Install all the necessary liberaries

In [1]:
```
import pandas as pd
import pandas_datareader as web
import yfinance as yf
import datetime
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
from pandas.plotting import scatter_matrix
from mplfinance.original_flavor import candlestick_ohlc
from matplotlib.dates import DateFormatter, date2num, WeekdayLocator, DayLocator, M
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 2
      1 import pandas as pd
----> 2 import pandas_datareader as web
      3 import yfinance as yf
      4 import datetime

ModuleNotFoundError: No module named 'pandas_datareader'
```

In [11]:
```
!pip install yfinance
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting yfinance
  Downloading yfinance-0.2.32-py2.py3-none-any.whl (68 kB)
     ---------------------------------------- 69.0/69.0 kB ? eta 0:00:00
Collecting peewee>=3.16.2
  Downloading peewee-3.17.0.tar.gz (2.9 MB)
     ---------------------------------------- 2.9/2.9 MB 4.9 MB/s eta 0:00:00
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Collecting multitasking>=0.0.7
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Requirement already satisfied: lxml>=4.9.1 in c:\programdata\anaconda3\lib\site-pac
kages (from yfinance) (4.9.1)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\programdata\anaconda3\l
ib\site-packages (from yfinance) (4.11.1)
Collecting frozendict>=2.3.4
  Downloading frozendict-2.3.9-cp310-cp310-win_amd64.whl (35 kB)
Requirement already satisfied: numpy>=1.16.5 in c:\programdata\anaconda3\lib\site-p
ackages (from yfinance) (1.23.5)
Requirement already satisfied: pandas>=1.3.0 in c:\programdata\anaconda3\lib\site-p
ackages (from yfinance) (1.5.3)
Requirement already satisfied: pytz>=2022.5 in c:\programdata\anaconda3\lib\site-pa
ckages (from yfinance) (2022.7)
Collecting html5lib>=1.1
  Downloading html5lib-1.1-py2.py3-none-any.whl (112 kB)
     ---------------------------------------- 112.2/112.2 kB ? eta 0:00:00
Requirement already satisfied: appdirs>=1.4.4 in c:\programdata\anaconda3\lib\site-
packages (from yfinance) (1.4.4)
Collecting requests>=2.31
  Downloading requests-2.31.0-py3-none-any.whl (62 kB)
     ---------------------------------------- 62.6/62.6 kB 3.3 MB/s eta 0:00:00
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\site-p
ackages (from beautifulsoup4>=4.11.1->yfinance) (2.3.2.post1)
Requirement already satisfied: six>=1.9 in c:\programdata\anaconda3\lib\site-packag
es (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in c:\programdata\anaconda3\lib\site-pa
ckages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\programdata\anaconda3\l
ib\site-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda
3\lib\site-packages (from requests>=2.31->yfinance) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\s
ite-packages (from requests>=2.31->yfinance) (2022.12.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\s
ite-packages (from requests>=2.31->yfinance) (1.26.14)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-pa
ckages (from requests>=2.31->yfinance) (3.4)
Building wheels for collected packages: peewee
  Building wheel for peewee (pyproject.toml): started
  Building wheel for peewee (pyproject.toml): finished with status 'done'
  Created wheel for peewee: filename=peewee-3.17.0-py3-none-any.whl size=135766 sha
256=a1671b24a5213e35d7cbc07eeb97a60852b707df8951e681eda4d4fa3de3b735
  Stored in directory: c:\users\reham\appdata\local\pip\cache\wheels\e2\b9\da\71651
4851b65304b2d24f2a161398b9470da589b08a5a586c8
Successfully built peewee
```

```
Installing collected packages: peewee, multitasking, requests, html5lib, frozendic
t, yfinance
Successfully installed frozendict-2.3.9 html5lib-1.1 multitasking-0.0.11 peewee-3.1
7.0 requests-2.31.0 yfinance-0.2.32
  WARNING: The script sample.exe is installed in 'C:\Users\Reham\AppData\Roaming\Py
thon\Python310\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warnin
g, use --no-warn-script-location.
ERROR: pip's dependency resolver does not currently take into account all the packa
ges that are installed. This behaviour is the source of the following dependency co
nflicts.
conda-repo-cli 1.0.41 requires requests_mock, which is not installed.
conda-repo-cli 1.0.41 requires clyent==1.2.1, but you have clyent 1.2.2 which is in
compatible.
conda-repo-cli 1.0.41 requires nbformat==5.4.0, but you have nbformat 5.7.0 which i
s incompatible.
conda-repo-cli 1.0.41 requires requests==2.28.1, but you have requests 2.31.0 which
is incompatible.
```

In [13]:
```
!pip install mplfinance
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting mplfinance
  Downloading mplfinance-0.12.10b0-py3-none-any.whl (75 kB)
     ---------------------------------------- 75.0/75.0 kB 518.2 kB/s eta 0:00:00
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-pack
ages (from mplfinance) (3.7.0)
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages
(from mplfinance) (1.5.3)
Requirement already satisfied: pillow>=6.2.0 in c:\programdata\anaconda3\lib\site-p
ackages (from matplotlib->mplfinance) (9.4.0)
Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda3\lib\site
-packages (from matplotlib->mplfinance) (22.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anaconda3\lib\si
te-packages (from matplotlib->mplfinance) (1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib->mplfinance) (3.0.9)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\site-pa
ckages (from matplotlib->mplfinance) (0.11.0)
Requirement already satisfied: numpy>=1.20 in c:\programdata\anaconda3\lib\site-pac
kages (from matplotlib->mplfinance) (1.23.5)
Requirement already satisfied: contourpy>=1.0.1 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib->mplfinance) (1.0.5)
Requirement already satisfied: python-dateutil>=2.7 in c:\programdata\anaconda3\li
b\site-packages (from matplotlib->mplfinance) (2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\lib\si
te-packages (from matplotlib->mplfinance) (4.25.0)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-pa
ckages (from pandas->mplfinance) (2022.7)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packag
es (from python-dateutil>=2.7->matplotlib->mplfinance) (1.16.0)
Installing collected packages: mplfinance
Successfully installed mplfinance-0.12.10b0
```

In [12]:
```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
from pandas.plotting import scatter_matrix
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[12], line 5
      3 get_ipython().run_line_magic('matplotlib', 'inline')
      4 from pandas.plotting import scatter_matrix
----> 5 from mplfinance.original_flavor import candlestick_ohlc
      6 from matplotlib.dates import DateFormatter, date2num, WeekdayLocator, DayLo
cator, MONDAY

ModuleNotFoundError: No module named 'mplfinance'
```

In [14]:
```
from matplotlib.dates import DateFormatter, date2num, WeekdayLocator, DayLocator, M
```

In [5]:
```
!pip install pandas_datareader
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting pandas_datareader
  Downloading pandas_datareader-0.10.0-py3-none-any.whl (109 kB)
     ------------------------------------ 109.5/109.5 kB 2.2 MB/s eta 0:00:00
Requirement already satisfied: pandas>=0.23 in c:\programdata\anaconda3\lib\site-pa
ckages (from pandas_datareader) (1.5.3)
Requirement already satisfied: requests>=2.19.0 in c:\programdata\anaconda3\lib\sit
e-packages (from pandas_datareader) (2.28.1)
Requirement already satisfied: lxml in c:\programdata\anaconda3\lib\site-packages
(from pandas_datareader) (4.9.1)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-pa
ckages (from pandas>=0.23->pandas_datareader) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in c:\programdata\anaconda3\lib\site-p
ackages (from pandas>=0.23->pandas_datareader) (1.23.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\programdata\anaconda3\l
ib\site-packages (from pandas>=0.23->pandas_datareader) (2.8.2)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\programdata\anaconda
3\lib\site-packages (from requests>=2.19.0->pandas_datareader) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda3\li
b\site-packages (from requests>=2.19.0->pandas_datareader) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\s
ite-packages (from requests>=2.19.0->pandas_datareader) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-pa
ckages (from requests>=2.19.0->pandas_datareader) (3.4)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packag
es (from python-dateutil>=2.8.1->pandas>=0.23->pandas_datareader) (1.16.0)
Installing collected packages: pandas_datareader
Successfully installed pandas_datareader-0.10.0
```

write function for date & time

In [8]:
```
import datetime

start = datetime.datetime(2016, 1, 1)
end = datetime.datetime(2022, 12, 31)
```

Download the Data from yahoo finance using stock ticker name & display the prices

In [17]:
```
price_rsnl = yfinance.download('RELIANCE.NS',start,end)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[17], line 1
----> 1 price_rsnl = yfinance.download('RELIANCE.NS',start,end)

NameError: name 'yfinance' is not defined
```

Display the last 5 row with prices [OHLC]

In [5]:
```
price_rsnl.tail()
```

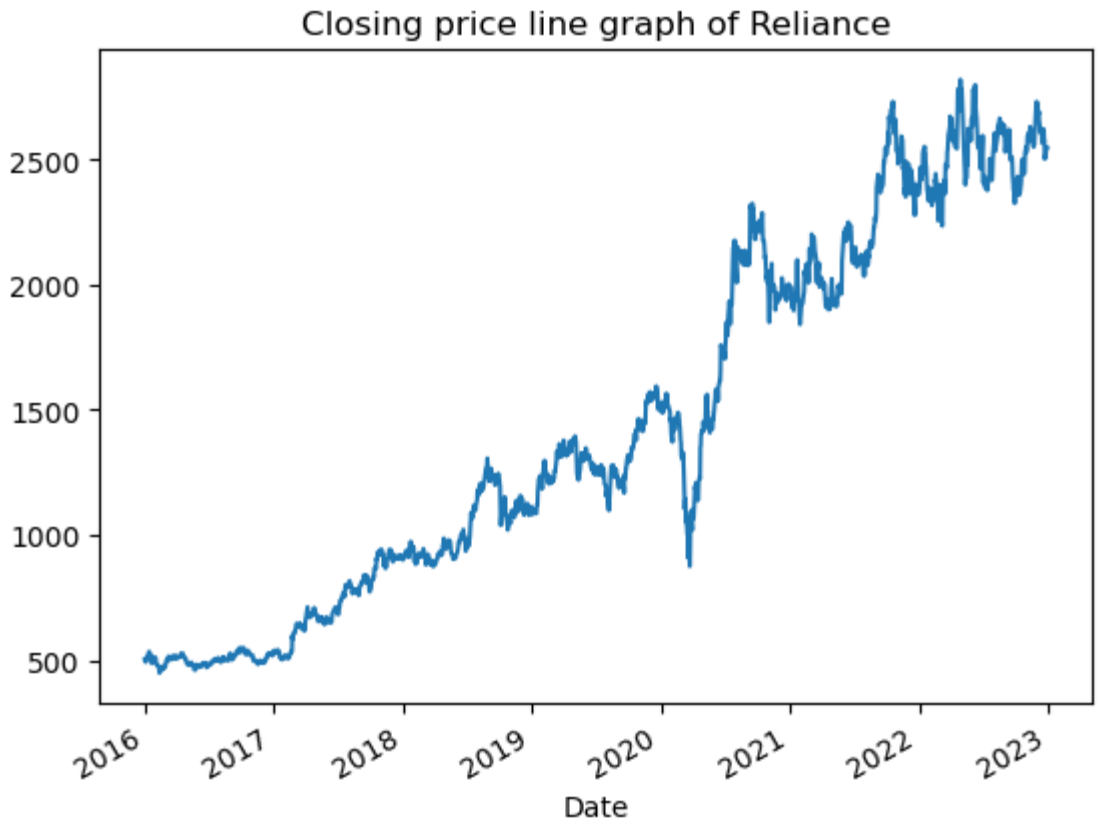
Out[5]:

