17/2019	1	1	0.96	0.99	55800	0.99	$40.94M	Consumer Services	Other Consumer Services
92706	92705	AACG	1/18/2019	0.99	1	0.96	0.98	5800	0.98	$40.94M	Consumer Services	Other Consumer Services
92707	92706	AACG	1/22/2019	1.02	1.02	0.96	0.96	12200	0.96	$40.94M	Consumer Services	Other Consumer Services
92708	92707	AACG	1/23/2019	0.96	1	0.96	0.96	23200	0.96	$40.94M	Consumer Services	Other Consumer Services
92709	92708	AACG	1/24/2019	0.98	0.99	0.96	0.98	34600	0.98	$40.94M	Consumer Services	Other Consumer Services
92710	92709	AACG	1/25/2019	0.98	1.11	0.98	1.07	262000	1.07	$40.94M	Consumer Services	Other Consumer Services
92711	92710	AACG	1/28/2019	1.07	1.07	1	1	22500	1	$40.94M	Consumer Services	Other Consumer Services
92712	92711	AACG	1/29/2019	1.03	1.05	1	1.05	9100	1.05	$40.94M	Consumer Services	Other Consumer Services
92713	92712	AACG	1/30/2019	1.05	1.05	1	1.01	16700	1.01	$40.94M	Consumer Services	Other Consumer Services
92714	92713	AACG	1/31/2019	1.02	1.11	1.02	1.08	141500	1.08	$40.94M	Consumer Services	Other Consumer Services
92715	92714	AACG	2/1/2019	1.09	1.19	1.05	1.18	338000	1.18	$40.94M	Consumer Services	Other Consumer Services

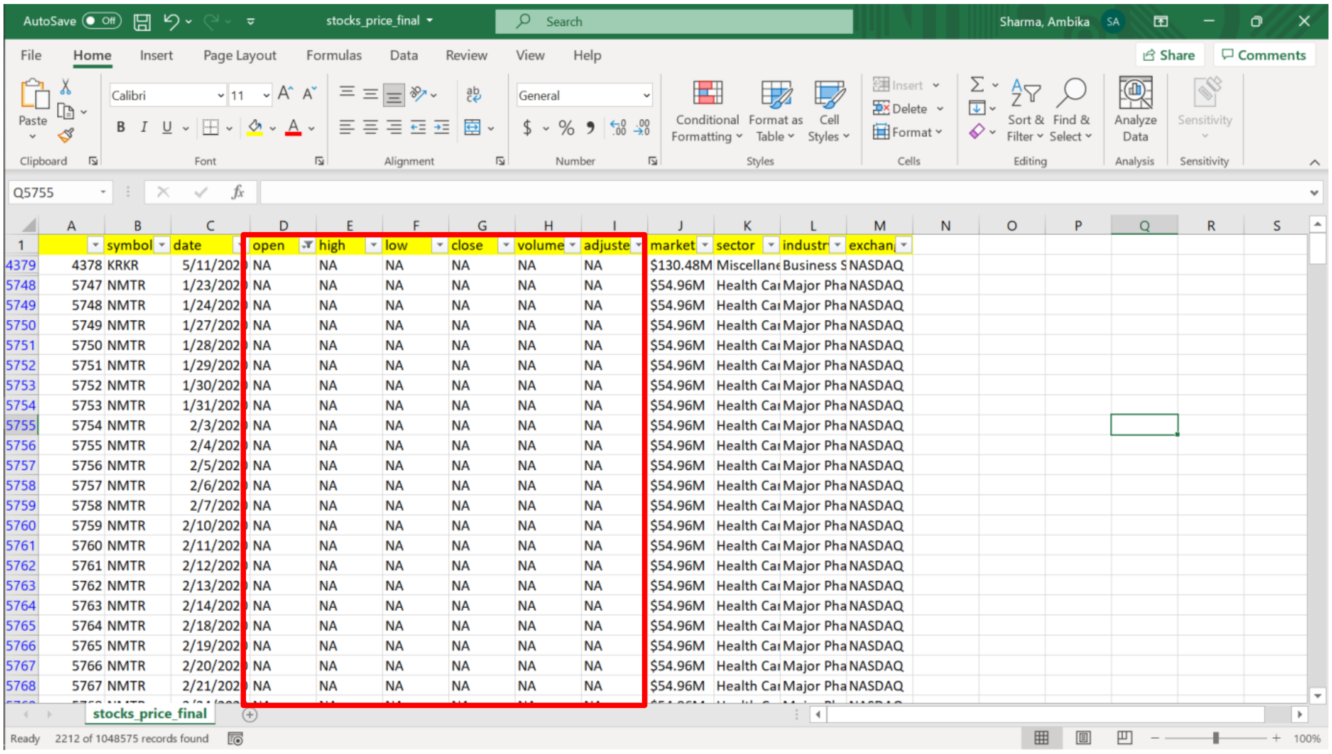
Ready 1176 of 1048575 records found

Data Set Description

Column Name	Description	Sample Data
Symbol	Symbols of the companies or organizations like FB- Facebook, TSLA- Tesla.	TXG
Date	The day of the stock when the market starts.	9/13/2019
Open	The value of the stock when the marketplace opens in the morning	52.75
High	Maximum worth the stock reached through that day	54.35
Low	Lowermost worth the stock is traded through the day	49.15
Close	The value of the stock when the marketplace locked in the evening.	52.27
Volume	The whole quantity of stocks traded on that day	1025200
Market.cap	Market cap measures what a corporation is worth on the open market, as well as the market's perception of its prospects, because it reflects what investors are willing to pay for its stock.	$9.31B
Sector	Huge section of the economy, that defines a wider aspect of industry.	Capital Goods
Industry	Group of corporations that are connected grounded on their chief commercial activities.	Apparel
Exchange	A market where securities, merchandises, derivatives, and other monetary gadgets are dealt.	NASDAQ

