

In [9]:

```
#Basic Python Packages
import pandas as pd
import numpy as np
import datetime as dt

#Technical Trading and Finance
import pandas_ta as ta
import yfinance as yf
import talib
import mplfinance as mpf

#Data Visualization
import plotly.express as px
import plotly.graph_objects as go
import panel as pn
pn.extension('plotly')
import hvplot.pandas
import matplotlib.pyplot as plt
from plotly.subplots import make_subplots
```

In [12]:

```
start = dt.datetime.today()-dt.timedelta(90)
end = dt.datetime.today()
ticker = 'MSFT'
inter = '1mo'
```

In [13]:

```
df = yf.download(ticker, start, end, inter)
```

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

In [14]:

```
def momentum(df):
    slowk, slowd = talib.STOCH(df['High'], df['Low'], df['Adj Close'], fastk_period=5, slowk_period=3, slo
df['slowk'] = slowk
df['slowd'] = slowd
df['RSI'] = talib.RSI(df['Adj Close'], timeperiod=14)
df['ROCR'] = talib.ROCR(df['Adj Close'], timeperiod=10)
macd, macdsignal, macdhist = talib.MACD(df['Adj Close'], fastperiod=12, slowperiod=26, signalperiod=9)
df['macd'] = macd
df['macdsignal'] = macdsignal
df['macdhist'] = macdhist
return df
```

In [15]:

```
data = momentum(df)
data['price_max'] = data['Adj Close'].rolling(5).max()
data['rsi_max'] = data['RSI'].rolling(5).max()
data['price_min'] = data['Adj Close'].rolling(5).min()
data['rsi_min'] = data['RSI'].rolling(5).min()
data
```

Out[15]:

	Open	High	Low	Close	Adj Close	Volume	Dividends	Stock Splits	slowk	slowd	
Date											
2022-03-14	280.339996	285.399994	275.820007	276.440002	275.797638	30660700	0.0	0	NaN	NaN	
2022-03-15	280.350006	287.820007	278.730011	287.149994	286.482758	34245100	0.0	0	NaN	NaN	
2022-03-16	289.109985	294.570007	283.200012	294.390015	293.705933	37826300	0.0	0	NaN	NaN	
2022-03-17	293.290009	295.609985	289.369995	295.220001	294.533997	30816600	0.0	0	NaN	NaN	
2022-03-18	295.369995	301.000000	292.730011	300.429993	299.731873	43390600	0.0	0	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	
2022-06-06	272.059998	274.179993	267.220001	268.750000	268.750000	22400300	0.0	0	59.213760	69.646619	49.57
2022-06-07	266.640015	273.130005	265.940002	272.500000	272.500000	22860700	0.0	0	54.837325	61.080531	52.72
2022-06-08	271.709991	273.000000	269.609985	270.410004	270.410004	17372300	0.0	0	59.897017	57.982701	50.81
2022-06-09	267.779999	272.709991	264.630005	264.790009	264.790009	26439700	0.0	0	45.642991	53.459111	46.00
2022-06-10	260.579987	260.579987	252.529999	252.990005	252.990005	31422800	0.0	0	23.769933	43.103314	37.88

63 rows x 19 columns

In [16]:

```

# Stochastics panel

slowk_plot = mpf.make_addplot(data["slowk"], panel=1, color='blue', title="Stochastic Oscillators")
slowd_plot = mpf.make_addplot(data["slowd"], panel=1, color='red')
# ax[1].axhline(80, color="tab:red", ls="--")
# ax[1].axhline(20, color="tab:green", ls="--")

# plot
plots_so = [slowk_plot, slowd_plot]
fig, axes = mpf.plot(df, type='candle', addplot=plots_so, figscale=1.5, figratio=(12, 8), title=f"\n{ticker}",
                    style='yahoo', volume=True, volume_panel=2, panel_ratios=(6, 3, 2), returnfig=True)

```

/Users/satishsurath/opt/anaconda3/envs/ibkr/lib/python3.9/site-packages/matplotlib/cbook/\_\_init\_\_.py:1377: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.

