

```
In [2]: import pandas as pd
import numpy as np
import yfinance
#from operator import itemgetter
import matplotlib.dates as mpl_dates
import matplotlib.pyplot as plt
%matplotlib inline
#from mpl_finance import candlestick_ohlc
from mplfinance.original_flavor import candlestick_ohlc

plt.rcParams['figure.figsize'] = [14, 7]
plt.rc('font', size=14)
```

```
In [5]: symbol = "AAPL TSLA SPY AMZN BABA GOOGL MSFT NFLX PYPL COST AAL"
num_name= symbol.split(' ')
data_frames_separate = []
for i in num_name:
    ticker = yfinance.Ticker(i)
    df=ticker.history(interval='1d',start="2020-03-15",end="2021-01-30")
    df['Date'] = pd.to_datetime(df.index) # just to crate another column for date!!!
    df['Date'] = df['Date'].apply(mpl_dates.date2num)
    df = df.loc[:,['Date', 'Open', 'High', 'Low', 'Close']]
    data_frames_separate.append(df)

len(data_frames_separate)
```

Out[5]: 11

```
In [6]: def isSupport(df,i):
    support = df['Low'][i] < df['Low'][i-1] and df['Low'][i] < df['Low'][i+1] \
    and df['Low'][i+1] < df['Low'][i+2] and df['Low'][i-1] < df['Low'][i-2]
```

```

    return support

def isResistance(df,i):
    resistance = df['High'][i] > df['High'][i-1] and df['High'][i] > df[
'High'][i+1] \
    and df['High'][i+1] > df['High'][i+2] and df['High'][i-1] > df['High'
][i-2]

    return resistance

```

```

In [7]: levels = [[] for i in range(len(num_name))]
        for a in range(len(levels)):
            for i in range(2,data_frames_separate[a].shape[0]-2):
                if isSupport(data_frames_separate[a],i):
                    levels[a].append((i,data_frames_separate[a]['Low'][i]))
                elif isResistance(data_frames_separate[a],i):
                    levels[a].append((i,data_frames_separate[a]['High'][i]))
        len(levels)

