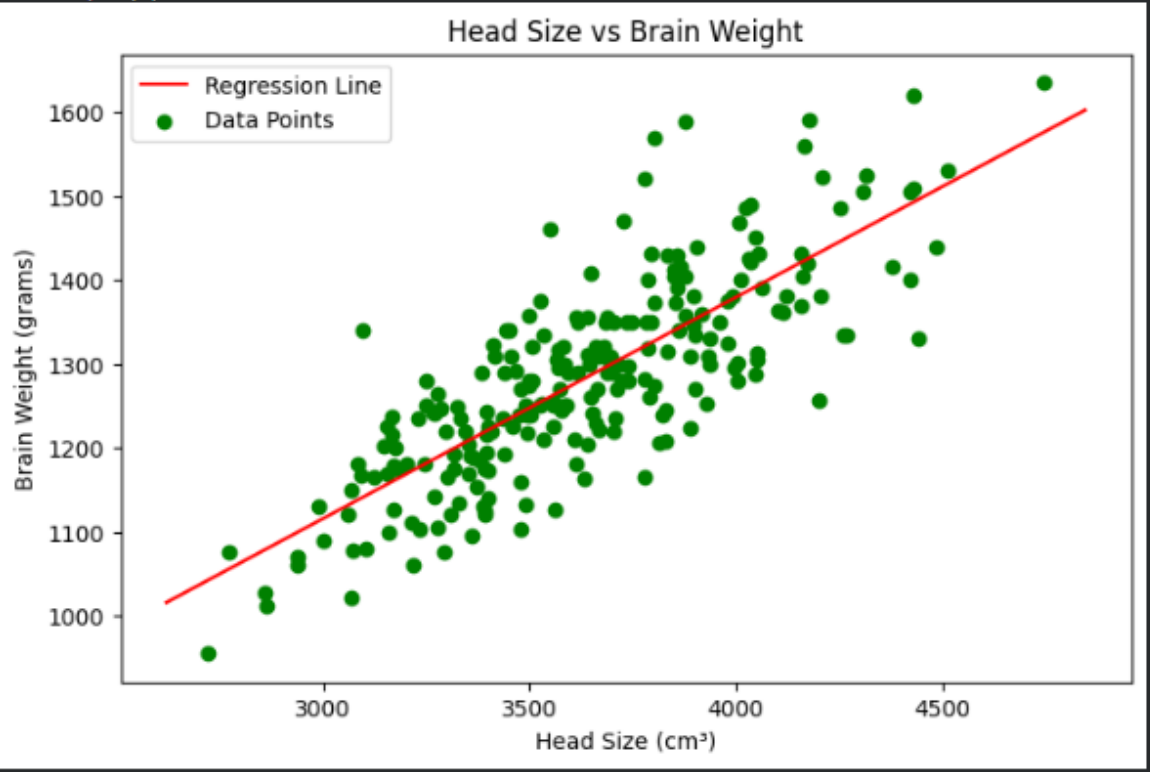
```
# ------------------------------------------------
# EXPERIMENT: LINEAR REGRESSION ON HEAD-BRAIN DATA
# ------------------------------------------------

# Step 1: Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Step 2: Upload the dataset file from your computer
from google.colab import files
uploaded = files.upload()  # 📤 Choose 'headbrain.csv' from your system

# Step 3: Load the dataset
data = pd.read_csv("headbrain.csv")

# Step 4: Display sample data
print(data.head())
print("Shape of data:", data.shape)

# Step 5: Separate features and target
x = np.array(data['Head Size(cm^3)'])
y = np.array(data['Brain Weight(grams)'])
print("\nSample values:")
print("X:", x[:5])
print("Y:", y[:5])

# ------------------------------------------------
# Step 6: Manual Implementation of Linear Regression
# ------------------------------------------------
```

```
def get_line(x, y):
    # Mean of X and Y
    x_m, y_m = np.mean(x), np.mean(y)
    print("\nMean of X:", x_m, " | Mean of Y:", y_m)

    # Calculate slope (m) and intercept (c)
    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)
    print("Intercept (c):", c)

    # Return line equation
    return lambda X: m * X + c

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

# Step 7: Plot regression line
X = np.linspace(np.min(x) - 100, np.max(x) + 100, 1000)
Y = np.array([lin(val) for val in X])

plt.figure(figsize=(8,5))
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.title('Head Size vs Brain Weight')
plt.legend()
plt.show()
```

```
# ------------------------------------------------
# Step 8: Calculate R² (Coefficient of Determination)
# ------------------------------------------------

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)     # Total Sum of Squares
    ss_r = np.sum((y - y_pred) ** 2)  # Residual Sum of Squares
    r2 = 1 - (ss_r / ss_t)
    return r2

r2_manual = get_error(lin, x, y)
print("\nManual R² Score:", r2_manual)

# ------------------------------------------------
# Step 9: Compare with sklearn LinearRegression
# ------------------------------------------------

from sklearn.linear_model import LinearRegression

x = x.reshape((len(x), 1))  # Reshape to 2D
reg = LinearRegression()
reg.fit(x, y)

print("\nSklearn R² Score:", reg.score(x, y))
print("Sklearn Slope (m):", reg.coef_[0])
print("Sklearn Intercept (c):", reg.intercept_)
```

```
Choose Files | headbrain.csv
headbrain.csv(text/csv) - 3370 bytes, last modified: 11/5/2025 - 100% done
Saving headbrain.csv to headbrain.csv
   Gender  Age Range  Head Size(cm^3)  Brain Weight(grams)
0       1          1             4512                 1530
1       1          1             3738                 1297
2       1          1             4261                 1335
3       1          1             3777                 1282
4       1          1             4177                 1590
Shape of data: (237, 4)

Sample values:
X: [4512 3738 4261 3777 4177]
Y: [1530 1297 1335 1282 1590]

Mean of X: 3633.9915611814345  | Mean of Y: 1282.873417721519
Slope (m): 0.2634293394893993
Intercept (c): 325.5734210494428
```




```
Manual R² Score: 0.639311719957

Sklearn R² Score: 0.639311719957
Sklearn Slope (m): 0.26342933948939934
Sklearn Intercept (c): 325.5734210494426
```