

Coca Cola Stock- Live and Updated

Data Collection

```
!pip install yfinance
import yfinance as yf
import pandas as pd
ticker = 'KO'
data = yf.download(ticker, start='2015-01-01',
end='2023-12-31')
data.reset_index(inplace=True)
```

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: yfinance in c:\users\nilam\appdata\roaming\python\python312\site-packages (0.2.66)
Requirement already satisfied: pandas>=1.3.0 in c:\programdata\anaconda3\lib\site-packages (from yfinance) (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in c:\programdata\anaconda3\lib\site-packages (from yfinance) (1.26.4)
Requirement already satisfied: requests>=2.31 in c:\programdata\anaconda3\lib\site-packages (from yfinance) (2.32.2)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\nilam\appdata\roaming\python\python312\site-packages (from yfinance) (0.0.12)
Requirement already satisfied: platformdirs>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from yfinance) (3.10.0)
Requirement already satisfied: peewee>=3.16.2 in c:\users\nilam\appdata\roaming\python\python312\site-packages (from yfinance) (3.18.2)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\programdata\anaconda3\lib\site-packages (from yfinance) (4.12.3)
Requirement already satisfied: curl_cffi>=0.7 in c:\users\nilam\appdata\roaming\python\python312\site-packages (from yfinance) (0.13.0)
Requirement already satisfied: protobuf>=3.19.0 in c:\programdata\anaconda3\lib\site-packages (from yfinance) (3.20.3)
Requirement already satisfied: websockets>=13.0 in c:\users\nilam\appdata\roaming\python\python312\site-packages (from yfinance) (15.0.1)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\site-packages (from beautifulsoup4>=4.11.1->yfinance) (2.5)
Requirement already satisfied: cffi>=1.12.0 in c:\programdata\anaconda3\lib\site-packages (from curl_cffi>=0.7->yfinance) (1.16.0)
Requirement already satisfied: tzdata>=2022.7 in c:\programdata\anaconda3\lib\site-packages (from pandas>=1.3.0->yfinance) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (2.2.2)
Requirement already satisfied: pycparser in c:\programdata\anaconda3\lib\site-packages (from cffi>=1.12.0->curl_cffi>=0.7->yfinance) (2.21)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas>=1.3.0->yfinance) (1.16.0)

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2264 entries, 0 to 2263
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   (Date, )              2264 non-null  datetime64[ns]
1   (Close, K0)           2264 non-null  float64
2   (High, K0)            2264 non-null  float64
3   (Low, K0)             2264 non-null  float64
4   (Open, K0)            2264 non-null  float64
5   (Volume, K0)          2264 non-null  int64
dtypes: datetime64[ns](1), float64(4), int64(1)
memory usage: 106.3 KB
None
```

```
print(data.head())
```

	Price Ticker	Date	Close KO	High KO	Low KO	Open KO	Volume KO
0		2015-01-02	29.992605	30.177658	29.750614	30.078013	9921100
1		2015-01-05	29.992605	30.583348	29.949903	30.384060	26292600
2		2015-01-06	30.220348	30.561982	30.063768	30.184762	16897500
3		2015-01-07	30.597584	30.682991	30.305772	30.462352	13412300
4		2015-01-08	30.967680	31.010385	30.675868	30.732808	21743600

Data Cleaning

```
print(data.isnull().sum())
data.fillna(method='ffill', inplace=True)
data.fillna(0, inplace=True)
print(data.isnull().sum())
```

Price	Ticker
Date	0
Close	KO 0
High	KO 0
Low	KO 0
Open	KO 0
Volume	KO 0

dtype: int64

Price	Ticker
Date	0
Close	KO 0
High	KO 0
Low	KO 0
Open	KO 0
Volume	KO 0

dtype: int64

C:\Users\NILAM\AppData\Local\Temp\ipykernel_1740\1662279237.py:2: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.

```
data.fillna(method='ffill', inplace=True)
```

Feature Engineering

```
data['MA_20'] = data['Close'].rolling(window=20).mean()
data['MA_50'] = data['Close'].rolling(window=50).mean()
data['Daily_Return'] = data['Close'].pct_change()
data['Volatility'] = data['Daily_Return'].rolling(window=20).std()
data.dropna(inplace=True)
print(data.head())
```

Price Ticker	Date	Close KO	High KO	Low KO	Open KO	Volume \ KO
49	2015-03-16	28.913834	28.956890	28.698541	28.705717	15238800
50	2015-03-17	29.071718	29.229601	28.935367	29.186541	21661800
51	2015-03-18	29.136295	29.272649	28.562182	29.050179	26247500
52	2015-03-19	28.720072	29.078893	28.677012	29.050187	16057600
53	2015-03-20	29.172186	29.179361	28.712892	28.755952	31608500

Price Ticker	MA_20	MA_50	Daily_Return	Volatility
-----------------	-------	-------	--------------	------------

49	29.748193	29.998750	0.009522	0.011377
50	29.713893	29.980333	0.005461	0.011464
51	29.685668	29.963206	0.002221	0.011486
52	29.623109	29.933201	-0.014285	0.011604
53	29.588138	29.904693	0.015742	0.012260

Exploratory Data Analysis (EDA)

