

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import r2_score


# Set a random seed for reproducibility

np.random.seed(42)


# Create the dataset

data = {

    'ID': range(1, 501),

    'TV': np.random.randint(1000, 5000, 500),

    'Radio': np.random.randint(100, 1000, 500),

    'Newspaper': np.random.randint(100, 1000, 500),

    'Sales': np.random.randint(10000, 50000, 500)

}


# Convert to DataFrame

df = pd.DataFrame(data)


# Display the first few rows of the dataset

print("Sample of the dataset:")

print(df.head())
```

```
# Define independent variable (feature) and target variable

X = df[['TV']] # Independent variable (only TV)

y = df['Sales'] # Target variable


# Split the data into training and testing sets (70:30 ratio)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)


# Print the shapes of the training and testing sets

print("\nTraining set shape:", X_train.shape, y_train.shape)

print("Testing set shape:", X_test.shape, y_test.shape)


# Build a simple linear regression model

model = LinearRegression()

model.fit(X_train, y_train)


# Print the model coefficients

print("\nModel coefficients:")

print("Intercept:", model.intercept_)

print("Coefficient (TV):", model.coef_[0])


# Predict sales on the testing set

y_pred = model.predict(X_test)


# Evaluate the model using R^2 score

r2 = r2_score(y_test, y_pred)

print("\nModel R^2 score on testing set:", r2)
```

```
# Plotting the regression line

plt.figure(figsize=(8, 6))

sns.scatterplot(x=X_test['TV'], y=y_test, color='blue', label='Actual Sales')

sns.lineplot(x=X_test['TV'], y=y_pred, color='red', label='Predicted Sales')

plt.title('Simple Linear Regression: TV vs Sales')

plt.xlabel('TV Advertising Budget')

plt.ylabel('Sales')

plt.legend()

plt.show()
```