Date	Open	High	Low	Close	Adj Close	Volume
2022-12-26	2514.750000	2542.000000	2492.399902	2524.050049	2524.050049	2764496
2022-12-27	2530.000000	2548.800049	2515.250000	2544.699951	2544.699951	2659749
2022-12-28	2538.000000	2549.800049	2521.500000	2544.449951	2544.449951	3442509
2022-12-29	2527.000000	2548.899902	2525.500000	2543.300049	2543.300049	3198493
2022-12-30	2545.100098	2577.000000	2541.100098	2547.199951	2547.199951	3364092

Plot the close price of the stock

In [7]:
```
price_rsnl.Close.plot()
plt.title('Closing price line graph of Reliance')
```

Out[7]:
```
Text(0.5, 1.0, 'Closing price line graph of Reliance')
```



Adjusted Data means corporate actions like stock splits, dividends, right offerings etc .....

In [9]:
```
price_rsnl.tail()
```

Out[9]:

Date	Open	High	Low	Close	Adj Close	Volume
2022-12-26	2514.750000	2542.000000	2492.399902	2524.050049	2524.050049	2764496
2022-12-27	2530.000000	2548.800049	2515.250000	2544.699951	2544.699951	2659749
2022-12-28	2538.000000	2549.800049	2521.500000	2544.449951	2544.449951	3442509
2022-12-29	2527.000000	2548.899902	2525.500000	2543.300049	2543.300049	3198493
2022-12-30	2545.100098	2577.000000	2541.100098	2547.199951	2547.199951	3364092

Inorder to temporary delete the Adjusted close price columns, add auto_adjust function to true & display last 5 rows

In [26]:
```
price_tcs = yf.download('TCS.NS',start,end,auto_adjust=True)
price_tcs.tail()
```

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

Out[26]:

Date	Open	High	Low	Close	Volume
2022-12-26	3140.893574	3183.360993	3137.682924	3164.778320	870157
2022-12-27	3180.636885	3185.112366	3143.958231	3171.199707	835883
2022-12-28	3161.762457	3177.620935	3138.607155	3168.864746	910795
2022-12-29	3143.569256	3182.874721	3140.553141	3180.199219	1037927
2022-12-30	3197.030340	3209.629474	3158.259789	3168.475342	1163131

For multiple data we need to use list of ticker, pass the list to the download & fetch the values

In [27]:
```
tickers_list = ['TCS.NS','INFY.NS','HCLTECH.NS','HDFCBANK.NS']
price_list = yf.download(tickers_list,start,end,auto_adjust=True)
price_list.tail()
```

```
[**********100%**********]  4 of 4 completed
```

Out[27]:

	Close					
	HCLTECH.NS	HDFCBANK.NS	INFY.NS	TCS.NS	HCLTECH.NS	HDFCBANK.NS
Date						
2022-12-26	1005.268311	1610.975464	1462.794922	3164.778320	1015.060882	1620.417236
2022-12-27	1009.799194	1612.606812	1474.916626	3171.199707	1014.086492	1617.401799
2022-12-28	1008.922241	1611.321533	1470.340576	3168.864746	1011.699321	1614.485204
2022-12-29	1017.691650	1622.691162	1477.545532	3180.199219	1019.201988	1624.866170
2022-12-30	1012.673706	1609.690308	1468.441895	3168.475342	1028.215028	1626.349241

Fetching minute level data, mention the time period and intervals

In [28]:
```
price_tcs = yf.download('TCS.NS',period="5d",interval="1m",auto_adjust=True)
price_tcs.tail()
```

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

Out[28]:

Datetime	Open	High	Low	Close	Volume
2023-07-07 15:25:00+05:30	3322.000000	3324.550049	3321.850098	3321.850098	5152
2023-07-07 15:26:00+05:30	3322.199951	3324.050049	3322.199951	3322.899902	3786
2023-07-07 15:27:00+05:30	3322.649902	3324.000000	3322.149902	3323.000000	7165
2023-07-07 15:28:00+05:30	3322.899902	3323.199951	3321.100098	3321.899902	4163
2023-07-07 15:29:00+05:30	3321.800049	3324.000000	3319.699951	3321.699951	6792

Fetching fundamental data

In [14]:
```
price_tcs = yf.Ticker("TCS.NS")
price_tcs.info
```

Out[14]:

```
{'address1': 'TCS House',
 'address2': 'Raveline Street Fort',
 'city': 'Mumbai',
 'zip': '400001',
 'country': 'India',
 'phone': '91 22 6778 9595',
 'website': 'https://www.tcs.com',
 'industry': 'Information Technology Services',
 'industryDisp': 'Information Technology Services',
 'sector': 'Technology',
 'longBusinessSummary': 'Tata Consultancy Services Limited provides information tec
hnology (IT) and IT enabled services worldwide. It operates through Banking, Financ
ial Services and Insurance; Manufacturing; Retail and Consumer Business; Communicat
ion, Media and Technology; Life Sciences and Healthcare; and Others segments. The c
ompany provides TCS ADD, a suite of technology platforms for clinical research and
drug development; TCS BaNCS, a financial solution platform; TCS BFSI Platforms, a c
loud-native, as-a-service that helps FIs and insurance firms; TCS CHROMA, a cloud-b
ased talent management solution; customer intelligence and insight solutions; TCS E
RP on Cloud, a hosted ERP applications and services platform; TCS HOBS, a connected
devices management platform; and ignio, an autonomous enterprise software. It also
offers TCS Intelligent Urban Exchange, smart cities and enterprises solution; TCS O
mniStore, a commerce platform; TCS OPTUMERA, a retail-connected strategic intellige
nce platform; TCS TAP, a procurement offering; TCS MasterCraft, a platform of intel
ligent automation products; Quartz- the Smart Ledgers, a blockchain solution; Jile,
an enterprise agile planning and delivery tool; TCS iON, an IT-as-a-Service model t
hat provides business solutions; and TCS TwinX, an enterprise digital twin platfor
m. In addition, the company provides cloud, cognitive business, consulting, cyberse
curity, data and analytics, enterprise solutions, Internet of Things and digital en
gineering, TCS interactive, and sustainability services. It serves capital markets;
communications, media, and information services; public services; energy, resource,
and utility industries; consumer goods and distribution industries; and banking, ed
ucation, healthcare, high tech, insurance, life science, manufacturing, retail, tra
vel, and logistics industries. The company was founded in 1968 and is headquartered
in Mumbai, India. Tata Consultancy Services Limited is a subsidiary of Tata Sons Pr
ivate Limited.',
 'fullTimeEmployees': 614795,
 'companyOfficers': [{'maxAge': 1,
   'name': 'Mr. Ganapathy Subramaniam Natarajan M.Sc.',
   'age': 63,
   'title': 'COO & Non-Independent Exec. Director',
   'yearBorn': 1959,
   'fiscalYear': 2023,
   'totalPay': 236020000,
   'exercisedValue': 0,
   'unexercisedValue': 0},
  {'maxAge': 1,
   'name': 'Mr. K.  Krithivasan B.E, M.Tech.',
   'age': 58,
   'title': 'MD, CEO & Director',
   'yearBorn': 1964,
   'exercisedValue': 0,
   'unexercisedValue': 0},
  {'maxAge': 1,
   'name': 'Mr. Samir  Seksaria',
   'title': 'Chief Financial Officer',
   'exercisedValue': 0,
   'unexercisedValue': 0},
  {'maxAge': 1,
   'name': 'Mr. K. Ananth Krishnan M.Sc (Engg.), M.Tech.',
```

```
  'age': 59,
  'title': 'Chief Technology Officer',
  'yearBorn': 1963,
  'exercisedValue': 0,
  'unexercisedValue': 0},
 {'maxAge': 1,
  'name': 'Mr. Kedar  Shirali',
  'title': 'Head of Global Investor Relations',
  'exercisedValue': 0,
  'unexercisedValue': 0},
 {'maxAge': 1,
  'name': 'Mr. Madhav  Anchan',
  'title': 'Gen. Counsel Legal',
  'exercisedValue': 0,
  'unexercisedValue': 0},
 {'maxAge': 1,
  'name': 'Mr. Pradeep Manohar Gaitonde',
  'title': 'Company Sec. & Compliance Officer',
  'exercisedValue': 0,
  'unexercisedValue': 0},
 {'maxAge': 1,
  'name': 'Mr. Vivek  Padiyar',
  'title': 'Head of Corp. Communications',
  'exercisedValue': 0,
  'unexercisedValue': 0},
 {'maxAge': 1,
  'name': 'Ms. Rajashree  R.',
  'title': 'Chief Marketing Officer',
  'exercisedValue': 0,
  'unexercisedValue': 0},
 {'maxAge': 1,
  'name': 'Mr. Milind  Lakkad M.Tech.',
  'age': 58,
  'title': 'Chief HR Officer',
  'yearBorn': 1964,
  'exercisedValue': 0,
  'unexercisedValue': 0}],
'auditRisk': 1,
'boardRisk': 10,
'compensationRisk': 1,
'shareHolderRightsRisk': 5,
'overallRisk': 5,
'governanceEpochDate': 1688169600,
'compensationAsOfEpochDate': 1703980800,
'maxAge': 86400,
'priceHint': 2,
'previousClose': 3322.9,
'open': 3302.0,
'dayLow': 3302.0,
'dayHigh': 3356.9,
'regularMarketPreviousClose': 3322.9,
'regularMarketOpen': 3302.0,
'regularMarketDayLow': 3302.0,
'regularMarketDayHigh': 3356.9,
'dividendRate': 96.0,
'dividendYield': 0.029000001,
'exDividendDate': 1686787200,
'payoutRatio': 0.3993,
'fiveYearAvgDividendYield': 1.27,
```

```
'beta': 0.565128,
'trailingPE': 28.78978,
'forwardPE': 23.1955,
'volume': 1787817,
'regularMarketVolume': 1787817,
'averageVolume': 1934281,
'averageVolume10days': 1475494,
'averageDailyVolume10Day': 1475494,
'bid': 0.0,
'ask': 0.0,
'bidSize': 0,
'askSize': 0,
'marketCap': 12181892497408,
'fiftyTwoWeekLow': 2926.1,
'fiftyTwoWeekHigh': 3575.0,
'priceToSalesTrailing12Months': 5.4031763,
'fiftyDayAverage': 3254.437,
'twoHundredDayAverage': 3272.7822,
'trailingAnnualDividendRate': 48.0,
'trailingAnnualDividendYield': 0.014445214,
'currency': 'INR',
'enterpriseValue': 11712542539776,
'profitMargins': 0.18694,
'floatShares': 995993784,
'sharesOutstanding': 3659049984,
'heldPercentInsiders': 0.72318,
'heldPercentInstitutions': 0.15995,
'impliedSharesOutstanding': 3659049984,
'bookValue': 247.124,
'priceToBook': 13.471982,
'lastFiscalYearEnd': 1680220800,
'nextFiscalYearEnd': 1711843200,
'mostRecentQuarter': 1680220800,
'earningsQuarterlyGrowth': 0.148,
'netIncomeToCommon': 421470011392,
'trailingEps': 115.64,
'forwardEps': 143.53,
'pegRatio': 2.47,
'lastSplitFactor': '2:1',
'lastSplitDate': 1527724800,
'enterpriseToRevenue': 5.195,
'enterpriseToEbitda': 20.325,
'52WeekChange': 1.7593265,
'SandP52WeekChange': 13.135683,
'lastDividendValue': 24.0,
'lastDividendDate': 1686787200,
'exchange': 'NSI',
'quoteType': 'EQUITY',
'symbol': 'TCS.NS',
'underlyingSymbol': 'TCS.NS',
'shortName': 'TATA CONSULTANCY S',
'longName': 'Tata Consultancy Services Limited',
'firstTradeDateEpochUtc': 1029123900,
'timeZoneFullName': 'Asia/Kolkata',
'timeZoneShortName': 'IST',
'uuid': 'cc4c841e-ad42-3fdb-8d9f-b26fa8acd1cf',
'messageBoardId': 'finmb_6411769',
'gmtOffSetMilliseconds': 19800000,
'currentPrice': 3329.25,
```

```
'targetHighPrice': 4450.0,
'targetLowPrice': 2638.0,
'targetMeanPrice': 3510.22,
'targetMedianPrice': 3541.0,
'recommendationMean': 2.6,
'recommendationKey': 'hold',
'numberOfAnalystOpinions': 41,
'totalCash': 455260012544,
'totalCashPerShare': 124.42,
'ebitda': 576249987072,
'totalDebt': 76880003072,
'quickRatio': 2.362,
'currentRatio': 2.532,
'totalRevenue': 2254579957760,
'debtToEquity': 8.429,
'revenuePerShare': 616.165,
'returnOnAssets': 0.23774,
'returnOnEquity': 0.4673,
'freeCashflow': 356872486912,
'operatingCashflow': 419650011136,
'earningsGrowth': 0.16,
'revenueGrowth': 0.169,
'grossMargins': 0.42604,
'ebitdaMargins': 0.25559,
'operatingMargins': 0.24056,
'financialCurrency': 'INR',
'trailingPegRatio': 2.105}
```


