

Balance_of_Power

September 29, 2021

1 Balance of Power

<http://www.binarytribune.com/forex-trading-indicators/balance-of-power>

```
[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

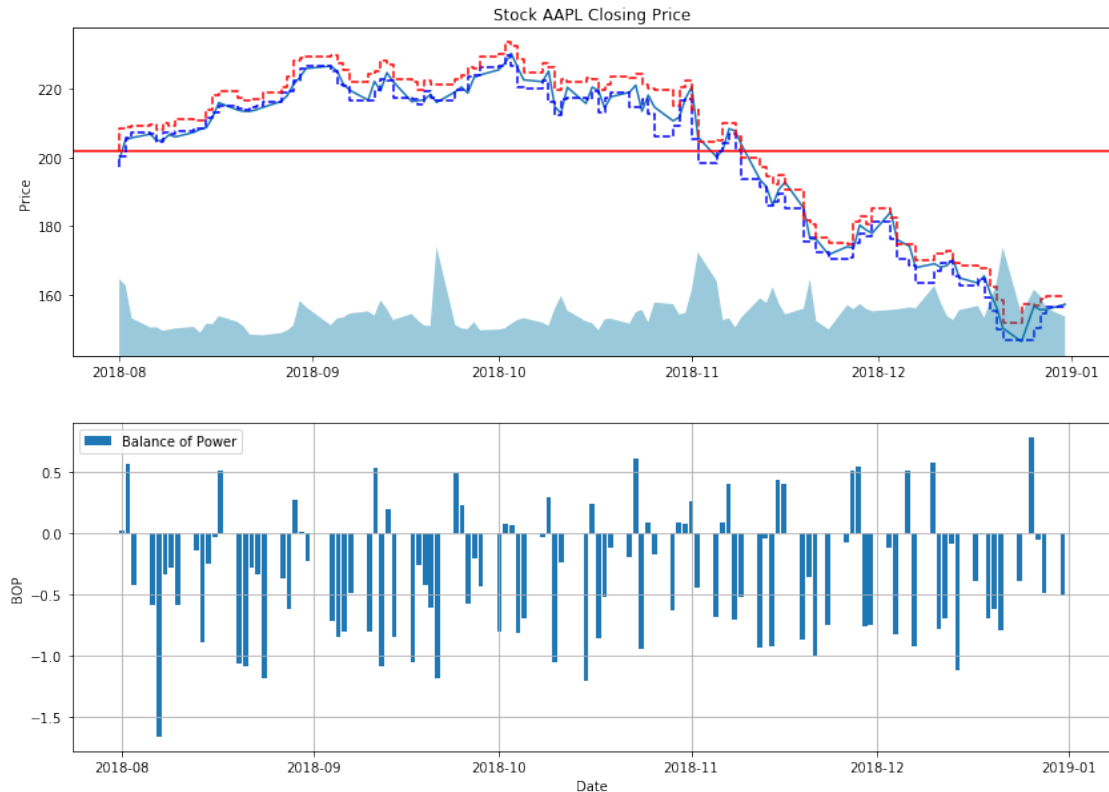
```

```
[3]: df['BOP'] = (df['Adj Close'] - df['Open']) / (df['High'] - df['Low'])
```

```
[4]: # Line Chart
fig = plt.figure(figsize=(14,10))
ax1 = plt.subplot(2, 1, 1)
ax1.plot(df.index, df['Adj Close'])
ax1.axhline(y=df['Adj Close'].mean(),color='r')
ax1.step(df.index, df['Low'], c='blue', linestyle='--')
ax1.step(df.index, df['High'], c='red', linestyle='--')
ax1v = ax1.twinx()
ax1v.fill_between(df.index[0:],0, df.Volume[0:], facecolor='#0079a3', 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.bar(df.index, df['BOP'], label='Balance of Power')
ax2.grid()
ax2.set_ylabel('BOP')
ax2.set_xlabel('Date')
ax2.legend(loc='best')
```

```
[4]: <matplotlib.legend.Legend at 0x25084ff7668>
```



1.1 Candlestick with BOP

```
[5]: from matplotlib import dates as mdates
import datetime as dt

dfc = df.copy()
dfc['BOP'] = (dfc['Adj Close'] - dfc['Open']) / (dfc['High'] - dfc['Low'])
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()
```

```
[5]:
```

	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	BOP	VolumePositive
--	--------	-----	----------------

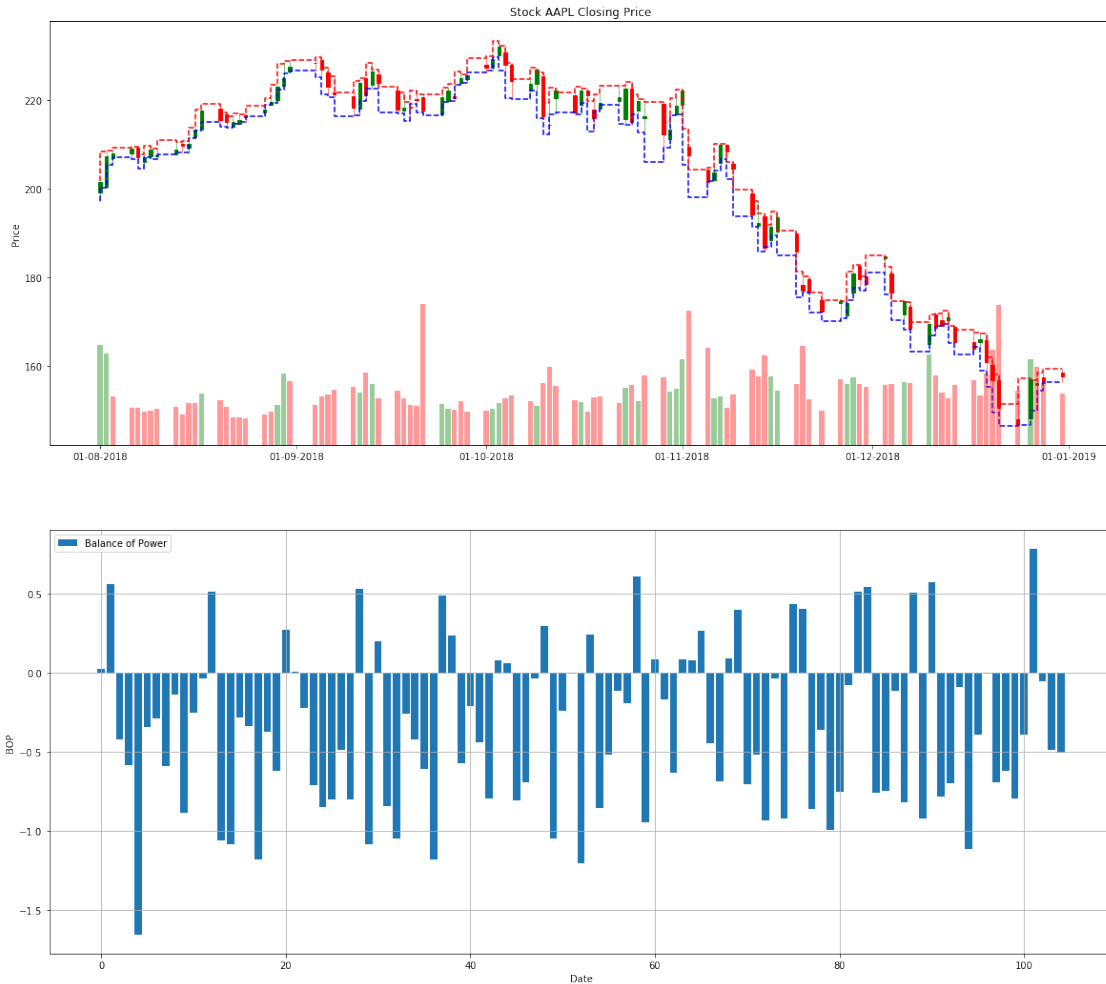
0	67935700	0.025412	True
1	62404000	0.558795	True
2	33447400	-0.420116	False
3	25425400	-0.583343	False
4	25587400	-1.653186	False

```
[6]: from mpl_finance import candlestick_ohlc

fig = plt.figure(figsize=(20,18))
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.axhline(y=dfc['Adj Close'].mean(),color='r')
ax1.step(dfc.Date, dfc['Low'], c='blue', linestyle='--')
ax1.step(dfc.Date, dfc['High'], c='red', linestyle='--')
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.bar(dfc.index, dfc['BOP'], label='Balance of Power')
ax2.grid()
ax2.set_ylabel('BOP')
ax2.set_xlabel('Date')
ax2.legend(loc='best')
```

```
[6]: <matplotlib.legend.Legend at 0x2508534ae48>
```



```
[7]: import matplotlib.patches as mpatches

fig = plt.figure(figsize=(20,18))
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.axhline(y=dfc['Adj Close'].mean(),color='r')
ax1.step(dfc.Date, dfc['Low'], c='blue', linestyle='--')
ax1.step(dfc.Date, dfc['High'], c='red', linestyle='--')
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)
status = []
for i in dfc['BOP']:
    if i >= 0:
        status.append(True) # Increase
    else:
        status.append(False) # Decrease
dfc['Status'] = status
#dfc['Positive'] = dfc['BOP'] > 0
colors2 = dfc.Status.map({True: 'g', False: 'r'})
Increase = mpatches.Patch(color='g', label='Increase')
Decrease = mpatches.Patch(color='r', label='Decrease')
ax2.bar(dfc.Date, dfc['BOP'], color = colors2)
ax2.grid()
ax2.set_ylabel('BOP')
ax2.set_xlabel('Date')
ax2.legend(handles=[Increase,Decrease])

```

[7]: <matplotlib.legend.Legend at 0x250858b39b0>

