Reliance comm vs Bharti Airtel Stock Analysis

by Bibhash Kalita

[35]
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

[3]
```
import pandas_datareader
import datetime
```

[4]
```
import pandas_datareader.data as web
```

[5]
```
start = datetime.datetime(2017, 1, 1)
end = datetime.datetime(2018, 5, 8)
relcom = web.DataReader("NSE/RCOM",'quandl', start, end)
```

[6]
```
relcom.head()
```

	Open	High	Low	Last	Close	TotalTradeQuantity	Tur
Date							
2018-05-07	15.20	15.8	15.15	15.50	15.55	46573218.0	719
2018-05-04	15.10	15.9	14.50	15.30	15.35	85210340.0	129
2018-05-03	16.00	16.1	14.85	15.10	15.05	72648942.0	111
2018-05-02	15.25	17.9	15.00	16.15	16.05	170980066.0	286
2018-04-30	15.65	15.9	14.65	15.25	15.30	78831548.0	119

[7]
```
relcom.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 333 entries, 2018-05-07 to 2017-01-02
Data columns (total 7 columns):
Open                 333 non-null float64
High                 333 non-null float64
Low                  333 non-null float64
Last                 333 non-null float64
Close                333 non-null float64
TotalTradeQuantity   333 non-null float64
TurnoverLacs         333 non-null float64
dtypes: float64(7)
memory usage: 20.8 KB
```

[8]
```
airtel = web.DataReader("NSE/BHARTIARTL",'quandl', start, end)
```

[9]
```
airtel.head()
```

	Open	High	Low	Last	Close	TotalTradeQuantity
Date						
2018-05-07	401.50	404.55	395.5	398.50	398.25	5906011.0
2018-05-04	410.85	418.35	394.0	398.00	396.75	14415306.0
2018-05-03	406.30	408.00	399.0	405.05	404.40	3716970.0
2018-05-02	412.90	412.90	406.0	409.60	408.45	6087447.0
2018-04-30	407.00	413.25	404.2	410.00	409.55	3274387.0

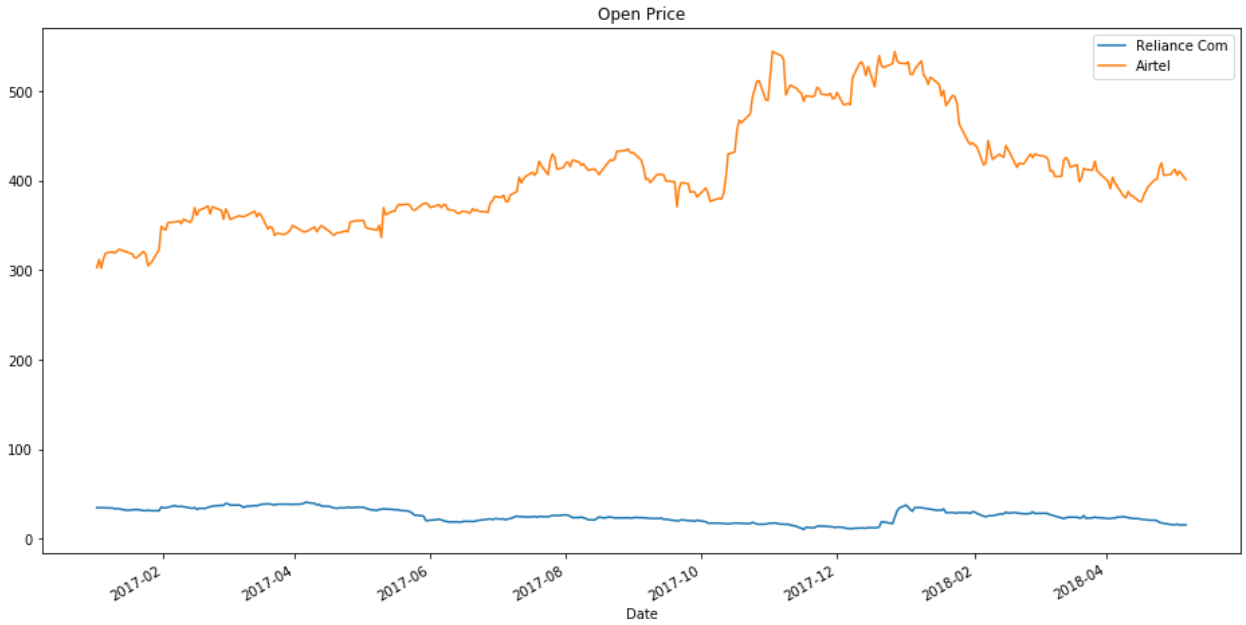
[10]
```
airtel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 333 entries, 2018-05-07 to 2017-01-02
Data columns (total 7 columns):
Open                 333 non-null float64
High                 333 non-null float64
Low                  333 non-null float64
Last                 333 non-null float64
Close                333 non-null float64
TotalTradeQuantity   333 non-null float64
TurnoverLacs         333 non-null float64
```

```
dtypes: float64(7)
memory usage: 20.8 KB
```

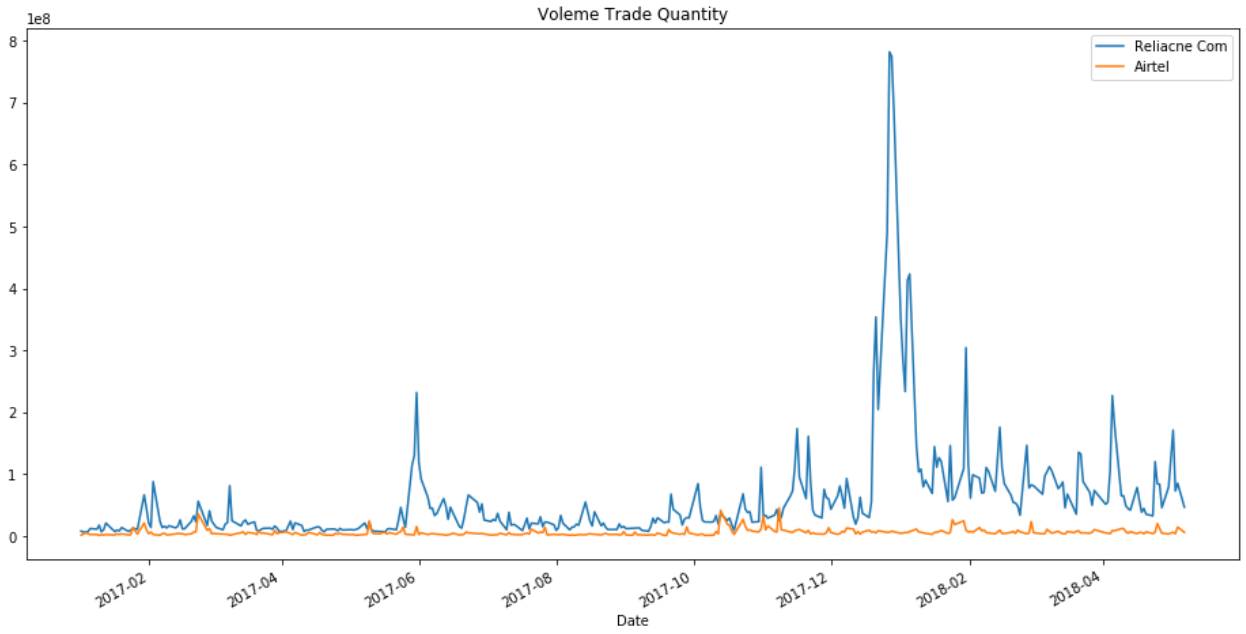
[11]
```
relcom['Open'].plot(label='Reliance Com',figsize=(16,8),title='Open Price
airtel['Open'].plot(label='Airtel')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x2e77244c588>
```



[12]
```
relcom['TotalTradeQuantity'].plot(label='Reliacne Com',figsize=(16,8),tit
airtel['TotalTradeQuantity'].plot(label='Airtel')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x2e772322630>
```



[13]
```
relcom['TotalTradeQuantity'].argmax()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
FutureWarning: 'argmax' is deprecated. Use 'idxmax' instead. The behavior
of 'argmax' will be corrected to return the positional maximum in the
future. Use 'series.values.argmax' to get the position of the maximum now.
  """Entry point for launching an IPython kernel.
Timestamp('2017-12-27 00:00:00')
```

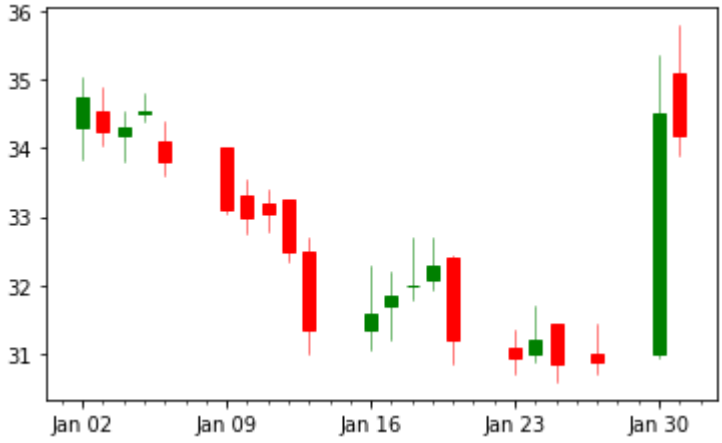
[14]
```
relcom['TurnoverLacs'].mean()
```

```
14781.744324324309
```

[15]
```
airtel['TurnoverLacs'].mean()
```

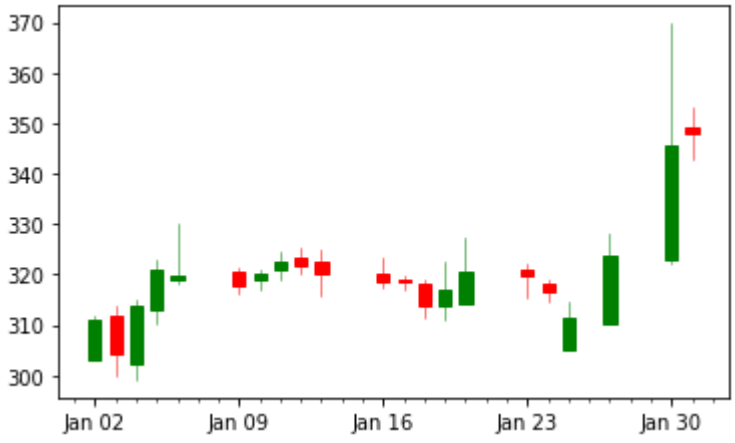
```
24936.075975975993
```

[16]
```
from matplotlib.finance import candlestick_ohlc
from matplotlib.dates import DateFormatter, date2num, WeekdayLocator, Day

# Rest the index to get a column of January Dates
relcom_reset = relcom.loc['2017-01':'2017-01'].reset_index()

# Create a new column of numerical "date" values for matplotlib to use
relcom_reset['date_ax'] = relcom_reset['Date'].apply(lambda date: date2nu
relcom_values = [tuple(vals) for vals in relcom_reset[['date_ax', 'Open',

mondays = WeekdayLocator(MONDAY)        # major ticks on the mondays
alldays = DayLocator()              # minor ticks on the days
weekFormatter = DateFormatter('%b %d')  # e.g., Jan 12
dayFormatter = DateFormatter('%d')      # e.g., 12

#Plot it
fig, ax = plt.subplots()
fig.subplots_adjust(bottom=0.2)
ax.xaxis.set_major_locator(mondays)
ax.xaxis.set_minor_locator(alldays)
ax.xaxis.set_major_formatter(weekFormatter)

candlestick_ohlc(ax, relcom_values, width=0.6, colorup='g',colordown='r')
```



[18]

```
from matplotlib.finance import candlestick_ohlc
from matplotlib.dates import DateFormatter, date2num, WeekdayLocator, Day

# Rest the index to get a column of January Dates
airtel_reset = airtel.loc['2017-01':'2017-01'].reset_index()

# Create a new column of numerical "date" values for matplotlib to use
airtel_reset['date_ax'] = airtel_reset['Date'].apply(lambda date: date2nu
airtel_values = [tuple(vals) for vals in airtel_reset[['date_ax', 'Open',

mondays = WeekdayLocator(MONDAY)        # major ticks on the mondays
alldays = DayLocator()              # minor ticks on the days
weekFormatter = DateFormatter('%b %d')  # e.g., Jan 12
dayFormatter = DateFormatter('%d')      # e.g., 12

#Plot it
fig, ax = plt.subplots()
fig.subplots_adjust(bottom=0.2)
ax.xaxis.set_major_locator(mondays)
ax.xaxis.set_minor_locator(alldays)
ax.xaxis.set_major_formatter(weekFormatter)

candlestick_ohlc(ax, airtel_values, width=0.6, colorup='g',colordown='r')
```



