In [1]:

```
pip install yfinance
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.7/dist-packages (0.1.59
)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (fro
m yfinance) (1.19.5)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.7/dist-packages (fr
om yfinance) (1.1.5)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.7/dist-packa
ges (from yfinance) (0.0.9)
Requirement already satisfied: lxml>=4.5.1 in /usr/local/lib/python3.7/dist-packages (fro
m yfinance) (4.6.3)
Requirement already satisfied: requests>=2.20 in /usr/local/lib/python3.7/dist-packages (
from yfinance) (2.23.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-pa
ckages (from pandas>=0.24->yfinance) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages (fr
om pandas>=0.24->yfinance) (2018.9)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packag
es (from requests>=2.20->yfinance) (2020.12.5)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (fr
om requests>=2.20->yfinance) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-package
s (from requests>=2.20->yfinance) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/
python3.7/dist-packages (from requests>=2.20->yfinance) (1.24.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from p
ython-dateutil>=2.7.3->pandas>=0.24->yfinance) (1.15.0)
```

In [2]:

```
import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
```

https://in.finance.yahoo.com/quote/INFY.NS/history?p=INFY.NS&.tsrc=fin-srch

https://in.finance.yahoo.com/quote/%5ENSEI/history?p=%5ENSEI&.tsrc=fin-srch

1

In [3]:

```
df1 = yf.download('INFY.NS',start='2020-03-17',end='2021-03-19',progress=False)
```

In [4]:

```
df2 = yf.download('^NSEI',start='2020-03-17',end='2021-03-19',progress=False)
```

In [5]:

```
df1t=df1
df2t=df2
```

In [6]:

```
df1t.drop(df1t.columns[[0,1,2,3,5]], axis = 1, inplace = True)
df2t.drop(df2t.columns[[0,1,2,3,5]], axis = 1, inplace = True)
```

In [7]:

```
df1t.rename(columns = {'Adj Close':'adjclose'}, inplace = True)
```

```
df2t.rename(columns = {'Adj Close':'adjclose'}, inplace = True)
```

In [8]:

```
# df1t
```

In [9]:

```
# df1t.adjclose
```

In [10]:

```
# df2t.adjclose
```

In [11]:

```
df3t=pd.merge(df1t, df2t, on='Date')
```

In [12]:

```
df3t
```

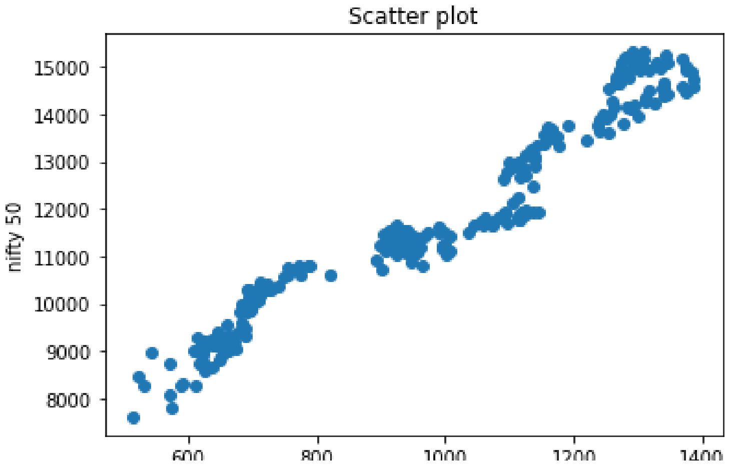
Out[12]:

	adjclose_x	adjclose_y
Date		
2020-03-17	541.678284	8967.049805
2020-03-18	521.522156	8468.799805
2020-03-19	532.503113	8263.450195
2020-03-20	571.204834	8745.450195
2020-03-23	513.859863	7610.250000
...	...	...
2021-03-12	1374.849976	15030.950195
2021-03-15	1373.599976	14929.500000
2021-03-16	1384.000000	14910.450195
2021-03-17	1387.000000	14721.299805
2021-03-18	1337.099976	14557.849609

250 rows × 2 columns

In [13]:

```
plt.scatter(df3t.adjclose_x, df3t.adjclose_y)
plt.title('Scatter plot')
plt.xlabel('infosys')
plt.ylabel('nifty 50')
plt.show()
```



In [14]:

```
len(df3t)
```

Out[14]:

```
250
```

In [14]:

```

```

In [15]:

```
def estimate_coef(x, y):
    # number of observations/points
    n = np.size(x)

    # mean of x and y vector
    m_x = np.mean(x)
    m_y = np.mean(y)

    # calculating cross-deviation and deviation about x
    SS_xy = np.sum(y*x) - n*m_y*m_x
    SS_xx = np.sum(x*x) - n*m_x*m_x

    # calculating regression coefficients
    b_1 = SS_xy / SS_xx
    b_0 = m_y - b_1*m_x

    return (b_0, b_1)
```

In [16]:

```
def plot_regression_line(x, y, b):
    # plotting the actual points as scatter plot
    plt.scatter(x, y,
               marker = "o", s = 30)

    # predicted response vector
    y_pred = b[0] + b[1]*x

    # plotting the regression line
    plt.plot(x, y_pred, color = "g")

    # putting labels
    plt.xlabel('infosys')
    plt.ylabel('nifty50')

    # function to show plot
    plt.show()
```

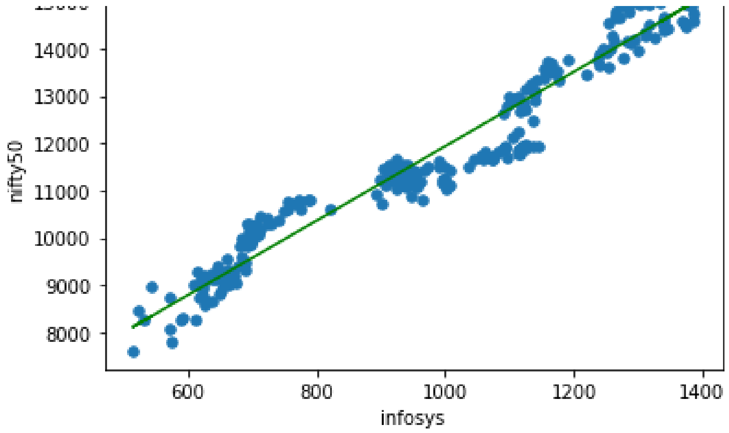
In [17]:

```
x = np.array(df3t['adjclose_x'].tolist())
y = np.array(df3t['adjclose_y'].tolist())

    # estimating coefficients
b = estimate_coef(x, y)
print("Estimated coefficients:\n intercept = {}  \
      \n slope = {}".format(b[0], b[1]))

plot_regression_line(x, y, b)
```

```
Estimated coefficients:
 intercept = 4074.4829670204153
 slope = 7.859117822076629
```



In [20]:

```
n = np.size(x)
y_pred = b[0] + b[1]*x
m_y = np.mean(y)
```

In [21]:

```
print('squared error is', np.sum((y - y_pred)**2))
print('mean squared error is', (((np.sum((y - y_pred)**2)))/n))
print('root mean square error is', (np.sqrt((np.sum((y - y_pred)**2))/n)))
print('R square is', 1-((((np.sum((y - y_pred)**2))))/(np.sum((y - m_y)**2))))
```

```
squared error is 59418578.00548892
mean squared error is 237674.31202195567
root mean square error is 487.5185247987564
R square is 0.9420521950177291
```

In [21]:

```

```

preparing data for using with skikit library for comparison

In [22]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: p
andas.util.testing is deprecated. Use the functions in the public API at pandas.testing i
nstead.
  import pandas.util.testing as tm
```

In [23]:

```
xst = df3t['adjclose_x'].tolist()
xst = np.array(xst)
xst = xst.reshape(-1,1)
```

In [24]:

```
#xst
```

In [25]:

```
regression_model = LinearRegression()

regression_model.fit(xst, df3t.adjclose_y)
ypst = regression_model.predict(xst)
mse=mean_squared_error(df3t.adjclose_y,ypst)

rmse = np.sqrt(mean_squared_error(df3t.adjclose_y, ypst))
```

```
r2 = r2_score(df3t.adjclose_y, ypst)

print('Slope:' ,regression_model.coef_)
print('Intercept:', regression_model.intercept_)
print('MSE:',mse)
print('Root mean squared error: ', rmse)
print('R2 score: ', r2)
```

```
Slope: [7.85911782]
Intercept: 4074.4829670204253
MSE: 237674.3120219556
Root mean squared error:  487.51852479875635
R2 score:  0.9420521950177292
```

In [43]:

```
# df3t.to_csv('reg.csv', index = True)
```

In [ ]:

```

```