

cpu92psd5

December 23, 2024

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import datetime
import warnings
warnings.filterwarnings('ignore')
```

```
[14]: df = pd.read_csv('AAPL_5min_sample[1].csv')
```

```
[16]: df
```

```
[16]:
```

	timestamp	open	high	low	close	volume
0	2024-12-02 04:00:00	237.1000	237.1200	236.8900	236.9200	2451
1	2024-12-02 04:05:00	236.8600	236.8600	236.8600	236.8600	100
2	2024-12-02 04:10:00	237.0700	237.1200	237.0700	237.0700	1227
3	2024-12-02 04:15:00	237.1200	237.1400	237.1200	237.1400	366
4	2024-12-02 04:20:00	237.1400	237.1700	237.1400	237.1700	200
...
1962	2024-12-16 19:35:00	250.8700	250.8907	250.8700	250.8907	400
1963	2024-12-16 19:40:00	250.8914	250.9286	250.8400	250.8500	2820
1964	2024-12-16 19:45:00	250.9000	250.9900	250.8700	250.9900	1955
1965	2024-12-16 19:50:00	251.0000	251.0000	250.9993	250.9999	1706
1966	2024-12-16 19:55:00	250.9000	251.0400	250.9000	251.0000	989

[1967 rows x 6 columns]

```
[17]: df.head(100)
```

```
[17]:
```

	timestamp	open	high	low	close	volume
0	2024-12-02 04:00:00	237.10	237.12	236.89	236.92	2451
1	2024-12-02 04:05:00	236.86	236.86	236.86	236.86	100
2	2024-12-02 04:10:00	237.07	237.12	237.07	237.07	1227
3	2024-12-02 04:15:00	237.12	237.14	237.12	237.14	366
4	2024-12-02 04:20:00	237.14	237.17	237.14	237.17	200
..

```

95  2024-12-02 12:25:00  240.18  240.20  239.76  239.84  202607
96  2024-12-02 12:30:00  239.84  239.95  239.74  239.92  143042
97  2024-12-02 12:35:00  239.93  240.00  239.78  239.86  104128
98  2024-12-02 12:40:00  239.87  239.90  239.71  239.74  110067
99  2024-12-02 12:45:00  239.74  239.84  239.61  239.64  163380

```

[100 rows x 6 columns]

```
[18]: df.sample(100)
```

```

[18]:          timestamp      open      high      low      close  volume
288  2024-12-03 13:30:00  241.4200  241.7400  241.34  241.73  255748
891  2024-12-09 05:50:00  242.6000  242.8400  242.60  242.84     870
1791 2024-12-16 04:05:00  248.4100  248.4900  248.41  248.49     608
1595 2024-12-12 19:00:00  246.7801  246.7801  246.70  246.78     476
445  2024-12-04 11:40:00  242.2950  242.3000  242.11  242.15  290604
...
1574 2024-12-12 17:15:00  248.0300  248.0300  247.70  247.74     7001
549  2024-12-05 04:50:00  242.8500  242.8500  242.85  242.85      200
536  2024-12-04 19:40:00  242.9000  242.9000  242.75  242.75     1906
1092 2024-12-10 07:30:00  247.5400  247.6100  247.54  247.61     1358
1682 2024-12-13 10:30:00  247.5600  247.9500  247.55  247.68  260298

```

[100 rows x 6 columns]

```
[19]: df.shape
```

```
[19]: (1967, 6)
```

```
[20]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1967 entries, 0 to 1966
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   timestamp   1967 non-null   object
1   open        1967 non-null   float64
2   high        1967 non-null   float64
3   low         1967 non-null   float64
4   close       1967 non-null   float64
5   volume      1967 non-null   int64
dtypes: float64(4), int64(1), object(1)
memory usage: 92.3+ KB

```

```
[21]: df.describe
```

```
[21]: <bound method NDFrame.describe of
low      close  volume
0      2024-12-02 04:00:00  237.1000  237.1200  236.8900  236.9200  2451
1      2024-12-02 04:05:00  236.8600  236.8600  236.8600  236.8600  100
2      2024-12-02 04:10:00  237.0700  237.1200  237.0700  237.0700  1227
3      2024-12-02 04:15:00  237.1200  237.1400  237.1200  237.1400  366
4      2024-12-02 04:20:00  237.1400  237.1700  237.1400  237.1700  200
...
1962   2024-12-16 19:35:00  250.8700  250.8907  250.8700  250.8907  400
1963   2024-12-16 19:40:00  250.8914  250.9286  250.8400  250.8500  2820
1964   2024-12-16 19:45:00  250.9000  250.9900  250.8700  250.9900  1955
1965   2024-12-16 19:50:00  251.0000  251.0000  250.9993  250.9999  1706
1966   2024-12-16 19:55:00  250.9000  251.0400  250.9000  251.0000  989

