

```
In [2]: #!pip install yfinance
```

```
import yfinance as yf
import pandas as pd
symbol = 'AAPL'
df1 = yf.download(symbol, start='2020-01-01')
```

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

```
In [3]: df=df1.copy()
df.tail()
```

```
Out[3]:
```

	Open	High	Low	Close	Adj Close	Volume
<b>Date</b>						
<b>2021-12-30</b>	179.470001	180.570007	178.089996	178.199997	178.199997	59773000
<b>2021-12-31</b>	178.089996	179.229996	177.259995	177.570007	177.570007	64025500
<b>2022-01-03</b>	177.830002	182.880005	177.710007	182.009995	182.009995	104487900
<b>2022-01-04</b>	182.630005	182.940002	179.119995	179.699997	179.699997	99218200
<b>2022-01-05</b>	179.610001	180.149994	177.970001	179.014999	179.014999	32573080

```
In [4]: len(df)
```

```
Out[4]: 508
```

```
In [5]: range(len(df))
```

```
Out[5]: range(0, 508)
```

```
In [6]: data = []
data.append(df)
```

```
In [7]: data[0].head()
```

```
Out[7]:
```

	Open	High	Low	Close	Adj Close	Volume
<b>Date</b>						
<b>2020-01-02</b>	74.059998	75.150002	73.797501	75.087502	73.988480	135480400
<b>2020-01-03</b>	74.287498	75.144997	74.125000	74.357498	73.269142	146322800
<b>2020-01-06</b>	73.447502	74.989998	73.187500	74.949997	73.852989	118387200
<b>2020-01-07</b>	74.959999	75.224998	74.370003	74.597504	73.505653	108872000
<b>2020-01-08</b>	74.290001	76.110001	74.290001	75.797501	74.688080	132079200

```
In [8]: import numpy as np
delta = df['Close'].diff()
up = delta.clip(lower=0)
down = -1*delta.clip(upper=0)
ema_up = up.ewm(com=14, adjust=False).mean()
```

```
ema_down = down.ewm(com=14, adjust=False).mean()
rs = ema_up/ema_down
df['RSI'] = 100 - (100/(1 + rs))
df['AP'] = (df['Close'] + df['Low'] + df['High'])/3
df['std'] = df['AP'].rolling(20).std(ddof=0)
df['BBM'] = df['AP'].rolling(20).mean()
df['BBU'] = df['BBM'] + 2*df['std']
df['BBL'] = df['BBM'] - 2*df['std']
df['MA200'] = df['Close'].rolling(200).mean()
df['10dVol'] = df['Volume'].rolling(10).mean()
df['30dVol'] = df['Volume'].rolling(30).mean()
df['90dVol'] = df['Volume'].rolling(90).mean()
df['change']=df['Close'].pct_change().round(3)*100

df = df.iloc[21:]

buy_price = [row['Close'] if row['Close'] - row['BBL'] < 0 and row['RSI'] < 30
             else np.nan for index, row in df.iterrows()]
sell_price = [row['Close'] if row['Close'] - row['BBU'] > 0 and row['RSI'] > 80
             else np.nan for index, row in df.iterrows()]
```

In [9]: df.tail()

	Open	High	Low	Close	Adj Close	Volume	RSI	AP	std	
Date										
2021-12-30	179.470001	180.570007	178.089996	178.199997	178.199997	59773000	62.794404	178.953334	5.276169	173.
2021-12-31	178.089996	179.229996	177.259995	177.570007	177.570007	64025500	61.586181	178.019999	4.620220	174.
2022-01-03	177.830002	182.880005	177.710007	182.009995	182.009995	104487900	66.459329	180.866669	3.883393	175.
2022-01-04	182.630005	182.940002	179.119995	179.699997	179.699997	99218200	62.070011	180.586665	3.374690	176.
2022-01-05	179.610001	180.149994	177.970001	179.014999	179.014999	32573080	60.794323	179.044998	3.163883	176.

In [10]:  
##Buy signal  
df[(df['RSI'] <30) & (df['Close']< df['BBL'])]

	Open	High	Low	Close	Adj Close	Volume	RSI	AP	std	
Date										
2020-02-27	70.275002	71.500000	68.239998	68.379997	67.539062	320605600	27.169637	69.373332	3.274724	78.
2020-02-28	64.315002	69.602501	64.092499	68.339996	67.499550	426510000	27.119258	67.344999	4.030666	77.
2021-02-25	124.680000	126.459999	120.540001	120.989998	120.428375	148199500	29.894646	122.663333	4.636192	132.

In [11]:  
###Sell signals  
df[(df['RSI'] >70) & (df['Close']> df['BBU'])]

Out[11]:

	Open	High	Low	Close	Adj Close	Volume	RSI	AP	std	
Date										
2020-06-08	82.562500	83.400002	81.830002	83.364998	82.562653	95654400	71.015610	82.865000	1.603209	79.0
2020-06-09	83.035004	86.402496	83.002502	85.997498	85.169815	147712400	75.820579	85.134165	2.002506	79.0
2020-06-10	86.974998	88.692497	86.522499	88.209999	87.361023	166651600	78.961266	87.808332	2.633838	80.0
2020-07-08	94.180000	95.375000	94.089996	95.342499	94.424881	117092000	71.831232	94.935832	2.740248	89.0
2020-07-31	102.885002	106.415001	100.824997	106.260002	105.237312	374336800	73.342944	104.500000	2.568680	95.0
2020-08-03	108.199997	111.637497	107.892502	108.937500	107.889038	308151200	75.764037	109.489166	3.855824	96.0
2020-08-04	109.132500	110.790001	108.387497	109.665001	108.609535	173071600	76.388338	109.614166	4.702275	97.0
2020-08-05	109.377502	110.392502	108.897499	110.062500	109.003220	121776800	76.739110	109.784167	5.365843	98.0
2020-08-06	110.404999	114.412498	109.797501	113.902496	112.806252	202428800	79.839139	112.704165	6.179025	99.0
2020-09-01	132.759995	134.800003	130.529999	134.179993	133.128189	151948100	81.421859	133.169998	6.879644	118.0
2020-12-28	133.990005	137.339996	133.509995	136.690002	135.852524	124486200	72.693602	135.846664	4.331476	125.0
2021-07-02	137.899994	140.000000	137.750000	139.960007	139.546967	78852600	74.276418	139.236669	3.762146	131.0
2021-07-06	140.070007	143.149994	140.070007	142.020004	141.600876	108181800	77.056773	141.746668	4.102213	132.0
2021-07-07	143.539993	144.889999	142.660004	144.570007	144.143356	104911600	79.933378	144.040003	4.627727	133.0
2021-07-09	142.750000	145.649994	142.649994	145.110001	144.681763	99890800	76.971075	144.469996	5.022961	134.0
2021-07-14	148.100006	149.570007	147.679993	149.149994	148.709824	127050800	79.894614	148.799998	5.685024	137.0
2021-11-18	153.710007	158.669998	153.050003	157.869995	157.869995	137827700	70.310084	156.529999	1.882444	150.0
2021-11-19	157.649994	161.020004	156.529999	160.550003	160.550003	117305600	73.758353	159.366669	2.688603	150.0
2021-11-22	161.679993	165.699997	161.000000	161.020004	161.020004	117467900	74.318799	162.573334	3.641295	151.0
2021-11-23	161.119995	161.800003	159.059998	161.410004	161.410004	96041900	74.797338	160.756668	4.118115	152.0
2021-11-24	160.750000	162.139999	159.639999	161.940002	161.940002	69463600	75.463065	161.240000	4.495897	152.0
2021-11-30	159.990005	165.520004	159.919998	165.300003	165.300003	174048100	70.831284	163.580002	5.077431	154.0



```

    ), row=1, col=1)
fig.add_trace(go.Scatter(x=df.index,
                        y=df['BBL'],name="BBL",
                        line=dict(color='red', width=2)
                        ), row=1, col=1)
fig.add_trace(go.Scatter(x=df.index,
                        y=df['BBM'],name="BBM",
                        line=dict(color='grey', width=2)
                        ), row=1, col=1)
fig.add_trace(go.Scatter(x=df.index,
                        y=df['Close'],name="Close",
                        line=dict(color='blue', width=2)
                        ), row=1, col=1)
fig.add_trace(go.Scatter(x=df.index,
                        y=df['change'],name="Change%",
                        line=dict(color='yellow', width=2)
                        ), row=1, col=1)
fig.add_trace(go.Scatter(x=df.index,
                        y=buy_price,name="BUY",
                        mode='markers',marker=dict(color='green'),marker_symbol='triangle-up',
                        marker_line_width=2, marker_size=12
                        ), row=1, col=1)

fig.add_trace(go.Scatter(x=df.index,
                        y=sell_price,name="SELL",
                        mode='markers',marker=dict(color='red'),marker_symbol='triangle-down',
                        marker_line_width=2, marker_size=12
                        ), row=1, col=1)
# Plot volume trace on 2nd row
colors = ['green' if row['Open'] - row['Close'] >= 0
          else 'red' for index, row in df.iterrows()]

fig.add_trace(go.Bar(x=df.index,
                    y=df['Volume'],name="Volume",
                    marker_color=colors
                    ), row=2, col=1)
# Plot MACD trace on 3rd row

fig.add_trace(go.Scatter(x=df.index,
                        y=df['RSI'],name="RSI",
                        line=dict(color='red', width=2)
                        ), row=3, col=1)
fig.add_shape(type='line',
              x0=df.index.min(),
              y0=30,
              x1=df.index.max(),
              y1=30,
              line=dict(color='Red',),
              xref='x',
              yref='y', row=3, col=1
              )
fig.add_shape(type='line',
              x0=df.index.min(),
              y0=70,
              x1=df.index.max(),
              y1=70,
              line=dict(color='Green',),
              xref='x',
              yref='y', row=3, col=1
              )

fig.add_trace(go.Bar(x=df.index,
                    y=df['Volume'],name="Volume",

```

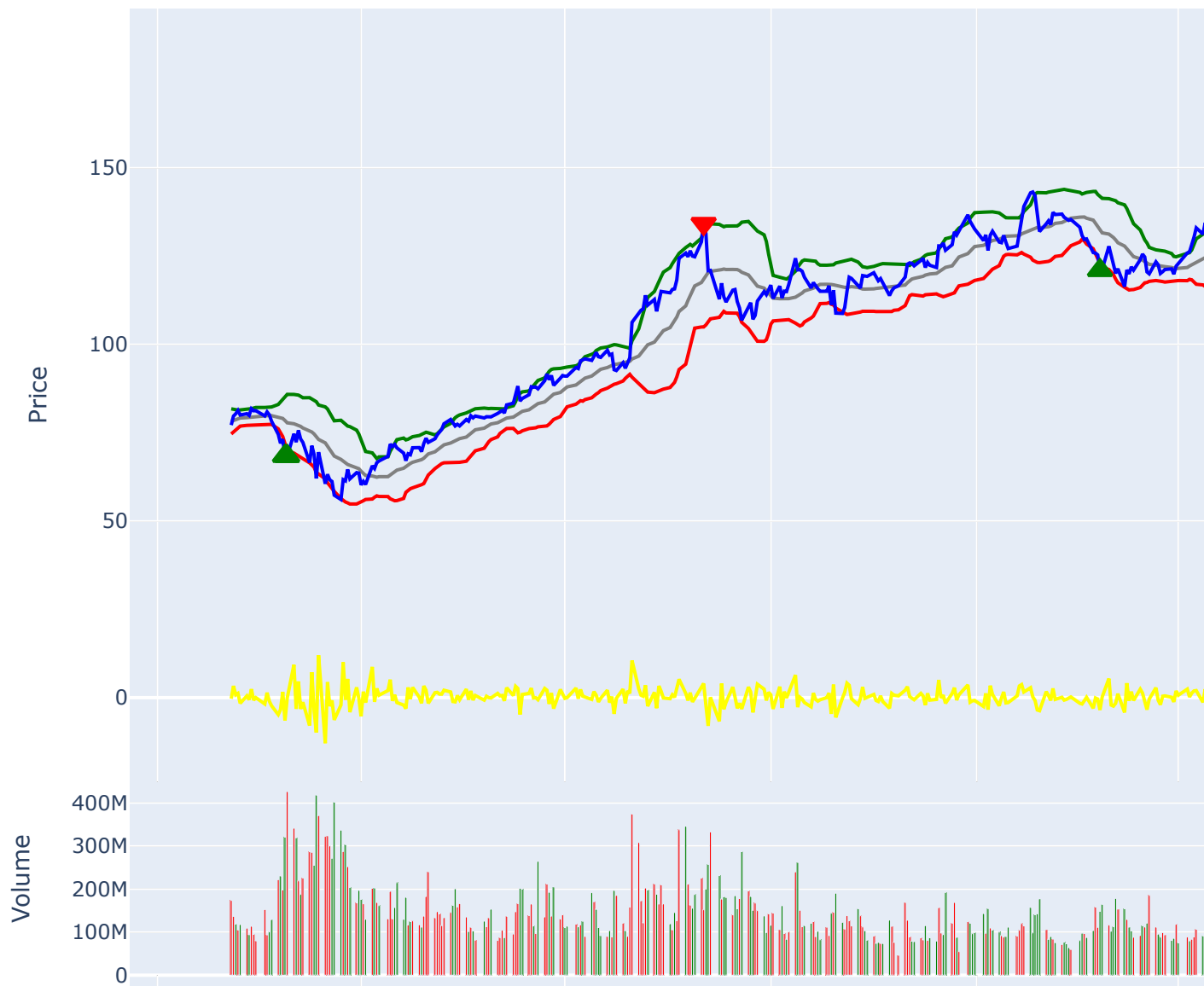
```

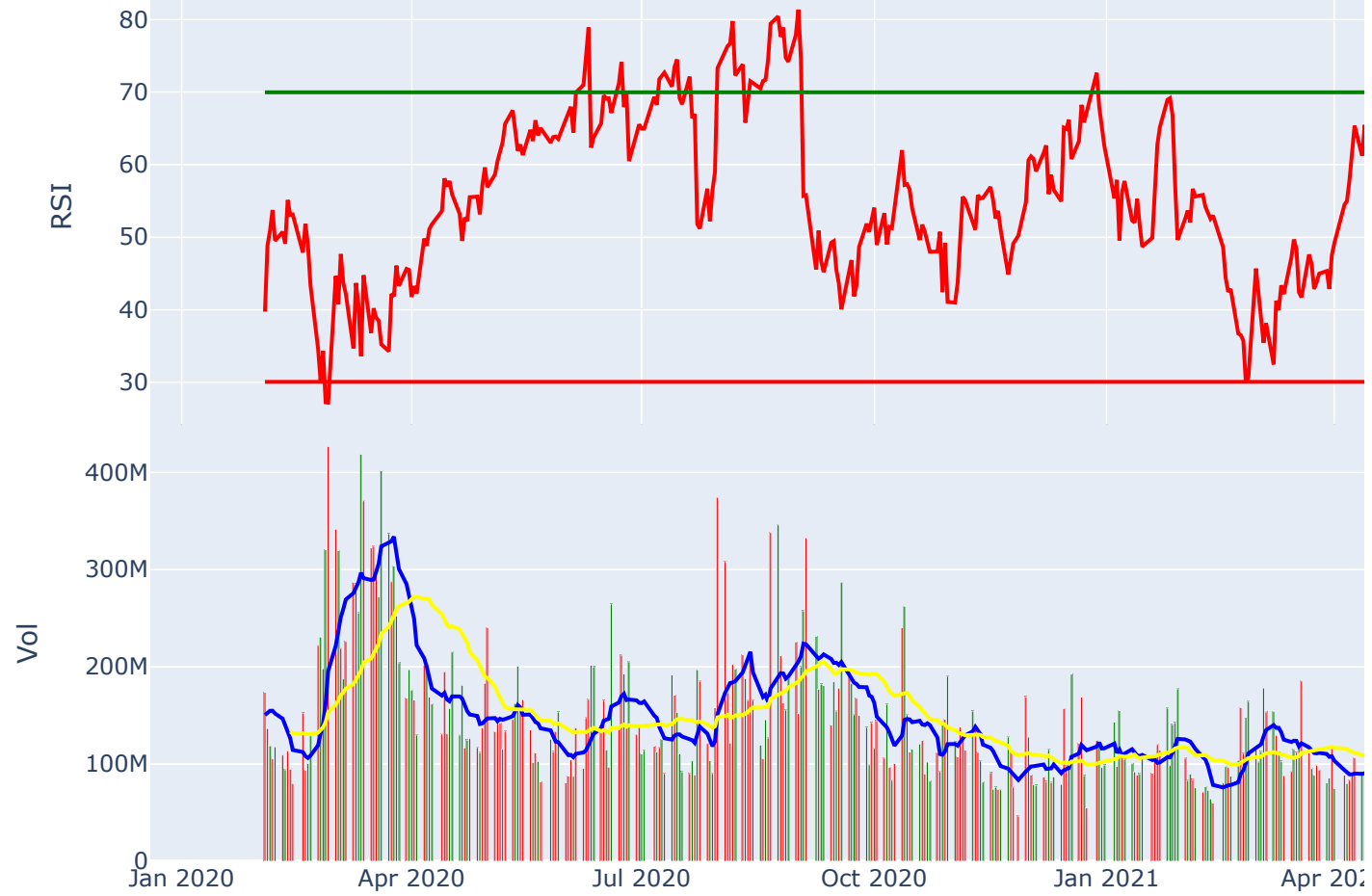
        marker_color=colors
    ), row=4, col=1)
fig.add_trace(go.Scatter(x=df.index,
                        y=df['10dVol'],name="10dVol",
                        line=dict(color='blue', width=2)
                        ), row=4, col=1)
fig.add_trace(go.Scatter(x=df.index,
                        y=df['30dVol'],name="30dVol",
                        line=dict(color='yellow', width=2)
                        ), row=4, col=1)

# update y-axis label
fig.update_yaxes(title_text="Price", row=1, col=1)
fig.update_yaxes(title_text="Volume", row=2, col=1)
fig.update_yaxes(title_text="RSI", row=3, col=1)
fig.update_yaxes(title_text="Vol", row=4, col=1)

# update layout by changing the plot size, hiding legends & rangeslider, and removing
fig.update_layout(height=1200, width=1200,
                  showlegend=False, hovermode="x unified",
                  xaxis_rangeslider_visible=False)

```





In [ ]: