Main

October 31, 2024

[1]:
```
import yfinance as yf
import pandas as pd
import pandas_ta as ta
import numpy as np
import matplotlib.pyplot as plt
```

Importing stocks and adding .NS to make it compatible with the Yfinance format

[2]:
```
ticker=␣
 ↪['SBIN','ONGC','CIPLA','JIOFIN','RELIANCE','TCS','ASIANPAINT','INFIBEAM','COALINDIA','WIPRO
for i in range(0,len(ticker)):
    ticker[i] = ticker[i] + ".NS"
```

Creating the DataFrame and Dropping stocks which do not have values

[3]:
```
stocks = yf.download(ticker,start="2010-01-01", end="2024-01-01")
stocks = stocks.dropna(axis=1, how='any')
stocks.head()
```

```
[**********************100%**********************]  21 of 21 completed
```

[3]:

Price	Adj Close				\
Ticker	ADANIENT.NS	ASIANPAINT.NS	BHARTIARTL.NS	CIPLA.NS	
Date					
2010-01-04 00:00:00+00:00	57.215855	157.902100	272.333496	307.199127	
2010-01-05 00:00:00+00:00	58.539829	158.400894	276.646271	301.693207	
2010-01-06 00:00:00+00:00	64.317162	157.924164	273.715210	313.888245	
2010-01-07 00:00:00+00:00	61.662884	156.313049	275.850739	311.294495	
2010-01-08 00:00:00+00:00	62.442066	158.895264	272.207886	310.429993	

Price					\
Ticker	HDFCBANK.NS	IDEA.NS	INFY.NS	ITC.NS	
Date					
2010-01-04 00:00:00+00:00	151.915100	34.550125	232.806366	57.650013	
2010-01-05 00:00:00+00:00	152.048721	35.767101	233.586075	58.218220	
2010-01-06 00:00:00+00:00	152.151138	36.390427	230.177734	58.354588	
2010-01-07 00:00:00+00:00	152.547455	36.242012	225.004868	58.172752	
2010-01-08 00:00:00+00:00	152.747864	35.945190	219.582611	58.297771	

```
Price                                                 …    Volume            \
Ticker                          LT.NS     ONGC.NS     …   INFY.NS    ITC.NS
Date                                                  …
2010-01-04 00:00:00+00:00  609.478394  105.796288     …   4069264   5161635
2010-01-05 00:00:00+00:00  610.559387  107.649467     …   6895528  10767225
2010-01-06 00:00:00+00:00  603.803040  109.008194     …   6817288   7529979
2010-01-07 00:00:00+00:00  600.902344  109.123970     …  10892600   4987434
2010-01-08 00:00:00+00:00  604.703857  108.313210     …  12649312   4881351

Price                                                                  \
Ticker                        LT.NS  ONGC.NS RELIANCE.NS   SBIN.NS
Date
2010-01-04 00:00:00+00:00  1060024  1905762    76646086  11031470
2010-01-05 00:00:00+00:00  2683989  4657392    21392825  11613740
2010-01-06 00:00:00+00:00  3076209  4203690    23691760  14527930
2010-01-07 00:00:00+00:00  2033021  6394536    26197920   8576510
2010-01-08 00:00:00+00:00  2940979  5678652    15110149  10453820

Price
Ticker                     SUNPHARMA.NS   TCS.NS TRIDENT.NS WIPRO.NS
Date
2010-01-04 00:00:00+00:00       831170  1963682    7309990  3409626
2010-01-05 00:00:00+00:00      1231990  2014488    5342380  4979701
2010-01-06 00:00:00+00:00      1882490  3349176    3667230  4575030
2010-01-07 00:00:00+00:00       923100  6474892   22172980  4582648
2010-01-08 00:00:00+00:00      3298080  6048178   19074960  2856839

[5 rows x 96 columns]
```

[4]:
```
#creating a new dataframe which has values of closing price of columns
close = stocks.loc[:,"Close"].copy()
close.head()
```

[4]:
```
Ticker                     ADANIENT.NS  ASIANPAINT.NS  BHARTIARTL.NS  \
Date
2010-01-04 00:00:00+00:00    69.219017     178.865005     293.088867
2010-01-05 00:00:00+00:00    70.820740     179.429993     297.730347
2010-01-06 00:00:00+00:00    77.810081     178.889999     294.575928
2010-01-07 00:00:00+00:00    74.598969     177.065002     296.874146
2010-01-08 00:00:00+00:00    75.541611     179.990005     292.953674

Ticker                       CIPLA.NS  HDFCBANK.NS    IDEA.NS     INFY.NS  \
Date
2010-01-04 00:00:00+00:00  337.549988   170.570007  35.092796  326.575012
2010-01-05 00:00:00+00:00  331.500000   170.720001  36.328884  327.668762
2010-01-06 00:00:00+00:00  344.899994   170.835007  36.962002  322.887512
2010-01-07 00:00:00+00:00  342.049988   171.279999  36.811256  315.631256
```

```
2010-01-08 00:00:00+00:00  341.100006   171.505005  36.509773  308.024994

Ticker                        ITC.NS       LT.NS     ONGC.NS  RELIANCE.NS  \
Date
2010-01-04 00:00:00+00:00  84.550003  751.733337  197.908340   245.841202
2010-01-05 00:00:00+00:00  85.383331  753.066650  201.375000   244.744003
2010-01-06 00:00:00+00:00  85.583336  744.733337  203.916672   248.698502
2010-01-07 00:00:00+00:00  85.316666  741.155579  204.133331   252.824417
2010-01-08 00:00:00+00:00  85.500000  745.844421  202.616669   252.161530

Ticker                       SBIN.NS  SUNPHARMA.NS      TCS.NS  TRIDENT.NS  \
Date
2010-01-04 00:00:00+00:00  229.119995    150.735001  375.825012       1.515
2010-01-05 00:00:00+00:00  229.205002    155.264999  375.924988       1.505
2010-01-06 00:00:00+00:00  230.580002    157.279999  367.424988       1.500
2010-01-07 00:00:00+00:00  229.289993    154.884995  357.200012       1.560
2010-01-08 00:00:00+00:00  228.604996    157.494995  349.899994       1.565

Ticker                       WIPRO.NS
Date
2010-01-04 00:00:00+00:00  156.105011
2010-01-05 00:00:00+00:00  158.568756
2010-01-06 00:00:00+00:00  155.362503
2010-01-07 00:00:00+00:00  152.606262
2010-01-08 00:00:00+00:00  150.311264
```