[1967 rows x 6 columns]>
```

```
[22]: df.describe().transpose()
```

```
[22]:
```

	count	mean	std	min	25%	50% \
open	1967.0	244.997475	3.394242	236.86	242.7343	246.36
high	1967.0	245.100157	3.404108	236.86	242.7900	246.46
low	1967.0	244.887455	3.387967	236.00	242.6500	246.17
close	1967.0	245.002633	3.392876	236.86	242.7300	246.36
volume	1967.0	159562.929842	656210.082150	100.00	1109.0000	10005.00

	75%	max
open	247.72500	2.512863e+02
high	247.80000	2.513800e+02
low	247.62500	2.511700e+02
close	247.72295	2.512859e+02
volume	171726.00000	1.717780e+07

```
[23]: df.isnull().sum()
```

```
[23]: timestamp    0
open              0
high              0
low               0
close             0
volume            0
dtype: int64
```

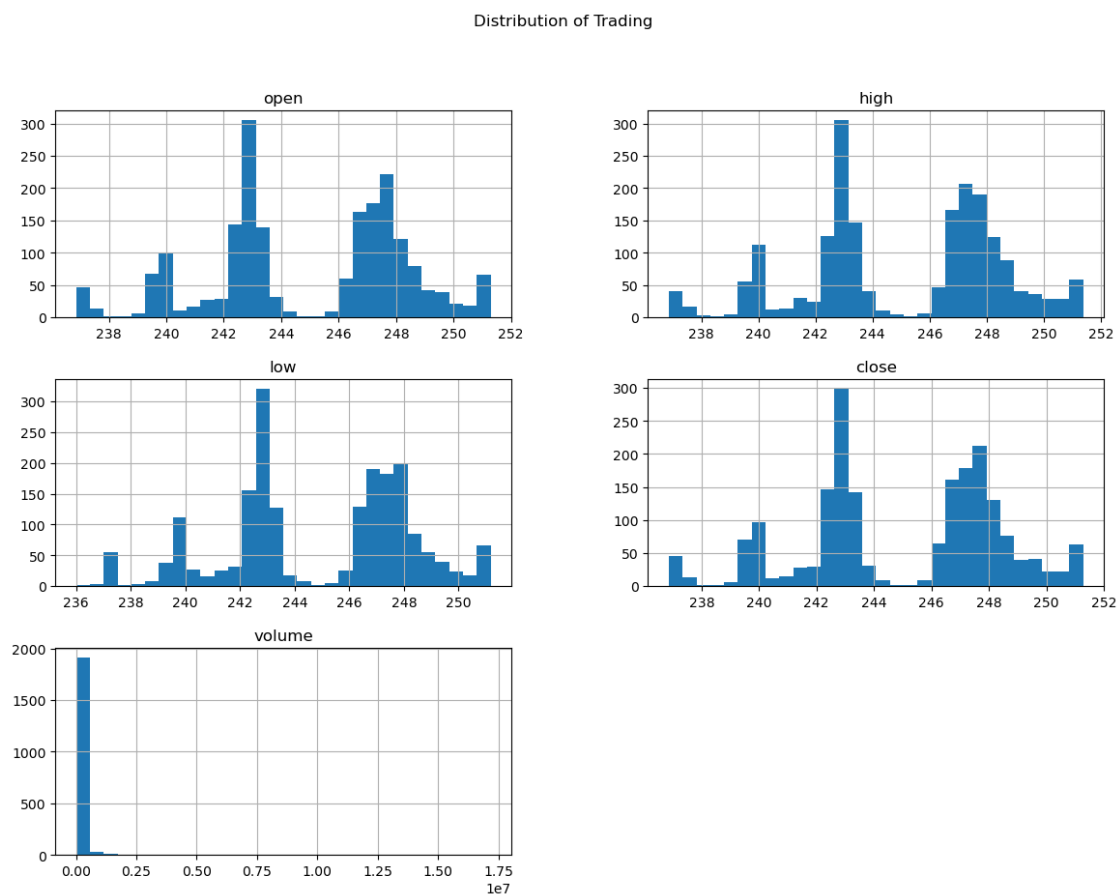
```
[24]: df.duplicated().sum()
```

```
[24]: 0
```

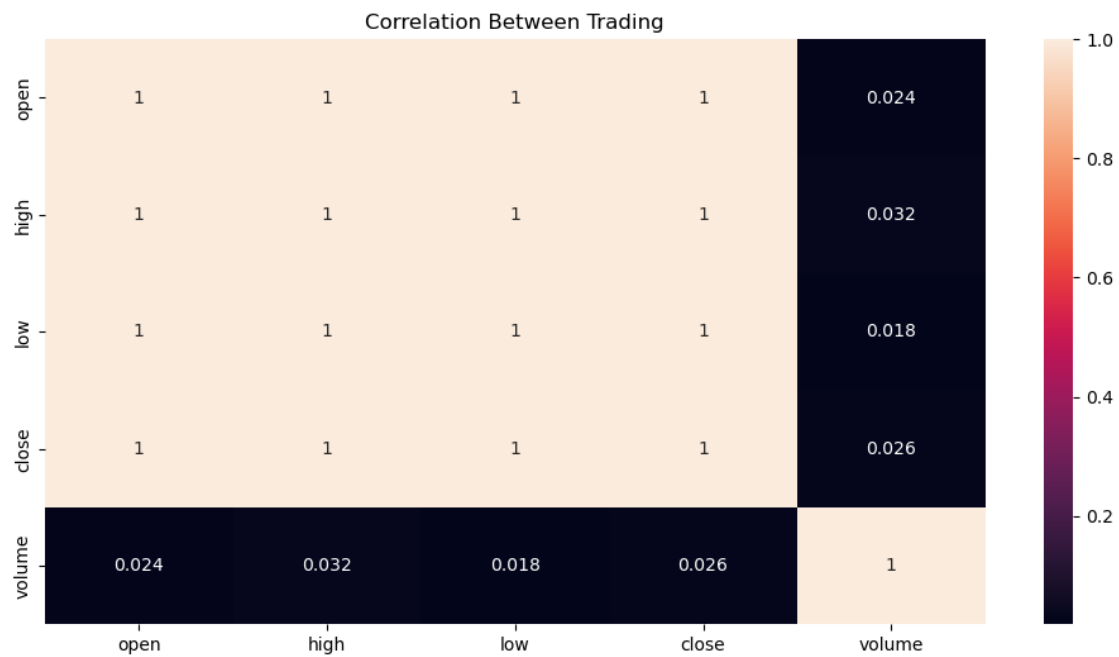
```
[25]: df.nunique()
```

```
[25]: timestamp    1967
      open         1021
      high         977
      low          937
      close        1061
      volume       1708
      dtype: int64
```

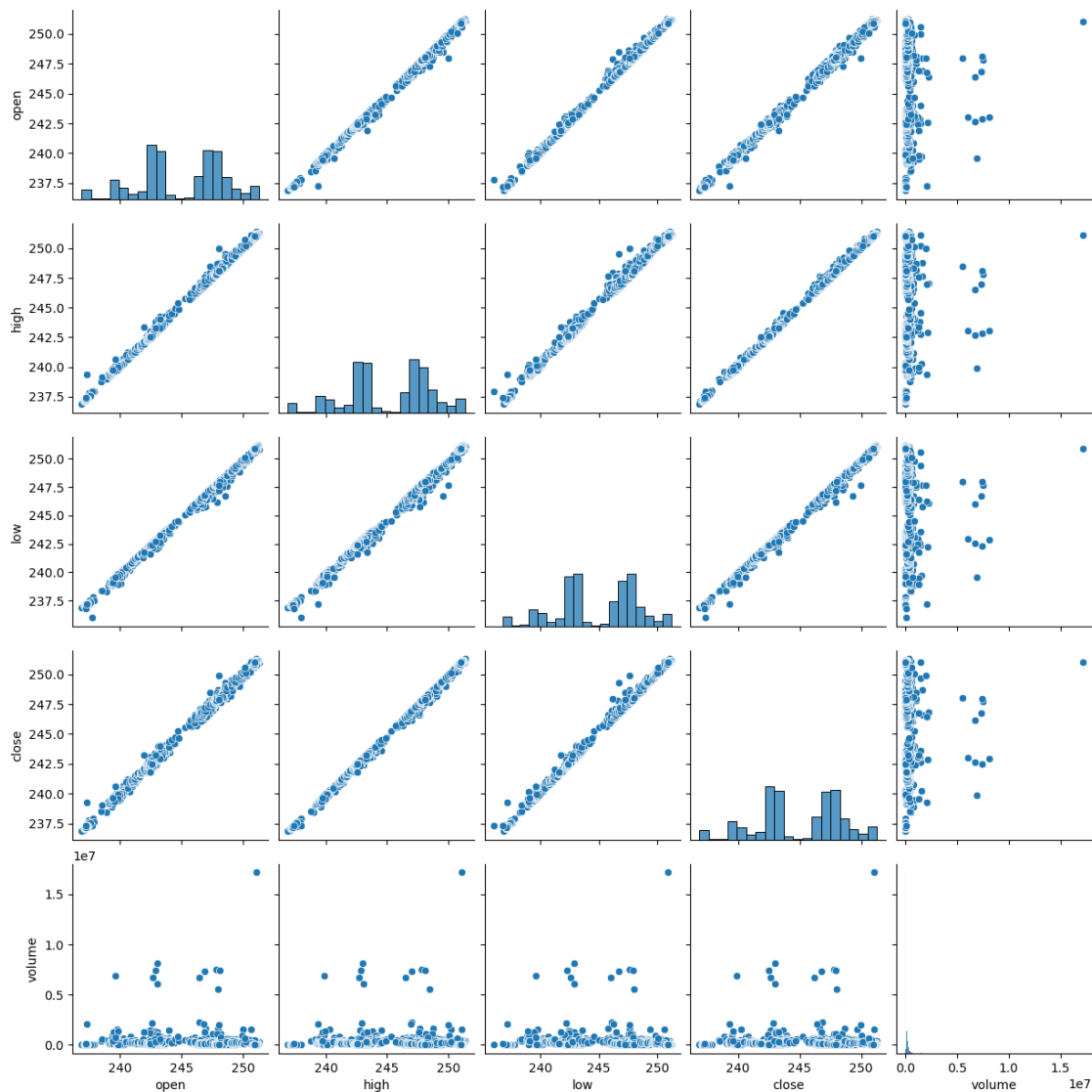
```
[26]: df[['timestamp', 'open', 'high', 'low', 'close', 'volume']].hist(bins=30,
      ↪figsize=(14, 10))
      plt.suptitle('Distribution of Trading')
      plt.show()
```



```
[27]: plt.figure(figsize=(12, 6))
      sns.heatmap(df.corr(numeric_only = True), annot=True,)
      plt.title('Correlation Between Trading')
      plt.show()
```



```
[28]: sns.pairplot(df)  
plt.show()
```



```
[29]: top_trading = df.sort_values('volume', ascending=False).head(10)
top_trading
```

```
[29]:
```

	timestamp	open	high	low	close	volume
1924	2024-12-16 16:00:00	251.05	251.1201	250.88	251.05	17177798
497	2024-12-04 16:00:00	243.01	243.0200	242.89	242.96	8136177
1194	2024-12-10 16:00:00	247.81	247.8100	247.65	247.75	7484209
830	2024-12-06 16:00:00	242.84	242.8400	242.28	242.50	7420243
1748	2024-12-13 16:00:00	248.10	248.1300	247.94	247.95	7388472
1011	2024-12-09 16:00:00	246.86	246.9800	246.73	246.76	7333756
138	2024-12-02 16:00:00	239.60	239.8700	239.59	239.87	6865052
318	2024-12-03 16:00:00	242.65	242.7000	242.56	242.65	6727562
1376	2024-12-11 16:00:00	246.41	246.4900	246.02	246.17	6719878

673 2024-12-05 16:00:00 243.04 243.0600 242.91 242.99 6066927

```
[31]: top_trading = df.sort_values('timestamp', ascending=False).head(10)
top_trading
```

```
[31]:
```

		timestamp	open	high	low	close	volume
1966	2024-12-16	19:55:00	250.9000	251.0400	250.9000	251.0000	989
1965	2024-12-16	19:50:00	251.0000	251.0000	250.9993	250.9999	1706
1964	2024-12-16	19:45:00	250.9000	250.9900	250.8700	250.9900	1955
1963	2024-12-16	19:40:00	250.8914	250.9286	250.8400	250.8500	2820
1962	2024-12-16	19:35:00	250.8700	250.8907	250.8700	250.8907	400
1961	2024-12-16	19:25:00	250.9000	250.9000	250.9000	250.9000	200
1960	2024-12-16	19:15:00	250.8800	250.8800	250.8800	250.8800	497
1959	2024-12-16	19:10:00	250.8300	250.8900	250.8000	250.8900	538
1958	2024-12-16	19:00:00	250.8100	250.8100	250.8100	250.8100	100
1957	2024-12-16	18:55:00	250.8800	250.9700	250.8700	250.8700	307

```
[33]: trading_corr= df.groupby('high')['low'].mean().sort_values(ascending=False).
↪reset_index()
trading_corr
```

```
[33]:
```

	high	low
0	251.2000	251.1650
1	251.1300	251.1300
2	251.1700	251.1100
3	251.0995	251.0995
4	251.3800	251.0700
..
972	237.4400	237.0500
973	237.1200	236.9800
974	236.8600	236.8600
975	237.3800	236.7800
976	237.9500	236.0000

[977 rows x 2 columns]

```
[34]: trading_corr= df.groupby('open')['close'].mean().sort_values(ascending=False).
↪reset_index()
trading_corr
```

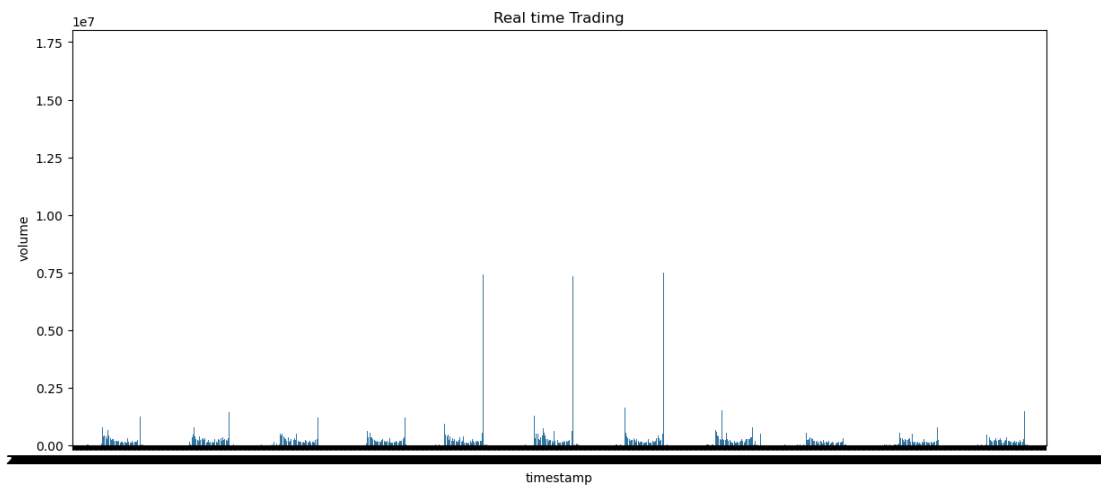
```
[34]:
```

	open	close
0	251.0882	251.285900
1	251.2000	251.200000
2	251.1100	251.170000
3	251.1700	251.160000
4	251.1050	251.150100
...

```
1016  237.1200  237.143333
1017  237.1300  237.140000
1018  237.0700  237.070000
1019  237.1000  236.920000
1020  236.8600  236.860000
```

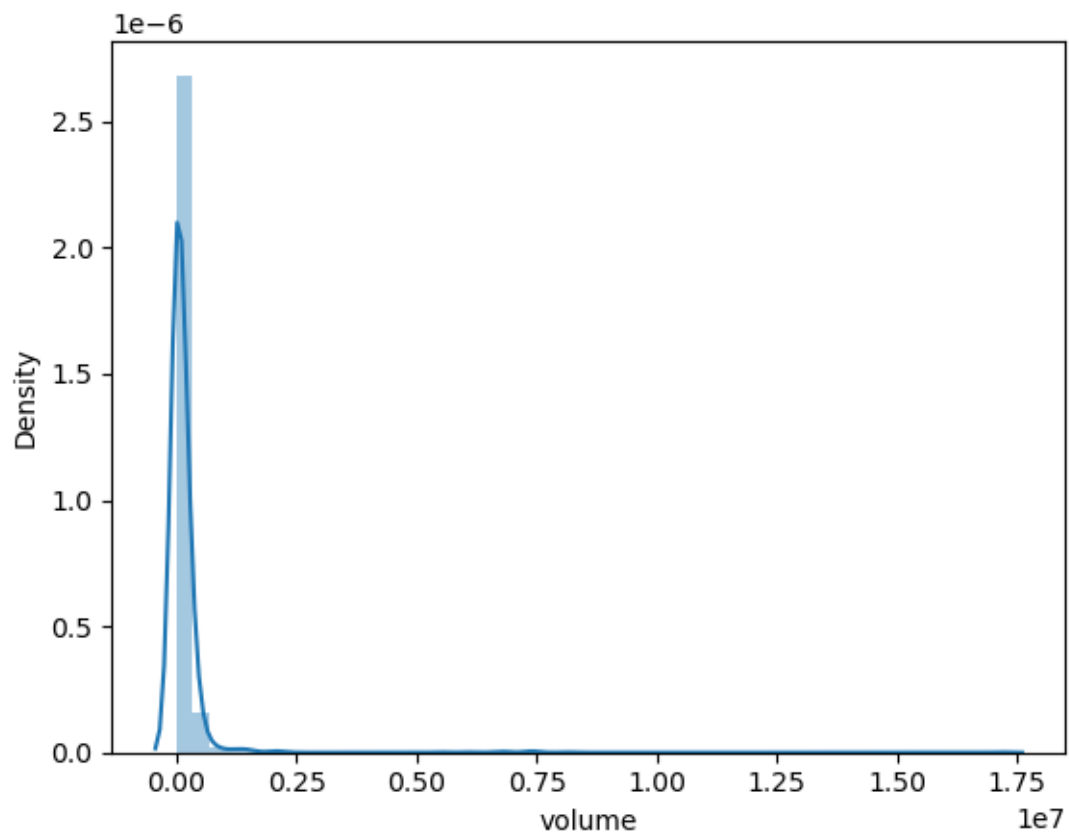
```
[1021 rows x 2 columns]
```

```
[47]: plt.figure(figsize=(14,6))
      sns.barplot(df,x='timestamp',y='volume')
      plt.title('Real time Trading')
      plt.show()
```

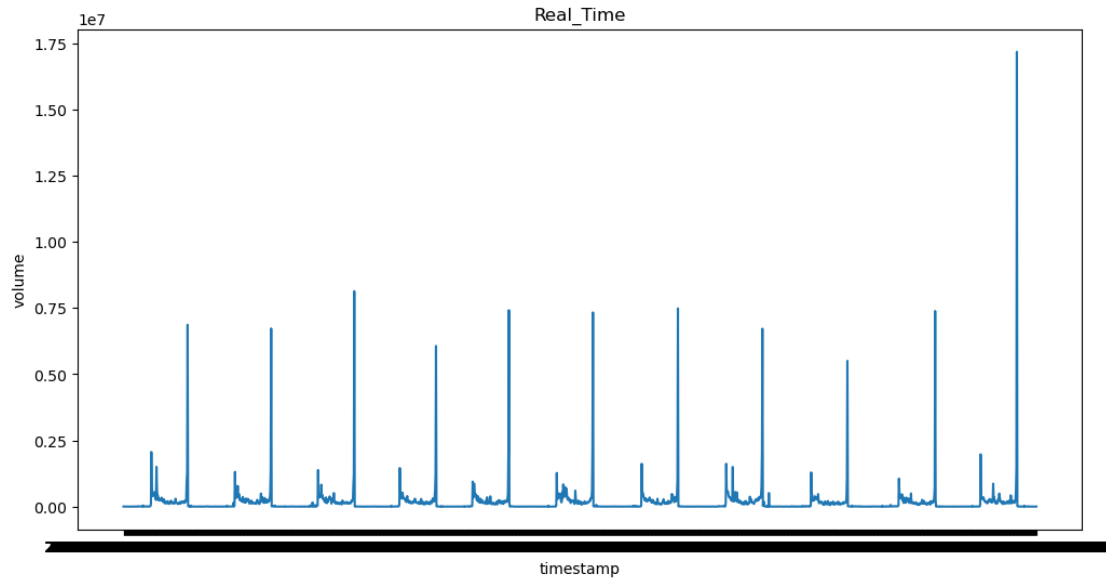


```
[48]: sns.distplot(df.volume)
```

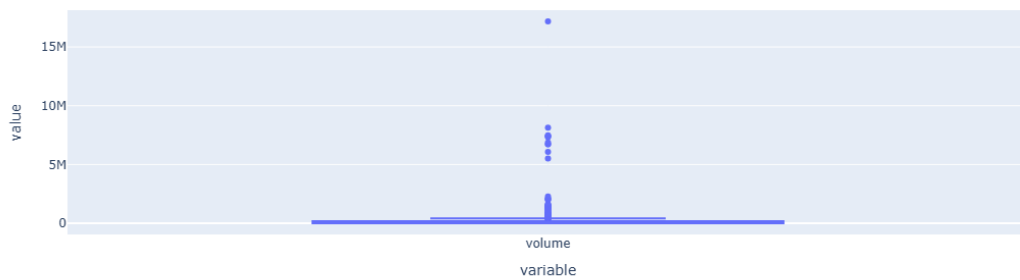
```
[48]: <Axes: xlabel='volume', ylabel='Density'>
```

```
[50]: plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='timestamp', y='volume')
plt.title('Real_Time')
plt.show()
```

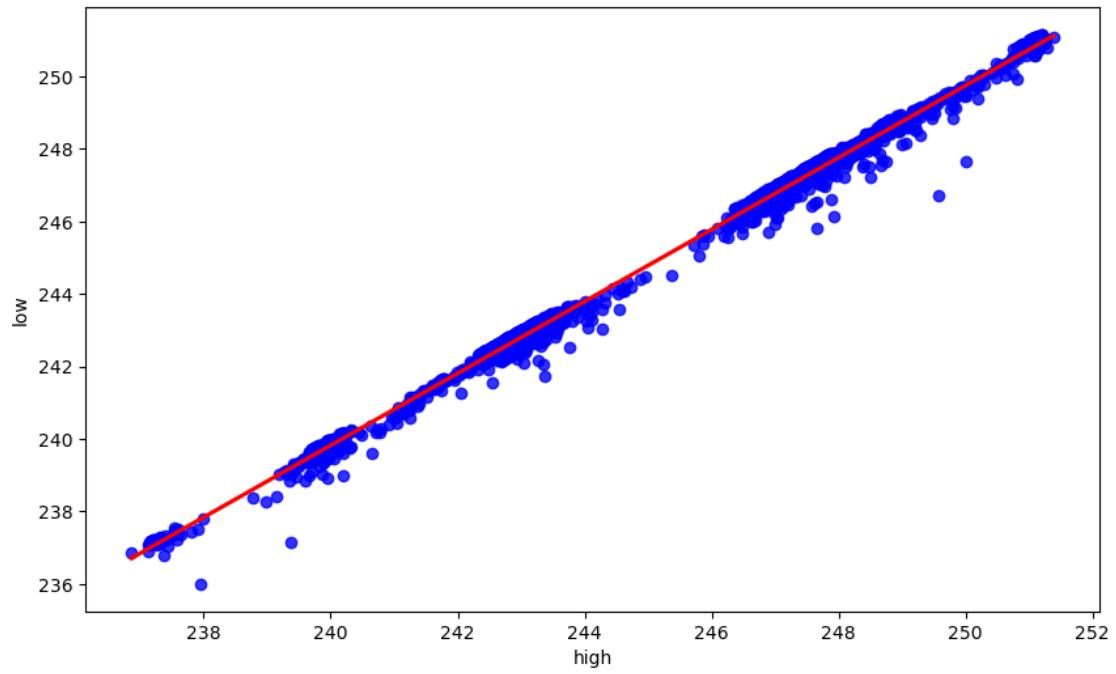


```
[52]: px.box(df.volume)
```



```
[53]: plt.figure(figsize = (10, 6))
sns.regplot(data = df, y = "low" , x = "high", color = "b", line_kws={"color":
↪ "red"})
```

```
[53]: <Axes: xlabel='high', ylabel='low'>
```



[]: