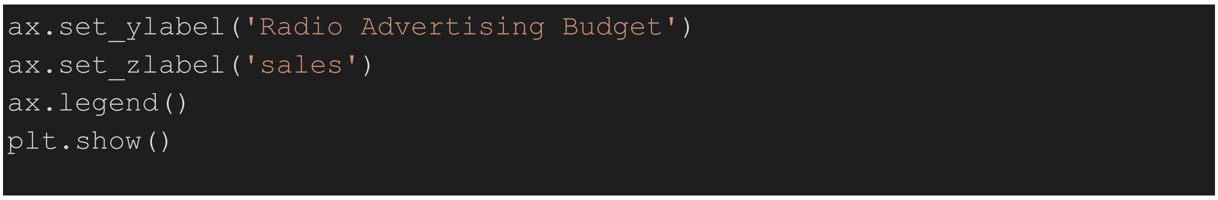
Roll. No.A176	Name: Yash Patil
Class MBA Tech	Batch: 1
Date of Experiment:	Date of Submission:
Grade:	

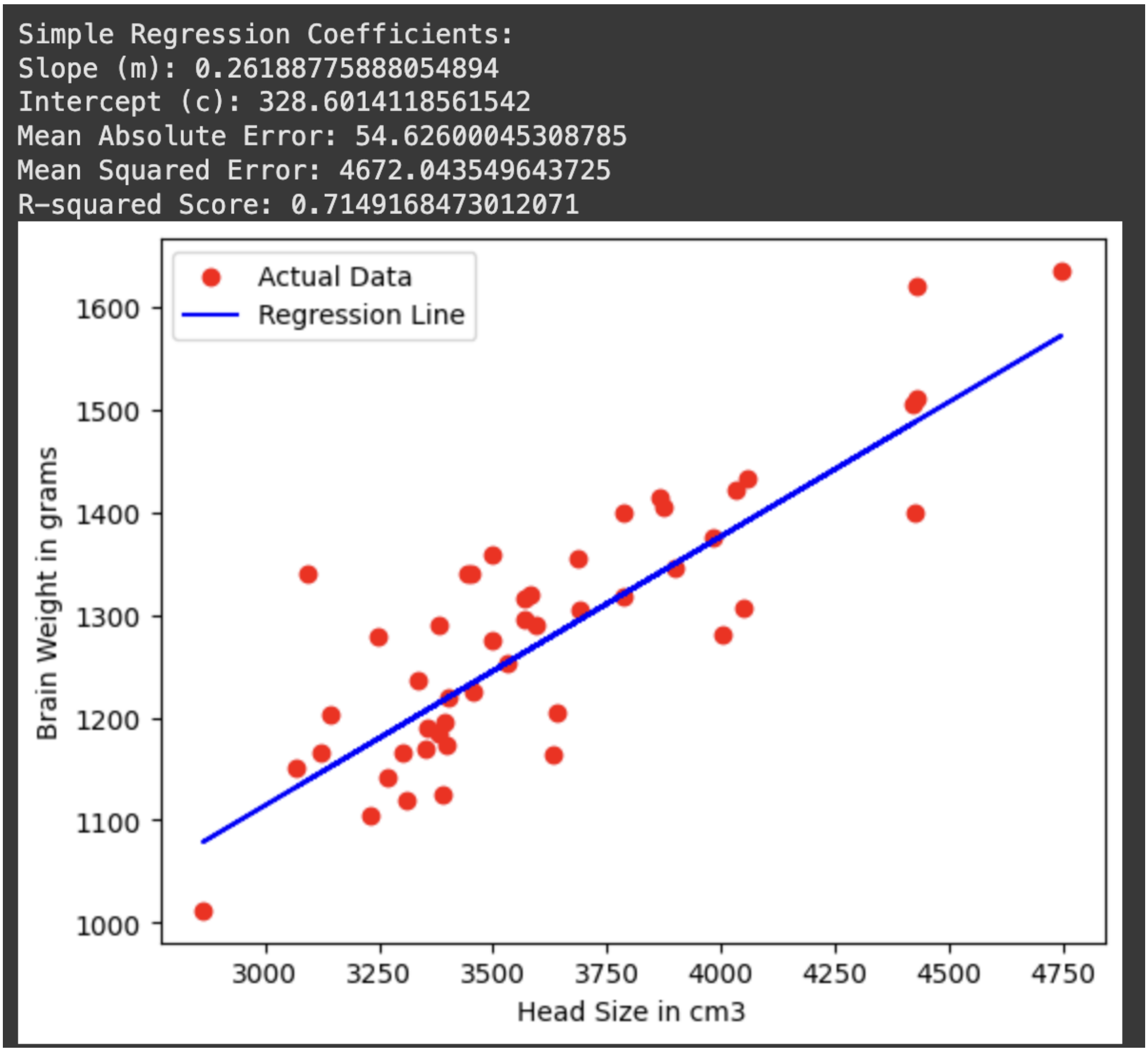
EXP 5

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

# ---------------- SIMPLE LINEAR REGRESSION ---------------- #
# Reading Data
data = pd.read_csv('headbrain.csv')  # Replace with your dataset file
X = data[['Head Size(cm^3)']].values
Y = data[['Brain Weight(grams)']].values

# Splitting Data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)

# Creating Model
model = LinearRegression()
model.fit(X_train, Y_train)

# Making Predictions
Y_pred = model.predict(X_test)

# Printing Model Coefficients
print("Simple Regression Coefficients:")
print("Slope (m):", model.coef_[0][0])
print("Intercept (c):", model.intercept_[0])

# Model Performance
mae = metrics.mean_absolute_error(Y_test, Y_pred)
mse = metrics.mean_squared_error(Y_test, Y_pred)
r2 = metrics.r2_score(Y_test, Y_pred)

print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("R-squared Score:", r2)

# Plotting
```

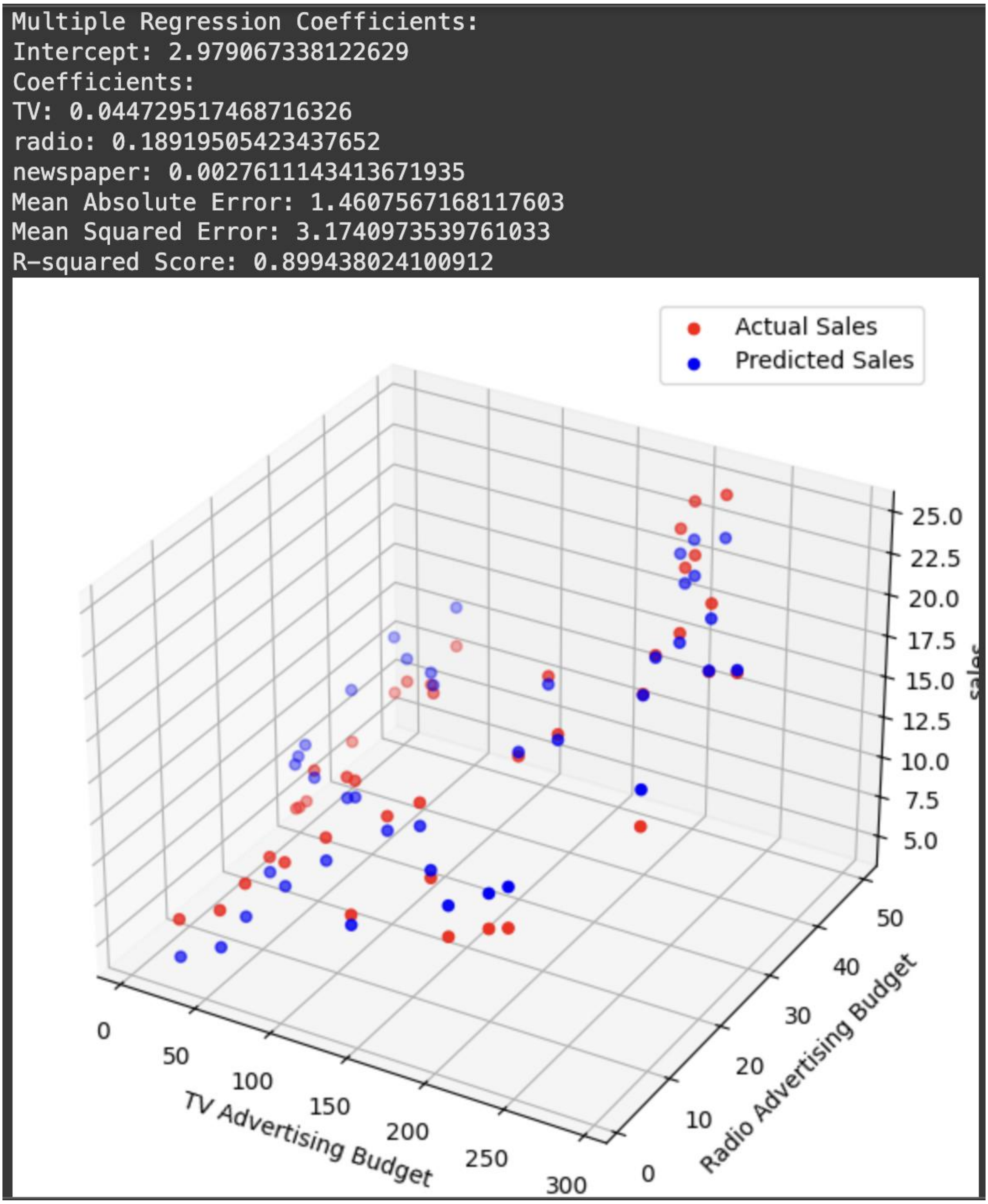
```
plt.scatter(X_test, Y_test, color='red', label='Actual Data')
plt.plot(X_test, Y_pred, color='blue', label='Regression Line')
plt.xlabel('Head Size in cm3')
plt.ylabel('Brain Weight in grams')
plt.legend()
plt.show()

