

A PYTHON PROGRAM TO IMPLEMENT SIMPLE LINEAR REGRESSION USING LEAST SQUARE METHOD

EXP NO. 2

Name: MOHAMED FUZAIL A

Roll no: 241801162

Code:

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
data = pd.read_csv("C:\\Users\\Shyam  
Ganesh\\Documents\\kaggle\\headbrain.csv")
```

```
x = np.array(list(data['Head Size(cm^3)']))
```

```
y = np.array(list(data['Brain Weight(grams)']))
```

```
print(x[:5], y[:5])
```

```
def get_line(x, y):  
    x_m, y_m = np.mean(x), np.mean(y)  
    print(x_m, y_m)  
    x_d, y_d = x - x_m, y - y_m  
    m = np.sum(x_d * y_d) / np.sum(x_d ** 2)  
    c = y_m - (m * x_m)  
    print(m, c)  
    return lambda x: m * x + c
```

```
lin = get_line(x, y)
```

```
X = np.linspace(np.min(x) - 100, np.max(x) + 100, 1000)
```

```
Y = np.array([lin(i) for i in X])
```

```
plt.plot(X, Y, color='red', label='Regression line')
```

```
plt.scatter(x, y, color='green', label='Scatter plot')
```

```
plt.xlabel('Head Size(cm^3)')
plt.ylabel('Brain Weight(grams)')
plt.legend()
plt.show()
```

```
def get_error(line_func, x, y):
    y_m = np.mean(y)
    y_pred = np.array([line_func(i) for i in x])
    ss_t = np.sum((y - y_m) ** 2)
    ss_r = np.sum((y - y_pred) ** 2)
    return 1 - (ss_r / ss_t)
```

```
print(get_error(lin, x, y))
```

```
from sklearn.linear_model import LinearRegression
```

```
x = x.reshape((len(x), 1))
```

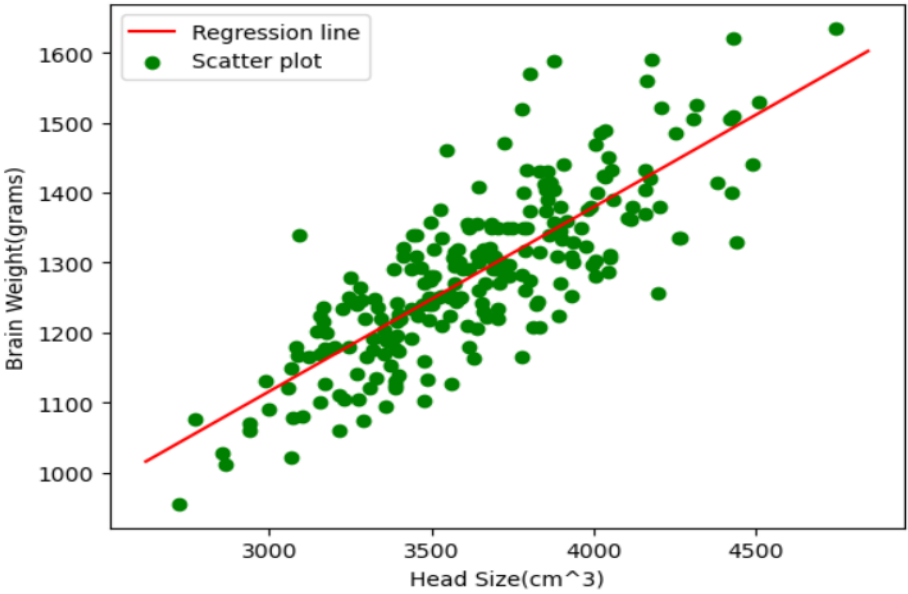
```
reg = LinearRegression()
```

```
reg = reg.fit(x, y)
```

```
print(reg.score(x, y))
```

OUTPUT:

[4512 3738 4261 3777 4177] [1530 1297 1335 1282 1590]
3633.9915611814345 1282.873417721519
0.2634293394893993 325.5734210494428



0.639311719957
0.639311719957