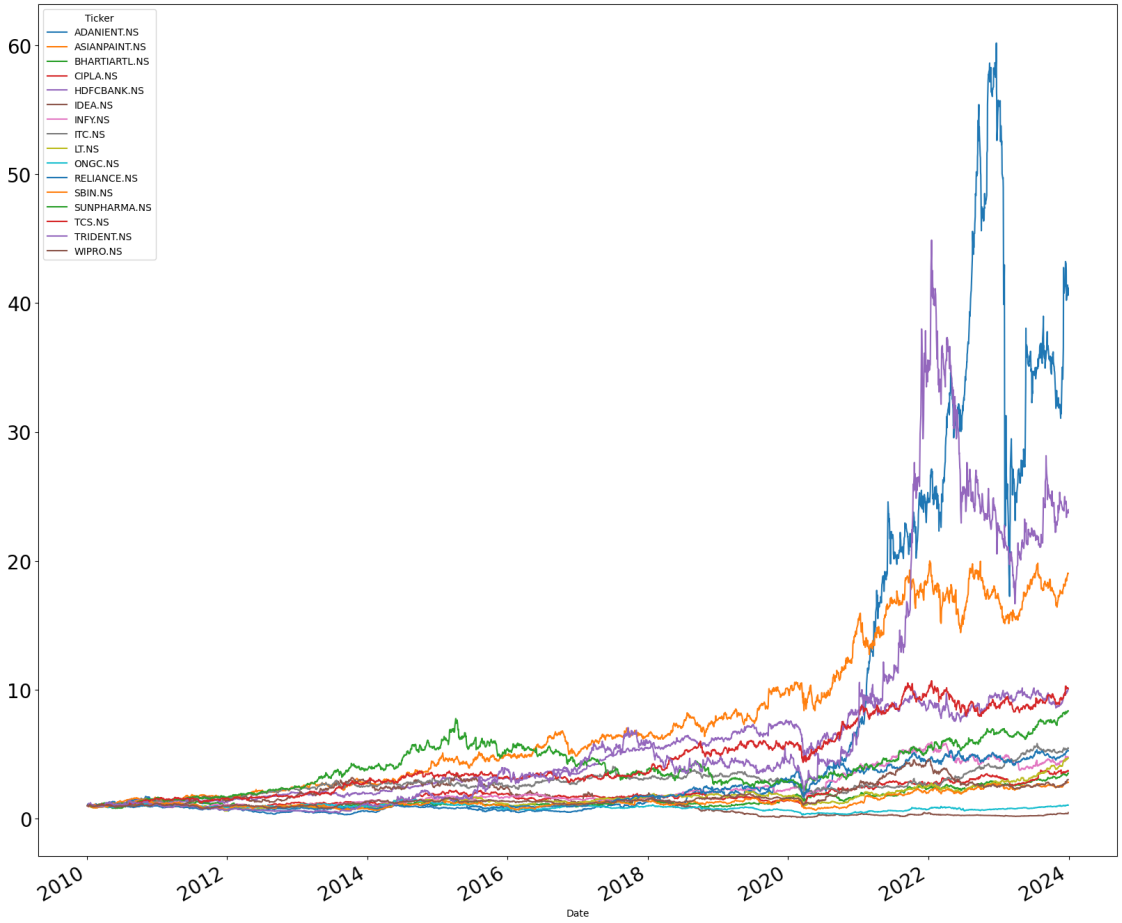
Normalizing the data

[5]:
```
normclose= close.div(close.iloc[0]).copy()
normclose.head()
```

[5]:
```
Ticker                     ADANIENT.NS  ASIANPAINT.NS  BHARTIARTL.NS  \
Date
2010-01-04 00:00:00+00:00     1.000000       1.000000       1.000000
2010-01-05 00:00:00+00:00     1.023140       1.003159       1.015836
2010-01-06 00:00:00+00:00     1.124114       1.000140       1.005074
2010-01-07 00:00:00+00:00     1.077724       0.989937       1.012915
2010-01-08 00:00:00+00:00     1.091342       1.006290       0.999539

Ticker                     CIPLA.NS  HDFCBANK.NS   IDEA.NS   INFY.NS  \
Date
2010-01-04 00:00:00+00:00  1.000000     1.000000  1.000000  1.000000
2010-01-05 00:00:00+00:00  0.982077     1.000879  1.035223  1.003349
2010-01-06 00:00:00+00:00  1.021775     1.001554  1.053265  0.988709
2010-01-07 00:00:00+00:00  1.013331     1.004162  1.048969  0.966489
2010-01-08 00:00:00+00:00  1.010517     1.005482  1.040378  0.943198

Ticker                       ITC.NS     LT.NS   ONGC.NS  RELIANCE.NS  \
```

```
Date
2010-01-04 00:00:00+00:00  1.000000  1.000000  1.000000     1.000000
2010-01-05 00:00:00+00:00  1.009856  1.001774  1.017516     0.995537
2010-01-06 00:00:00+00:00  1.012222  0.990688  1.030359     1.011623
2010-01-07 00:00:00+00:00  1.009068  0.985929  1.031454     1.028405
2010-01-08 00:00:00+00:00  1.011236  0.992166  1.023790     1.025709

Ticker                      SBIN.NS  SUNPHARMA.NS    TCS.NS  TRIDENT.NS  \
Date
2010-01-04 00:00:00+00:00  1.000000      1.000000  1.000000    1.000000
2010-01-05 00:00:00+00:00  1.000371      1.030053  1.000266    0.993399
2010-01-06 00:00:00+00:00  1.006372      1.043421  0.977649    0.990099
2010-01-07 00:00:00+00:00  1.000742      1.027532  0.950442    1.029703
2010-01-08 00:00:00+00:00  0.997752      1.044847  0.931018    1.033003

Ticker                     WIPRO.NS
Date
2010-01-04 00:00:00+00:00  1.000000
2010-01-05 00:00:00+00:00  1.015783
2010-01-06 00:00:00+00:00  0.995244
2010-01-07 00:00:00+00:00  0.977587
2010-01-08 00:00:00+00:00  0.962886
```

[6]:
```
%matplotlib inline
normclose.plot(figsize=(20,18),fontsize=20)
plt.show()
```



[7]:
```
#calculating the percentage change of the close series over period=1 unit
ret = close.pct_change(periods=1)
ret.head()
```

[7]:
```
Ticker                     ADANIENT.NS  ASIANPAINT.NS  BHARTIARTL.NS  \
Date
2010-01-04 00:00:00+00:00          NaN            NaN            NaN
2010-01-05 00:00:00+00:00     0.023140       0.003159       0.015836
2010-01-06 00:00:00+00:00     0.098691      -0.003009      -0.010595
2010-01-07 00:00:00+00:00    -0.041269      -0.010202       0.007802
2010-01-08 00:00:00+00:00     0.012636       0.016519      -0.013206

Ticker                     CIPLA.NS  HDFCBANK.NS   IDEA.NS   INFY.NS  \
Date
2010-01-04 00:00:00+00:00       NaN          NaN       NaN       NaN
2010-01-05 00:00:00+00:00 -0.017923     0.000879  0.035223  0.003349
2010-01-06 00:00:00+00:00  0.040422     0.000674  0.017427 -0.014592
2010-01-07 00:00:00+00:00 -0.008263     0.002605 -0.004078 -0.022473
```

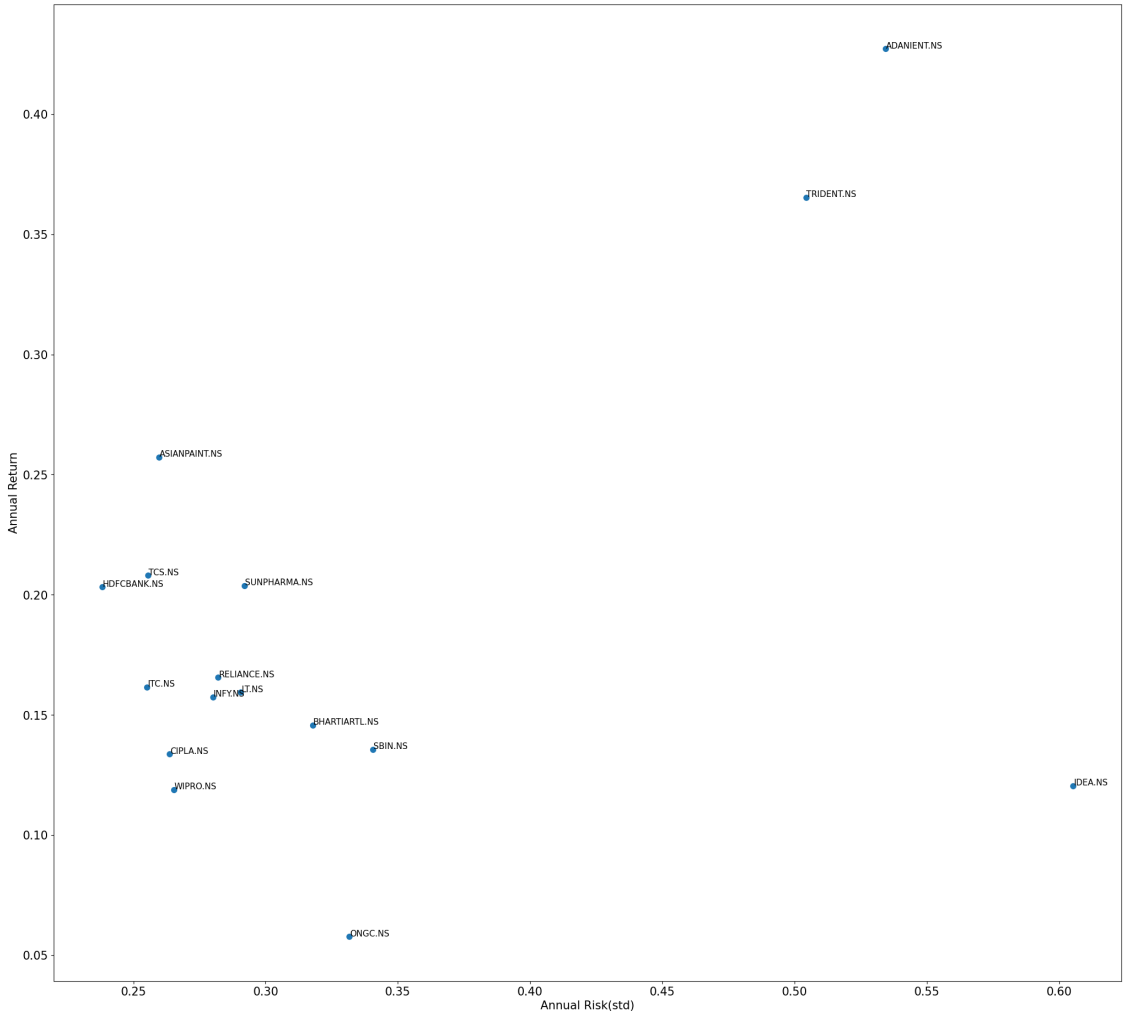
```
2010-01-08 00:00:00+00:00 -0.002777     0.001314 -0.008190 -0.024099

Ticker                       ITC.NS     LT.NS   ONGC.NS  RELIANCE.NS  \
Date
2010-01-04 00:00:00+00:00       NaN       NaN       NaN          NaN
2010-01-05 00:00:00+00:00  0.009856  0.001774  0.017516    -0.004463
2010-01-06 00:00:00+00:00  0.002342 -0.011066  0.012622     0.016158
2010-01-07 00:00:00+00:00 -0.003116 -0.004804  0.001062     0.016590
2010-01-08 00:00:00+00:00  0.002149  0.006326 -0.007430    -0.002622

Ticker                      SBIN.NS  SUNPHARMA.NS    TCS.NS  TRIDENT.NS  \
Date
2010-01-04 00:00:00+00:00       NaN           NaN       NaN         NaN
2010-01-05 00:00:00+00:00  0.000371      0.030053  0.000266   -0.006601
2010-01-06 00:00:00+00:00  0.005999      0.012978 -0.022611   -0.003322
2010-01-07 00:00:00+00:00 -0.005595     -0.015228 -0.027829    0.040000
2010-01-08 00:00:00+00:00 -0.002987      0.016851 -0.020437    0.003205

Ticker                      WIPRO.NS
Date
2010-01-04 00:00:00+00:00       NaN
2010-01-05 00:00:00+00:00  0.015783
2010-01-06 00:00:00+00:00 -0.020220
2010-01-07 00:00:00+00:00 -0.017741
2010-01-08 00:00:00+00:00 -0.015039
```

[8]:
```
summary = ret.describe().T
summary["mean"] = summary["mean"]*262
summary["std"] = summary["std"]*np.sqrt(262)
summary.head()
```

[8]:
```
                count      mean       std       min       25%       50%  \
Ticker
ADANIENT.NS    3453.0  0.427182  0.534390 -0.387493 -0.013025  0.000931
ASIANPAINT.NS  3453.0  0.257271  0.259602 -0.140279 -0.007777  0.000482
BHARTIARTL.NS  3453.0  0.145806  0.317845 -0.119529 -0.010560 -0.000220
CIPLA.NS       3453.0  0.133680  0.263610 -0.088256 -0.008413 -0.000477
HDFCBANK.NS    3453.0  0.203257  0.238146 -0.126069 -0.006990  0.000507

                    75%       max
Ticker
ADANIENT.NS    0.015861  0.273680
ASIANPAINT.NS  0.009473  0.088527
BHARTIARTL.NS  0.010710  0.112908
CIPLA.NS       0.008635  0.130375
HDFCBANK.NS    0.007996  0.115996
```

Plotting Risk vs Return of various stocks

[9]:
```
summary.plot.scatter(x="std",y="mean",figsize=(25,23),s=50,fontsize=15)
for i in summary.index:
    plt.annotate(i,xy = (summary.loc[i,"std"]+0.0002,summary.loc[i,"mean"]+0.
 ↪0002),size=11)

plt.xlabel("Annual Risk(std)",fontsize = 15)
plt.ylabel("Annual Return",fontsize = 15)
plt.show()
```



