

# ROI

September 29, 2021

## 1 Return on Investment (ROI)

<https://www.investopedia.com/terms/r/returnoninvestment.asp>

[https://en.wikipedia.org/wiki/Return\\_on\\_investment](https://en.wikipedia.org/wiki/Return_on_investment)

```
[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	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]: df['ROI'] = (df['Adj Close'] - df['Adj Close'].shift(1)) / df['Adj Close'].
      ↪shift(1) * 100
```

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

```
[4]:
```

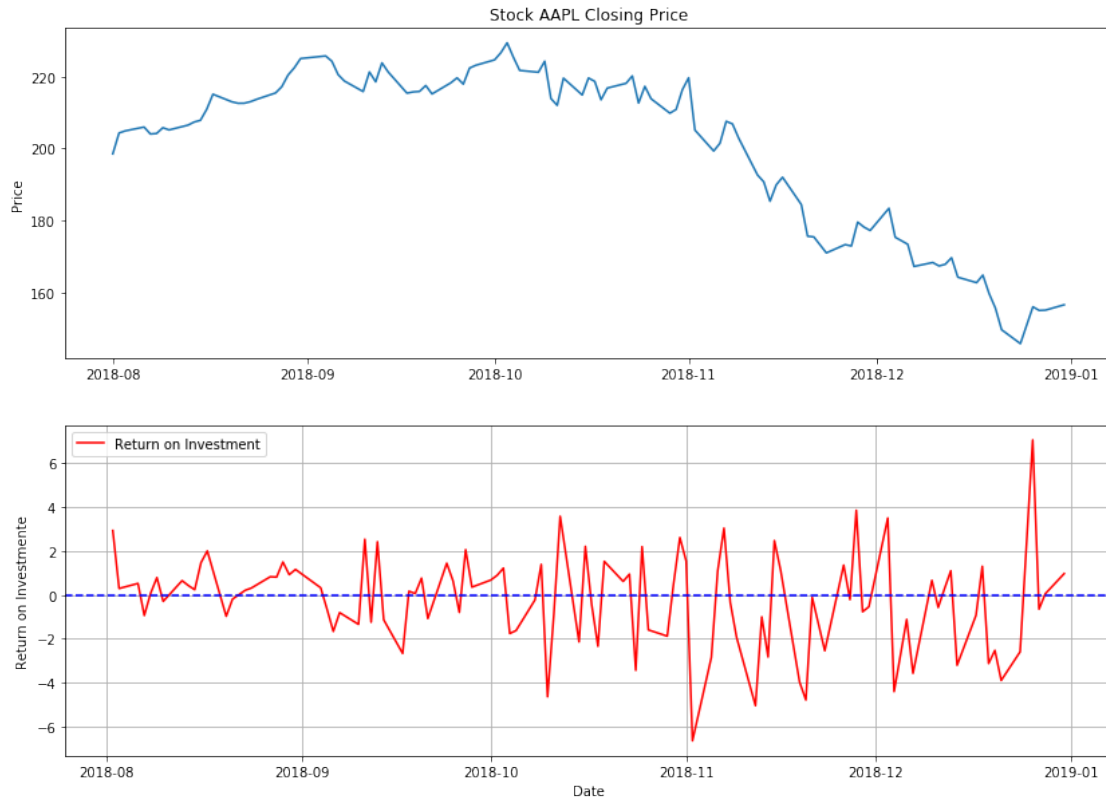
	Open	High	Low	Close	Adj Close \
Date					
2018-08-01	199.130005	201.759995	197.309998	201.500000	198.478760
2018-08-02	200.580002	208.380005	200.350006	207.389999	204.280457
2018-08-03	207.029999	208.740005	205.479996	207.990005	204.871445
2018-08-06	208.000000	209.250000	207.070007	209.070007	205.935257
2018-08-07	209.320007	209.500000	206.759995	207.110001	204.004639

	Volume	ROI
Date		
2018-08-01	67935700	NaN
2018-08-02	62404000	2.923082
2018-08-03	33447400	0.289302
2018-08-06	25425400	0.519258
2018-08-07	25587400	-0.937488

```
[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['ROI'], label='Return on Investment', color='red')
      ax2.axhline(y=0, color='blue', linestyle='--')
      ax2.grid()
      ax2.set_ylabel('Return on Investmente')
      ax2.set_xlabel('Date')
      ax2.legend(loc='best')
```

```
[5]: <matplotlib.legend.Legend at 0x284fc539710>
```



## 1.1 Candlestick with Return On Investment (ROI)

```
[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	\
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      ROI    VolumePositive

0	67935700	NaN	False
1	62404000	2.923082	True
2	33447400	0.289302	False
3	25425400	0.519258	False
4	25587400	-0.937488	False

```
[7]: 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['ROI'], label='Return on Investment', color='red')
ax2.axhline(y=0, color='blue', linestyle='--')
ax2.grid()
ax2.set_ylabel('Return on Investmente')
ax2.set_xlabel('Date')
ax2.legend(loc='best')
```

```
[7]: <matplotlib.legend.Legend at 0x284fc9cf7f0>
```

