```
# Linear Regression: Simple Implementation

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_boston # You can replace with fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load dataset
boston = load_boston()  # You may use fetch_california_housing() due to deprecation
df = pd.DataFrame(boston.data, columns=boston.feature_names)
df['PRICE'] = boston.target

# Select features and target variable (Simple Linear Regression using 'RM')
X = df[['RM']]   # Average number of rooms per dwelling
y = df['PRICE']  # Target: House price

# Split data into training and testing sets (80/20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on test data
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Display results
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("R² Score:", r2)
print("Model Coefficient (slope):", model.coef_[0])
print("Model Intercept:", model.intercept_)

# Plot regression line
plt.scatter(X_test, y_test, color='blue', label='Actual')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted')
plt.title('Linear Regression: RM vs PRICE')
plt.xlabel('Average number of rooms per dwelling (RM)')
plt.ylabel('House Price')
plt.legend()
plt.show()
```