Linear_Weighted_Moving_Average

September 29, 2021

1 Linearly Weighted Moving Average

https://www.investopedia.com/terms/l/linearlyweightedmovingaverage.asp

```python
[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()
```

```python
[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  198.478760
     2018-08-02  200.580002  208.380005  200.350006  207.389999  204.280457
     2018-08-03  207.029999  208.740005  205.479996  207.990005  204.871445
     2018-08-06  208.000000  209.250000  207.070007  209.070007  205.935257
     2018-08-07  209.320007  209.500000  206.759995  207.110001  204.004639

                     Volume
     Date
```

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

[3]:
```python
def linear_weight_moving_average(close, n):
    lwma = [np.nan] * n
    for i in range(n, len(close)):
        lwma.append((close[i - n : i] * (np.arange(n) + 1)).sum()/(np.arange(n
    + 1).sum()))
    return lwma
```

[4]:
```python
df['LWMA'] = linear_weight_moving_average(df['Adj Close'], 5)
```

[5]:
```python
df.head(10)
```

[5]:
```
                 Open        High         Low       Close    Adj Close  \
Date
2018-08-01  199.130005  201.759995  197.309998  201.500000  198.478760
2018-08-02  200.580002  208.380005  200.350006  207.389999  204.280457
2018-08-03  207.029999  208.740005  205.479996  207.990005  204.871445
2018-08-06  208.000000  209.250000  207.070007  209.070007  205.935257
2018-08-07  209.320007  209.500000  206.759995  207.110001  204.004639
2018-08-08  206.050003  207.809998  204.520004  207.250000  204.142532
2018-08-09  209.529999  209.779999  207.199997  208.880005  205.748108
2018-08-10  207.360001  209.100006  206.669998  207.529999  205.135254
2018-08-13  209.309998  210.949997  207.699997  208.869995  206.459793
2018-08-14  210.160004  210.559998  208.259995  209.750000  207.329651

              Volume        LWMA
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  204.361215
2018-08-09  23492600  204.570689
2018-08-10  24611200  204.937770
2018-08-13  25890900  205.002722
2018-08-14  20748000  205.491601
```

[6]:
```python
fig = plt.figure(figsize=(14,10))
ax1 = plt.subplot(2, 1, 1)
ax1.plot(df['Adj Close'])
ax1.set_title('Stock '+ symbol +' Closing Price')
```
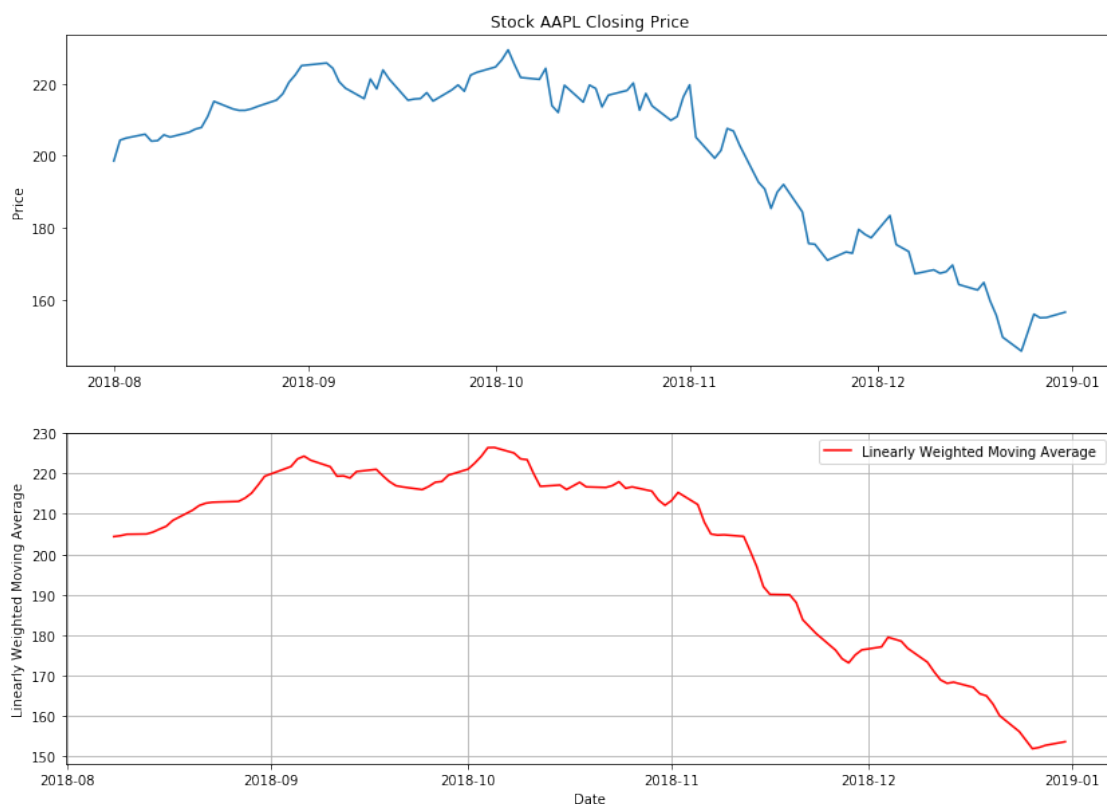
```
ax1.set_ylabel('Price')

ax2 = plt.subplot(2, 1, 2)
ax2.plot(df['LWMA'], label='Linearly Weighted Moving Average', color='red')
#ax2.axhline(y=0, color='blue', linestyle='--')
#ax2.axhline(y=0.5, color='darkblue')
#ax2.axhline(y=-0.5, color='darkblue')
ax2.grid()
ax2.set_ylabel('Linearly Weighted Moving Average')
ax2.set_xlabel('Date')
ax2.legend(loc='best')
```

[6]: <matplotlib.legend.Legend at 0x2105d3625c0>

1.1 Candlestick with Linearly Weighted Moving Average

```
[7]: 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'] = pd.to_datetime(dfc['Date'])
dfc['Date'] = dfc['Date'].apply(mdates.date2num)
dfc.head()
```

[7]:
```
       Date        Open        High         Low       Close    Adj Close  \
0  736907.0  199.130005  201.759995  197.309998  201.500000  198.478760
1  736908.0  200.580002  208.380005  200.350006  207.389999  204.280457
2  736909.0  207.029999  208.740005  205.479996  207.990005  204.871445
3  736912.0  208.000000  209.250000  207.070007  209.070007  205.935257
4  736913.0  209.320007  209.500000  206.759995  207.110001  204.004639

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

[8]:
```python
from mpl_finance import candlestick_ohlc

fig = plt.figure(figsize=(14,10))
ax1 = plt.subplot(2, 1, 1)
candlestick_ohlc(ax1,dfc.values, width=0.5, colorup='g', colordown='r', alpha=1.
 →0)
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*df.Volume.max())
ax1.set_title('Stock '+ symbol +' Closing Price')
ax1.set_ylabel('Price')

ax2 = plt.subplot(2, 1, 2)
ax2.plot(df['LWMA'], label='Linearly Weighted Moving Average', color='red')
ax2.grid()
ax2.set_ylabel('Linearly Weighted Moving Average')
ax2.set_xlabel('Date')
ax2.legend(loc='best')
```

[8]: <matplotlib.legend.Legend at 0x2105d7e29e8>

Stock AAPL Closing Price