Advertising Dataset

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```python
df = pd.read_csv('/content/advertising.csv')
```

```python
print("Dataset Head:")
display(df.head())
```

Dataset Head:

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

```python
X = df[['TV']]
y = df['Sales']
```

```python
# Check for missing values
print("Missing values before cleaning:")
display(df.isnull().sum())
```

Missing values before cleaning:

	0
TV	0
Radio	0
Newspaper	0
Sales	0

dtype: int64

```python
df_cleaned = df.dropna()
print("\nMissing values after dropping rows:")
display(df_cleaned.isnull().sum())
print("\nFirst 5 rows of the cleaned DataFrame:")
display(df_cleaned.head())
```

```
Missing values after dropping rows:
                0
    TV          0
   Radio        0
Newspaper       0
   Sales        0
```

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

First 5 rows of the cleaned DataFrame:

```python
print("Training data shape:", X_train.shape, y_train.shape)
print("Testing data shape:", X_test.shape, y_test.shape)
```

```
Training data shape: (160, 1) (160,)
Testing data shape: (40, 1) (40,)
  2   17.2   45.9      69.3    12.0
```

```python
model = LinearRegression()
```

```
  4  180.8   10.8      58.4    17.9
```

```python
model.fit(X_train, y_train)
```

```
▾ LinearRegression  ⓘ ⓘ
LinearRegression()
```

```python
y_pred = model.predict(X_test)
```

```python
print("Model training complete. Coefficients and Intercept:")
print(f"Coefficient: {model.coef_[0]:.2f}")
print(f"Intercept: {model.intercept_:.2f}")
```
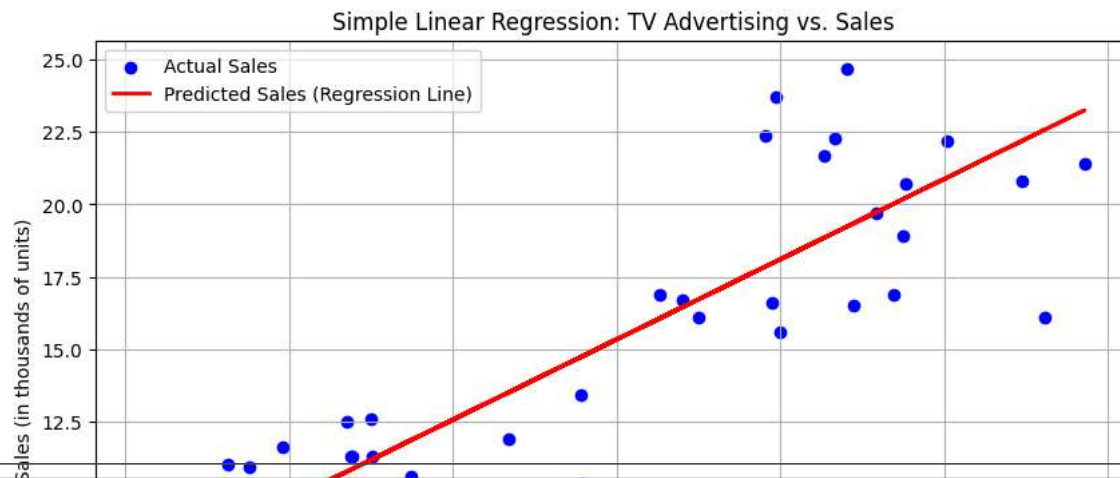
```
Model training complete. Coefficients and Intercept:
Coefficient: 0.06
Intercept: 7.01
```

```python
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```python
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"R-squared (R2) Score: {r2:.2f}")
```

```
Mean Squared Error (MSE): 6.10
R-squared (R2) Score: 0.80
```

```python
plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color='blue', label='Actual Sales')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted Sales (Regression Line)')
plt.xlabel('TV Advertising (in thousands of dollars)')
plt.ylabel('Sales (in thousands of units)')
plt.title('Simple Linear Regression: TV Advertising vs. Sales')
plt.legend()
plt.grid(True)
plt.show()
```

Simple Linear Regression: TV Advertising vs. Sales

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='green', label='Actual vs. Predicted Sales')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', label='Ideal Fit (y=x)'
plt.xlabel('Actual Sales (in thousands of units)')
plt.ylabel('Predicted Sales (in thousands of units)')
plt.title('Actual vs. Predicted Sales with Regression Line')
plt.legend()
plt.grid(True)
plt.show()
```


Actual vs. Predicted Sales with Regression Line