```
print(data.describe())
```

Price Ticker	Date	Close KO	High KO	Low \ KO
count	2215	2215.000000	2215.000000	2215.000000
mean	2019-08-06 07:49:22.889390592	42.048377	42.337400	41.754463
min	2015-03-16 00:00:00	27.488174	27.872285	26.453481
25%	2017-05-24 12:00:00	33.936790	34.151400	33.717890
50%	2019-08-07 00:00:00	39.998840	40.339882	39.651807
75%	2021-10-16 12:00:00	49.775475	50.087506	49.514108
max	2023-12-29 00:00:00	59.625156	60.516683	59.396649
std	NaN	9.340581	9.416087	9.259001

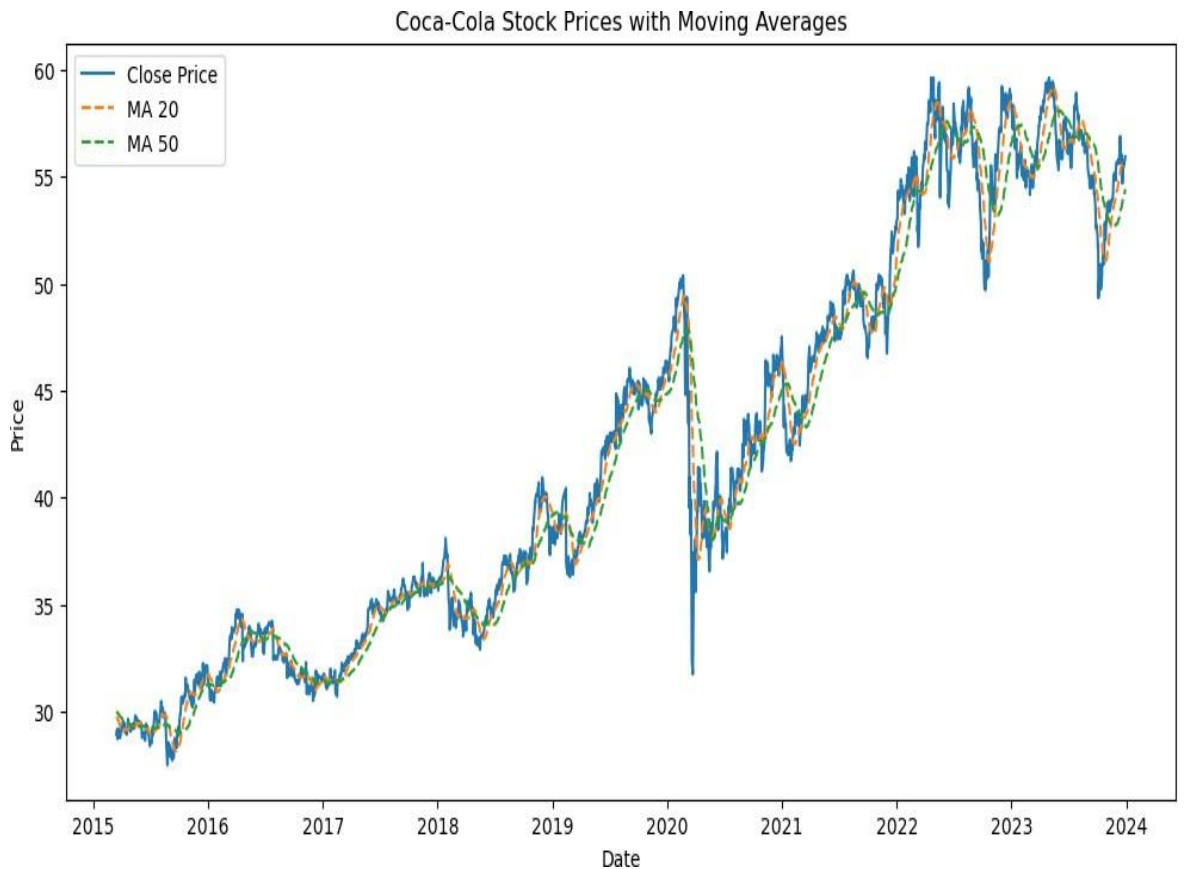
Price Ticker	Open KO	Volume KO	MA_20	MA_50	Daily_Return \
count	2215.000000	2.215000e+03	2215.000000	2215.000000	2215.000000
mean	42.052575	1.410888e+07	41.936514	41.768241	0.000369
min	27.488177	2.996300e+06	28.141970	28.942742	-0.096725
25%	33.951560	1.021795e+07	33.831000	33.648286	-0.004557
50%	40.007284	1.279290e+07	39.589039	39.271224	0.000642
75%	49.768199	1.603545e+07	49.404318	48.932102	0.005890
max	60.336576	6.784570e+07	59.194986	58.078910	0.064796
std	9.346432	6.239615e+06	9.291641	9.226694	0.011520

Price Ticker	Volatility
count	2215.000000
mean	0.009985
min	0.003597
25%	0.006704
50%	0.008680
75%	0.011269
max	0.053882
std	0.005898

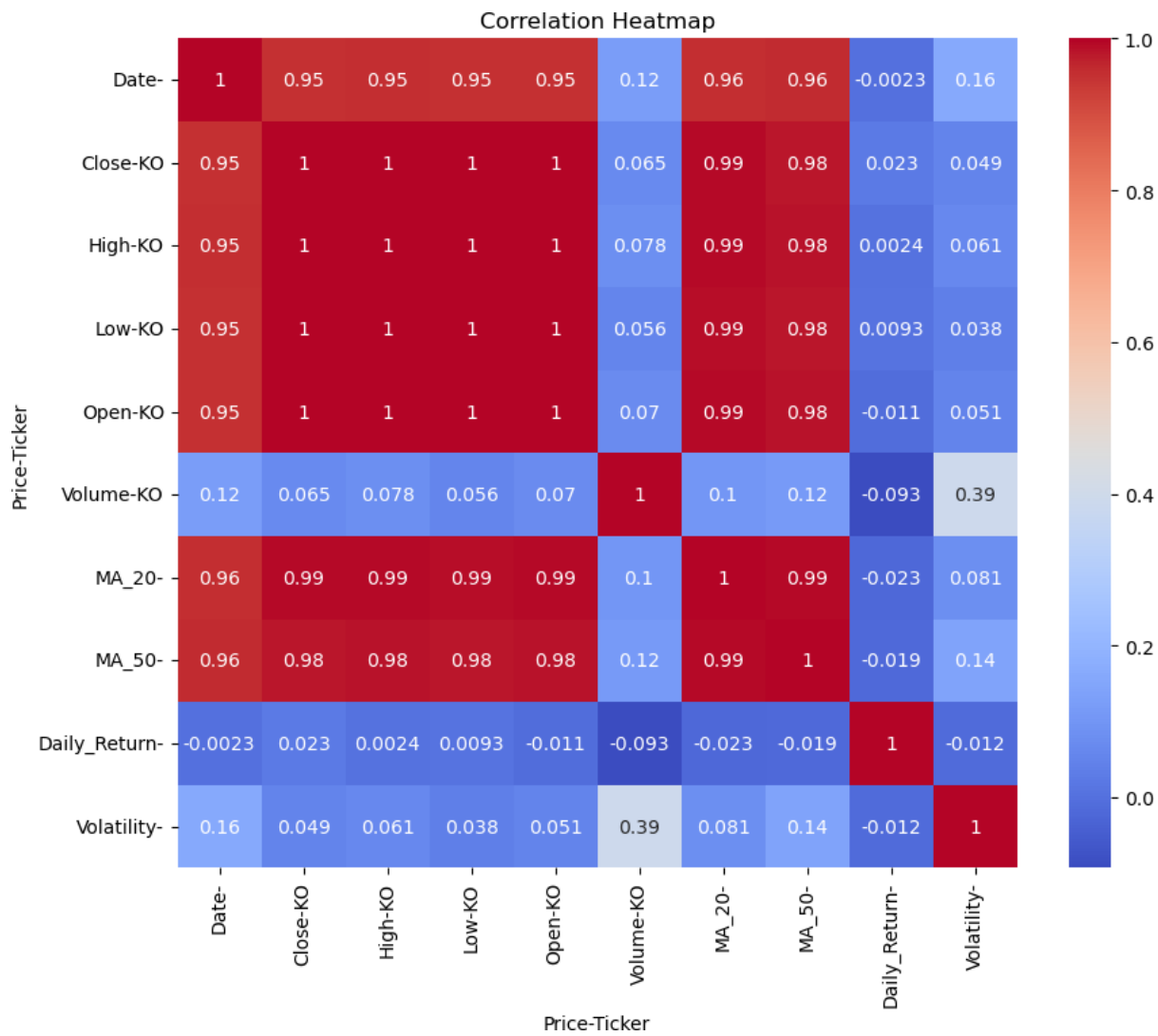
```
import matplotlib.pyplot as plt
orn as sns
```

```
plt.figure(figsize=(12, 6))
plt.plot(data['Date'], data['Close'], label='Close Price')
plt.plot(data['Date'], data['MA_20'], label='MA 20',
linestyle='--')
plt.plot(data['Date'], data['MA_50'], label='MA 50',
linestyle='--')
plt.title('Coca-Cola Stock Prices with Moving Averages')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
```

<matplotlib.legend.Legend at 0x15ea071e150>



```
plt.show()
plt.figure(figsize=(10, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



Data Splitting

```
import yfinance as yf
import pandas as pd
from sklearn.model_selection import train_test_split

ticker = yf.Ticker("KO")
data = ticker.history(start="2015-01-01", end="2023-12-31")
data.reset_index(inplace=True)

data['MA_20'] = data['Close'].rolling(window=20).mean()
data['MA_50'] = data['Close'].rolling(window=50).mean()
data['Daily_Return'] = data['Close'].pct_change()
data['Volatility'] = data['Daily_Return'].rolling(window=20).std()
data.fillna(0, inplace=True)
print(data.columns)
features = ['Open', 'High', 'Low', 'Volume', 'Dividends',
            'Stock Splits', 'MA_20', 'MA_50', 'Daily_Return', 'Volatility']
target = 'Close'
X = data[features]
y = data[target]
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, shuffle=False)
print(X_train.shape, X_test.shape)
```

```
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Dividends',
      'Stock Splits', 'MA_20', 'MA_50', 'Daily_Return', 'Volatility'],
      dtype='object')
(1811, 10) (453, 10)
```

Model Training

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error
# Initialize the model
model = RandomForestRegressor(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"Mean Absolute Error: {mae}")
```

```
Mean Squared Error: 2.607216526301235
Mean Absolute Error: 1.1668836845414787
```

Live Prediction System

```
import yfinance as yf
ticker = "KO"
live_data = yf.download(ticker, period='1d', interval='1m')
live_data['MA_20'] = live_data['Close'].rolling(window=20).mean()
live_data['MA_50'] = live_data['Close'].rolling(window=50).mean()
live_data['Daily_Return'] = live_data['Close'].pct_change()
live_data['Volatility'] = live_data['Daily_Return'].rolling(window=20).std()
print(live_data.tail())
```

C:\Users\NILAM\AppData\Local\Temp\ipykernel_1740\4160867952.py:3: FutureWarning: YF.download() has changed argument auto_adjust default to True

```
live_data = yf.download(ticker, period='1d', interval='1m')
[*****100%*****] 1 of 1 completed
```

Price		Close	High	Low	Open	Volume	\
Ticker		KO	KO	KO	KO	KO	
Datetime							
2025-09-17	19:55:00+00:00	67.035004	67.080002	67.025002	67.040001	137242	
2025-09-17	19:56:00+00:00	66.964996	67.035004	66.964996	67.032501	102830	
2025-09-17	19:57:00+00:00	66.974998	66.999901	66.959999	66.964996	95988	
2025-09-17	19:58:00+00:00	66.992500	67.019997	66.959999	66.980003	242684	
2025-09-17	19:59:00+00:00	67.035004	67.035004	66.970001	66.992500	360500	

Price		MA_20	MA_50	Daily_Return	Volatility
Ticker					
Datetime					
2025-09-17	19:55:00+00:00	67.013000	66.945402	-0.000075	0.000445
2025-09-17	19:56:00+00:00	67.013000	66.948002	-0.001044	0.000496
2025-09-17	19:57:00+00:00	67.014250	66.950502	0.000149	0.000494
2025-09-17	19:58:00+00:00	67.014625	66.953452	0.000261	0.000483
2025-09-17	19:59:00+00:00	67.018876	66.957796	0.000634	0.000486

```
live_data.fillna(0, inplace=True)
live_data['Dividends'] = 0
live_data['Stock Splits'] = 0
latest_features = live_data[features].iloc[-1:].dropna()
live_prediction = model.predict(latest_features)
print(f"Predicted Closing Price: {live_prediction[0]}")
```

Predicted Closing Price: 55.6706640625

C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but RandomForestRegressor was fitted with feature names

```
warnings.warn(
```


Deploy the System

```
import streamlit as st
data = pd.read_csv(r"C:\Users\NILAM\Desktop\Coca-Cola_stock_history.csv")

data['MA_20'] = data['Close'].rolling(window=20).mean()
data['MA_50'] = data['Close'].rolling(window=50).mean()

st.title('Coca-Cola Stock Price Prediction')
st.line_chart(data[['Close', 'MA_20', 'MA_50']])
live_prediction = [data['Close'].iloc[-1] * 1.02]
st.write(f"Predicted Closing Price: {live_prediction[0]}")
```

Stock Analysis of Coca Cola Stock

```
!pip install pandas-datareader
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
plt.style.use("fivethirtyeight")
%matplotlib inline
from pandas_datareader.data import DataReader
from datetime import datetime
from math import sqrt
from math import sqrt
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import MinMaxScaler
import warnings
warnings.filterwarnings('ignore')
```

Defaulting to user installation because normal site-packages is not writeable
Collecting pandas-datareader

Downloading pandas_datareader-0.10.0-py3-none-any.whl.metadata (2.9 kB)

Requirement already satisfied: lxml in c:\programdata\anaconda3\lib\site-packages (from pandas-datareader) (5.2.1)

Requirement already satisfied: pandas>=0.23 in c:\programdata\anaconda3\lib\site-packages (from pandas-datareader) (2.2.2)

Requirement already satisfied: requests>=2.19.0 in c:\programdata\anaconda3\lib\site-packages (from pandas-datareader) (2.32.2)

Requirement already satisfied: numpy>=1.26.0 in c:\programdata\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (1.26.4)

Requirement already satisfied: python-dateutil>=2.8.2 in c:\programdata\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (2024.1)

Requirement already satisfied: tzdata>=2022.7 in c:\programdata\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (2023.3)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (3.7)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (2.2.2)

Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (2024.8.30)

Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas>=0.23->pandas-datareader) (1.16.0)

Downloading pandas_datareader-0.10.0-py3-none-any.whl (109 kB)

```

----- 0.0/109.5 kB ? eta -:--:--
--- ----- 10.2/109.5 kB ? eta -:--:--
--- ----- 10.2/109.5 kB ? eta -:--:--
--- ----- 10.2/109.5 kB ? eta -:--:--
--- ----- 10.2/109.5 kB ? eta -:--:--
----- 81.9/109.5 kB 328.8 kB/s eta 0:00:01
----- 81.9/109.5 kB 328.8 kB/s eta 0:00:01
----- 109.5/109.5 kB 318.0 kB/s eta 0:00:00

```

Installing collected packages: pandas-datareader

Successfully installed pandas-datareader-0.10.0

```

import pandas as pd
KO_Data = pd.read_csv(r'C:\Users\NILAM\Desktop\Coca-Cola_stock_history.csv')
print(KO_Data.head())

```

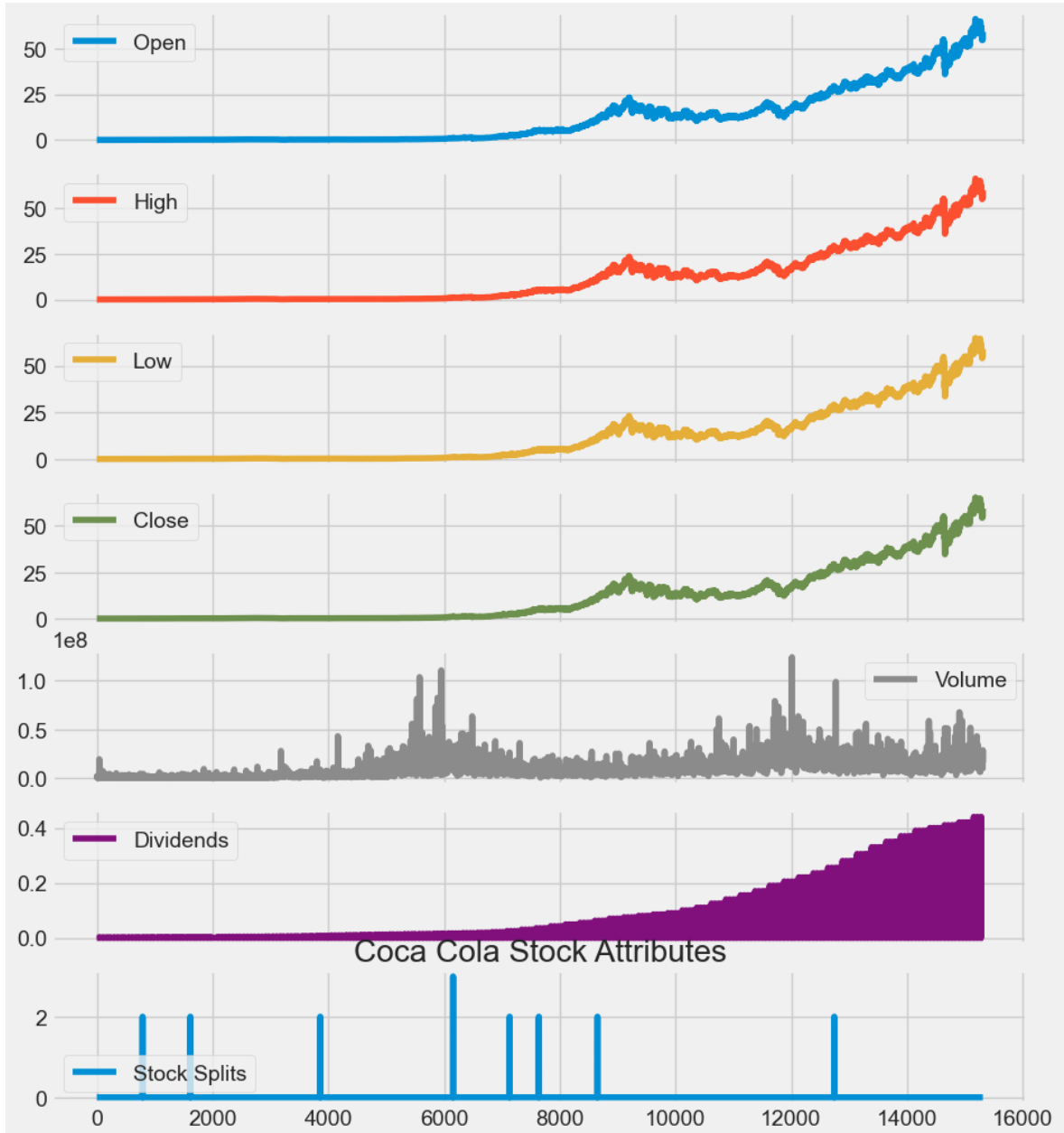
	Date	Open	High	Low	Close	Volume	Dividends	\
0	1962-01-02	0.050016	0.051378	0.050016	0.050016	806400	0.0	
1	1962-01-03	0.049273	0.049273	0.048159	0.048902	1574400	0.0	
2	1962-01-04	0.049026	0.049645	0.049026	0.049273	844800	0.0	
3	1962-01-05	0.049273	0.049892	0.048035	0.048159	1420800	0.0	
4	1962-01-08	0.047787	0.047787	0.046735	0.047664	2035200	0.0	

Stock Splits

0	0
1	0
2	0
3	0
4	0

Basic EDA

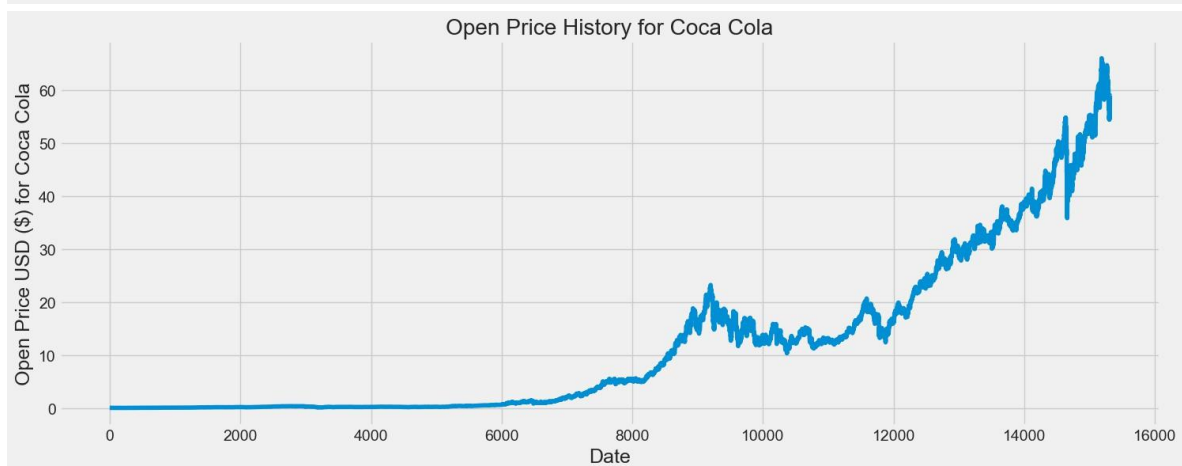
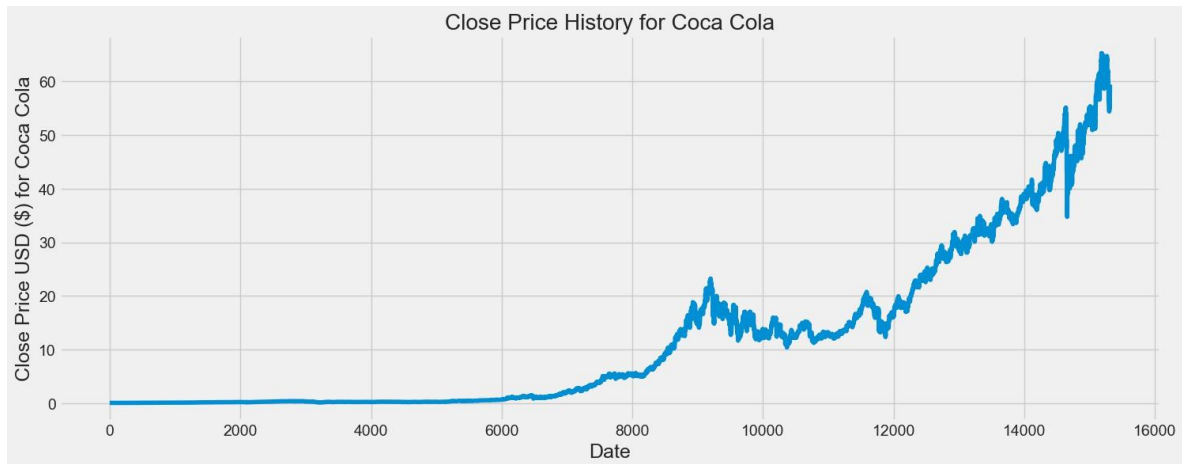
```
KO_Data.plot(subplots = True, figsize = (10,12))  
plt.title('Coca Cola Stock Attributes')  
plt.show()
```



```
def plot_close_val(data_frame, column, stock):  
    plt.figure(figsize=(16,6))  
    plt.title(column + ' Price History for ' + stock )  
    plt.plot(data_frame[column])  
    plt.xlabel('Date', fontsize=18)  
    plt.ylabel(column + ' Price USD ($) for ' + stock, fontsize=18)  
    plt.show()
```

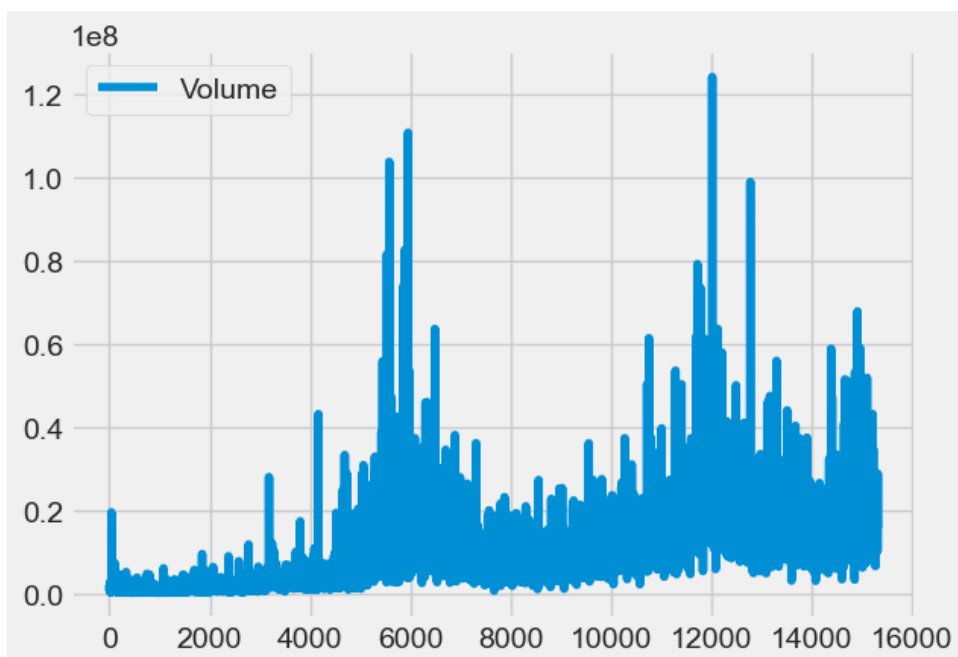
Test the Function

```
plot_close_val(KO_Data, 'Close', 'Coca Cola')  
plot_close_val(KO_Data, 'Open', 'Coca Cola')
```



