

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Requirement already satisfied: seaborn in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (0.13.2)

Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from seaborn) (2.2.6)

Requirement already satisfied: pandas>=1.2 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from seaborn) (2.2.3)

Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from seaborn) (3.10.3)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.2)

Requirement already satisfied: cycler>=0.10 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.58.1)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)

Requirement already satisfied: packaging>=20.0 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.2)

Requirement already satisfied: pillow>=8 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.2.1)

Requirement already satisfied: pyparsing>=2.3.1 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.3)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\yadav\appdata\roaming\python\python313\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in c:\users\yadav\appdata\local\programs\python\python313\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)

Requirement already satisfied: six>=1.5 in c:\users\yadav\appdata\roaming\python\python313\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)

Note: you may need to restart the kernel to use updated packages.

```
df = pd.read_csv(r"D:\Machine learning\Linear Regression\placement.csv")
```

```
df.head()
```

	cgpa	package
0	6.89	3.26
1	5.12	1.98
2	7.82	3.25
3	7.42	3.67
4	6.94	3.57

```
df.shape
```

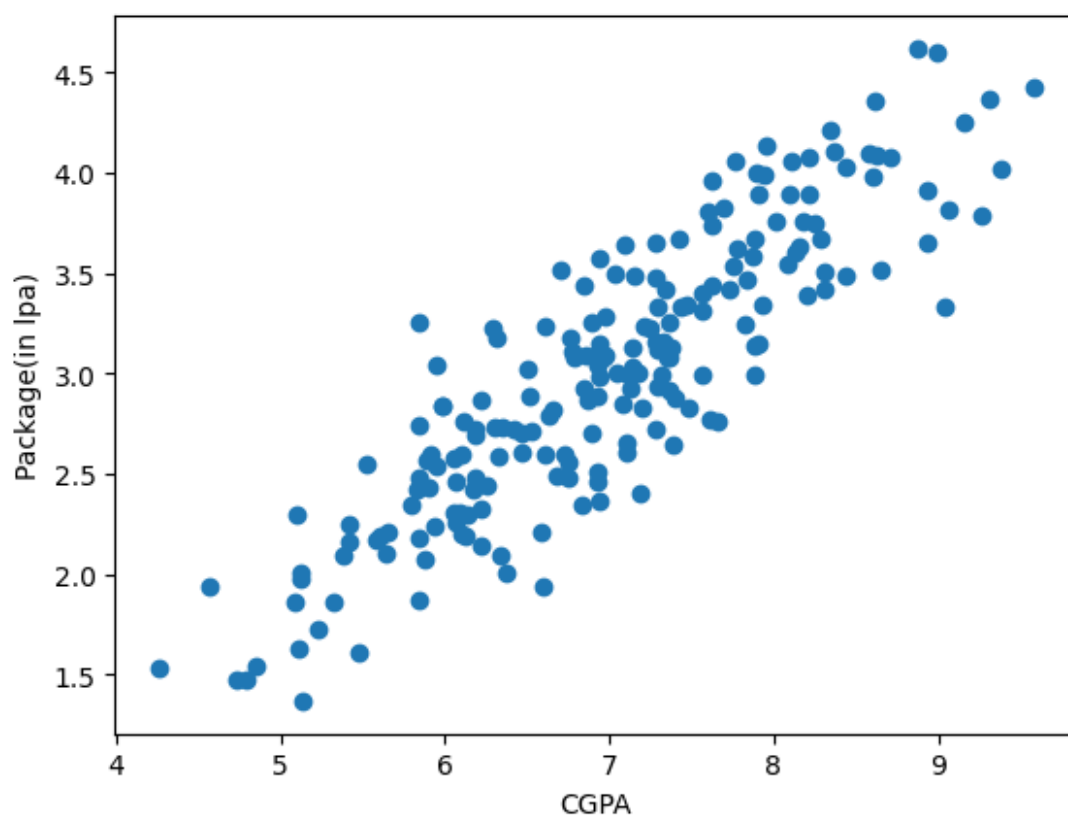
```
(200, 2)
```

```
plt.scatter(df['cgpa'],df['package'])
```

```
plt.xlabel('CGPA')
```

```
plt.ylabel('Package(in lpa)')
```

```
Text(0, 0.5, 'Package(in lpa)')
```



```
x =df.iloc[:,0:1]
```

```
x
```

```
      cgpa
0      6.89
1      5.12
2      7.82
3      7.42
4      6.94
..      ...
195    6.93
196    5.89
197    7.21
198    7.63
199    6.22
```

```
[200 rows x 1 columns]
```

```
y = df.iloc[:, -1]
```

```
y
```

```
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
```

```
from sklearn.model_selection import train_test_split
```

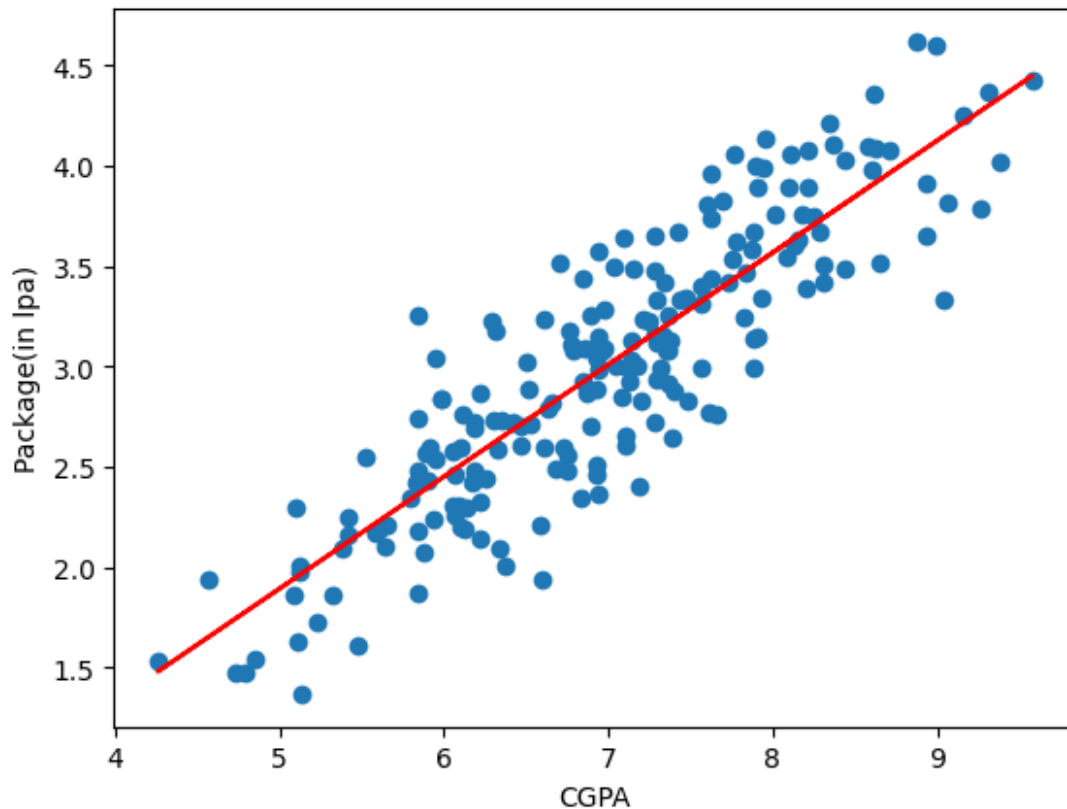
```
x_train,x_test,y_train,y_test =  
train_test_split(x,y,test_size=0.2,random_state=2)
```

```
from sklearn.linear_model import LinearRegression  
lr = LinearRegression()  
lr.fit(x_train,y_train)
```

```
LinearRegression()
```

