```
import pandas as pd

#Importing the data

data = pd.read_excel('/BankNiftyFutures_Data5Min.xlsx')

data.head()
```

	Ticker	Date	Final Date format	Time	Open 5	High 5	Low 5	Clo
0	BANKNIFTY_F1	2015-01-01	20150101	09:20:00	18861.3496	18867.9492	18793.0000	18808
1	BANKNIFTY_F1	2015-01-01	20150101	09:25:00	18808.0000	18835.0000	18802.1992	18829
2	BANKNIFTY_F1	2015-01-01	20150101	09:30:00	18826.9492	18833.4004	18820.0000	18826
3	BANKNIFTY_F1	2015-01-01	20150101	09:35:00	18825.0508	18829.0996	18810.0000	18810
4	BANKNIFTY_F1	2015-01-01	20150101	09:40:00	18805.0000	18808.9492	18775.0000	18803

```
data.shape
```

```
(91737, 10)
```

```
data.isnull().sum()
```

```
Ticker              0
Date                0
Final Date format   0
Time                0
Open 5              0
High 5              0
Low 5               0
Close 5             0
Volume 5            0
Year                0
dtype: int64
```

```
data.dtypes
```

```
Ticker                      object
Date                datetime64[ns]
Final Date format            int64
Time                        object
Open 5                     float64
High 5                     float64
Low 5                      float64
```

```
Close 5                    float64
Volume 5                     int64
Year                         int64
dtype: object
```

```
import datetime
```

```
data['Date '] = pd.to_datetime(data['Date'], errors='coerce').dt.floor('d')
```

```
data.head()
```

	Ticker	Date	Final Date format	Time	Open 5	High 5	Low 5	Clo
0	BANKNIFTY_F1	2015-01-01	20150101	09:20:00	18861.3496	18867.9492	18793.0000	18808
1	BANKNIFTY_F1	2015-01-01	20150101	09:25:00	18808.0000	18835.0000	18802.1992	18829
2	BANKNIFTY_F1	2015-01-01	20150101	09:30:00	18826.9492	18833.4004	18820.0000	18826
3	BANKNIFTY_F1	2015-01-01	20150101	09:35:00	18825.0508	18829.0996	18810.0000	18810
4	BANKNIFTY_F1	2015-01-01	20150101	09:40:00	18805.0000	18808.9492	18775.0000	18803

```
data.dtypes
```

```
Ticker                      object
Date                datetime64[ns]
Final Date format            int64
Time                        object
Open 5                     float64
High 5                     float64
Low 5                      float64
Close 5                    float64
Volume 5                     int64
Year                         int64
Date                datetime64[ns]
dtype: object
```

```
data.loc[:,'Date'] = pd.to_datetime(data.Date.astype(str)+' '+data.Time.astype(str))
```

```
data.drop(data.columns[[-1,]], axis=1, inplace=True)
```

```
data.head(5)
```

	Ticker	Date	Final Date format	Time	Open 5	High 5	Low 5	
0	BANKNIFTY_F1	2015-01-01 09:20:00	20150101	09:20:00	18861.3496	18867.9492	18793.0000	188
1	BANKNIFTY_F1	2015-01-01 09:25:00	20150101	09:25:00	18808.0000	18835.0000	18802.1992	188
2	BANKNIFTY_F1	2015-01-01	20150101	09:30:00	18826.9492	18833.4004	18820.0000	188

```
del data['Time']
```

```
del data['Final Date format']
```

```
data.rename(columns = {'Date':'Date_Time'}, inplace = True)
```

```
data.head(5)
```

	Ticker	Date_Time	Open 5	High 5	Low 5	Close 5	Volume 5	
0	BANKNIFTY_F1	2015-01-01 09:20:00	18861.3496	18867.9492	18793.0000	18808.3496	84825	2
1	BANKNIFTY_F1	2015-01-01 09:25:00	18808.0000	18835.0000	18802.1992	18829.0000	40925	2
2	BANKNIFTY_F1	2015-01-						

```
data.dtypes
```

```
Ticker              object
Date_Time    datetime64[ns]
Open 5              float64
High 5              float64
Low 5               float64
Close 5             float64
Volume 5              int64
Year                  int64
dtype: object
```

```
data['Timestamp'] = data[['Date_Time']].apply(lambda x: x[0].timestamp(), axis=1).astype(i
```

```
data.head()
```

	Ticker	Date_Time	Open 5	High 5	Low 5	Close 5	Volume 5
0	BANKNIFTY_F1	2015-01-01 09:20:00	18861.3496	18867.9492	18793.0000	18808.3496	84825

```
# Convert the Timestamp column to the correct format
data['Timestamp'] = pd.to_datetime(data['Timestamp'], unit='s')
```

```
# Index by time to allow us to use .resample()
data.set_index('Timestamp', inplace=True)
```

```
import numpy as np
```

```
# Resample and Aggregate appropriately.
df = data.groupby(pd.Grouper(freq='15Min',closed='right',label='right')).agg({
                                    "Open 5":  "first",
                                    "High 5":  "max",
                                    "Low 5":   "min",
                                    "Close 5": "last",
                                    "Volume 5": "sum"

                                })
```

```
df
```

Timestamp	Open 5	High 5	Low 5	Close 5	Volume 5
2015-01-01 09:30:00	18861.3496	18867.9492	18793.0000	18826.0996	152750
2015-01-01 09:45:00	18825.0508	18829.9004	18775.0000	18821.0000	104675
2015-01-01 10:00:00	18821.0000	18850.0000	18820.0000	18838.9492	66825
2015-01-01 10:15:00	18835.0508	18858.5996	18830.6504	18836.4004	49100
2015-01-01 10:30:00	18835.6992	18848.0000	18828.5996	18830.0000	16600
...	...	...	...	...	...
2019-12-31 14:30:00	32444.0000	32450.0000	32384.0000	32409.8496	96960
2019-12-31 14:45:00	32409.8496	32459.0000	32401.1992	32426.0996	56520
2019-12-31 15:00:00	32426.0996	32439.9492	32374.3008	32382.9492	114820
2019-12-31 15:15:00	32380.6504	32407.6992	32376.6504	32382.0000	95220
2019-12-31 15:30:00	32383.5000	32392.6992	32320.0000	32379.9004	259300

175225 rows × 5 columns

```
df.reset_index(level=0, inplace=True)
```

```
df['year'] = pd.Index(df['Timestamp']).year
df.head()
```

	Timestamp	Open 5	High 5	Low 5	Close 5	Volume 5	year
0	2015-01-01 09:30:00	18861.3496	18867.9492	18793.0000	18826.0996	152750	2015
1	2015-01-01 09:45:00	18825.0508	18829.9004	18775.0000	18821.0000	104675	2015
2	2015-01-01 10:00:00	18821.0000	18850.0000	18820.0000	18838.9492	66825	2015
3	2015-01-01 10:15:00	18835.0508	18858.5996	18830.6504	18836.4004	49100	2015
4	2015-01-01 10:30:00	18835.6992	18848.0000	18828.5996	18830.0000	16600	2015

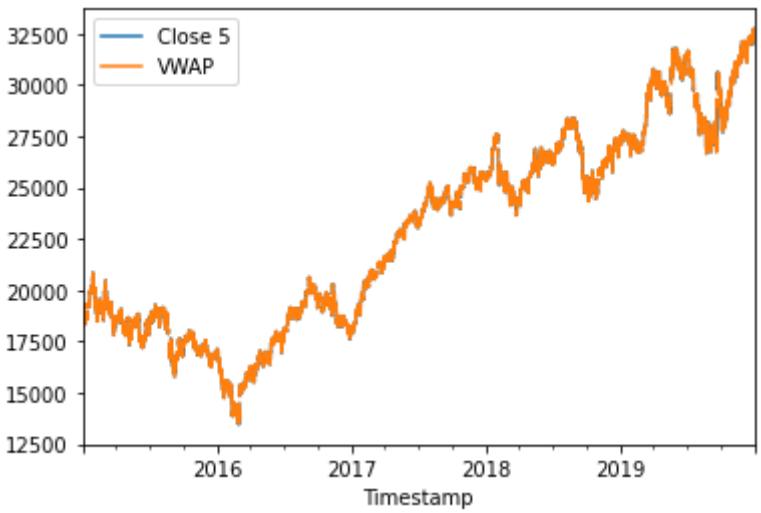
```
def calculateVwap(df):
    df['TP'] = (df['High 5']+df['Low 5']+df['Close 5'])/3.0
    df['TradedValue']  = df['TP']*df['Volume 5']
    df['CumVolume'] = df['Volume 5'].cumsum()
    df['CumTradedValue'] = df['TradedValue'].cumsum()
    df['VWAP'] = df['CumTradedValue'] /df['CumVolume']
    return df


VWAP = df.groupby('Timestamp').apply(calculateVwap)
VWAP
```

	Timestamp	Open 5	High 5	Low 5	Close 5	Volume 5	year	
0	2015-01-01 09:30:00	18861.3496	18867.9492	18793.0000	18826.0996	152750	2015	1882
1	2015-01-01 09:45:00	18825.0508	18829.9004	18775.0000	18821.0000	104675	2015	1880
2	2015-01-01 10:00:00	18821.0000	18850.0000	18820.0000	18838.9492	66825	2015	1883
3	2015-01-01	18835.0508	18858.5996	18830.6504	18836.4004	49100	2015	1884

```
import matplotlib.pyplot as plt

VWAP.plot(x="Timestamp", y=["Close 5", "VWAP"])
plt.show()
```




```
VWAP['Buy/Sell'] = np.where(VWAP['VWAP']<VWAP['Close 5'], 'Buy', 'Sell')

VWAP.head(20)
```

5	2015-01-01 10:45:00	18830.0000	18841.1504	18825.0000	18829.1992	14175	2015	18831.783
6	2015-01-01 11:00:00	18830.1504	18846.2500	18830.0996	18843.2500	15650	2015	18839.866
7	2015-01-01 11:15:00	18843.5000	18848.6992	18832.0000	18840.0000	18675	2015	18840.233
8	2015-01-01 11:30:00	18840.0000	18889.9004	18840.0000	18872.0000	87525	2015	18867.300
9	2015-01-01 11:45:00	18871.0508	18874.0000	18851.9492	18869.0000	18075	2015	18864.983
10	2015-01-01 12:00:00	18868.0508	18889.0000	18857.0000	18878.0996	47275	2015	18874.699
11	2015-01-01 12:15:00	18875.9492	18880.0000	18865.0000	18875.0000	20675	2015	18873.333
12	2015-01-01 12:30:00	18875.0000	18875.0000	18855.3496	18857.0000	13800	2015	18862.449
13	2015-01-01 12:45:00	18858.0996	18867.9004	18850.0000	18857.0000	16025	2015	18858.300
14	2015-01-01 13:00:00	18857.4492	18873.0000	18857.0000	18865.0000	9925	2015	18865.000
15	2015-01-01 13:15:00	18865.0000	18865.0000	18857.3496	18860.0000	6575	2015	18860.783
16	2015-01-01 13:30:00	18860.0000	18869.7500	18853.0000	18856.0000	9825	2015	18859.583
17	2015-01-01 13:45:00	18857.0000	18879.5996	18853.0508	18873.8008	15625	2015	18868.817
18	2015-01-01 14:00:00	18873.8008	18909.9492	18861.1504	18892.3008	60975	2015	18887.800
19	2015-01-01 14:15:00	18897.0000	18897.0000	18870.0000	18887.3496	34000	2015	18884.783

```
VWAP['transaction'] = np.where(VWAP['Buy/Sell'] == 'Sell', -1 * VWAP['Close 5'], VWAP['Clo
```

```
VWAP['transaction']
```

```
0        -18826.0996
1         18821.0000
2         18838.9492
3        -18836.4004
4        -18830.0000
              ...
175220   -32409.8496
175221   -32426.0996
175222   -32382.9492
175223   -32382.0000
175224    32379.9004
Name: transaction, Length: 175225, dtype: float64
```

```
VWAP['transaction'].sum()
```

```
10137370.475199997
```

```
TransactionBy_date=VWAP.resample('D', on='Timestamp').transaction.sum()
```

```
TransactionBy_date.astype(int)
```

```
Timestamp
2015-01-01    -18661
2015-01-02    211085
2015-01-03         0
2015-01-04         0
2015-01-05     19284
               ...
2019-12-27     97075
2019-12-28         0
2019-12-29         0
2019-12-30    -31677
2019-12-31    -97205
Freq: D, Name: transaction, Length: 1826, dtype: int64
```

```
TransactionBy_year=VWAP.resample('Y', on='Timestamp').transaction.sum()
```

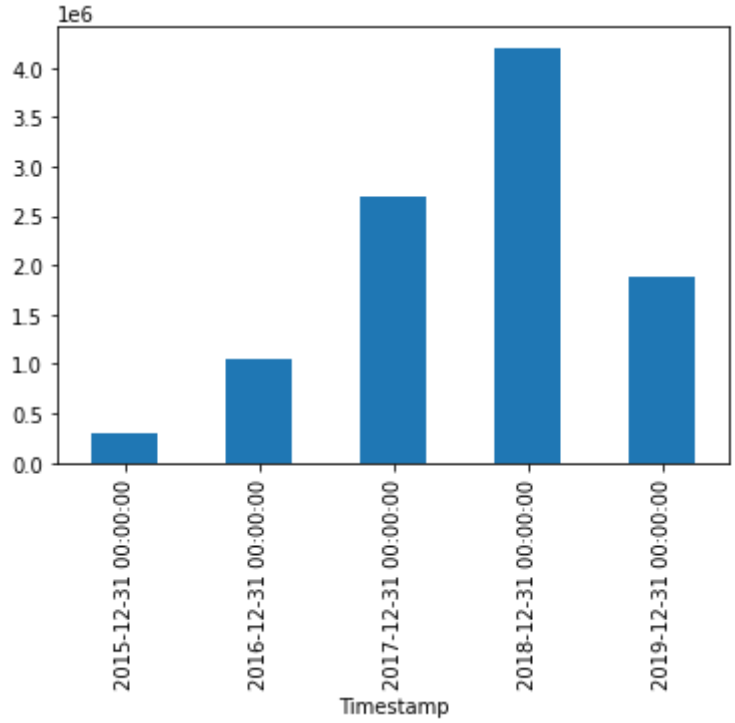
```
TransactionBy_year.astype(int)
```

```
Timestamp
2015-12-31     295679
2016-12-31    1053645
2017-12-31    2697544
2018-12-31    4203946
```

```
2019-12-31    1886553
Freq: A-DEC, Name: transaction, dtype: int64
```

```
TransactionBy_year.plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8d5673c490>
```



Best Performing Year is 2018 with Total profit = 4203946

✓ 0s completed at 10:55 AM