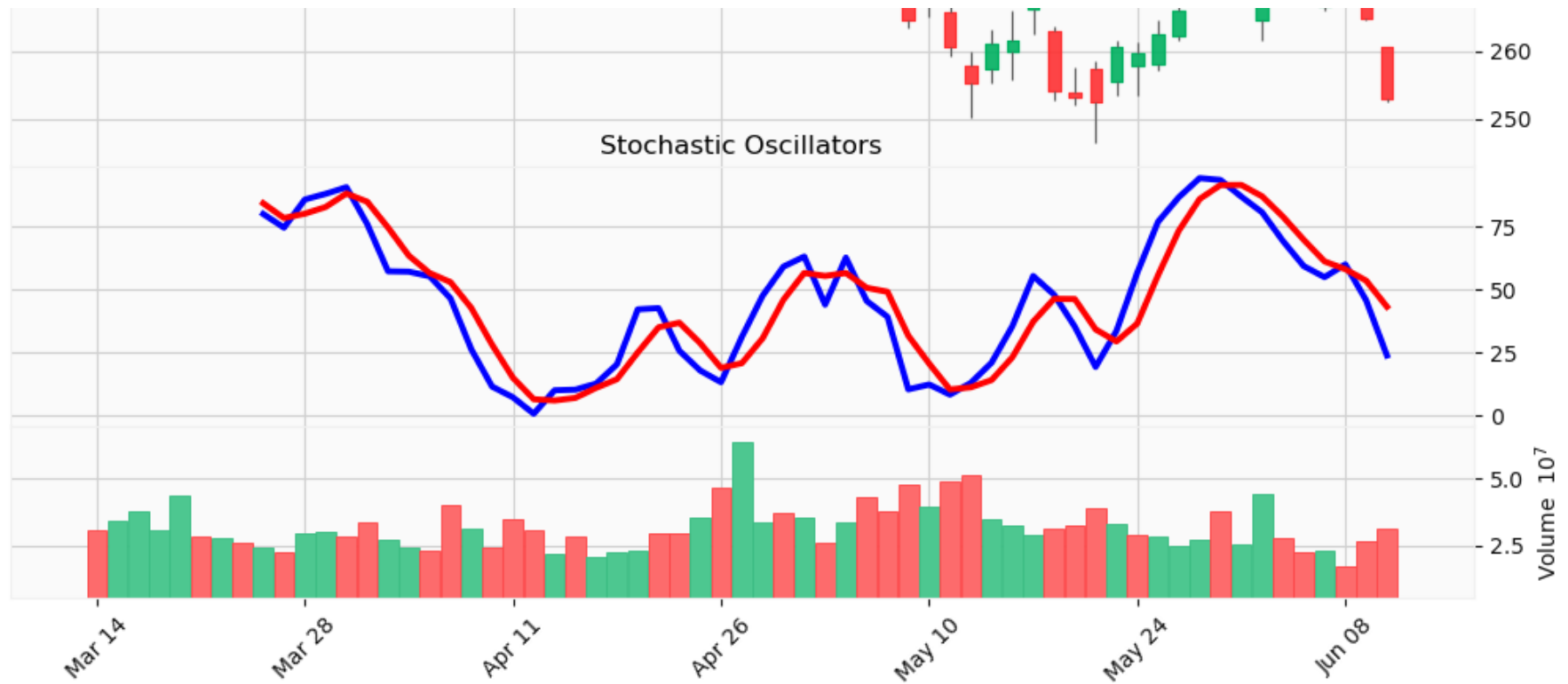
x[:, None]

/Users/satishsurath/opt/anaconda3/envs/ibkr/lib/python3.9/site-packages/matplotlib/axes/\_base.py:239: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.

y = y[:, np.newaxis]