# ---------------- MULTIPLE LINEAR REGRESSION ---------------- #
# Reading Data
data = pd.read_csv('Advertising.csv')  # Replace with your dataset file
X = data[['TV', 'radio', 'newspaper']]
y = data['sales']

# Splitting Data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Creating Model
mlr = LinearRegression()
mlr.fit(X_train, y_train)

# Making Predictions
y_pred_mlr = mlr.predict(X_test)

# Printing Model Coefficients
print("Multiple Regression Coefficients:")
print("Intercept:", mlr.intercept_)
print("Coefficients:")
for feature, coef in zip(X.columns, mlr.coef_):
    print(f"{feature}: {coef}")

# Model Performance
mae_mlr = metrics.mean_absolute_error(y_test, y_pred_mlr)
mse_mlr = metrics.mean_squared_error(y_test, y_pred_mlr)
r2_mlr = metrics.r2_score(y_test, y_pred_mlr)

print("Mean Absolute Error:", mae_mlr)
print("Mean Squared Error:", mse_mlr)
print("R-squared Score:", r2_mlr)

# Plotting for Multiple Regression (3D Plot)
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_test['TV'], X_test['radio'], y_test, color='red',
label='Actual Sales')
ax.scatter(X_test['TV'], X_test['radio'], y_pred_mlr, color='blue',
label='Predicted Sales')
ax.set_xlabel('TV Advertising Budget')
```



```
ax.set_ylabel('Radio Advertising Budget')
ax.set_zlabel('sales')
ax.legend()
plt.show()
```



```
Simple Regression Coefficients:
Slope (m): 0.26188775888054894
Intercept (c): 328.6014118561542
Mean Absolute Error: 54.62600045308785
Mean Squared Error: 4672.043549643725
R-squared Score: 0.7149168473012071
```



```
Multiple Regression Coefficients:
Intercept: 2.979067338122629
Coefficients:
TV: 0.044729517468716326
radio: 0.18919505423437652
newspaper: 0.0027611143413671935
Mean Absolute Error: 1.4607567168117603
Mean Squared Error: 3.1740973539761033
R-squared Score: 0.899438024100912
```