DATA CLEANING

Find NA



	A	symbol	date	open	high	low	close	volume	adjuste	market	sector	industr	exchan
4379	4378	KRKR	5/11/202	NA	NA	NA	NA	NA	NA	$130.48M	Miscellane	Business S	NASDAQ
5748	5747	NMTR	1/23/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5749	5748	NMTR	1/24/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5750	5749	NMTR	1/27/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5751	5750	NMTR	1/28/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5752	5751	NMTR	1/29/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5753	5752	NMTR	1/30/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5754	5753	NMTR	1/31/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5755	5754	NMTR	2/3/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5756	5755	NMTR	2/4/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5757	5756	NMTR	2/5/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5758	5757	NMTR	2/6/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5759	5758	NMTR	2/7/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5760	5759	NMTR	2/10/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5761	5760	NMTR	2/11/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5762	5761	NMTR	2/12/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5763	5762	NMTR	2/13/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5764	5763	NMTR	2/14/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5765	5764	NMTR	2/18/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5766	5765	NMTR	2/19/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5767	5766	NMTR	2/20/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ
5768	5767	NMTR	2/21/202	NA	NA	NA	NA	NA	NA	$54.96M	Health Ca	Major Pha	NASDAQ

stocks_price_final

Ready 2212 of 1048575 records found

In [10]:
```
df.isna()
```

Out[10]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
0	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...
1729029	False	False	False	False	False	False	False	False	False	False	False	False
1729030	False	False	False	False	False	False	False	False	False	False	False	False
1729031	False	False	False	False	False	False	False	False	False	False	False	False
1729032	False	False	False	False	False	False	False	False	False	False	False	False
1729033	False	False	False	False	False	False	False	False	False	False	False	False

1729034 rows × 12 columns

Before Data Cleaning:

In [11]:
```
df.isna().sum()
```

Out[11]:
```
symbol           0
date             0
open          2733
high          2733
low           2733
close         2733
volume        2733
adjusted      2733
market.cap       0
sector           0
industry         0
exchange         0
dtype: int64
```

After Data Cleaning:

```
In [20]: df = df.fillna(0)

In [21]: df.isna().any()

Out[21]: symbol        False
         date          False
         open          False
         high          False
         low           False
         close         False
         volume        False
         adjusted      False
         market.cap    False
         sector        False
         industry      False
         exchange      False
         dtype: bool

In [22]: df.isna().sum()

Out[22]: symbol        0
         date          0
         open          0
         high          0
         low           0
         close         0
         volume        0
         adjusted      0
         market.cap    0
         sector        0
         industry      0
         exchange      0
         dtype: int64
```

...	...	...	...	...	...	...	...	...	...	...	...	...
1595246	SOS	2020-05-19	0.0	0.0	0.0	0.0	0.0	0.0	$29.26M	Finance	Finance: Consumer Services	NYSE
1595247	SOS	2020-05-20	0.0	0.0	0.0	0.0	0.0	0.0	$29.26M	Finance	Finance: Consumer Services	NYSE
1595248	SOS	2020-05-21	0.0	0.0	0.0	0.0	0.0	0.0	$29.26M	Finance	Finance: Consumer Services	NYSE
1595249	SOS	2020-05-22	0.0	0.0	0.0	0.0	0.0	0.0	$29.26M	Finance	Finance: Consumer Services	NYSE
294496	MOHO	2020-05-11	0.0	0.0	0.0	0.0	0.0	0.0	$72.13M	Consumer Services	Catalog/Specialty Distribution	NASDAQ

df.isna()

df.isna().sum()

df = df.fillna(0)

df.isna().any()

df.isna().sum()

There was several "NA", once we explored the excel sheet for the columns open, high, low, close, volume and adjusted. To clean NA, we first determined NA in the columns, followed by replacing it with "0" so that our data for visualization is captured appropriately.

Removing $ from Column

Before Data Cleaning:

```
In [76]: df.head()
```

Out[76]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange	year
0	TXG	2019-09-12	54.00	58.00	51.00	52.75	7326300.0	52.75	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
1	TXG	2019-09-13	52.75	54.36	49.15	52.27	1025200.0	52.27	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
2	TXG	2019-09-16	52.45	56.00	52.01	55.20	269900.0	55.20	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
3	TXG	2019-09-17	56.21	60.90	55.42	56.78	602800.0	56.78	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
4	TXG	2019-09-18	56.85	62.27	55.65	62.00	1589600.0	62.00	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019

After Data Cleaning:

```
In [84]: df['market.cap'] = df['market.cap'].map(lambda x: x.lstrip('$'))
         df.head()
```

Out[84]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange	year
0	TXG	2019-09-12	54.00	58.00	51.00	52.75	7326300.0	52.75	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
1	TXG	2019-09-13	52.75	54.36	49.15	52.27	1025200.0	52.27	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
2	TXG	2019-09-16	52.45	56.00	52.01	55.20	269900.0	55.20	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
3	TXG	2019-09-17	56.21	60.90	55.42	56.78	602800.0	56.78	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
4	TXG	2019-09-18	56.85	62.27	55.65	62.00	1589600.0	62.00	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019

df.head()

df['market.cap'] = df['market.cap'].map(lambda x: x.lstrip('$'))

df.head()

For the column “Market.cap” it had both numbers and character in the string, but for the visualization purpose, it was important to change the values in this column to one single format, so the first step involves the **removal of “$” sign** from the column, so the data before cleaning contains $, whereas data after cleaning has removed $ from the column.

Replacing B and M with number of zeroes

Before Data Cleaning:

```
In [84]: df['market.cap'] = df['market.cap'].map(lambda x: x.lstrip('$'))
         df.head()
```

Out[84]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange	year
0	TXG	2019-09-12	54.00	58.00	51.00	52.75	7326300.0	52.75	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
1	TXG	2019-09-13	52.75	54.36	49.15	52.27	1025200.0	52.27	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
2	TXG	2019-09-16	52.45	56.00	52.01	55.20	269900.0	55.20	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
3	TXG	2019-09-17	56.21	60.90	55.42	56.78	602800.0	56.78	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
4	TXG	2019-09-18	56.85	62.27	55.65	62.00	1589600.0	62.00	9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019

After Data Cleaning:

```
In [31]: def value_to_float(x):
             if type(x) == float or type(x) == int:
                 return x
             if 'K' in x:
                 if len(x) > 1:
                     return float(x.replace('K', '')) * 1000
                 return 1000.0
             if 'M' in x:
                 if len(x) > 1:
                     return float(x.replace('M', '')) * 1000000
                 return 1000000.0
             if 'B' in x:
                 return float(x.replace('B', '')) * 1000000000
             return 0.0

         df['market.cap'] = df['market.cap'].apply(value_to_float)
         df['market.cap']
         df.head()
```

Out[31]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange	year
0	TXG	2019-09-12	54.00	58.00	51.00	52.75	7326300.0	52.75	9.310000e+09	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
1	TXG	2019-09-13	52.75	54.36	49.15	52.27	1025200.0	52.27	9.310000e+09	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
2	TXG	2019-09-16	52.45	56.00	52.01	55.20	269900.0	55.20	9.310000e+09	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
		2019-09-								Capital	Biotechnology: Laboratory Analytical		

```
def value_to_float(x):
    if type(x) == float or type(x) == int:
        return x
    if 'K' in x:
        if len(x) > 1:
            return float(x.replace('K', '')) * 1000
        return 1000.0
```

```
    if 'M' in x:
        if len(x) > 1:
            return float(x.replace('M', '')) * 1000000
        return 1000000.0
    if 'B' in x:
        return float(x.replace('B', '')) * 1000000000
    return 0.0
df['market.cap'] = df['market.cap'].apply(value_to_float)
df['market.cap']
df.head()
```

Once we remove the “$” sign from the market.cap, there were three other symbols to replace and re evaluate the value of this column. “M”, “B”, and “K” , the string is checked, and **every K is replaced by *1000, every M is replaced by *1000000 and every B is replaced by *1000000000.** This will provide us numeric value for the column that would be easy to use for visualization purpose.

Decimal to Fixed 2

Before Data Cleaning:

In [2]:
```
import pandas as pd
import numpy as np
df = pd.read_csv('stocks_price_final.csv')
df.head()
```

Out[2]:

	Unnamed: 0	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
0	1	TXG	2019-09-12	54.000000	58.000000	51.000000	52.750000	7326300.0	52.750000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
1	2	TXG	2019-09-13	52.750000	54.355000	49.150002	52.270000	1025200.0	52.270000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
2	3	TXG	2019-09-16	52.450001	56.000000	52.009998	55.200001	269900.0	55.200001	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
3	4	TXG	2019-09-17	56.209999	60.900002	55.423000	56.779999	602800.0	56.779999	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
4	5	TXG	2019-09-18	56.849998	62.270000	55.650002	62.000000	1589600.0	62.000000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ

After Data Cleaning:

```
In [12]: df = df.round({"open":2, "high":2, "low":2, "close":2, "volume":2, "adjusted":2})

In [13]: df.head()
Out[13]:
```

	Unnamed: 0	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
0	1	TXG	2019-09-12	54.00	58.00	51.00	52.75	7326300.0	52.75	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
1	2	TXG	2019-09-13	52.75	54.36	49.15	52.27	1025200.0	52.27	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
2	3	TXG	2019-09-16	52.45	56.00	52.01	55.20	269900.0	55.20	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
3	4	TXG	2019-09-17	56.21	60.90	55.42	56.78	602800.0	56.78	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
4	5	TXG	2019-09-18	56.85	62.27	55.65	62.00	1589600.0	62.00	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ

```
In [14]: df.tail()
Out[14]:
```

	Unnamed: 0	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
1729029	1729030	ZYME	2020-07-16	30.57	31.67	30.30	31.15	467900.0	31.15	$1.44B	Health Care	Major Pharmaceuticals	NYSE
1729030	1729031	ZYME	2020-07-17	31.20	33.08	31.00	33.03	600800.0	33.03	$1.44B	Health Care	Major Pharmaceuticals	NYSE
1729031	1729032	ZYME	2020-07-20	33.32	33.32	31.59	32.11	303500.0	32.11	$1.44B	Health Care	Major Pharmaceuticals	NYSE
1729032	1729033	ZYME	2020-07-21	32.37	32.49	30.34	30.65	337900.0	30.65	$1.44B	Health Care	Major Pharmaceuticals	NYSE
1729033	1729034	ZYME	2020-07-22	30.80	32.12	30.52	31.70	369900.0	31.70	$1.44B	Health Care	Major Pharmaceuticals	NYSE

df = df.round({"open":2, "high":2, "low":2, "close":2, "volume":2, "adjusted":2})

df.head()

df.tail()

Adjusting the data for “open, high, clow, close, volume and adjusted”. There is minimum 6 numbers after decimal before the data cleaning, once we use **round the decimal to 2**, we see the data cleaner and easier to predict.

