Rithik Tripathi

Connect with me on Linkedin (https://www.linkedin.com/in/rithik-tripathi-data-scientist/)

```
Problem Statement

Write a program in R/Python (preferably R) to detect the occurrence of following four patterns in OHLC data. You can download
sample OHLC data in .csv/.xlsx/.xls format for your reference using Google.

1) Bullish Engulfing

2) Bearish Engulfing

3) Bullish Rising Three

4) Bearish Falling Three
```

Note : The techniqies are not limited to the conditions used below, there might be many more factors affecting the patterns, These are coded to get a high level review. Again, We could use High/ Low & Open/ Close to Analyze & I have tried using both to better understand the patterns

In [1]:

```python
# importing required libraries
import plotly.express as px
import plotly.graph_objects as go
import numpy as np
import pandas as pd
import time
import os
import glob

from nsepy import get_history
from datetime import date, timedelta, datetime


class NIFTY():

    def fetch_data(self,start_year, end_year):
        '''
        Generated NIFTY 50 separate .csv files for each year from start to end as passed in parameters.
        sleeps for 10 seconds after each call to allow api to process request.

        Params:
            start_year : year from which NIFTY data is to be fetched
            end_year   : year(included) upto NIFTY data is to be fetched

        Returns:
            yearly_data_list : Fetched yearly data (dtype = List)
            available_years  : dictionory containing index position of available years in yearly_data_list
        '''

        years = [year for year in range(start_year, end_year+1)]

        available_years = []
        yearly_data_list = []

        print('\nFetching Data...')
        for year in years:
            year_start = datetime(year,1,1)
            year_end = datetime(year,12,31)

            data = get_history(symbol = "NIFTY 50", start= year_start, end=year_end, index=True)
            data[["Open", "High", "Low", "Close"]]

            data = pd.DataFrame(data)
            data = data.reset_index()
            yearly_data_list.append(data)

            print('\nData for year ', year, ' Fetched Successfully.')

            df_name = 'df_'+str(year)+'.csv'
            available_years.append(df_name)

            data.to_csv(df_name)
            print('Exported ', year, ' data in ',df_name)

            time.sleep(10)

        return yearly_data_list, dict(enumerate(available_years))


```

In [2]:

```
nifty = NIFTY()
```

In [3]:

```
yearly_data_list, available_years = nifty.fetch_data(start_year= 2017, end_year= 2022)
```

```
Fetching Data...

Data for year  2017  Fetched Successfully.
Exported  2017  data in  df_2017.csv

Data for year  2018  Fetched Successfully.
Exported  2018  data in  df_2018.csv

Data for year  2019  Fetched Successfully.
Exported  2019  data in  df_2019.csv

Data for year  2020  Fetched Successfully.
Exported  2020  data in  df_2020.csv

Data for year  2021  Fetched Successfully.
Exported  2021  data in  df_2021.csv

Data for year  2022  Fetched Successfully.
Exported  2022  data in  df_2022.csv
```

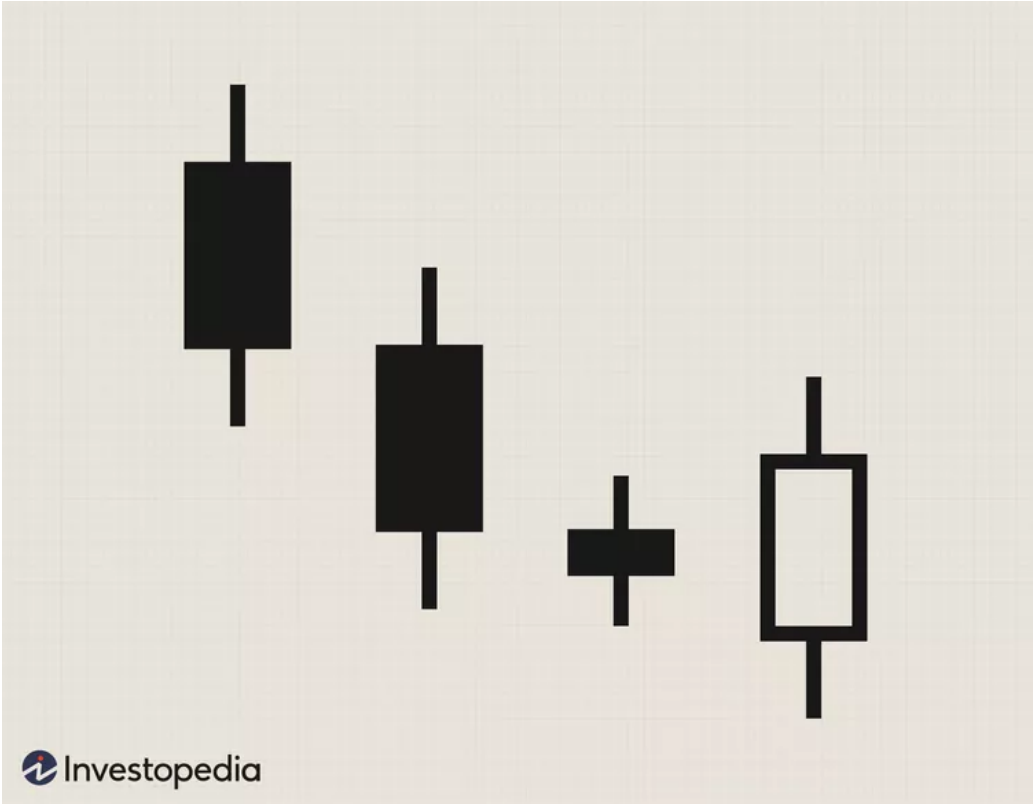
In [4]:

```
import pandas as pd

df = pd.read_csv("df_2022.csv")

df.drop(['Unnamed: 0'], axis=1, inplace=True)

df['Previous_Close'] = df['Close'].shift(1)
df['Previous_Open'] = df['Open'].shift(1)
df.head()
```

Out[4]:

	Date	Open	High	Low	Close	Volume	Turnover	Previous_Close	Previous_Open
0	2022-01-03	17387.15	17646.65	17383.30	17625.70	200456430	1.618136e+11	NaN	NaN
1	2022-01-04	17681.40	17827.60	17593.55	17805.25	247437472	1.860416e+11	17625.70	17387.15
2	2022-01-05	17820.10	17944.70	17748.85	17925.25	251460038	2.373731e+11	17805.25	17681.40
3	2022-01-06	17768.50	17797.95	17655.55	17745.90	236454824	2.264382e+11	17925.25	17820.10
4	2022-01-07	17797.60	17905.00	17704.55	17812.70	239338015	2.144789e+11	17745.90	17768.50

1. Bullish Engulfing




```
> Conditions required to be met in order to declare a candle as Bullish Engulfing
  ------------------------------------------------------------------------------

->> The current candle's open price is less than the previous candle's close price. This means that the current candle is opening below the previous candle's close, which is a bearish indication.

->> The current candle's close price is greater than the previous candle's open price. This means that the current candle is closing above the previous candle's open, which is a bullish indication.

->> The current candle's close price is greater than the previous candle's close price. This means that the current candle is closing above the previous candle's close, which is a bullish indication.
```

In [5]:

```
# DFunction to detect the Bullish Engulfing pattern
def detect_bullish_engulfing(data):

    # empty list to store values of dates where bullish engulfing pattern is True
    bullish_engulfing = []

    for i in range(1, len(data)):
        # defining conditions
        if (
              (data.iloc[i]['Open'] < data.iloc[i-1]['Close']) and
              (data.iloc[i]['Close'] > data.iloc[i-1]['Open']) and
              (data.iloc[i]['Close'] > data.iloc[i-1]['Close'])
           ):

                    bullish_engulfing.append(data.iloc[i]['Date'])

    return bullish_engulfing

# Calling function to detect the Bullish Engulfing pattern
bullish_engulfing = detect_bullish_engulfing(df)

# Print the dates where the Bullish Engulfing pattern was detected
print(bullish_engulfing)

```

```
['2022-01-11', '2022-01-17', '2022-02-28', '2022-03-08', '2022-04-01', '2022-06-02', '2022-06-09', '2022-07-04', '2022-07-19', '2022-07-27', '2022-08-02', '2022-08-18', '2022-08-24', '2022-09-30', '2022-11-16', '2022-11-28', '2022-11-29', '2022-12-29']
```

In [ ]:

```

```

In [6]:

```
# comlpete details with Bullish Engulfing using bullish_engulfing list (Note : only top 5 records are displayed)
df[df['Date'].isin(bullish_engulfing)].head()
```

Out[6]:

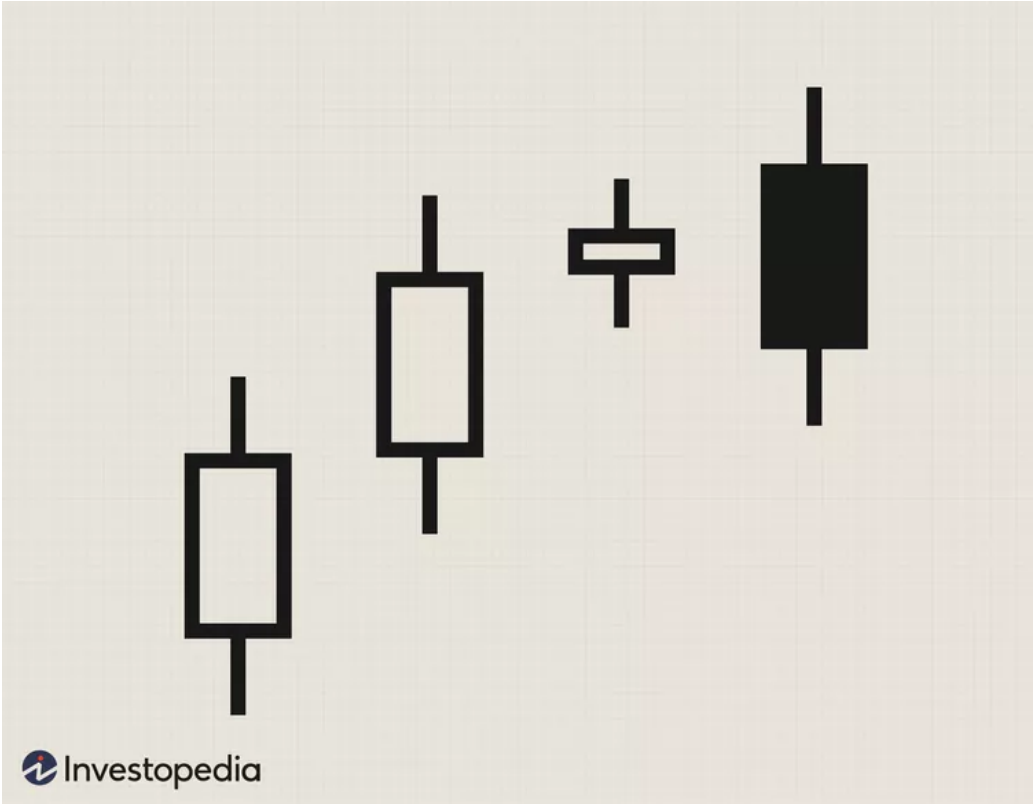
	Date	Open	High	Low	Close	Volume	Turnover	Previous_Close	Previous_Open
6	2022-01-11	17997.75	18081.25	17964.40	18055.75	220238796	2.078146e+11	18003.30	17913.30
10	2022-01-17	18235.65	18321.55	18228.75	18308.10	266702919	2.385938e+11	18255.75	18185.00
39	2022-02-28	16481.60	16815.90	16356.30	16793.90	404214666	3.383157e+11	16658.40	16515.65
44	2022-03-08	15747.75	16028.75	15671.45	16013.45	543600673	3.870440e+11	15863.15	15867.95
61	2022-04-01	17436.90	17703.70	17422.70	17670.45	291773447	2.113213e+11	17464.75	17519.20

In [7]:

```
fig = go.Figure(data=[go.Candlestick(x=df.index, open=df['Open'], high=df['High'], low=df['Low'], close=df['Close'], increasing=d

fig.add_trace(go.Scatter(x=df[df['Date'].isin(bullish_engulfing)].index, y=df[df['Date'].isin(bullish_engulfing)]['Close'], mode=

fig.update_layout(
    title='NIFTY Bullish Engulfing',
    xaxis_title='Year (increasing order)',
    yaxis_title='Value (increasing order)')

fig.show()
```

Screenshot

2. Bearish Engulfing




```
> Conditions required to be met in order to declare a candle as Bearish Engulfing
  ------------------------------------------------------------------------------

->> The current candle's open price is greater than the previous candle's close price. This means that the current candle is opening above the previous candle's close, which is a bullish indication.

->> The current candle's close price is less than the previous candle's open price. This means that the current candle is closing below the previous candle's open, which is a bearish indication.

->> The current candle's close price is less than the previous candle's close price. This means that the current candle is closing below the previous candle's close, which is a bearish indication.
```

In [8]:

```python
# Function to detect the Bearish Engulfing pattern

def detect_bearish_engulfing(data):
    # empty list to store values of dates where bullish engulfing pattern is True
    bearish_engulfing = []

    for i in range(1, len(data)):
        # defining conditions
        if (
               (data.iloc[i]['Open'] > data.iloc[i-1]['Close']) and
               (data.iloc[i]['Close'] < data.iloc[i-1]['Open']) and
               (data.iloc[i]['Close'] < data.iloc[i-1]['Close'])
           ):

                    bearish_engulfing.append(data.iloc[i]['Date'])

    return bearish_engulfing

# Call the function to detect the Bearish Engulfing pattern
bearish_engulfing = detect_bearish_engulfing(df)

# Print the dates where the Bearish Engulfing pattern was detected
print(bearish_engulfing)

```

```
['2022-01-18', '2022-01-19', '2022-02-04', '2022-02-17', '2022-03-03', '2022-03-21', '2022-03-31', '2022-04-13', '2022-
04-19', '2022-04-29', '2022-05-04', '2022-05-11', '2022-05-13', '2022-05-24', '2022-05-25', '2022-06-01', '2022-06-08',
'2022-06-16', '2022-07-13', '2022-07-14', '2022-07-26', '2022-08-19', '2022-08-25', '2022-09-27', '2022-09-29', '2022-1
0-11', '2022-10-25', '2022-11-02', '2022-11-09', '2022-11-18', '2022-12-09', '2022-12-21', '2022-12-22']
```

In [9]:

```
# comlpete details with Bearish Engulfing using bearish_engulfing list (Note : only top 5 records are displayed)
df[df['Date'].isin(bearish_engulfing)].head()
```

Out[9]:

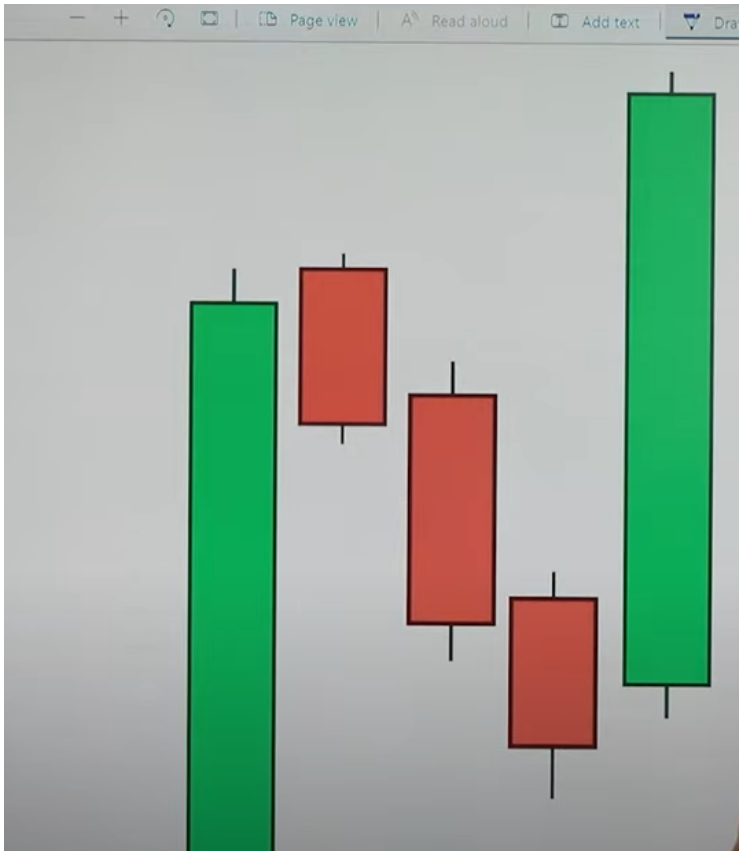
	Date	Open	High	Low	Close	Volume	Turnover	Previous_Close	Previous_Open
11	2022-01-18	18337.20	18350.95	18085.90	18113.05	227507319	2.085267e+11	18308.10	18235.65
12	2022-01-19	18129.20	18129.20	17884.90	17938.40	276662654	2.531019e+11	18113.05	18337.20
23	2022-02-04	17590.20	17617.80	17462.55	17516.30	261434170	2.065185e+11	17560.20	17767.75
32	2022-02-17	17396.55	17442.90	17235.85	17304.60	232136131	1.938126e+11	17322.20	17408.45
41	2022-03-03	16723.20	16768.95	16442.95	16498.05	442068263	3.141300e+11	16605.95	16593.10

In [10]:

```
fig = go.Figure(data=[go.Candlestick(x=df.index, open=df['Open'], high=df['High'], low=df['Low'], close=df['Close'], increasing=d

fig.add_trace(go.Scatter(x=df[df['Date'].isin(bearish_engulfing)].index, y=df[df['Date'].isin(bearish_engulfing)]['Close'], mode=

fig.update_layout(
    title='NIFTY Bearish Engulfing',
    xaxis_title='Year (increasing order)',
    yaxis_title='Value (increasing order)')

fig.show()

```

Screenshot

3. Bullish Rising Three



In [11]:

```
# Function to detect the Bullish Rising Three

def detect_bullish_rising_three(df):

    # empty list to store values of dates where Bullish Rising Three pattern is True
    bullish_rising_three = []

    # starting from 4th record (indexing starts from 0) as previous 3 records are required to detect this pattern
    for i in range(3, df.shape[0] - 1):
        if (
            (df.iloc[i]['Low'] >= df.iloc[i-1]['Low'] and df.iloc[i]['High'] > df.iloc[i-1]['High'] and df.iloc[i]['High'] > df.i
            (df.iloc[i-1]['Low'] < df.iloc[i-2]['Low'] and df.iloc[i-1]['Low'] < df.iloc[i-3]['Low'] and  df.iloc[i-1]['High'] <
            (df.iloc[i-2]['Low'] < df.iloc[i-3]['Low'] and df.iloc[i-2]['High'] >= df.iloc[i-3]['Low'] and  df.iloc[i-2]['High']
        ):
            bullish_rising_three.append(df.iloc[i]['Date'])
    return bullish_rising_three


```

In [12]:

```
# 2017 => Using 2017 NIFTY data for this dataframe
df2 = pd.read_csv('df_2017.csv')
```

In [13]:

```
bullish_rising_three2 = detect_bullish_rising_three(df2)
bullish_rising_three2
```

Out[13]:

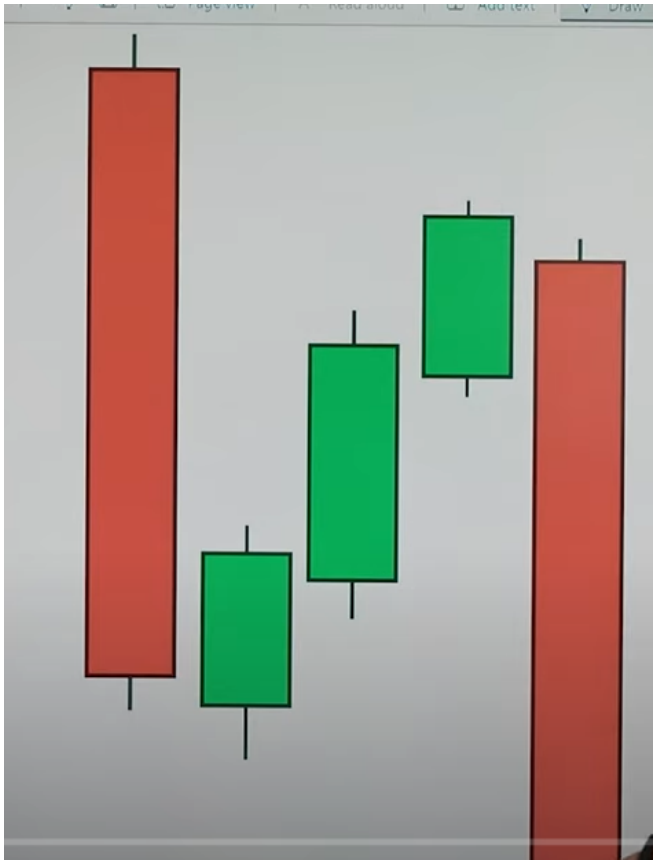
```
['2017-01-24', '2017-02-09', '2017-05-25', '2017-12-07', '2017-12-15']
```

In [14]:

```
import plotly.graph_objects as go

fig = go.Figure(data=[go.Candlestick(x=df2.index, open=df2['Open'], high=df2['High'], low=df2['Low'], close=df2['Close'], increas

fig.add_trace(go.Scatter(x=df2[df2['Date'].isin(bullish_rising_three2)].index, y=df2[df2['Date'].isin(bullish_rising_three2)]['Cl

fig.update_layout(
    title='NIFTY Bullish Rising Three',
    xaxis_title='Year (increasing order)',
    yaxis_title='Value (increasing order)')

fig.show()

```

Screenshot

4. Bearish Falling Three



In [72]:

```
def detect_Bearish_Falling_Three(df):
    Bearish_Falling_Three = []
    for i in range(3, df.shape[0] - 1):
        if (
            ( df.iloc[i]['High'] > df.iloc[i-2]['High'] and df.iloc[i]['High'] > df.iloc[i-3]['High']) and
            ( df.iloc[i]['Low'] < df.iloc[i-1]['Low'] and df.iloc[i]['Low'] < df.iloc[i-2]['Low'] and df.iloc[i]['Low'] < df.iloc
            ( df.iloc[i-1]['High'] >= df.iloc[i-2]['High'] and df.iloc[i-1]['High'] >= df.iloc[i-3]['High'] and df.iloc[i-1]['Low


        ):
            Bearish_Falling_Three.append(df.iloc[i]['Date'])
    return Bearish_Falling_Three
```

In [88]:

```
df = pd.read_csv("df_2016.csv")
```

In [89]:

```
Bearish_Falling_Three = detect_Bearish_Falling_Three(df)
Bearish_Falling_Three
```

Out[89]:

```
['2016-02-23']
```

In [90]:

```
import plotly.graph_objects as go

fig = go.Figure(data=[go.Candlestick(x=df.index, open=df['Open'], high=df['High'], low=df['Low'], close=df['Close'], increasing=d

fig.add_trace(go.Scatter(x=df[df['Date'].isin(Bearish_Falling_Three)].index, y=df[df['Date'].isin(Bearish_Falling_Three)]['Close'

fig.update_layout(
    title='NIFTY Bearish Falling Three',
    xaxis_title='Year (increasing order)',
    yaxis_title='Value (increasing order)')

fig.show()

```

Screenshot

In [ ]:

```
1
```