Efficiency_Ratio

September 29, 2021

1 Efficiency Ratio (ER)

https://www.marketvolume.com/technicalanalysis/efficiencyratio.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 = '2018-12-01'
end = '2019-02-01'

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

# View Columns
df.head()
```

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

[2]:

Date	Open	High	Low	Close	Adj Close	Volume
2018-12-03	184.460007	184.940002	181.210007	184.820007	183.324753	
2018-12-04	180.949997	182.389999	176.270004	176.690002	175.260513	
2018-12-06	171.759995	174.779999	170.419998	174.720001	173.306473	
2018-12-07	173.490005	174.490005	168.300003	168.490005	167.126862	
2018-12-10	165.000000	170.089996	163.330002	169.600006	168.227890	

```
2018-12-03  40802500
2018-12-04  41344300
2018-12-06  43098400
2018-12-07  42281600
2018-12-10  62026000
```

[3]:
```
n = 10
change = df['Adj Close'].diff(n).abs()
vol = df['Adj Close'].diff().abs().rolling(n).sum()
df['ER'] = change / vol
```

[4]:
```
df.head(20)
```

[4]:

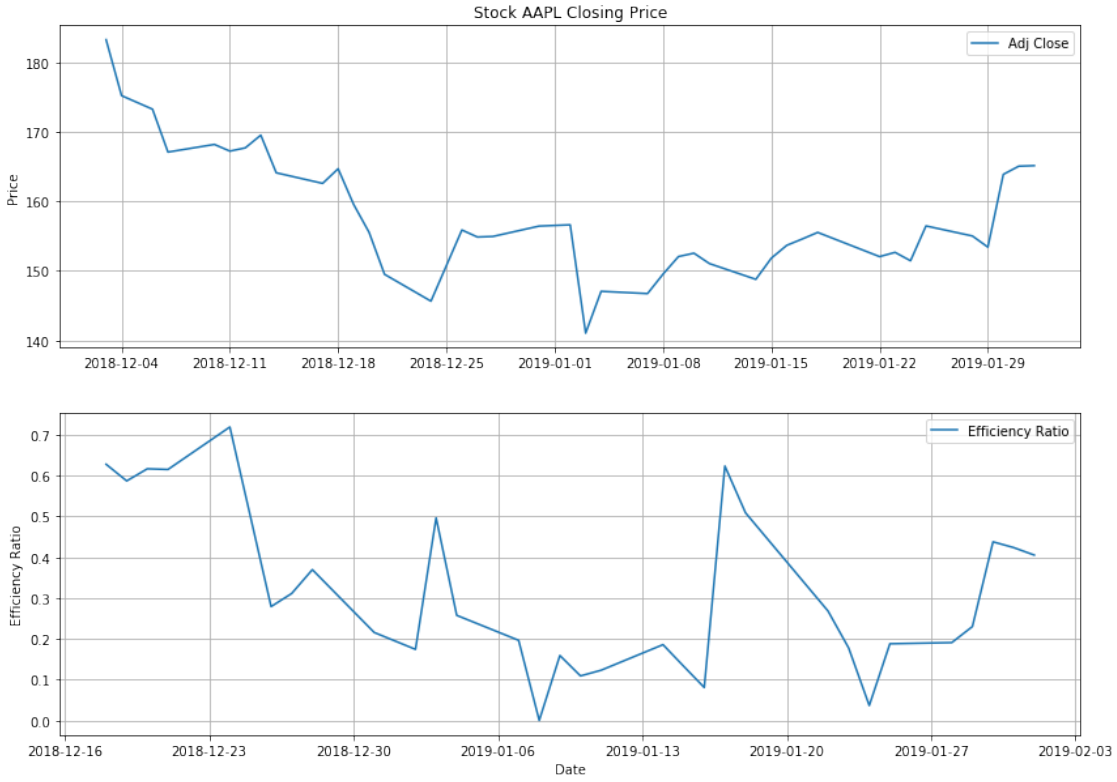
Date	Open	High	Low	Close	Adj Close	Volume	ER
2018-12-03	184.460007	184.940002	181.210007	184.820007	183.324753	40802500	NaN
2018-12-04	180.949997	182.389999	176.270004	176.690002	175.260513	41344300	NaN
2018-12-06	171.759995	174.779999	170.419998	174.720001	173.306473	43098400	NaN
2018-12-07	173.490005	174.490005	168.300003	168.490005	167.126862	42281600	NaN
2018-12-10	165.000000	170.089996	163.330002	169.600006	168.227890	62026000	NaN
2018-12-11	171.660004	171.789993	167.000000	168.630005	167.265732	47281700	NaN
2018-12-12	170.399994	171.919998	169.020004	169.100006	167.731934	35627700	NaN
2018-12-13	170.490005	172.570007	169.550003	170.949997	169.566956	31898600	NaN
2018-12-14	169.000000	169.080002	165.279999	165.479996	164.141220	40703700	NaN
2018-12-17	165.449997	168.350006	162.729996	163.940002	162.613678		
2018-12-18	165.380005	167.529999	164.389999	166.070007	164.726440		
2018-12-19	166.000000	167.449997	159.089996	160.889999	159.588348		
2018-12-20	160.399994	162.110001	155.300003	156.830002	155.561188		
2018-12-21	156.860001	158.160004	149.630005	150.729996	149.510544		
2018-12-24	148.149994	151.550003	146.589996	146.830002	145.642090		
2018-12-26	148.300003	157.229996	146.720001	157.169998	155.898438		
2018-12-27	155.839996	156.770004	150.070007	156.149994	154.886688		
2018-12-28	157.500000	158.520004	154.550003	156.229996	154.966034		
2018-12-31	158.529999	159.360001	156.479996	157.740005	156.463837		
2019-01-02	154.889999	158.850006	154.229996	157.919998	156.642365		