To obtain balance sheet

In [15]:
```
price_tcs.balance_sheet
```

Out[15]:

	2023-03-31	2022-03-31	2021-03-31	2020-03-31
Treasury Shares Number	NaN	0.0	NaN	NaN
Ordinary Shares Number	3659051373.0	3659051373.0	3699051373.0	3752384706.0
Share Issued	3659051373.0	3659051373.0	3699051373.0	3752384706.0
Total Debt	76880000000.0	78180000000.0	77950000000.0	81760000000.0
Tangible Book Value	876990000000.0	861180000000.0	841300000000.0	821330000000.0
...	...	...	...	...
Cash Cash Equivalents And Short Term Investments	492480000000.0	484330000000.0	383830000000.0	357250000000.0
Other Short Term Investments	421250000000.0	359450000000.0	315380000000.0	270790000000.0
Cash And Cash Equivalents	71230000000.0	124880000000.0	68450000000.0	86460000000.0
Cash Equivalents	NaN	NaN	NaN	8050000000.0
Cash Financial	NaN	124880000000.0	68450000000.0	86460000000.0

76 rows × 4 columns

To fetch data for various keys .....

In [16]:
```
pb = price_tcs.info['priceToBook']

print('Price to Book ratio is: %.2f' % pb)
```

```
Price to Book ratio is: 13.47
```

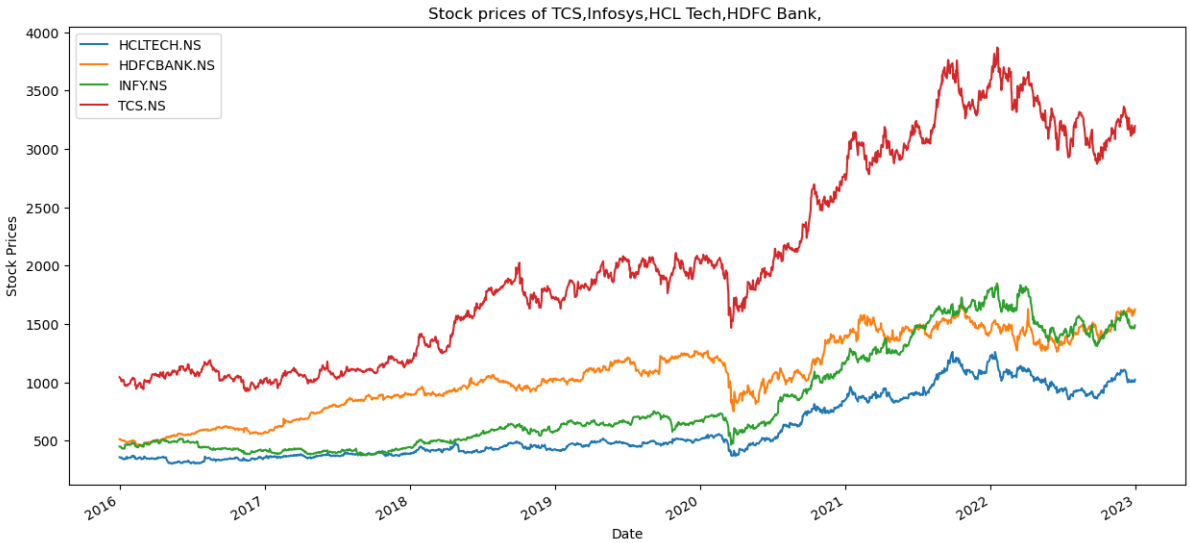
In [17]:
```
pe = price_tcs.info['totalRevenue']
print('Price to Earnings ratio is: %.2f' % pe)
```

```
Price to Earnings ratio is: 2254579957760.00
```

Displaying only open price of all the stocks

In [18]:
```
price_list['Open'].plot(figsize=(15,7))
plt.ylabel('Stock Prices')
plt.title('Stock prices of TCS,Infosys,HCL Tech,HDFC Bank, ')
```

Out[18]:
```
Text(0.5, 1.0, 'Stock prices of TCS,Infosys,HCL Tech,HDFC Bank, ')
```



Volumes of each stocks

In [18]:
```
price_list['Volume'].plot(figsize=(15,7))
plt.ylabel('Volume Traded')
plt.legend()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 price_list['Volume'].plot(figsize=(7,5))
      2 plt.ylabel('Volume Traded')
      3 plt.legend()

NameError: name 'price_list' is not defined
```

In [22]:
```
price_HDFCB = yf.download('HDFCBANK.NS',start,end,auto_adjust=True)
```

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

Locating the spike in the volume of HDFC Bank

In [23]:
```
price_HDFCB.iloc[[price_HDFC['Volume'].argmax()]]
```

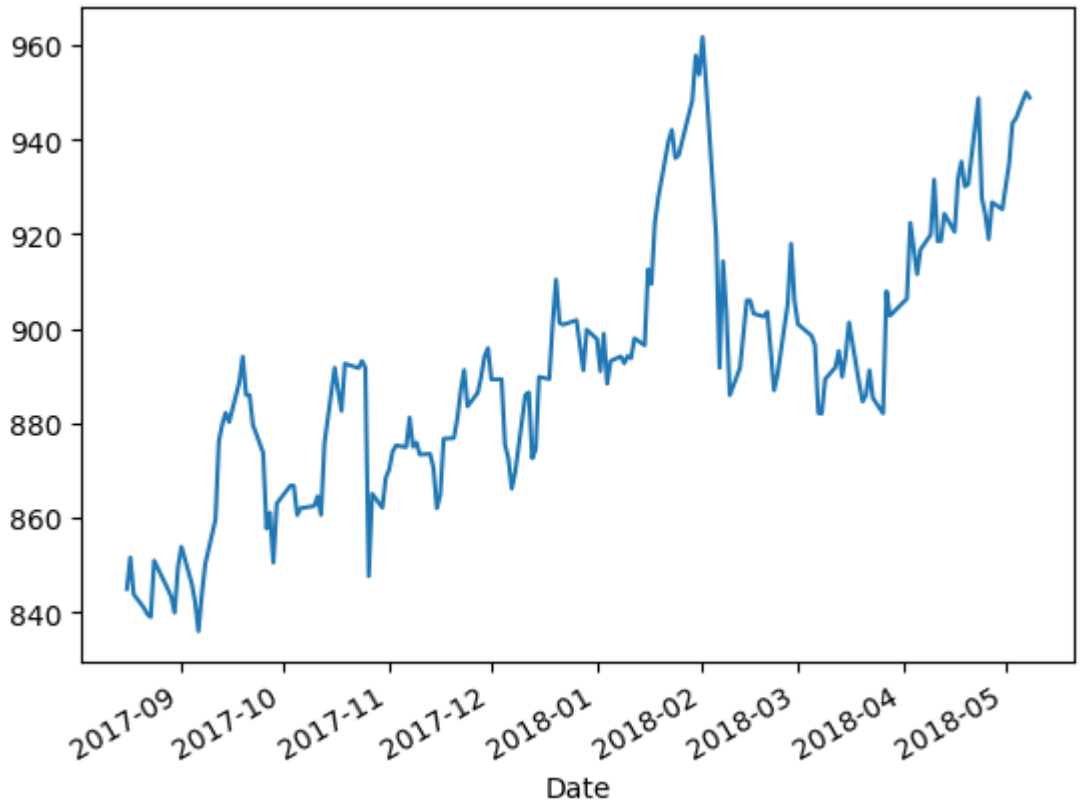
Out[23]:

Date	Open	High	Low	Close	Volume
2017-02-17	688.164131	692.450275	647.898172	655.803772	201129980

Plotting the strike price range with volume, for indepth insights

In [24]:
```
price_HDFCB.iloc[400:580]['Open'].plot()
```

Out[24]:
```
<Axes: xlabel='Date'>
```



In [25]:
```
price_list['Volume']
```

Out[25]:

Date	HCLTECH.NS	HDFCBANK.NS	INFY.NS	TCS.NS
2016-01-01	970392	1597538	1806550	712262
2016-01-04	2172072	2593768	3975362	1870184
2016-01-05	1465382	1580436	4949786	2678020
2016-01-06	2797870	2082768	5588328	2653228
2016-01-07	2901314	3027714	5294088	3199580
...	...	...	...	...
2022-12-26	1680715	4953661	4115459	870157
2022-12-27	554319	3963386	4860076	835883
2022-12-28	1397806	4345935	5029860	910795
2022-12-29	1277244	5506448	4624745	1037927
2022-12-30	1860560	3561320	5060544	1163131

1730 rows × 4 columns

Total Traded Capital

In [29]:
```
price_tcs = yf.download('TCS.NS',start,end,auto_adjust=True)
price_tcs.tail()
```

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

Out[29]:

Date	Open	High	Low	Close	Volume
2022-12-26	3140.893574	3183.360993	3137.682924	3164.778320	870157
2022-12-27	3180.636885	3185.112366	3143.958231	3171.199707	835883
2022-12-28	3161.762457	3177.620935	3138.607155	3168.864746	910795
2022-12-29	3143.569256	3182.874721	3140.553141	3180.199219	1037927
2022-12-30	3197.030340	3209.629474	3158.259789	3168.475342	1163131

In [31]:
```
price_HDFCBank = yf.download('HDFCBANK.NS',start,end,auto_adjust=True)
price_HDFCBank.tail()
```

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

Out[31]:

Date	Open	High	Low	Close	Volume
2022-12-26	1581.365081	1620.417236	1571.972791	1610.975464	4953661
2022-12-27	1614.485294	1617.401799	1595.206382	1612.606812	3963386
2022-12-28	1604.697425	1614.485204	1604.697425	1611.321533	4345935
2022-12-29	1601.731453	1624.866170	1592.734652	1622.691162	5506448
2022-12-30	1626.349241	1626.349241	1601.632687	1609.690308	3561320

In [32]:
```
price_infy = yf.download('INFY.NS',start,end,auto_adjust=True)
price_infy.tail()
```

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

Out[32]:

Date	Open	High	Low	Close	Volume
2022-12-26	1460.944984	1470.097213	1456.904360	1462.794922	4115459
2022-12-27	1470.145772	1481.342618	1458.024018	1474.916626	4860076
2022-12-28	1465.423653	1484.750358	1458.997661	1470.340576	5029860
2022-12-29	1464.352680	1481.196678	1461.431763	1477.545532	4624745
2022-12-30	1487.719989	1490.056746	1464.401389	1468.441895	5060544

In [33]:
```
price_hcltech = yf.download('HCLTECH.NS',start,end,auto_adjust=True)
price_hcltech.tail()
```

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

Out[33]:

Date	Open	High	Low	Close	Volume
2022-12-26	1004.586292	1015.060882	1002.637531	1005.268311	1680715
2022-12-27	1012.381388	1014.086554	1003.952972	1009.799255	554319
2022-12-28	1003.611915	1011.699321	1000.201583	1008.922241	1397806
2022-12-29	1003.611960	1019.202049	1001.760613	1017.691711	1277244
2022-12-30	1023.099530	1028.215028	1007.801804	1012.673706	1860560

To find the total capital traded in the market

In [34]:
```
price_hcltech['Total Capital Traded'] = price_hcltech['Open'] * price_hcltech['Volu
price_tcs['Total Capital Traded'] = price_tcs['Open'] * price_tcs['Volume']
price_infy['Total Capital Traded'] = price_infy['Open'] * price_infy['Volume']
price_HDFCBank['Total Capital Traded'] = price_HDFCBank['Open'] * price_HDFCBank['V
```

In [36]:
```
price_tcs.head()
```

Out[36]:

Date	Open	High	Low	Close	Volume	Total Capital Traded
2016-01-01	1043.815073	1043.815073	1032.366917	1034.142944	712262	7.434698e+08
2016-01-04	1031.468298	1033.116019	1012.594949	1014.114258	1870184	1.929036e+09
2016-01-05	1020.704773	1021.389564	1001.874161	1005.276489	2678020	2.733468e+09
2016-01-06	1005.811604	1021.197051	1005.811604	1019.249878	2653228	2.668648e+09
2016-01-07	1014.285413	1019.806165	1010.005728	1014.820374	3199580	3.245287e+09

In [37]:
```
price_tcs.tail()
```

Out[37]:

Date	Open	High	Low	Close	Volume	Total Capital Traded
2022-12-26	3140.893574	3183.360993	3137.682924	3164.778320	870157	2.733071e+09
2022-12-27	3180.636885	3185.112366	3143.958231	3171.199707	835883	2.658640e+09
2022-12-28	3161.762457	3177.620935	3138.607155	3168.864746	910795	2.879717e+09
2022-12-29	3143.569256	3182.874721	3140.553141	3180.199219	1037927	3.262795e+09
2022-12-30	3197.030340	3209.629474	3158.259789	3168.475342	1163131	3.718565e+09

In [55]:
```
price_hcltech.head()
```

Out[55]:

Date	Open	High	Low	Close	Volume	Total Capital Traded
2016-01-01	1043.815073	1043.815073	1032.366917	1034.142944	712262	7.434698e+08
2016-01-04	1031.467925	1033.115645	1012.594583	1014.113892	1870184	1.929035e+09
2016-01-05	1020.704835	1021.389626	1001.874222	1005.276550	2678020	2.733468e+09
2016-01-06	1005.811664	1021.197112	1005.811664	1019.249939	2653228	2.668648e+09
2016-01-07	1014.285352	1019.806104	1010.005667	1014.820312	3199580	3.245287e+09

In [38]:
```
price_infy.head()
```

Out[38]:

Date	Open	High	Low	Close	Volume	Total Capital Traded
2016-01-01	448.324685	451.972398	445.553203	450.464417	1806550	8.099210e+08
2016-01-04	448.304343	449.323262	438.563508	439.725067	3975362	1.782172e+09
2016-01-05	442.415016	442.415016	432.939061	437.748383	4949786	2.189860e+09
2016-01-06	437.992802	437.992802	431.573608	435.832672	5588328	2.447647e+09
2016-01-07	432.042465	435.241864	427.090504	428.272461	5294088	2.287271e+09

In [39]:
```
price_HDFCBank.head()
```

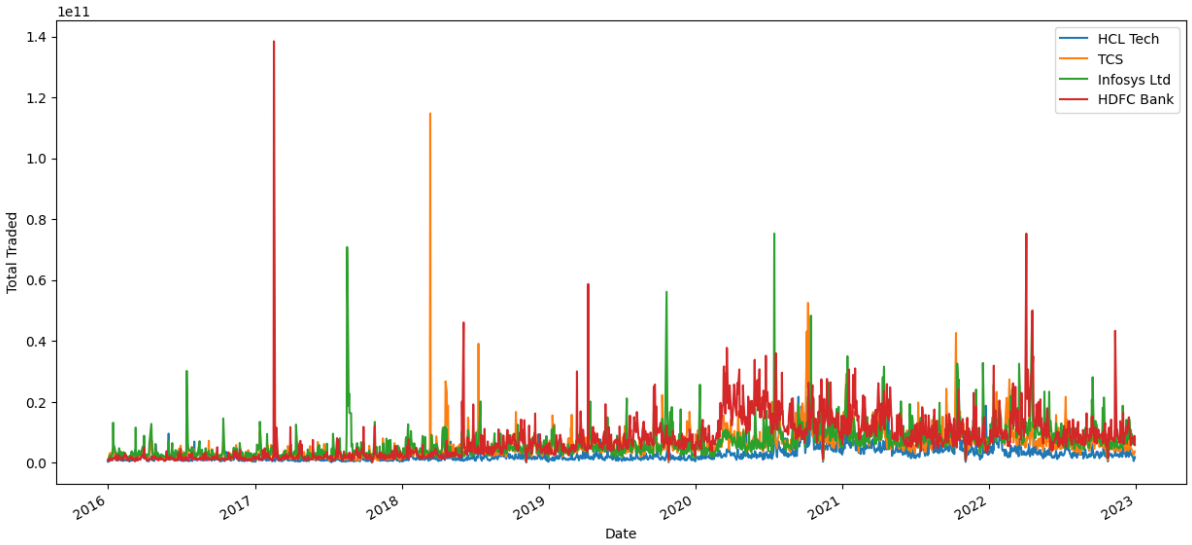
Out[39]:

Date	Open	High	Low	Close	Volume	Total Capital Traded
2016-01-01	511.283782	514.991806	508.331524	514.283264	1597538	8.167953e+08
2016-01-04	512.039535	512.039535	504.528980	505.662659	2593768	1.328112e+09
2016-01-05	505.520900	507.693807	501.340515	501.836517	1580436	7.989434e+08
2016-01-06	499.120429	508.614876	499.120429	504.056580	2082768	1.039552e+09
2016-01-07	500.750088	503.017444	495.837520	498.907867	3027714	1.516128e+09

Volume chart of total capital traded

In [40]:
```
price_hcltech['Total Capital Traded'].plot(label='HCL Tech',figsize=(15,7))
price_tcs['Total Capital Traded'].plot(label='TCS')
price_infy['Total Capital Traded'].plot(label='Infosys Ltd')
price_HDFCBank['Total Capital Traded'].plot(label='HDFC Bank')
plt.legend()
plt.ylabel('Total Traded')
```

Out[40]:
```
Text(0, 0.5, 'Total Traded')
```



Max spiked value

In [41]:
```
price_tcs['Total Capital Traded'].argmax()
```

Out[41]:
```
542
```

In [42]:
```
price_tcs.iloc[[price_tcs['Total Capital Traded'].argmax()]]
```

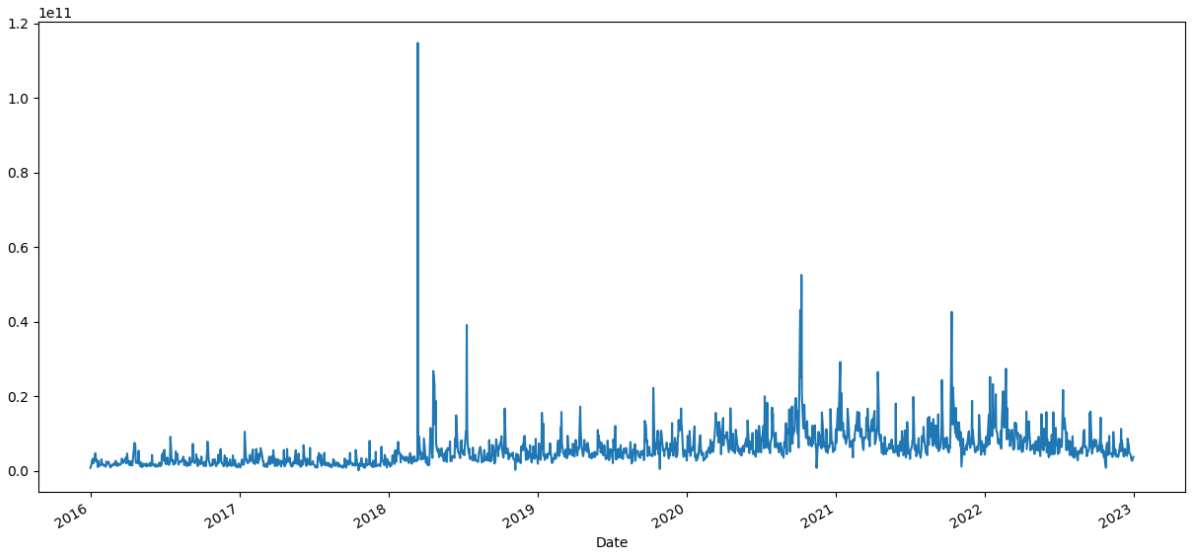
Out[42]:

Date	Open	High	Low	Close	Volume	Total Capital Traded
2018-03-13	1302.682288	1309.273885	1279.121904	1285.713501	88067154	1.147235e+11

Graph plotted for Total Captial Traded

In [43]:
```
price_tcs['Total Capital Traded'].plot(figsize=(15,7))
```

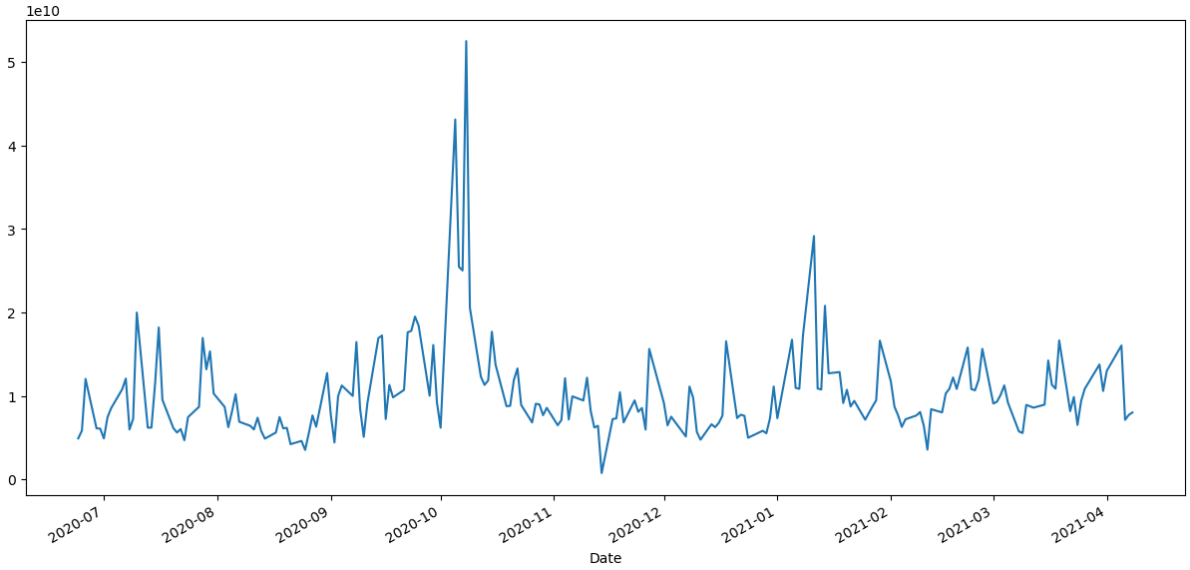
Out[43]:
```
<Axes: xlabel='Date'>
```



Graph plotted for Total Captial Traded in-depth

In [44]:
```
price_tcs.iloc[1100:1300]['Total Capital Traded'].plot(figsize=(15,7))
```

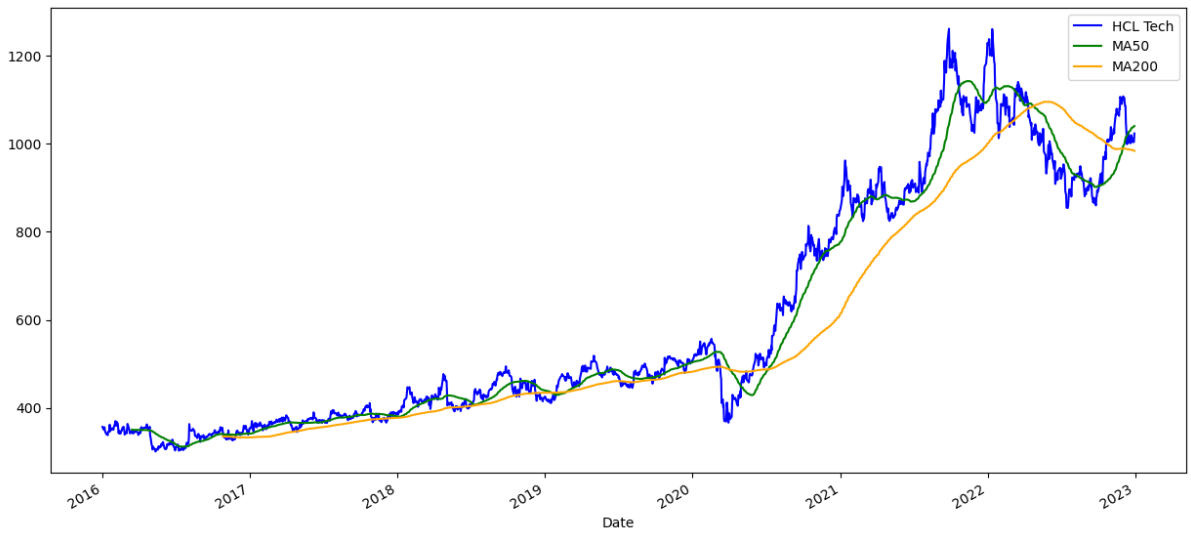
Out[44]:
```
<Axes: xlabel='Date'>
```



Plotting moving average of HCL Tech stock

In [46]:
```
price_hcltech['Open'].plot(label='HCL Tech',figsize=(15,7),color='Blue')
price_hcltech['MA50'] = price_hcltech['Open'].rolling(50).mean()
price_hcltech['MA50'].plot(label='MA50',color='Green')
price_hcltech['MA200'] = price_hcltech['Open'].rolling(200).mean()
price_hcltech['MA200'].plot(label='MA200',color='Orange')
plt.legend()
```

Out[46]:
```
<matplotlib.legend.Legend at 0x1e41fd8b190>
```



Plotting moving average of TCS stock

In [47]:
```
price_tcs['Open'].plot(label='TCS',figsize=(15,7),color='Cyan')
price_tcs['MA50'] = price_tcs['Open'].rolling(50).mean()
price_tcs['MA50'].plot(label='MA50',color='Green')
price_tcs['MA200'] = price_tcs['Open'].rolling(200).mean()
price_tcs['MA200'].plot(label='MA200',color='Orange')
plt.legend()
```

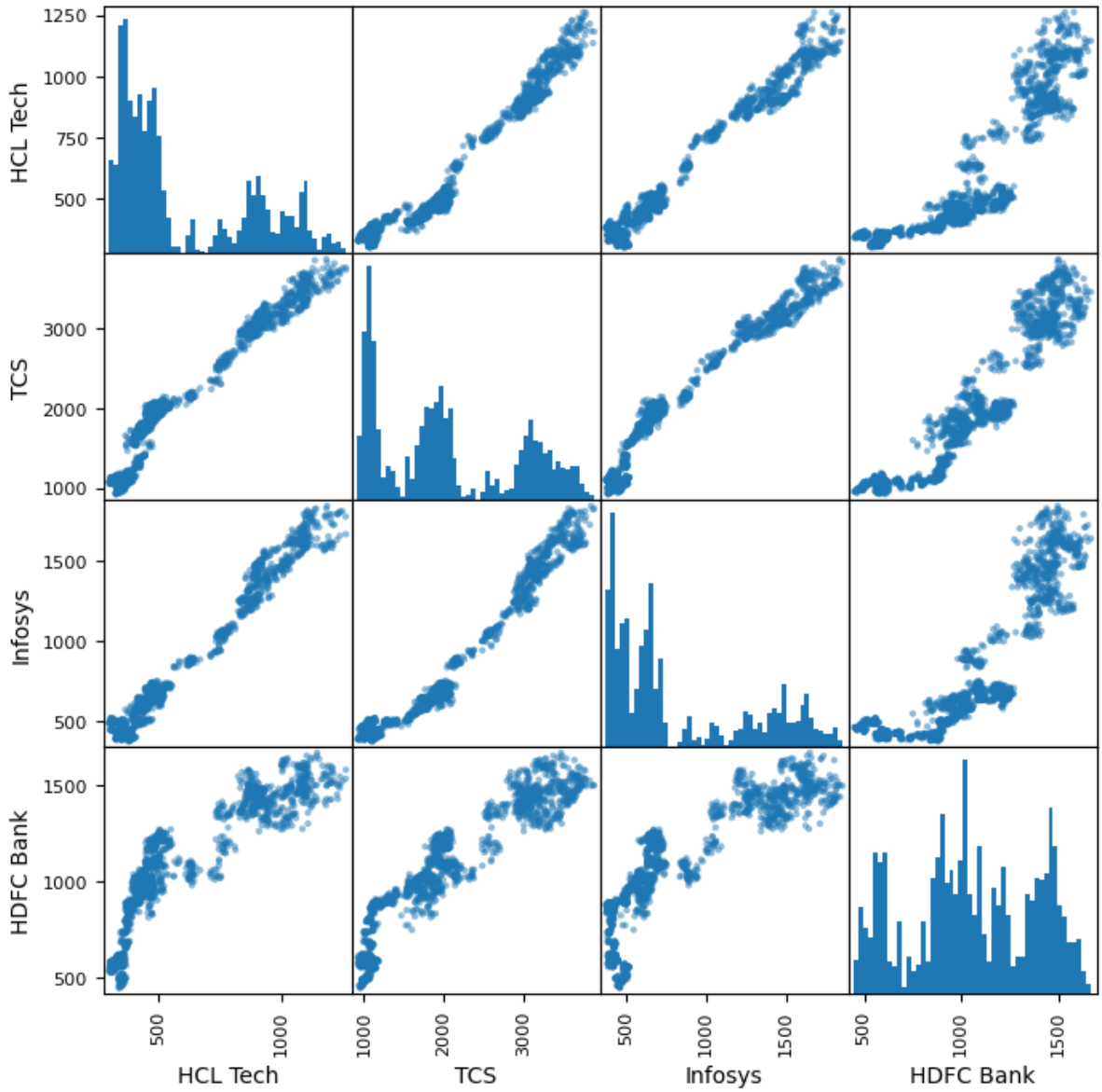
Out[47]:
```
<matplotlib.legend.Legend at 0x1e420f30c10>
```

Correlation & Scatter Matrix between stocks

In [49]:
```
stock_list = pd.concat([price_hcltech['Open'],price_tcs['Open'],price_infy['Open'],
stock_list.columns = ['HCL Tech','TCS','Infosys','HDFC Bank']
```

In [50]:
```
scatter_matrix(stock_list,figsize=(8,8),hist_kwds={'bins': 50})
```

Out[50]:
```
array([[<Axes: xlabel='HCL Tech', ylabel='HCL Tech'>,
        <Axes: xlabel='TCS', ylabel='HCL Tech'>,
        <Axes: xlabel='Infosys', ylabel='HCL Tech'>,
        <Axes: xlabel='HDFC Bank', ylabel='HCL Tech'>],
       [<Axes: xlabel='HCL Tech', ylabel='TCS'>,
        <Axes: xlabel='TCS', ylabel='TCS'>,
        <Axes: xlabel='Infosys', ylabel='TCS'>,
        <Axes: xlabel='HDFC Bank', ylabel='TCS'>],
       [<Axes: xlabel='HCL Tech', ylabel='Infosys'>,
        <Axes: xlabel='TCS', ylabel='Infosys'>,
        <Axes: xlabel='Infosys', ylabel='Infosys'>,
        <Axes: xlabel='HDFC Bank', ylabel='Infosys'>],
       [<Axes: xlabel='HCL Tech', ylabel='HDFC Bank'>,
        <Axes: xlabel='TCS', ylabel='HDFC Bank'>,
        <Axes: xlabel='Infosys', ylabel='HDFC Bank'>,
        <Axes: xlabel='HDFC Bank', ylabel='HDFC Bank'>]], dtype=object)
```



Ploting candle stick patterns, install necessary candle sticks liberaries

Use the below two liberaries

from mplfinance.original_flavor import candlestick_ohlc

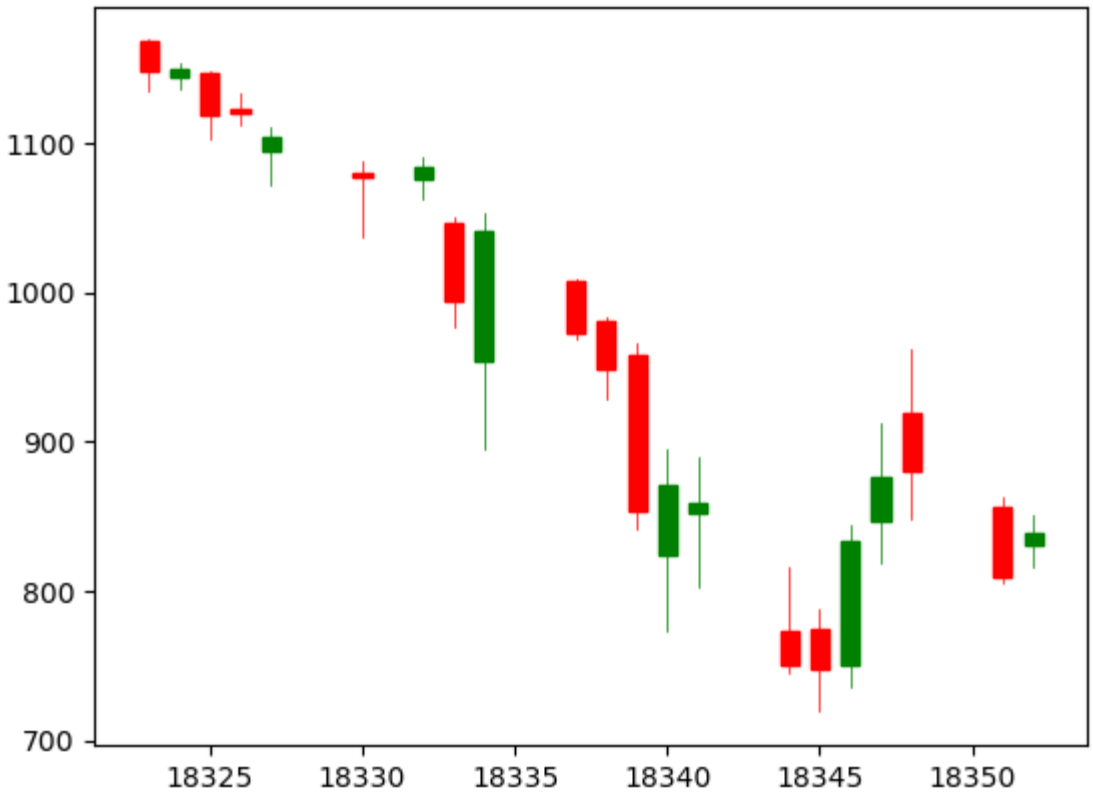
from matplotlib.dates import DateFormatter, date2num, WeekdayLocator, DayLocator, DAYNAME

In [51]:
```
HDFCBank_reset = price_HDFCBank.loc['2020-03':'2020-03'].reset_index()
HDFCBank_reset['date_ax'] = HDFCBank_reset['Date'].apply(lambda date: date2num(date
HDFCBank_values = [tuple(vals) for vals in HDFCBank_reset[['date_ax','Open','High',

mondays = WeekdayLocator(MONDAY)
alldays = DayLocator()
weekFormatter = DateFormatter('%b %d')
dayFormatter = DateFormatter('%d')

fig, ax = plt.subplots()
candlestick_ohlc(ax,HDFCBank_values,width=0.6,colorup='g',colordown='r')
```

Out[51]:
```
([<matplotlib.lines.Line2D at 0x1e423c897b0>,
  <matplotlib.lines.Line2D at 0x1e423ce0280>,
  <matplotlib.lines.Line2D at 0x1e423ce0700>,
  <matplotlib.lines.Line2D at 0x1e423ce0b80>,
  <matplotlib.lines.Line2D at 0x1e423ce1000>,
  <matplotlib.lines.Line2D at 0x1e423ce1510>,
  <matplotlib.lines.Line2D at 0x1e423ce1930>,
  <matplotlib.lines.Line2D at 0x1e423ce1db0>,
  <matplotlib.lines.Line2D at 0x1e423ce2230>,
  <matplotlib.lines.Line2D at 0x1e423ce26b0>,
  <matplotlib.lines.Line2D at 0x1e423ce2b30>,
  <matplotlib.lines.Line2D at 0x1e423ce2fb0>,
  <matplotlib.lines.Line2D at 0x1e423ce3430>,
  <matplotlib.lines.Line2D at 0x1e423ce38b0>,
  <matplotlib.lines.Line2D at 0x1e423ce3d30>,
  <matplotlib.lines.Line2D at 0x1e423d201f0>,
  <matplotlib.lines.Line2D at 0x1e423d20670>,
  <matplotlib.lines.Line2D at 0x1e423d20af0>,
  <matplotlib.lines.Line2D at 0x1e423d20f70>,
  <matplotlib.lines.Line2D at 0x1e423d213f0>,
  <matplotlib.lines.Line2D at 0x1e423d21870>],
 [<matplotlib.patches.Rectangle at 0x1e423c178b0>,
  <matplotlib.patches.Rectangle at 0x1e423ce02b0>,
  <matplotlib.patches.Rectangle at 0x1e423ce0730>,
  <matplotlib.patches.Rectangle at 0x1e423ce0bb0>,
  <matplotlib.patches.Rectangle at 0x1e423ce1030>,
  <matplotlib.patches.Rectangle at 0x1e423ce14b0>,
  <matplotlib.patches.Rectangle at 0x1e423ce1960>,
  <matplotlib.patches.Rectangle at 0x1e423ce1de0>,
  <matplotlib.patches.Rectangle at 0x1e423ce2260>,
  <matplotlib.patches.Rectangle at 0x1e423ce26e0>,
  <matplotlib.patches.Rectangle at 0x1e423ce2b60>,
  <matplotlib.patches.Rectangle at 0x1e423ce2fe0>,
  <matplotlib.patches.Rectangle at 0x1e423ce3460>,
  <matplotlib.patches.Rectangle at 0x1e423ce38e0>,
  <matplotlib.patches.Rectangle at 0x1e423ce3d60>,
  <matplotlib.patches.Rectangle at 0x1e423d20220>,
  <matplotlib.patches.Rectangle at 0x1e423d206a0>,
  <matplotlib.patches.Rectangle at 0x1e423d20b20>,
  <matplotlib.patches.Rectangle at 0x1e423d20fa0>,
  <matplotlib.patches.Rectangle at 0x1e423d21420>,
  <matplotlib.patches.Rectangle at 0x1e423d218a0>])
```



Daily Percentage Change or Volatile Market

rt = [pt/pt-1]-1

In [52]:
```
price_hcltech['Returns'] = (price_hcltech['Close']/price_hcltech['Close'].shift(1))
price_tcs['Returns'] = (price_tcs['Close']/price_tcs['Close'].shift(1)) - 1
price_infy['Returns'] = (price_infy['Close']/price_infy['Close'].shift(1)) - 1
price_HDFCBank['Returns'] = (price_HDFCBank['Close']/price_HDFCBank['Close'].shift(
```

In [53]:
```
price_hcltech.tail()
```

Out[53]:

Date	Open	High	Low	Close	Volume	Total Capital Traded	MA
2022-12-26	1004.586292	1015.060882	1002.637531	1005.268311	1680715	1.688423e+09	1036.7762
2022-12-27	1012.381388	1014.086554	1003.952972	1009.799255	554319	5.611822e+08	1037.6329
2022-12-28	1003.611915	1011.699321	1000.201583	1008.922241	1397806	1.402855e+09	1038.3006
2022-12-29	1003.611960	1019.202049	1001.760613	1017.691711	1277244	1.281857e+09	1039.0783
2022-12-30	1023.099530	1028.215028	1007.801804	1012.673706	1860560	1.903538e+09	1039.9260

In [54]:
```
price_tcs.tail()
```

Out[54]:

Date	Open	High	Low	Close	Volume	Total Capital Traded	MA
2022-12-26	3140.893574	3183.360993	3137.682924	3164.778320	870157	2.733071e+09	3184.8269
2022-12-27	3180.636885	3185.112366	3143.958231	3171.199707	835883	2.658640e+09	3187.4015
2022-12-28	3161.762457	3177.620935	3138.607155	3168.864746	910795	2.879717e+09	3190.8486
2022-12-29	3143.569256	3182.874721	3140.553141	3180.199219	1037927	3.262795e+09	3192.4266
2022-12-30	3197.030340	3209.629474	3158.259789	3168.475342	1163131	3.718565e+09	3194.8988

In [55]:
```
price_infy.tail()
```

Out[55]:

Date	Open	High	Low	Close	Volume	Total Capital Traded	Returns
2022-12-26	1460.944984	1470.097213	1456.904360	1462.794922	4115459	6.012459e+09	0.003540
2022-12-27	1470.145772	1481.342618	1458.024018	1474.916626	4860076	7.145020e+09	0.008287
2022-12-28	1465.423653	1484.750358	1458.997661	1470.340576	5029860	7.370876e+09	-0.003103
2022-12-29	1464.352680	1481.196678	1461.431763	1477.545532	4624745	6.772258e+09	0.004900
2022-12-30	1487.719989	1490.056746	1464.401389	1468.441895	5060544	7.528672e+09	-0.006161

In [56]:
```
price_HDFCBank.tail()
```

Out[56]:

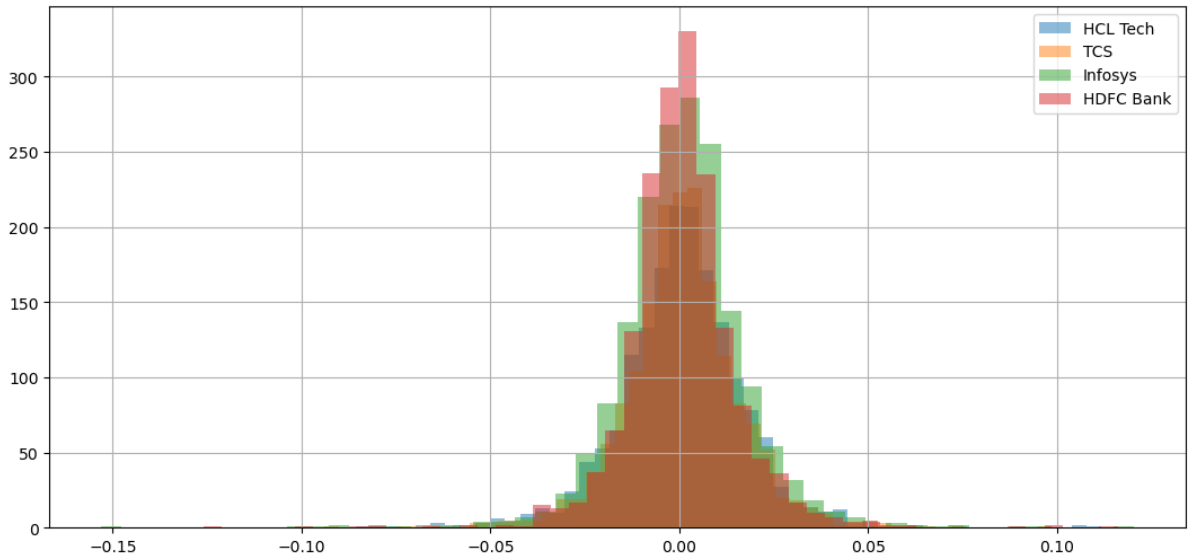
Date	Open	High	Low	Close	Volume	Total Capital Traded	Returns
2022-12-26	1581.365081	1620.417236	1571.972791	1610.975464	4953661	7.833547e+09	0.019904
2022-12-27	1614.485294	1617.401799	1595.206382	1612.606812	3963386	6.398828e+09	0.001013
2022-12-28	1604.697425	1614.485204	1604.697425	1611.321533	4345935	6.973911e+09	-0.000797
2022-12-29	1601.731453	1624.866170	1592.734652	1622.691162	5506448	8.819851e+09	0.007056
2022-12-30	1626.349241	1626.349241	1601.632687	1609.690308	3561320	5.791950e+09	-0.008012

Plotting the histogram with volatile values

Higher the values higher the volatile and not good for investment for now

In [57]:
```
price_hcltech['Returns'].hist(bins=50,label='HCL Tech',alpha=0.5,figsize=(13,6))
price_tcs['Returns'].hist(bins=50,label='TCS',alpha=0.5)
price_infy['Returns'].hist(bins=50,label='Infosys',alpha=0.5)
price_HDFCBank['Returns'].hist(bins=50,label='HDFC Bank',alpha=0.5)
plt.legend()
```

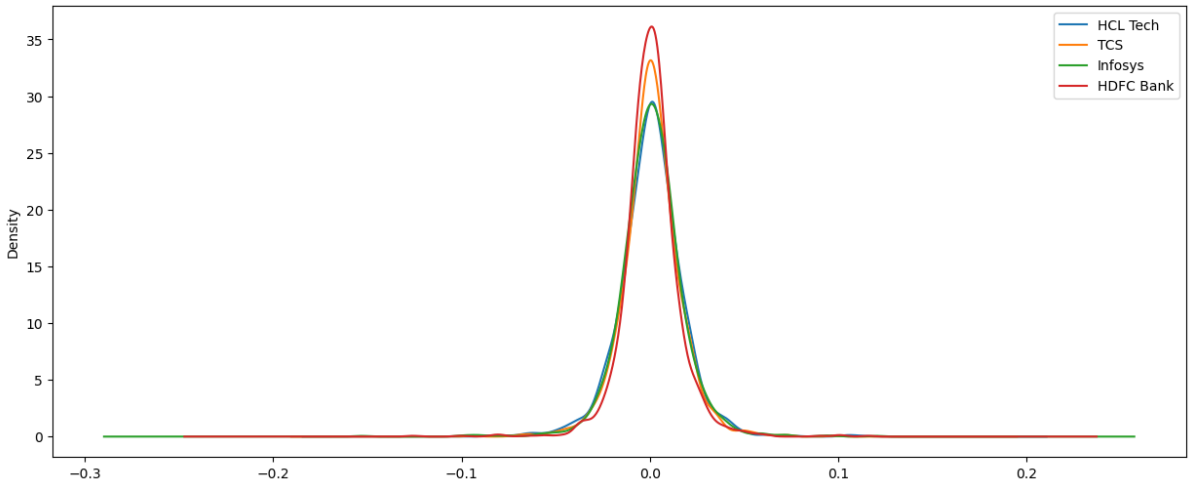
Out[57]:
```
<matplotlib.legend.Legend at 0x1e420e332b0>
```



Plotting KDE[Kernal Density Estimate] of the stocks for validating the volatile changes

In [58]:
```
price_hcltech['Returns'].plot(kind='kde',label='HCL Tech',figsize=(15,6))
price_tcs['Returns'].plot(kind='kde',label='TCS')
price_infy['Returns'].plot(kind='kde',label='Infosys')
price_HDFCBank['Returns'].plot(kind='kde',label='HDFC Bank')
plt.legend()
```

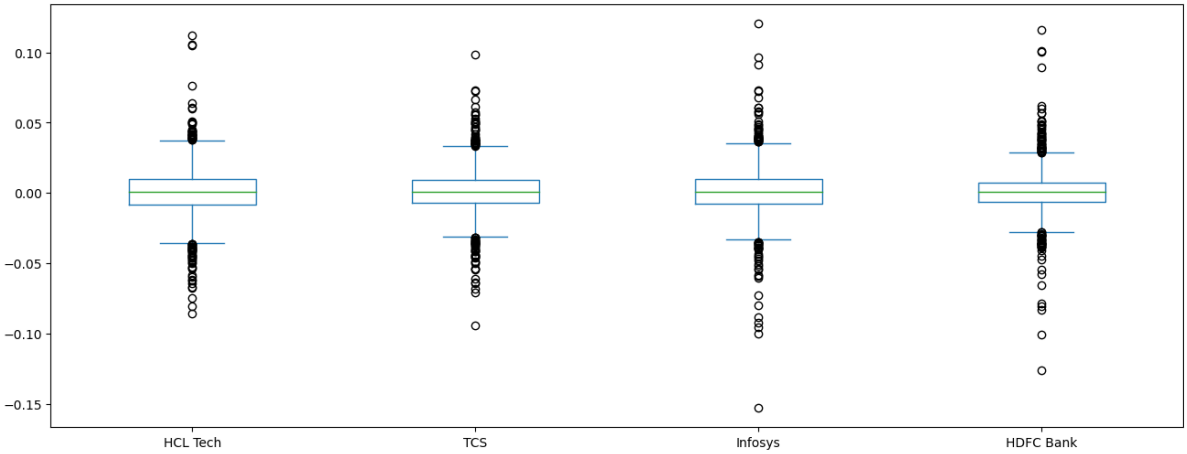
Out[58]:
```
<matplotlib.legend.Legend at 0x1e4251fc4c0>
```



Box plot for validating the volatile changes of stocks

In [59]:
```
box_df = pd.concat([price_hcltech['Returns'],price_tcs['Returns'],price_infy['Retur
box_df.columns = ['HCL Tech','TCS','Infosys','HDFC Bank']
box_df.plot(kind='box',figsize=(16,6))
```

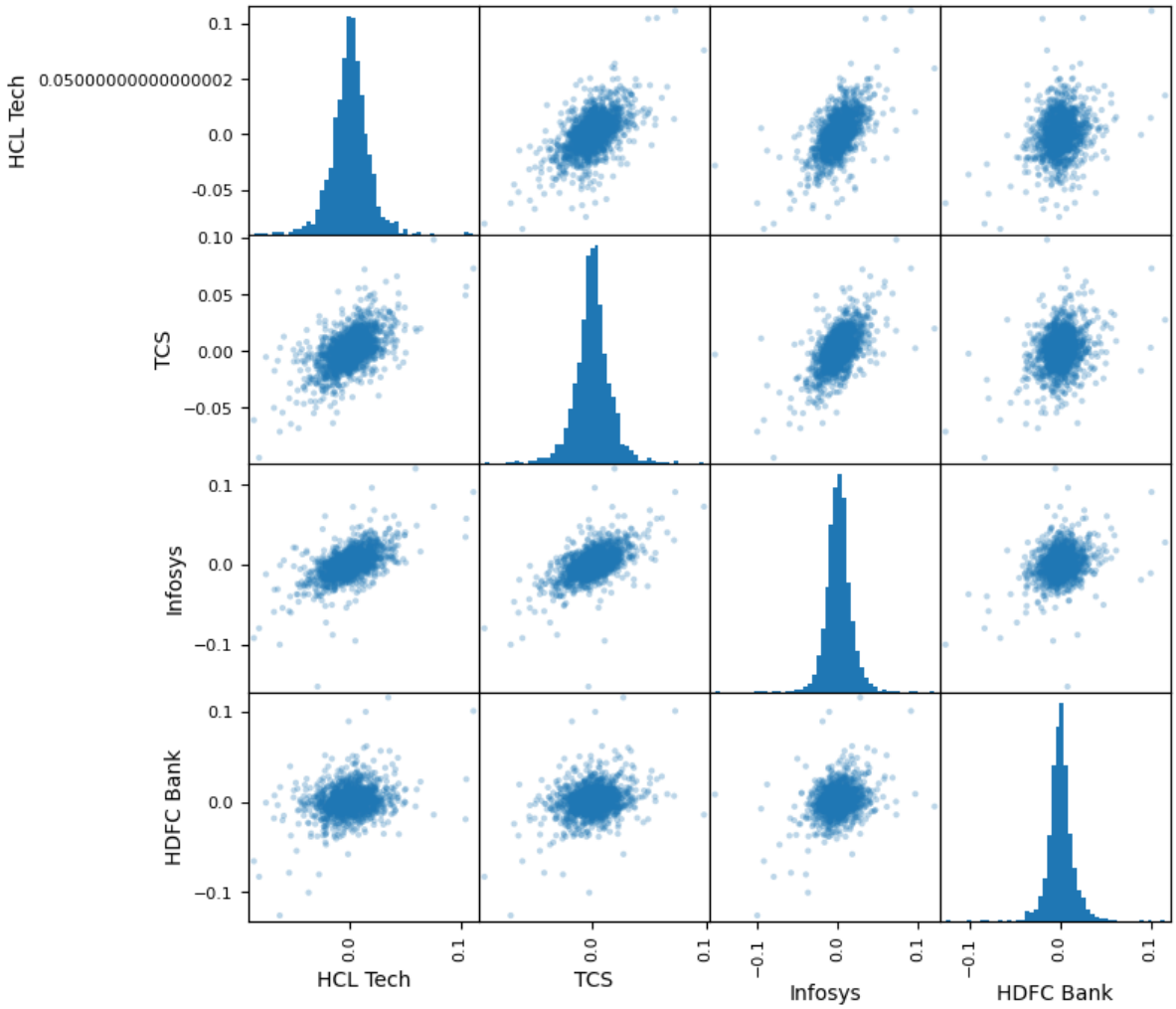
Out[59]:
```
<Axes: >
```



Compare the volatility of other stocks, create a scatter plot matrix

In [60]:
```
scatter_matrix(box_df,figsize=(8,8),hist_kwds={'bins':50},alpha=0.3)
```

Out[60]:
```
array([[<Axes: xlabel='HCL Tech', ylabel='HCL Tech'>,
        <Axes: xlabel='TCS', ylabel='HCL Tech'>,
        <Axes: xlabel='Infosys', ylabel='HCL Tech'>,
        <Axes: xlabel='HDFC Bank', ylabel='HCL Tech'>],
       [<Axes: xlabel='HCL Tech', ylabel='TCS'>,
        <Axes: xlabel='TCS', ylabel='TCS'>,
        <Axes: xlabel='Infosys', ylabel='TCS'>,
        <Axes: xlabel='HDFC Bank', ylabel='TCS'>],
       [<Axes: xlabel='HCL Tech', ylabel='Infosys'>,
        <Axes: xlabel='TCS', ylabel='Infosys'>,
        <Axes: xlabel='Infosys', ylabel='Infosys'>,
        <Axes: xlabel='HDFC Bank', ylabel='Infosys'>],
       [<Axes: xlabel='HCL Tech', ylabel='HDFC Bank'>,
        <Axes: xlabel='TCS', ylabel='HDFC Bank'>,
        <Axes: xlabel='Infosys', ylabel='HDFC Bank'>,
        <Axes: xlabel='HDFC Bank', ylabel='HDFC Bank'>]], dtype=object)
```



Cumulative Returns

it=(1+rt)it-1 = (1+ [pt/pt-1]-1)it-1 = [pt/pt-1]it-1

In [61]:
```
price_hcltech['Cumulative Returns'] = (1+ price_hcltech['Returns']).cumprod()
price_tcs['Cumulative Returns'] = (1+ price_tcs['Returns']).cumprod()
price_infy['Cumulative Returns'] = (1+ price_infy['Returns']).cumprod()
price_HDFCBank['Cumulative Returns'] = (1+ price_HDFCBank['Returns']).cumprod()
```

In [62]:
```
price_hcltech.head()
```

Out[62]:

Date	Open	High	Low	Close	Volume	Total Capital Traded	MA50	MA20
2016-01-01	357.171176	357.171176	352.325922	353.307495	970392	3.465961e+08	NaN	Nal
2016-01-04	351.281708	357.108561	349.798895	353.349304	2172072	7.630092e+08	NaN	Nal
2016-01-05	354.602405	355.479555	349.360335	352.033569	1465382	5.196280e+08	NaN	Nal
2016-01-06	356.586376	356.586376	349.610862	351.448730	2797870	9.976823e+08	NaN	Nal
2016-01-07	349.235004	349.986876	342.092425	344.619476	2901314	1.013240e+09	NaN	Nal

In [63]:
```
price_tcs.head()
```

Out[63]:

Date	Open	High	Low	Close	Volume	Total Capital Traded	MA50	M
2016-01-01	1043.815073	1043.815073	1032.366917	1034.142944	712262	7.434698e+08	NaN	
2016-01-04	1031.468298	1033.116019	1012.594949	1014.114258	1870184	1.929036e+09	NaN	
2016-01-05	1020.704773	1021.389564	1001.874161	1005.276489	2678020	2.733468e+09	NaN	
2016-01-06	1005.811604	1021.197051	1005.811604	1019.249878	2653228	2.668648e+09	NaN	
2016-01-07	1014.285413	1019.806165	1010.005728	1014.820374	3199580	3.245287e+09	NaN	

In [64]:
```
price_infy.head()
```

Out[64]:

Date	Open	High	Low	Close	Volume	Total Capital Traded	Returns	Cu
2016-01-01	448.324685	451.972398	445.553203	450.464417	1806550	8.099210e+08	NaN	
2016-01-04	448.304343	449.323262	438.563508	439.725067	3975362	1.782172e+09	-0.023841	
2016-01-05	442.415016	442.415016	432.939061	437.748383	4949786	2.189860e+09	-0.004495	
2016-01-06	437.992802	437.992802	431.573608	435.832672	5588328	2.447647e+09	-0.004376	
2016-01-07	432.042465	435.241864	427.090504	428.272461	5294088	2.287271e+09	-0.017347	

In [65]:
```
price_HDFCBank.head()
```

Out[65]:

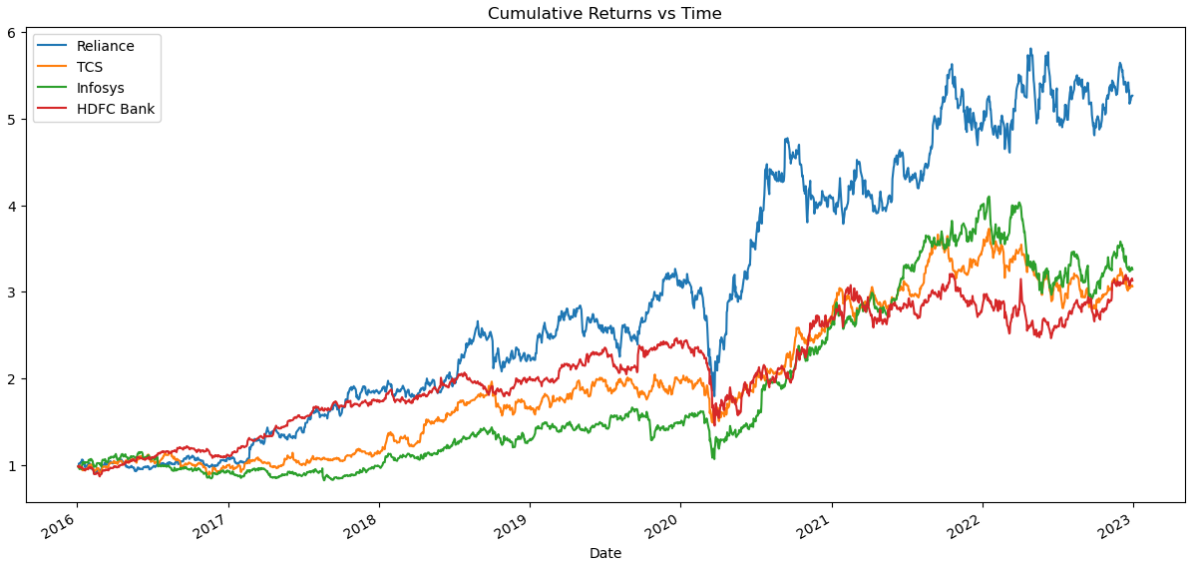
Date	Open	High	Low	Close	Volume	Total Capital Traded	Returns	Cu
2016-01-01	511.283782	514.991806	508.331524	514.283264	1597538	8.167953e+08	NaN	
2016-01-04	512.039535	512.039535	504.528980	505.662659	2593768	1.328112e+09	-0.016762	
2016-01-05	505.520900	507.693807	501.340515	501.836517	1580436	7.989434e+08	-0.007567	
2016-01-06	499.120429	508.614876	499.120429	504.056580	2082768	1.039552e+09	0.004424	
2016-01-07	500.750088	503.017444	495.837520	498.907867	3027714	1.516128e+09	-0.010215	

In [116…

```
price_rsl['Cumulative Returns'].plot(label='Reliance',figsize=(15,7))
price_tcs['Cumulative Returns'].plot(label='TCS')
price_infy['Cumulative Returns'].plot(label='Infosys')
price_HDFCBank['Cumulative Returns'].plot(label='HDFC Bank')
plt.title('Cumulative Returns vs Time')
plt.legend()
```

Out[116]:

```
<matplotlib.legend.Legend at 0x13ae0ceaec0>
```



In [ ]: