

# Accumulation\_Distribution

September 29, 2021

## 1 Accumulation/Distribution Line

<https://www.investopedia.com/terms/a/accumulationdistribution.asp>

```
[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 = '2018-09-01'
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					
2018-09-04	223.062759	228.360001	229.179993	226.630005	228.410004
2018-09-05	221.607346	226.869995	229.669998	225.100006	228.990005
2018-09-06	217.924789	223.100006	227.350006	221.300003	226.229996
2018-09-07	216.166550	221.300003	225.369995	220.710007	221.850006
2018-09-10	213.265411	218.330002	221.850006	216.470001	220.949997

	Volume
Date	
2018-09-04	27390100

```

2018-09-05 33333000
2018-09-06 34290000
2018-09-07 37619800
2018-09-10 39516500

```

```

[16]: def accumulation_distribution_line(df, n):
        adl = 0
        money_flow_multiplier = ((df['Adj Close'] - df['Low'].shift(n)) -
        ↪(df['High'].shift(n) - df['Adj Close'])) / (df['High'].shift(n) - df['Low'].
        ↪shift(n))
        money_flow_volume = money_flow_multiplier * df['Volume'].shift(n)
        adl = adl + money_flow_volume
        return adl

```

```

[17]: df['ADL'] = accumulation_distribution_line(df, 14)

```

```

[19]: df.head(20)

```

```

[19]:      Adj Close    Close    High    Low    Open  \
Date
2018-09-04  223.062759  228.360001  229.179993  226.630005  228.410004
2018-09-05  221.607346  226.869995  229.669998  225.100006  228.990005
2018-09-06  217.924789  223.100006  227.350006  221.300003  226.229996
2018-09-07  216.166550  221.300003  225.369995  220.710007  221.850006
2018-09-10  213.265411  218.330002  221.850006  216.470001  220.949997
2018-09-11  218.657410  223.850006  224.300003  216.559998  218.009995
2018-09-12  215.941879  221.070007  225.000000  219.839996  224.940002
2018-09-13  221.157990  226.410004  228.350006  222.570007  223.520004
2018-09-14  218.647614  223.839996  226.839996  222.520004  225.750000
2018-09-17  212.825882  217.880005  222.949997  217.270004  222.149994
2018-09-18  213.177521  218.240005  221.850006  217.119995  217.789993
2018-09-19  213.304474  218.369995  219.619995  215.300003  218.500000
2018-09-20  214.926010  220.029999  222.279999  219.149994  220.240005
2018-09-21  212.610977  217.660004  221.360001  217.289993  220.779999
2018-09-24  215.668365  220.789993  221.259995  216.630005  216.820007
2018-09-25  217.035889  222.190002  222.820007  219.699997  219.750000
2018-09-26  215.306931  220.419998  223.750000  219.759995  221.000000
2018-09-27  219.731857  224.949997  226.440002  223.539993  223.820007
2018-09-28  220.503540  225.740005  225.839996  224.020004  224.789993
2018-10-01  221.988266  227.259995  229.419998  226.350006  227.949997

```

```

      Volume    ADL
Date
2018-09-04  27390100    NaN
2018-09-05  33333000    NaN
2018-09-06  34290000    NaN
2018-09-07  37619800    NaN

```

2018-09-10	39516500	NaN
2018-09-11	35749000	NaN
2018-09-12	49278700	NaN
2018-09-13	41706400	NaN
2018-09-14	31999300	NaN
2018-09-17	37195100	NaN
2018-09-18	31571700	NaN
2018-09-19	27123800	NaN
2018-09-20	26608800	NaN
2018-09-21	96246700	NaN
2018-09-24	27693400	-2.628739e+08
2018-09-25	24554400	-1.509705e+08
2018-09-26	23984700	-1.022247e+08
2018-09-27	30181200	-5.341288e+07
2018-09-28	22929400	1.973673e+07
2018-10-01	23600800	1.439442e+07

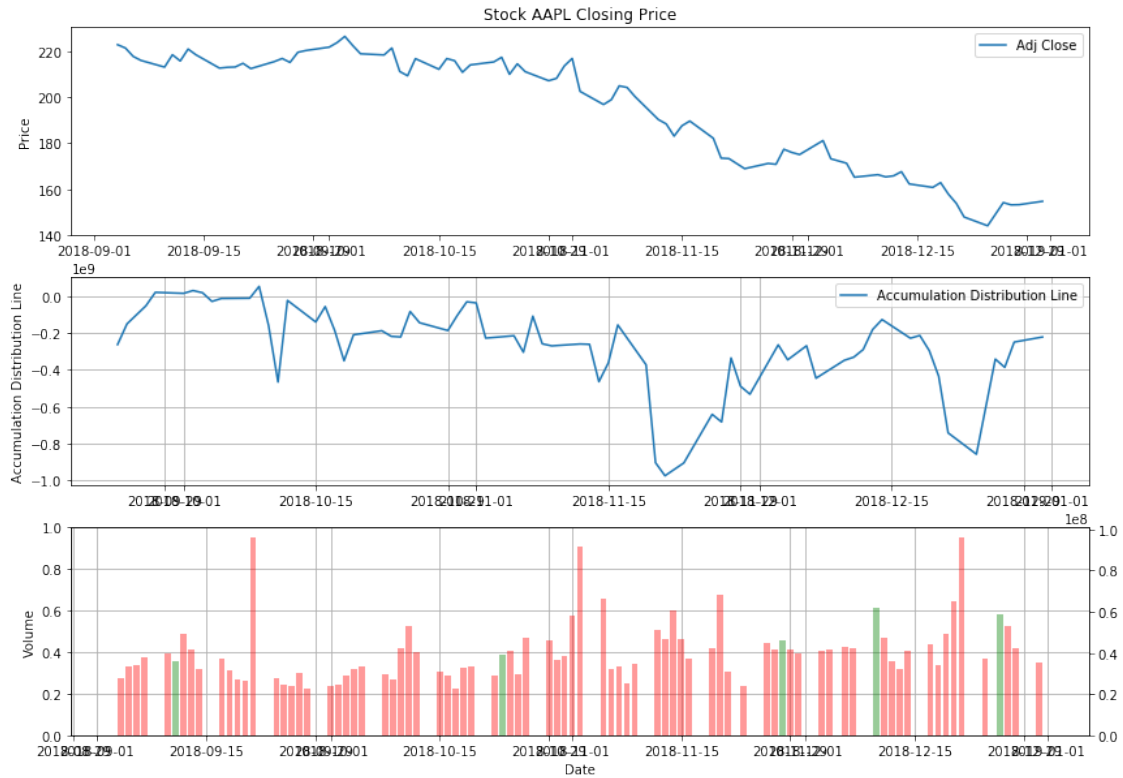
```
[20]: df['VolumePositive'] = df['Open'] < df['Adj Close']
```

```
[21]: fig = plt.figure(figsize=(14,10))
ax1 = plt.subplot(3, 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(3, 1, 2)
ax2.plot(df['ADL'], label='Accumulation Distribution Line')
ax2.grid()
ax2.legend(loc='best')
ax2.set_ylabel('Accumulation Distribution Line')

ax3 = plt.subplot(3, 1, 3)
ax3v = ax3.twinx()
colors = df.VolumePositive.map({True: 'g', False: 'r'})
ax3v.bar(df.index, df['Volume'], color=colors, alpha=0.4)
ax3.set_ylabel('Volume')
ax3.grid()
ax3.set_xlabel('Date')
```

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



## 1.1 Candlestick with Accumulation/Distribution Line

```
[22]: 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()
```

```
[22]:
```

	Date	Adj Close	Close	High	Low	Open \
0	736941.0	223.062759	228.360001	229.179993	226.630005	228.410004
1	736942.0	221.607346	226.869995	229.669998	225.100006	228.990005
2	736943.0	217.924789	223.100006	227.350006	221.300003	226.229996
3	736944.0	216.166550	221.300003	225.369995	220.710007	221.850006
4	736947.0	213.265411	218.330002	221.850006	216.470001	220.949997

```

      Volume  ADL  VolumePositive
0  27390100  NaN              False
```

1	33333000	NaN	False
2	34290000	NaN	False
3	37619800	NaN	False
4	39516500	NaN	False

```
[23]: from mpl_finance import candlestick_ohlc

fig = plt.figure(figsize=(14,10))
ax1 = plt.subplot(3, 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(3, 1, 2)
ax2.plot(df['ADL'], label='Accumulation Distribution Line')
ax2.grid()
ax2.legend(loc='best')
ax2.set_ylabel('Accumulation Distribution Line')

ax3 = plt.subplot(3, 1, 3)
ax3v = ax3.twinx()
colors = df.VolumePositive.map({True: 'g', False: 'r'})
ax3v.bar(df.index, df['Volume'], color=colors, alpha=0.4)
ax3.set_ylabel('Volume')
ax3.grid()
ax3.set_xlabel('Date')
```

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