[10]:
```
ret.cov()
ret.head()
```

[10]:
```
Ticker                     ADANIENT.NS  ASIANPAINT.NS  BHARTIARTL.NS  \
Date
2010-01-04 00:00:00+00:00          NaN            NaN            NaN
2010-01-05 00:00:00+00:00     0.023140       0.003159       0.015836
```

```
2010-01-06 00:00:00+00:00     0.098691      -0.003009      -0.010595
2010-01-07 00:00:00+00:00    -0.041269      -0.010202       0.007802
2010-01-08 00:00:00+00:00     0.012636       0.016519      -0.013206

Ticker                     CIPLA.NS  HDFCBANK.NS   IDEA.NS   INFY.NS  \
Date
2010-01-04 00:00:00+00:00       NaN          NaN       NaN       NaN
2010-01-05 00:00:00+00:00 -0.017923     0.000879  0.035223  0.003349
2010-01-06 00:00:00+00:00  0.040422     0.000674  0.017427 -0.014592
2010-01-07 00:00:00+00:00 -0.008263     0.002605 -0.004078 -0.022473
2010-01-08 00:00:00+00:00 -0.002777     0.001314 -0.008190 -0.024099

Ticker                       ITC.NS     LT.NS   ONGC.NS  RELIANCE.NS  \
Date
2010-01-04 00:00:00+00:00       NaN       NaN       NaN          NaN
2010-01-05 00:00:00+00:00  0.009856  0.001774  0.017516    -0.004463
2010-01-06 00:00:00+00:00  0.002342 -0.011066  0.012622     0.016158
2010-01-07 00:00:00+00:00 -0.003116 -0.004804  0.001062     0.016590
2010-01-08 00:00:00+00:00  0.002149  0.006326 -0.007430    -0.002622

Ticker                      SBIN.NS  SUNPHARMA.NS    TCS.NS  TRIDENT.NS  \
Date
2010-01-04 00:00:00+00:00       NaN           NaN       NaN         NaN
2010-01-05 00:00:00+00:00  0.000371      0.030053  0.000266   -0.006601
2010-01-06 00:00:00+00:00  0.005999      0.012978 -0.022611   -0.003322
2010-01-07 00:00:00+00:00 -0.005595     -0.015228 -0.027829    0.040000
2010-01-08 00:00:00+00:00 -0.002987      0.016851 -0.020437    0.003205

Ticker                     WIPRO.NS
Date
2010-01-04 00:00:00+00:00       NaN
2010-01-05 00:00:00+00:00  0.015783
2010-01-06 00:00:00+00:00 -0.020220
2010-01-07 00:00:00+00:00 -0.017741
2010-01-08 00:00:00+00:00 -0.015039
```

[11]:
```
#Forming a co-relation matrix
ret.corr()
```

[11]:
```
Ticker         ADANIENT.NS  ASIANPAINT.NS  BHARTIARTL.NS  CIPLA.NS  \
Ticker
ADANIENT.NS       1.000000       0.208710       0.222835  0.168680
ASIANPAINT.NS     0.208710       1.000000       0.209627  0.171803
BHARTIARTL.NS     0.222835       0.209627       1.000000  0.178340
CIPLA.NS          0.168680       0.171803       0.178340  1.000000
HDFCBANK.NS       0.283016       0.305503       0.285022  0.179338
IDEA.NS           0.137206       0.110682       0.377980  0.109583
```

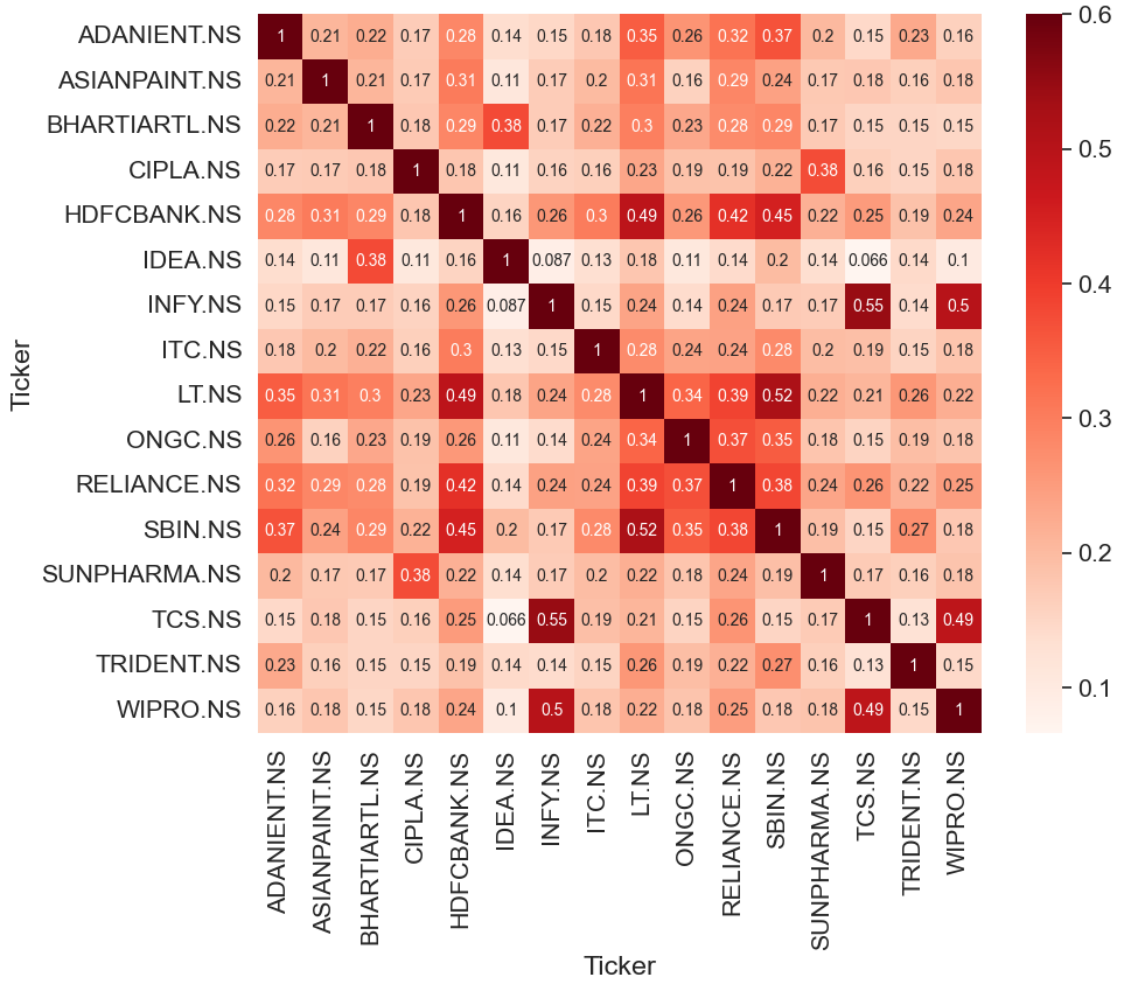
INFY.NS	0.148048	0.168891	0.169675	0.157710
ITC.NS	0.177577	0.201228	0.217966	0.163119
LT.NS	0.351235	0.314665	0.299133	0.233923
ONGC.NS	0.263128	0.158975	0.231987	0.187319
RELIANCE.NS	0.316825	0.285429	0.276749	0.186615
SBIN.NS	0.366567	0.243246	0.285073	0.216124
SUNPHARMA.NS	0.201836	0.174525	0.173284	0.376125
TCS.NS	0.146186	0.178479	0.153556	0.163640
TRIDENT.NS	0.229360	0.163831	0.152296	0.151563
WIPRO.NS	0.160620	0.179699	0.150915	0.179108

Ticker	HDFCBANK.NS	IDEA.NS	INFY.NS	ITC.NS	LT.NS	ONGC.NS	\
Ticker							
ADANIENT.NS	0.283016	0.137206	0.148048	0.177577	0.351235	0.263128	
ASIANPAINT.NS	0.305503	0.110682	0.168891	0.201228	0.314665	0.158975	
BHARTIARTL.NS	0.285022	0.377980	0.169675	0.217966	0.299133	0.231987	
CIPLA.NS	0.179338	0.109583	0.157710	0.163119	0.233923	0.187319	
HDFCBANK.NS	1.000000	0.155571	0.259544	0.302594	0.492956	0.256670	
IDEA.NS	0.155571	1.000000	0.087404	0.134508	0.178599	0.108454	
INFY.NS	0.259544	0.087404	1.000000	0.146126	0.237867	0.139013	
ITC.NS	0.302594	0.134508	0.146126	1.000000	0.277815	0.238287	
LT.NS	0.492956	0.178599	0.237867	0.277815	1.000000	0.339297	
ONGC.NS	0.256670	0.108454	0.139013	0.238287	0.339297	1.000000	
RELIANCE.NS	0.419414	0.141410	0.241203	0.241047	0.388942	0.371325	
SBIN.NS	0.449914	0.198983	0.172428	0.280301	0.519080	0.353225	
SUNPHARMA.NS	0.216819	0.136630	0.169516	0.196294	0.219882	0.181026	
TCS.NS	0.252949	0.065945	0.546335	0.191833	0.207964	0.154815	
TRIDENT.NS	0.187023	0.144900	0.139864	0.154335	0.258332	0.194504	
WIPRO.NS	0.235939	0.100910	0.501027	0.181384	0.221430	0.184390	

Ticker	RELIANCE.NS	SBIN.NS	SUNPHARMA.NS	TCS.NS	TRIDENT.NS	\
Ticker						
ADANIENT.NS	0.316825	0.366567	0.201836	0.146186	0.229360	
ASIANPAINT.NS	0.285429	0.243246	0.174525	0.178479	0.163831	
BHARTIARTL.NS	0.276749	0.285073	0.173284	0.153556	0.152296	
CIPLA.NS	0.186615	0.216124	0.376125	0.163640	0.151563	
HDFCBANK.NS	0.419414	0.449914	0.216819	0.252949	0.187023	
IDEA.NS	0.141410	0.198983	0.136630	0.065945	0.144900	
INFY.NS	0.241203	0.172428	0.169516	0.546335	0.139864	
ITC.NS	0.241047	0.280301	0.196294	0.191833	0.154335	
LT.NS	0.388942	0.519080	0.219882	0.207964	0.258332	
ONGC.NS	0.371325	0.353225	0.181026	0.154815	0.194504	
RELIANCE.NS	1.000000	0.383600	0.236101	0.262873	0.217081	
SBIN.NS	0.383600	1.000000	0.191234	0.152227	0.270948	
SUNPHARMA.NS	0.236101	0.191234	1.000000	0.172277	0.162587	
TCS.NS	0.262873	0.152227	0.172277	1.000000	0.126495	
TRIDENT.NS	0.217081	0.270948	0.162587	0.126495	1.000000	

```
WIPRO.NS          0.248127  0.176763      0.183777  0.489727    0.145823

Ticker         WIPRO.NS
Ticker
ADANIENT.NS    0.160620
ASIANPAINT.NS  0.179699
BHARTIARTL.NS  0.150915
CIPLA.NS       0.179108
HDFCBANK.NS    0.235939
IDEA.NS        0.100910
INFY.NS        0.501027
ITC.NS         0.181384
LT.NS          0.221430
ONGC.NS        0.184390
RELIANCE.NS    0.248127
SBIN.NS        0.176763
SUNPHARMA.NS   0.183777
TCS.NS         0.489727
TRIDENT.NS     0.145823
WIPRO.NS       1.000000
```

