KAMA

September 29, 2021

1 Kaufman's Adaptive Moving Average (KAMA)

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

[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 = '2017-01-01'
end = '2019-01-01'

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

# View Columns
df.head()
```

```
[**********************100%**********************]  1 of 1 downloaded
```

[2]:

Date	Open	High	Low	Close	Adj Close	Volume
2017-01-03	115.800003	116.330002	114.760002	116.150002	112.140007	
2017-01-04	115.849998	116.510002	115.750000	116.019997	112.014503	
2017-01-05	115.919998	116.860001	115.809998	116.610001	112.584129	
2017-01-06	116.779999	118.160004	116.470001	117.910004	113.839249	
2017-01-09	117.949997	119.430000	117.940002	118.989998	114.881950	

```
2017-01-03  28781900
2017-01-04  21118100
2017-01-05  22193600
2017-01-06  31751900
2017-01-09  33561900
```

[3]:
```
n = 10
df['Change'] = abs(df['Adj Close'] - df['Adj Close'].shift(10))
df['Volatility'] = abs(df['Adj Close'] - df['Adj Close'].shift()).rolling(n).
 ↪sum()
df['ER'] = df['Change']/df['Volatility']
df['SC'] = np.square(df['ER']*(2.0/(2+1)-2.0/(30+1))+2.0/(30+1))
df['KAMA'] = df['Adj Close'].rolling(n).mean()
df['KAMA'][:n]= np.nan
```

[4]:
```
i = 1
while i<len(df['KAMA'][n+1:]):
        s = df['KAMA']
        s.iloc[n+i] = df['KAMA'][n+i-1] + df['SC'][n+i]*(df['Adj Close'][n+i] -␣
 ↪df['KAMA'][n+i-1])
        df['KAMA'] = s
        i = i + 1
df = df.drop(['Change','Volatility','ER','SC'],axis=1)
```

[5]:
```
df.head(20)
```

[5]:

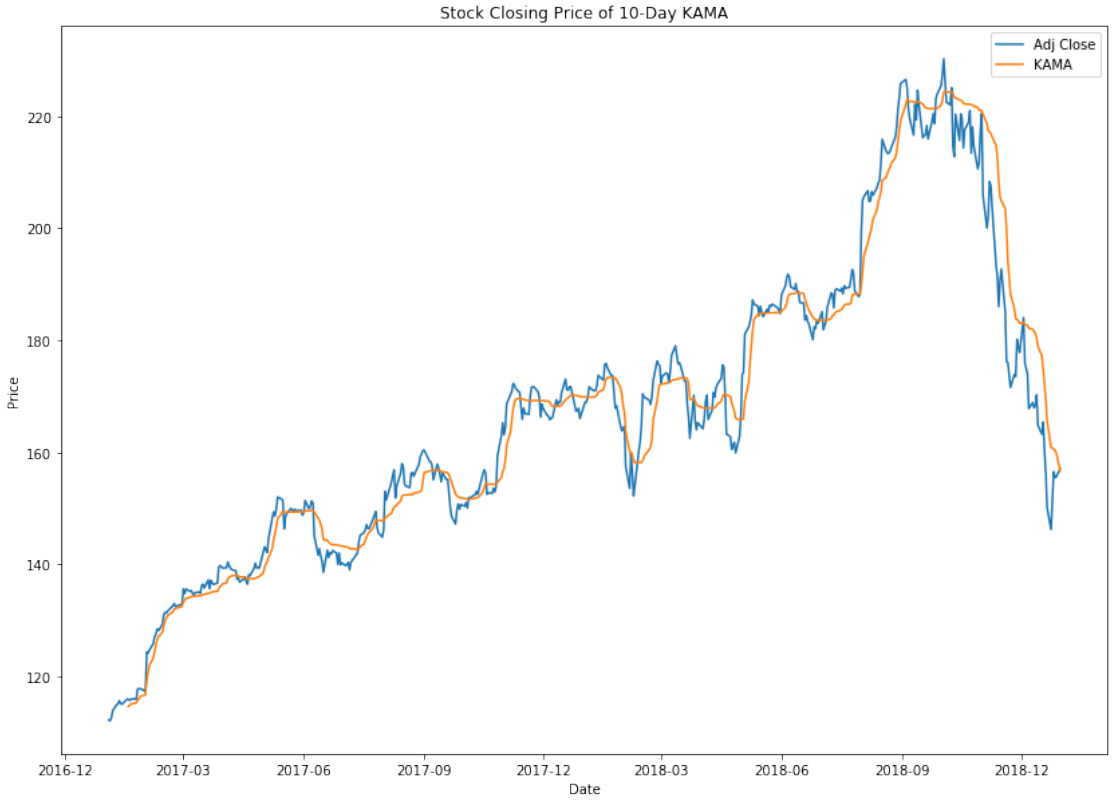
Date	Open	High	Low	Close	Adj Close
2017-01-03	115.800003	116.330002	114.760002	116.150002	112.140007
2017-01-04	115.849998	116.510002	115.750000	116.019997	112.014503
2017-01-05	115.919998	116.860001	115.809998	116.610001	112.584129
2017-01-06	116.779999	118.160004	116.470001	117.910004	113.839249
2017-01-09	117.949997	119.430000	117.940002	118.989998	114.881950
2017-01-10	118.769997	119.379997	118.300003	119.110001	114.997818
2017-01-11	118.739998	119.930000	118.599998	119.750000	115.615723
2017-01-12	118.900002	119.300003	118.209999	119.250000	115.132988
2017-01-13	119.110001	119.620003	118.809998	119.040001	114.930237
2017-01-17	118.339996	120.239998	118.220001	120.000000	115.857086
2017-01-18	120.000000	120.500000	119.709999	119.989998	115.847435
2017-01-19	119.400002	120.089996	119.370003	119.779999	115.644691
2017-01-20	120.449997	120.449997	119.730003	120.000000	115.857086
2017-01-23	120.000000	120.809998	119.769997	120.080002	115.934326
2017-01-24	119.550003	120.099998	119.500000	119.970001	115.828125
2017-01-25	120.419998	122.099998	120.279999	121.879997	117.672188
2017-01-26	121.669998	122.440002	121.599998	121.940002	117.730118
2017-01-27	122.139999	122.349998	121.599998	121.949997	117.739769
2017-01-30	120.930000	121.629997	120.660004	121.629997	117.430817

```
2017-01-31  121.150002  121.389999  120.620003  121.349998  117.160492

              Volume        KAMA
Date
2017-01-03  28781900         NaN
2017-01-04  21118100         NaN
2017-01-05  22193600         NaN
2017-01-06  31751900         NaN
2017-01-09  33561900         NaN
2017-01-10  24462100         NaN
2017-01-11  27588600         NaN
2017-01-12  27086200         NaN
2017-01-13  26111900         NaN
2017-01-17  34439800         NaN
2017-01-18  23713000  114.570112
2017-01-19  25597300  114.804847
2017-01-20  32597900  115.021096
2017-01-23  22050200  115.159111
2017-01-24  23211000  115.203424
2017-01-25  32377600  115.615244
2017-01-26  26337600  115.910036
2017-01-27  20562900  116.357689
2017-01-30  30377500  116.590162
2017-01-31  49201000  116.647749
```

[6]:
```
plt.figure(figsize=(14,10))
plt.plot(df['Adj Close'])
plt.plot(df['KAMA'])
plt.ylabel('Price')
plt.xlabel('Date')
plt.title('Stock Closing Price of ' + str(n) + '-Day KAMA')
plt.legend(loc='best')
```

[6]:
```
<matplotlib.legend.Legend at 0x1a7347a4860>
```



1.1 Candlestick with KAMA

[7]:
```
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()
```

[7]:

	Date	Open	High	Low	Close	Adj Close	Volume	KAMA	VolumePositive
0	736332.0	115.800003	116.330002	114.760002	116.150002	112.140007	28781900	NaN	False
1	736333.0	115.849998	116.510002	115.750000	116.019997	112.014503			
2	736334.0	115.919998	116.860001	115.809998	116.610001	112.584129			
3	736335.0	116.779999	118.160004	116.470001	117.910004	113.839249			
4	736338.0	117.949997	119.430000	117.940002	118.989998	114.881950			

```
1  21118100   NaN          False
2  22193600   NaN          False
3  31751900   NaN          False
4  33561900   NaN          False
```

[8]:
```
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.plot(df['KAMA'], label='KAMA')
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')
```

[8]:
```
Text(0.5,0,'Date')
```