## MSFT





In [17]:

```
# RSI panel

rsi_plot = mpf.make_addplot(data["RSI"], panel=1, color='blue', title="RSI")

# plot
rsi_p = [rsi_plot]
fig, axes = mpf.plot(df, type='candle', addplot=rsi_p, figscale=1.5, figratio=(12,8), title=f"\n{ticker}",
                    style='yahoo', volume=True, volume_panel=2, panel_ratios=(6,3,2), returnfig=True)
```

```
/Users/satishsurath/opt/anaconda3/envs/ibkr/lib/python3.9/site-packages/matplotlib/cbook/__init__.py:1377:
FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.
```

```
x[:, None]
```

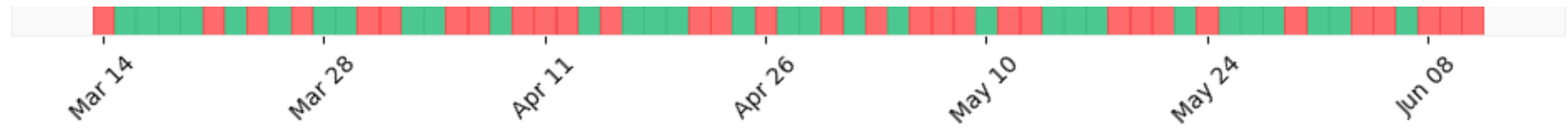
```
/Users/satishsurath/opt/anaconda3/envs/ibkr/lib/python3.9/site-packages/matplotlib/axes/_base.py:239: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.
```

reWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.

```
y = y[:, np.newaxis]
```

## MSFT





In [18]:

```
# macd panel
colors = ['g' if v >= 0 else 'r' for v in data["macdhist"]]
macd_plot = mpf.make_addplot(data["macd"], panel=1, color='blue', title="MACD")
macd_hist_plot = mpf.make_addplot(data["macdhist"], type='bar', panel=1, color=colors) # color='dimgray'
macd_signal_plot = mpf.make_addplot(data["macdsignal"], panel=1, color='red')

# plot
plots = [macd_plot, macd_signal_plot, macd_hist_plot]
mpf.plot(data, type='candle', style='yahoo', figratio=(16,8), addplot=plots, title=f"\n{ticker}", volume=T
```

```
/Users/satishsurath/opt/anaconda3/envs/ibkr/lib/python3.9/site-packages/matplotlib/cbook/__init__.py:1377:
FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.
```

```
x[:, None]
```

```
/Users/satishsurath/opt/anaconda3/envs/ibkr/lib/python3.9/site-packages/matplotlib/axes/_base.py:239: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.
```

```
y = y[:, np.newaxis]
```

## MSFT





In [19]:

```

# ROCR panel

rocr_plot = mpf.make_addplot(data["ROCR"], panel=1, color='blue', title="Rate of Change")

# plot
rocr_p = [rocr_plot]
fig, axes = mpf.plot(df, type='candle', addplot=rocr_p, figscale=1.5, figratio=(12, 8), title=f"\n{ticker}",
                    style='yahoo', volume=True, volume_panel=2, panel_ratios=(6, 3, 2), returnfig=True)
# Configure chart legend and title
# axes[0].legend([None]*(len(rocr_p)+2))
# handles = axes[0].get_legend().legendHandles
# axes[0].legend(handles=handles[2:], labels=list(rocr_p.keys()))

```

/Users/satishsurath/opt/anaconda3/envs/ibkr/lib/python3.9/site-packages/matplotlib/cbook/\_\_init\_\_.py:1377: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.

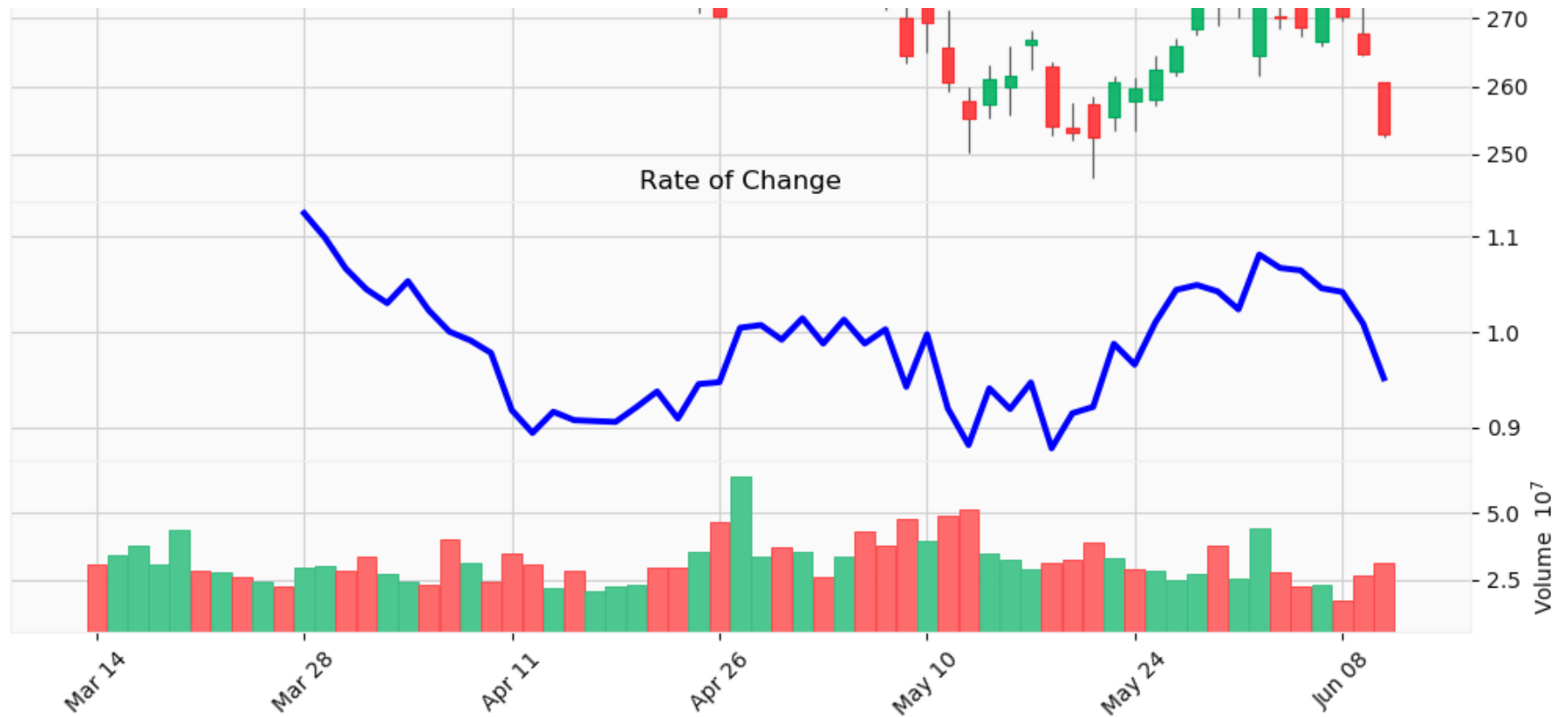
```
x[:, None]
```

/Users/satishsurath/opt/anaconda3/envs/ibkr/lib/python3.9/site-packages/matplotlib/axes/\_base.py:239: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version. Convert to a numpy array before indexing instead.

```
y = y[:, np.newaxis]
```

## MSFT





In [20]:

```

dashboard_title = pn.Column(
    '## Momentum Indicators'
)

# Define a welcome text
# welcome_message = ''
# The purpose of this project is to identify correlations between variables like annual income, interest rate
# ''

# MACD
# Construct a 2 x 1 Plotly figure
fig = make_subplots(rows=2, cols=1)
# Candlestick chart for pricing

```

```

fig.append_trace(
    go.Candlestick(
        x=data.index,
        open=data['Open'],
        high=data['High'],
        low=data['Low'],
        close=data['Adj Close'],
        increasing_line_color='green',
        decreasing_line_color='red',
        showlegend=False
    ), row=1, col=1
)
# Fast Signal (%k)
fig.append_trace(
    go.Scatter(
        x=data.index,
        y=data['macd'],
        line=dict(color='#ff9900', width=2),
        name='macd',
        # showlegend=False,
        legendgroup='2',
    ), row=2, col=1
)
# Slow signal (%d)
fig.append_trace(
    go.Scatter(
        x=data.index,
        y=data['macdsignal'],
        line=dict(color='#000000', width=2),
        # showlegend=False,
        legendgroup='2',
        name='signal'
    ), row=2, col=1
)
# Colorize the histogram values
colors = np.where(data['macdhist'] < 0, 'red', 'green')
# Plot the histogram
fig.append_trace(
    go.Bar(

```

```

        x=data.index,
        y=data['macdhist'],
        name='histogram',
        marker_color=colors,
    ), row=2, col=1
)
# Make it pretty
layout = go.Layout(
    height=1000, width=1200,
    plot_bgcolor='#efefef',
    # Font Families
    font_family='Monospace',
    font_color='#000000',
    font_size=20,
    xaxis=dict(
        rangeslider=dict(
            visible=False
        )
    )
)

# Update options and show plot
fig.update_layout(layout)
# Create Columns

MACD = pn.Column(pn.Column(fig))

# rate_dist = pn.Column(pn.Column(fig4,fig5))
# revol = pn.Column(fig6)

```

In [21]:

```

# RSI

# Candlestick chart for pricing
fig_rsi= make_subplots(rows=2, cols=1)

fig_rsi.append_trace(
    go.Candlestick(
        x=data.index,

```

```

        open=data['Open'],
        high=data['High'],
        low=data['Low'],
        close=data['Adj Close'],
        increasing_line_color='green',
        decreasing_line_color='red',
        showlegend=False
    ), row=1, col=1
)

fig_rsi.append_trace(go.Scatter(x=data.index, y=data['RSI'], name='RSI',
                               line = dict(color='green', width=4)), row = 2, col = 1)

# Make it pretty
layout = go.Layout(
    height=1000, width=1200,
    plot_bgcolor='#efefef',
    # Font Families
    font_family='Monospace',
    font_color='#000000',
    font_size=20,
    xaxis=dict(
        rangelslider=dict(
            visible=False
        )
    )
)

# Update options and show plot
fig_rsi.update_layout(layout)
# Create Columns
RSI_ind = pn.Column(fig_rsi)

```

In [22]:

```

# Stochastic Oscillator

# Candlestick chart for pricing
fig_so= make_subplots(rows=2, cols=1)

```

```

fig_so.append_trace(
    go.Candlestick(
        x=data.index,
        open=data['Open'],
        high=data['High'],
        low=data['Low'],
        close=data['Adj Close'],
        increasing_line_color='green',
        decreasing_line_color='red',
        showlegend=False
    ), row=1, col=1
)

fig_so.append_trace(go.Scatter(x=data.index, y=data['slowk'], name='K',
                               line = dict(color='blue', width=4)), row = 2, col = 1)
fig_so.append_trace(go.Scatter(x=data.index, y=data['slowd'], name='D',
                               line = dict(color='red', width=4)), row = 2, col = 1)

# Make it pretty
layout = go.Layout(
    height=1000, width=1200,
    plot_bgcolor='#efefef',
    # Font Families
    font_family='Monospace',
    font_color='#000000',
    font_size=20,
    xaxis=dict(
        ranglider=dict(
            visible=False
        )
    )
)

# Update options and show plot
fig_so.update_layout(layout)
# Create Columns

sto_ind = pn.Column(fig_so)

```

```
In [23]: tabs = pn.Tabs(('Moving Average Convergence Divergence ', MACD), ('RSI', RSI_ind), ('Stochastics', sto_ind)
         dashboard = pn.Column("## Technical Analysis", tabs)
```

```
In [24]: dashboard.servable()
```

Out[24]:

```
In [25]: fig
```





In [ ]: