```
from tabulate import tabulate
import pandas as pd
import numpy as np
Lingyi_TS_Monthly_df = pd.read_csv('/content/MSFT_Stock.csv')
microsoft  = pd.read_csv('MSFT_Stock.csv', index_col='Date', parse_dates=['Date'])
microsoft.head()
```

	Open	High	Low	Close	Volume
Date					
2015-04-01 16:00:00	40.60	40.76	40.31	40.72	36865322
2015-04-02 16:00:00	40.66	40.74	40.12	40.29	37487476
2015-04-06 16:00:00	40.34	41.78	40.18	41.55	39223692
2015-04-07 16:00:00	41.61	41.91	41.31	41.53	28809375
2015-04-08 16:00:00	41.48	41.69	41.04	41.42	24753438

Next steps: View recommended plots

```
import matplotlib.pyplot as plt
Lingyi_TS_df = microsoft
```

```
#Part1:Data Exploration
```

```
Lingyi_TS_df.head()
```

	Open	High	Low	Close	Volume
Date					
2015-04-01 16:00:00	40.60	40.76	40.31	40.72	36865322
2015-04-02 16:00:00	40.66	40.74	40.12	40.29	37487476
2015-04-06 16:00:00	40.34	41.78	40.18	41.55	39223692
2015-04-07 16:00:00	41.61	41.91	41.31	41.53	28809375
2015-04-08 16:00:00	41.48	41.69	41.04	41.42	24753438

Next steps: View recommended plots

```
Lingyi_TS_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1511 entries, 2015-04-01 16:00:00 to 2021-03-31 16:00:00
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Open    1511 non-null   float64
 1   High    1511 non-null   float64
 2   Low     1511 non-null   float64
 3   Close   1511 non-null   float64
 4   Volume  1511 non-null   int64  
```

```
dtypes: float64(4), int64(1)
memory usage: 70.8 KB
```

```
Lingyi_TS_df.describe()
```

	Open	High	Low	Close	Volume
count	1511.000000	1511.000000	1511.000000	1511.000000	1.511000e+03
mean	107.385976	108.437472	106.294533	107.422091	3.019863e+07
std	56.691333	57.382276	55.977155	56.702299	1.425266e+07
min	40.340000	40.740000	39.720000	40.290000	1.016120e+05
25%	57.860000	58.060000	57.420000	57.855000	2.136213e+07
50%	93.990000	95.100000	92.920000	93.860000	2.662962e+07
75%	139.440000	140.325000	137.825000	138.965000	3.431962e+07
max	245.030000	246.130000	242.920000	244.990000	1.352271e+08

```
Lingyi_TS_df.nunique()
```

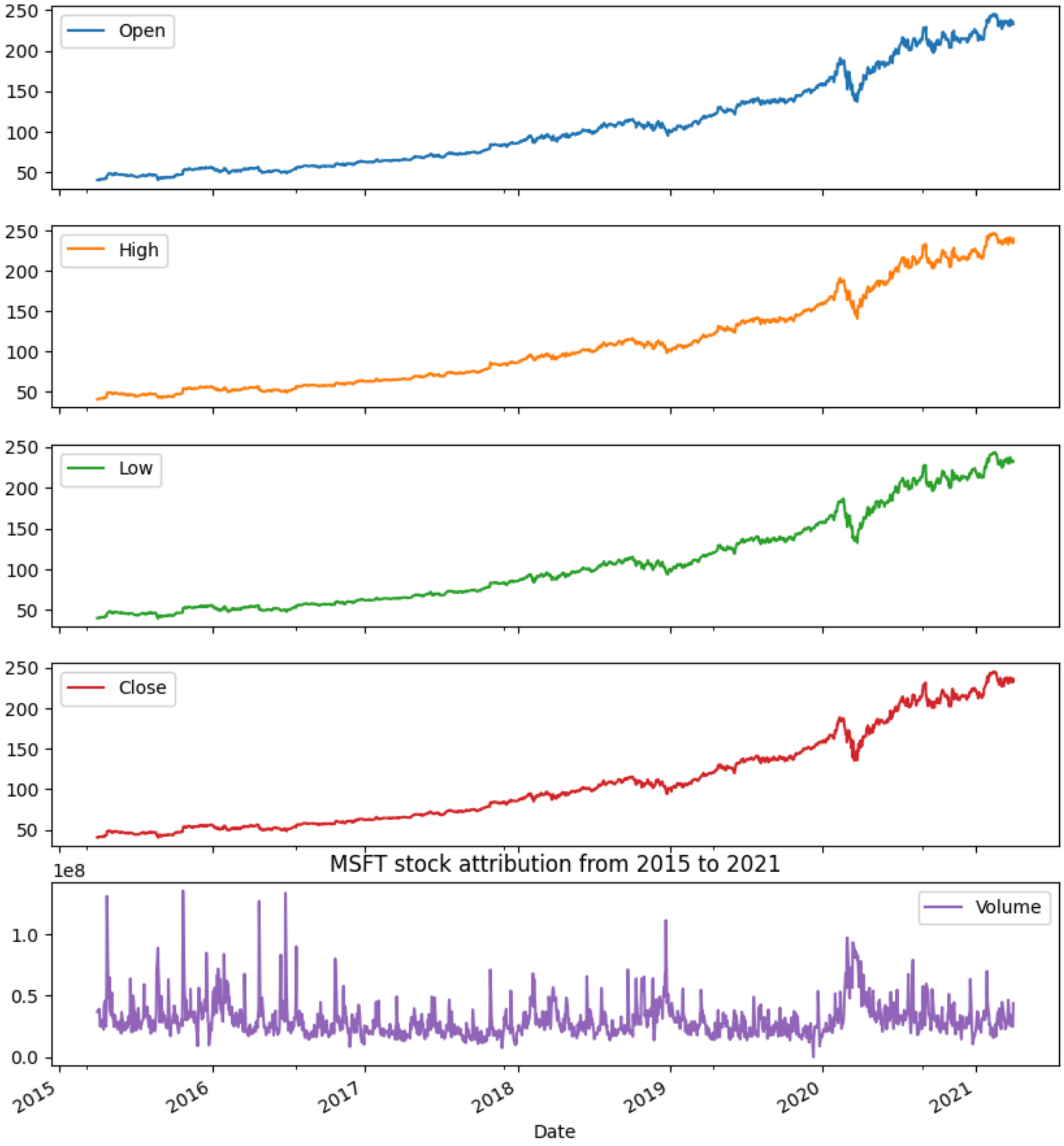
```
Open      1409
High      1400
Low       1397
Close     1398
Volume    1511
dtype: int64
```

```
Lingyi_TS_df.isnull().sum()
```

```
Open      0
High      0
Low       0
Close     0
Volume    0
dtype: int64
```

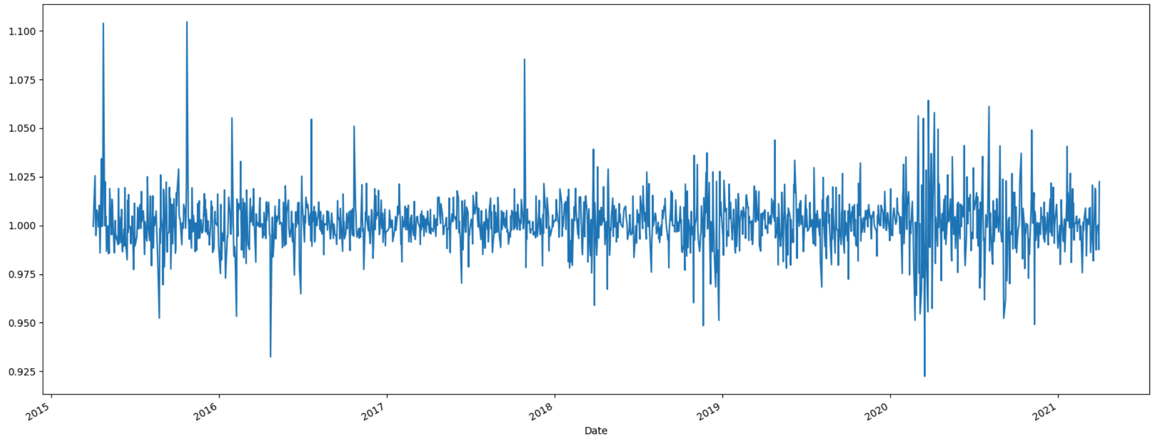
```
#Exploratory Data Analysis
```

```
Lingyi_TS_df['2015':'2021'].plot(subplots = True,figsize = (10,12))
plt.title('MSFT stock attribution from 2015 to 2021')
plt.savefig('stocks.png')
```



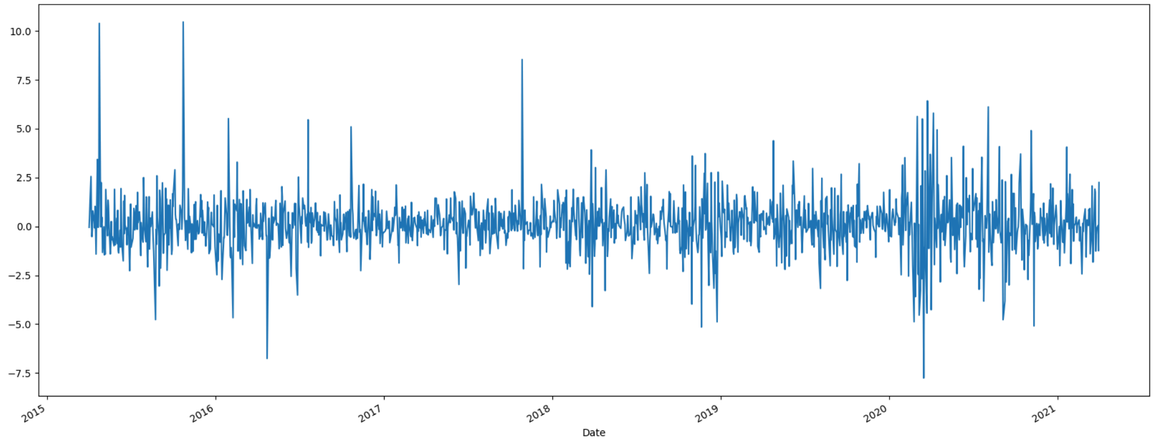
```
Lingyi_TS_df['Change'] = Lingyi_TS_df.High.div(Lingyi_TS_df.High.shift())
Lingyi_TS_df['Change'].plot(figsize=(20,8))
```

```
<Axes: xlabel='Date'>
```



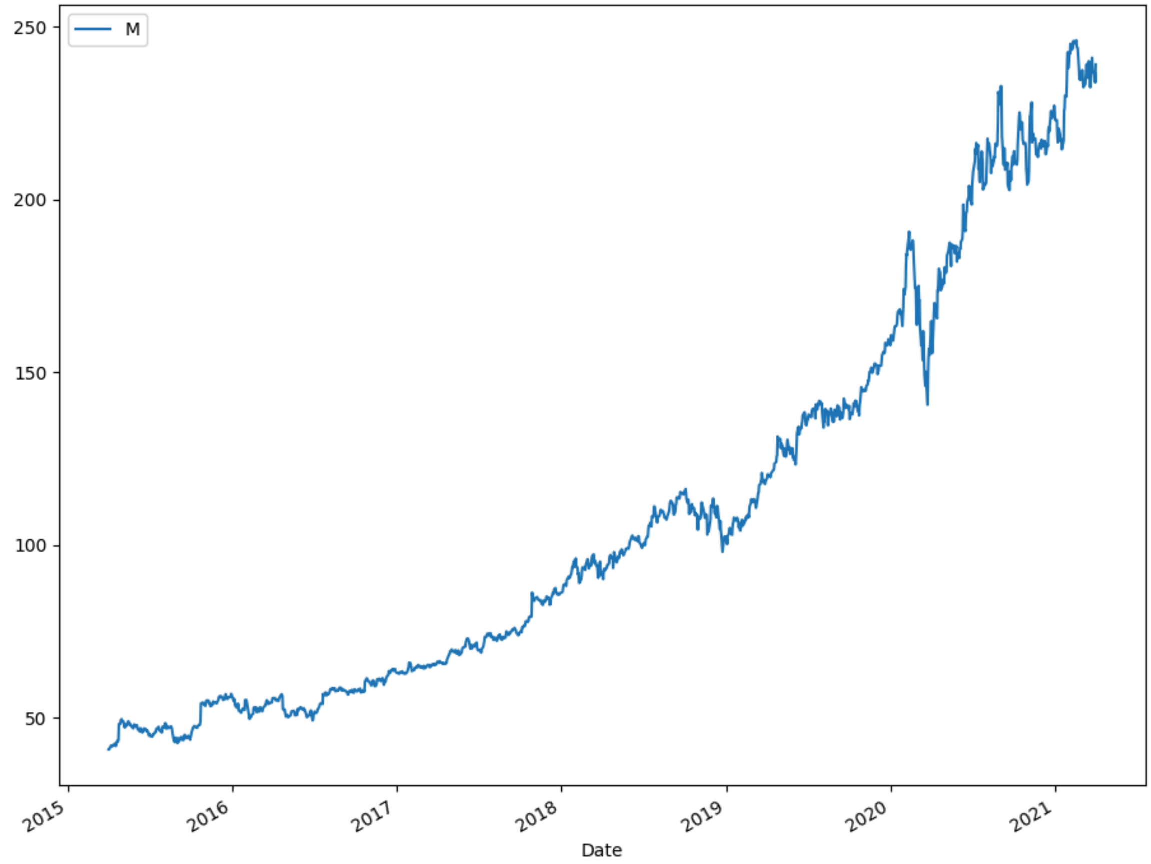
```
Lingyi_TS_df['Return'] = Lingyi_TS_df.Change.sub(1).mul(100)#revenue rate
Lingyi_TS_df['Return'].plot(figsize=(20,8))
```

```
<Axes: xlabel='Date'>
```



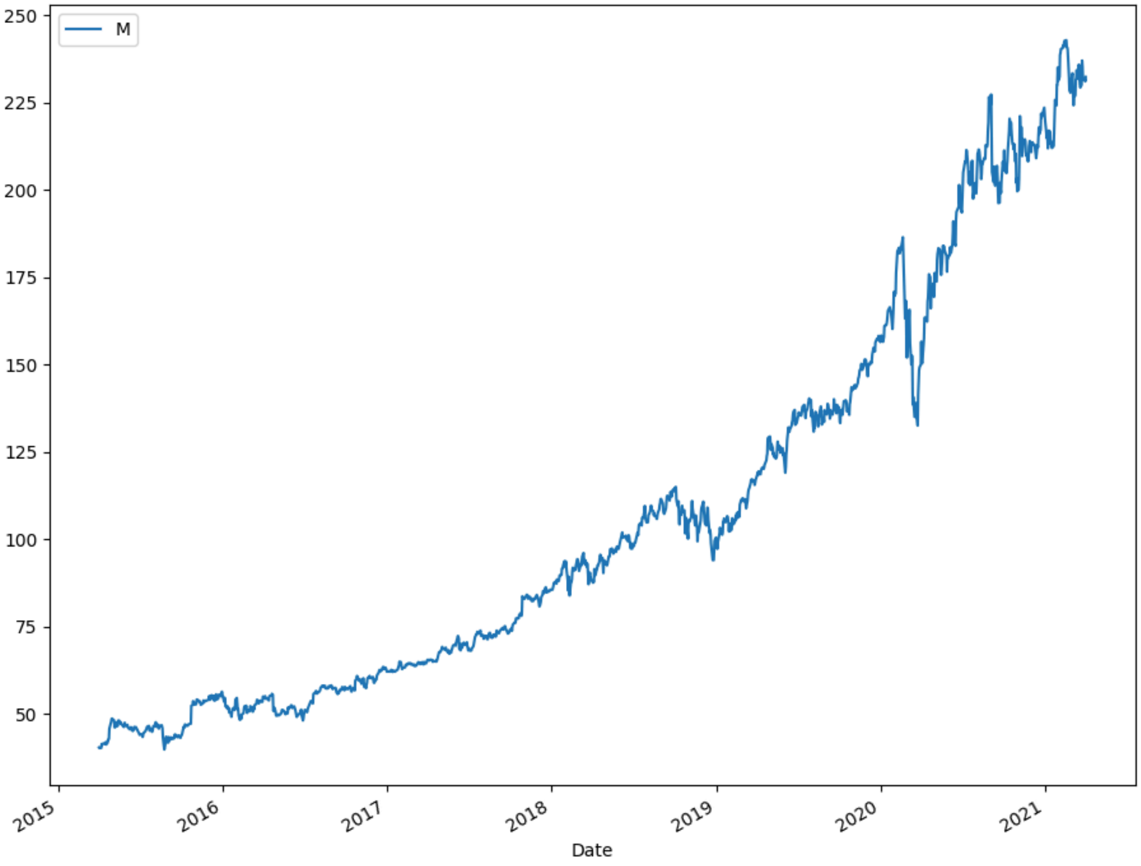
```
Lingyi_TS_df.High.plot()
plt.legend(('Microsoft High'))
```

```
<matplotlib.legend.Legend at 0x7aa49e96df90>
```



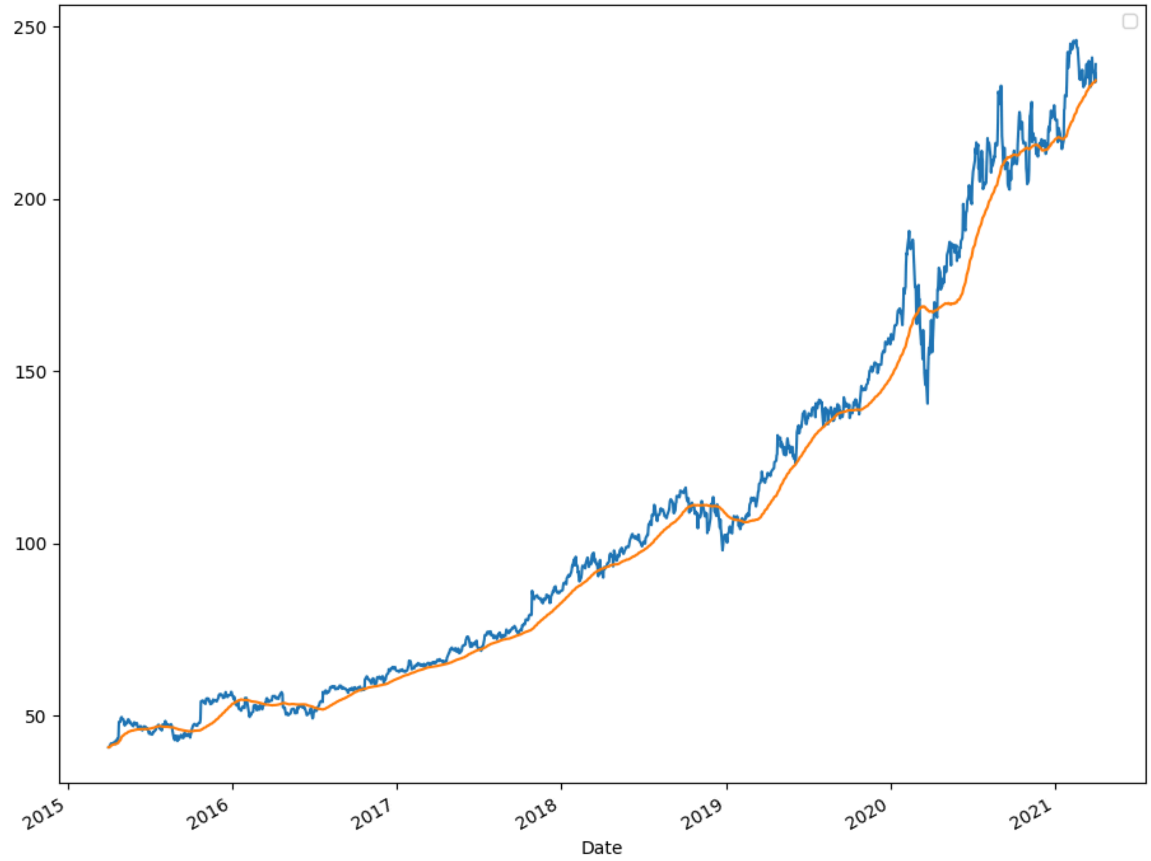
```
Lingyi_TS_df.Low.plot()
plt.legend(('Microsoft Low'))
```

```
<matplotlib.legend.Legend at 0x7aa49e9f8b20>
```



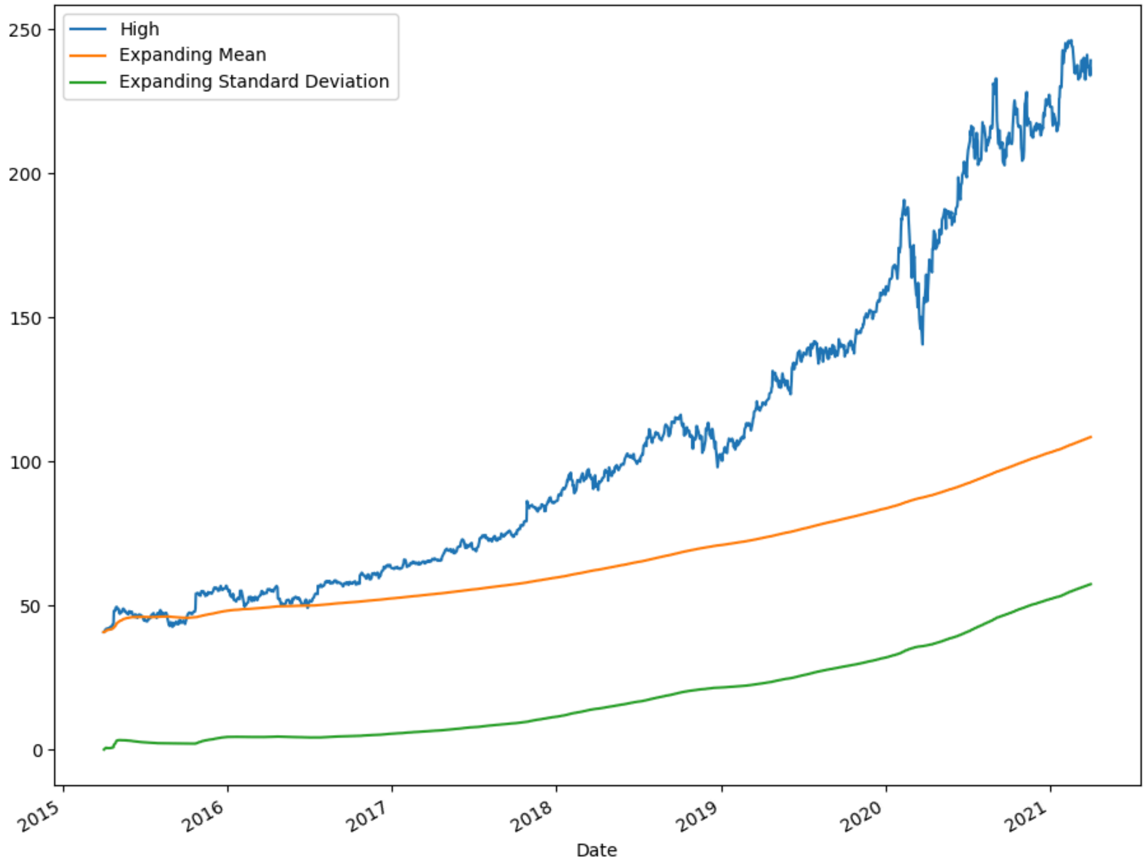
```
rolling_MSFT =Lingyi_TS_df.High.rolling('90D').mean()
Lingyi_TS_df.High.plot()
rolling_MSFT.plot()
plt.legend('HIgh','Rolling Mean')
```

```
<ipython-input-48-32757c33cba0>:4: UserWarning: Legend does not support handles
A proxy artist may be used instead.
See: https://matplotlib.org/stable/tutorials/intermediate/legend_guide.html#cont
  plt.legend('HIgh','Rolling Mean')
<matplotlib.legend.Legend at 0x7aa49ea6c520>
```



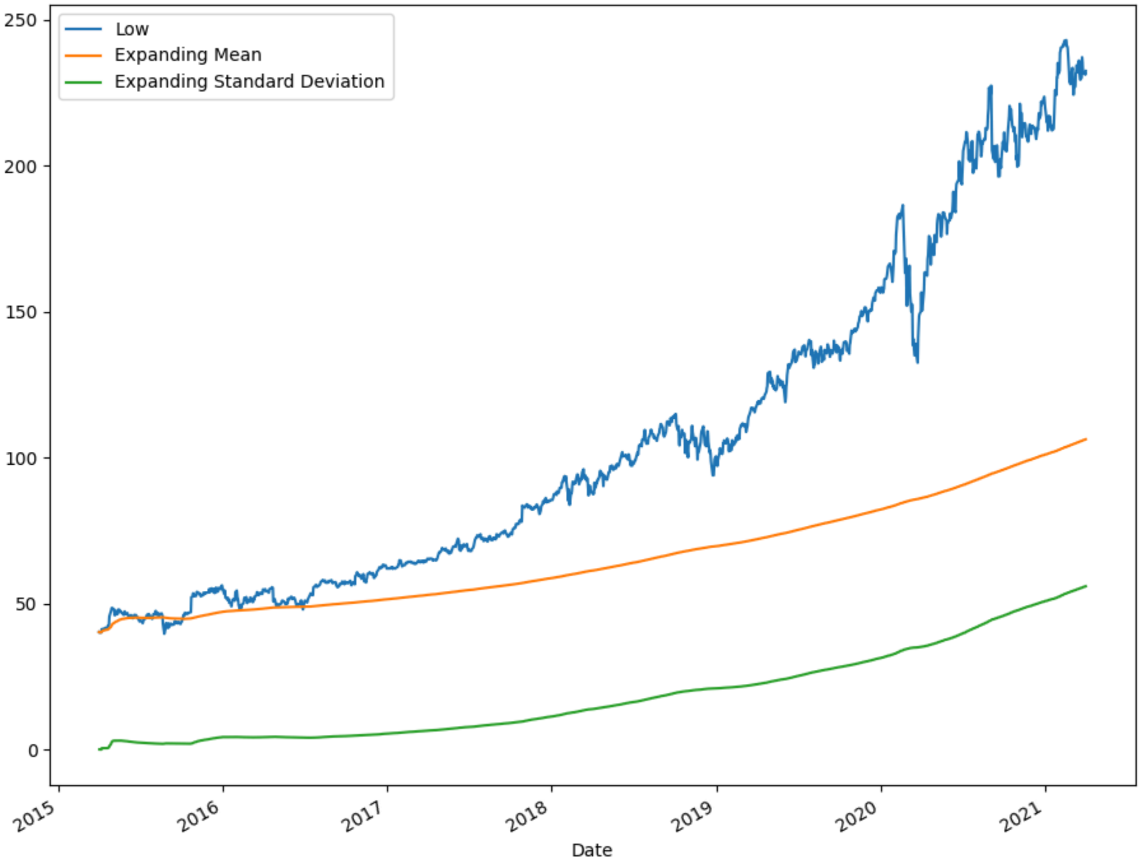
```
microsoft_mean = Lingyi_TS_df.High.expanding().mean()
microsoft_std = Lingyi_TS_df.High.expanding().std()
Lingyi_TS_df.High.plot()
microsoft_mean.plot()
microsoft_std.plot()
plt.legend(['High','Expanding Mean','Expanding Standard Deviation'])
```

```
<matplotlib.legend.Legend at 0x7aa49ebfdd80>
```

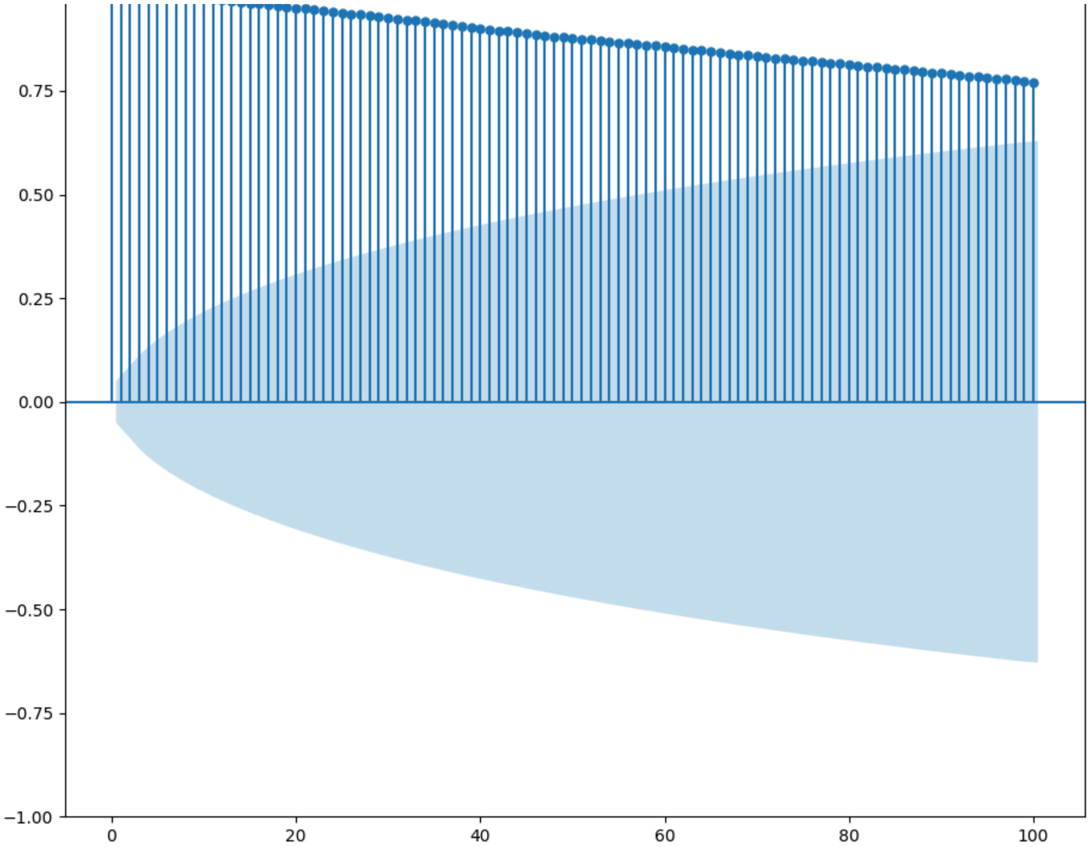


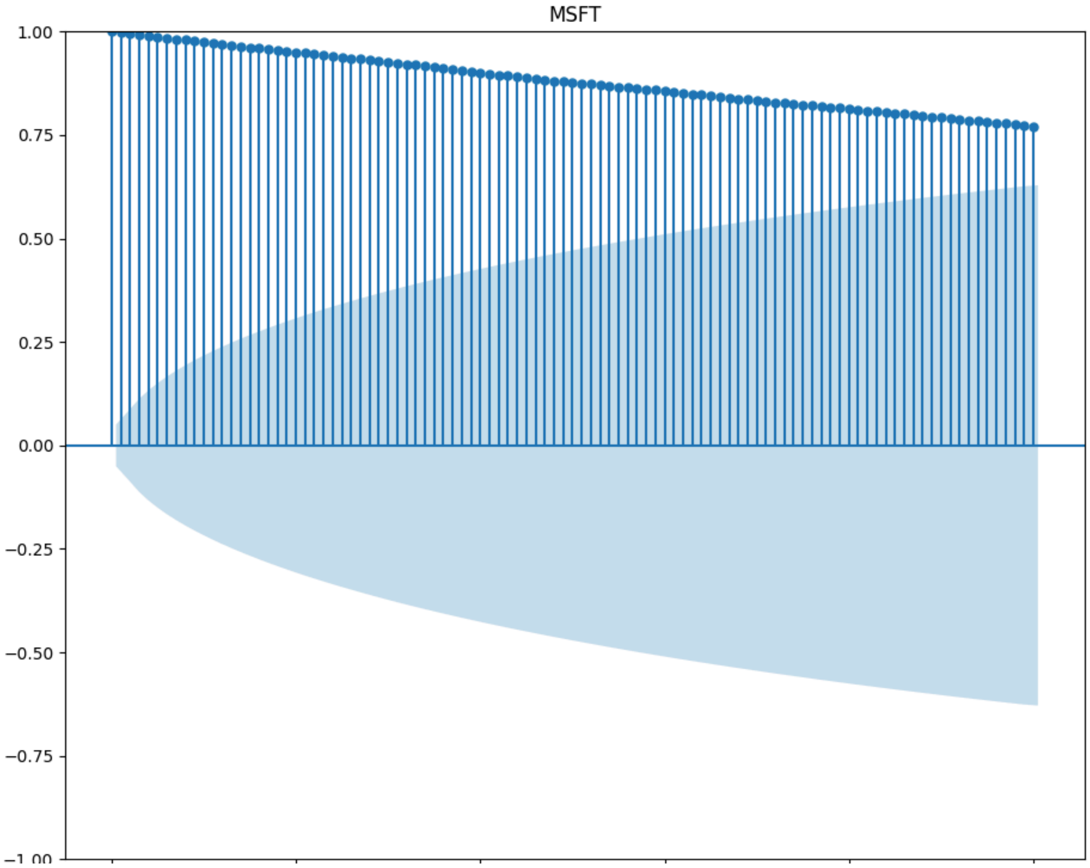
```
microsoft_mean = Lingyi_TS_df.Low.expanding().mean()
microsoft_std = Lingyi_TS_df.Low.expanding().std()
Lingyi_TS_df.Low.plot()
microsoft_mean.plot()
microsoft_std.plot()
plt.legend(['Low','Expanding Mean','Expanding Standard Deviation'])
```

```
<matplotlib.legend.Legend at 0x7aa49ea87700>
```

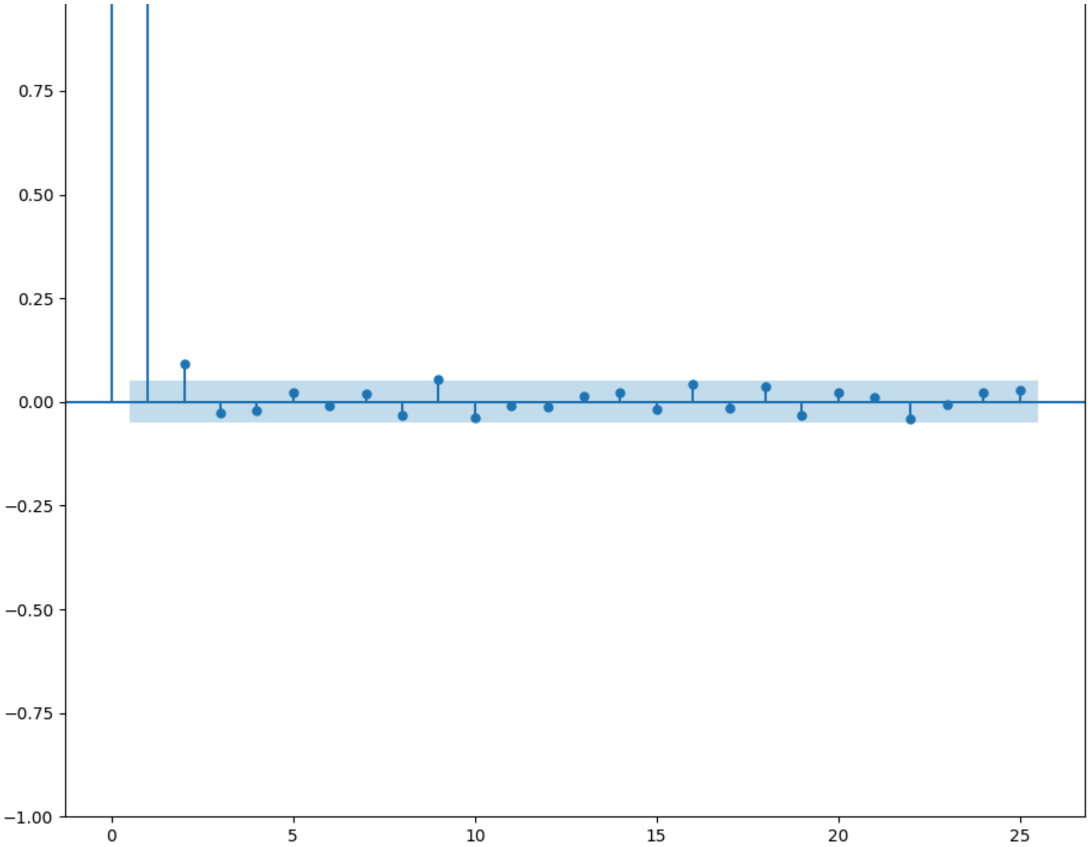


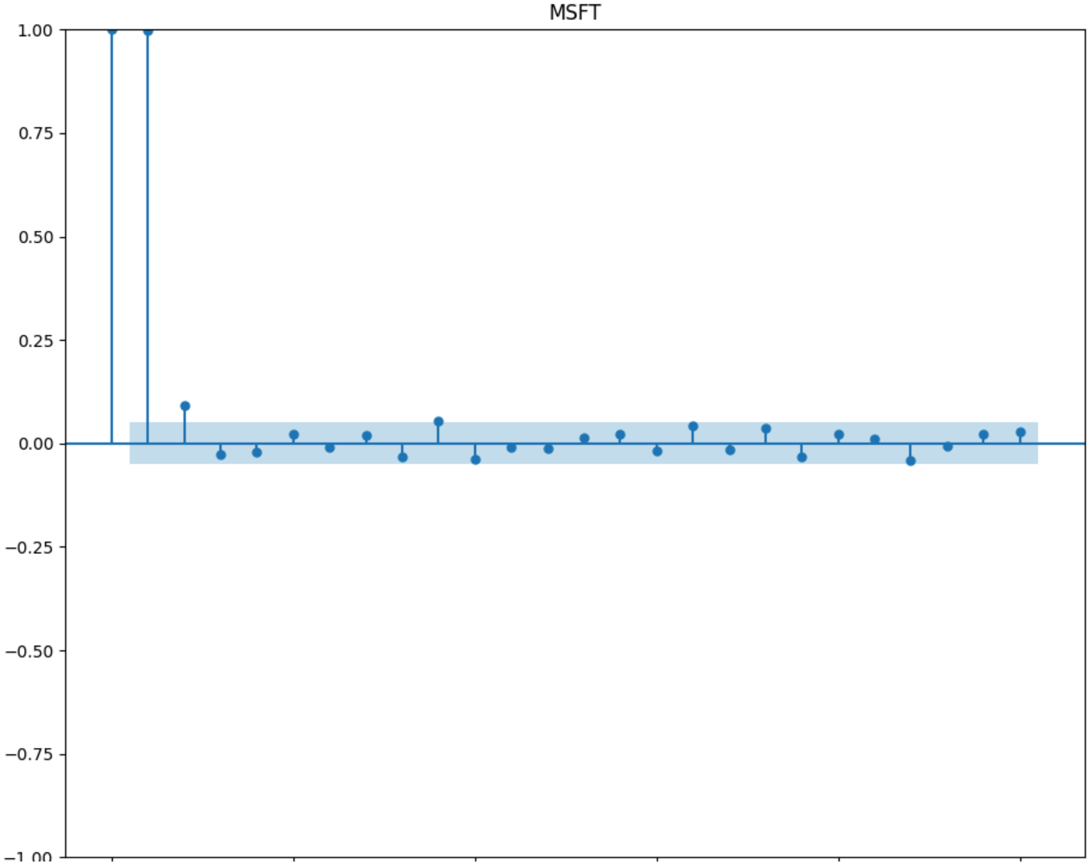
```
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
plot_acf(Lingyi_TS_df['High'],lags = 100,title = 'MSFT')
```


0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
20
40
60
80
100


MSFT
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

```
plot_pacf(Lingyi_TS_df['Close'],lags = 25,title ='MSFT')
```


0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25


MSFT
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

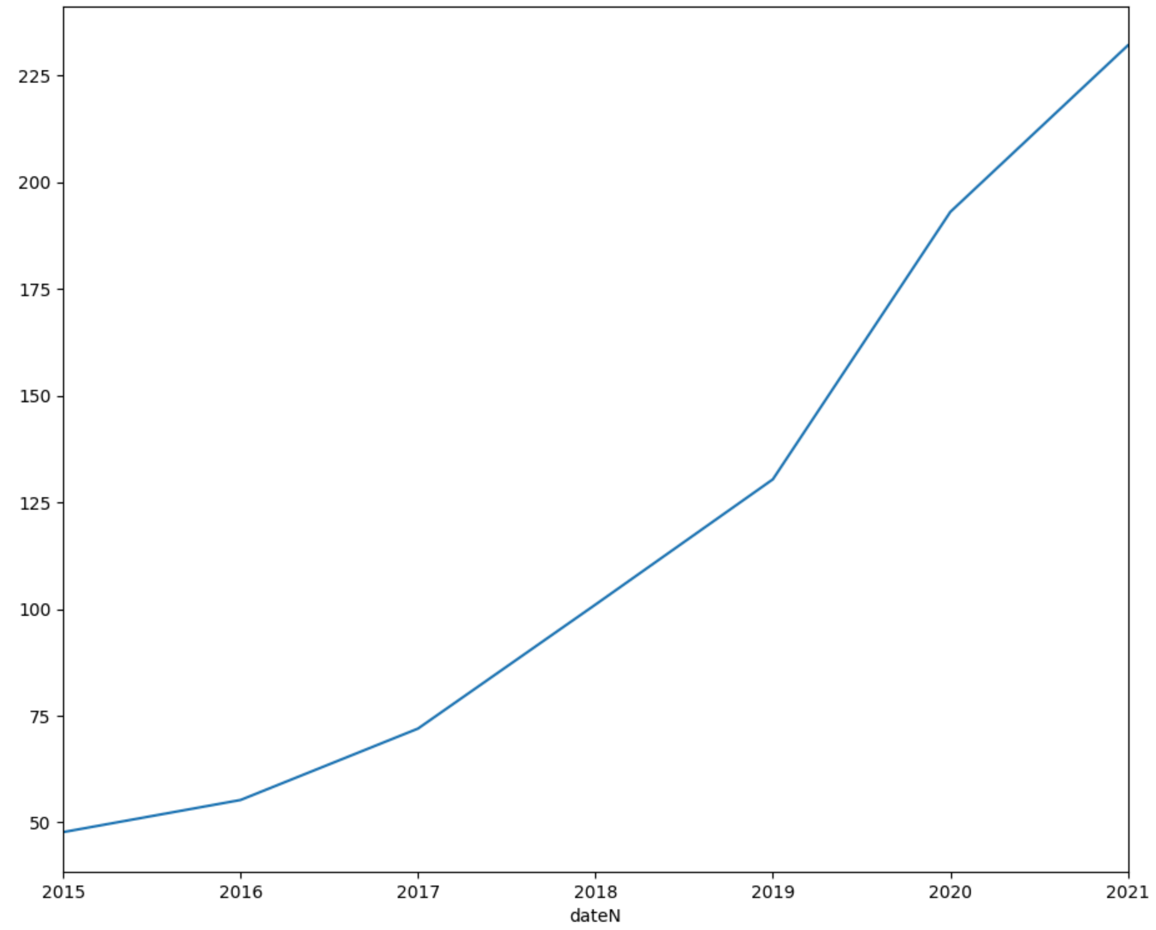
```
Lingyi_TS_Monthly_df['dateN']=pd.to_datetime(Lingyi_TS_Monthly_df['Date'])
Lingyi_TS_Monthly_df.set_index('dateN',inplace=True)
Lingyi_TS_Monthly_df.head()
```

```
                                   Date   Open   High    Low  Close    Volume
              dateN
2015-04-01 16:00:00  4/1/2015 16:00:00  40.60  40.76  40.31  40.72  36865322
2015-04-02 16:00:00  4/2/2015 16:00:00  40.66  40.74  40.12  40.29  37487476
2015-04-06 16:00:00  4/6/2015 16:00:00  40.34  41.78  40.18  41.55  39223692
2015-04-07 16:00:00  4/7/2015 16:00:00  41.61  41.91  41.31  41.53  28809375
2015-04-08 16:00:00  4/8/2015 16:00:00  41.48  41.69  41.04  41.42  24753438
```

Next steps: View recommended plots

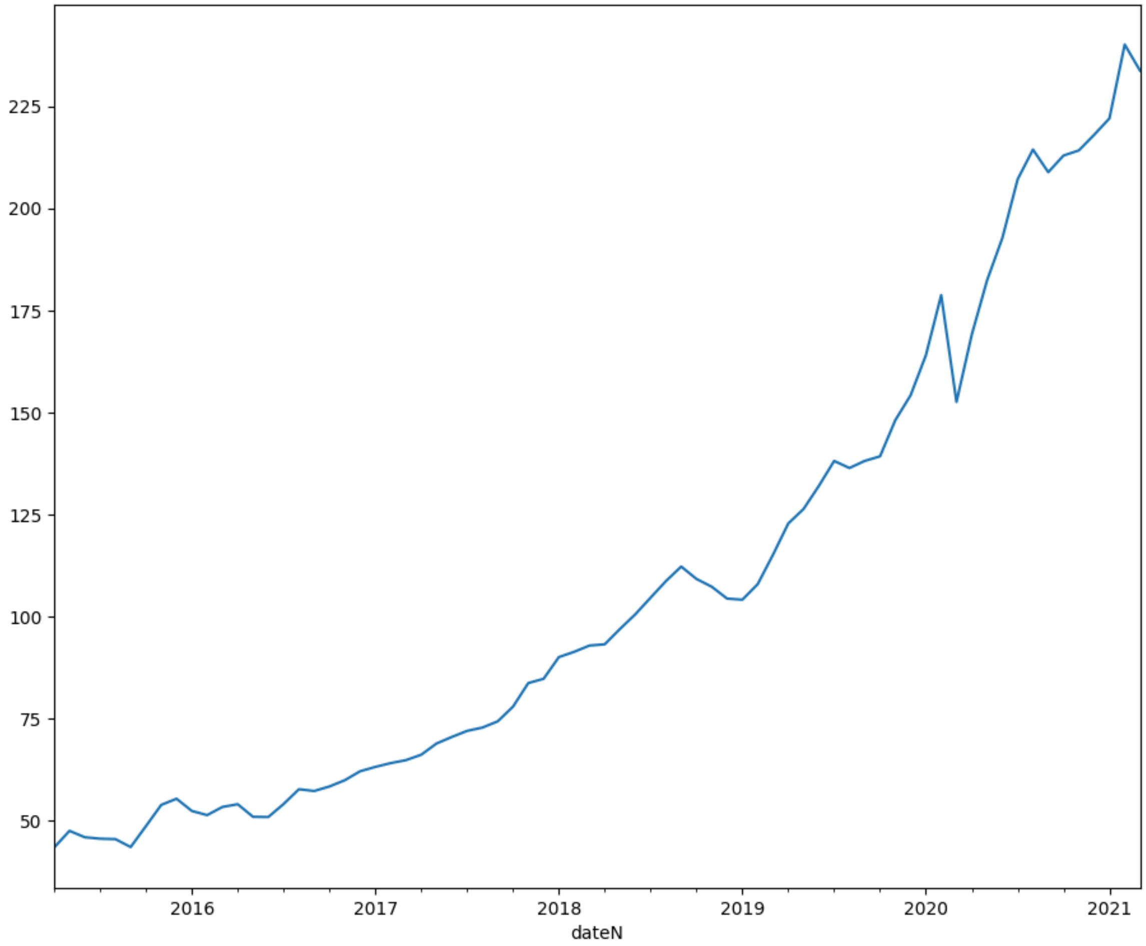
```
Lingyi_TS_Monthly_df['Close'].resample('Y').mean().plot()
```

```
<Axes: xlabel='dateN'>
```



```
Lingyi_TS_Monthly_df['Close'].resample('M').mean().plot()
```

```
<Axes: xlabel='dateN'>
```



```
Lingyi_TS_Monthly_df.describe()
```

	Open	High	Low	Close	Volume
count	1511.000000	1511.000000	1511.000000	1511.000000	1.511000e+03
mean	107.385976	108.437472	106.294533	107.422091	3.019863e+07
std	56.691333	57.382276	55.977155	56.702299	1.425266e+07
min	40.340000	40.740000	39.720000	40.290000	1.016120e+05
25%	57.860000	58.060000	57.420000	57.855000	2.136213e+07
50%	93.990000	95.100000	92.920000	93.860000	2.662962e+07
75%	139.440000	140.325000	137.825000	138.965000	3.431962e+07
max	245.030000	246.130000	242.920000	244.990000	1.352271e+08