



### Experiment 3

**Aim:** Visualize and prepare data analysis on stock data prices, volume, and moving averages using a web API-Alpha Vantage

**Theory:** Alpha Vantage offers free stock APIs in JSON and CSV formats for realtime and historical stock market data, forex, commodity, cryptocurrency feeds and over. One of the main investing principles is that past performance even if it is not an indicator of future performance. However, it is good to look at the historical price and volume charts to get a sense of the range a stock has been trading in, notice trends and patterns, and locate the price levels at which investors are particularly active.

Alpha Vantage provides enterprise-grade financial market data through a set of powerful and developer-friendly data APIs and spreadsheets. From traditional asset classes (e.g., stocks, ETFs, mutual funds) to economic indicators, from foreign exchange rates to commodities, from fundamental data to technical indicators, Alpha Vantage is your one-stop-shop for real-time and historical global market data delivered through cloud-based APIs, Excel, and Google Sheets. the Alpha Academy is an open knowledge base for essential content in quantitative investing, machine learning, software development, block chain technologies and more, all delivered to you by industry experts.

#### **TIME\_SERIES\_INTRADAY Trending**

This API returns intraday time series of the equity specified, covering extended trading hours where applicable (e.g., 4:00am to 8:00pm Eastern Time for the US market). The intraday data is derived from the Securities Information Processor (SIP) market-aggregated data. You can query both raw (as-traded) and split/dividend-adjusted intraday data from this endpoint.

This API returns the most recent 1-2 months of intraday data and is best suited for short-term/medium-term charting and trading strategy development. If you are targeting a deeper intraday history, please use the Extended Intraday API.

#### **Lab Experiment to be done by students:**

1. Get free API key by filling out a couple of fields on this link:  
<https://www.alphavantage.co/support/#api-key>
2. Get Historical Stock Price and Volume Data from Web API alpha Vantage
3. Set specific Time Periods for given stock data.
4. Analyse Stock Splits and Dividends
5. Create a DatetimeIndex and year wise analysis
6. Perform Frequency Settings (Intraday) at various time intervals



**SHRI VILEPARLE KELAVANI MANDAL'S  
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**  
(Autonomous College Affiliated to the University of Mumbai)  
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)



**A.Y.: 2023-24**

**Class/Sem: T.Y.B.Tech/ Sem-VI**

**Sub: FMC**

7. Plot bands using Technical Indicators
8. Implement foreign exchange Currencies / FX
9. Display daily digital Cryptocurrencies
10. Plot any technical indicator you are interested in or zoom into the specific period.  
Moving averages smooth out the sharp increases or decreases, with 50, 100, and 200-day averages.



**SHRI VILEPARLE KELAVANI MANDAL'S  
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**  
(Autonomous College Affiliated to the University of Mumbai)  
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)



A.Y.: 2023-24

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: FMC

```
!pip install alpha_vantage

Collecting alpha_vantage
  Downloading alpha_vantage-2.3.1-py3-none-any.whl (31 kB)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from alpha_vantage) (3.9.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from alpha_vantage) (2.31.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->alpha_vantage) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->alpha_vantage) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->alpha_vantage) (1.4.1)
Requirement already satisfied: multidict>=4.0.4 in /usr/local/lib/python3.10/dist-packages (from aiohttp->alpha_vantage) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->alpha_vantage) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->alpha_vantage) (4.0.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->alpha_vantage) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->alpha_vantage) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->alpha_vantage) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->alpha_vantage) (2024.2.2)
Installing collected packages: alpha_vantage
Successfully installed alpha_vantage-2.3.1

[2] !pip install yfinance

Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-packages (0.2.36)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.5.3)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.25.2)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.31.0)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.10/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.9.4)
Requirement already satisfied: appdirs>=1.4.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.4.4)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2023.4)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.4.0)
Requirement already satisfied: numba>=0.3.16 in /usr/local/lib/python3.10/dist-packages (from yfinance) (0.57.1)
1s completed at 7:45AM

[3] import pandas as pd
from alpha_vantage.timeseries import TimeSeries

[4] mykey= 'CVQ3EZOTRZZCDR80'

[5] ts=TimeSeries(key=mykey, output_format='pandas')

[6] GE = ts.get_daily("GE")

[7] type(GE)

tuple

[8] GE

(
      date      1. open  2. high   3. low  4. close   5. volume
2024-02-21  148.47  149.390  147.660   149.07  4194372.0
2024-02-20  149.02  150.340  147.815   148.62  5719073.0
2024-02-16  148.95  150.360  148.510   149.16  6876932.0
2024-02-15  147.00  148.890  146.785   148.37  6885487.0
2024-02-14  143.00  146.900  142.350   146.50  10467749.0
...
2023-10-04  107.97  109.210  107.360   108.85  3891936.0
2023-10-03  108.31  109.990  107.040   107.76  6883407.0
2023-10-02  110.45  111.200  108.100   108.79  4301080.0
2023-09-29  112.63  112.750  109.910   110.55  3929920.0
)

[8] [100 rows x 5 columns],
{'1. Information': 'Daily Prices (open, high, low, close) and Volumes',
 '2. Symbol': 'GE',
 '3. Last Refreshed': '2024-02-21',
 '4. Output Size': 'Compact',
 '5. Time Zone': 'US/Eastern'})

[9] len(GE)

2

[10] GE[0]

      1. open  2. high   3. low  4. close   5. volume
date
2024-02-21  148.47  149.390  147.660   149.07  4194372.0
2024-02-20  149.02  150.340  147.815   148.62  5719073.0
2024-02-16  148.95  150.360  148.510   149.16  6876932.0
2024-02-15  147.00  148.890  146.785   148.37  6885487.0
2024-02-14  143.00  146.900  142.350   146.50  10467749.0
...
2023-10-04  107.97  109.210  107.360   108.85  3891936.0
2023-10-03  108.31  109.990  107.040   107.76  6883407.0
1s completed at 7:45AM
```



SHRI VILEPARLE KELAVANI MANDAL'S  
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING  
(Autonomous College Affiliated to the University of Mumbai)  
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)



A.Y.: 2023-24

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: FMC

```
[11] GE[1]
{'1. Information': 'Daily Prices (open, high, low, close) and Volumes',
 '2. Symbol': 'GE',
 '3. Last Refreshed': '2024-02-21',
 '4. Output Size': 'Compact',
 '5. Time Zone': 'US/Eastern'}
```

```
[12] GE= ts.get_daily("GE",outputsize="full")[0]
```

```
[13] GE.head()
```

	1. open	2. high	3. low	4. close	5. volume
date					
2024-02-21	148.47	149.39	147.660	149.07	4194372.0
2024-02-20	149.02	150.34	147.815	148.62	5719073.0
2024-02-16	148.95	150.36	148.510	149.16	6876932.0
2024-02-15	147.00	148.89	146.785	148.37	6885487.0
2024-02-14	143.00	146.90	142.350	146.50	10467749.0

```
[14] GE.tail()
```

	1. open	2. high	3. low	4. close	5. volume
date					
1999-11-05	133.19	134.81	133.19	133.75	4688900.0
1999-11-04	132.50	133.56	130.50	131.88	4353600.0
1999-11-03	132.88	132.94	130.00	131.38	4589000.0
1999-11-02	129.69	133.13	128.19	129.00	6340600.0
1999-11-01	133.63	134.38	129.25	129.38	6795500.0

```
[15] GE= ts.get_daily_adjusted("GE",outputsize="full")[0]
```

```
ValueError                                Traceback (most recent call last)
<ipython-input-15-3937514d6d42> in <cell line: 1>()
----> 1 GE= ts.get_daily_adjusted("GE",outputsize="full")[0]
```

```
↕ 2 frames ↕
/usr/local/lib/python3.10/dist-packages/alpha_vantage/alphavantage.py in _handle_api_call(self, url)
    359         raise ValueError(json_response["Error Message"])
    360     elif "Information" in json_response and self.treat_info_as_error:
--> 361         raise ValueError(json_response["Information"])
    362     elif "Note" in json_response and self.treat_info_as_error:
    363         raise ValueError(json_response["Note"])

ValueError: Thank you for using Alpha Vantage! This is a premium endpoint. You may subscribe to any of the premium plans at https://www.alphavantage.co/premium/ to instantly unlock all premium endpoints
```



✓ [16] GE.info()

0s

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 6115 entries, 2024-02-21 to 1999-11-01
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   1. open      6115 non-null   float64
1   2. high      6115 non-null   float64
2   3. low       6115 non-null   float64
3   4. close     6115 non-null   float64
4   5. volume    6115 non-null   float64
dtypes: float64(5)
memory usage: 286.6 KB
```

✓ GE.loc["2017"]

0s



1. open 2. high 3. low 4. close 5. volume



date



2017-12-29	17.27	17.530	17.27	17.45	75906686.0
2017-12-28	17.35	17.400	17.25	17.36	60756258.0
2017-12-27	17.46	17.630	17.31	17.38	58655208.0
2017-12-26	17.45	17.660	17.40	17.43	55337900.0
2017-12-22	17.51	17.560	17.40	17.50	46370400.0





✓  
1s

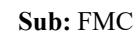
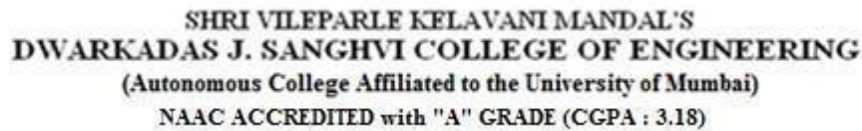
```
[18] ticker="GE"  
GEwa= ts.get_weekly_adjusted(ticker)[0]
```


✓  
0s


```
▶ GEwa.iloc[:, -1].value_counts()
```

```
⇒ 0.0000    1170  
   0.0100     11  
   0.0800     10  
   0.2300      8  
   0.2200      8  
   0.1900      8  
   0.3100      6  
   0.1200      5  
   0.1000      5  
   0.2400      4  
   0.1800      4  
   0.1700      4  
   0.1600      4  
   0.2800      4  
   0.2500      4  
   0.2000      4  
   0.1367      2  
   0.1500      2  
   0.1400      2  
   0.4100      2  
   0.4203      1
```

```
Name: 7. dividend amount, dtype: int64
```



0s  GEwa



	1. open	2. high	3. low	4. close	5. adjusted close	6. volume	7. dividend amount
date							
2024-02-21	133.03	150.36	132.47	149.07	149.0700	76849143.0	0.00
2024-01-31	127.01	134.47	123.65	132.42	132.4200	106954079.0	0.00
2023-12-29	121.70	129.20	118.17	127.63	127.6300	85947490.0	0.08
2023-11-30	108.30	121.84	105.83	121.80	121.7240	96927283.0	0.00
2023-10-31	110.45	114.89	105.53	108.63	108.5623	114374662.0	0.00
...	...	...	...	...	...	...	...
2000-04-28	155.25	167.94	143.06	157.25	173.5309	134577200.0	0.00
2000-03-31	133.50	164.88	126.25	155.63	171.7432	187565000.0	0.41
2000-02-29	134.25	143.13	124.94	132.38	145.6515	148506000.0	0.00
2000-01-31	153.00	154.94	133.06	134.00	147.4339	140256100.0	0.00
1999-12-31	130.25	159.50	130.06	154.75	170.2642	130042100.0	0.41

291 rows × 7 columns



```
[23] ts = TimeSeries(key=mykey,output_format='pandas')
```

```
ts.get_intraday("MSFT",outputsize="full", interval="60min")[0]
```

	1. open	2. high	3. low	4. close	5. volume
date					
2024-02-21 19:00:00	403.325	405.000	403.310	404.850	72982.0
2024-02-21 18:00:00	403.300	403.500	330.530	403.490	49889.0
2024-02-21 17:00:00	402.820	405.650	402.180	403.300	273413.0
2024-02-21 16:00:00	402.180	429.134	382.582	402.800	7056257.0
2024-02-21 15:00:00	398.770	402.290	398.140	402.180	3132887.0
...	...	...	...	...	...
2024-01-31 08:00:00	404.430	411.803	400.114	405.260	473256.0
2024-01-31 07:00:00	404.550	406.757	403.558	406.657	199540.0
2024-01-31 06:00:00	402.134	404.482	400.963	404.471	62444.0
2024-01-31 05:00:00	401.466	403.004	400.953	402.136	52747.0
2024-01-31 04:00:00	406.257	407.865	398.068	401.517	161235.0

240 rows × 5 columns





SHRI VILEPARLE KELAVANI MANDAL'S  
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING  
(Autonomous College Affiliated to the University of Mumbai)  
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)



A.Y.: 2023-24

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: FMC

✓  
2s

```
[25] ts.get_intraday("MSFT",outputsize="full", interval="1min")[0]
```

	1. open	2. high	3. low	4. close	5. volume
date					
2024-02-21 19:59:00	404.890	404.900	404.850	404.850	1529.0
2024-02-21 19:58:00	404.900	404.900	404.750	404.899	420.0
2024-02-21 19:57:00	404.980	404.980	404.510	404.800	476.0
2024-02-21 19:56:00	405.000	405.000	404.600	404.740	839.0
2024-02-21 19:55:00	404.980	405.000	404.690	405.000	1644.0
...	...	...	...	...	...
2024-01-31 04:04:00	404.909	404.949	404.357	404.379	2867.0
2024-01-31 04:03:00	405.199	405.239	404.906	404.958	1016.0
2024-01-31 04:02:00	405.638	405.758	405.125	405.237	3011.0
2024-01-31 04:01:00	406.257	406.257	405.504	405.757	2964.0
2024-01-31 04:00:00	406.257	407.864	405.255	405.786	5247.0

14359 rows × 5 columns

✓  
0s



```
from alpha_vantage.techindicators import TechIndicators
```



✓ [27] `import matplotlib.pyplot as plt`

✓ [28] `ti = TechIndicators(key=mykey, output_format='pandas')`

✓ [29] `ti`

<alpha\_vantage.techindicators.TechIndicators at 0x791257ceee90>

✓ [30] `sma= ti.get_sma("MSFT",interval="daily",time_period=50)[0]`

✓ `sma.head()`



SMA



date



2000-01-11 31.0110

2000-01-12 31.0943

2000-01-13 31.1889

2000-01-14 31.3145

2000-01-18 31.4606



```
[32] msft= ts.get_daily("MSFT",outputsize="full")[0]
      close = msft.loc["1999-11-09":,"4. close"].to_frame()
```

close

date	4. close
2024-02-21	402.18
2024-02-20	402.79
2024-02-16	404.06
2024-02-15	406.56
2024-02-14	409.49
...	...
1999-11-15	87.00
1999-11-12	89.19
1999-11-11	89.62
1999-11-10	87.12
1999-11-09	88.87

6109 rows × 1 columns

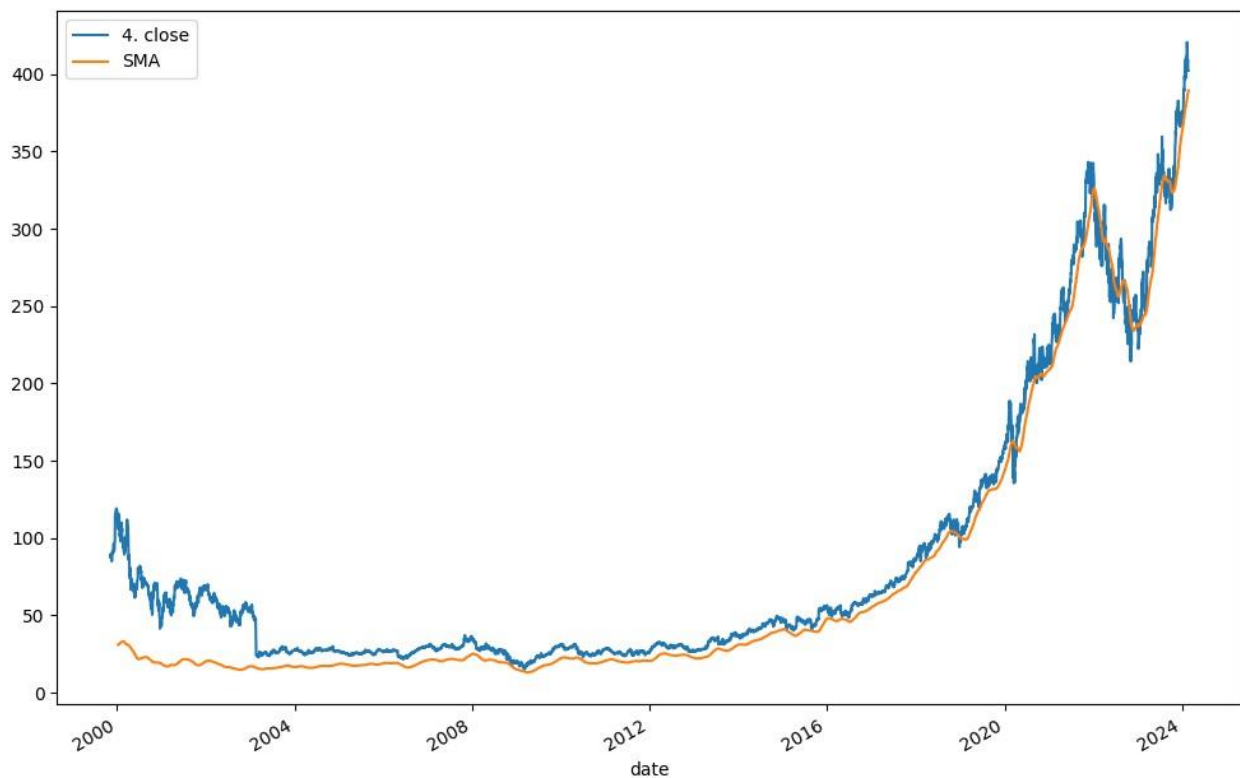


```
[34] close.head()
```

date	4. close
2024-02-21	402.18
2024-02-20	402.79
2024-02-16	404.06
2024-02-15	406.56
2024-02-14	409.49

```
[35] close["SMA"]=sma
```

```
close.plot(figsize=(12,8))  
plt.show()
```





SHRI VILEPARLE KELAVANI MANDAL'S  
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING  
(Autonomous College Affiliated to the University of Mumbai)  
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)



A.Y.: 2023-24

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: FMC

✓ 3s [37] `bbands=ti.get_bbands("MSFT",interval="daily", time_period=50)[0]`

✓ 0s [38] `bbands.head()`

	Real Upper Band	Real Middle Band	Real Lower Band
date			
2024-02-21	422.2276	389.2732	356.3188
2024-02-20	421.8001	388.6350	355.4700
2024-02-16	421.3427	387.9418	354.5408
2024-02-15	420.6728	387.2973	353.9219
2024-02-14	419.8564	386.5354	353.2145

Next steps:

[Generate code with bbands](#)

[View recommended plots](#)

✓ 0s [39] `bbands["Close"]=close.iloc[:,0]`

✓ 0s `bbands`



	Real Upper Band	Real Middle Band	Real Lower Band	Close
date				
2024-02-21	422.2276	389.2732	356.3188	402.18
2024-02-20	421.8001	388.6350	355.4700	402.79
2024-02-16	421.3427	387.9418	354.5408	404.06
2024-02-15	420.6728	387.2973	353.9219	406.56
2024-02-14	419.8564	386.5354	353.2145	409.49
...	...	...	...	...
2000-01-18	38.8485	31.4606	24.0727	115.31
2000-01-14	38.6458	31.3145	23.9831	112.25
2000-01-13	38.4920	31.1889	23.8859	107.81
2000-01-12	38.4015	31.0943	23.7872	105.81
2000-01-11	38.3332	31.0110	23.6888	109.37

6066 rows × 4 columns





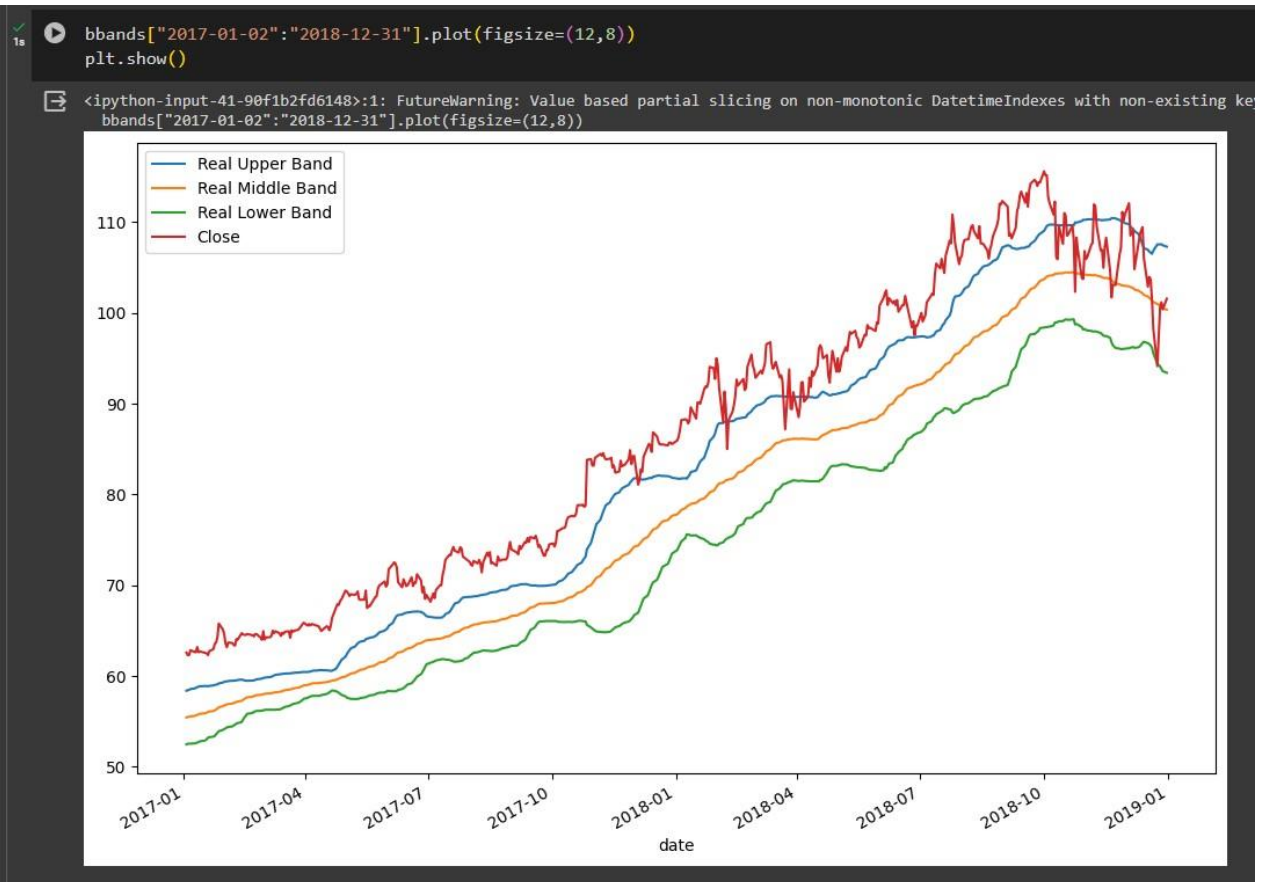
SHRI VILEPARLE KELAVANI MANDAL'S  
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING  
(Autonomous College Affiliated to the University of Mumbai)  
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)



A.Y.: 2023-24

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: FMC





## TSLA Price and Volume

Closing price on 2021-03-19: \$654.87  
Shares traded on 2021-03-19: 42,893,978

