# Twitter Thread by Indian Quant ■■



**Indian Quant ■■**
@indian_quant

## Nano Course On Python For Trading
==========
## Module 3

## In this post, I will attempt to teach you how to write an algorithm in python to automatically find support and resistance. We shall use the data we downloaded in module 2.

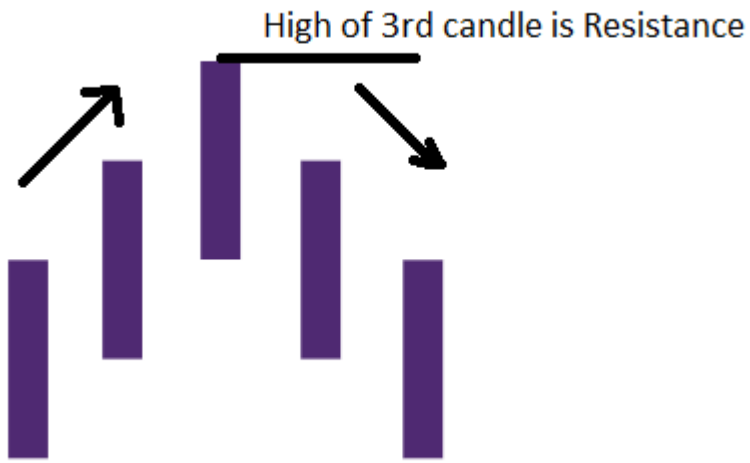
**+**

To visualize price action, I have used the daily candlestick chart. To quantify the meaning of support and resistance, I have taken below definitions:
In a series of 5 continuous candles, if the first three candles make increasing high and then 2 decreasing highs, then the

+

high of the third candle is considered resistance. Similarly, if the first 3 candles make lower lows and the next 2 make high lows then the low of the third candle is support.

+




The above explanation is for the sake of quantifying support and resistance at a very basic level. These are called fractals. Now let's get started with the code.

mpl_finance is the new module here. We will use its candlestick_ohlc function.

+

```
import pandas as pd
import numpy as np
import yfinance
from mpl_finance import candlestick_ohlc
import matplotlib.dates as mpl_dates
import matplotlib.pyplot as plt

plt.rcParams['figure.figsize'] = [12, 7]

plt.rc('font', size=14)
```

Now, if you remember, you downloaded data for nifty 50 stocks in module 2. We shall use that data. Let's work on ITC data this time.
We will use data from 1 Jan 2021 to 30 June 2021.

+

```
df = pd.read_csv(r"C:\Users\pvarshney\Downloads\BlackField Research\Content\Nifty 50 Data\ITC.csv")
df = df.drop(['Unnamed: 0'], axis = 1)
df
start_date = "2021-01-01"
end_date = "2021-06-31"
df = df[(df['Date'] >= start_date) & (df['Date'] <= end_date)]

df
```

| | Date | Open | High | Low | Close |
|---|---|---|---|---|---|
| **1730** | 2021-01-01 | 209.90 | 214.20 | 209.30 | 213.85 |
| **1731** | 2021-01-04 | 214.00 | 216.00 | 212.10 | 213.40 |
| **1732** | 2021-01-05 | 212.45 | 213.25 | 210.60 | 211.45 |
| **1733** | 2021-01-06 | 211.45 | 211.50 | 204.50 | 205.40 |
| **1734** | 2021-01-07 | 206.10 | 207.60 | 202.10 | 202.80 |
| **...** | ... | ... | ... | ... | ... |
| **1847** | 2021-06-24 | 204.45 | 204.65 | 202.55 | 203.25 |
| **1848** | 2021-06-25 | 204.00 | 205.60 | 203.75 | 205.05 |
| **1849** | 2021-06-28 | 205.90 | 205.90 | 203.35 | 203.60 |
| **1850** | 2021-06-29 | 203.60 | 205.30 | 203.15 | 203.30 |
| **1851** | 2021-06-30 | 204.30 | 204.40 | 202.55 | 202.70 |

Let's plot the candlestick chart for this data frame. We are going to use the candlestick_ohlc function from the mpl_finance module. I have defined a function plot_chart() which you can see in the google colab link in the last thread

+




We will iterate over this dataframe and see if the low of the current candle is a support or high is a resistance. For that, we have created an array named levels.

Again remember the definition of fractals, we shall leave the first two candles and the last 2 candles as we

+

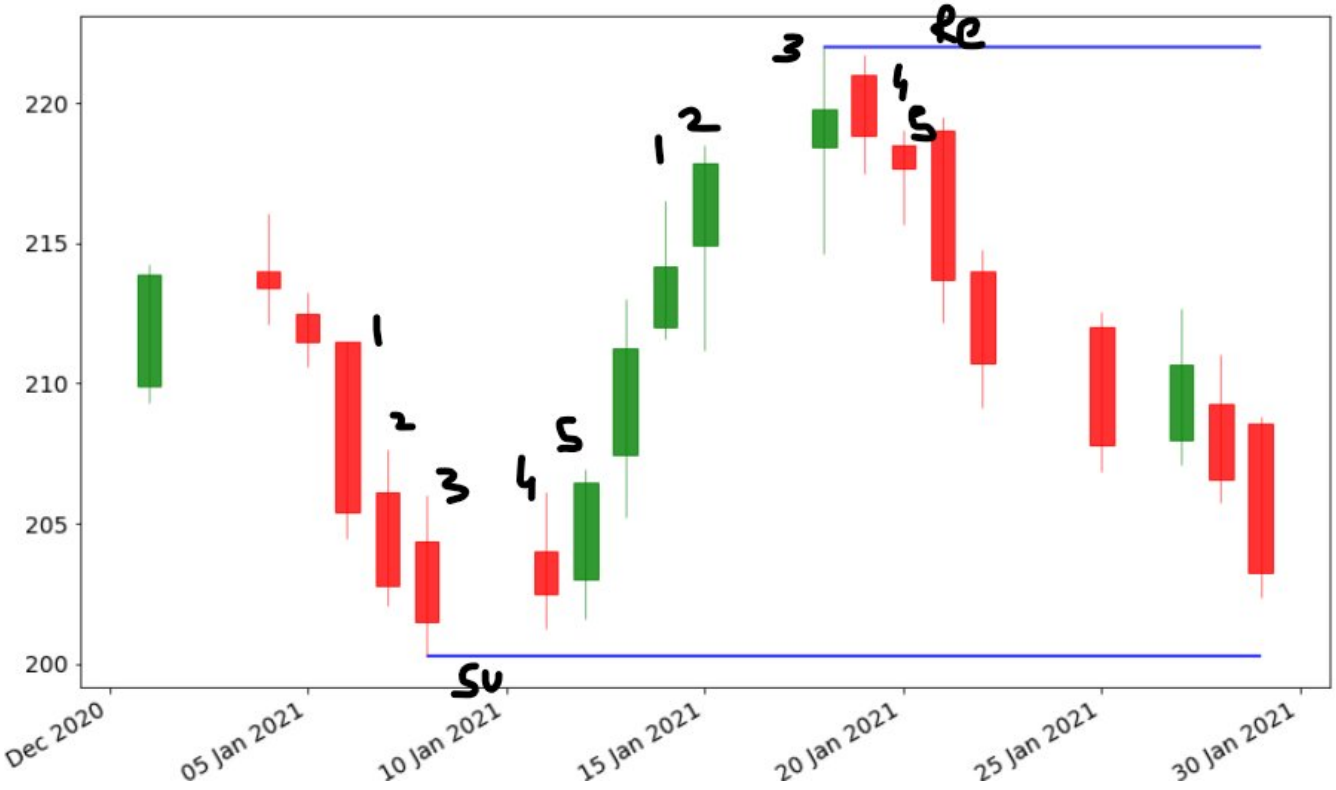
need two candles to the right of a candle in order to consider it a fractal. Similarly, we will leave the last two candles. We will iterate over the data frame from index 2 till it is 2 less than its size. We also need to define two functions "isSupport" and "isResistance"

+

```
levels = []
for i in range(2,df.shape[0]-2):
    print(i)
    if isSupport(df,i):
        levels.append((i,df['Low'][i]))
    elif isResistance(df,i):
        levels.append((i,df['High'][i]))
```

determines and returns true/false if a candle low is a support or its high is resistance. In this image below 1-2-3-4-5 is a fractal as explained above, low of the candle on 10 Jan is support and high on 20 Jan is resistance. Repeating this process for the entire data frame.

+




To plot the levels we add it in our plot function (named it plot_all() ) and execute it.

+

```
12
13      for level in levels:
14          plt.hlines(level[1],xmin=df['Date'][level[0]],\
15                     xmax=max(df['Date']),colors='blue')
16          fig.show()
```

But, alas it looks so ugly currently, I don't wanna look at it. There is a way that we can reduce the number of levels marked. What we will do that will first find the average length of a candle and reject all other price levels which are closed then twice this average

+




let's find the average length of a candle i.e. average of high - low

s = 2 * np.mean(df['High'] - df['Low'])

Using this number, we will reject the current price level if there is already another price level within the range +- s. For this, we defined a function isFarFromLevel

Finally, we have reduced the number of price levels and the chart looks like below:

+




If you want to work on the code and tinker with the code, go to Google Colab https://t.co/G8DuZvWO0g

I have shared the entire code on colab and added references for you to look into.

D: these won't help you become profitable but this will teach you how quant traders think &

+

automate their day-to-day tasks and quantify certain ideas. If you have any doubt reach out to me in DM.

+

In Module 4, we will learn about cloud computing and the use of cloud in trading. Until then, happy learning.