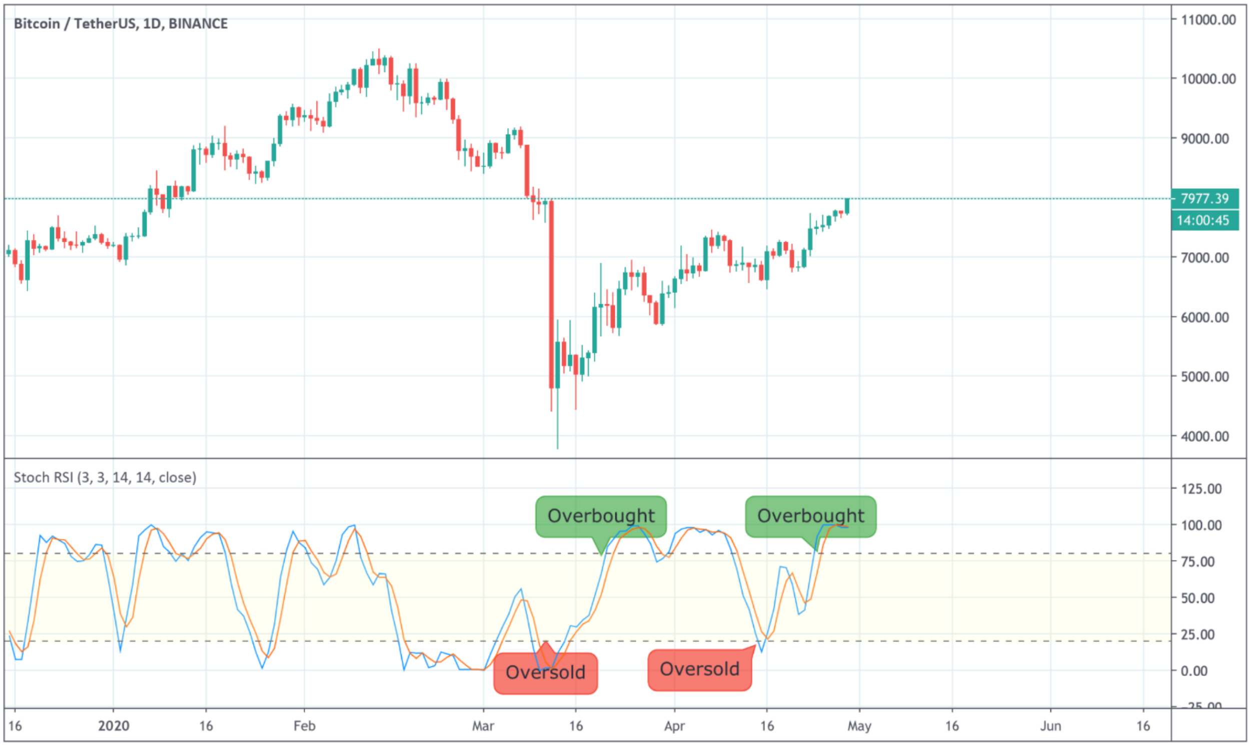
Jawher Sakka

<u>This program uses a technical indicator (stochastic RSI) to determinate when to buy and sell stocks</u>

Difference between RSI and stochastic RSI:

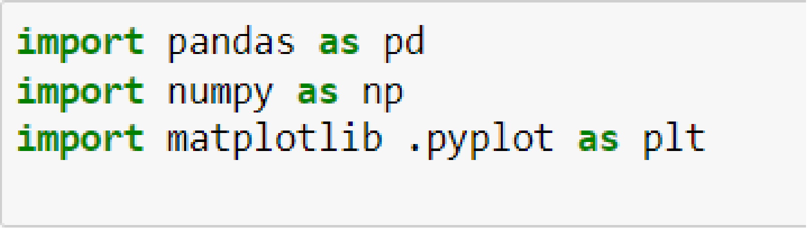
- **The relative strength index (RSI) was designed to measure the speed of price movements and it is more useful in trending markets**
- **The stochastic RSI (stochRSI) works best when the market is trading in consistent ranges and it are more useful in sideways or choppy markets**

This is an exemple for a CryptoHero Stoch RSI



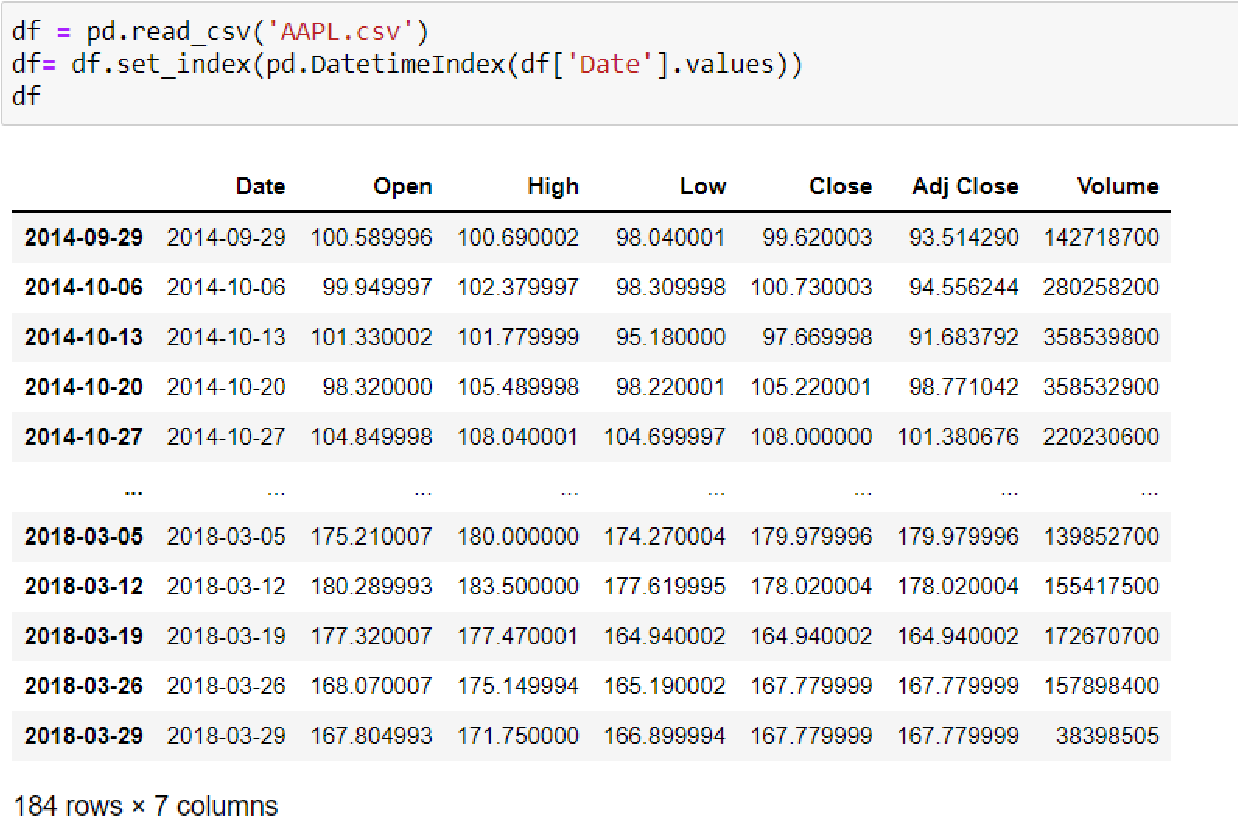
⇨ a buy signal is generated when Stoch RSI is oversold, a sell signal is generated when Stoch RSI is overbought

1- import libraries:



```
import pandas as pd
import numpy as np
import matplotlib .pyplot as plt
```

2-Read the file and show the data :

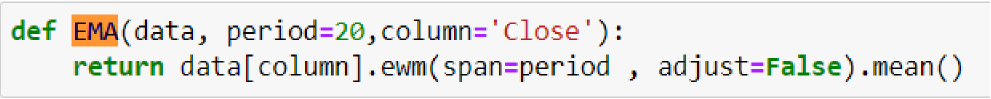


```
df = pd.read_csv('AAPL.csv')
df= df.set_index(pd.DatetimeIndex(df['Date'].values))
df
```

	Date	Open	High	Low	Close	Adj Close	Volume
2014-09-29	2014-09-29	100.589996	100.690002	98.040001	99.620003	93.514290	142718700
2014-10-06	2014-10-06	99.949997	102.379997	98.309998	100.730003	94.556244	280258200
2014-10-13	2014-10-13	101.330002	101.779999	95.180000	97.669998	91.683792	358539800
2014-10-20	2014-10-20	98.320000	105.489998	98.220001	105.220001	98.771042	358532900
2014-10-27	2014-10-27	104.849998	108.040001	104.699997	108.000000	101.380676	220230600
...	...	...	...	...	...	...	...
2018-03-05	2018-03-05	175.210007	180.000000	174.270004	179.979996	179.979996	139852700
2018-03-12	2018-03-12	180.289993	183.500000	177.619995	178.020004	178.020004	155417500
2018-03-19	2018-03-19	177.320007	177.470001	164.940002	164.940002	164.940002	172670700
2018-03-26	2018-03-26	168.070007	175.149994	165.190002	167.779999	167.779999	157898400
2018-03-29	2018-03-29	167.804993	171.750000	166.899994	167.779999	167.779999	38398505

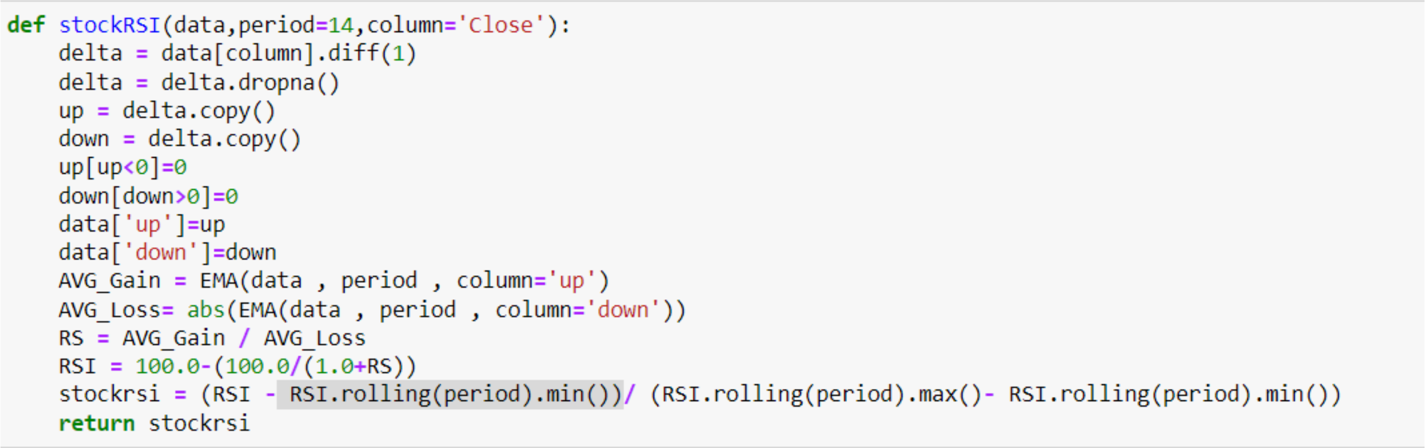
184 rows × 7 columns

3-Create an Exponential Moving Average indicator function:



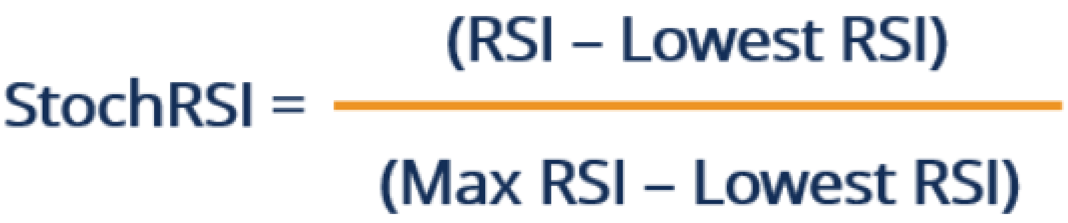
```
def EMA(data, period=20,column='Close'):
    return data[column].ewm(span=period , adjust=False).mean()
```

4-Create a function to calculate the Stochastic Relative Strength Index (StochRSI):



```
def stockRSI(data,period=14,column='Close'):
    delta = data[column].diff(1)
    delta = delta.dropna()
    up = delta.copy()
    down = delta.copy()
    up[up<0]=0
    down[down>0]=0
    data['up']=up
    data['down']=down
    AVG_Gain = EMA(data , period , column='up')
    AVG_Loss= abs(EMA(data , period , column='down'))
    RS = AVG_Gain / AVG_Loss
    RSI = 100.0-(100.0/(1.0+RS))
    stockrsi = (RSI - RSI.rolling(period).min())/ (RSI.rolling(period).max()- RSI.rolling(period).min())
    return stockrsi
```

How to Calculate Stochastic RSI :



$$\text{StochRSI} = \frac{(\text{RSI} - \text{Lowest RSI})}{(\text{Max RSI} - \text{Lowest RSI})}$$

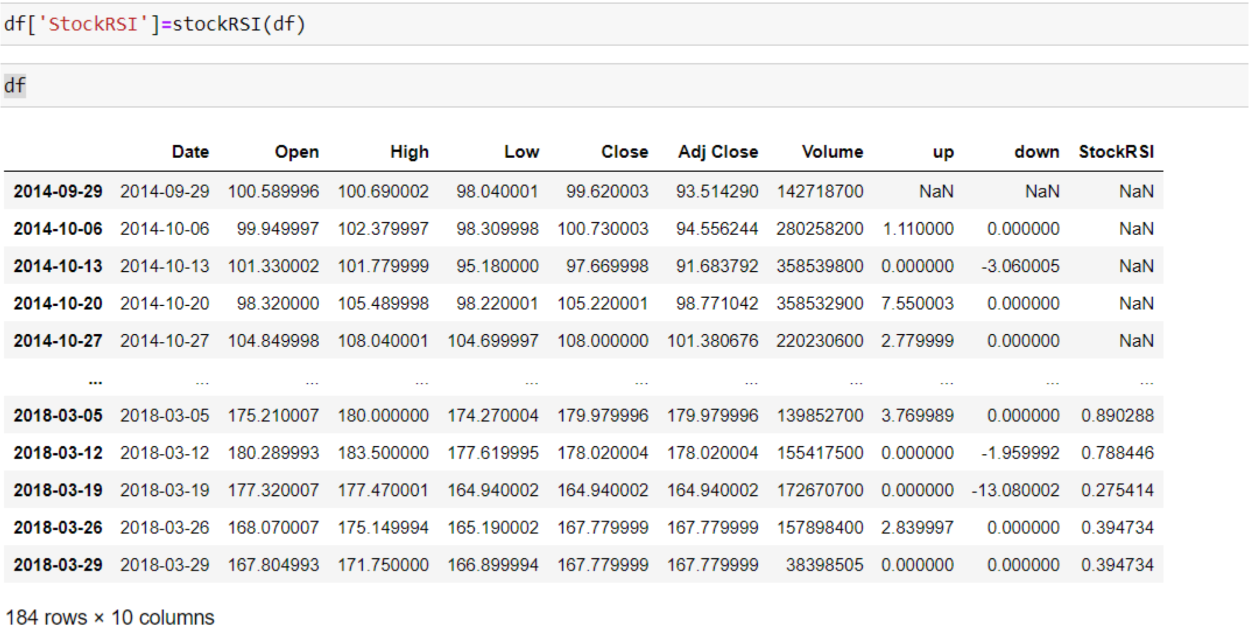
Where:

RSI = Current RSI reading

Lower RSI = Minimum RSI reading since the last 14 oscillations

Max RSI = Maximum RSI reading for the last 14 periods

5- Store the StockRSI data in new column and show the data :



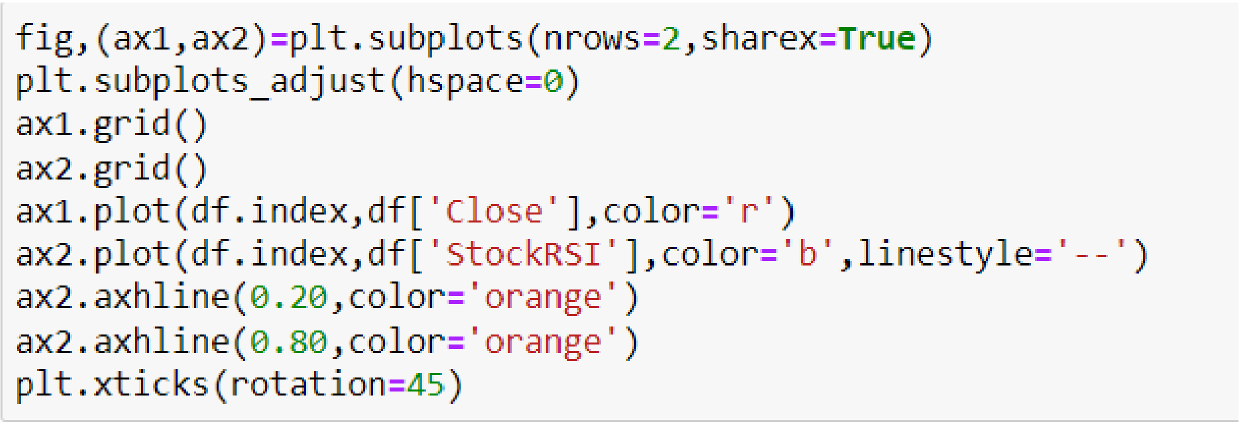
```
df['StockRSI']=stockRSI(df)
```

```
df
```

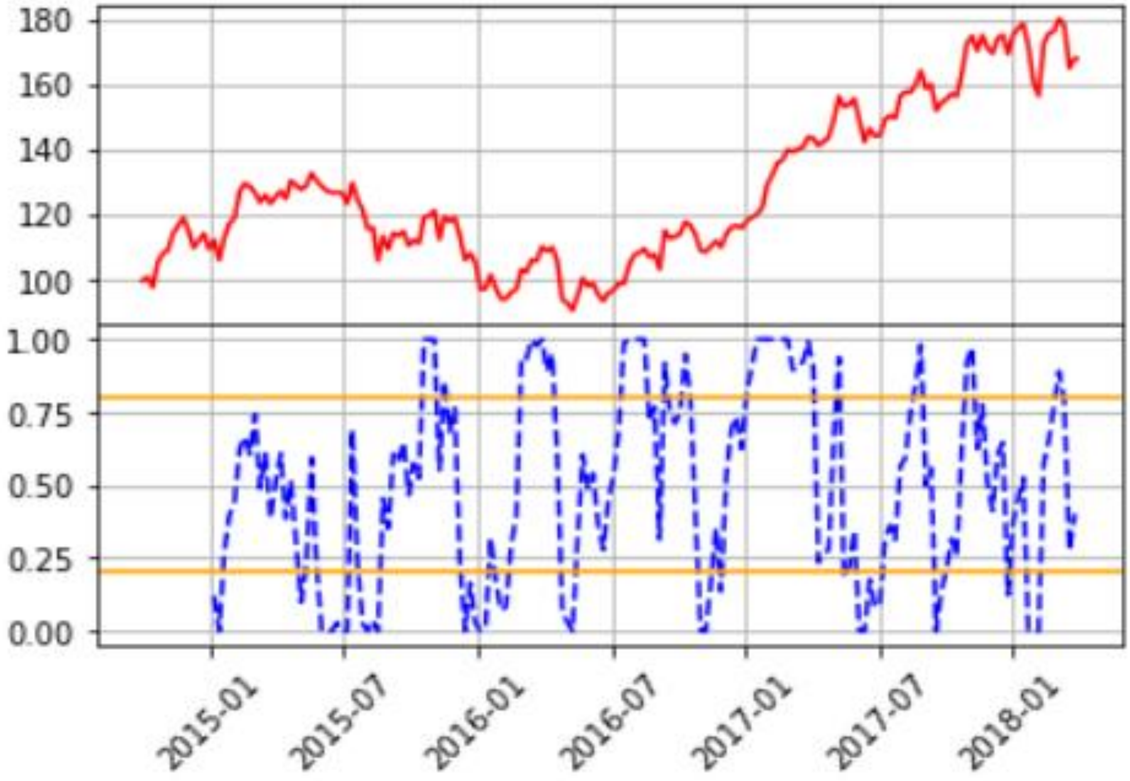
	Date	Open	High	Low	Close	Adj Close	Volume	up	down	StockRSI
2014-09-29	2014-09-29	100.589996	100.690002	98.040001	99.620003	93.514290	142718700	NaN	NaN	NaN
2014-10-06	2014-10-06	99.949997	102.379997	98.309998	100.730003	94.556244	280258200	1.110000	0.000000	NaN
2014-10-13	2014-10-13	101.330002	101.779999	95.180000	97.669998	91.683792	358539800	0.000000	-3.060005	NaN
2014-10-20	2014-10-20	98.320000	105.489998	98.220001	105.220001	98.771042	358532900	7.550003	0.000000	NaN
2014-10-27	2014-10-27	104.849998	108.040001	104.699997	108.000000	101.380676	220230600	2.779999	0.000000	NaN
...	...	...	...	...	...	...	...	...	...	...
2018-03-05	2018-03-05	175.210007	180.000000	174.270004	179.979996	179.979996	139852700	3.769989	0.000000	0.890288
2018-03-12	2018-03-12	180.289993	183.500000	177.619995	178.020004	178.020004	155417500	0.000000	-1.959992	0.788446
2018-03-19	2018-03-19	177.320007	177.470001	164.940002	164.940002	164.940002	172670700	0.000000	-13.080002	0.275414
2018-03-26	2018-03-26	168.070007	175.149994	165.190002	167.779999	167.779999	157898400	2.839997	0.000000	0.394734
2018-03-29	2018-03-29	167.804993	171.750000	166.899994	167.779999	167.779999	38398505	0.000000	0.000000	0.394734

184 rows × 10 columns

6- Plot the data



```
fig,(ax1,ax2)=plt.subplots(nrows=2,sharex=True)
plt.subplots_adjust(hspace=0)
ax1.grid()
ax2.grid()
ax1.plot(df.index,df['Close'],color='r')
ax2.plot(df.index,df['StockRSI'],color='b',linestyle='--')
ax2.axhline(0.20,color='orange')
ax2.axhline(0.80,color='orange')
plt.xticks(rotation=45)
```



- ⇨ According to the technical traders, StochRSI oscillates between 20 and 80 for a long period without reaching an extreme range , the StochRSI uses 20 and 80 as the price overbought and oversold, respectively