```
2018-12-17  44287900       NaN
2018-12-18  33841500  0.627720
2018-12-19  49047300  0.586924
2018-12-20  64773000  0.616684
2018-12-21  95744600  0.614959
2018-12-24  37169200  0.718978
2018-12-26  58582500  0.279240
2018-12-27  53117100  0.311373
2018-12-28  42291400  0.369664
2018-12-31  35003500  0.215839
2019-01-02  37039700  0.174493
```

[5]:
```
fig = plt.figure(figsize=(14,10))
ax1 = plt.subplot(2, 1, 1)
ax1.plot(df['Adj Close'])
ax1.grid(True, which='both')
ax1.legend(loc='best')
ax1.set_title('Stock '+ symbol +' Closing Price')
ax1.set_ylabel('Price')

ax2 = plt.subplot(2, 1, 2)
ax2.plot(df['ER'], '-', label='Efficiency Ratio')
#ax2.axhline(y=0,color='r')
ax2.grid(True, which='both')
ax2.set_ylabel('Efficiency Ratio')
ax2.set_xlabel('Date')
ax2.legend(loc='best')
```

[5]:
```
<matplotlib.legend.Legend at 0x2793fa55c50>
```



1.1 Candlestick with ER

[6]:
```
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()
```

[6]:

	Date	Open	High	Low	Close	Adj Close	Volume	ER	VolumePositive
0	737031.0	184.460007	184.940002	181.210007	184.820007	183.324753	40802500	NaN	False
1	737032.0	180.949997	182.389999	176.270004	176.690002	175.260513			
2	737034.0	171.759995	174.779999	170.419998	174.720001	173.306473			
3	737035.0	173.490005	174.490005	168.300003	168.490005	167.126862			
4	737038.0	165.000000	170.089996	163.330002	169.600006	168.227890			

```
1  41344300 NaN           False
2  43098400 NaN            True
3  42281600 NaN           False
4  62026000 NaN            True
```

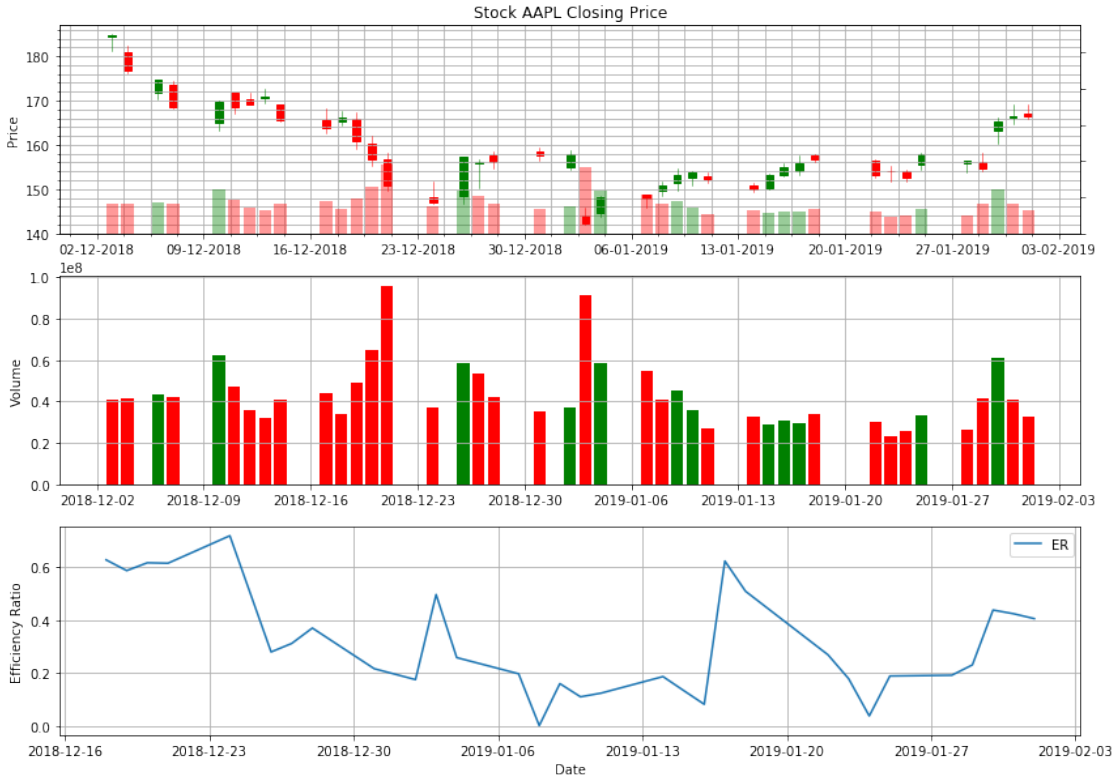
[7]:
```
from mpl_finance import candlestick_ohlc

fig = plt.figure(figsize=(14,10))
ax1 = plt.subplot(3, 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(3, 1, 2)
df['VolumePositive'] = df['Open'] < df['Adj Close']
ax2.bar(df.index, df['Volume'], color=df.VolumePositive.map({True: 'g', False:␣
 ↪'r'}), label='macdhist')
ax2.grid()
ax2.set_ylabel('Volume')

ax3 = plt.subplot(3, 1, 3)
ax3.plot(df['ER'])
ax3.grid()
ax3.set_ylabel('Efficiency Ratio')
ax3.set_xlabel('Date')
ax3.legend()
```

[7]:
```
<matplotlib.legend.Legend at 0x2793fe57be0>
```


Stock AAPL Closing Price
Price
180
170
160
150
140
02-12-2018
09-12-2018
16-12-2018
23-12-2018
30-12-2018
06-01-2019
13-01-2019
20-01-2019
27-01-2019
03-02-2019
1e8
Volume
1.0
0.8
0.6
0.4
0.2
0.0
2018-12-02
2018-12-09
2018-12-16
2018-12-23
2018-12-30
2019-01-06
2019-01-13
2019-01-20
2019-01-27
2019-02-03
Efficiency Ratio
ER
0.6
0.4
0.2
0.0
2018-12-16
2018-12-23
2018-12-30
2019-01-06
2019-01-13
2019-01-20
2019-01-27
2019-02-03
Date