

# Realised\_Volatility

September 29, 2021

## 1 Realised Volatility Indicator

<https://www.investopedia.com/terms/h/historicalvolatility.asp>

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

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	197.135651	
2018-08-02	200.580002	208.380005	200.350006	207.389999	202.898071	
2018-08-03	207.029999	208.740005	205.479996	207.990005	203.485062	
2018-08-06	208.000000	209.250000	207.070007	209.070007	204.541672	
2018-08-07	209.320007	209.500000	206.759995	207.110001	202.624130	

Volume

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

```
[3]: n = 20
rets = df['Adj Close'].pct_change().dropna()
std = rets.rolling(n).std()
```

```
[4]: historical_vol_annually = std*math.sqrt(252)
df['RV'] = 100*historical_vol_annually
```

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

```
[5]:
```

	Open	High	Low	Close	Adj Close	\
Date						
2018-08-01	199.130005	201.759995	197.309998	201.500000	197.135651	
2018-08-02	200.580002	208.380005	200.350006	207.389999	202.898071	
2018-08-03	207.029999	208.740005	205.479996	207.990005	203.485062	
2018-08-06	208.000000	209.250000	207.070007	209.070007	204.541672	
2018-08-07	209.320007	209.500000	206.759995	207.110001	202.624130	

```

      Volume  RV
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
```

```
[6]: df.tail()
```

```
[6]:
```

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

```

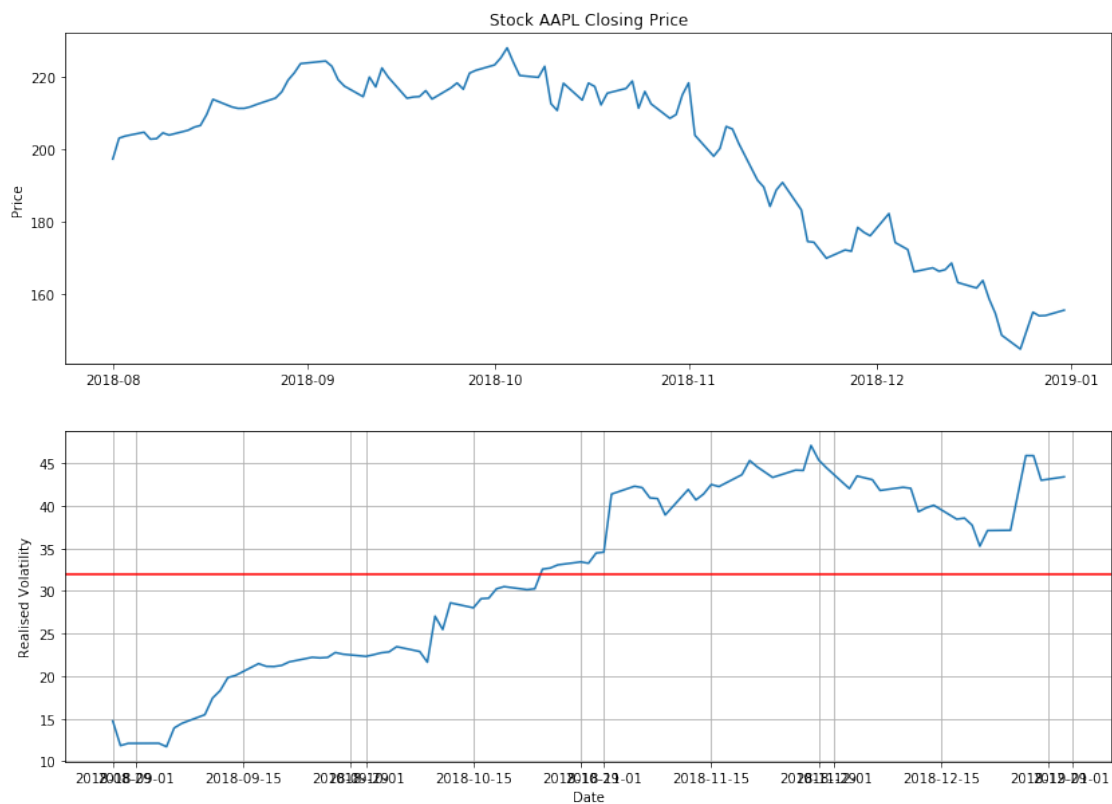
      Volume      RV
Date
2018-12-24  37169200  37.129966
2018-12-26  58582500  45.901041
2018-12-27  53117100  45.893158
```

2018-12-28 42291400 43.005838  
2018-12-31 35003500 43.406725

```
[7]: 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['RV'], label='Realised Volatility')
ax2.axhline(y=df['RV'].mean(), color='red')
ax2.grid()
ax2.set_ylabel('Realised Volatility')
ax2.set_xlabel('Date')
```

[7]: Text(0.5, 0, 'Date')



## 1.1 Candlestick with Historical Volatility

```
[8]: 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()
```

```
[8]:
```

	Date	Open	High	Low	Close	Adj Close	\
0	736907.0	199.130005	201.759995	197.309998	201.500000	197.135651	
1	736908.0	200.580002	208.380005	200.350006	207.389999	202.898071	
2	736909.0	207.029999	208.740005	205.479996	207.990005	203.485062	
3	736912.0	208.000000	209.250000	207.070007	209.070007	204.541672	
4	736913.0	209.320007	209.500000	206.759995	207.110001	202.624130	

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

```
[9]: 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['RV'], label='Realised Volatility')
```

```

ax2.axhline(y=df['RV'].mean(), color='red')
ax2.grid()
ax2.set_ylabel('Realised Volatility')
ax2.set_xlabel('Date')

```

[9]: Text(0.5, 0, 'Date')