[12]:
```
import seaborn as sns
plt.figure(figsize=(10,8))
sns.set(font_scale=1.4)
sns.heatmap(ret.corr(),cmap="Reds",annot=True,annot_kws={"size":10},vmax=0.6)
plt.show()
```



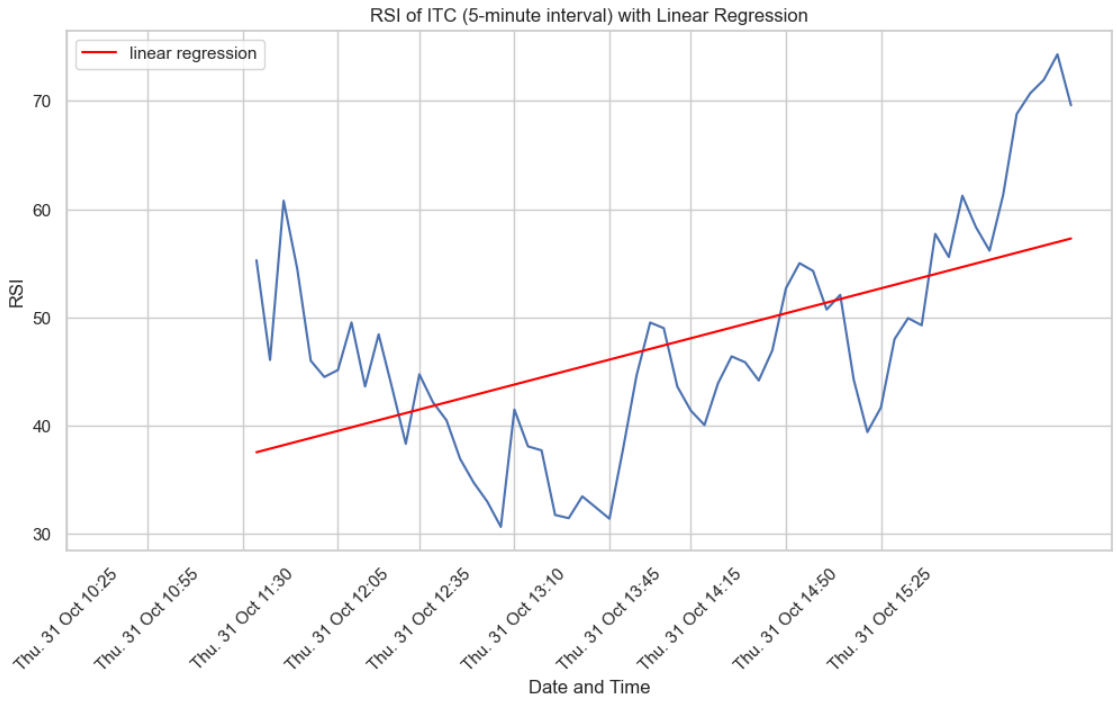
Predictive model for RSI

[13]:
```
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler

# Function to fetch historical data from Yahoo Finance
def fetch_historical_data(symbol, period="1d", interval="5m"):    #******Change␣
 ↪your timeframe here****
    data = yf.download(tickers=symbol, period=period, interval=interval)
    return data

# Function to calculate the selected indicator
def calculate_indicator(data, indicator='RSI', length=14):
    if indicator == 'RSI':
        data[indicator] = ta.rsi(data['Close'], length=length)
    elif indicator == 'SMA':
```

```
        data[indicator] = ta.sma(data['Close'], length=length)
    return data[indicator]

# Function to plot the selected indicator
def plot_indicator(data, indicator, model='linear'):
    # Get the last 100 bars of the selected indicator
    history = data[indicator].tail(100).reset_index()
    history['index'] = history.index

    # Drop rows with NaN values
    history = history.dropna(subset=[indicator])

    # Remove timezone information from the 'Datetime' column if present
    if 'Datetime' not in history.columns:
        history['Datetime'] = data.index[-len(history):]

    # Convert Datetime to timezone-naive (removing timezone information)
    if history['Datetime'].dtype == 'datetime64[ns, Asia/Kolkata]':
        history['Datetime'] = history['Datetime'].dt.tz_localize(None)

    # Create the regression model and fit it to the data
    if model == 'linear':
        reg = LinearRegression()
    elif model == 'svr':
        reg = SVR(kernel='rbf', C=1e3, gamma=0.1)
        scaler = StandardScaler()
        history[indicator] = scaler.fit_transform(history[[indicator]])

    reg.fit(history[['index']], history[indicator])
    history[f'{model}_pred'] = reg.predict(history[['index']])

    # Plot the indicator and the regression line
    sns.set(style="whitegrid")
    plt.figure(figsize=(12, 6))
    sns.lineplot(x='index', y=indicator, data=history)
    sns.lineplot(x='index', y=f'{model}_pred', data=history, color='red',␣
 ↪label=f'{model} regression')

    # Set the number of ticks and labels dynamically
    num_ticks = 10  # Set the number of ticks you want (e.g., 10 evenly spaced␣
 ↪ticks)
    tick_positions = np.linspace(0, len(history) - 1, num_ticks, dtype=int)
    tick_labels = history['Datetime'].iloc[tick_positions].dt.strftime('%a. %d␣
 ↪%b %H:%M')

    # Set the plot title, axes labels, and tick labels
```

```
    plt.title(f"{indicator} of ITC (5-minute interval) with {model.
 ↪capitalize()} Regression")
    plt.xlabel("Date and Time")
    plt.ylabel(indicator)
    plt.xticks(tick_positions, tick_labels, rotation=45)
    plt.legend()
    plt.show()

if __name__ == "__main__":
    # Set the symbol and indicator to be plotted
    symbol = "ITC.NS"  #*******Change your Stock here**********
    indicator = 'RSI'

    # Fetch the historical data and calculate the selected indicator
    data = fetch_historical_data(symbol)
    indicator_data = calculate_indicator(data, indicator=indicator)

    # Plot the indicator with support vector regression
    plot_indicator(data, indicator, 'linear')
```

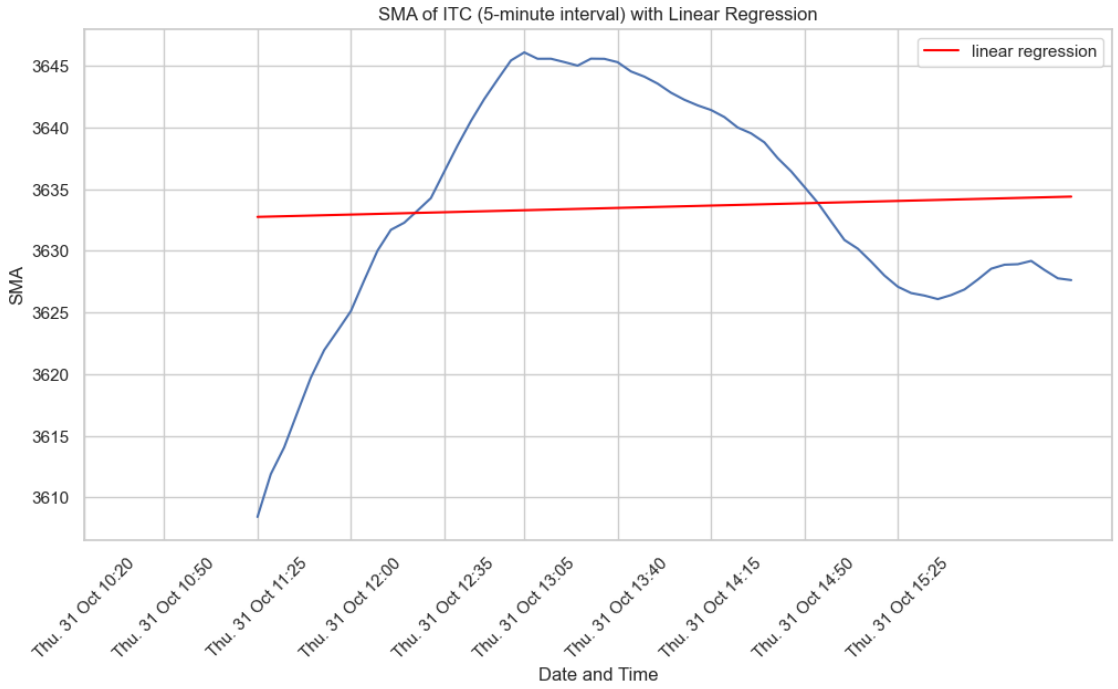
```
[**********************100%**********************]  1 of 1 completed
```



Predictive model for SMA

[14]:
```
if __name__ == "__main__":
    # Set the symbol and indicator to be plotted
    symbol = "LT.NS"  #*******Change your Stock here**********
    indicator = 'SMA'

    # Fetch the historical data and calculate the selected indicator
    data = fetch_historical_data(symbol)
    indicator_data = calculate_indicator(data, indicator=indicator)

    # Plot the indicator with support vector regression
    plot_indicator(data, indicator, 'linear')
```

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



[15]:
```
#shivsharan
```

[16]:
```
x = close.columns
x
```

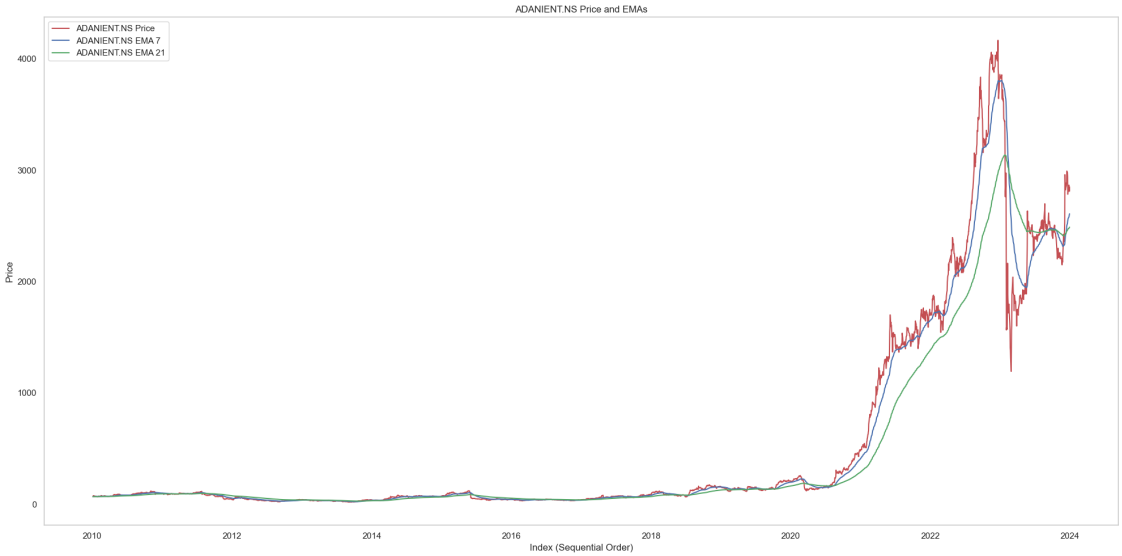
[16]:
```
Index(['ADANIENT.NS', 'ASIANPAINT.NS', 'BHARTIARTL.NS', 'CIPLA.NS',
       'HDFCBANK.NS', 'IDEA.NS', 'INFY.NS', 'ITC.NS', 'LT.NS', 'ONGC.NS',
       'RELIANCE.NS', 'SBIN.NS', 'SUNPHARMA.NS', 'TCS.NS', 'TRIDENT.NS',
       'WIPRO.NS'],
      dtype='object', name='Ticker')
```

[17]:
```
for company in x: #FIND THE EMA's
    close[f"EMA_{company}_7"] = close[company].ewm(span=50, adjust=False).mean()
    close[f"EMA_{company}_21"] = close[company].ewm(span=200, adjust=False).
 ↪mean()
```

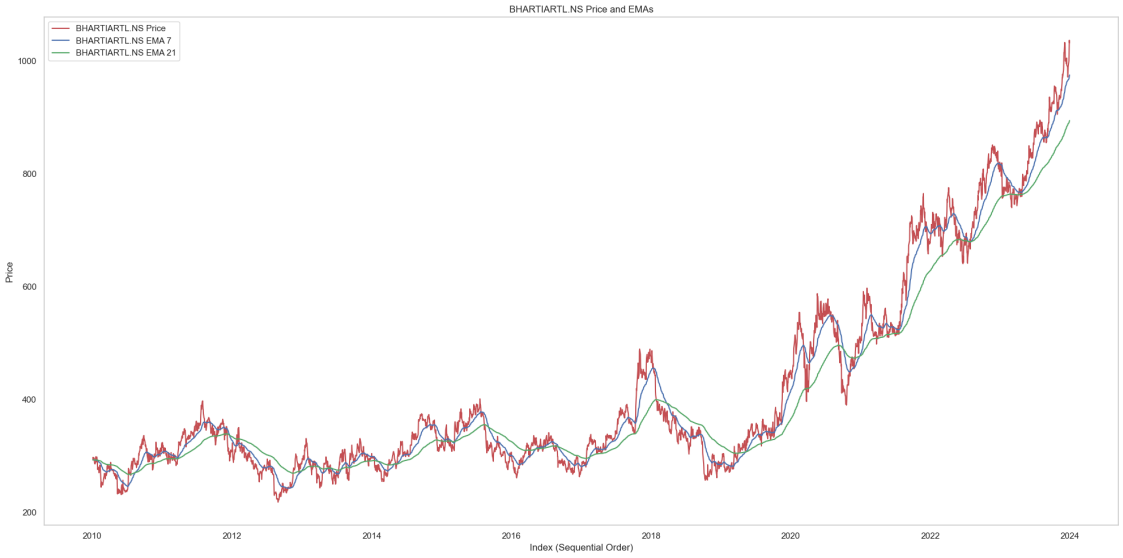
[18]:
```
close.columns
```

[18]:
```
Index(['ADANIENT.NS', 'ASIANPAINT.NS', 'BHARTIARTL.NS', 'CIPLA.NS',
       'HDFCBANK.NS', 'IDEA.NS', 'INFY.NS', 'ITC.NS', 'LT.NS', 'ONGC.NS',
       'RELIANCE.NS', 'SBIN.NS', 'SUNPHARMA.NS', 'TCS.NS', 'TRIDENT.NS',
       'WIPRO.NS', 'EMA_ADANIENT.NS_7', 'EMA_ADANIENT.NS_21',
       'EMA_ASIANPAINT.NS_7', 'EMA_ASIANPAINT.NS_21', 'EMA_BHARTIARTL.NS_7',
       'EMA_BHARTIARTL.NS_21', 'EMA_CIPLA.NS_7', 'EMA_CIPLA.NS_21',
       'EMA_HDFCBANK.NS_7', 'EMA_HDFCBANK.NS_21', 'EMA_IDEA.NS_7',
       'EMA_IDEA.NS_21', 'EMA_INFY.NS_7', 'EMA_INFY.NS_21', 'EMA_ITC.NS_7',
       'EMA_ITC.NS_21', 'EMA_LT.NS_7', 'EMA_LT.NS_21', 'EMA_ONGC.NS_7',
       'EMA_ONGC.NS_21', 'EMA_RELIANCE.NS_7', 'EMA_RELIANCE.NS_21',
       'EMA_SBIN.NS_7', 'EMA_SBIN.NS_21', 'EMA_SUNPHARMA.NS_7',
       'EMA_SUNPHARMA.NS_21', 'EMA_TCS.NS_7', 'EMA_TCS.NS_21',
       'EMA_TRIDENT.NS_7', 'EMA_TRIDENT.NS_21', 'EMA_WIPRO.NS_7',
       'EMA_WIPRO.NS_21'],
      dtype='object', name='Ticker')
```

[19]:
```
# ALL COMPANY EMA
for company in x:
    plt.figure(figsize=(20, 10))
    plt.plot(close.index,close[company], label=f"{company} Price", color='r')
    plt.plot(close.index,close[f"EMA_{company}_7"],label=f"{company} EMA 7" ,␣
 ↪color='b')
    plt.plot(close.index,close[f"EMA_{company}_21"],label=f"{company} EMA 21" ,␣
 ↪color='g')
    plt.title(f"{company} Price and EMAs")
    plt.xlabel('Index (Sequential Order)')
    plt.ylabel('Price')
    plt.legend()
    plt.grid()
    plt.tight_layout()
    plt.show()
```


ADANIENT.NS Price and EMAs
ADANIENT.NS Price
ADANIENT.NS EMA 7
ADANIENT.NS EMA 21
Price
Index (Sequential Order)
0
1000
2000
3000
4000
2010
2012
2014
2016
2018
2020
2022
2024


ASIANPAINT.NS Price and EMAs
ASIANPAINT.NS Price
ASIANPAINT.NS EMA 7
ASIANPAINT.NS EMA 21
Price
Index (Sequential Order)
500
1000
1500
2000
2500
3000
3500
2010
2012
2014
2016
2018
2020
2022
2024


BHARTIARTL.NS Price and EMAs
BHARTIARTL.NS Price
BHARTIARTL.NS EMA 7
BHARTIARTL.NS EMA 21
Price
1000
800
600
400
200
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)


CIPLA.NS Price and EMAs
CIPLA.NS Price
CIPLA.NS EMA 7
CIPLA.NS EMA 21
Price
1200
1000
800
600
400
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)


HDFCBANK.NS Price and EMAs
HDFCBANK.NS Price
HDFCBANK.NS EMA 7
HDFCBANK.NS EMA 21
Price
Index (Sequential Order)

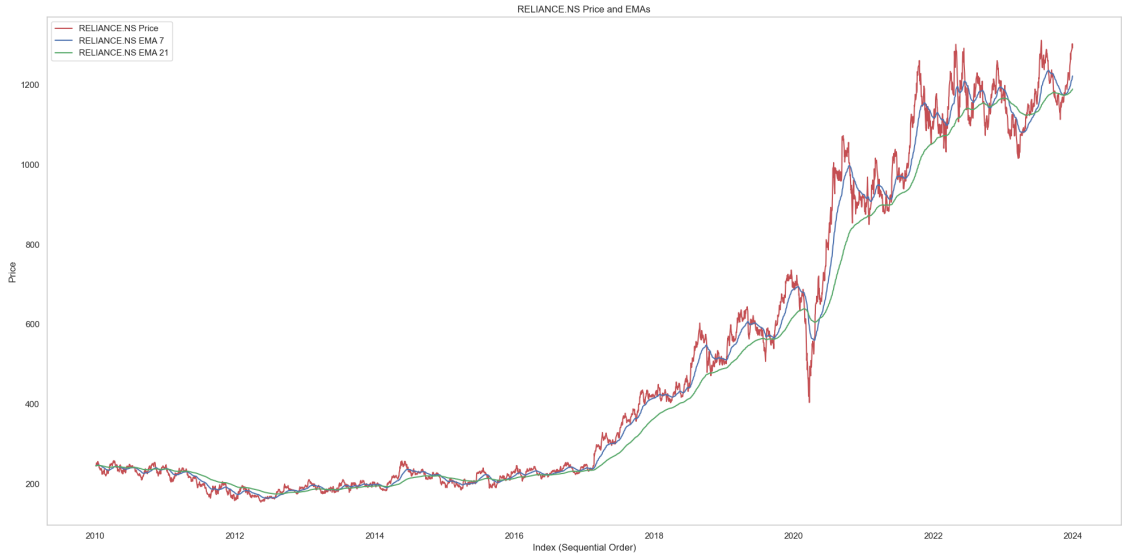

IDEA.NS Price and EMAs
IDEA.NS Price
IDEA.NS EMA 7
IDEA.NS EMA 21
Price
Index (Sequential Order)


INFY.NS Price and EMAs
INFY.NS Price
INFY.NS EMA 7
INFY.NS EMA 21
Price
Index (Sequential Order)
2000
1750
1500
1250
1000
750
500
250
2010
2012
2014
2016
2018
2020
2022
2024


ITC.NS Price and EMAs
ITC.NS Price
ITC.NS EMA 7
ITC.NS EMA 21
Price
Index (Sequential Order)
500
400
300
200
100
2010
2012
2014
2016
2018
2020
2022
2024


LT.NS Price and EMAs
LT.NS Price
LT.NS EMA 7
LT.NS EMA 21
Price
3500
3000
2500
2000
1500
1000
500
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)

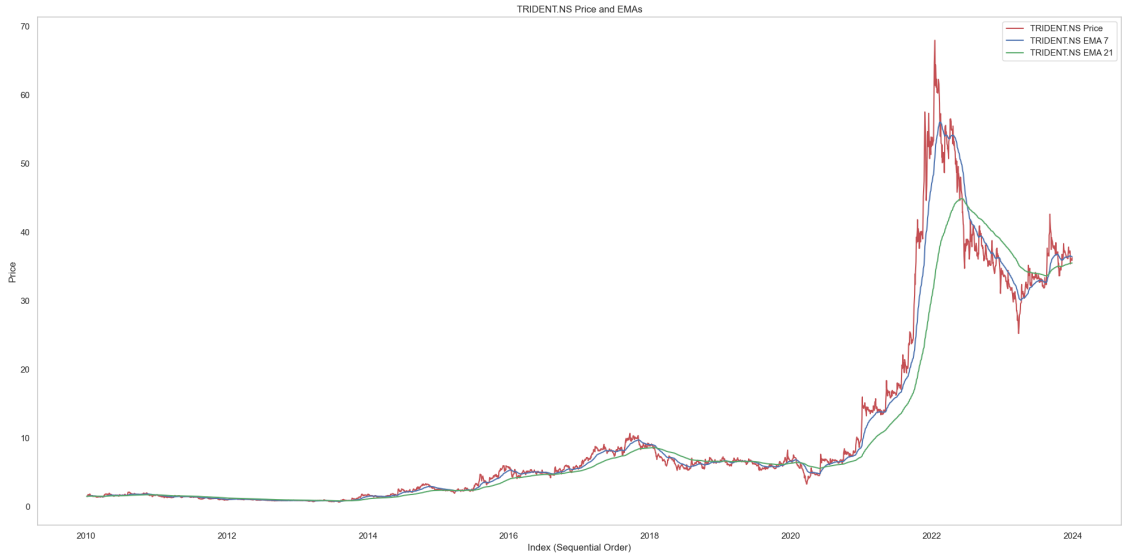

ONGC.NS Price and EMAs
ONGC.NS Price
ONGC.NS EMA 7
ONGC.NS EMA 21
Price
300
250
200
150
100
50
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)


RELIANCE.NS Price and EMAs
RELIANCE.NS Price
RELIANCE.NS EMA 7
RELIANCE.NS EMA 21
Price
Index (Sequential Order)


SBIN.NS Price and EMAs
SBIN.NS Price
SBIN.NS EMA 7
SBIN.NS EMA 21
Price
Index (Sequential Order)

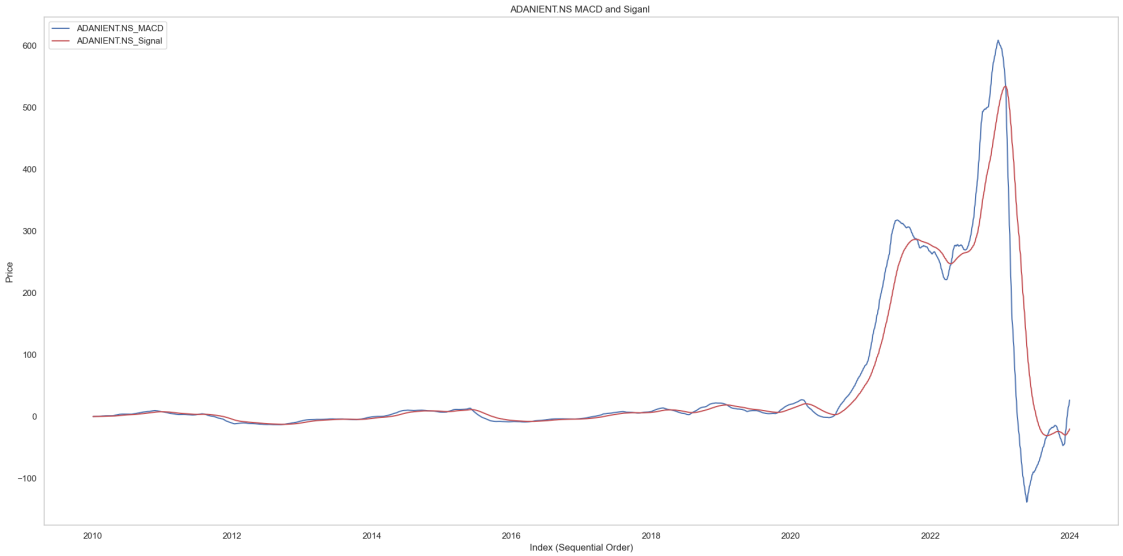

SUNPHARMA.NS Price and EMAs
SUNPHARMA.NS Price
SUNPHARMA.NS EMA 7
SUNPHARMA.NS EMA 21
Price
Index (Sequential Order)
200
400
600
800
1000
1200
2010
2012
2014
2016
2018
2020
2022
2024

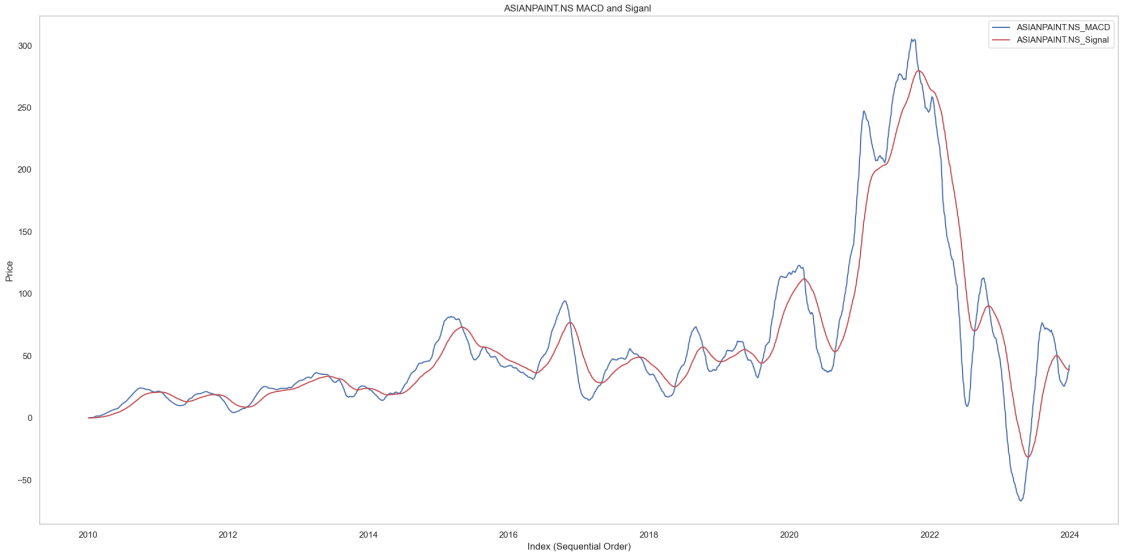

TCS.NS Price and EMAs
TCS.NS Price
TCS.NS EMA 7
TCS.NS EMA 21
Price
Index (Sequential Order)
500
1000
1500
2000
2500
3000
3500
4000
2010
2012
2014
2016
2018
2020
2022
2024

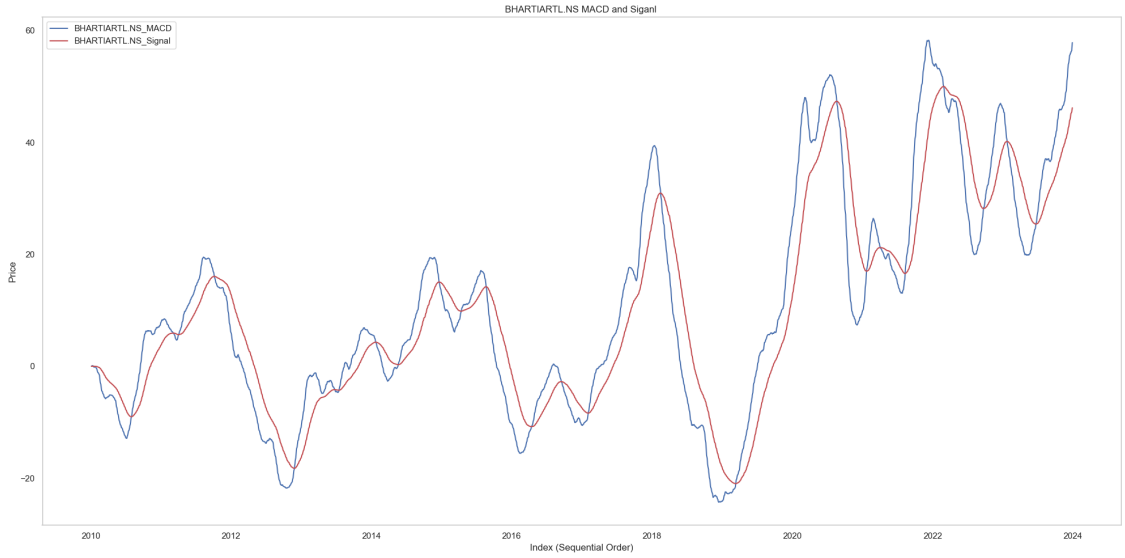


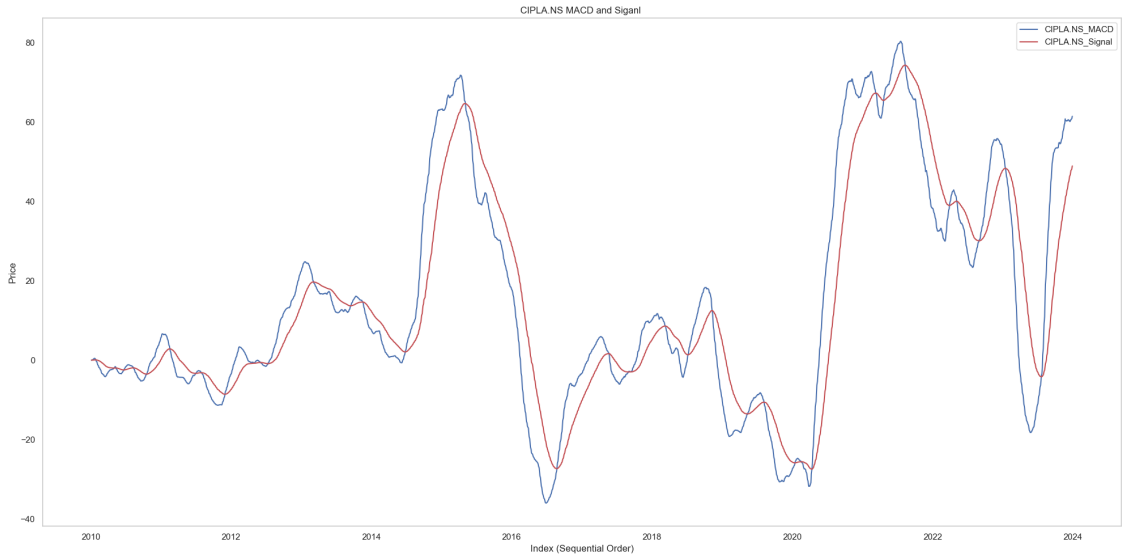
[20]:
```
#FIND ALL COMPANY MACD for signal line = 90
for company in x:
    plt.figure(figsize=(20,10))
    close[f"MACD_{company}"] = close[company].ewm(span=120, adjust=False).
 ↪mean()-close[company].ewm(span=260, adjust=False).mean();
    plt.plot(close.
 ↪index,close[f"MACD_{company}"],label=f"{company}_MACD",color='b')
    plt.plot(close.index,close[f"MACD_{company}"].ewm(span=90,adjust=False).
 ↪mean(),label=f"{company}_Signal",color='r')
    plt.title(f"{company} MACD and Siganl ")
```

```
    plt.xlabel('Index (Sequential Order)')
    plt.ylabel('Price')
    plt.legend()
    plt.grid()
    plt.tight_layout()
    plt.show()
```

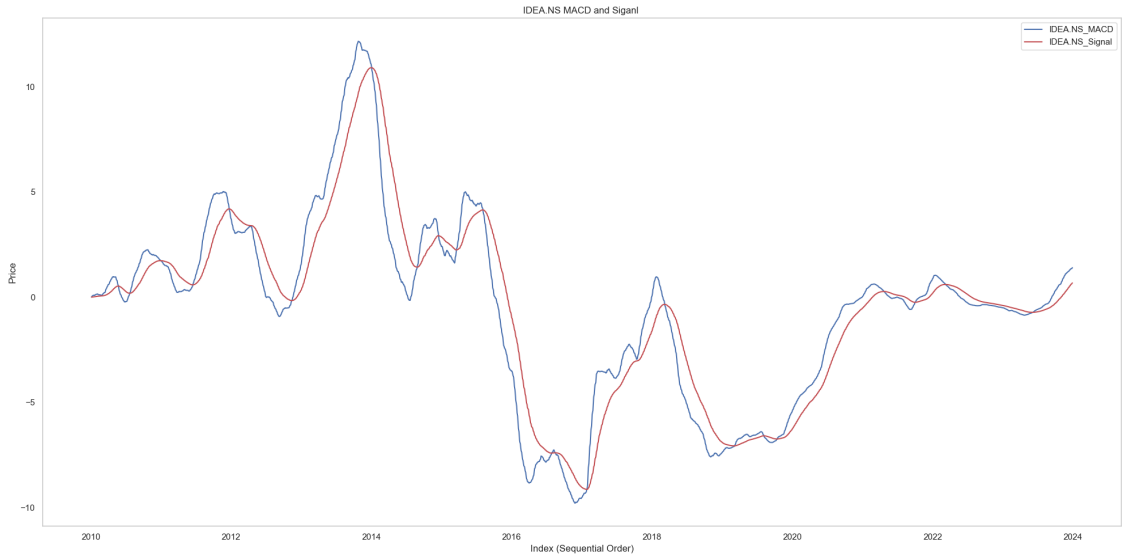


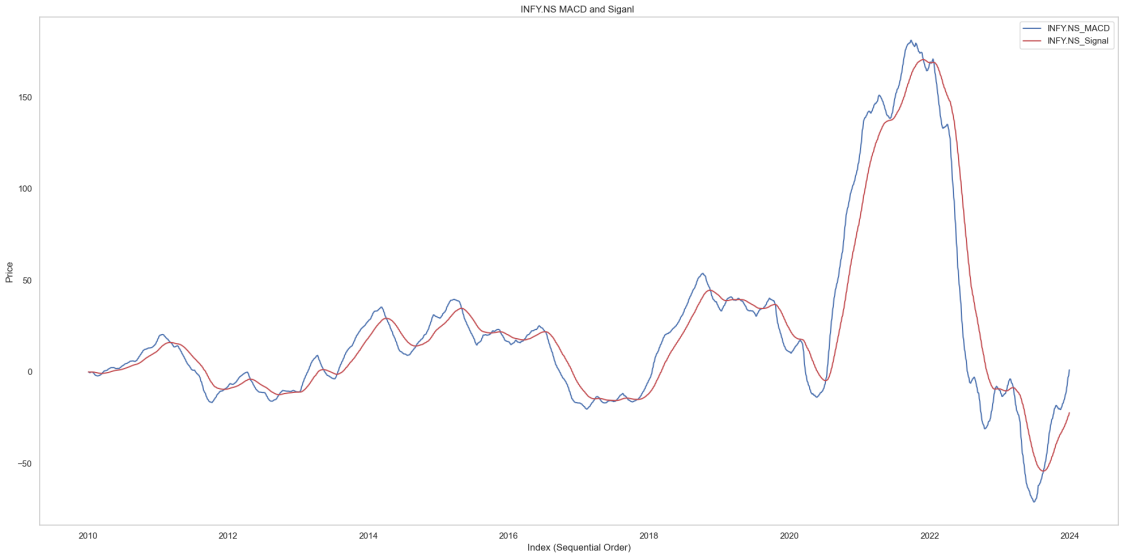



BHARTIARTL.NS MACD and Siganl
BHARTIARTL.NS_MACD
BHARTIARTL.NS_Signal
Price
Index (Sequential Order)

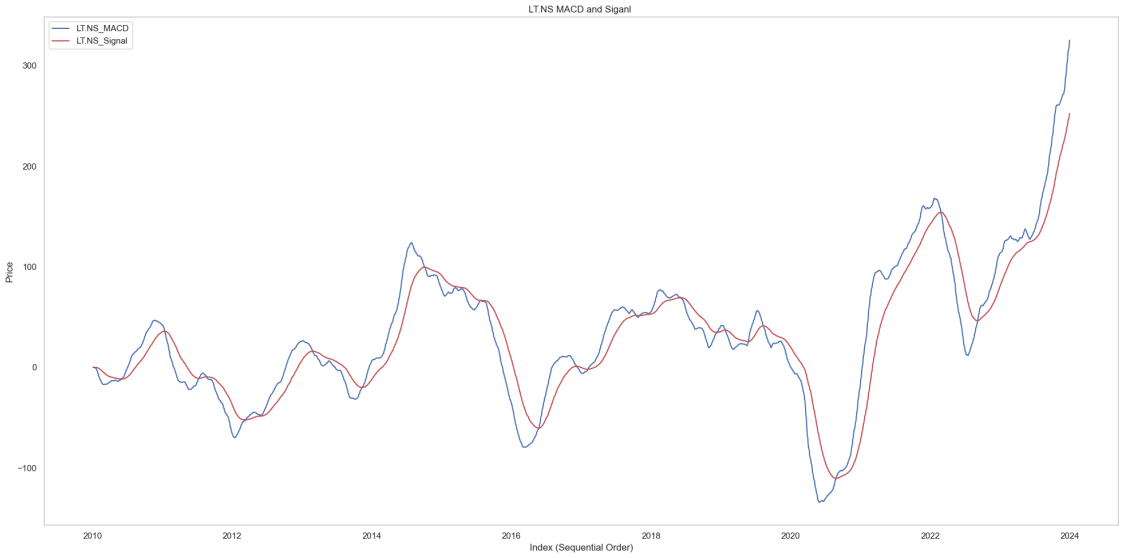

CIPLA.NS MACD and Siganl
CIPLA.NS_MACD
CIPLA.NS_Signal
Price
Index (Sequential Order)

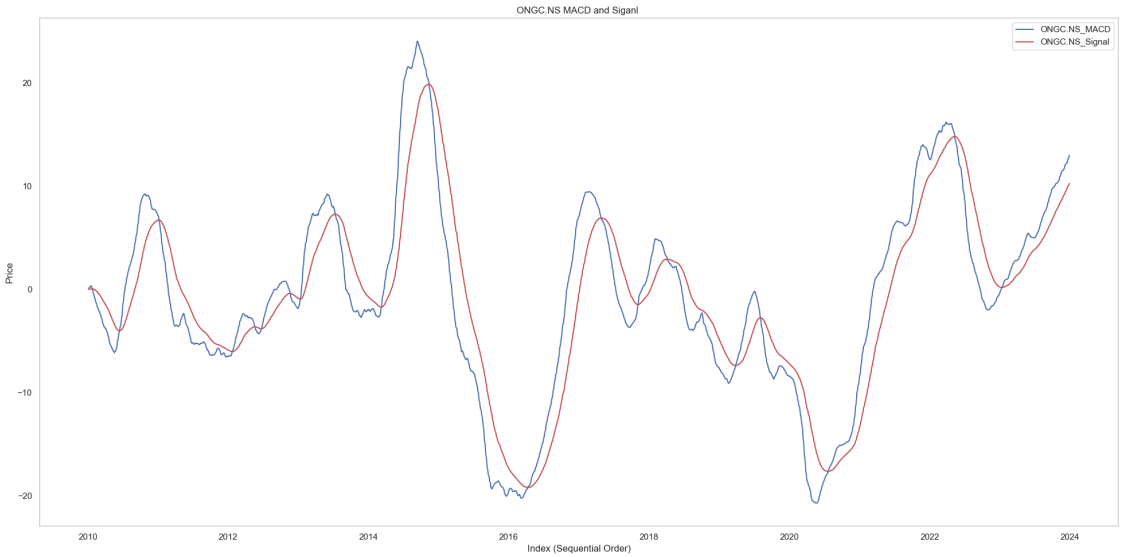

HDFCBANK.NS MACD and Siganl
HDFCBANK.NS_MACD
HDFCBANK.NS_Signal
Price
125
100
75
50
25
0
−25
−50
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)

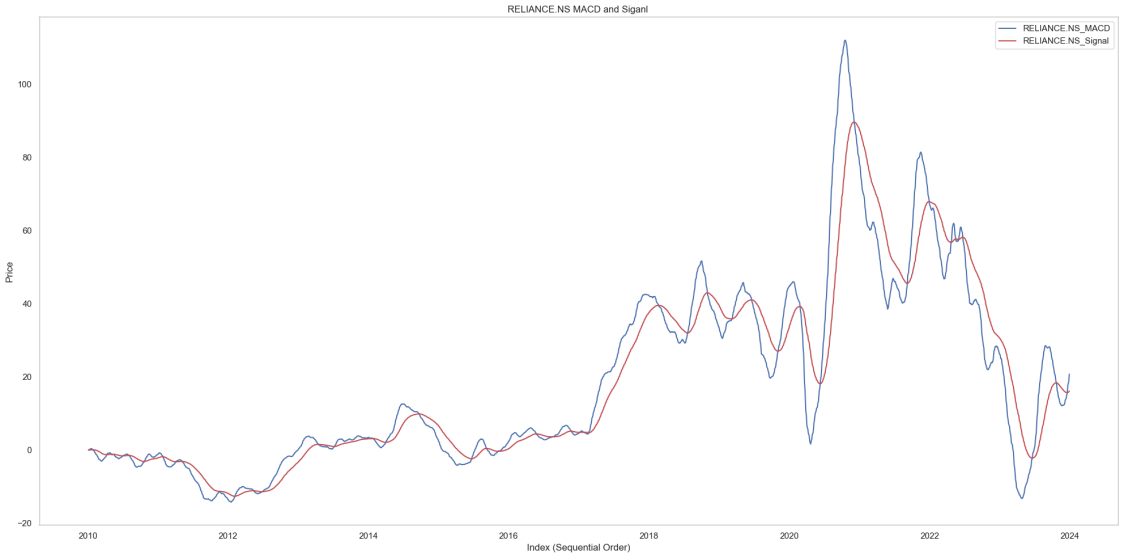

IDEA.NS MACD and Siganl
IDEA.NS_MACD
IDEA.NS_Signal
Price
10
5
0
−5
−10
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)

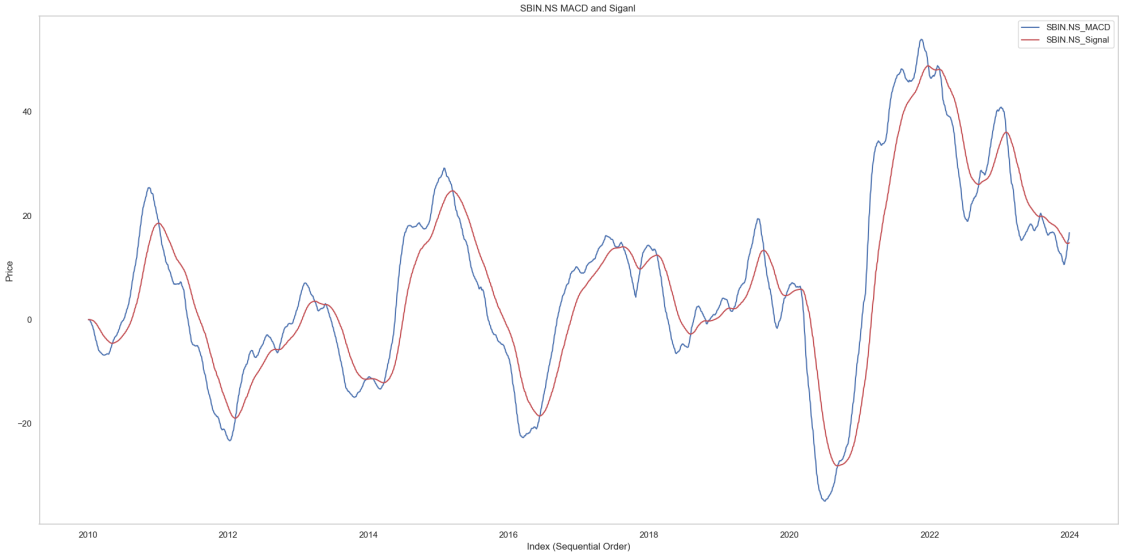

INFY.NS MACD and Siganl
INFY.NS_MACD
INFY.NS_Signal
150
100
50
0
−50
Price
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)


ITC.NS MACD and Siganl
ITC.NS_MACD
ITC.NS_Signal
40
30
20
10
0
−10
−20
−30
Price
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)


LT.NS MACD and Siganl
LT.NS_MACD
LT.NS_Signal
300
200
100
0
−100
Price
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)

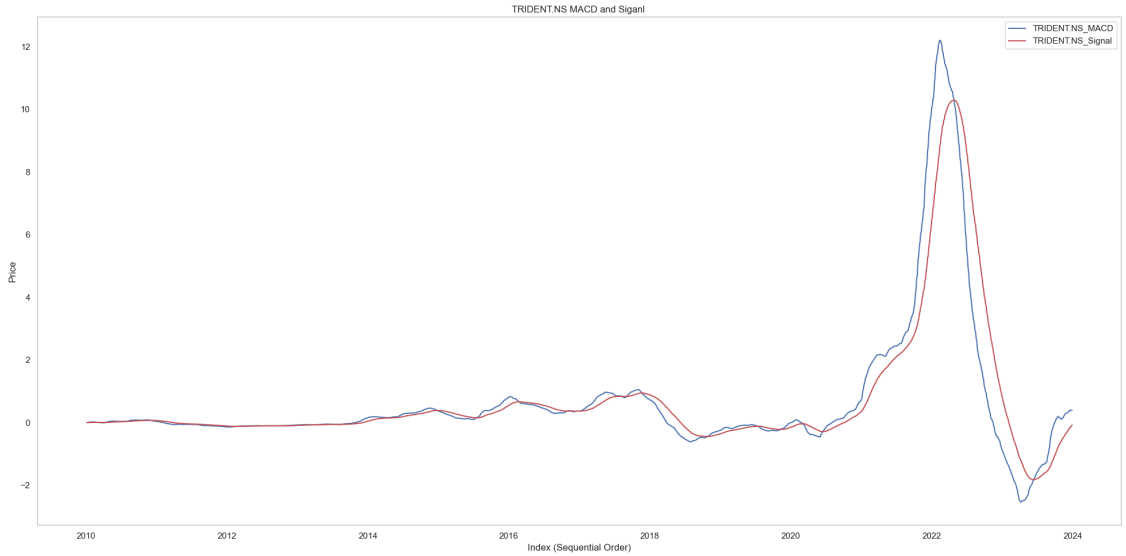

ONGC.NS MACD and Siganl
ONGC.NS_MACD
ONGC.NS_Signal
20
10
0
−10
−20
Price
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)


RELIANCE.NS MACD and Siganl
RELIANCE.NS_MACD
RELIANCE.NS_Signal
Price
Index (Sequential Order)


SBIN.NS MACD and Siganl
SBIN.NS_MACD
SBIN.NS_Signal
Price
Index (Sequential Order)


SUNPHARMA.NS MACD and Siganl
SUNPHARMA.NS_MACD
SUNPHARMA.NS_Signal
Price
100
75
50
25
0
−25
−50
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)


TCS.NS MACD and Siganl
TCS.NS_MACD
TCS.NS_Signal
Price
300
250
200
150
100
50
0
−50
2010
2012
2014
2016
2018
2020
2022
2024
Index (Sequential Order)



[21]:
```
close = close.iloc[:,:16]
```

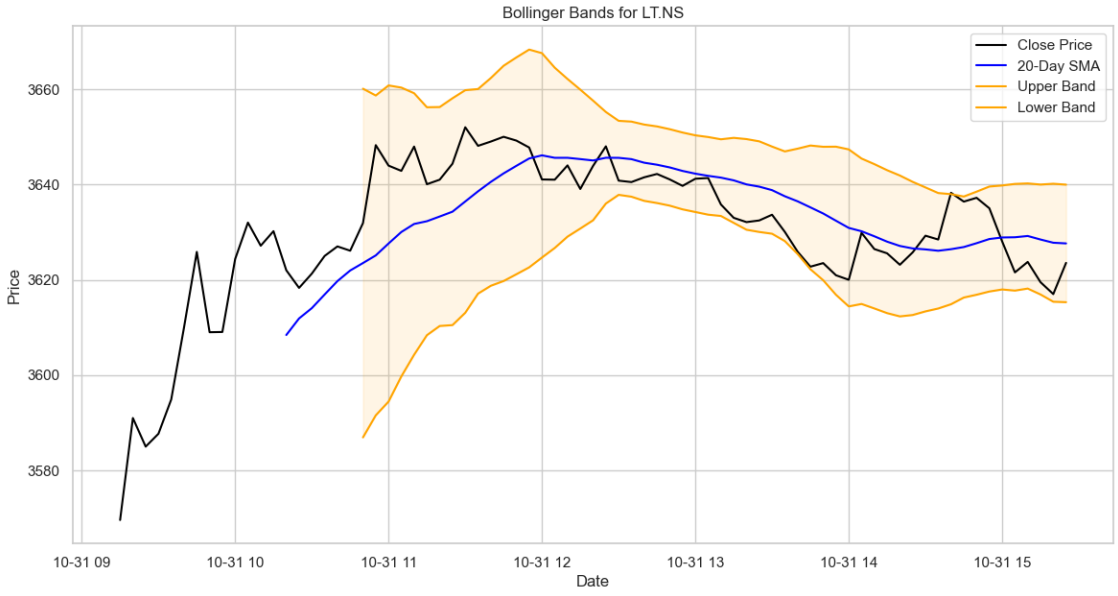
[22]:
```
close.columns
```

[22]:
```
Index(['ADANIENT.NS', 'ASIANPAINT.NS', 'BHARTIARTL.NS', 'CIPLA.NS',
       'HDFCBANK.NS', 'IDEA.NS', 'INFY.NS', 'ITC.NS', 'LT.NS', 'ONGC.NS',
       'RELIANCE.NS', 'SBIN.NS', 'SUNPHARMA.NS', 'TCS.NS', 'TRIDENT.NS',
       'WIPRO.NS'],
      dtype='object', name='Ticker')
```

[30]:
```
def calculate_bollinger_bands(data,window):
    data['STD'] = data['Close'].rolling(window=window).std()
    data['Upper_Band'] = data['SMA'] + (data['STD'] * 2)
    data['Lower_Band'] = data['SMA'] - (data['STD'] * 2)
    return data
```

[33]:
```
if __name__ == "__main__":
    # Set the symbol and indicator to be plotted
    symbol = "LT.NS"  #*******Change your Stock here*********
    indicator = 'SMA'

    # Fetch the historical data and calculate the selected indicator
    data = fetch_historical_data(symbol)
    indicator_data = calculate_indicator(data, indicator=indicator)
    bollinger  = calculate_bollinger_bands(data,20)
    plt.figure(figsize=(14, 7))
    plt.plot(data['Close'], label='Close Price', color='black')
    plt.plot(data['SMA'], label='20-Day SMA', color='blue')
    plt.plot(data['Upper_Band'], label='Upper Band', color='orange')
    plt.plot(data['Lower_Band'], label='Lower Band', color='orange')
    plt.fill_between(data.index,
                     data['Upper_Band'],
                     data['Lower_Band'],
                     color='orange', alpha=0.1)
    plt.title(f'Bollinger Bands for {symbol}')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.show()
```

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


Bollinger Bands for LT.NS
Close Price
20-Day SMA
Upper Band
Lower Band
Price
3660
3640
3620
3600
3580
10-31 09
10-31 10
10-31 11
10-31 12
10-31 13
10-31 14
10-31 15
Date

[ ]:

[ ]: