

Moving_Average_Ribbon

September 29, 2021

1 Moving Average Ribbon

<https://www.investopedia.com/terms/m/movingaverageribbon.asp>

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")

# fix_yahoo_finance is used to fetch data
import fix_yahoo_finance as yf
yf.pdr_override()
```

```
[2]: # input
symbol = 'AAPL'
start = '2016-08-01'
end = '2019-01-01'

# Read data
df = yf.download(symbol,start,end)

# View Columns
df.head()
```

[*****100%*****] 1 of 1 downloaded

```
[2]:
```

	Open	High	Low	Close	Adj Close	\
Date						
2016-08-01	104.410004	106.150002	104.410004	106.050003	101.316849	
2016-08-02	106.050003	106.070000	104.000000	104.480003	99.816925	
2016-08-03	104.809998	105.839996	104.769997	105.790001	101.068451	
2016-08-04	105.580002	106.000000	105.279999	105.870003	101.692818	
2016-08-05	106.269997	107.650002	106.180000	107.480003	103.239288	

	Volume
Date	

```
2016-08-01    38167900
2016-08-02    33816600
2016-08-03    30202600
2016-08-04    27408700
2016-08-05    40553400
```

```
[3]: df['MA10'] = df['Adj Close'].rolling(10).mean()
df['MA20'] = df['Adj Close'].rolling(20).mean()
df['MA30'] = df['Adj Close'].rolling(30).mean()
df['MA40'] = df['Adj Close'].rolling(40).mean()
df['MA50'] = df['Adj Close'].rolling(50).mean()
df['MA60'] = df['Adj Close'].rolling(60).mean()
```

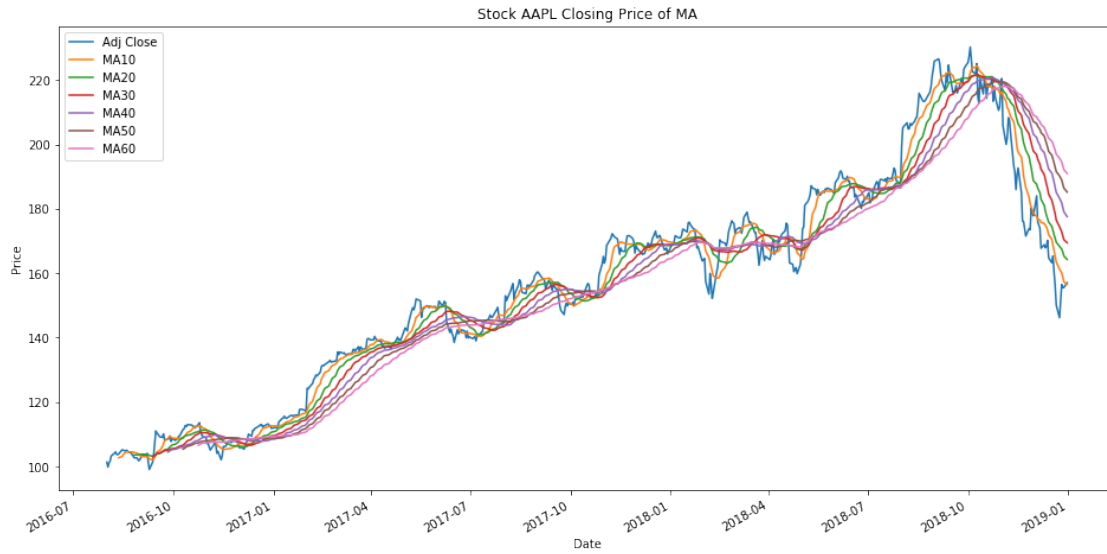
```
[4]: import talib as ta
```

```
[5]: df['SMA5'] = ta.SMA(df['Adj Close'], timeperiod=5)
df['SMA15'] = ta.SMA(df['Adj Close'], timeperiod=15)
df['SMA25'] = ta.SMA(df['Adj Close'], timeperiod=25)
df['SMA35'] = ta.SMA(df['Adj Close'], timeperiod=35)
df['SMA45'] = ta.SMA(df['Adj Close'], timeperiod=45)
```

```
[6]: df['EMA5'] = ta.EMA(df['Adj Close'], timeperiod=5)
df['EMA15'] = ta.EMA(df['Adj Close'], timeperiod=15)
df['EMA25'] = ta.EMA(df['Adj Close'], timeperiod=25)
df['EMA35'] = ta.EMA(df['Adj Close'], timeperiod=35)
df['EMA45'] = ta.EMA(df['Adj Close'], timeperiod=45)
```

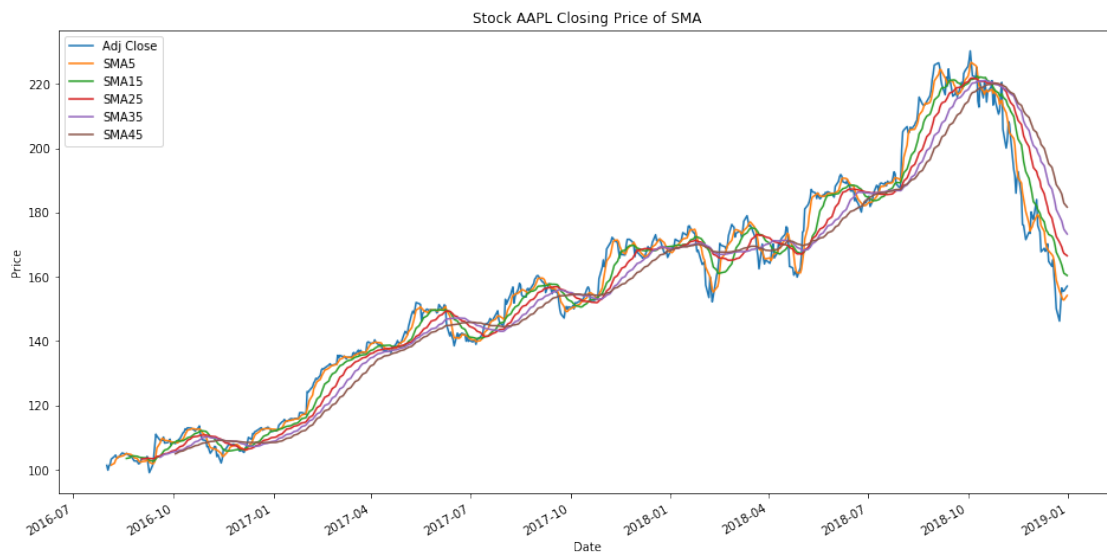
```
[7]: # Line Chart
df[['Adj Close', 'MA10', 'MA20', 'MA30', 'MA40', 'MA50', 'MA60']].
    .plot(figsize=(16,8))
plt.title('Stock ' + symbol + ' Closing Price of MA')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend(loc='Best')
```

```
[7]: <matplotlib.legend.Legend at 0x18f488e2b00>
```



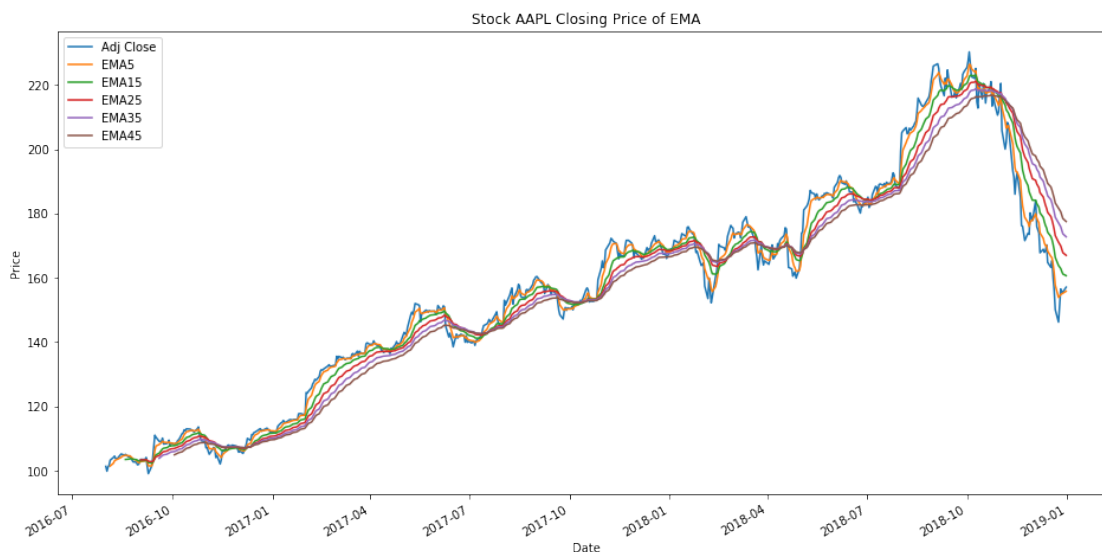
```
[8]: df[['Adj Close', 'SMA5', 'SMA15', 'SMA25', 'SMA35', 'SMA45']].
      ↪plot(figsize=(16,8))
plt.title('Stock ' + symbol + ' Closing Price of SMA')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend(loc='Best')
```

[8]: <matplotlib.legend.Legend at 0x18f48f33160>



```
[9]: df[['Adj Close', 'EMA5', 'EMA15', 'EMA25', 'EMA35', 'EMA45']].
      ↪plot(figsize=(16,8))
plt.title('Stock ' + symbol + ' Closing Price of EMA')
plt.xlabel('Date')
plt.ylabel('Price')
```

```
[9]: Text(0,0.5,'Price')
```



```
[10]: from matplotlib import dates as mdates
import datetime as dt

dfc = df.copy()
dfc['VolumePositive'] = dfc['Open'] < dfc['Adj Close']
#dfc = dfc.dropna()
dfc = dfc.reset_index()
dfc['Date'] = mdates.date2num(dfc['Date'].astype(dt.date))
dfc.head()
```

```
[10]:
```

	Date	Open	High	Low	Close	Adj Close	\
0	736177.0	104.410004	106.150002	104.410004	106.050003	101.316849	
1	736178.0	106.050003	106.070000	104.000000	104.480003	99.816925	
2	736179.0	104.809998	105.839996	104.769997	105.790001	101.068451	
3	736180.0	105.580002	106.000000	105.279999	105.870003	101.692818	
4	736181.0	106.269997	107.650002	106.180000	107.480003	103.239288	

	Volume	MA10	MA20	MA30	...	SMA15	SMA25	SMA35	SMA45	\
0	38167900	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	
1	33816600	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	

2	30202600	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
3	27408700	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
4	40553400	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN

	EMA5	EMA15	EMA25	EMA35	EMA45	VolumePositive
0	NaN	NaN	NaN	NaN	NaN	False
1	NaN	NaN	NaN	NaN	NaN	False
2	NaN	NaN	NaN	NaN	NaN	False
3	NaN	NaN	NaN	NaN	NaN	False
4	101.426866	NaN	NaN	NaN	NaN	False

[5 rows x 24 columns]

```
[11]: from mpl_finance import candlestick_ohlc

fig = plt.figure(figsize=(16,8))
ax1 = plt.subplot(111)
candlestick_ohlc(ax1,dfc.values, width=0.5, colorup='g', colordown='r', alpha=1.
↪0)
ax1.plot(dfc.Date, dfc[['EMA5', 'EMA15', 'EMA25', 'EMA35', 'EMA45']])
ax1.xaxis_date()
ax1.xaxis.set_major_formatter(mdates.DateFormatter('%d-%m-%Y'))
ax1.grid(True, which='both')
ax1.minorticks_on()
ax1v = ax1.twinx()
ax1v.fill_between(dfc.Date, 0, dfc.Volume[0:], facecolor='#0079a3', alpha=0.4)
ax1v.axes.yaxis.set_ticklabels([])
ax1v.set_ylim(0, 3*df.Volume.max())
ax1.set_title('Stock ' + symbol + ' Closing Price')
ax1.set_ylabel('Price')
```

```
[11]: Text(0,0.5,'Price')
```

