1. Simple Linear Regression - Headbrain Data

Code:

```
python
Copy code
import numpy as np
import pandas as pd

# Load the dataset
data = pd.read_csv("headbrain.csv")  # Replace with the correct file path
X = data['Head Size(cm^3)'].values
Y = data['Brain Weight(grams)'].values

# Calculate mean of X and Y
mean_x = np.mean(X)
mean_y = np.mean(Y)

# Calculate coefficients
n = len(X)
numerator = np.sum((X - mean_x) * (Y - mean_y))
denominator = np.sum((X - mean_x) ** 2)
m = numerator / denominator  # Slope
c = mean_y - m * mean_x  # Intercept

print(f"Coefficients: Slope = {m}, Intercept = {c}")

# Predict brain weight for a new head size
new_head_size = 4000  # Replace with the head size value
predicted_brain_weight = m * new_head_size + c
print(f"Predicted Brain Weight for head size {new_head_size}:
{predicted_brain_weight}")
```

---

2. Simple Linear Regression - Housing Price Data

Code:

```
python
Copy code
# Load the dataset
data = pd.read_csv("hprice.csv")  # Replace with the correct file path
X = data['total_sqft_int'].values
Y = data['price'].values

# Calculate mean of X and Y
mean_x = np.mean(X)
mean_y = np.mean(Y)

# Calculate coefficients
numerator = np.sum((X - mean_x) * (Y - mean_y))
denominator = np.sum((X - mean_x) ** 2)
m = numerator / denominator  # Slope
c = mean_y - m * mean_x  # Intercept

print(f"Coefficients: Slope = {m}, Intercept = {c}")

# Predictions for testing
predictions = m * X + c

# Calculate RMSE and R-squared
```

```
rmse = np.sqrt(np.sum((Y - predictions) ** 2) / n)
ss_total = np.sum((Y - mean_y) ** 2)
ss_residual = np.sum((Y - predictions) ** 2)
r_squared = 1 - (ss_residual / ss_total)

print(f"Root Mean Square Error: {rmse}")
print(f"R-squared value: {r_squared}")
```

---

3. Predict Prices for New Inputs

Code:

python
Copy code

```
# Predict for one new total_sqft_int value
new_sqft = 1200  # Replace with the value
predicted_price = m * new_sqft + c
print(f"Predicted Price for {new_sqft} sqft: {predicted_price}")

# Predict for three new total_sqft_int values
new_sqfts = [1500, 1800, 2200]  # Replace with values
predicted_prices = [m * sqft + c for sqft in new_sqfts]
print(f"Predicted Prices for {new_sqfts}: {predicted_prices}")
```

---

4. Implement Using Sklearn API (Train-Test Split and Plotting)

Code:

python
Copy code

```
import matplotlib.pyplot as plt

# Split data into training and testing sets
split_ratio = 0.75
split_index = int(split_ratio * len(X))
X_train, X_test = X[:split_index], X[split_index:]
Y_train, Y_test = Y[:split_index], Y[split_index:]

# Train using the train set
mean_x_train = np.mean(X_train)
mean_y_train = np.mean(Y_train)

numerator_train = np.sum((X_train - mean_x_train) * (Y_train - mean_y_train))
denominator_train = np.sum((X_train - mean_x_train) ** 2)
m_train = numerator_train / denominator_train
c_train = mean_y_train - m_train * mean_x_train

# Predict on the test set
predictions_test = m_train * X_test + c_train

# Plotting
plt.scatter(X, Y, color='blue', label='Actual Data')
plt.plot(X, m_train * X + c_train, color='red', label='Regression Line')
plt.title("Simple Linear Regression")
plt.xlabel("Total Square Feet")
plt.ylabel("Price")
plt.legend()
plt.show()
```