```
KO_Data[["Volume"]].plot()
```

<Axes: >



Basic Company Info

```
ko_info = pd.read_csv(r'C:\Users\NILAM\Desktop/Coca-Cola_stock_info.csv',
                      header=None, names=['Description', 'Information'])
ko_info = ko_info.dropna()
ko_info.drop(ko_info.loc[ko_info['Information']=='nan'].index, inplace=True)
ko = ko_info.sort_values('Information').style
ko
```

13	industry	Beverages—Non-Alcoholic
44	shortName	Coca-Cola Company (The)
2	sector	Consumer Defensive
50	quoteType	EQUITY
47	exchangeTimezoneShortName	EST
146	tradeable	False
48	isEsgPopulated	False
7	state	GA
51	symbol	KO
43	exchange	NYQ
12	address1	One Coca-Cola Plaza
45	longName	The Coca-Cola Company
127	currency	USD
39	financialCurrency	USD
8	country	United States
0	Key	Value
9	companyOfficers	[]

4	longBusinessSummary	<p>The Coca-Cola Company, a beverage company, manufactures, markets, and sells various nonalcoholic beverages worldwide. The company provides sparkling soft drinks; water, enhanced water, and sports drinks; juice, dairy, and plant-based beverages; tea and coffee; and energy drinks. It also offers beverage concentrates and syrups, as well as fountain syrups to fountain retailers, such as restaurants and convenience stores. The company sells its products under the Coca-Cola, Diet Coke/Coca-Cola Light, Coca-Cola Zero Sugar, Fanta, Fresca, Schweppes, Sprite, Thums Up, Aquarius, Ciel, Dasani, glacéau smartwater, glacéau vitaminwater, Ice Dew, I LOHAS, Powerade, Topo Chico, AdeS, Del Valle, fairlife, innocent, Minute Maid, Minute Maid Pulpy, Simply, Ayataka, Costa, dogadan, FUZE TEA, Georgia, Gold Peak, HONEST TEA, and Kochakaden brands.</p>
---	---------------------	---

```
adj_close_px = KO_Data['Close']
```

```
moving_avg = adj_close_px.rolling(window=40).mean()
```

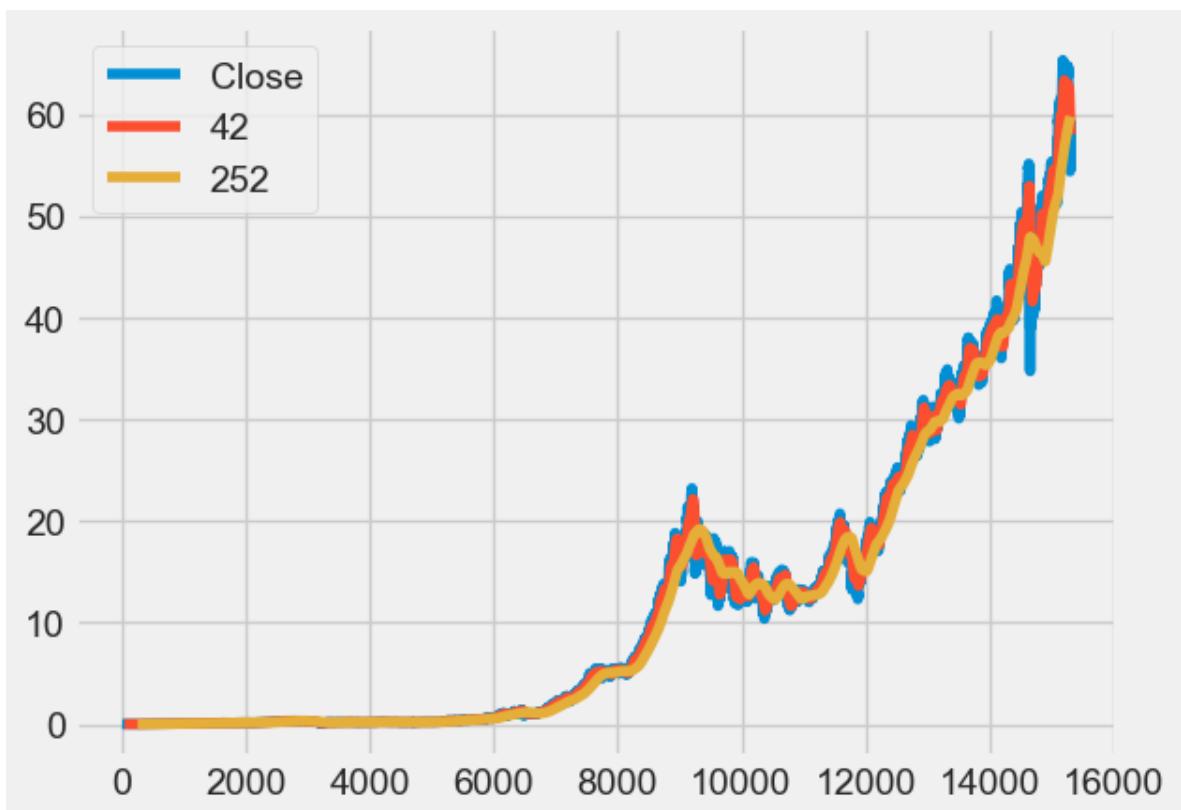
```
moving_avg[-10:]
```

```
15301    59.573229
15302    59.329031
15303    59.103823
15304    58.921440
15305    58.725320
15306    58.504966
15307    58.298918
15308    58.171838
15309    58.088689
15310    58.030935
Name: Close, dtype: float64
```

```
KO_Data['42'] = adj_close_px.rolling(window=40).mean()
```

```
KO_Data['252'] = adj_close_px.rolling(window=252).mean()
```

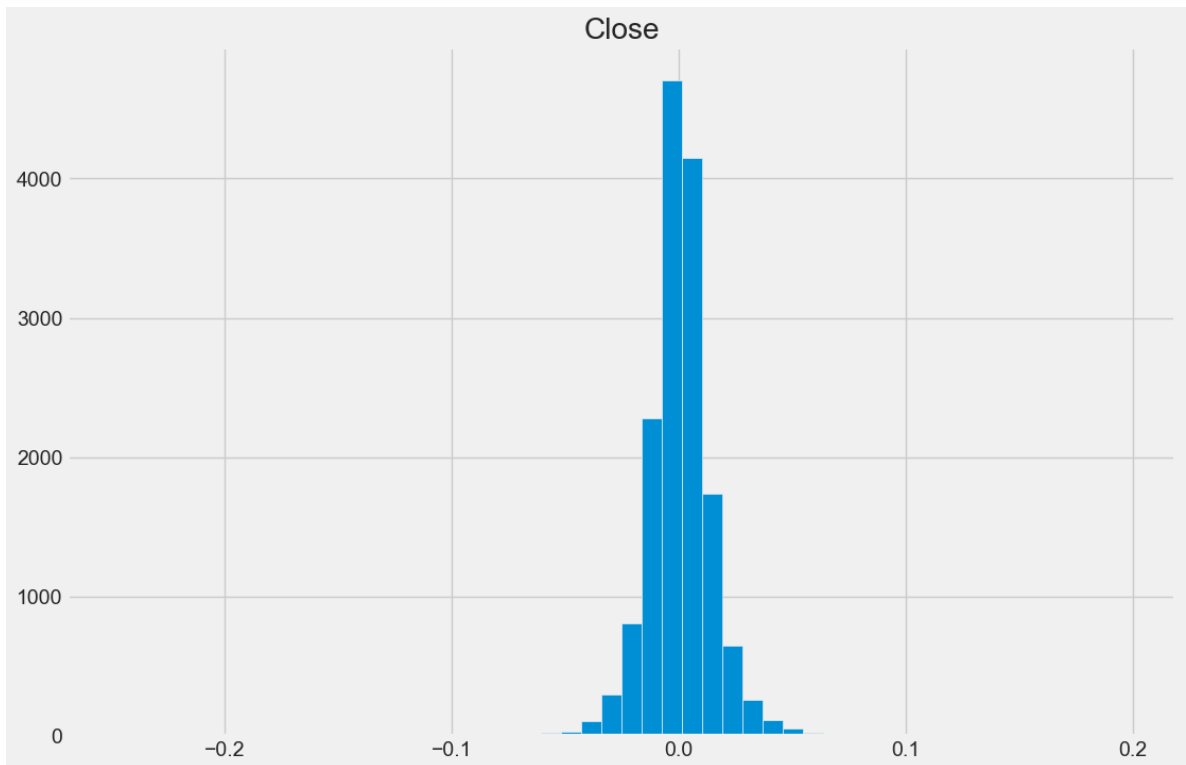
```
KO_Data[['Close', '42', '252']].plot()
plt.show()
```



```

daily_close_px = KO_Data[['Close']]
daily_pct_change = daily_close_px.pct_change()
daily_pct_change.hist(bins=50, sharex=True, figsize=(12,8))
plt.show()

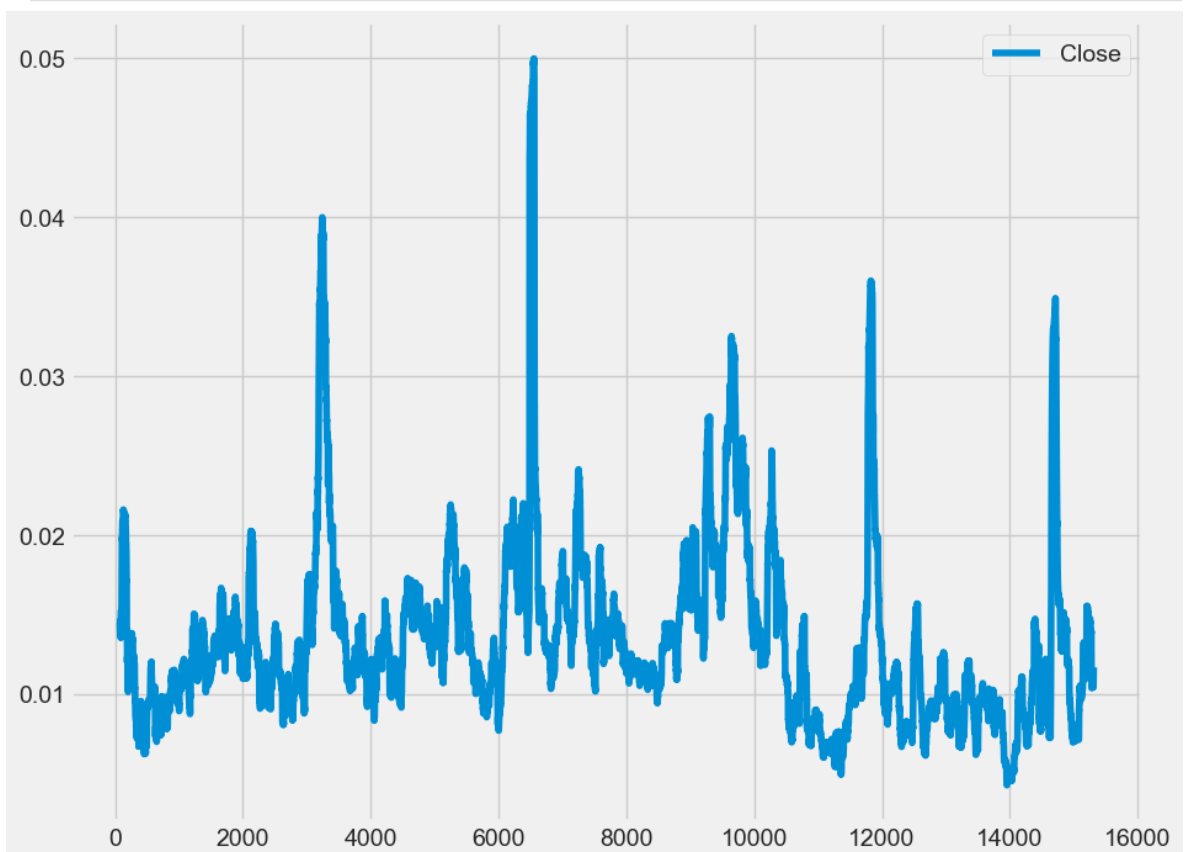
```



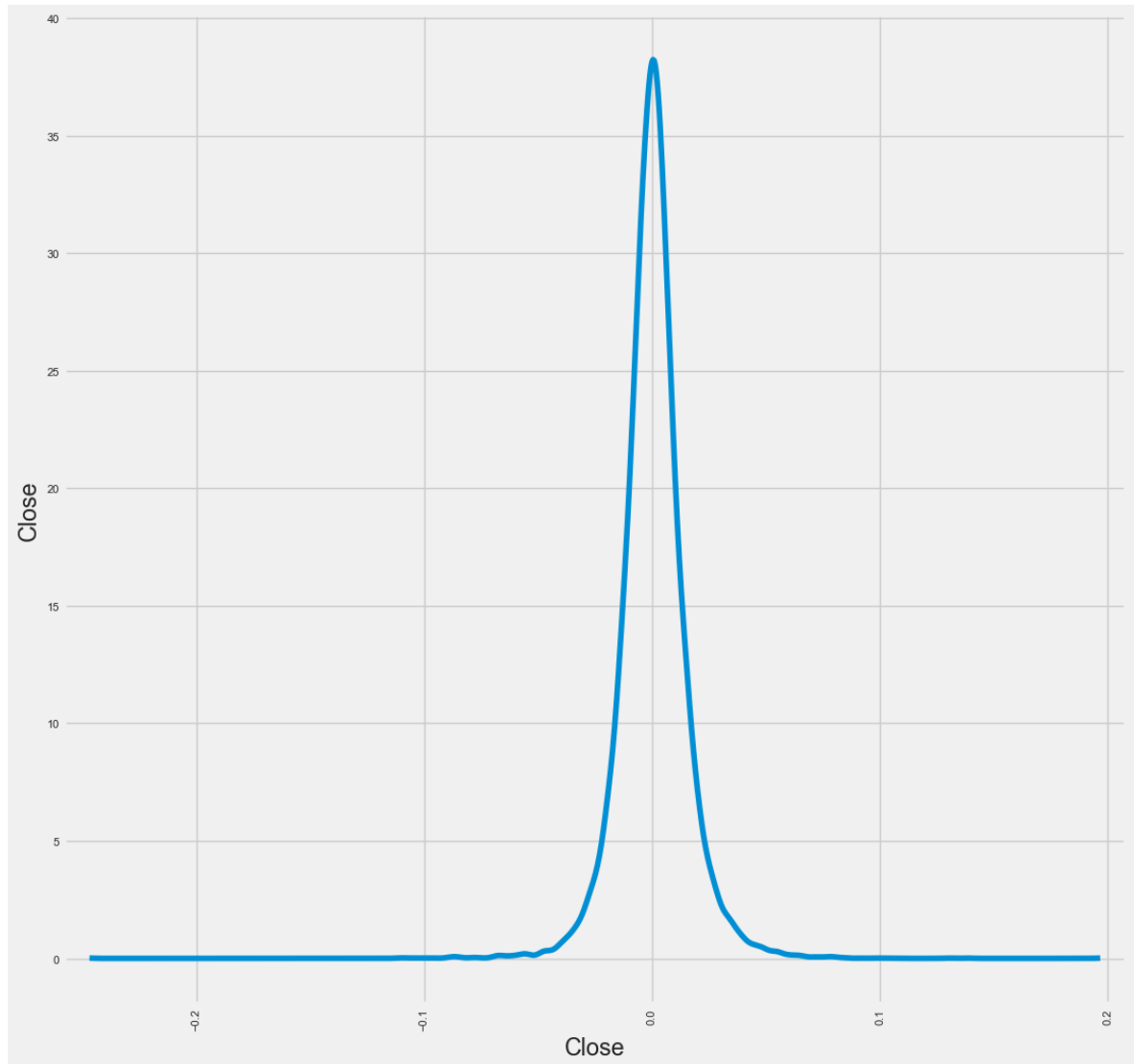
```

min_periods=75
vol=daily_pct_change.rolling(min_periods).std()
np.sqrt(min_periods)
vol.plot(figsize=(10,8))
plt.show()

```

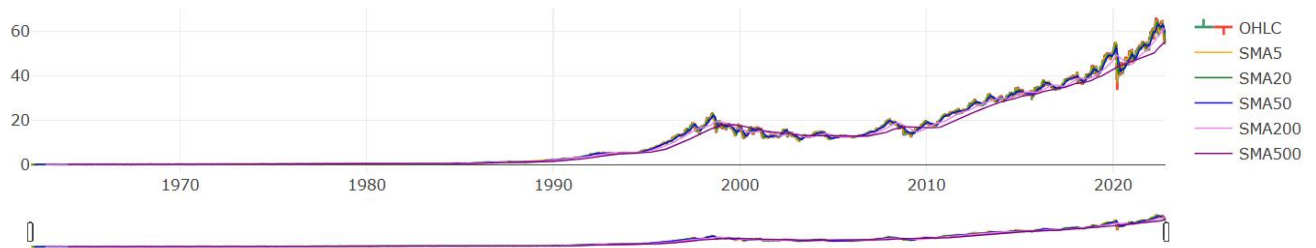


```
pd.plotting.scatter_matrix(daily_pct_change, diagonal='kde',
alpha=0.1,figsize=(12,12))
plt.show()
```



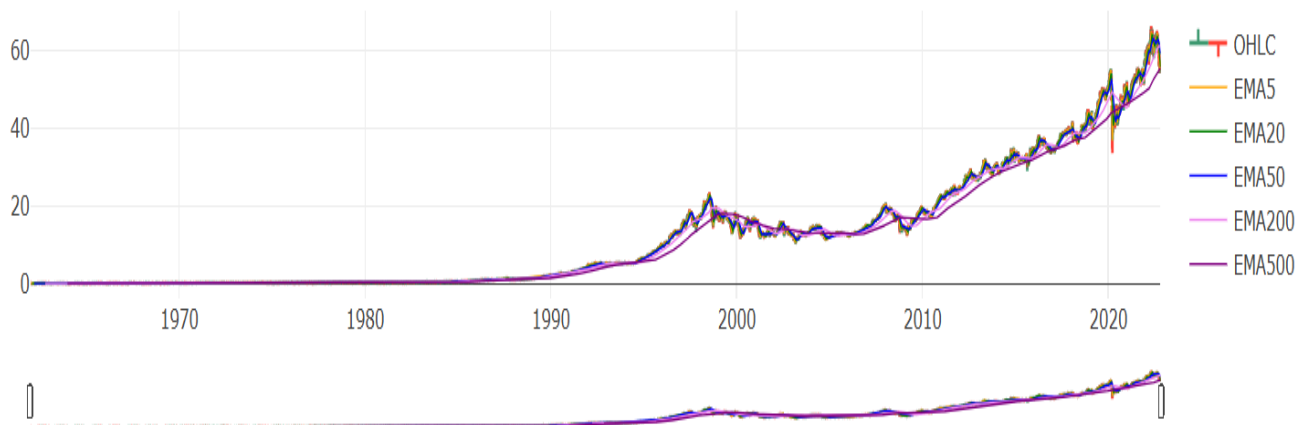
Basic MACD

```
import plotly.graph_objects as go
KO_Data=KO_Data.reset_index()
fig = go.Figure(data=go.Ohlc(x=KO_Data['Date'],
open=KO_Data['Open'],
high=KO_Data['High'],
low=KO_Data['Low'],
close=KO_Data['Close'])))
fig.show()
```

Basic SMA

```
KO_Data['SMA5'] = KO_Data.Close.rolling(5).mean()
KO_Data['SMA20'] = KO_Data.Close.rolling(20).mean()
KO_Data['SMA50'] = KO_Data.Close.rolling(50).mean()
KO_Data['SMA200'] = KO_Data.Close.rolling(200).mean()
KO_Data['SMA500'] = KO_Data.Close.rolling(500).mean()
fig = go.Figure(data=[go.Ohlc(x=KO_Data['Date'], open=KO_Data['Open'],
high=KO_Data['High'], low=KO_Data['Low'], close=KO_Data['Close'], name = "OHLC"),
go.Scatter(x=KO_Data.Date, y=KO_Data.SMA5, line=dict(color='orange', width=1),
name="SMA5"), go.Scatter(x=KO_Data.Date, y=KO_Data.SMA20, line=dict(color='green',
name="SMA20"), go.Scatter(x=KO_Data.Date, y=KO_Data.SMA50, line=dict(color='blue',
name="SMA50"), go.Scatter(x=KO_Data.Date, y=KO_Data.SMA200, line=dict(color='violet',
name="SMA200"), go.Scatter(x=KO_Data.Date, y=KO_Data.SMA500, line=dict(color='purple',
name="SMA500"))])
fig.show()
```



Basic EMA

```
KO_Data['EMA5'] = KO_Data.Close.ewm(span=5,adjust=False).mean()
KO_Data['EMA20'] = KO_Data.Close.ewm(span=20,adjust=False).mean()
KO_Data['EMA50'] = KO_Data.Close.ewm(span=50,adjust=False).mean()
KO_Data['EMA200'] = KO_Data.Close.ewm(span=200,adjust=False).mean()
KO_Data['EMA500'] = KO_Data.Close.ewm(span=500,adjust=False).mean()
fig = go.Figure(data=[go.Ohlc(x=KO_Data['Date'],open=KO_Data['Open'],high=KO_Data['High'],low=KO_Data['Low'],close=KO_Data['Close'],
name="OHLC"),go.Scatter(x=KO_Data.Date,y=KO_Data.SMA5, line=dict(color='orange',name="EMA5"),go.Scatter(x=KO_Data.Date, y=KO_Data.SMA20, line=dict(color='green',name="EMA20"),go.Scatter(x=KO_Data.Date,y=KO_Data.SMA50, line=dict(color='blue',name="EMA50"),go.Scatter(x=KO_Data.Date,y=KO_Data.SMA200,line=dict(color='violet',name="EMA200"),go.Scatter(x=KO_Data.Date,y=KO_Data.SMA500,line=dict(color='purple',name="EMA500"))])
fig.show()
```

KO_Data.head()

	index	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0		1962-01-02	0.050016	0.051378	0.050016	0.050016	806400	0.0	0 NaN
1		1962-01-03	0.049273	0.049273	0.048159	0.048902	1574400	0.0	0 NaN
2		1962-01-04	0.049026	0.049645	0.049026	0.049273	844800	0.0	0 NaN
3		1962-01-05	0.049273	0.049892	0.048035	0.048159	1420800	0.0	0 NaN
4		1962-01-08	0.047787	0.047787	0.046735	0.047664	2035200	0.0	0 NaN

FINTA Tech Analysis Ratios

```
!pip install finta backtesting
from finta import TA
from backtesting import Backtest, Strategy
from backtesting.lib import crossover
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting finta
  Downloading finta-1.3-py3-none-any.whl.metadata (6.4 kB)
Collecting backtesting
  Downloading backtesting-0.6.5-py3-none-any.whl.metadata (7.0 kB)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (from finta) (1.26.4)
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (from finta) (2.2.2)
Requirement already satisfied: bokeh!=3.0.*,!=3.2.*,>=3.0.0 in c:\programdata\anaconda3\lib\site-packages (from backtesting) (3.4.1)
Requirement already satisfied: Jinja2>=2.9 in c:\programdata\anaconda3\lib\site-packages (from bokeh!=3.0.*,!=3.2.*,>=3.0.0->backtesting) (3.1.4)
Requirement already satisfied: contourpy>=1.2 in c:\programdata\anaconda3\lib\site-packages (from bokeh!=3.0.*,!=3.2.*,>=3.0.0->backtesting) (1.2.0)
Requirement already satisfied: packaging>=16.8 in c:\programdata\anaconda3\lib\site-packages (from bokeh!=3.0.*,!=3.2.*,>=3.0.0->backtesting) (23.2)
Requirement already satisfied: xyzservices>=2021.09.1 in c:\programdata\anaconda3\lib\site-packages (from bokeh!=3.0.*,!=3.2.*,>=3.0.0->backtesting) (2022.9.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\programdata\anaconda3\lib\site-packages (from pandas->finta) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas->finta) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\programdata\anaconda3\lib\site-packages (from pandas->finta) (2023.3)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3\lib\site-packages (from Jinja2>=2.9->bokeh!=3.0.*,!=3.2.*,>=3.0.0->backtesting) (2.1.3)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas->finta) (1.16.0)
Downloading finta-1.3-py3-none-any.whl (29 kB)
Downloading backtesting-0.6.5-py3-none-any.whl (192 kB)
----- 0.0/192.1 kB ? eta -:-:-
----- 61.4/192.1 kB 1.7 MB/s eta 0:00:01
----- 153.6/192.1 kB 1.5 MB/s eta 0:00:01
----- 192.1/192.1 kB 1.7 MB/s eta 0:00:00
Installing collected packages: finta, backtesting
Successfully installed backtesting-0.6.5 finta-1.3
```



BokehJS 3.4.1 successfully loaded.

```
fin_ma = pd.read_csv(r'C:\Users\NILAM\Desktop\Coca-Cola_stock_history.csv', pars
print(fin_ma.head())
ohlcl=fin_ma
print(TA.SMA(ohlcl, 42))
```

	Date	Open	High	Low	Close	Volume	Dividends	\
0	1962-01-02	0.050016	0.051378	0.050016	0.050016	806400	0.0	
1	1962-01-03	0.049273	0.049273	0.048159	0.048902	1574400	0.0	
2	1962-01-04	0.049026	0.049645	0.049026	0.049273	844800	0.0	
3	1962-01-05	0.049273	0.049892	0.048035	0.048159	1420800	0.0	
4	1962-01-08	0.047787	0.047787	0.046735	0.047664	2035200	0.0	

	Stock Splits
0	0
1	0
2	0
3	0
4	0
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN

```

...
15306    58.759467
15307    58.572686
15308    58.422110
15309    58.297065
15310    58.219369
Name: 42 period SMA, Length: 15311, dtype: float64

```

```

class DemaCross(Strategy):
    def init(self):
        self.ma1 = self.I(TA.DEMA, ohlc, 10)
        self.ma2 = self.I(TA.DEMA, ohlc, 20)
    def next(self):
        if crossover(self.ma1, self.ma2):
            self.buy()
        elif crossover(self.ma2, self.ma1):
            self.sell()

```

```

ohlc.head()
print(ohlc.Date)

```

0	1962-01-02
1	1962-01-03
2	1962-01-04
3	1962-01-05
4	1962-01-08
...	
15306	2022-10-20 00:00:00-04:00
15307	2022-10-21 00:00:00-04:00
15308	2022-10-24 00:00:00-04:00
15309	2022-10-25 00:00:00-04:00
15310	2022-10-26 00:00:00-04:00

Name: Date, Length: 15311, dtype: object

```

bt = Backtest(ohlc, DemaCross,
    cash=100000, commission=0.015,
    exclusive_orders=True)
bt.run()

```

Backtest.run: 0%| | 0/15310 [00:00<?, ?bar/s]

Start	0.0
End	15310.0
Duration	15310.0
Exposure Time [%]	26.88263
Equity Final [\$]	0.19041
Equity Peak [\$]	100000.0
Commissions [\$]	77449.51548
Return [%]	-99.99981
Buy & Hold Return [%]	118642.19364
Return (Ann.) [%]	0.0
Volatility (Ann.) [%]	NaN
Sharpe Ratio	NaN
Sortino Ratio	NaN
Calmar Ratio	0.0
Alpha [%]	-88.72432
Beta	-0.0001
Max. Drawdown [%]	-99.99981
Avg. Drawdown [%]	-99.99981
Max. Drawdown Duration	15304.0
Avg. Drawdown Duration	15304.0
# Trades	362.0
Win Rate [%]	12.43094
Best Trade [%]	45.6298
Worst Trade [%]	-17.64692
Avg. Trade [%]	-3.79272
Max. Trade Duration	59.0
Avg. Trade Duration	11.35635
Profit Factor	0.15259
Expectancy [%]	-3.68244
SQN	-5.3684
Kelly Criterion	-1.02421
_strategy	DemaCross
_equity_curve	Equ...
_trades	Size En...
dtype:	object

```
bt.plot()
```

GridPlot(id = 'p1311', ...)

```
data=ohlcv
```

```
from backtesting import Strategy
from backtesting.lib import crossover
from backtesting.test import SMA
```

```

def BBANDS(data, n_lookback, n_std):
    hlc3 = (data.High + data.Low + data.Close) / 3
    mean = hlc3.rolling(n_lookback).mean()
    std = hlc3.rolling(n_lookback).std()

    upper = mean + n_std*std
    lower = mean - n_std*std

    return upper, lower
upper, lower = BBANDS(data, 20, 2)

```

```

import pandas as pd
data = pd.read_csv(r"C:\Users\NILAM\Desktop\Coca-Cola_stock_history.csv")
print(data.head())
close = data.Close.values
def SMA(series, n):
    return series.rolling(n).mean()
sma10 = SMA(data.Close, 10)
sma20 = SMA(data.Close, 20)
sma50 = SMA(data.Close, 50)
sma100 = SMA(data.Close, 100)

upper, lower = BBANDS(data, 20, 2)

data['X_SMA10'] = (close - sma10) / close
data['X_SMA20'] = (close - sma20) / close
data['X_SMA50'] = (close - sma50) / close
data['X_SMA100'] = (close - sma100) / close
data['X_DELTA_SMA10'] = (sma10 - sma20) / close
data['X_DELTA_SMA20'] = (sma20 - sma50) / close
data['X_DELTA_SMA50'] = (sma50 - sma100) / close
data['X_MOM'] = data.Close.pct_change(periods=2)
data['X_BB_upper'] = (upper - close) / close
data['X_BB_lower'] = (lower - close) / close
data['X_BB_width'] = (upper - lower) / close

```

	Date	Open	High	Low	Close	Volume	Dividends	\
0	1962-01-02	0.050016	0.051378	0.050016	0.050016	806400	0.0	
1	1962-01-03	0.049273	0.049273	0.048159	0.048902	1574400	0.0	
2	1962-01-04	0.049026	0.049645	0.049026	0.049273	844800	0.0	
3	1962-01-05	0.049273	0.049892	0.048035	0.048159	1420800	0.0	
4	1962-01-08	0.047787	0.047787	0.046735	0.047664	2035200	0.0	

	Stock Splits
0	0
1	0
2	0
3	0
4	0

```

!pip install scikit-optimize
%%time
stats_skopt, heatmap, optimize_result = backtest.optimize(
    n1=[10, 100],
    n2=[20, 200],
    n_enter=[10, 40],
    n_exit=[10, 30],

    onstraint=lambda p: p.n_exit < p.n_enter < p.n1 < p.n2,
    maximize='Equity Final [$]',
    method='skopt',
    max_tries=200,
    random_state=0,
    return_heatmap=True,
    return_optimization=True)
from skopt.plots import plot_objective
_ = plot_objective(optimize_result, n_points=10)

```

Defaulting to user installation because normal site-packages is not writeable
Collecting scikit-optimize

Downloading scikit_optimize-0.10.2-py2.py3-none-any.whl.metadata (9.7 kB)

Requirement already satisfied: joblib>=0.11 in c:\programdata\anaconda3\lib\site-packages (from scikit-optimize) (1.4.2)

Collecting pyaml>=16.9 (from scikit-optimize)

Downloading pyaml-25.7.0-py3-none-any.whl.metadata (12 kB)

Requirement already satisfied: numpy>=1.20.3 in c:\programdata\anaconda3\lib\site-packages (from scikit-optimize) (1.26.4)

Requirement already satisfied: scipy>=1.1.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-optimize) (1.13.1)

Requirement already satisfied: scikit-learn>=1.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-optimize) (1.4.2)

Requirement already satisfied: packaging>=21.3 in c:\programdata\anaconda3\lib\site-packages (from scikit-optimize) (23.2)

Requirement already satisfied: PyYAML in c:\programdata\anaconda3\lib\site-packages (from pyaml>=16.9->scikit-optimize) (6.0.1)

Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn>=1.0.0->scikit-optimize) (2.2.0)

Downloading scikit_optimize-0.10.2-py2.py3-none-any.whl (107 kB)

----- 0.0/107.8 kB ? eta -:-:-

----- 20.5/107.8 kB 682.7 kB/s eta 0:00:01

----- 41.0/107.8 kB 393.8 kB/s eta 0:00:01

----- 102.4/107.8 kB 737.3 kB/s eta 0:00:01

----- 107.8/107.8 kB 696.8 kB/s eta 0:00:00

Downloading pyaml-25.7.0-py3-none-any.whl (26 kB)

Installing collected packages: pyaml, scikit-optimize

Successfully installed pyaml-25.7.0 scikit-optimize-0.10

WARNING: The script pyaml.exe is installed in 'C:\Users\NILAM\AppData\Roaming\Python\Python312\Scripts' which is not on PATH.

Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

UsageError: Line magic function `%%time` not found.