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

```
Text(0, 0.5, 'Package(in lpa)')
```



```
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
y_pred = lr.predict(x_test)
y_test.values
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]])
print("MAE",mean_absolute_error(y_test,y_pred))
MAE 0.2884710931878175
print("MSE",mean_squared_error(y_test,y_pred))
MSE 0.12129235313495527
print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred)))
RMSE 0.34827051717731616
```

```

print("MSE",r2_score(y_test,y_pred))
r2 = r2_score(y_test,y_pred)

MSE 0.780730147510384

#Adjusted R2 score
x_test.shape

(40, 1)

1 - ((1-r2)*(40-1)/(40-1-1))

0.7749598882343415

new_df1 = df.copy()
new_df1['random_feature'] = np.random.random(200)
new_df1 = new_df1[['cgpa', 'random_feature', 'package']]
new_df1.head()

```

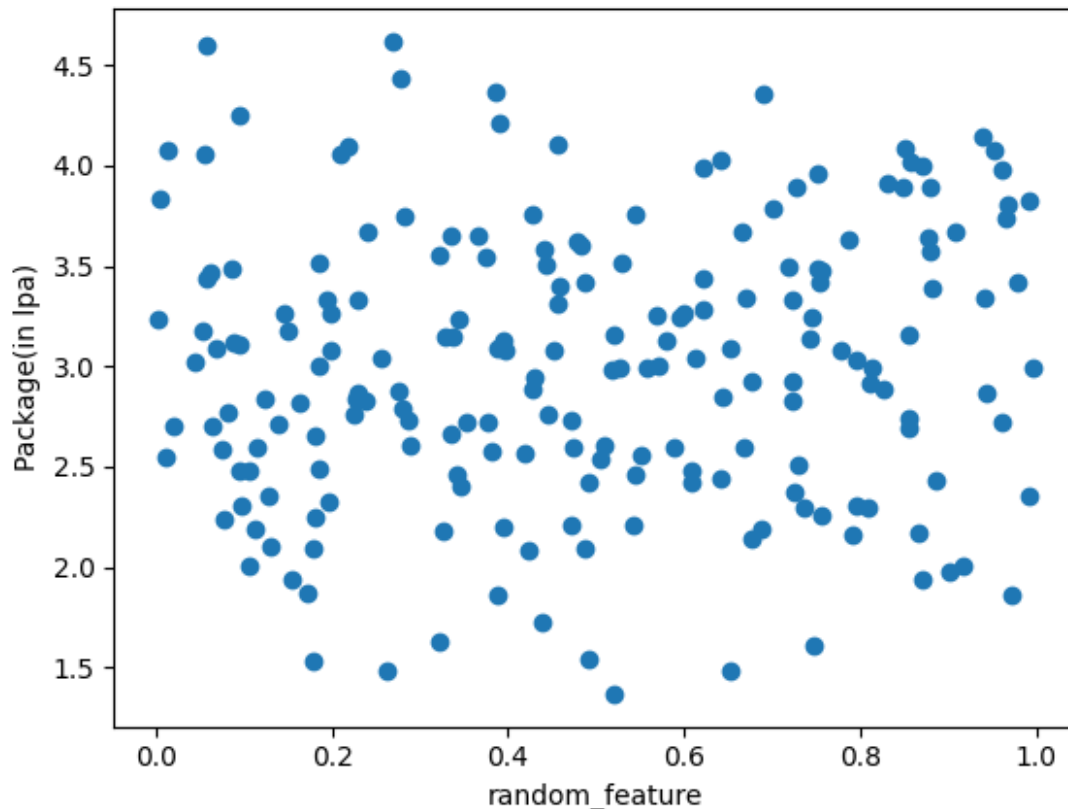
	cgpa	random_feature	package
0	6.89	0.198620	3.26
1	5.12	0.901528	1.98
2	7.82	0.568093	3.25
3	7.42	0.666366	3.67
4	6.94	0.879066	3.57

```

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

Text(0, 0.5, 'Package(in lpa)')

```



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

X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=2)

lr = LinearRegression()
lr.fit(X_train,y_train)
LinearRegression()

y_pred = lr.predict(X_test)

print("R2 score",r2_score(y_test,y_pred))
r2 = r2_score(y_test,y_pred)

R2 score 0.7834331907856025

1 - ((1-r2)*(40-1)/(40-1-2))

0.7717268767740134

new_df2 = df.copy()

new_df2['iq'] = new_df2['package'] + (np.random.randint(-
```

```

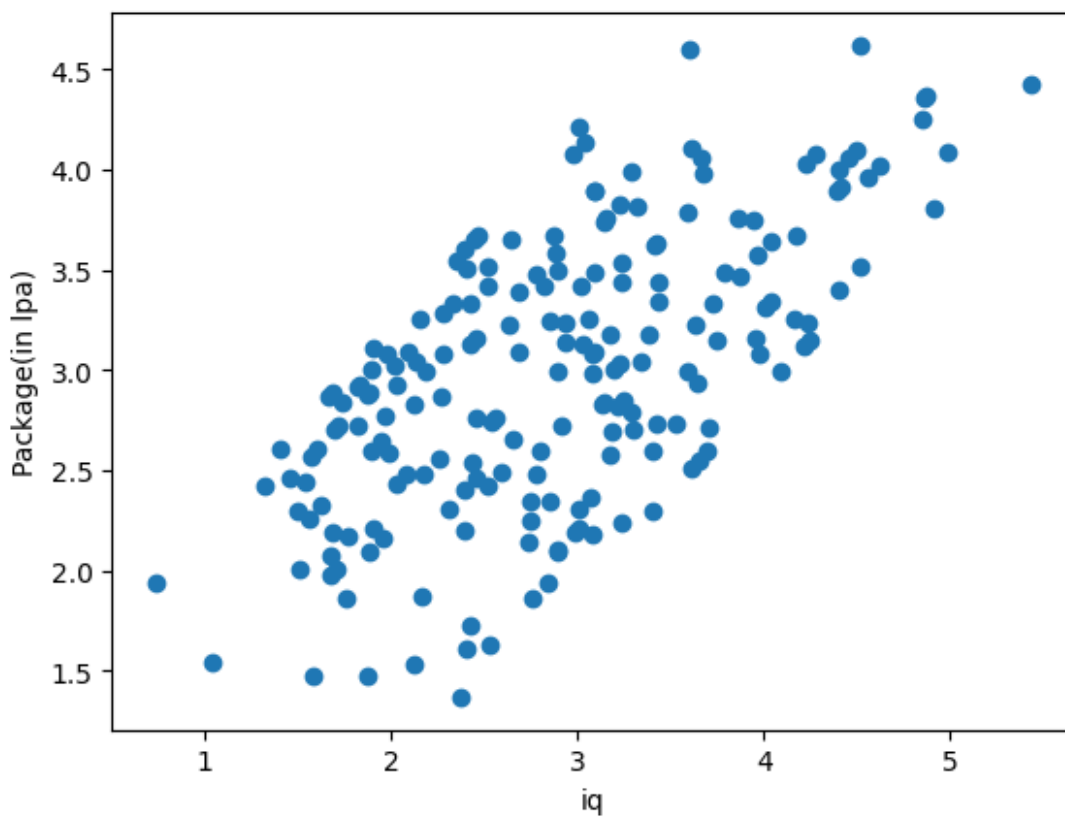
12,12,200)/10)

new_df2 = new_df2[['cgpa','iq','package']]
new_df2.sample(5)

   cgpa    iq  package
34  7.34  2.82    3.42
180  6.19  1.72    2.72
23   6.19  2.08    2.48
93   6.79  2.28    3.08
190  6.19  3.19    2.69

plt.scatter(new_df2['iq'],new_df2['package'])
plt.xlabel('iq')
plt.ylabel('Package(in lpa)')
Text(0, 0.5, 'Package(in lpa)')

```



```

np.random.randint(-100,100)

-58

X = new_df2.iloc[:,0:2]
y = new_df2.iloc[:, -1]

```

```
X_train,X_test,y_train,y_test =  
train_test_split(X,y,test_size=0.2,random_state=2)  
lr = LinearRegression()  
lr.fit(X_train,y_train)  
y_pred = lr.predict(X_test)  
  
print("R2 score",r2_score(y_test,y_pred))  
r2 = r2_score(y_test,y_pred)  
  
R2 score 0.8399396228404709  
  
1 - ((1-r2)*(40-1)/(40-1-2))  
  
0.8312877105615774
```