In [12]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

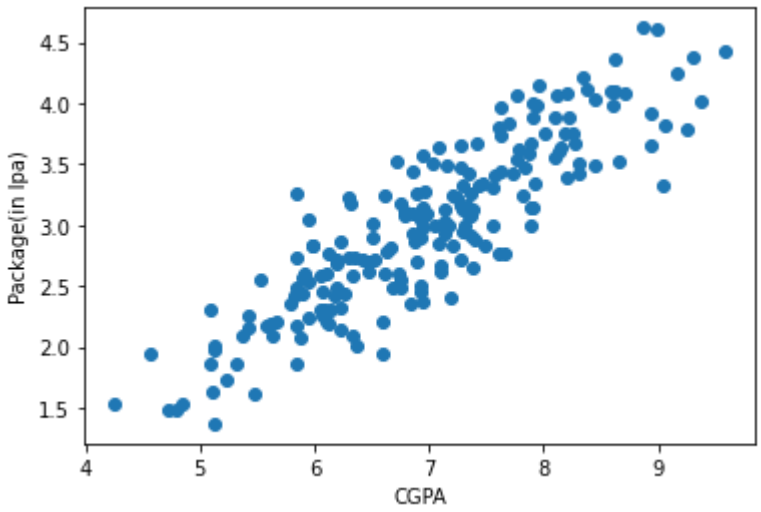
In [13]:
```
df = pd.read_csv('placement.csv')
```

In [37]:
```
df.head()
df.shape
```

Out[37]: (200, 2)

In [15]:
```
plt.scatter(df['cgpa'],df['package'])
plt.xlabel('CGPA')
plt.ylabel('Package(in lpa)')
```

Out[15]: Text(0, 0.5, 'Package(in lpa)')



In [16]:
```
X = df.iloc[:,0:1]
y = df.iloc[:,-1]
```

In [6]:
```
y
```

Out[6]:
```
0      3.26
1      1.98
2      3.25
3      3.67
4      3.57
       ...
195    2.46
196    2.57
197    3.24
198    3.96
199    2.33
Name: package, Length: 200, dtype: float64
```

In [17]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_st
```

In [18]:
```
from sklearn.linear_model import LinearRegression
```

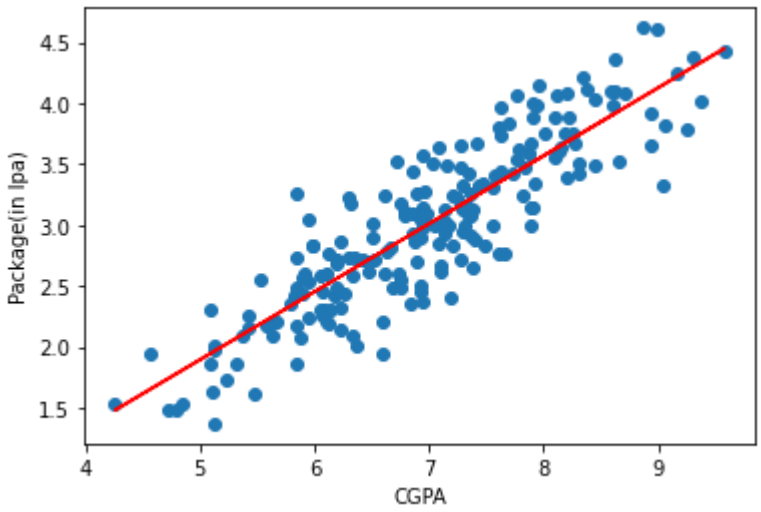
In [19]:
```
lr = LinearRegression()
```

In [20]:
```
lr.fit(X_train,y_train)
```

Out[20]:
```
LinearRegression()
```

In [21]:
```
plt.scatter(df['cgpa'],df['package'])
plt.plot(X_train,lr.predict(X_train),color='red')
plt.xlabel('CGPA')
plt.ylabel('Package(in lpa)')
```

Out[21]:
```
Text(0, 0.5, 'Package(in lpa)')
```



In [22]:
```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

In [26]:
```
y_pred = lr.predict(X_test)
```

In [25]:
```
y_test.values
```

Out[25]:
```
array([4.1 , 3.49, 2.08, 2.33, 1.94, 1.48, 1.86, 3.09, 4.21, 2.87, 3.65,
       4.  , 2.89, 2.6 , 2.99, 3.25, 1.86, 3.67, 2.37, 3.42, 2.48, 3.65,
       2.6 , 2.83, 4.08, 2.56, 3.58, 3.81, 4.09, 2.01, 3.63, 2.92, 3.51,
       1.94, 2.21, 3.34, 3.34, 3.23, 2.01, 2.61])
```

In [27]:
```
print("MAE",mean_absolute_error(y_test,y_pred))
```

```
MAE 0.2884710931878175
```

In [28]:
```
print("MSE",mean_squared_error(y_test,y_pred))
```

```
MSE 0.12129235313495527
```

In [29]:
```
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
```

```
RMSE 0.34827051717731616
```

In [31]:
```
print("MSE",r2_score(y_test,y_pred))
r2 = r2_score(y_test,y_pred)
```

```
MSE 0.780730147510384
```

In [33]:
```
# Adjusted R2 score
X_test.shape
```

Out[33]: (40, 1)

In [34]:
```
1 - ((1-r2)*(40-1)/(40-1-1))
```

Out[34]: 0.7749598882343415

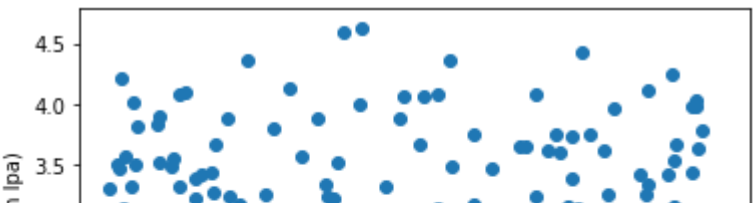
In [157…
```
new_df1 = df.copy()
new_df1['random_feature'] = np.random.random(200)

new_df1 = new_df1[['cgpa','random_feature','package']]
new_df1.head()
```

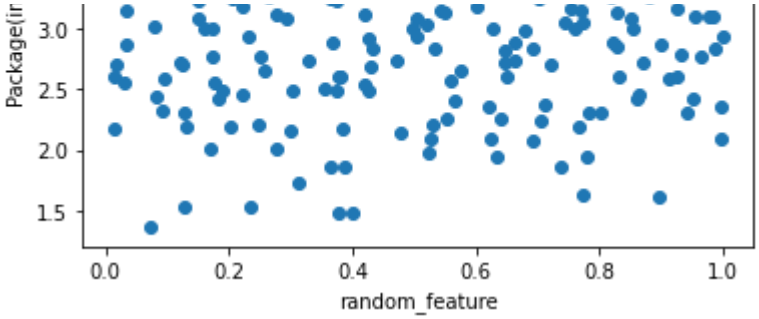
Out[157…

	cgpa	random_feature	package
0	6.89	0.309720	3.26
1	5.12	0.302370	1.98
2	7.82	0.171968	3.25
3	7.42	0.585555	3.67
4	6.94	0.171229	3.57

In [42]:
```
plt.scatter(new_df1['random_feature'],new_df1['package'])
plt.xlabel('random_feature')
plt.ylabel('Package(in lpa)')
```

Out[42]: Text(0, 0.5, 'Package(in lpa)')





In [50]:
```
X = new_df1.iloc[:,0:2]
y = new_df1.iloc[:,-1]
```

In [51]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_st
```

In [52]:
```
lr = LinearRegression()
```

In [53]:
```
lr.fit(X_train,y_train)
```

Out[53]:
```
LinearRegression()
```

In [54]:
```
y_pred = lr.predict(X_test)
```

In [55]:
```
print("R2 score",r2_score(y_test,y_pred))
r2 = r2_score(y_test,y_pred)
```

```
R2 score 0.781133851209665
```

In [56]:
```
1 - ((1-r2)*(40-1)/(40-1-2))
```

Out[56]:
```
0.7693032485723497
```

In [148…
```
new_df2 = df.copy()

new_df2['iq'] = new_df2['package'] + (np.random.randint(-12,12,200)/10)

new_df2 = new_df2[['cgpa','iq','package']]
```

In [149…
```
new_df2.sample(5)
```

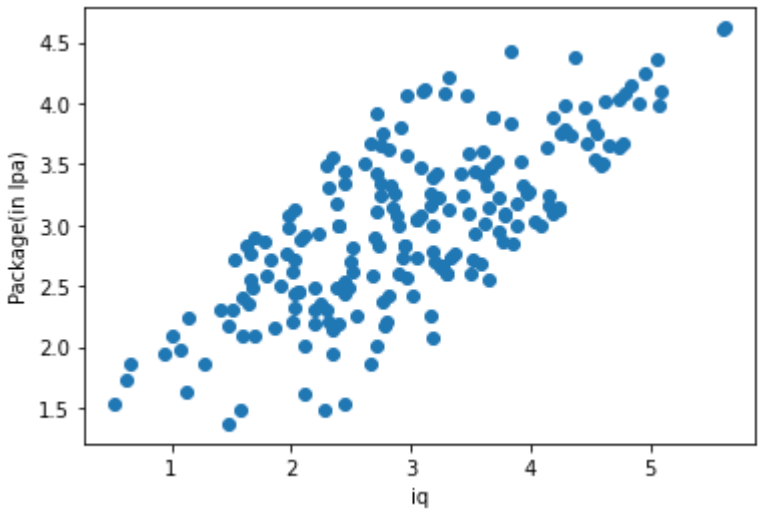
Out[149…

	cgpa	iq	package
42	5.95	2.44	2.54
180	6.19	1.52	2.72
38	8.62	5.06	4.36
23	6.19	2.38	2.48

98	7.69	3.83	3.83

In [150…
```
plt.scatter(new_df2['iq'],new_df2['package'])
plt.xlabel('iq')
plt.ylabel('Package(in lpa)')
```

Out[150…
```
Text(0, 0.5, 'Package(in lpa)')
```



In [126…
```
np.random.randint(-100,100)
```

Out[126…
```
44
```

In [151…
```
X = new_df2.iloc[:,0:2]
y = new_df2.iloc[:,-1]
```

In [152…
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_st
```

In [154…
```
lr = LinearRegression()
lr.fit(X_train,y_train)
y_pred = lr.predict(X_test)
```

In [155…
```
print("R2 score",r2_score(y_test,y_pred))
r2 = r2_score(y_test,y_pred)
```

```
R2 score 0.8000928965773431
```

In [156…
```
1 - ((1-r2)*(40-1)/(40-1-2))
```

Out[156…
```
0.7892871072031453
```

In [ ]:
```
```