**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 R² 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:**

**A graph with a red line and green dots

AI-generated content may be incorrect.**

**RESULT:**

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