```

Out[7]: 11

In [ ]:

```

In [9]: def plot_all(num):
        fig, ax = plt.subplots()

        candlestick_ohlc(ax,data_frames_separate[num].values,width=0.6, \
                           colorup='green', colordown='red', alpha=0.8)

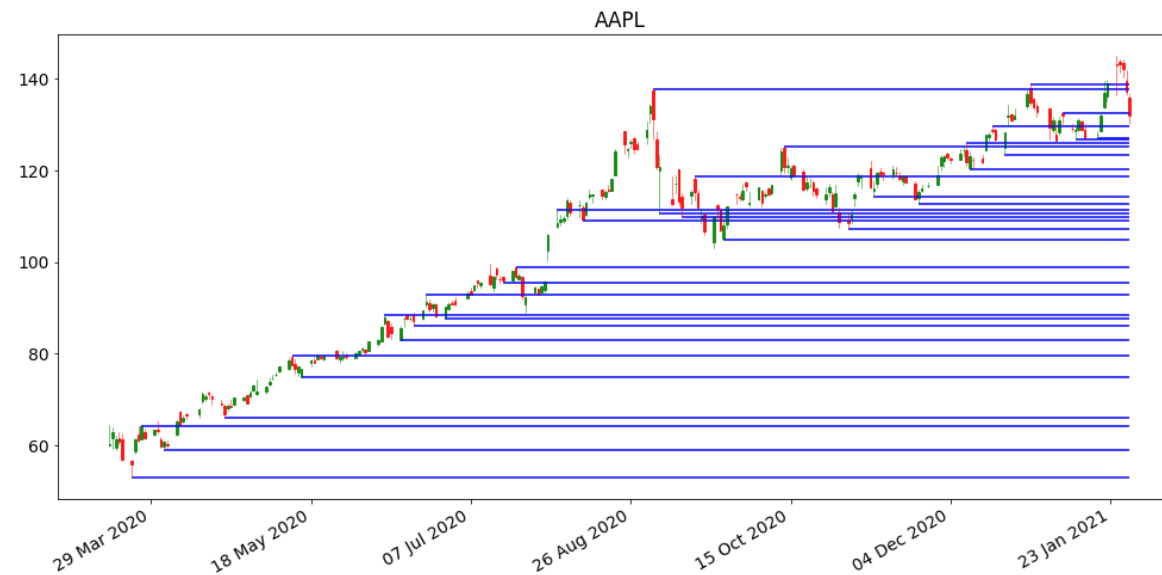
        date_format = mpl_dates.DateFormatter('%d %b %Y')
        ax.xaxis.set_major_formatter(date_format)
        fig.autofmt_xdate()
        plt.title(num_name[num])
        fig.tight_layout()

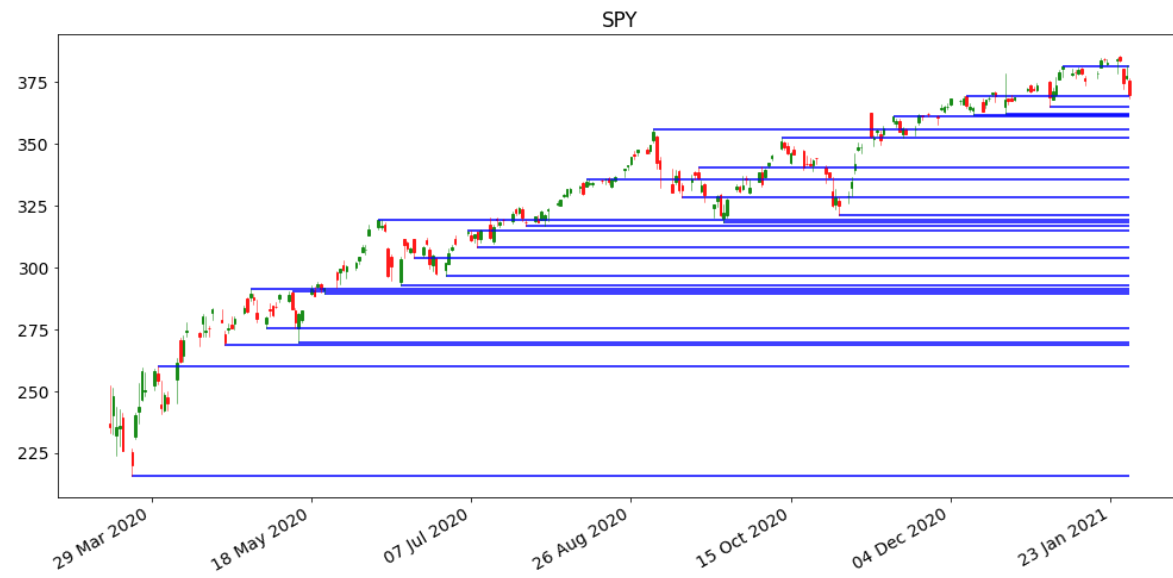
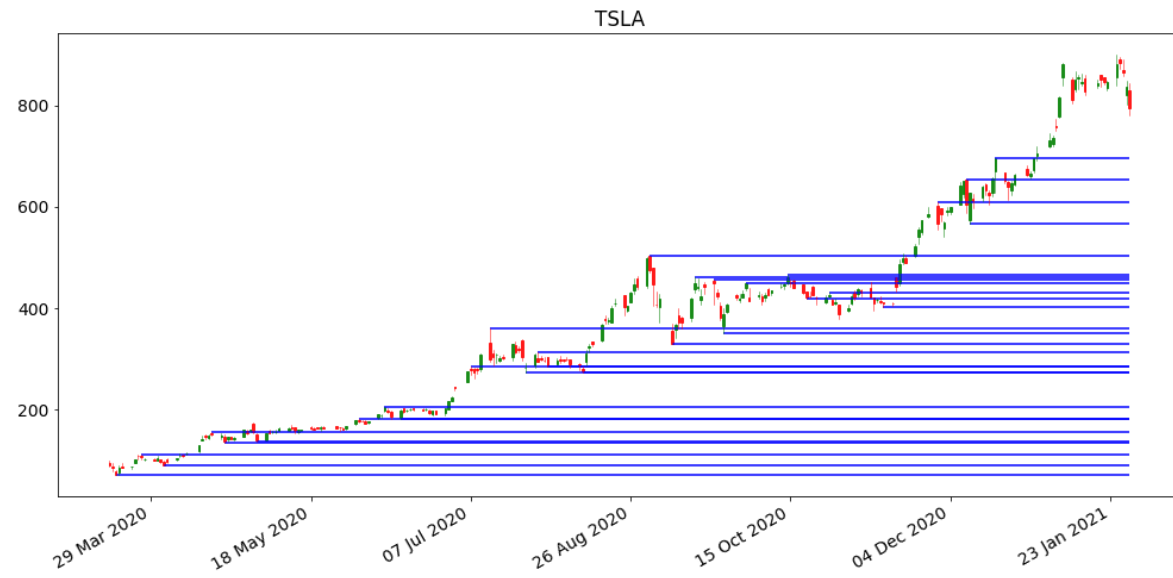
        for level in levels[num]:
            plt.hlines(level[1],xmin=data_frames_separate[num]['Date'][level[0]

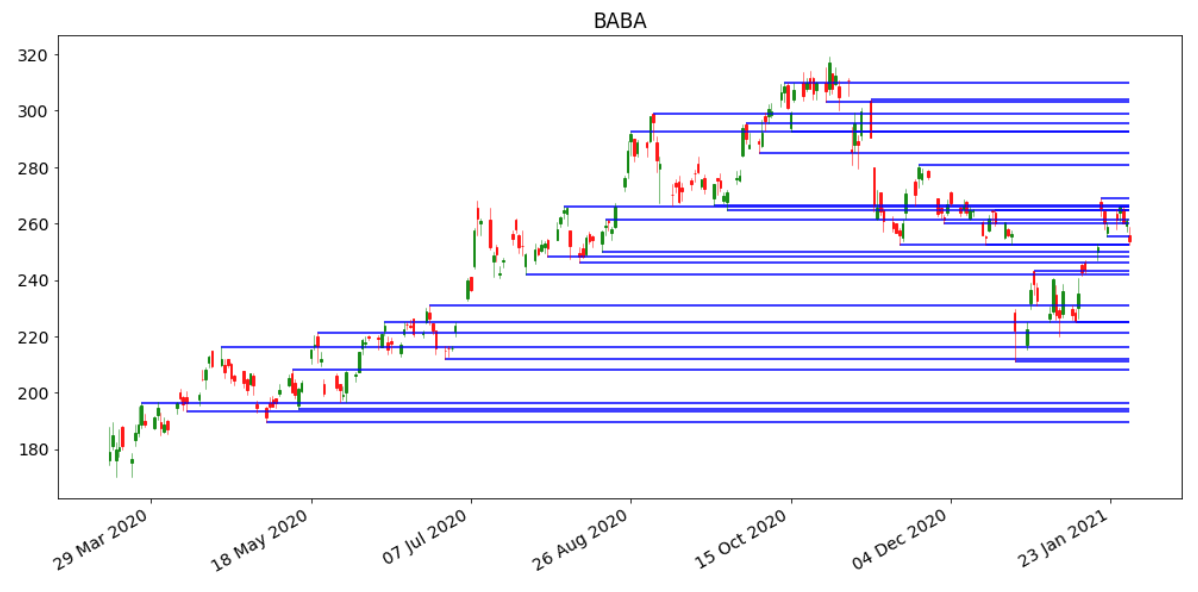
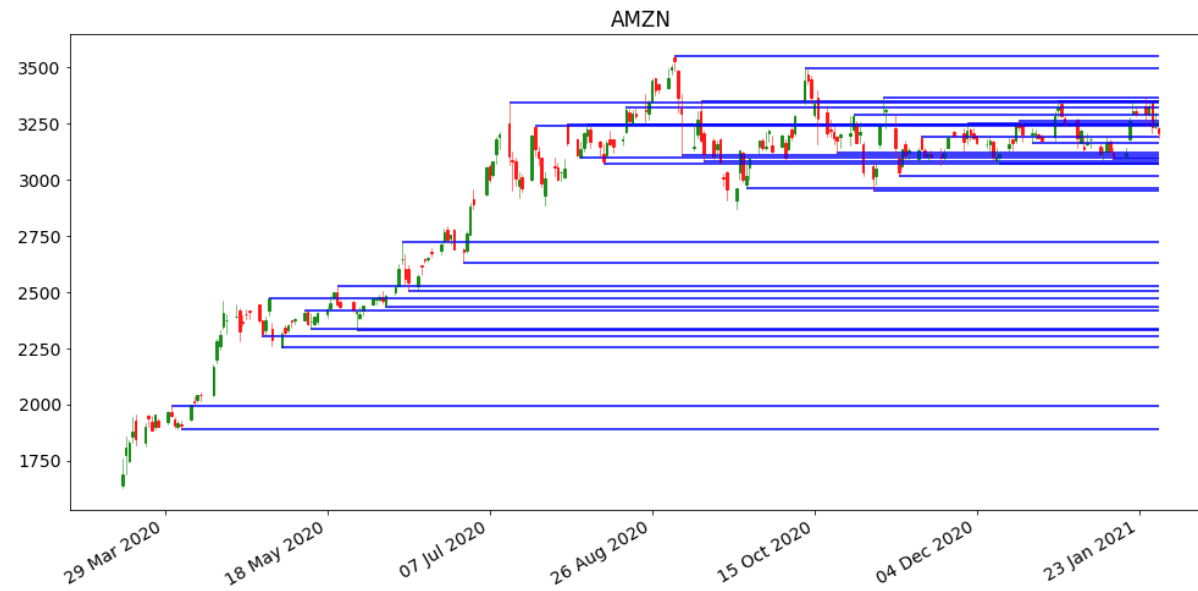
```

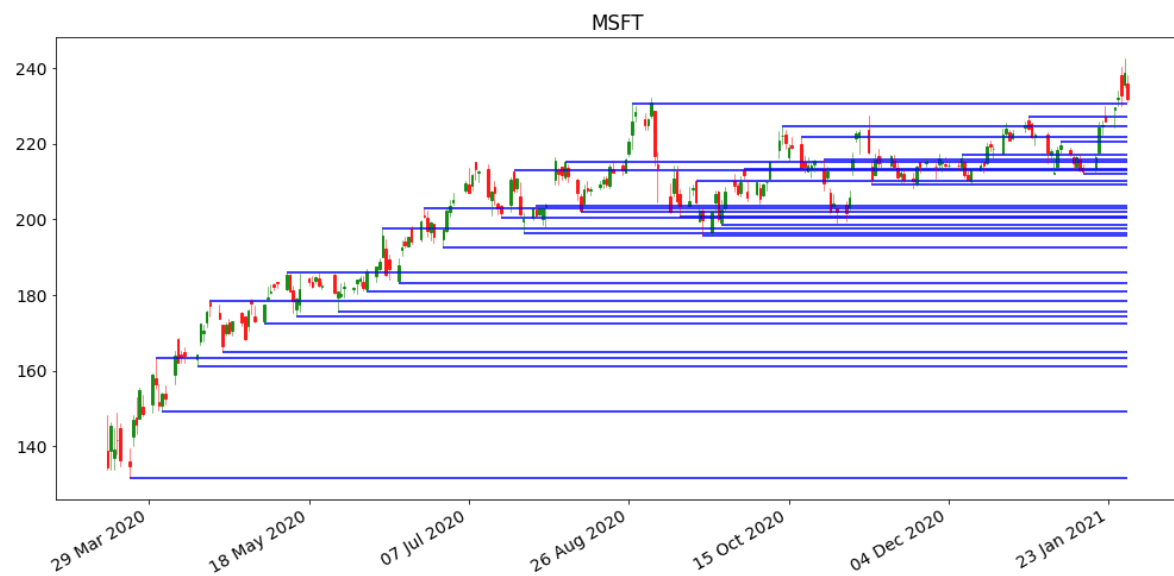
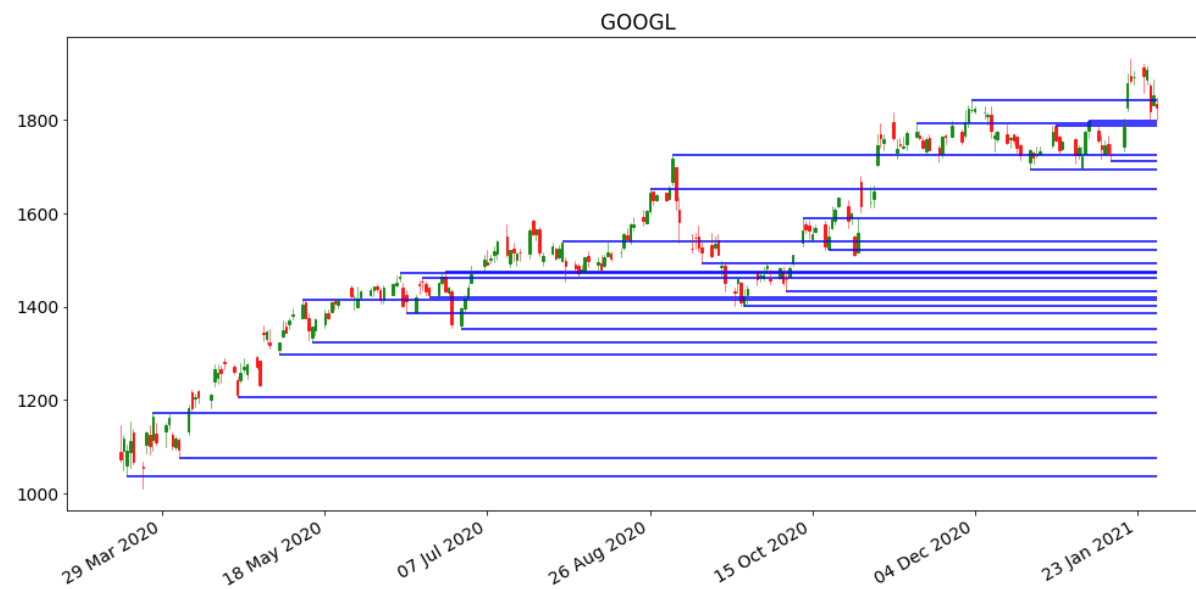
```
]],\
                                xmax=max(data_frames_separate[num]['Date']),colors='blue')
e')
```

```
In [10]: for i in range(len(num_name)):
          plot_all(i)
```

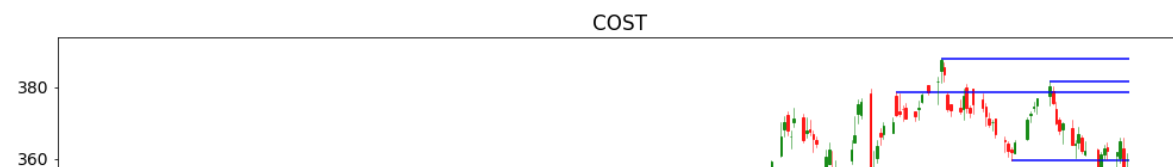
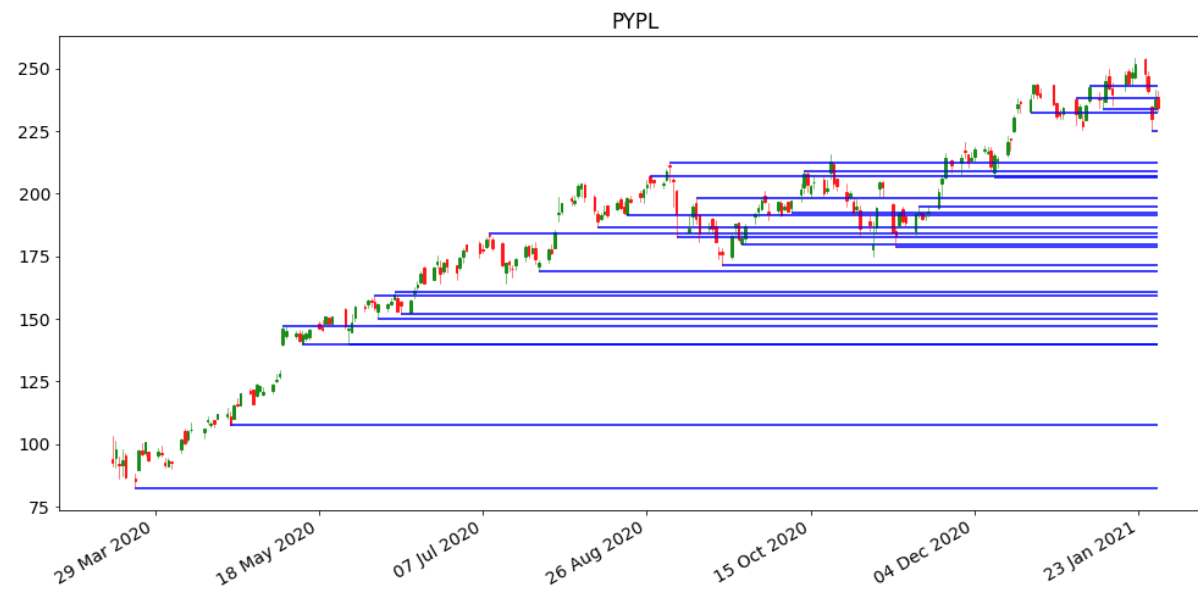
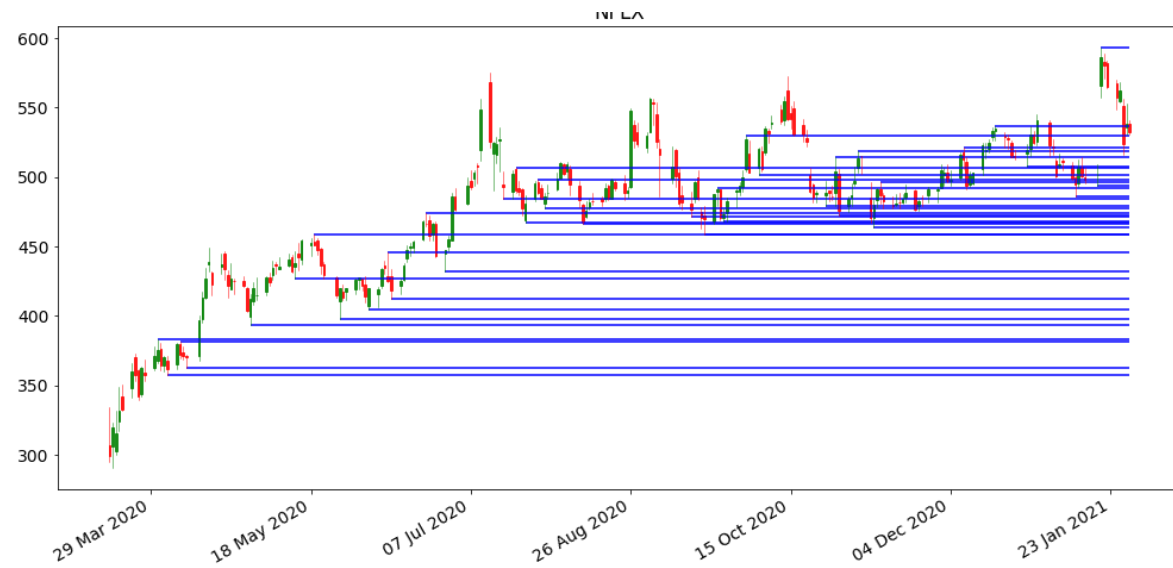


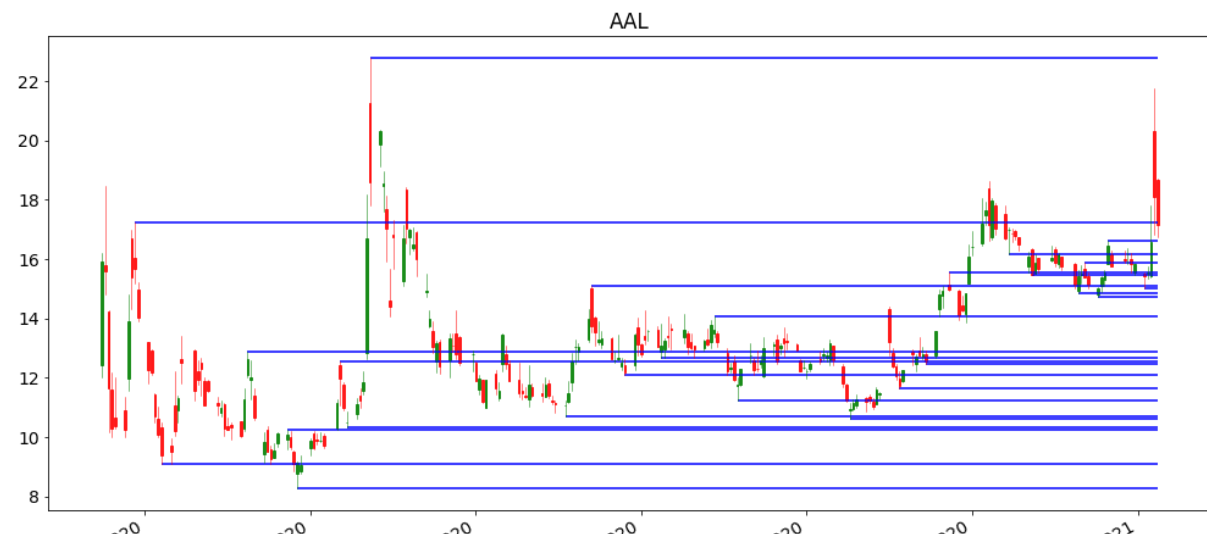
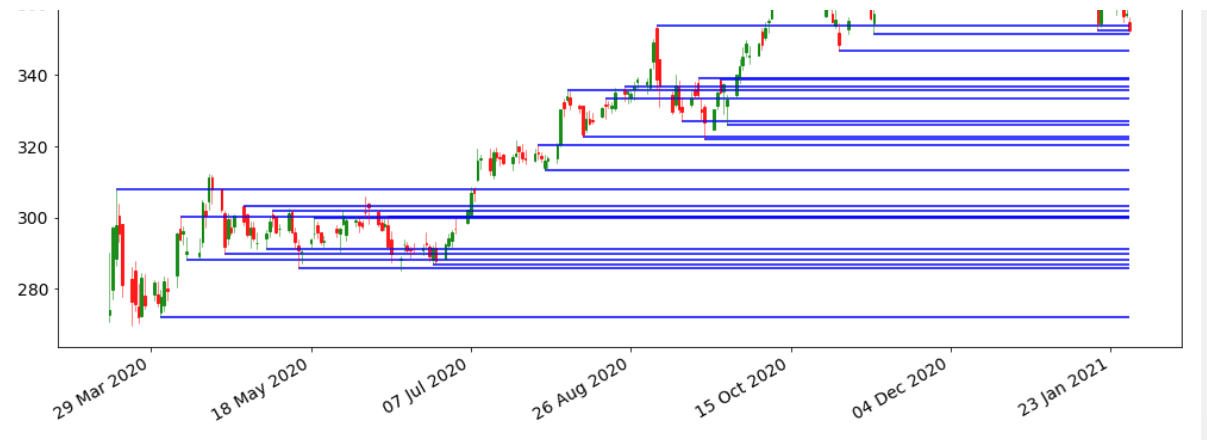






NFI Y







29 Mar 2022

18 May 2022

07 Jul 2022

26 Aug 2022

15 Oct 2022

04 Dec 2022

23 Jan 2023

```
In [11]: s=[]
         for num in range(len(num_name)):
             s.append(np.mean(data_frames_separate[num]['High'] - data_frames_separate[num]['Low']))
         s
```

```
Out[11]: [2.983300014289442,
          19.814532271376603,
          5.406140015381496,
          80.70508032446509,
          6.7394776902757245,
          38.77206833298142,
          5.073229853463499,
          16.859997173687358,
          6.043658213572459,
          6.184103050978104,
          0.9480630341950838]
```

```
In [12]: def isFarFromLevel(l,num):
         return np.sum([abs(l-x) < s[num] for x in levels[num]]) == 0
```

```
In [13]: len(levels)
```

```
Out[13]: 11
```

```

In [15]: levels = [[] for i in range(len(num_name))]
          for num in range(len(levels)):
              for i in range(2,df.shape[0]-2):
                  if isSupport(data_frames_separate[num],i):
                      l = data_frames_separate[num]['Low'][i]

                      if isFarFromLevel(l,num):
                          levels[num].append((i,l))

                  elif isResistance(data_frames_separate[num],i):
                      l = data_frames_separate[num]['High'][i]

                      if isFarFromLevel(l,num):
                          levels[num].append((i,l))

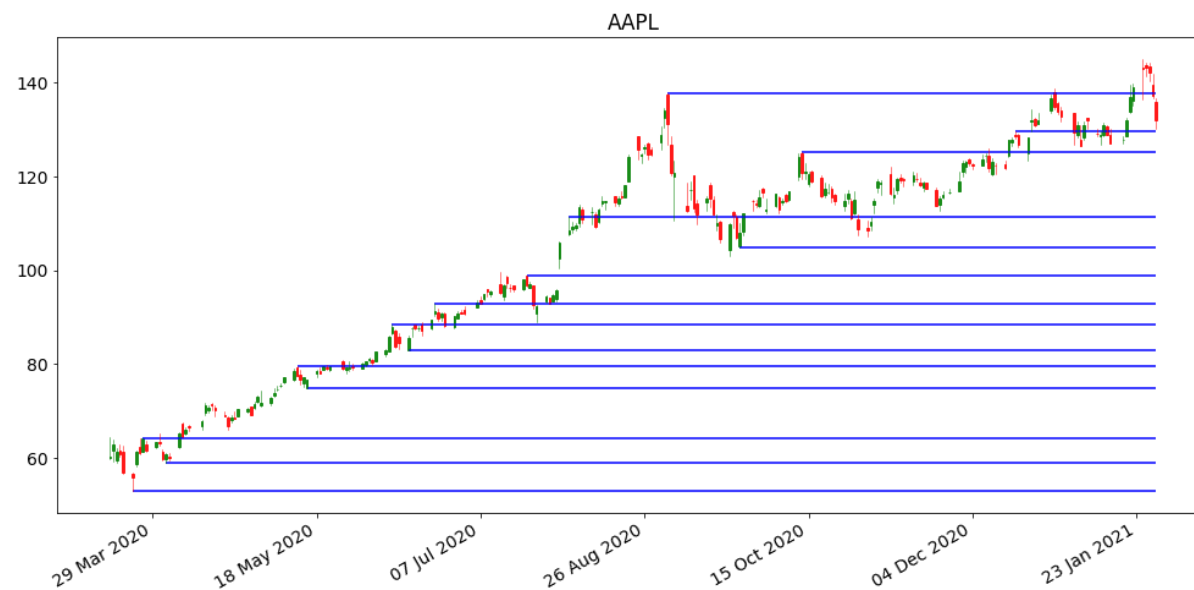
#levels

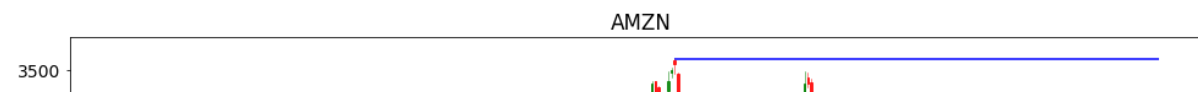
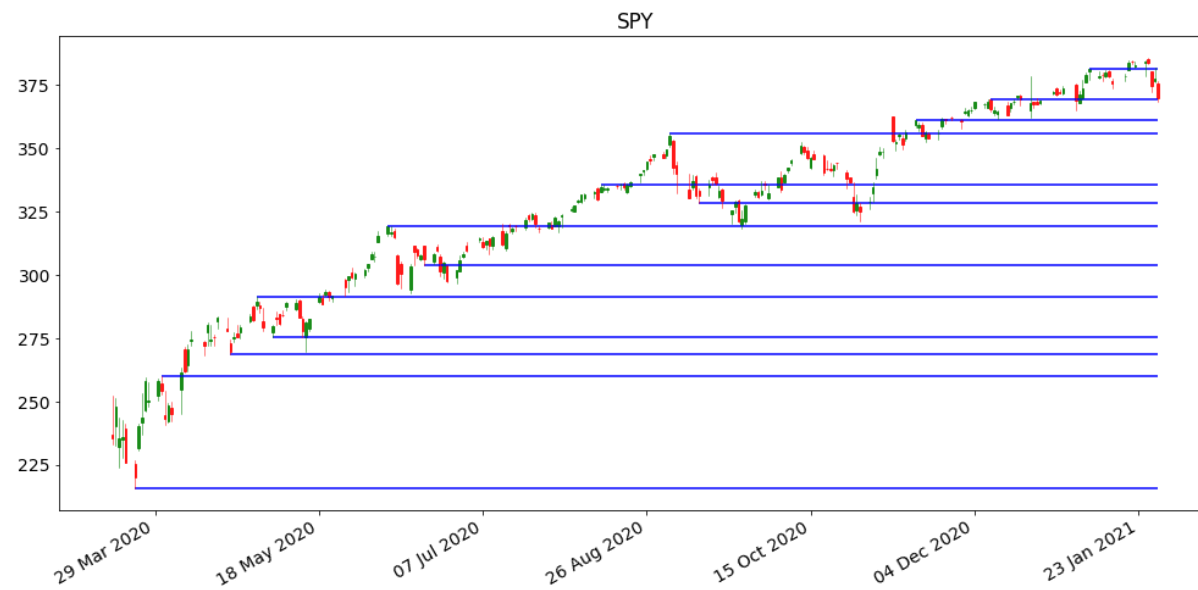
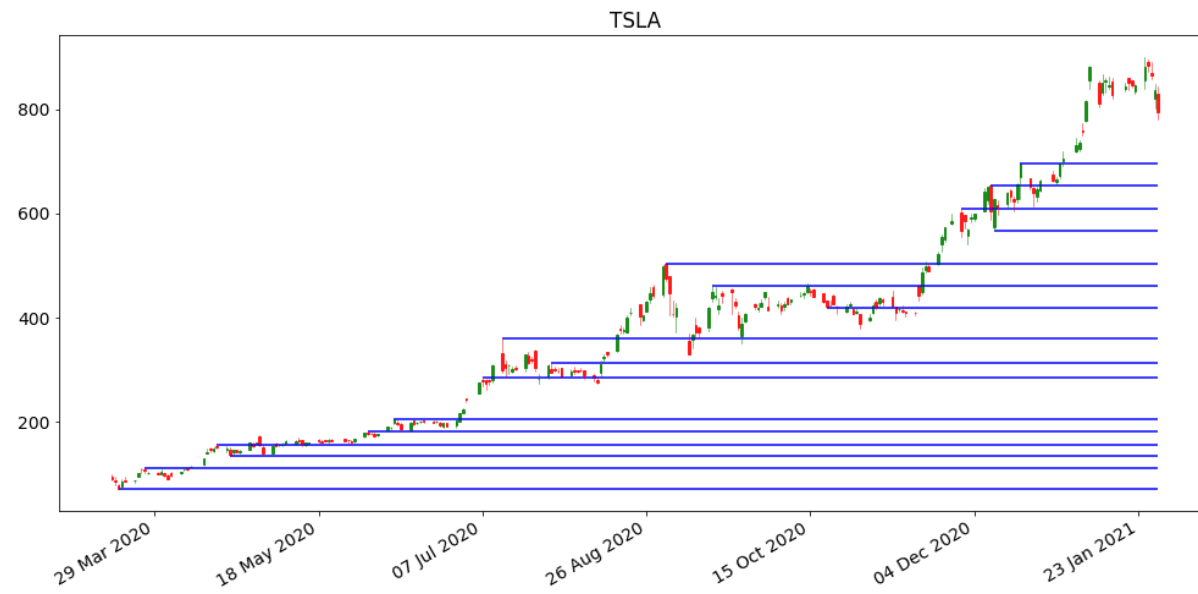
```

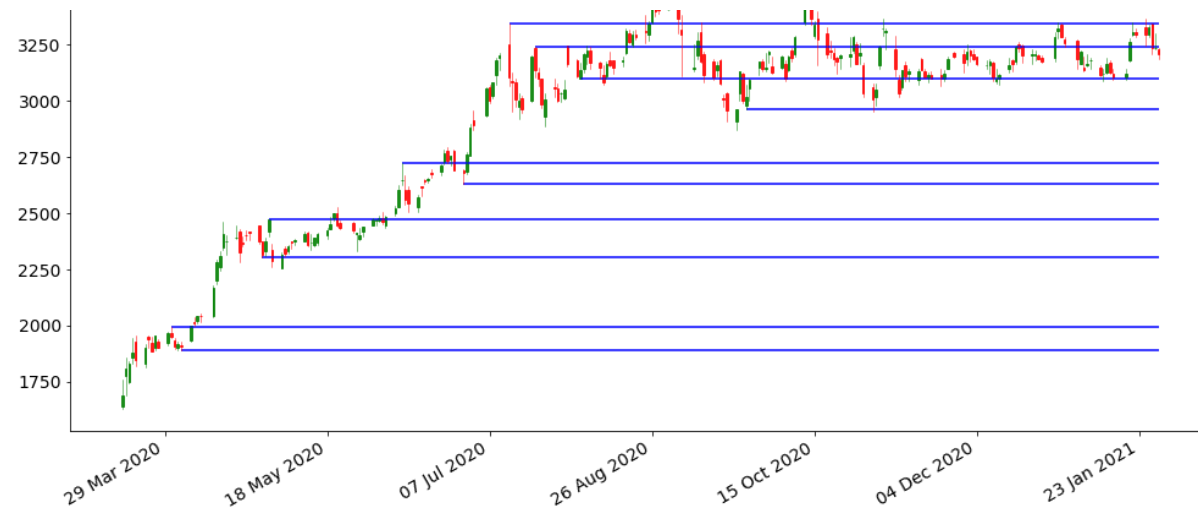
```

In [16]: for i in range(len(num_name)):
          plot_all(i)

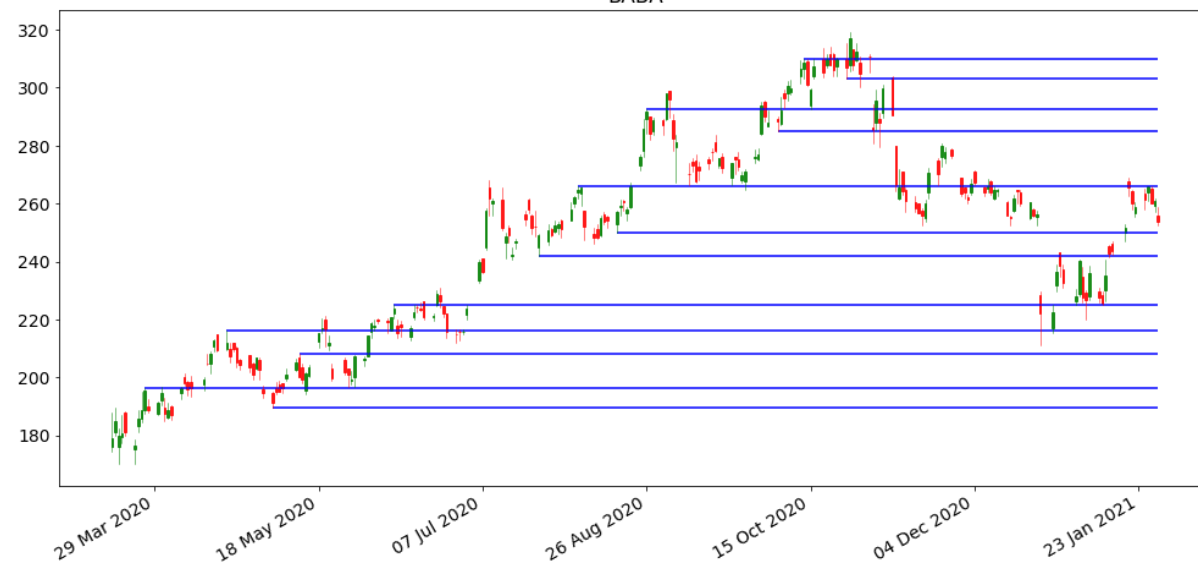
```



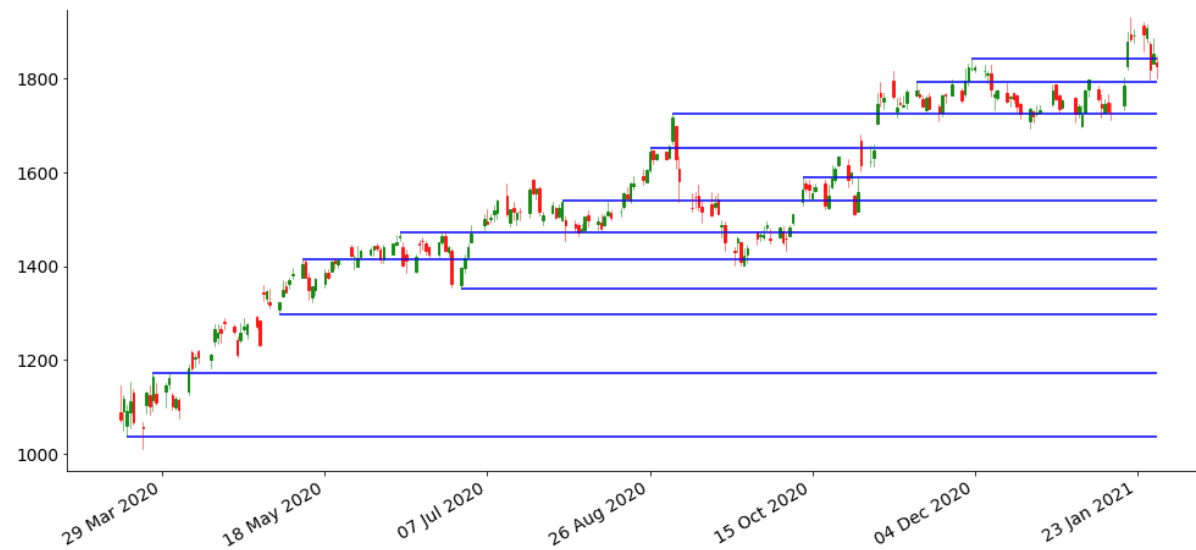




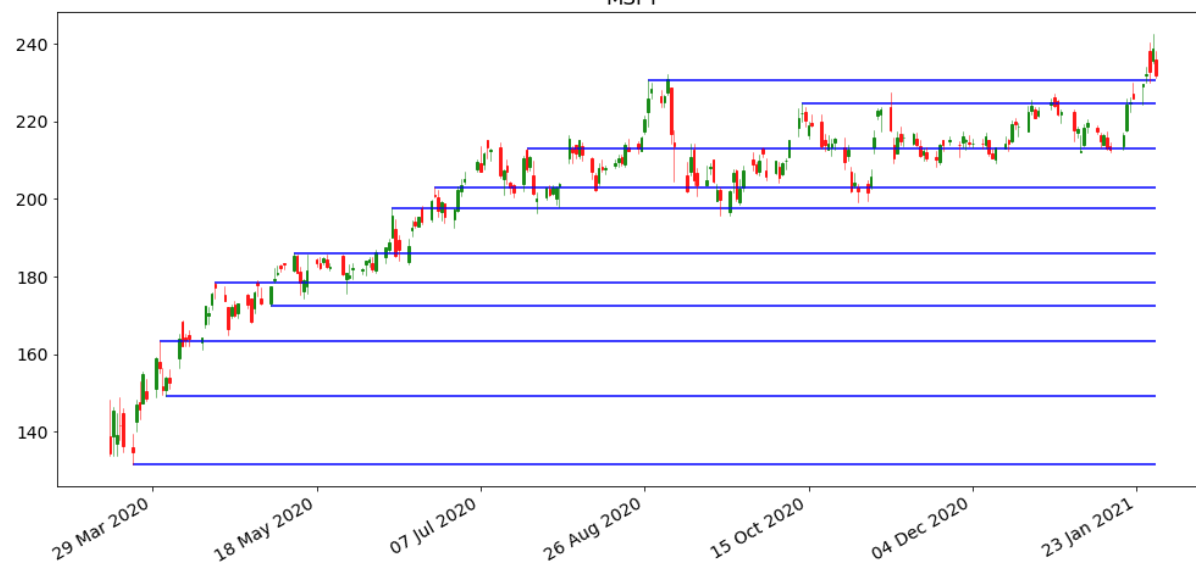
BABA



GOOGL

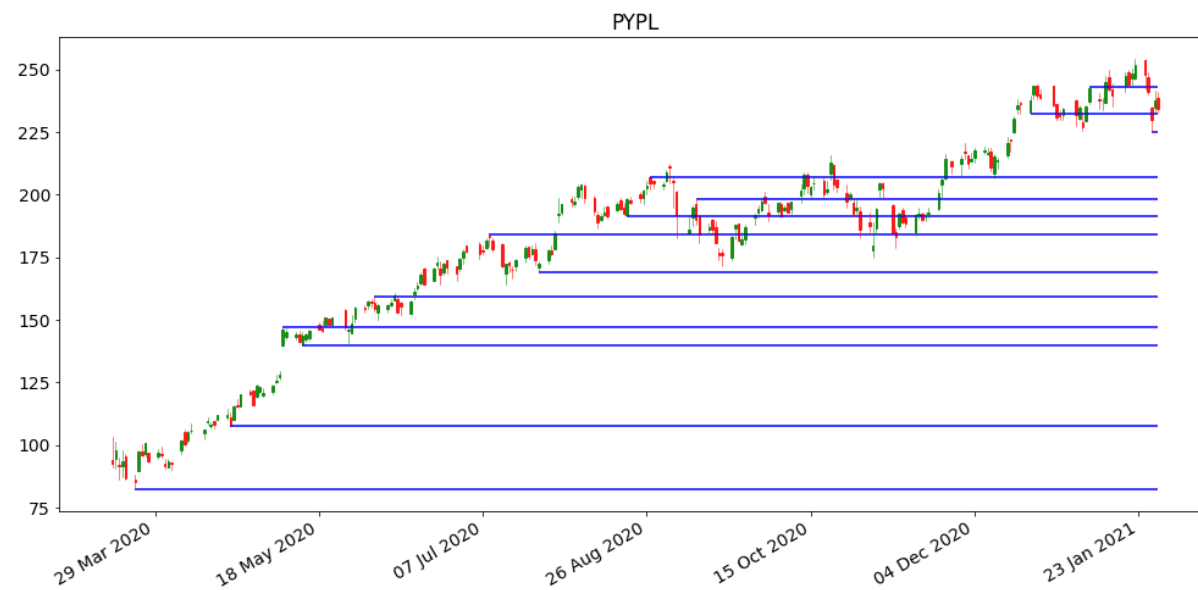
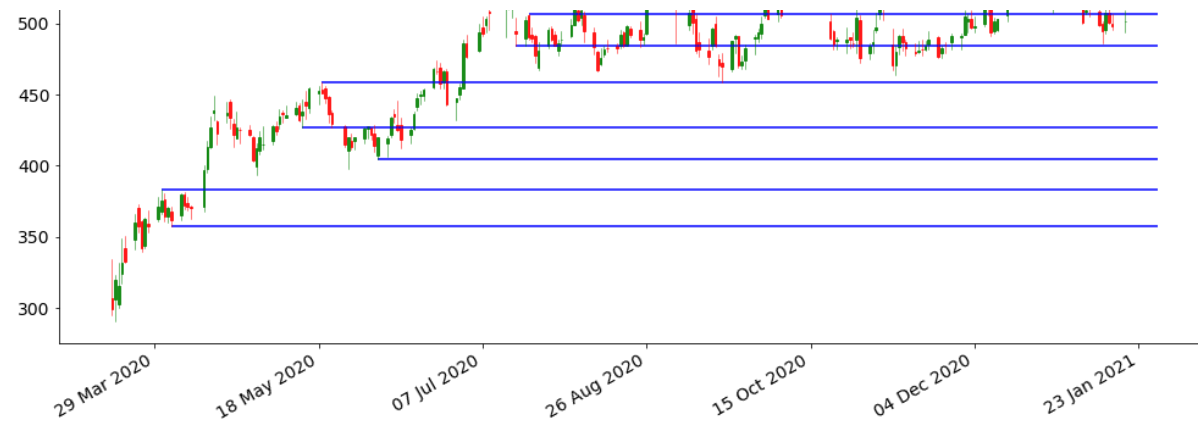


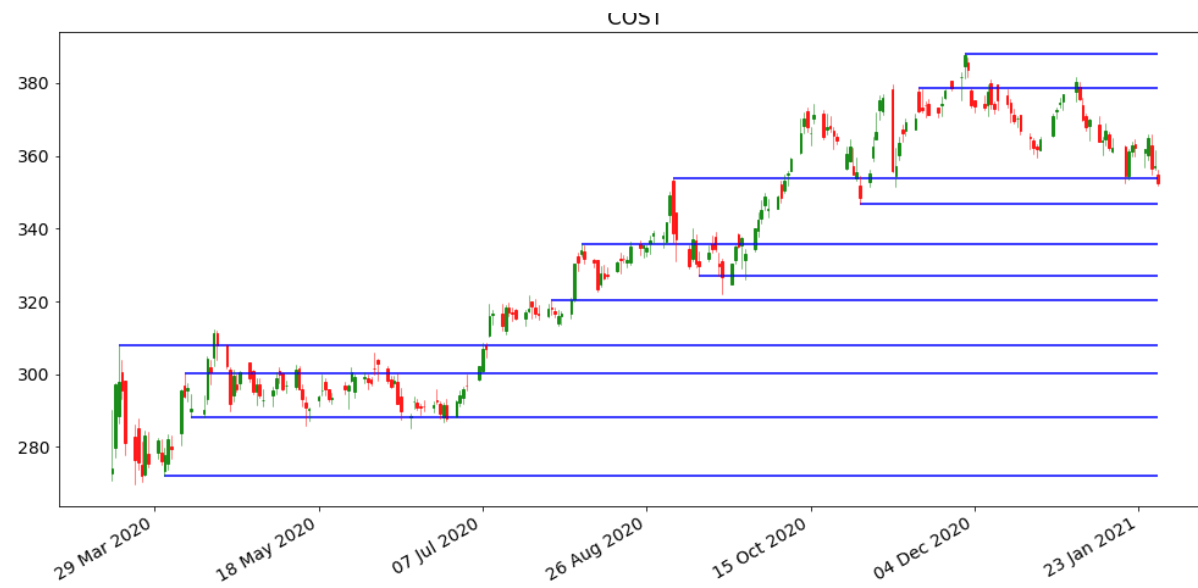
MSFT

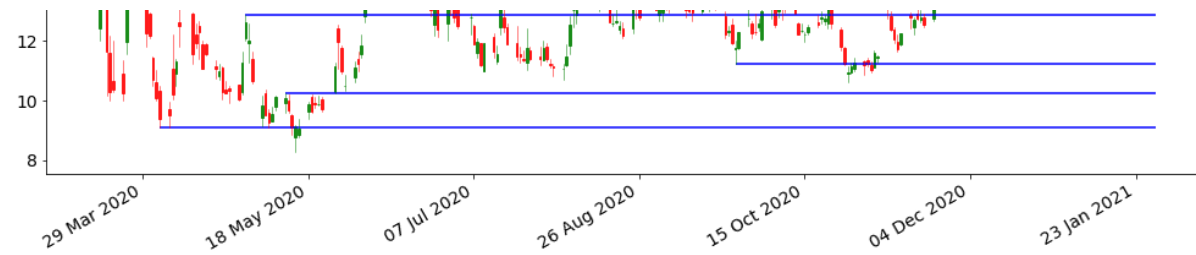


NFLX









In [ ]: