

SMMA

September 29, 2021

1 Smoothed Moving Average (SMMA)

https://www.metatrader5.com/en/terminal/help/indicators/trend_indicators/ma#smma

```
[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 = '2018-08-01'
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						
2018-08-01	199.130005	201.759995	197.309998	201.500000	199.243088	
2018-08-02	200.580002	208.380005	200.350006	207.389999	205.067123	
2018-08-03	207.029999	208.740005	205.479996	207.990005	205.660416	
2018-08-06	208.000000	209.250000	207.070007	209.070007	206.728317	
2018-08-07	209.320007	209.500000	206.759995	207.110001	204.790268	

	Volume
Date	

```

2018-08-01    67935700
2018-08-02    62404000
2018-08-03    33447400
2018-08-06    25425400
2018-08-07    25587400

```

```
[3]: df.shape
```

```
[3]: (105, 6)
```

```

[4]: n = 10
SMMA = np.array([np.nan] * len(df['Adj Close']))
SMMA[n - 2] = df['Adj Close'][:n - 1].mean()
for i in range(n - 1, len(df['Adj Close'])):
    SMMA[i] = (SMMA[i - 1] * (n - 2) + 2 * df['Adj Close'][i]) / n

```

```
[5]: SMMA
```

```

[5]: array([
         nan,         nan,         nan,         nan,
         nan,         nan,         nan,         nan,
    205.12649544, 205.72680976, 206.3043024 , 207.37753612,
    209.0815333 , 210.02401264, 210.69464351, 211.23313181,
    211.75124245, 212.29869296, 213.08990337, 214.07214849,
    215.50886939, 217.06507892, 218.82602553, 220.37965283,
    221.32685766, 221.33645553, 220.98691642, 220.11787794,
    220.51811215, 220.28659972, 221.16113098, 221.35072478,
    220.31962102, 219.56617962, 218.9892261 , 218.85709748,
    218.28106058, 218.44138587, 218.84748429, 218.82109963,
    219.69898491, 220.55807413, 221.5469933 , 222.73900464,
    224.24630011, 224.64244729, 224.22508463, 223.78799931,
    224.05353484, 222.18022108, 220.30252046, 220.32051677,
    219.39225941, 219.60024253, 219.57611683, 218.53080926,
    218.34747761, 218.46673629, 218.97492703, 217.86529622,
    217.91230838, 217.2553289 , 215.92402932, 215.06934746,
    215.48900157, 216.49152785, 214.36834428, 211.50090703,
    209.63958982, 209.37697346, 209.02150977, 207.93657121,
    205.01741497, 202.29574658, 199.03705206, 197.34815965,
    196.41923512, 194.1486449 , 190.56375532, 187.65601525,
    184.4356582 , 182.32338316, 180.55788873, 180.47977038,
    180.14046311, 179.67584709, 180.54682387, 179.62455009,
    178.49441248, 176.34962538, 174.8548467 , 173.46585056,
    172.44825425, 172.0025944 , 170.55673852, 169.09337062,
    168.34685829, 166.71807203, 164.60651023, 161.70246898,
    158.60256959, 158.18181767, 157.64208493, 157.22623195,
    157.19426076])

```

```
[6]: x = SMMA.reshape(-1,1)
      data = pd.DataFrame.from_records(x)
      data
```

```
[6]:      0
0      NaN
1      NaN
2      NaN
3      NaN
4      NaN
5      NaN
6      NaN
7      NaN
8    205.126495
9    205.726810
10   206.304302
11   207.377536
12   209.081533
13   210.024013
14   210.694644
15   211.233132
16   211.751242
17   212.298693
18   213.089903
19   214.072148
20   215.508869
21   217.065079
22   218.826026
23   220.379653
24   221.326858
25   221.336456
26   220.986916
27   220.117878
28   220.518112
29   220.286600
..      ...
75   197.348160
76   196.419235
77   194.148645
78   190.563755
79   187.656015
80   184.435658
81   182.323383
82   180.557889
83   180.479770
84   180.140463
85   179.675847
```

```

86    180.546824
87    179.624550
88    178.494412
89    176.349625
90    174.854847
91    173.465851
92    172.448254
93    172.002594
94    170.556739
95    169.093371
96    168.346858
97    166.718072
98    164.606510
99    161.702469
100   158.602570
101   158.181818
102   157.642085
103   157.226232
104   157.194261

```

[105 rows x 1 columns]

```
[7]: df['SMMA'] = data.values
      df.head(10)
```

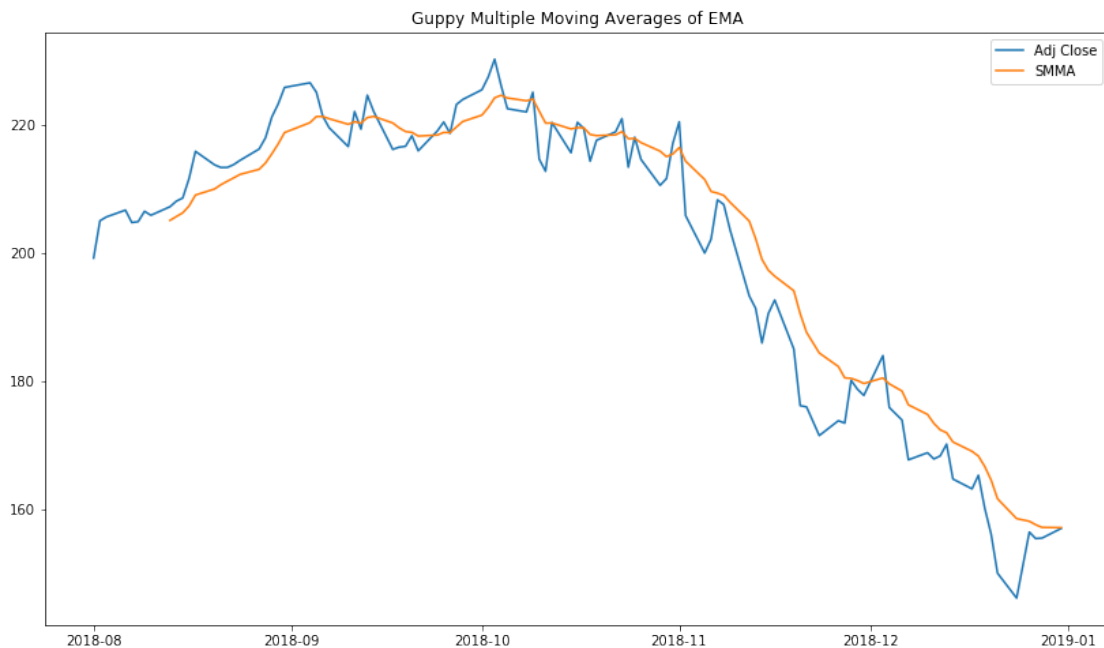
```
[7]:
```

	Open	High	Low	Close	Adj Close \
Date					
2018-08-01	199.130005	201.759995	197.309998	201.500000	199.243088
2018-08-02	200.580002	208.380005	200.350006	207.389999	205.067123
2018-08-03	207.029999	208.740005	205.479996	207.990005	205.660416
2018-08-06	208.000000	209.250000	207.070007	209.070007	206.728317
2018-08-07	209.320007	209.500000	206.759995	207.110001	204.790268
2018-08-08	206.050003	207.809998	204.520004	207.250000	204.928696
2018-08-09	207.279999	209.779999	207.199997	208.880005	206.540436
2018-08-10	207.360001	209.100006	206.669998	207.529999	205.925232
2018-08-13	207.699997	210.949997	207.699997	208.869995	207.254883
2018-08-14	210.160004	210.559998	208.259995	209.750000	208.128067

	Volume	SMMA
Date		
2018-08-01	67935700	NaN
2018-08-02	62404000	NaN
2018-08-03	33447400	NaN
2018-08-06	25425400	NaN
2018-08-07	25587400	NaN
2018-08-08	22525500	NaN
2018-08-09	23469200	NaN

2018-08-10	24611200	NaN
2018-08-13	25869100	205.126495
2018-08-14	20748000	205.726810

```
[8]: # Line Chart
plt.figure(figsize=(14,8))
plt.plot(df['Adj Close'])
plt.plot(df['SMMA'])
plt.title('Guppy Multiple Moving Averages of EMA')
plt.legend(loc='best')
plt.show()
```



1.1 Candlestick with SMMA

```
[9]: 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()
```

```
[9]:
```

	Date	Open	High	Low	Close	Adj Close	\
0	736907.0	199.130005	201.759995	197.309998	201.500000	199.243088	
1	736908.0	200.580002	208.380005	200.350006	207.389999	205.067123	
2	736909.0	207.029999	208.740005	205.479996	207.990005	205.660416	
3	736912.0	208.000000	209.250000	207.070007	209.070007	206.728317	
4	736913.0	209.320007	209.500000	206.759995	207.110001	204.790268	

	Volume	SMMA	VolumePositive
0	67935700	NaN	True
1	62404000	NaN	True
2	33447400	NaN	False
3	25425400	NaN	False
4	25587400	NaN	False

```
[10]: from mpl_finance import candlestick_ohlc

fig = plt.figure(figsize=(14,10))
ax1 = plt.subplot(111)
candlestick_ohlc(ax1,dfc.values, width=0.5, colorup='g', colordown='r', alpha=1.
↪0)
ax1.plot(df['SMMA'], color='orange')
ax1.xaxis_date()
ax1.xaxis.set_major_formatter(mdates.DateFormatter('%d-%m-%Y'))
ax1.grid(True, which='both')
ax1.minorticks_on()
ax1v = ax1.twinx()
colors = dfc.VolumePositive.map({True: 'g', False: 'r'})
ax1v.bar(dfc.Date, dfc['Volume'], color=colors, alpha=0.4)
ax1v.axes.yaxis.set_ticklabels([])
ax1v.set_ylim(0, 3*dfc.Volume.max())
ax1.set_title('Stock ' + symbol + ' Closing Price of SMMA')
ax1.set_ylabel('Price')
```

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

