# EXPERIMENT -2

# A python program to implement Simple linear regression using Least Square Method

## AIM:

**A python program to implement Simple linear regression using Least Square Method**

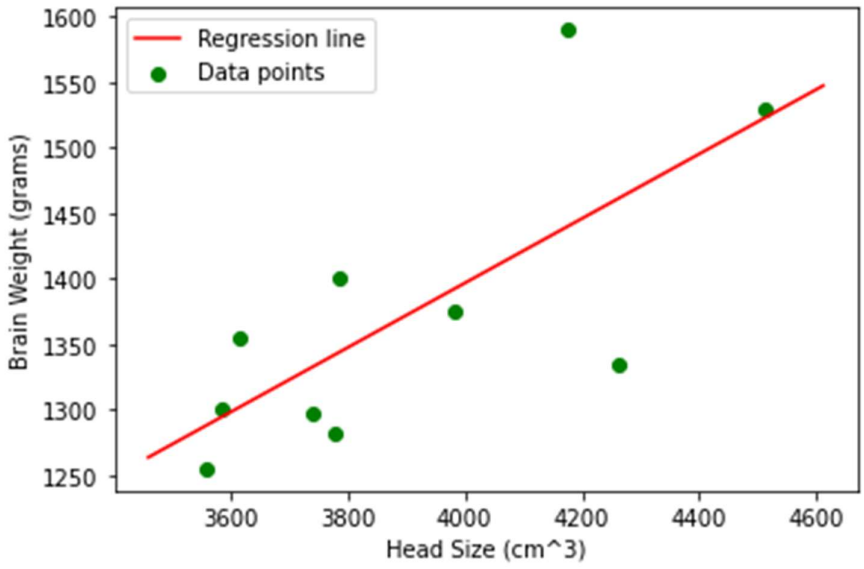
## CODE:

```
# Import required libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression

# Load the dataset
data = pd.read_csv('headbrain.csv')
x = np.array(list(data['Head Size(cm^3)']))
y = np.array(list(data['Brain Weight(grams)']))

# Display first few rows
print(x[:5], y[:5])

# Function to get regression line
```

```
def get_line(x, y):
    x_m, y_m = np.mean(x), np.mean(y)
    print("Mean of X:", x_m, " Mean of Y:", 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("Slope (m):", m, "Intercept (c):", c)
    return lambda x: m * x + c

# Generate regression line
lin = get_line(x, y)

# Plot
X = np.linspace(np.min(x) - 100, np.max(x) + 100, 1000)
Y = np.array([lin(val) for val in X])
plt.plot(X, Y, color='red', label='Regression line')
plt.scatter(x, y, color='green', label='Data points')
plt.xlabel('Head Size (cm^3)')
plt.ylabel('Brain Weight (grams)')
plt.legend()
plt.show()

# Calculate R^2 manually
def get_error(line_func, x, y):
    y_m = np.mean(y)
    y_pred = np.array([line_func(val) for val 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("Manual R2:", get_error(lin, x, y))

# Using sklearn
x = x.reshape((len(x), 1))
reg = LinearRegression()
reg.fit(x, y)
print("Sklearn R2:", reg.score(x, y))
```

## OUTPUT:




## RESULT:

**Thus a python program to implement Simple linear regression using Least Square Method is written and the output is verified successfully.**