

Stochastic_Fast

September 29, 2021

1 Fast Stochastic

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

```
[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
     df['Fast_%D'] = df['Fast_%K'].rolling(s).mean()

```

```

[4]: df.head()

```

```

[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

      Volume  High_Highest  Low_Lowest  Fast_%K  Fast_%D
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

```

```

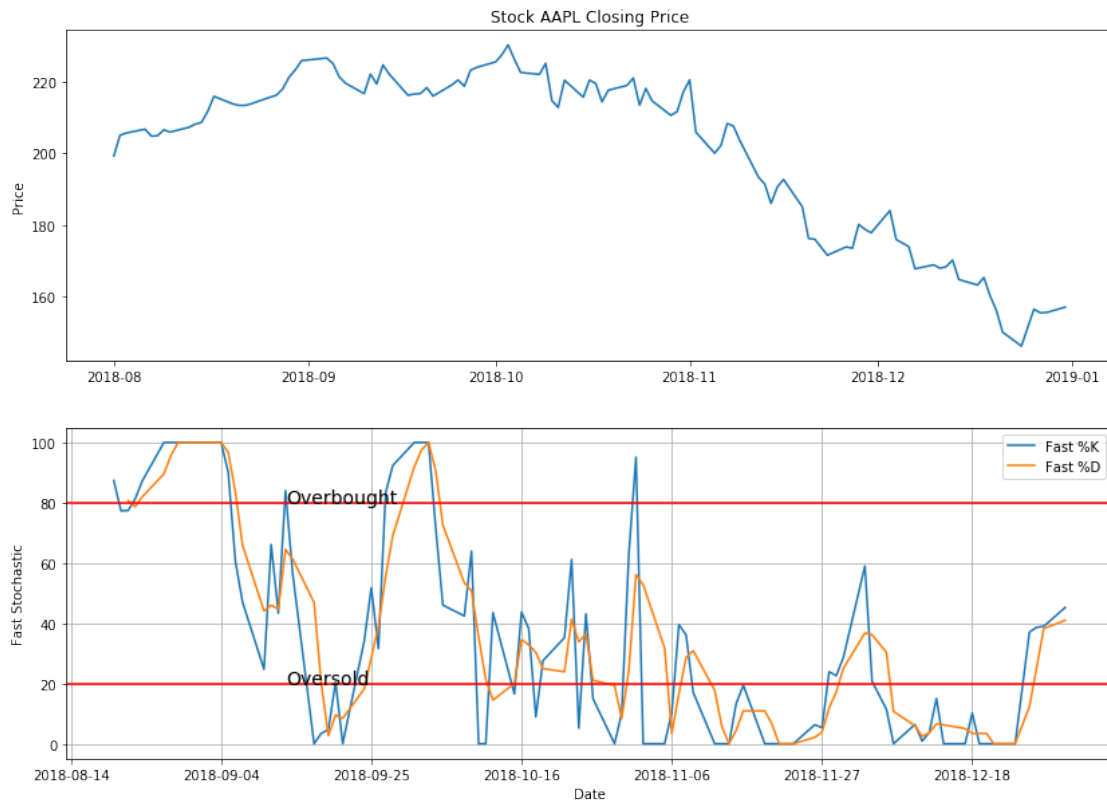
[5]: fig = plt.figure(figsize=(14,10))
     ax1 = plt.subplot(2, 1, 1)
     ax1.plot(df['Adj Close'])
     ax1.set_title('Stock ' + symbol + ' Closing Price')
     ax1.set_ylabel('Price')

     ax2 = plt.subplot(2, 1, 2)
     ax2.plot(df['Fast_%K'], label='Fast %K')
     ax2.plot(df['Fast_%D'], label='Fast %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('Fast Stochastic')

```

```
ax2.legend(loc='best')
ax2.set_xlabel('Date')
```

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



1.1 Candlestick with Fast 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	Fast_%D	VolumePositive
0	67935700	NaN	NaN	NaN	NaN	True
1	62404000	NaN	NaN	NaN	NaN	True
2	33447400	NaN	NaN	NaN	NaN	False
3	25425400	NaN	NaN	NaN	NaN	False
4	25587400	NaN	NaN	NaN	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['Fast_%K'], label='Fast %K')
ax2.plot(df['Fast_%D'], label='Fast %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('Fast Stochastic')
ax2.legend(loc='best')
ax2.set_xlabel('Date')

```

```

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

```