Extracting Year from Date

Before Data Cleaning

```
Out[1]:
```

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
0	TXG	2019-09-12	54.000000	58.000000	51.000000	52.750000	7326300.0	52.750000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
1	TXG	2019-09-13	52.750000	54.355000	49.150002	52.270000	1025200.0	52.270000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
2	TXG	2019-09-16	52.450001	56.000000	52.009998	55.200001	269900.0	55.200001	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
3	TXG	2019-09-17	56.209999	60.900002	55.423000	56.779999	602800.0	56.779999	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
4	TXG	2019-09-18	56.849998	62.270000	55.650002	62.000000	1589600.0	62.000000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ

After Data Cleaning:

In [14]:
```python
df['date'] = pd.to_datetime(df['date'],format='%Y-%m-%d')
df['year'] = pd.DatetimeIndex(df['date']).year
```

In [15]:
```python
df.head()
```

Out[15]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange	year
0	TXG	2019-09-12	54.00	58.00	51.00	52.75	7326300.0	52.75	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
1	TXG	2019-09-13	52.75	54.36	49.15	52.27	1025200.0	52.27	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
2	TXG	2019-09-16	52.45	56.00	52.01	55.20	269900.0	55.20	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
3	TXG	2019-09-17	56.21	60.90	55.42	56.78	602800.0	56.78	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019
4	TXG	2019-09-18	56.85	62.27	55.65	62.00	1589600.0	62.00	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ	2019

df['date'] = pd.to_datetime(df['date'],format='%Y-%m-%d')

df['year'] = pd.DatetimeIndex(df['date']).year

Splitting the date column, separating the year from date, and creating a new column under year, to keep the track of how the stock prices has change throughout the year rather than keeping count of dates.

Sorting

Before Data Cleaning:

In [6]:
```python
df.head()
```

Out[6]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
0	TXG	2019-09-12	54.00	58.00	51.00	52.75	7326300.0	52.75	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
1	TXG	2019-09-13	52.75	54.36	49.15	52.27	1025200.0	52.27	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
2	TXG	2019-09-16	52.45	56.00	52.01	55.20	269900.0	55.20	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
3	TXG	2019-09-17	56.21	60.90	55.42	56.78	602800.0	56.78	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
4	TXG	2019-09-18	56.85	62.27	55.65	62.00	1589600.0	62.00	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ

After Data Cleaning:

In [13]:
```python
df.sort_values(by='open', ascending=False)
```

Out[13]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
1682069	VHI	2019-02-25	160168176.0	161601456.0	151210224.0	158376592.0	0.0	157249392.0	$320.29M	Basic Industries	Major Chemicals	NYSE
1682070	VHI	2019-02-26	159451552.0	160168176.0	155151728.0	156943312.0	0.0	155826304.0	$320.29M	Basic Industries	Major Chemicals	NYSE
1682071	VHI	2019-02-27	157659952.0	157659952.0	139744048.0	145835456.0	0.0	144797504.0	$320.29M	Basic Industries	Major Chemicals	NYSE
1682078	VHI	2019-03-08	150493600.0	150851904.0	142252272.0	144760512.0	0.0	143731152.0	$320.29M	Basic Industries	Major Chemicals	NYSE
1682073	VHI	2019-03-01	146193776.0	148343680.0	135085920.0	140819008.0	0.0	139816768.0	$320.29M	Basic Industries	Major Chemicals	NYSE

df.sort_values(by='open', ascending=False)

Dropping Columns

Before Data Cleaning:

```
In [2]: import pandas as pd
        import numpy as np
        df = pd.read_csv('stocks_price_final.csv')
        df.head()
```

Out[2]:

	Unnamed: 0	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
0	1	TXG	2019-09-12	54.000000	58.000000	51.000000	52.750000	7326300.0	52.750000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
1	2	TXG	2019-09-13	52.750000	54.355000	49.150002	52.270000	1025200.0	52.270000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
2	3	TXG	2019-09-16	52.450001	56.000000	52.009998	55.200001	269900.0	55.200001	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
3	4	TXG	2019-09-17	56.209999	60.900002	55.423000	56.779999	602800.0	56.779999	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
4	5	TXG	2019-09-18	56.849998	62.270000	55.650002	62.000000	1589600.0	62.000000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ

After Data Cleaning:

```
In [3]: to_drop = ['Unnamed: 0']
        df.drop(to_drop, inplace=True, axis=1)
        df.head()
```

Out[3]:

	symbol	date	open	high	low	close	volume	adjusted	market.cap	sector	industry	exchange
0	TXG	2019-09-12	54.000000	58.000000	51.000000	52.750000	7326300.0	52.750000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
1	TXG	2019-09-13	52.750000	54.355000	49.150002	52.270000	1025200.0	52.270000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
2	TXG	2019-09-16	52.450001	56.000000	52.009998	55.200001	269900.0	55.200001	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
3	TXG	2019-09-17	56.209999	60.900002	55.423000	56.779999	602800.0	56.779999	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ
4	TXG	2019-09-18	56.849998	62.270000	55.650002	62.000000	1589600.0	62.000000	$9.31B	Capital Goods	Biotechnology: Laboratory Analytical Instruments	NASDAQ

```
import pandas as pd

import numpy as np

df = pd.read_csv('stocks_price_final.csv')

df.head()

to_drop = ['Unnamed: 0']

df.drop(to_drop, inplace=True, axis=1)

df.head()
```

Dropping an unnamed column that was just keeping the count of the serial number of the records.

APPLY/ SHOW SUMMARY STATISTICS

Open – Column

```
In [41]: df['open'].describe().round(1)
Out[41]: count      1729034.0
         mean         15046.3
         std        1110942.9
         min              0.0
         25%              7.0
         50%             18.3
         75%             44.6
         max      160168176.0
         Name: open, dtype: float64
```

```
In [42]: df['open'].mean()

Out[42]: 15046.251172215247
```

```
In [19]: df['open'].std()
Out[19]: 1110942.9122433085
```

```
In [45]: df['open'].mode()
Out[45]: 0    0.0
         dtype: float64
```


```
In [45]: df['open'].mode()
Out[45]: 0    0.0
         dtype: float64
```


```
In [43]: df['open'].min()
Out[43]: 0.0
```

```
In [44]: df['open'].max()
Out[44]: 160168176.0
```

```
In [46]: df['open'].median()
Out[46]: 18.33
```

df['open'].describe().round(1)

df['open'].mean()

```
df['open'].std()

df['open'].mode()

df['open'].min()

df['open'].max()

df['open'].median()
```

The describe function along with the round is used to round the values after the decimal for the describe function for opening rate of stock for one day. The **mean** is the average of the opening values which is the sum of all values divided by the total number of days that comes out to be **15046.25.**

The **standard deviation** defines in what way the data is spread out around the mean, for opening values it comes out to be **1110942.91**. The most common occurrence seems to be "**0**" for opening values which is the **mode**. The minimum value for the open values is "**0**". If minimum value is extremely small, even when you contemplate the midpoint, the spread, and the outline of the statistics, examine the source of the extreme value. The maximum value for open is "**160168176**" and the median for the open value is "**18.33**". These statistical values give us an idea whether our data is symmetrical or not. As the value of mean and median is not similar the data seems to be asymmetric.

Close – Column

```
In [47]: df['close'].describe().round(1)

Out[47]: count      1729034.0
         mean         15009.0
         std        1108878.7
         min              0.0
         25%              7.0
         50%             18.3
         75%             44.6
         max      158376592.0
         Name: close, dtype: float64

In [48]: df['close'].mean()

Out[48]: 15008.953372698428

In [49]: df['close'].std()

Out[49]: 1108878.6742145666

In [50]: df['close'].mode()

Out[50]: 0    0.0
         dtype: float64

In [51]: df['close'].min()

Out[51]: 0.0

In [52]: df['close'].max()

Out[52]: 158376592.0

In [53]: df['close'].median()

Out[53]: 18.32
```

df['close'].describe().round(1)

df['close'].mean()

df['close'].std()

df['close'].mode()

df['close'].min()

df['close'].max()

df['close'].median()

The describe function along with the round is used to round the values after the decimal for the describe function for closing rate of stock for one day. The **mean** is the average of the closing values which is the sum of all values divided by the total number of days that comes out to be **15008.95.**

The **standard deviation** defines in what way the data is spread out around the mean, for closing values it comes out to be **1100878.67**. The most common occurrence seems to be "**0**" for opening values which is the **mode**. The minimum value for the close values is "**0**". If minimum value is extremely small, even when you contemplate the midpoint, the spread, and the outline of the statistics, examine the source of the extreme value. The maximum value for close is "**158376592**" and the median for the close value is "**18.32**". These statistical values give us an idea whether our data is symmetrical or not. As the value of mean and median is not similar the data seems to be asymmetric.

High – Column

```
In [54]: df['high'].describe().round(1)

Out[54]: count      1729034.0
         mean         15530.5
         std        1147339.5
         min              0.0
         25%              7.2
         50%             18.7
         75%             45.4
         max      161601456.0
         Name: high, dtype: float64

In [55]: df['high'].mean()

Out[55]: 15530.480131645862

In [56]: df['high'].std()

Out[56]: 1147339.5130791045

In [57]: df['high'].mode()

Out[57]: 0    0.0
         dtype: float64

In [58]: df['high'].min()

Out[58]: 0.0
```

```
In [59]: df['high'].max()
Out[59]: 161601456.0

In [60]: df['high'].median()
Out[60]: 18.7
```

df['high'].describe().round(1)

df['high'].mean()

df['high'].std()

df['high'].mode()

df['high'].min()

df['high'].max()

df['high'].median()

The describe function along with the round is used to round the values after the decimal for the describe function for high rate of stock for one day. The **mean** is the average of the high values which is the sum of all values divided by the total number of days that comes out to be **15530.48.**

The **standard deviation** defines in what way the data is spread out around the mean, for high values it comes out to be **114739.31**. The most common occurrence seems to be "**0**" for high values which is the **mode**. The minimum value for the high values is "**0**". If minimum value is extremely small, even when you contemplate the midpoint, the spread, and the outline of the statistics, examine the source of the extreme value. The maximum value for open is "**161601456**" and the median for the open value is "**18.7**". These statistical values give us an idea whether our data is symmetrical or not. As the value of mean and median is not similar the data seems to be asymmetric.

ANALYSIS & VISUALIZATION

List the sectors that cover the trade market

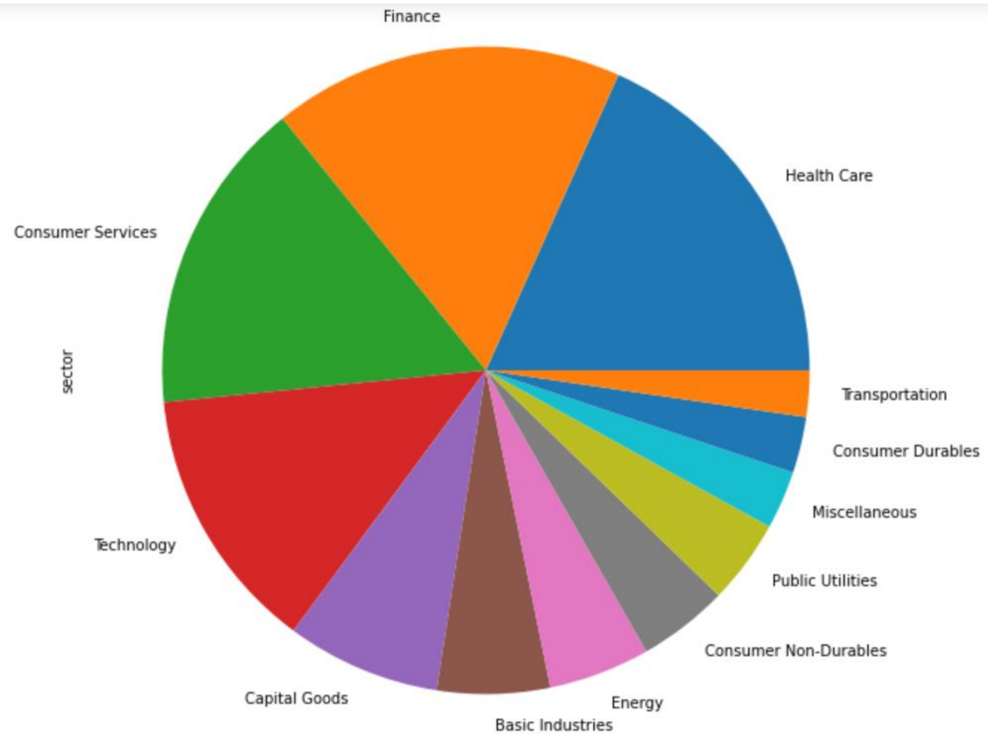
Q1- List down the sectors that cover the stock market.

```
In [69]: sector_count = df['sector'].value_counts()
         sector_count

Out[69]: Health Care               316175
         Finance                   303180
         Consumer Services         272393
         Technology                229799
         Capital Goods             133122
         Basic Industries           97323
         Energy                     87494
         Consumer Non-Durables      78080
         Public Utilities           72836
         Miscellaneous              50221
         Consumer Durables          48404
         Transportation             40007
         Name: sector, dtype: int64
```

```
In [70]: sector_count.plot(kind='pie', figsize=[10,10])

Out[70]: <AxesSubplot:ylabel='sector'>
```



```
sector_count = df['sector'].value_counts()

sector_count

sector_count.plot(kind='pie', figsize=[10,10])
```

There are total 12 sectors which cover the stock market, as per our data. **Healthcare** sector being the major one with the count of **316175**. It consists of shares of corporations involved in a range of health-related businesses, from medicinal creators to health devices and health care facility providers, as well as biotech shares and insurance corporations. Examples of great healthcare businesses contain UnitedHealth Group (UNH) and Pfizer (PFE). Followed by the **Finance** sector which includes an extensive variety of economic corporations, from commercial banks to investment banks, coverage companies, and economical service benefactors, as well as asset organization businesses and financial advisors. The financial sector includes some of the largest economical organization in the world like Bank of America (BAC),Visa (V), and JPMorgan Chase (JPM). Followed by the **Consumer Services** which involves luxury goods, companies that deliver clients with utility services, such as gas, ecommerce, water, electric, hotel, retail, and the vacation and travel businesses.

Technology sector includes several industries and sub-sectors, from semi-conductor creators to computer software and computer hardware providers, as well as internet shares and cloud computing. The sector includes companies with some of the major marketplace capitalizations in the world, such as Microsoft (MSFT), Facebook (FB), Apple (AAPL), and Amazon (AMZN).

Transportation is the **least count** sector, that has least coverage in the expanse with some “40007” count throughout the market.

These observation are from year 2019 to 2020.

List the top 5 industries which has highest volume

Q2- List down the top 5 industries with has the highest volume in stock market.

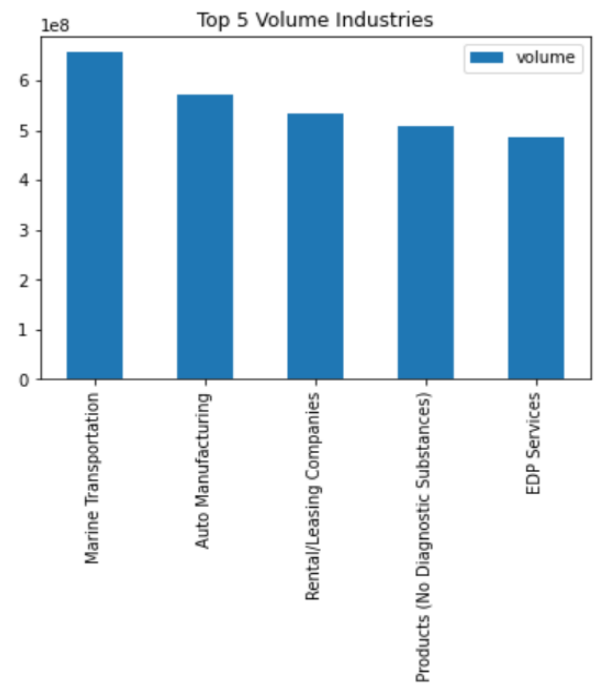
In [31]:
```python
# Grab the `Industry` and `Volume` columns
volume = df.loc[:, ['industry', 'volume']]
# Set the 'Indusrty' as the index
volume.set_index(volume['industry'], inplace=True)
# Drop the extra 'Industry' column
volume.drop(columns=['industry'], inplace=True)
# Filter down to 5 companies with the largest volume
top_5_volume = volume.nlargest(5, 'volume')
# Display the DataFrame
top_5_volume
```

Out[31]:

industry	volume
Marine Transportation	656504200.0
Auto Manufacturing	570911400.0
Rental/Leasing Companies	533891800.0
Biotechnology: Biological Products (No Diagnostic Substances)	507617300.0
EDP Services	486194300.0

In [29]:
```python
top_5_volume.plot(kind='bar', title='Top 5 Volume Industries')
```

Out[29]: <AxesSubplot:title={'center':'Top 5 Volume Industries'}, xlabel='industry'>



Grab the `Industry` and `Volume` columns

volume = df.loc[:, ['industry', 'volume']]

```
# Set the 'Indusrty' as the index
volume.set_index(volume['industry'], inplace=True)
# Drop the extra 'Industry' column
volume.drop(columns=['industry'], inplace=True)
# Filter down to 5 companies with the largest volume
top_5_volume = volume.nlargest(5, 'volume')
# Display the DataFrame
top_5_volume
top_5_volume.plot(kind='bar', title='Top 5 Volume Industries')
```

The graph above states the **highest volume** which is **656504200** for the "**Marine Transportation Industry**". Marine Transportation subdivision corporations have their shares listed and available for trading on the various stock exchanges of the U.S, there are several subcategories of this industry and hence it has one of the highest volume of stock in this category.

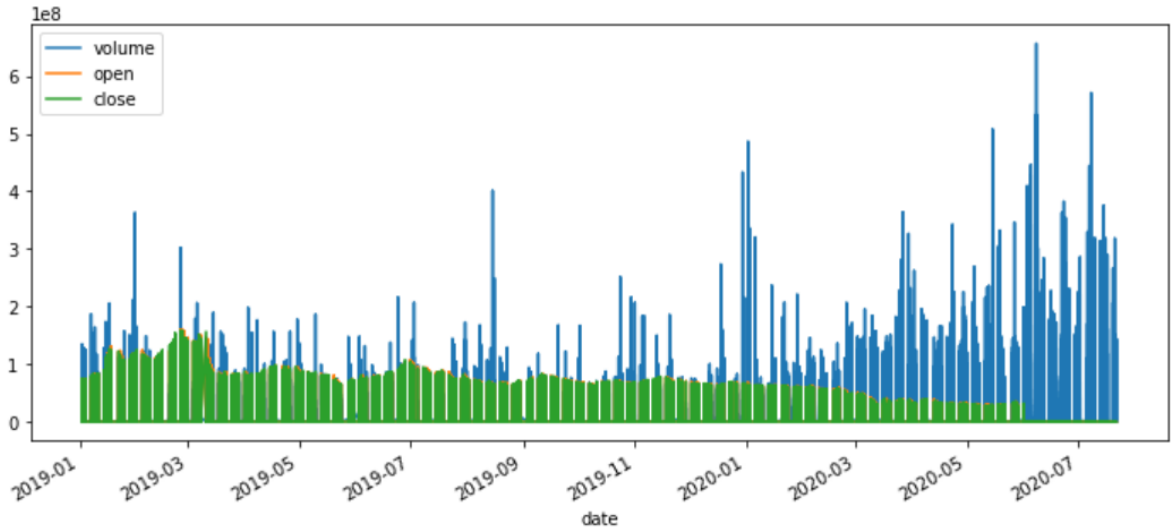
The other four industries which made up to the first five categories – **"Auto Manufacturing", "Rental/Leasing Companies", "Products" and "EDP services".**

These observation are from year 2019 to 2020.

Compare the open, close and volume of the stock market

Q3- Compare the opening, closing prices and volume of the stock market.

```
In [78]: df[['volume','open','close']].plot(subplots=False,figsize=(12,5))
Out[78]: <AxesSubplot:xlabel='date'>
```



df[['volume','open','close']].plot(subplots=False,figsize=(12,5))

The observation from year 2019 to 2020, shows the opening, closing prices and the volume of the market.

We see the **volume is highest during the months of June and July for the year 2020.**

The recorded closing cost is the last cost anybody paid for a portion of that stock during the business hours of the trade where the stock exchanges. The initial cost (open cost) is the cost of the main exchange of a workday. In some cases, these costs are unique. During an ordinary exchanging day, the harmony among market interest vacillates as the appeal of the stock's cost increments and diminishes. These variances are the reason for shutting, and they are not generally indistinguishable from open costs. In the hours between the end ringer and the accompanying exchanging day's initial chime, a few elements can influence the allure of a specific stock.

Volume is a marker that implies the sum of the number of offers that have been traded in a particular timeframe or during the exchanging day. It will likewise include the trading of every offer during a specific time span. Volume assists a few financial backers with dissecting the patterns and examples in the offer market. Whether a financial backer is discussing a whole securities exchange or portions of a singular stock, the data on volume can be found in any place.

The volume additionally implies the absolute number of offers made an into move during the exchanging day, whether they have a purchase or a sell request.

In this manner, on the off chance that stocks are effectively exchanged in the financial exchange, the volume is high, and on the off chance that stocks are not effectively exchanged, the volume is low.

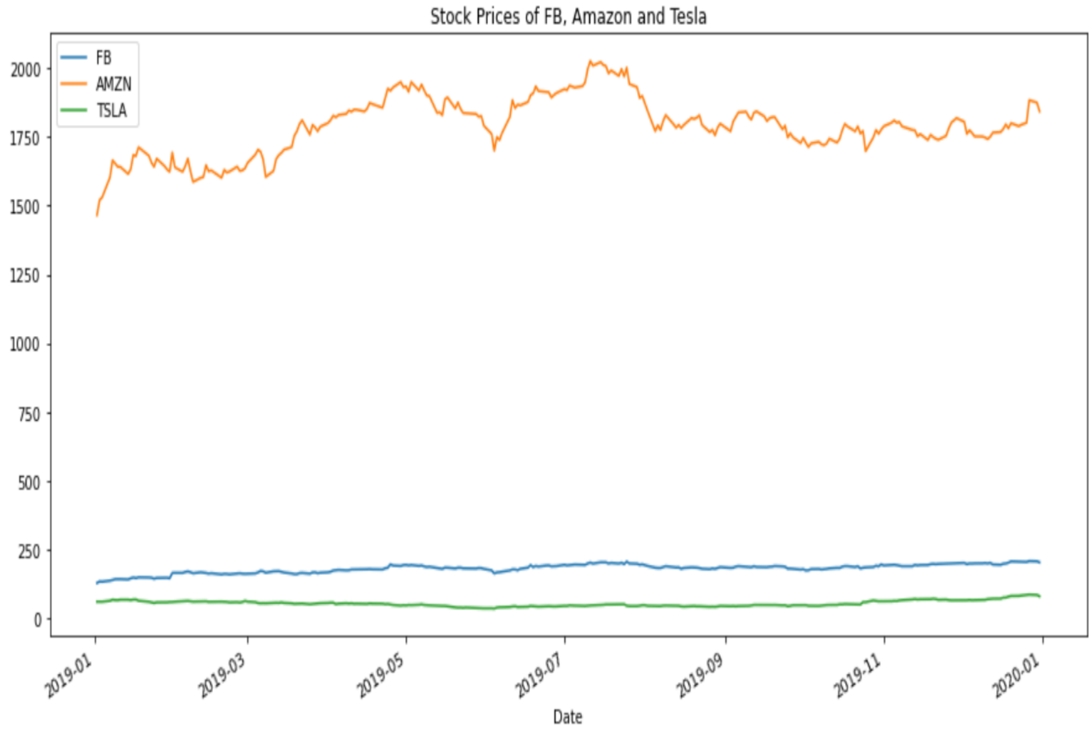
Compare the Opening Price of Amazon , Tesla, and FB for the year 2020 and 2019

Q4- Compare the opening price of Amazon, Tesla, and FB for the year 2020 and 2019.

In [58]:

```
import pandas as pd
import datetime
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
!pip install yfinance
import yfinance as yf

%matplotlib inline
start = "2019-01-01"
end = '2020-1-01'
fb = yf.download('FB',start,end)
amzn = yf.download('AMZN',start,end)
tsla = yf.download('TSLA',start,end)
fb['Open'].plot(label = 'FB', figsize = (15,7))
amzn['Open'].plot(label = "AMZN")
tsla['Open'].plot(label = 'TSLA')
plt.title('Stock Prices of FB, Amazon and Tesla')
plt.legend()
```




```
import datetime

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from pandas.plotting import scatter_matrix

!pip install yfinance

import yfinance as yf

%matplotlib inline

start = "2019-01-01"

end = '2020-1-01'

fb = yf.download('FB',start,end)

amzn = yf.download('AMZN',start,end)

tsla = yf.download('TSLA',start,end)
```

```
fb['Open'].plot(label = 'FB', figsize = (15,7))

amzn['Open'].plot(label = "AMZN")

tsla['Open'].plot(label = 'TSLA')

plt.title('Stock Prices of FB, Amazon and Tesla')

plt.legend()
```

For the above plot, we used the "**Yahoo finance API functionality".** We found out that it is possible for us to find the dynamic data for the stock market using the API downloading it and plotting the graph accordingly.

The API serves ongoing and authentic information for crypto and trade markets. It gives broad monetary information to public organizations, shared reserves, etc., securities, digital forms of money, and public monetary forms, including choice chains and market examination.

The chart is the depiction of exposed trade amounts for three companies via line diagram by leveraging matplotlib collection in python. The chart portrays that the opening price for **Amazon** is more when comparing it to other two companies. These stock varies drastically with time.

Tesla has the lowest opening prices in the year 2019 to 2020, followed by Facebook.

Amazon boom due to – online retail high pricing, during pandemic there was a whole new generation doing online shopping, that raised the market price for Amazon.

Obviously, the pandemic has pushed an entirely different age of customers on the web. Amazon may be predominant on the internet-based retail world, however before the pandemic around 90% of retail occurred face to face. Amazon was developing quickly, however, the move online was all the while going at its own place. The pandemic, notwithstanding, logical pushed an entire host of new customers to begin investigating web-based looking for their retail arrangements. Contenders like Walmart are yet playing to get up to speed, so getting the time span to speed up just plays for Amazon's potential benefit.

CONCLUSION

This article covers the initial step of trade exchange investigation which is making the analytic dataset. The above various investigation can be utilized to perceive a stock's present moment and long-haul conduct. A choice emotionally supportive network can be made which stock to pick from industry for generally safe low addition or high-risk high increase contingent upon the gamble apatite of the financial backer. A financial exchange is where individuals go to exchange stocks. Two of the main stock trades in the United States are the **NYSE and Nasdaq**.

Stocks are unstable. Costs change as indicated by market interest. Many individuals have various sentiments on why stock costs move in the way they do. One of the main factors that impact costs is profit. Figuring out how to peruse stock tables or a stock statement is an unquestionable necessity on the off chance that you are intending to be thoughtful financial backer in stocks. It isn't difficult to peruse a stock statement once you know what the various terms and images represent. Acquisition of stocks is generally done through a financier. You can likewise get a profit reinvestment plan.

Continuously recall the old financial exchange saying:

"***Bulls bring in cash, bears bring in cash, yet pigs get butchered!***".

This will maybe save you commonly from losing on your venture.