

# Acceleration\_Bands

September 29, 2021

## 1 Acceleration Bands (ABANDS)

<https://www.tradingtechnologies.com/xtrader-help/x-study/technical-indicator-definitions/acceleration-bands-abands/>

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

import warnings
warnings.filterwarnings("ignore")

import yfinance as yf
yf.pdr_override()
```

```
[2]: # input
symbol = 'AAPL'
start = '2018-01-01'
end = '2019-01-01'

# Read data
df = yf.download(symbol,start,end)

# View Columns
df.head()
```

[\*\*\*\*\*100%\*\*\*\*\*] 1 of 1 completed

```
[2]:
```

	Adj Close	Close	High	Low	Open \
Date					
2018-01-02	167.199890	172.259995	172.300003	169.259995	170.160004
2018-01-03	167.170776	172.229996	174.550003	171.960007	172.529999
2018-01-04	167.947266	173.029999	173.470001	172.080002	172.539993
2018-01-05	169.859406	175.000000	175.369995	173.050003	173.440002
2018-01-08	169.228500	174.350006	175.610001	173.929993	174.350006

	Volume
Date	

```

2018-01-02  25555900
2018-01-03  29517900
2018-01-04  22434600
2018-01-05  23660000
2018-01-08  20567800

```

```

[3]: n = 7
      UBB = df['High'] * ( 1 + 4 * (df['High'] - df['Low']) / (df['High'] +
      ↪df['Low']))
      df['Upper_Band'] = UBB.rolling(n, center=False).mean()
      df['Middle_Band'] = df['Adj Close'].rolling(n).mean()
      LBB = df['Low'] * ( 1 - 4 * (df['High'] - df['Low']) / (df['High'] + df['Low']))
      df['Lower_Band'] = LBB.rolling(n, center=False).mean()

```

```

[4]: df.head(20)

```

```

[4]:      Adj Close      Close      High      Low      Open \
Date
2018-01-02  167.199890  172.259995  172.300003  169.259995  170.160004
2018-01-03  167.170776  172.229996  174.550003  171.960007  172.529999
2018-01-04  167.947266  173.029999  173.470001  172.080002  172.539993
2018-01-05  169.859406  175.000000  175.369995  173.050003  173.440002
2018-01-08  169.228500  174.350006  175.610001  173.929993  174.350006
2018-01-09  169.209091  174.330002  175.059998  173.410004  174.550003
2018-01-10  169.170258  174.289993  174.300003  173.000000  173.160004
2018-01-11  170.131180  175.279999  175.490005  174.490005  174.589996
2018-01-12  171.888031  177.089996  177.360001  175.649994  176.179993
2018-01-16  171.014465  176.190002  179.389999  176.139999  177.899994
2018-01-17  173.839005  179.100006  179.250000  175.070007  176.149994
2018-01-18  173.994293  179.259995  180.100006  178.250000  179.369995
2018-01-19  173.217789  178.460007  179.580002  177.410004  178.610001
2018-01-22  171.800674  177.000000  177.779999  176.600006  177.300003
2018-01-23  171.839493  177.039993  179.440002  176.820007  177.300003
2018-01-24  169.102325  174.220001  177.300003  173.199997  177.250000
2018-01-25  166.083679  171.110001  174.949997  170.529999  174.509995
2018-01-26  166.471924  171.509995  172.000000  170.059998  172.000000
2018-01-29  163.026215  167.960007  170.160004  167.070007  170.160004
2018-01-30  162.065277  166.970001  167.369995  164.699997  165.529999

```

```

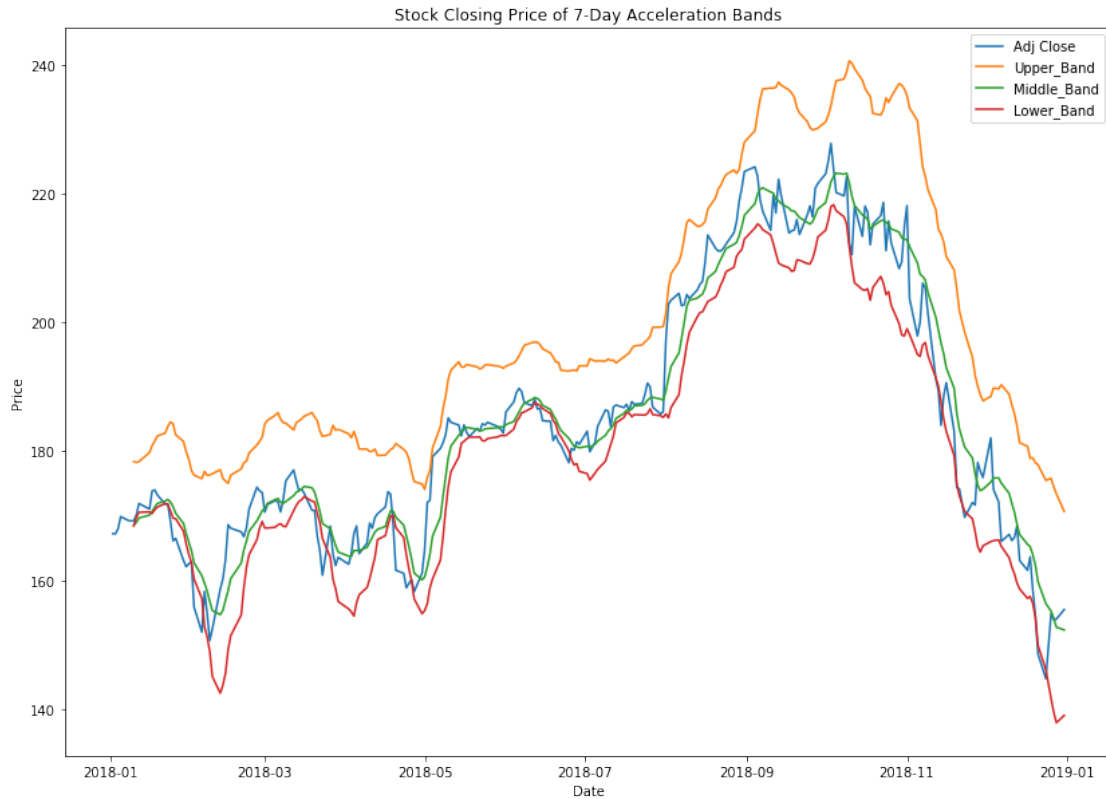
      Volume  Upper_Band  Middle_Band  Lower_Band
Date
2018-01-02  25555900      NaN      NaN      NaN
2018-01-03  29517900      NaN      NaN      NaN
2018-01-04  22434600      NaN      NaN      NaN
2018-01-05  23660000      NaN      NaN      NaN
2018-01-08  20567800      NaN      NaN      NaN
2018-01-09  21584000      NaN      NaN      NaN

```

2018-01-10	23959900	178.396632	168.540741	168.418060
2018-01-11	18667700	178.262573	168.959497	169.741149
2018-01-12	25418100	178.409411	169.633390	170.516552
2018-01-16	29565900	179.793445	170.071562	170.572013
2018-01-17	34386800	180.888835	170.640076	170.338832
2018-01-18	31193400	181.579257	171.320903	170.907827
2018-01-19	32425100	182.375081	171.893574	171.332219
2018-01-22	27108600	182.837667	172.269348	171.880527
2018-01-23	32689100	183.869497	172.513393	171.755217
2018-01-24	51105100	184.555119	172.115435	170.733697
2018-01-25	41529000	184.262787	171.411037	169.605652
2018-01-26	39143000	182.576129	170.358597	169.518987
2018-01-29	50640400	181.515773	168.791728	167.572923
2018-01-30	46048200	179.916708	167.198512	165.616716

```
[5]: plt.figure(figsize=(14,10))
plt.plot(df['Adj Close'])
plt.plot(df['Upper_Band'])
plt.plot(df['Middle_Band'])
plt.plot(df['Lower_Band'])
plt.ylabel('Price')
plt.xlabel('Date')
plt.title('Stock Closing Price of ' + str(n) + '-Day Acceleration Bands')
plt.legend(loc='best')
```

```
[5]: <matplotlib.legend.Legend at 0x1f12faeb6a0>
```



```
[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'] = pd.to_datetime(dfc['Date'])
dfc['Date'] = dfc['Date'].apply(mdates.date2num)
dfc.head()
```

```
[6]:      Date  Adj Close  Close  High  Low  Open \
0  736696.0  167.199890  172.259995  172.300003  169.259995  170.160004
1  736697.0  167.170776  172.229996  174.550003  171.960007  172.529999
2  736698.0  167.947266  173.029999  173.470001  172.080002  172.539993
3  736699.0  169.859406  175.000000  175.369995  173.050003  173.440002
4  736702.0  169.228500  174.350006  175.610001  173.929993  174.350006

      Volume  Upper_Band  Middle_Band  Lower_Band  VolumePositive
0  25555900         NaN         NaN         NaN             False
1  29517900         NaN         NaN         NaN             False
2  22434600         NaN         NaN         NaN             False
```

3	23660000	NaN	NaN	NaN	False
4	20567800	NaN	NaN	NaN	False

```
[7]: from mpl_finance import candlestick_ohlc

fig = plt.figure(figsize=(18,14))
ax1 = plt.subplot(2, 1, 1)
candlestick_ohlc(ax1,dfc.values, width=0.5, colorup='g', colordown='r', alpha=1.
    ↪0)
ax1.plot(df['Upper_Band'], label='Upper Band')
ax1.plot(df['Middle_Band'], label='Middle Band')
ax1.plot(df['Lower_Band'], label='Lower Band')
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.legend(loc='best')
ax1.set_ylabel('Price')
ax1.set_xlabel('Date')
```

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

