

Ulcer_Index

September 29, 2021

1 Ulcer Index

https://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:ulcer_index

```
[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 = '2017-01-01'
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						
2017-01-03	115.800003	116.330002	114.760002	116.150002	111.709831	
2017-01-04	115.849998	116.510002	115.750000	116.019997	111.584778	
2017-01-05	115.919998	116.860001	115.809998	116.610001	112.152229	
2017-01-06	116.779999	118.160004	116.470001	117.910004	113.402542	
2017-01-09	117.949997	119.430000	117.940002	118.989998	114.441246	

	Volume
Date	

```

2017-01-03  28781900
2017-01-04  21118100
2017-01-05  22193600
2017-01-06  31751900
2017-01-09  33561900

```

```

[3]: n = 14
df[str(n)+'-period Max Close'] = df['Adj Close'].rolling(n).max()
df['Percent_Drawdown'] = 100* (df['Adj Close'] - df[str(n)+'-period Max_
↳Close'])/df[str(n)+'-period Max Close']
df['Percent_Drawdown_Squared'] = df['Percent_Drawdown'] ** 2
df['Squared_Average'] = df['Percent_Drawdown_Squared'].rolling(n).sum()/14
df['Ulcer_Index'] = df['Squared_Average'].pow(0.5)
df = df.drop([str(n)+'-period Max_
↳Close', 'Percent_Drawdown', 'Percent_Drawdown_Squared', 'Squared_Average'],axis=1)

```

```

[4]: df.head(20)

```

```

[4]:
      Open      High      Low      Close  Adj Close  \
Date
2017-01-03  115.800003  116.330002  114.760002  116.150002  111.709831
2017-01-04  115.849998  116.510002  115.750000  116.019997  111.584778
2017-01-05  115.919998  116.860001  115.809998  116.610001  112.152229
2017-01-06  116.779999  118.160004  116.470001  117.910004  113.402542
2017-01-09  117.949997  119.430000  117.940002  118.989998  114.441246
2017-01-10  118.769997  119.379997  118.300003  119.110001  114.556656
2017-01-11  118.739998  119.930000  118.599998  119.750000  115.172195
2017-01-12  118.900002  119.300003  118.209999  119.250000  114.691307
2017-01-13  119.110001  119.620003  118.809998  119.040001  114.489334
2017-01-17  118.339996  120.239998  118.220001  120.000000  115.412643
2017-01-18  120.000000  120.500000  119.709999  119.989998  115.403008
2017-01-19  119.400002  120.089996  119.370003  119.779999  115.201050
2017-01-20  120.449997  120.449997  119.730003  120.000000  115.412643
2017-01-23  120.000000  120.809998  119.769997  120.080002  115.489578
2017-01-24  119.550003  120.099998  119.500000  119.970001  115.383789
2017-01-25  120.419998  122.099998  120.279999  121.879997  117.220779
2017-01-26  121.669998  122.440002  121.599998  121.940002  117.278488
2017-01-27  122.139999  122.349998  121.599998  121.949997  117.288101
2017-01-30  120.930000  121.629997  120.660004  121.629997  116.980324
2017-01-31  121.150002  121.389999  120.620003  121.349998  116.711029

      Volume  Ulcer_Index
Date
2017-01-03  28781900      NaN
2017-01-04  21118100      NaN
2017-01-05  22193600      NaN
2017-01-06  31751900      NaN

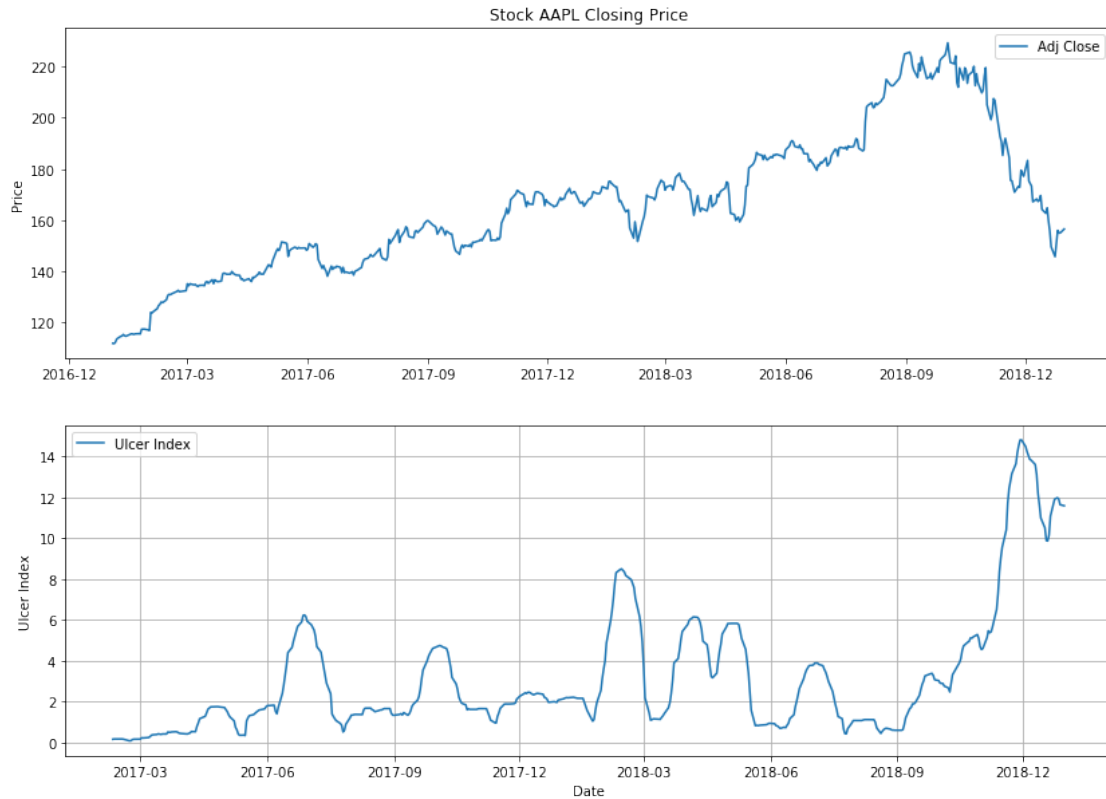
```

2017-01-09	33561900	NaN
2017-01-10	24462100	NaN
2017-01-11	27588600	NaN
2017-01-12	27086200	NaN
2017-01-13	26111900	NaN
2017-01-17	34439800	NaN
2017-01-18	23713000	NaN
2017-01-19	25597300	NaN
2017-01-20	32597900	NaN
2017-01-23	22050200	NaN
2017-01-24	23211000	NaN
2017-01-25	32377600	NaN
2017-01-26	26337600	NaN
2017-01-27	20562900	NaN
2017-01-30	30377500	NaN
2017-01-31	49201000	NaN

```
[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')
ax1.legend(loc='best')

ax2 = plt.subplot(2, 1, 2)
ax2.plot(df['Ulcer_Index'], label='Ulcer Index')
# ax2.axhline(y=0, color='red')
ax2.grid()
ax2.legend(loc='best')
ax2.set_ylabel('Ulcer Index')
ax2.set_xlabel('Date')
```

```
[5]: Text(0.5,0,'Date')
```



1.1 Candlestick with Ulcer Index

```
[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'] = mdates.date2num(dfc['Date'].astype(dt.date))
dfc.head()
```

```
[6]:      Date      Open      High      Low      Close  Adj Close  \
0  736332.0  115.800003  116.330002  114.760002  116.150002  111.709831
1  736333.0  115.849998  116.510002  115.750000  116.019997  111.584778
2  736334.0  115.919998  116.860001  115.809998  116.610001  112.152229
3  736335.0  116.779999  118.160004  116.470001  117.910004  113.402542
4  736338.0  117.949997  119.430000  117.940002  118.989998  114.441246

      Volume  Ulcer_Index  VolumePositive
0  28781900         NaN             False
```

1	21118100	NaN	False
2	22193600	NaN	False
3	31751900	NaN	False
4	33561900	NaN	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['Ulcer_Index'], label='Ulcer Index')
# ax2.axhline(y=0, color='red')
ax2.grid()
ax2.legend(loc='best')
ax2.set_ylabel('Ulcer Index')
ax2.set_xlabel('Date')
```

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