Daily Percentage Change

[19]
```
relcom['returns'] = relcom['Close'].pct_change(1)
```

[20]
```
relcom.head()
```

	Open	High	Low	Last	Close	TotalTradeQuantity	Tur
Date							
2018-05-07	15.20	15.8	15.15	15.50	15.55	46573218.0	719
2018-05-04	15.10	15.9	14.50	15.30	15.35	85210340.0	129
2018-05-03	16.00	16.1	14.85	15.10	15.05	72648942.0	111
2018-05-02	15.25	17.9	15.00	16.15	16.05	170980066.0	286
2018-04-30	15.65	15.9	14.65	15.25	15.30	78831548.0	119

[21]
```
airtel['returns'] = airtel['Close'].pct_change(1)
```

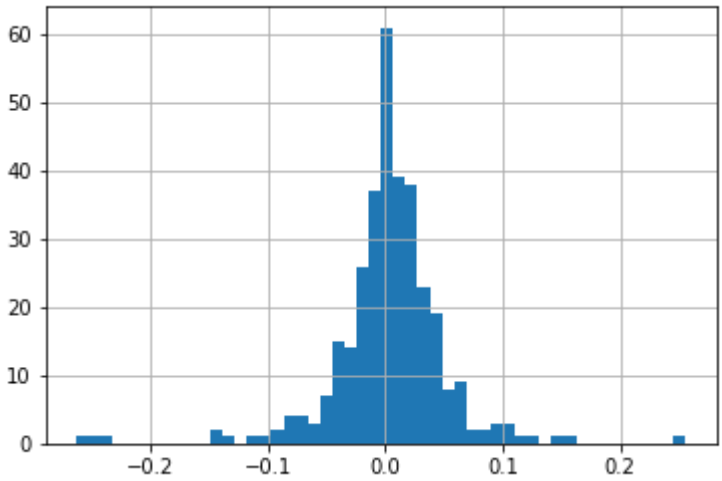
[22]
```
airtel.head()
```

	Open	High	Low	Last	Close	TotalTradeQuantity
Date						
2018-05-07	401.50	404.55	395.5	398.50	398.25	5906011.0
2018-05-04	410.85	418.35	394.0	398.00	396.75	14415306.0
2018-05-03	406.30	408.00	399.0	405.05	404.40	3716970.0
2018-05-02	412.90	412.90	406.0	409.60	408.45	6087447.0

	Open	High	Low	Last	Close	TotalTradeQuantity
Date						
2018-04-30	407.00	413.25	404.2	410.00	409.55	3274387.0

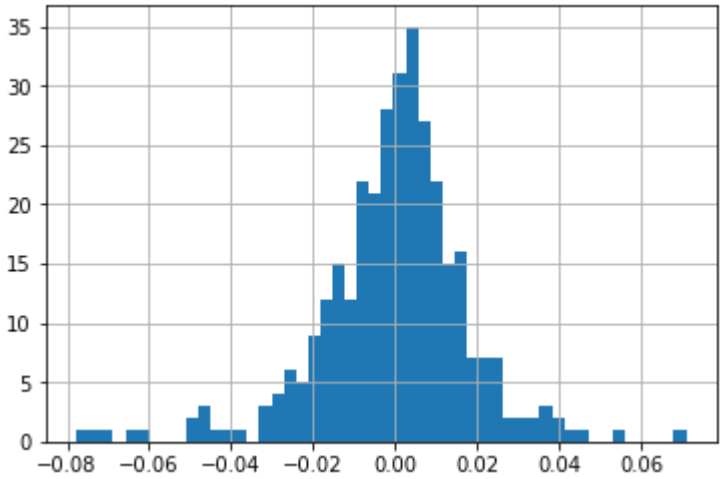
[23]
```
relcom['returns'].hist(bins=50)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e773ba7978>
```



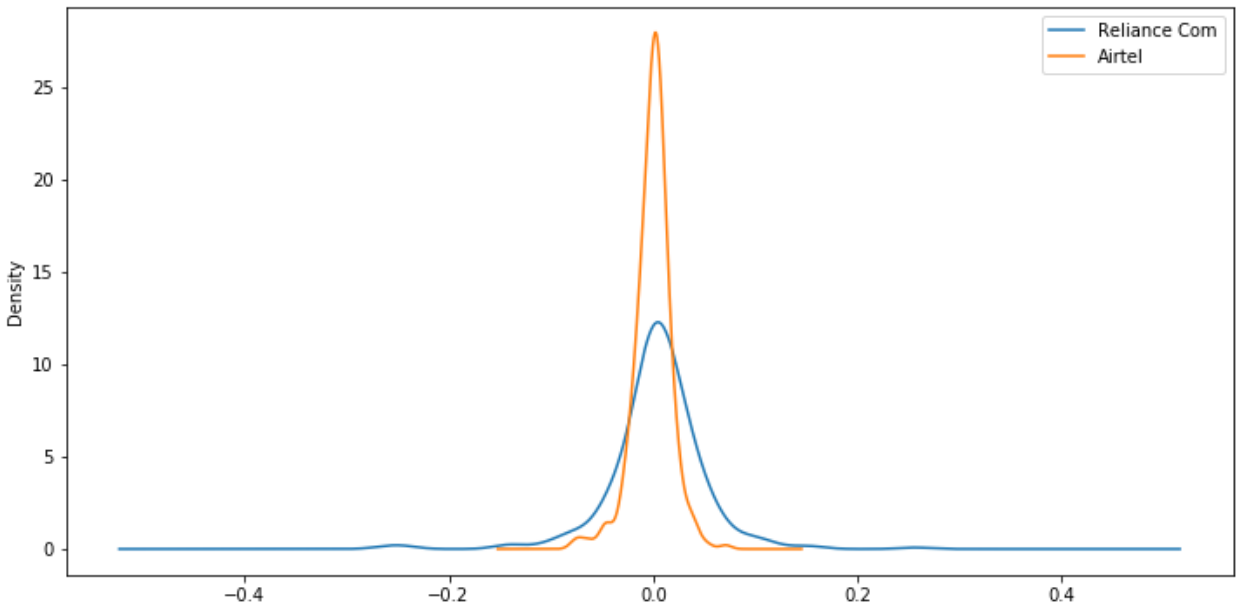
[24]
```
airtel['returns'].hist(bins=50)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e773157a90>
```



[25]
```
relcom['returns'].plot(kind='kde',label='Reliance Com',figsize=(12,6))
airtel['returns'].plot(kind='kde',label='Airtel')
plt.legend()
```

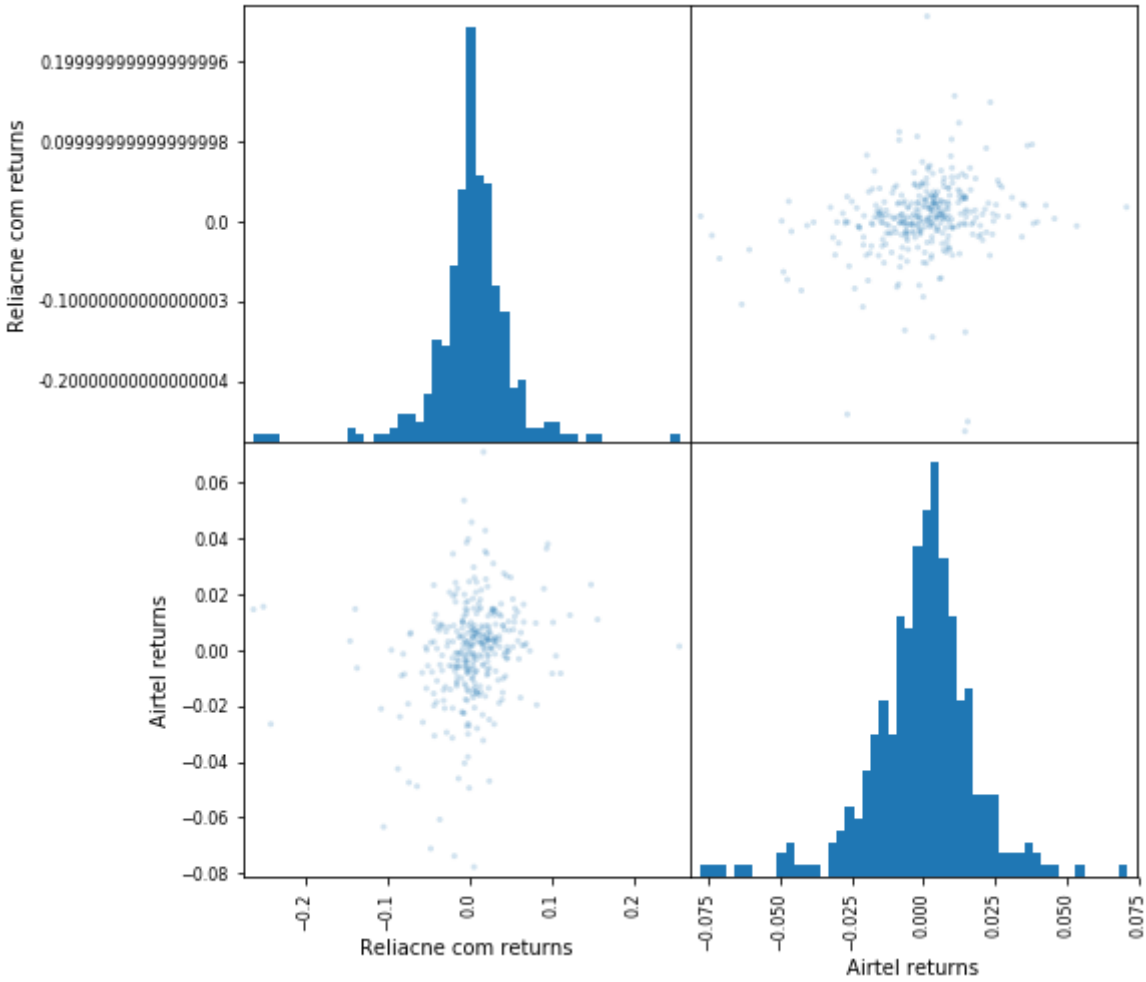
```
<matplotlib.legend.Legend at 0x2e77331c668>
```



[26]
```
box_df = pd.concat([relcom['returns'],airtel['returns']],axis=1)
box_df.columns = ['Reliacne com returns','Airtel returns']
box_df.plot(kind='box',figsize=(8,11),colormap='jet')
```

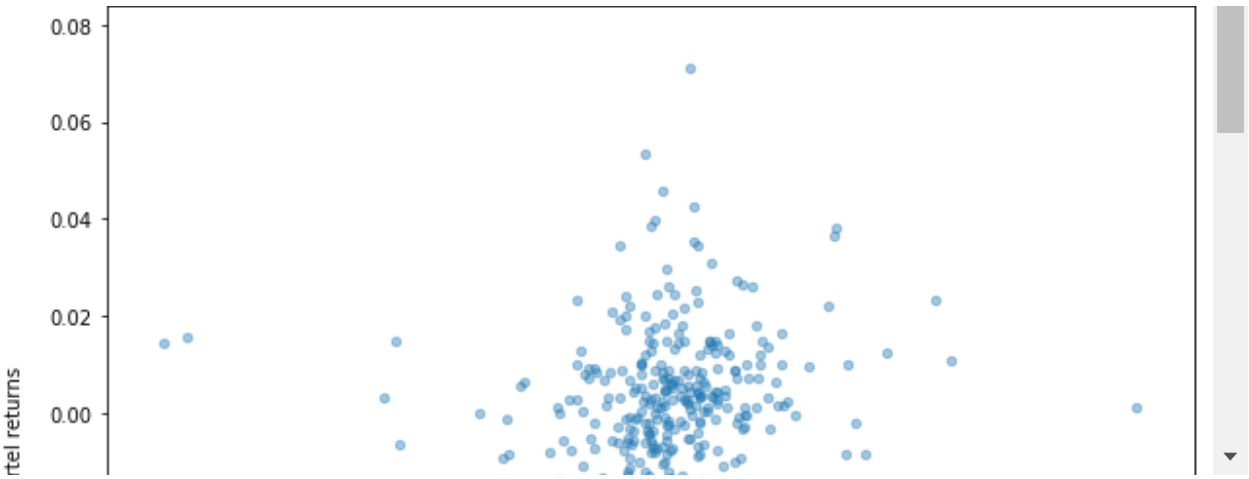
```
<matplotlib.axes._subplots.AxesSubplot at 0x2e7754b0898>
```

[27]
```
from pandas.plotting import scatter_matrix
scatter_matrix(box_df,figsize=(8,8),alpha=0.2,hist_kwds={'bins':50});
```



[29]
```
box_df.plot(kind='scatter',x='Reliacne com returns',y='Airtel returns',al
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e7754cc9b0>
```



Cumulative Daily Returns

```
[30] relcom['Cumulative Return'] = (1 + relcom['returns']).cumprod()
```

```
[31] airtel['Cumulative Return'] = (1 + airtel['returns']).cumprod()
```

```
[32] relcom.head()
```

	Open	High	Low	Last	Close	TotalTradeQuantity	Tu
Date							
2018-05-07	15.20	15.8	15.15	15.50	15.55	46573218.0	71
2018-05-04	15.10	15.9	14.50	15.30	15.35	85210340.0	12
2018-05-03	16.00	16.1	14.85	15.10	15.05	72648942.0	11
2018-05-02	15.25	17.9	15.00	16.15	16.05	170980066.0	28
2018-04-30	15.65	15.9	14.65	15.25	15.30	78831548.0	11

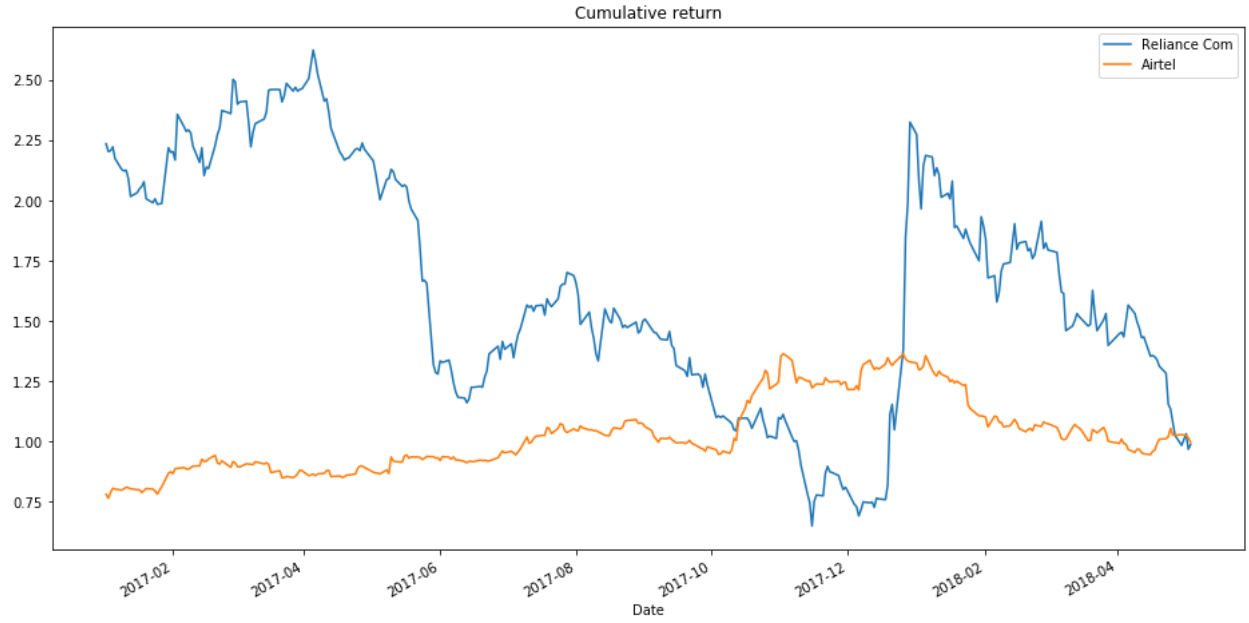
```
[33] airtel.head()
```

	Open	High	Low	Last	Close	TotalTradeQuantity

Date	Open	High	Low	Last	Close	TotalTradeQuantity
2018-05-07	401.50	404.55	395.5	398.50	398.25	5906011.0
Date						
2018-05-04	410.85	418.35	394.0	398.00	396.75	14415306.0
2018-05-03	406.30	408.00	399.0	405.05	404.40	3716970.0
2018-05-02	412.90	412.90	406.0	409.60	408.45	6087447.0
2018-04-30	407.00	413.25	404.2	410.00	409.55	3274387.0

[34]
```
relcom['Cumulative Return'].plot(label='Reliance Com',figsize=(16,8),titl
airtel['Cumulative Return'].plot(label='Airtel')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x2e775c11400>
```



Realiance vs Airtel who wins you decide