

```
In [83]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [84]: wipro = pd.read_csv('wipro.csv')
tcs = pd.read_csv('tcs.csv')
hcl = pd.read_csv('hcltech.csv')
infosys = pd.read_csv('infosys.csv')
```

```
In [85]: wipro['Date'] = pd.to_datetime(wipro['Date'], infer_datetime_format=True)
wipro['Date'] = pd.to_datetime(wipro['Date']).dt.date

tcs['Date'] = pd.to_datetime(tcs['Date'], infer_datetime_format=True)
tcs['Date'] = pd.to_datetime(tcs['Date']).dt.date

hcl['Date'] = pd.to_datetime(hcl['Date'], infer_datetime_format=True)
hcl['Date'] = pd.to_datetime(hcl['Date']).dt.date

infosys['Date'] = pd.to_datetime(infosys['Date'], infer_datetime_format=True)
infosys['Date'] = pd.to_datetime(infosys['Date']).dt.date
```

```
In [86]: wipro = wipro.set_index('Date')
tcs = tcs.set_index('Date')
hcl = hcl.set_index('Date')
infosys = infosys.set_index('Date')
```

```
In [66]: wipro.head()
```

```
Out[66]:
```

	Open	High	Low	Close	Volume
Date					
2006-01-02	103.95	104.51	102.44	103.88	645681
2006-01-03	103.73	107.10	103.14	106.43	1150446
2006-01-04	106.99	109.13	106.99	108.25	1151665
2006-01-05	108.45	108.90	106.26	106.66	839636
2006-01-06	106.88	106.88	101.48	106.36	994361

```
In [88]: tcs.head()
```

```
Out[88]:
```

	Open	High	Low	Close	Volume
Date					
2006-01-02	213.50	213.86	210.17	210.95	206688
2006-01-03	210.66	217.49	210.25	216.65	424617
2006-01-04	217.50	218.74	215.28	217.89	443202
2006-01-05	218.13	218.25	215.24	216.55	178978
2006-01-06	216.88	217.49	211.50	213.20	370718

```
In [89]: hcl.head()
```

Out[89]:

	Open	High	Low	Close	Volume
Date					
2006-01-02	67.50	68.10	66.75	67.36	291639
2006-01-03	68.00	69.00	67.36	68.71	412209
2006-01-04	71.25	71.63	68.52	69.04	646847
2006-01-05	69.76	71.25	68.79	70.35	503329
2006-01-06	71.00	74.06	69.50	73.30	1177316

In [90]: `infosys.head()`

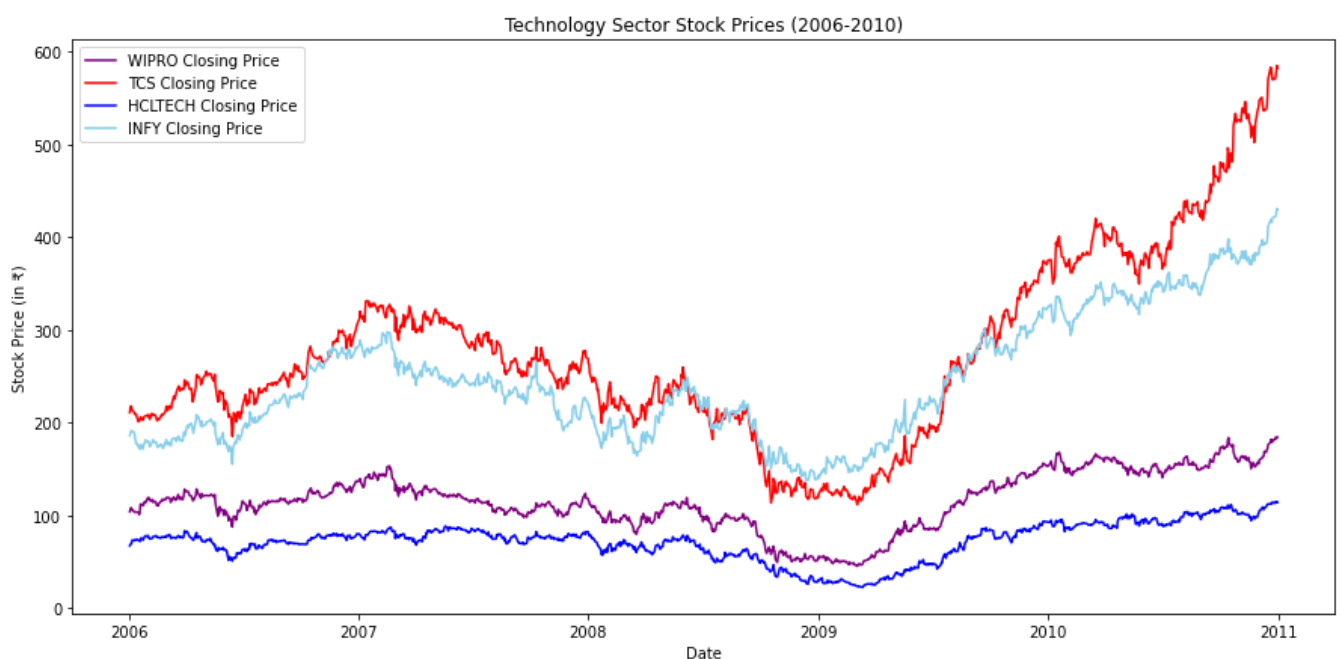
Out[90]:

	Open	High	Low	Close	Volume
Date					
2006-01-02	187.94	187.94	185.81	186.21	415500
2006-01-03	186.88	188.44	186.19	187.93	497496
2006-01-04	188.19	191.56	187.20	190.96	784558
2006-01-05	191.50	191.87	190.08	190.93	472285
2006-01-06	190.63	191.88	188.64	190.83	1062988

In [102]...

```
plt.figure(figsize=(15,7))
wipro['Close'].plot(label='WIPRO Closing Price', color = 'purple')
tcs['Close'].plot(label='TCS Closing Price', color = 'red')
hcl['Close'].plot(label='HCLTECH Closing Price', color = 'blue')
infosys['Close'].plot(label='INFY Closing Price', color = 'skyblue')
plt.title("Technology Sector Stock Prices (2006-2010)")
plt.xlabel("Date")
plt.ylabel("Stock Price (in ₹)")
plt.legend()
```

Out[102]: <matplotlib.legend.Legend at 0x17bfcf422b0>

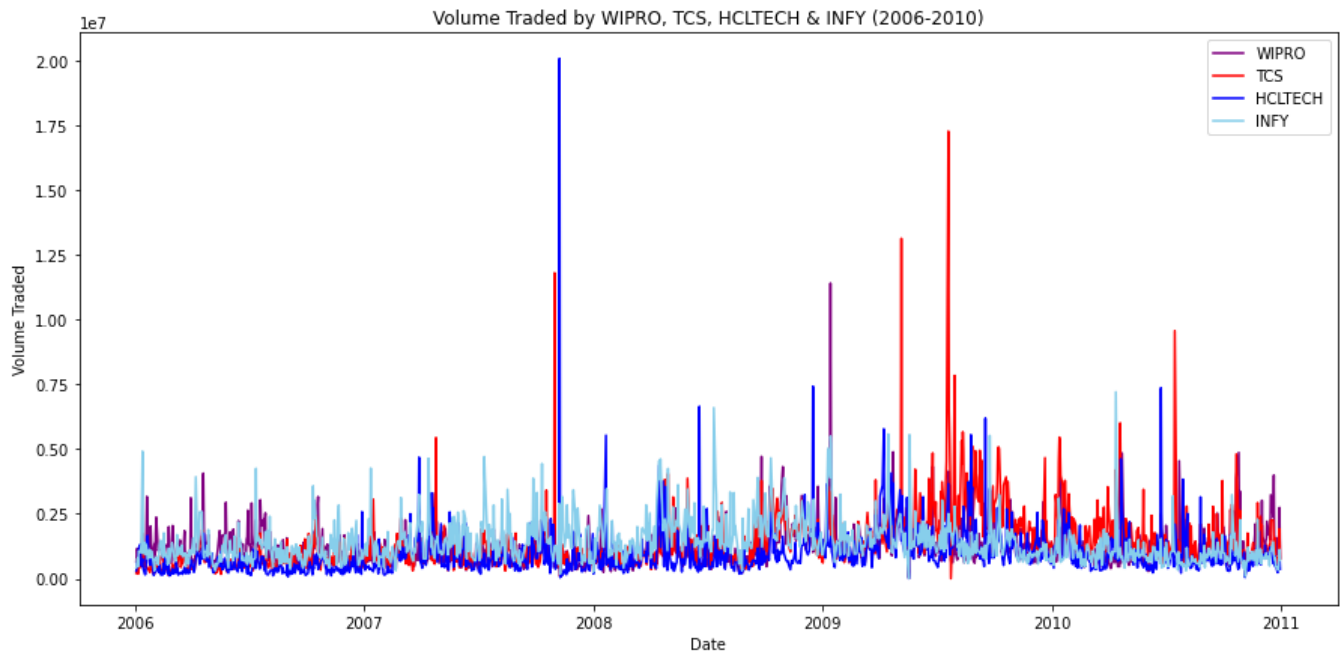


In [105]...

```
plt.figure(figsize=(15,7))
wipro['Volume'].plot(label='WIPRO', color = 'purple')
tcs['Volume'].plot(label='TCS', color = 'red')
hcl['Volume'].plot(label='HCLTECH', color = 'blue')
```

```
infosys['Volume'].plot(label='INFY', color='skyblue')
plt.title("Volume Traded by WIPRO, TCS, HCLTECH & INFY (2006-2010)")
plt.legend()
plt.xlabel('Date')
plt.ylabel('Volume Traded')
```

Out[105]: Text(0, 0.5, 'Volume Traded')



In [121... hcl.iloc[[hcl['Volume'].argmax()]]

Out[121]:

	Open	High	Low	Close	Volume	Total Traded
Date						
2007-11-08	74.0	80.5	73.0	75.16	20077384	32909206.0

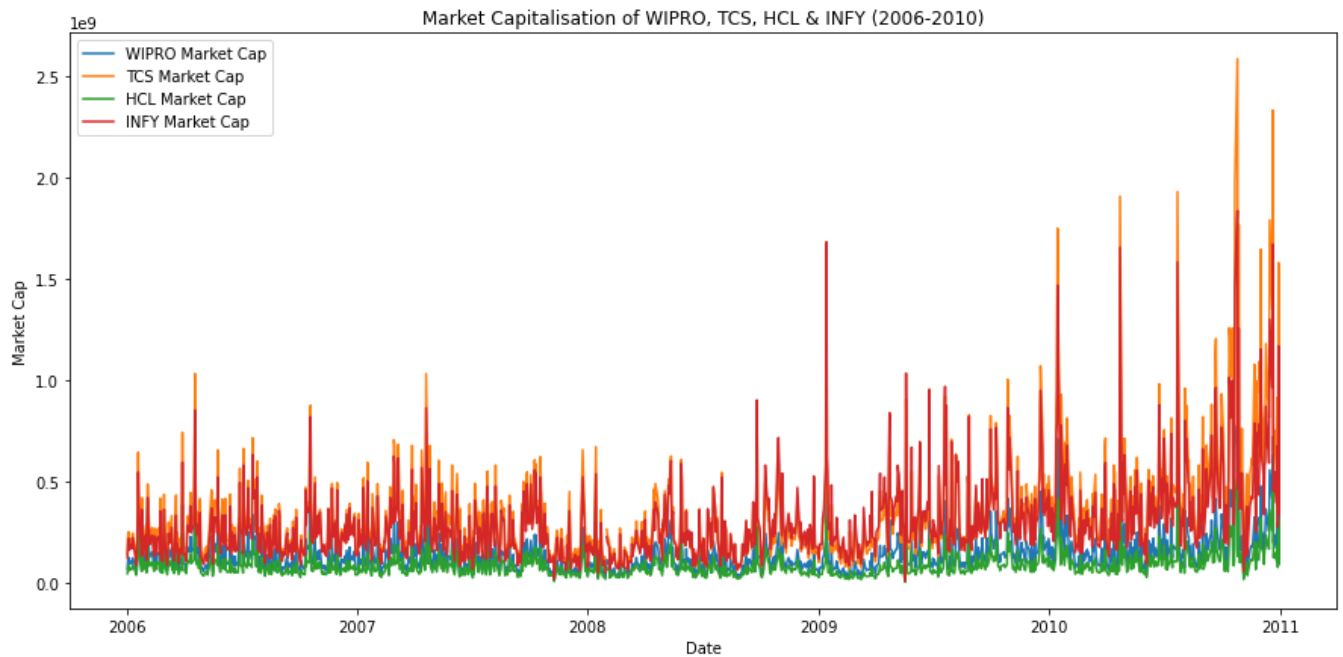
In [122... tcs.iloc[[tcs['Volume'].argmax()]]

Out[122]:

	Open	High	Low	Close	Volume	Total Traded
Date						
2009-07-20	222.5	257.45	222.5	250.28	17269768	921594777.5

In [117... plt.figure(figsize = (15,7))  
wipro['Total Traded'].plot(label = 'WIPRO Market Cap')  
tcs['Total Traded'].plot(label = 'TCS Market Cap')  
hcl['Total Traded'].plot(label = 'HCL Market Cap')  
infosys['Total Traded'].plot(label = 'INFY Market Cap')  
plt.legend()  
plt.xlabel('Date')  
plt.ylabel('Market Cap')  
plt.title('Market Capitalisation of WIPRO, TCS, HCL & INFY (2006-2010)')

Out[117]: Text(0.5, 1.0, 'Market Capitalisation of WIPRO, TCS, HCL & INFY (2006-2010)')



```
In [118... tcs.iloc[[tcs['Total Traded'].argmax()]]
```

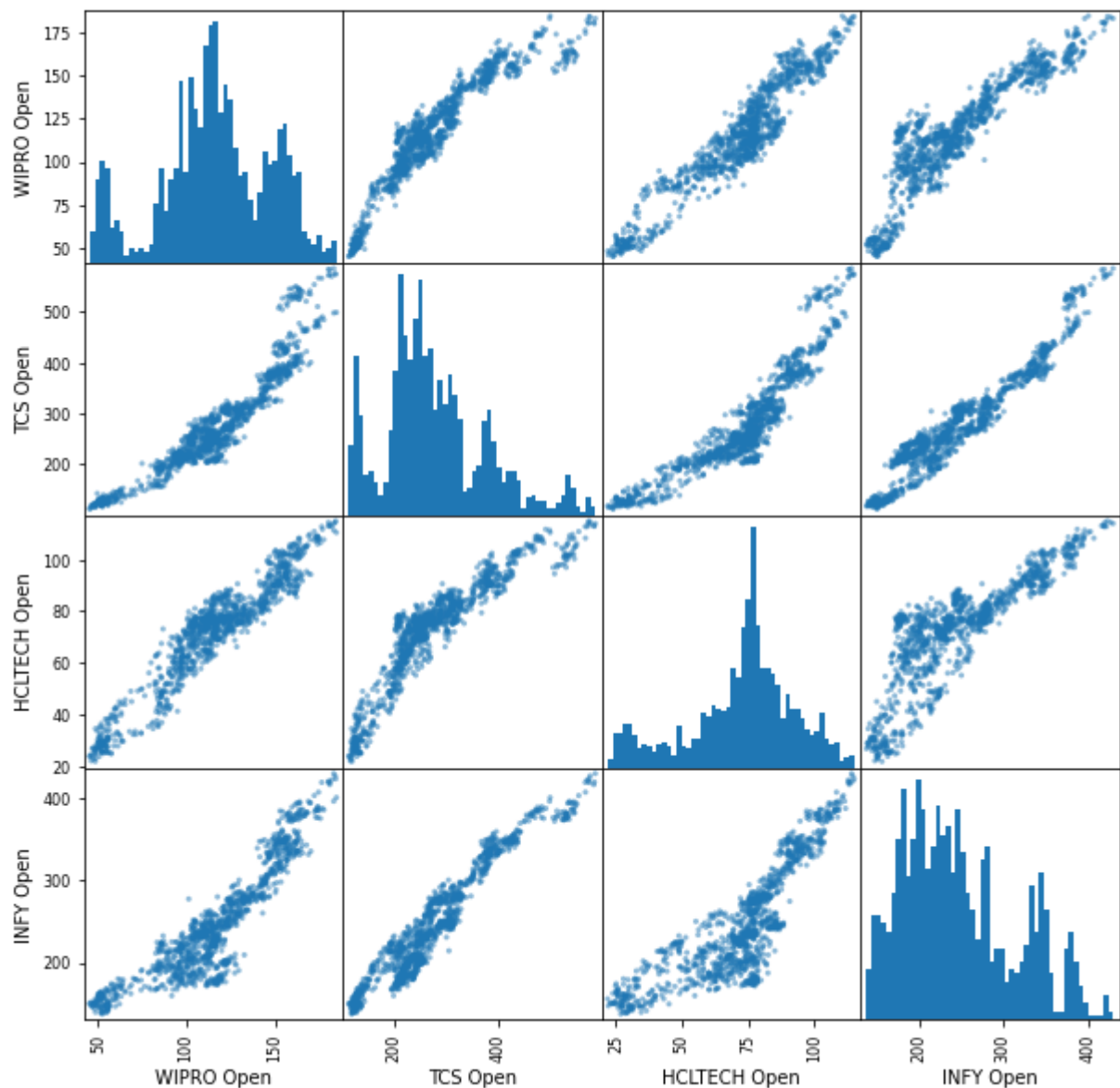
```
Out[118]:
```

	Open	High	Low	Close	Volume	Total Traded
Date						
2010-10-26	531.75	533.88	520.78	531.3	1753512	2.587501e+09

```
In [123... from pandas.plotting import scatter_matrix
```

```
In [126... tech = pd.concat([wipro['Open'], tcs['Open'], hcl['Open'], infosys['Open']], axis = 1)
tech.columns = ['WIPRO Open', 'TCS Open', 'HCLTECH Open', 'INFY Open']
```

```
In [171... scatter_matrix(tech, figsize = (10, 10), hist_kwds={'bins':50})
plt.show()
```



In [131... `!pip install mpl_finance`

```
Collecting mpl_finance
  Downloading mpl_finance-0.10.1-py3-none-any.whl (8.4 kB)
Requirement already satisfied: matplotlib in c:\users\urmil\anaconda3\lib\site-packages (from mpl_finance) (3.5.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\urmil\anaconda3\lib\site-packages (from matplotlib->mpl_finance) (1.3.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\urmil\anaconda3\lib\site-packages (from matplotlib->mpl_finance) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\urmil\anaconda3\lib\site-packages (from matplotlib->mpl_finance) (3.0.4)
Requirement already satisfied: packaging>=20.0 in c:\users\urmil\anaconda3\lib\site-packages (from matplotlib->mpl_finance) (21.3)
Requirement already satisfied: numpy>=1.17 in c:\users\urmil\anaconda3\lib\site-packages (from matplotlib->mpl_finance) (1.21.5)
Requirement already satisfied: pillow>=6.2.0 in c:\users\urmil\anaconda3\lib\site-packages (from matplotlib->mpl_finance) (9.0.1)
Requirement already satisfied: cycler>=0.10 in c:\users\urmil\anaconda3\lib\site-packages (from matplotlib->mpl_finance) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\urmil\anaconda3\lib\site-packages (from matplotlib->mpl_finance) (4.25.0)
Requirement already satisfied: six>=1.5 in c:\users\urmil\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->mpl_finance) (1.16.0)
Installing collected packages: mpl-finance
Successfully installed mpl-finance-0.10.1
```

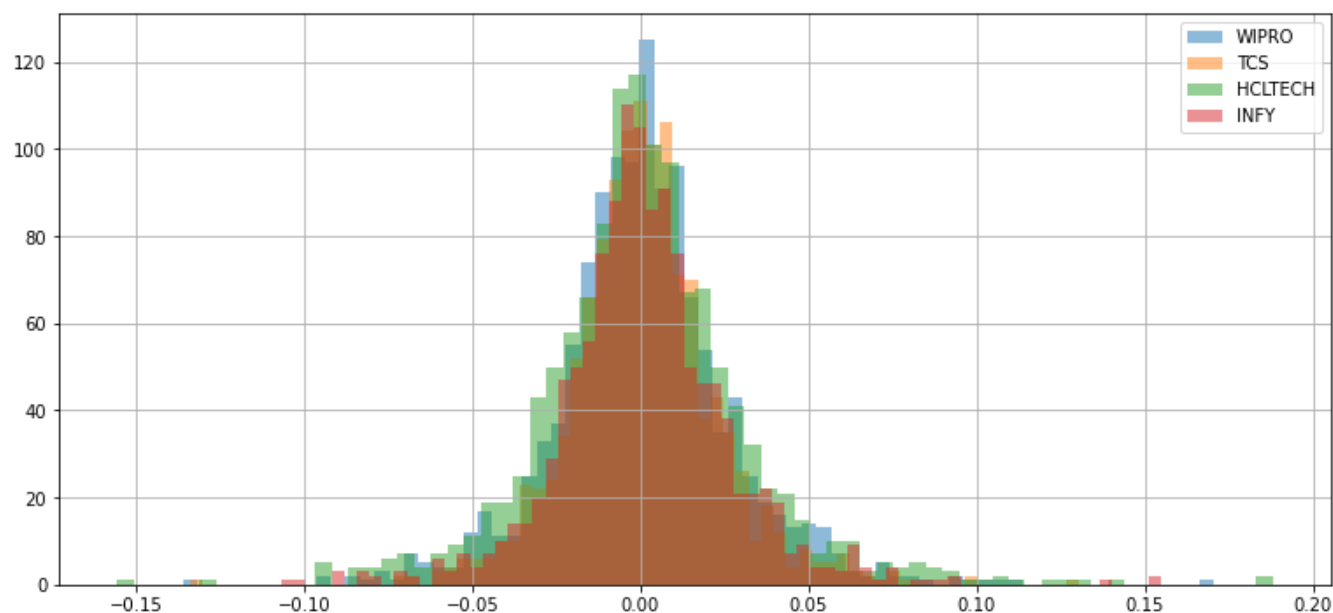
```
from mpl_finance import candlestick_ohlc from matplotlib.dates import DateFormatter, date2num, DayLocator, MONDAY
```

```
wipro_reset = wipro.iloc['2006-01':'2006-01'].reset_index()
```

```
In [143... wipro['returns'] = (wipro['Close']/wipro['Close'].shift(1)) - 1
tcs['returns'] = (tcs['Close']/tcs['Close'].shift(1)) - 1
hcl['returns'] = (hcl['Close']/hcl['Close'].shift(1)) - 1
infosys['returns'] = (infosys['Close']/infosys['Close'].shift(1)) - 1
```

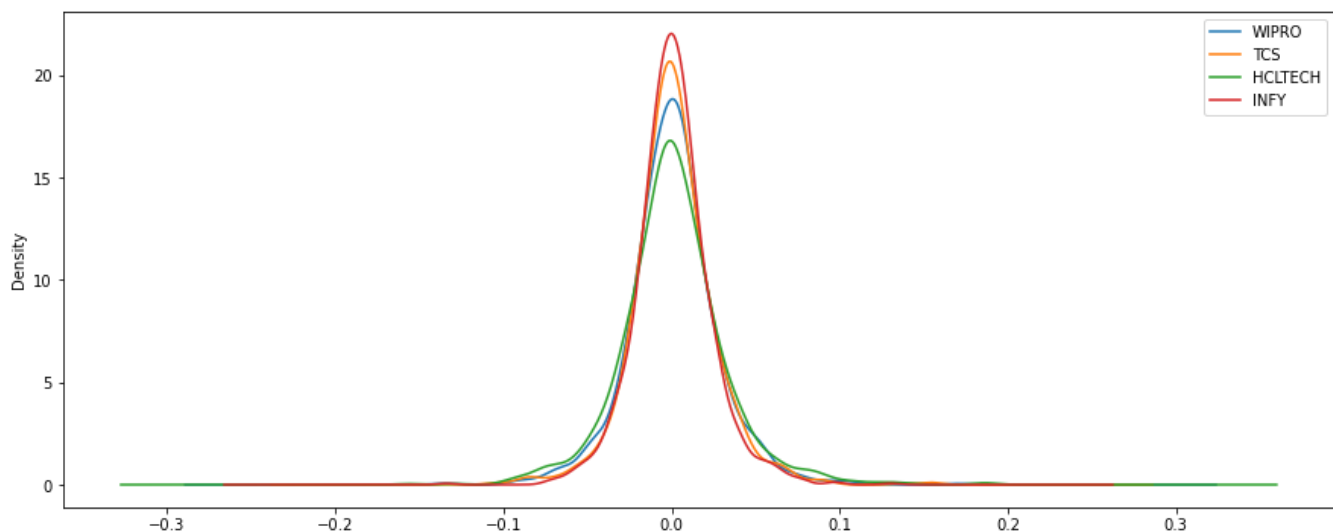
```
In [154... wipro['returns'].hist(bins = 70, label = 'WIPRO', alpha = 0.5, figsize = (13, 6))
infosys['returns'].hist(bins = 70, label = 'TCS', alpha = 0.5)
hcl['returns'].hist(bins = 70, label = 'HCLTECH', alpha = 0.5)
tcs['returns'].hist(bins = 70, label = 'INFY', alpha = 0.5)
plt.legend()
```

Out[154]: <matplotlib.legend.Legend at 0x17b89b83880>



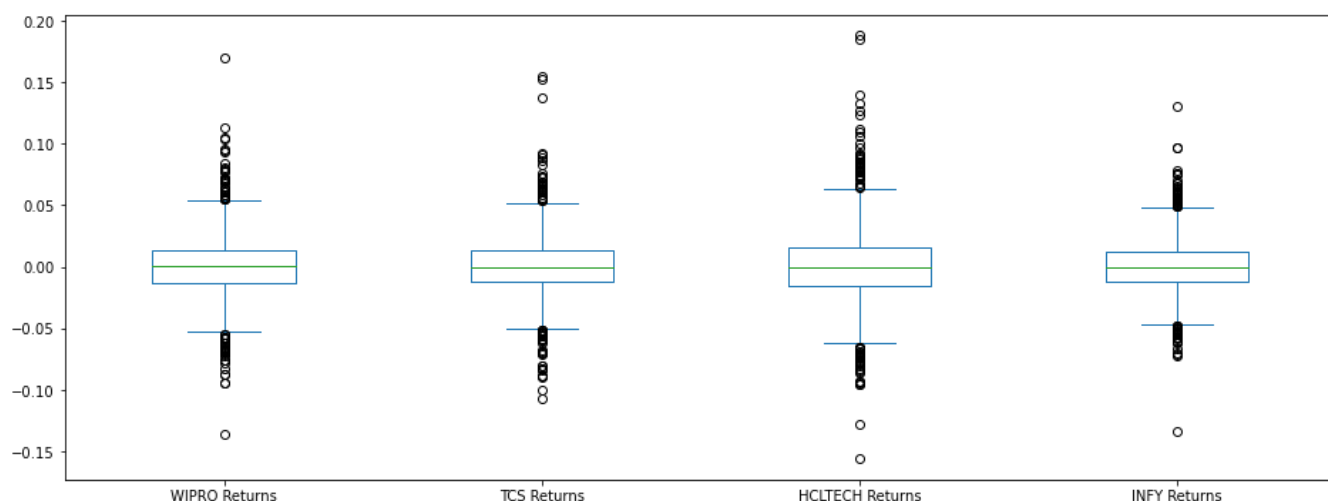
```
In [156... wipro['returns'].plot(kind='kde', label = 'WIPRO', figsize = (15, 6))
tcs['returns'].plot(kind='kde', label = 'TCS')
hcl['returns'].plot(kind='kde', label = 'HCLTECH')
infosys['returns'].plot(kind='kde', label = 'INFY')
plt.legend()
```

Out[156]: <matplotlib.legend.Legend at 0x17b89f4d850>

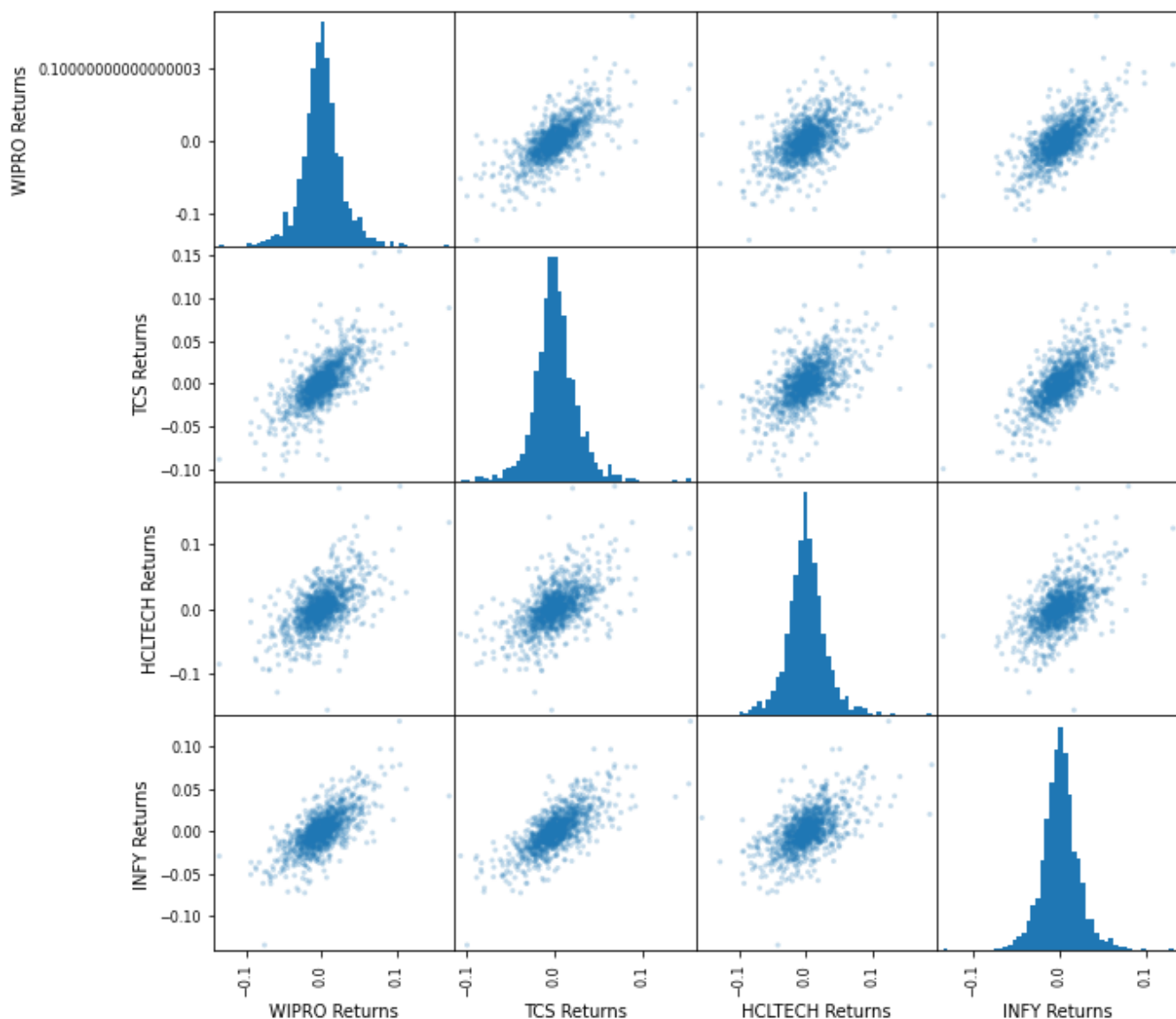


```
In [157... box_df = pd.concat([wipro['returns'], tcs['returns'], hcl['returns'], infosys['returns']], a
box_df.columns = ['WIPRO Returns', 'TCS Returns', 'HCLTECH Returns', 'INFY Returns']
box_df.plot(kind = 'box', figsize = (16,6))
```

Out[157]: <AxesSubplot:>



```
In [170... scatter_matrix(box_df, figsize = (10, 10), hist_kws={'bins':50}, alpha = 0.25)
plt.show()
```



```
In [165... wipro['cumulative_returns'] = (1 + wipro['returns']).cumprod()
tcs['cumulative_returns'] = (1 + tcs['returns']).cumprod()
```

```
hcl['cumulative_returns'] = (1 + hcl['returns']).cumprod()  
infosys['cumulative_returns'] = (1 + infosys['returns']).cumprod()
```

In [169...

```
wipro['cumulative_returns'].plot(label='WIPRO', figsize = (15, 7))  
tcs['cumulative_returns'].plot(label='TCS')  
hcl['cumulative_returns'].plot(label='HCLTECH')  
infosys['cumulative_returns'].plot(label='INFY')  
plt.title('Cumulative Return v/s Time')  
plt.legend()  
plt.show()
```

