

Stochastic_Full

September 29, 2021

1 Full Stochastic

https://www.investopedia.com/university/indicator_oscillator/ind_osc8.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 = '2018-08-01'
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	199.243088	
2018-08-02	200.580002	208.380005	200.350006	207.389999	205.067123	
2018-08-03	207.029999	208.740005	205.479996	207.990005	205.660416	
2018-08-06	208.000000	209.250000	207.070007	209.070007	206.728317	
2018-08-07	209.320007	209.500000	206.759995	207.110001	204.790268	

	Volume
Date	

```

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

```

```

[3]: n = 14 # number of days
      s = 3 # smoothing
      df['High_Highest'] = df['Adj Close'].rolling(n).max()
      df['Low_Lowest'] = df['Adj Close'].rolling(n).min()
      df['Fast_%K'] = (((df['Adj Close'] - df['Low_Lowest']) / (df['High_Highest'] -
      ↪df['Low_Lowest'])) * 100).rolling(3).mean()
      df['Full_%K'] = df['Fast_%K'].rolling(s).mean()
      df['Full_%D'] = df['Full_%K'].rolling(s).mean()

```

```

[4]: df.head(20)

```

```

[4]:
      Date      Open      High      Low      Close  Adj Close  \
2018-08-01  199.130005  201.759995  197.309998  201.500000  199.243088
2018-08-02  200.580002  208.380005  200.350006  207.389999  205.067123
2018-08-03  207.029999  208.740005  205.479996  207.990005  205.660416
2018-08-06  208.000000  209.250000  207.070007  209.070007  206.728317
2018-08-07  209.320007  209.500000  206.759995  207.110001  204.790268
2018-08-08  206.050003  207.809998  204.520004  207.250000  204.928696
2018-08-09  207.279999  209.779999  207.199997  208.880005  206.540436
2018-08-10  207.360001  209.100006  206.669998  207.529999  205.925232
2018-08-13  207.699997  210.949997  207.699997  208.869995  207.254883
2018-08-14  210.160004  210.559998  208.259995  209.750000  208.128067
2018-08-15  209.220001  210.740005  208.330002  210.240005  208.614273
2018-08-16  211.750000  213.809998  211.470001  213.320007  211.670471
2018-08-17  213.440002  217.949997  213.160004  217.580002  215.897522
2018-08-20  218.100006  219.179993  215.110001  215.460007  213.793930
2018-08-21  216.800003  217.190002  214.029999  215.039993  213.377167
2018-08-22  214.100006  216.360001  213.839996  215.050003  213.387085
2018-08-23  214.649994  217.050003  214.600006  215.490005  213.823685
2018-08-24  216.600006  216.899994  215.110001  216.160004  214.488495
2018-08-27  217.149994  218.740005  216.330002  217.940002  216.254745
2018-08-28  219.009995  220.539993  218.919998  219.699997  218.001129

      Volume  High_Highest  Low_Lowest  Fast_%K  Full_%K  \
Date
2018-08-01  67935700      NaN      NaN      NaN      NaN
2018-08-02  62404000      NaN      NaN      NaN      NaN
2018-08-03  33447400      NaN      NaN      NaN      NaN
2018-08-06  25425400      NaN      NaN      NaN      NaN
2018-08-07  25587400      NaN      NaN      NaN      NaN

```

2018-08-08	22525500	NaN	NaN	NaN	NaN
2018-08-09	23469200	NaN	NaN	NaN	NaN
2018-08-10	24611200	NaN	NaN	NaN	NaN
2018-08-13	25869100	NaN	NaN	NaN	NaN
2018-08-14	20748000	NaN	NaN	NaN	NaN
2018-08-15	28807600	NaN	NaN	NaN	NaN
2018-08-16	28500400	NaN	NaN	NaN	NaN
2018-08-17	35427000	NaN	NaN	NaN	NaN
2018-08-20	30287700	215.897522	199.243088	NaN	NaN
2018-08-21	26159800	215.897522	204.790268	NaN	NaN
2018-08-22	19018100	215.897522	204.790268	80.692109	NaN
2018-08-23	18883200	215.897522	204.790268	78.678712	NaN
2018-08-24	18476400	215.897522	204.790268	82.013853	80.461558
2018-08-27	20525100	216.254745	204.928696	89.547780	83.413448
2018-08-28	22776800	218.001129	205.925232	95.771451	89.111028

	Full_%D
Date	
2018-08-01	NaN
2018-08-02	NaN
2018-08-03	NaN
2018-08-06	NaN
2018-08-07	NaN
2018-08-08	NaN
2018-08-09	NaN
2018-08-10	NaN
2018-08-13	NaN
2018-08-14	NaN
2018-08-15	NaN
2018-08-16	NaN
2018-08-17	NaN
2018-08-20	NaN
2018-08-21	NaN
2018-08-22	NaN
2018-08-23	NaN
2018-08-24	NaN
2018-08-27	NaN
2018-08-28	84.328678

```
[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')

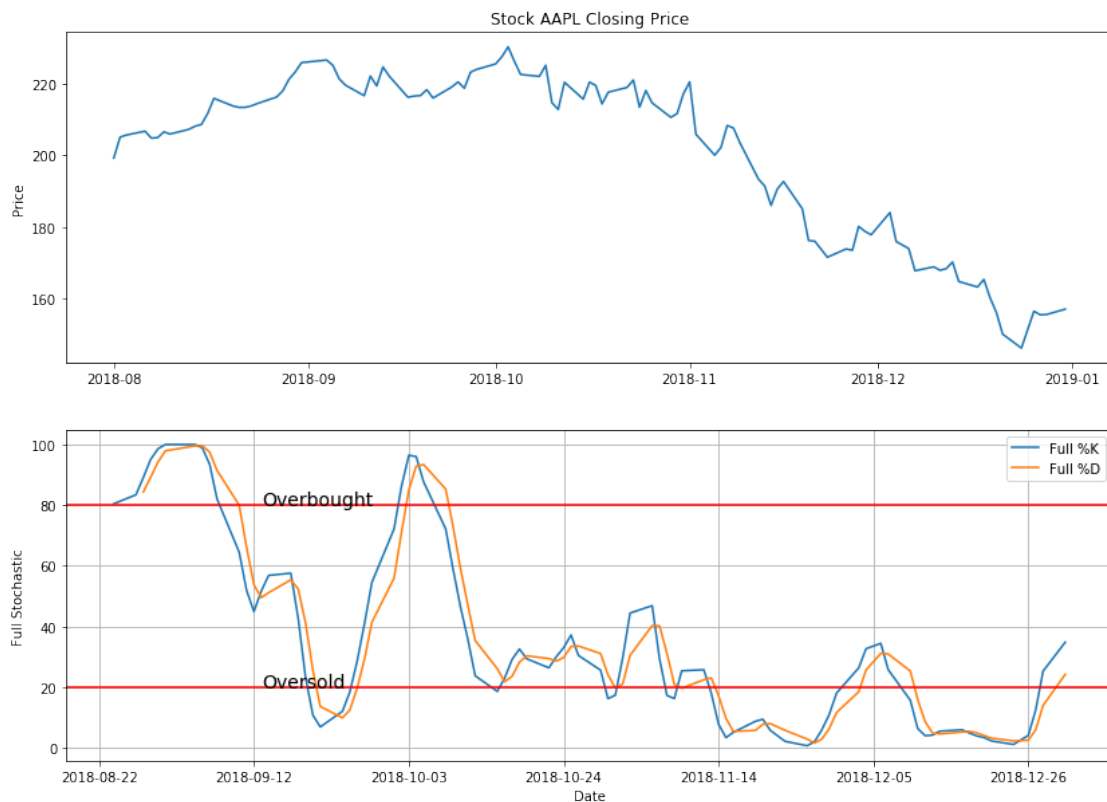
ax2 = plt.subplot(2, 1, 2)
ax2.plot(df['Full_%K'], label='Full %K')
```

```

ax2.plot(df['Full_%D'], label='Full %D')
ax2.text(s='Overbought', x=df.index[30], y=80, fontsize=14)
ax2.text(s='Oversold', x=df.index[30], y=20, fontsize=14)
ax2.axhline(y=80, color='red')
ax2.axhline(y=20, color='red')
ax2.grid()
ax2.set_ylabel('Full Stochastic')
ax2.legend(loc='best')
ax2.set_xlabel('Date')

```

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



1.1 Candlestick with Full Stochastic

```

[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	736907.0	199.130005	201.759995	197.309998	201.500000	199.243088	
1	736908.0	200.580002	208.380005	200.350006	207.389999	205.067123	
2	736909.0	207.029999	208.740005	205.479996	207.990005	205.660416	
3	736912.0	208.000000	209.250000	207.070007	209.070007	206.728317	
4	736913.0	209.320007	209.500000	206.759995	207.110001	204.790268	

	Volume	High_Highest	Low_Lowest	Fast_%K	Full_%K	Full_%D	\
0	67935700	NaN	NaN	NaN	NaN	NaN	
1	62404000	NaN	NaN	NaN	NaN	NaN	
2	33447400	NaN	NaN	NaN	NaN	NaN	
3	25425400	NaN	NaN	NaN	NaN	NaN	
4	25587400	NaN	NaN	NaN	NaN	NaN	

	VolumePositive
0	True
1	True
2	False
3	False
4	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*dfc.Volume.max())
ax1.set_title('Stock ' + symbol + ' Closing Price')
ax1.set_ylabel('Price')

ax2 = plt.subplot(2, 1, 2)
ax2.plot(df['Full_%K'], label='Full %K')
ax2.plot(df['Full_%D'], label='Full %D')
ax2.text(s='Overbought', x=df.index[30], y=80, fontsize=14)
ax2.text(s='Oversold', x=df.index[30], y=20, fontsize=14)
```

```

ax2.axhline(y=80, color='red')
ax2.axhline(y=20, color='red')
ax2.grid()
ax2.set_ylabel('Full Stochastic')
ax2.legend(loc='best')
ax2.set_xlabel('Date')

```

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

