

ROC_Part2

September 29, 2021

1 Rate of Change (ROC) Part 2

<https://www.tradingtechnologies.com/xtrader-help/x-study/technical-indicator-definitions/rate-of-change-roc/>

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

import warnings
warnings.filterwarnings("ignore")

import yfinance as yf
yf.pdr_override()
```

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

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

# View Columns
df.head()
```

[*****100%*****] 1 of 1 completed

```
[2]:
```

	Adj Close	Close	High	Low	Open \
Date					
2016-01-04	98.446655	105.349998	105.370003	102.000000	102.610001
2016-01-05	95.979675	102.709999	105.849998	102.410004	105.750000
2016-01-06	94.101387	100.699997	102.370003	99.870003	100.559998
2016-01-07	90.129868	96.449997	100.129997	96.430000	98.680000
2016-01-08	90.606438	96.959999	99.110001	96.760002	98.550003

	Volume
Date	

```

2016-01-04  67649400
2016-01-05  55791000
2016-01-06  68457400
2016-01-07  81094400
2016-01-08  70798000

```

```

[3]: n = 12
df['ROC'] = ((df['Adj Close']/df['Adj Close'].shift(n))-1.0) * 100

```

```

[4]: df.tail()

```

```

[4]:
      Adj Close      Close      High      Low      Open \
Date
2018-12-24  144.656540  146.830002  151.550003  146.589996  148.149994
2018-12-26  154.843475  157.169998  157.229996  146.720001  148.300003
2018-12-27  153.838562  156.149994  156.770004  150.070007  155.839996
2018-12-28  153.917389  156.229996  158.520004  154.550003  157.500000
2018-12-31  155.405045  157.740005  159.360001  156.479996  158.529999

      Volume      ROC
Date
2018-12-24  37169200 -15.962676
2018-12-26  58582500  -6.718509
2018-12-27  53117100  -7.930431
2018-12-28  42291400  -7.353385
2018-12-31  35003500  -6.717913

```

```

[5]: 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')

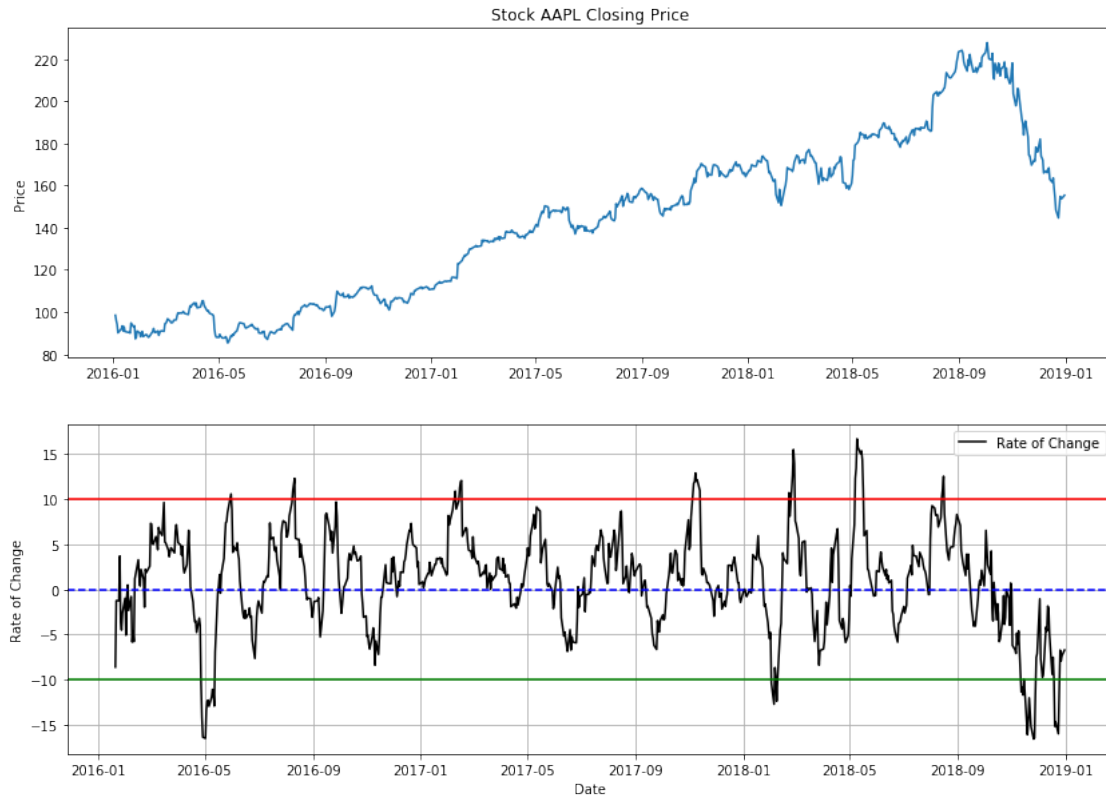
ax2 = plt.subplot(2, 1, 2)
ax2.plot(df['ROC'], label='Rate of Change', color='black')
ax2.axhline(y=0, color='blue', linestyle='--')
ax2.axhline(y=10, color='red')
ax2.axhline(y=-10, color='green')
ax2.grid()
ax2.set_ylabel('Rate of Change')
ax2.set_xlabel('Date')
ax2.legend(loc='best')

```

```

[5]: <matplotlib.legend.Legend at 0x1d157fe1b70>

```



1.1 Candlestick with (ROC)

```
[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  Adj Close  Close  High  Low  Open  \
0  735967.0  98.446655  105.349998  105.370003  102.000000  102.610001
1  735968.0  95.979675  102.709999  105.849998  102.410004  105.750000
2  735969.0  94.101387  100.699997  102.370003  99.870003  100.559998
3  735970.0  90.129868  96.449997  100.129997  96.430000  98.680000
4  735971.0  90.606438  96.959999  99.110001  96.760002  98.550003

      Volume  ROC  VolumePositive
```

0	67649400	NaN	False
1	55791000	NaN	False
2	68457400	NaN	False
3	81094400	NaN	False
4	70798000	NaN	False

```
[8]: 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['ROC'], label='Rate of Change', color='black')
ax2.axhline(y=0, color='blue', linestyle='--')
ax2.axhline(y=10, color='red')
ax2.axhline(y=-10, color='green')
ax2.grid()
ax2.set_ylabel('Rate of Change')
ax2.set_xlabel('Date')
ax2.legend(loc='best')
```

```
[8]: <matplotlib.legend.Legend at 0x1d15a2b7898>
```

