IMPORT LIBRARIES

In [1]:
```
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 mean_absolute_error,mean_squared_error,r2_score
```

LOAD AND VIEW THE DATASET

In [2]:
```
df=pd.read_csv('Advertising.csv')
df.head()
```

Out[2]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

DATA PREPROCESSING

In [3]:
```
print(df.isnull().sum())
```

```
Unnamed: 0    0
TV            0
Radio         0
Newspaper     0
Sales         0
dtype: int64
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  200 non-null    int64  
 1   TV          200 non-null    float64
 2   Radio       200 non-null    float64
 3   Newspaper   200 non-null    float64
 4   Sales       200 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

In [5]:
```
df.describe()
```

Out[5]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

SPLIT THE DATA INTO TRAIN-TEST SETS

In [6]:
```
X=df[['TV', 'Radio', 'Newspaper']]
y=df['Sales']
X_train, X_test, y_train, y_test= train_test_split(X,y,test_size=0.2, random_sta
print(f"Training data shape: {X_train.shape}")
print(f"Testing data shape: {X_test.shape}")
```

```
Training data shape: (160, 3)
Testing data shape: (40, 3)
```

TRAIN THE LINEAR REGRESSION MODEL

In [7]:
```
#create the model
lr=LinearRegression()
# Train the model
lr.fit(X_train,y_train)
# Predict the test set
y_pred=lr.predict(X_test)
```

EVALUATE THE MODEL

In [8]:
```
mae = mean_absolute_error(y_test,y_pred)
mse = mean_squared_error(y_test,y_pred)
r2 = r2_score(y_test,y_pred)
print(f"Mean Absolute Error (MAE): {mae: 2f}")
print(f"Mean Squared Error (MSE): {mse: 2f}")
print(f"R-Squared (R² Score): {mae: 2f}")
```

```
Mean Absolute Error (MAE):  1.460757
Mean Squared Error (MSE):  3.174097
R-Squared (R² Score):  1.460757
```

VISUALIZE THE RESULTS

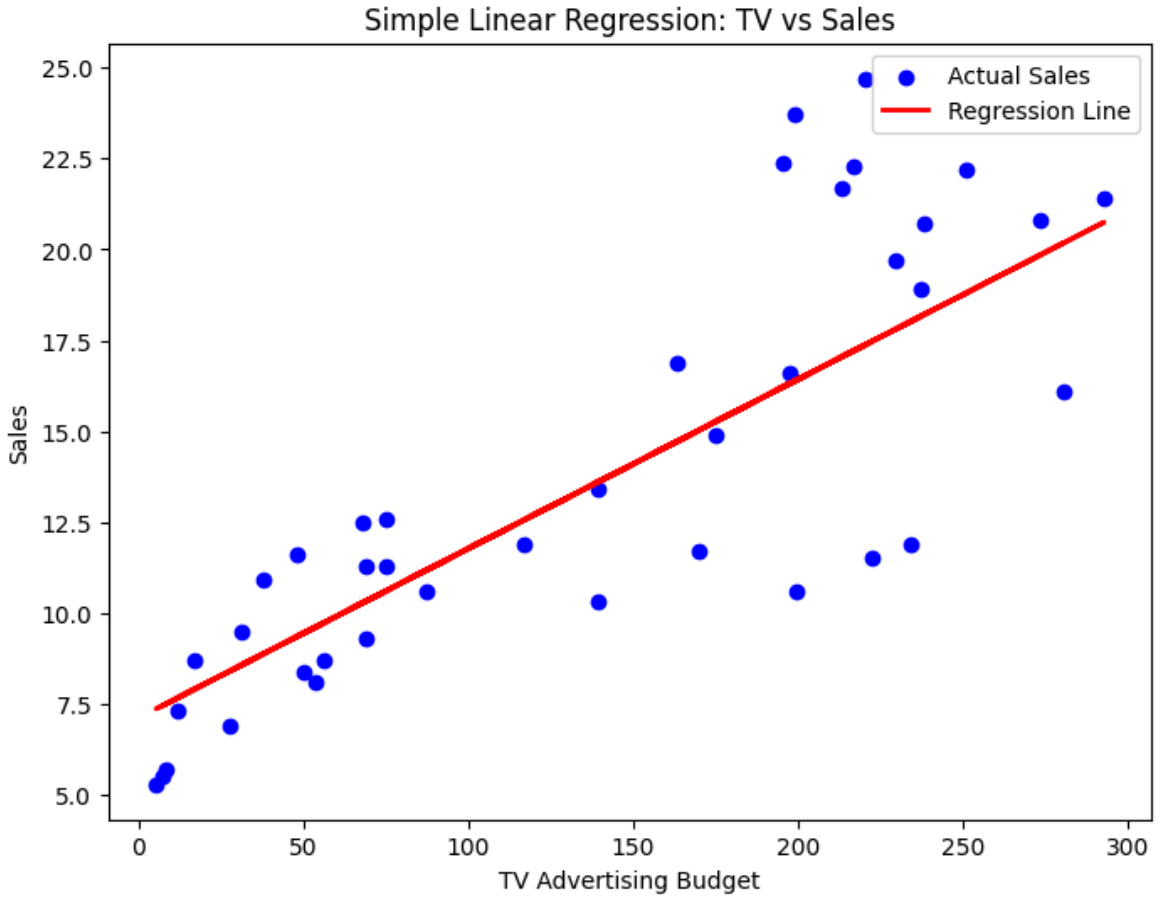
In [9]:
```
X_tv = df[['TV']]
y_sales = df['Sales']

X_train_tv, X_test_tv, y_train_tv, y_test_tv = train_test_split(X_tv, y_sales, t

lr_tv = LinearRegression()
lr_tv.fit(X_train_tv, y_train_tv)
```

```
y_pred_tv = lr_tv.predict(X_test_tv)
```

In [10]:
```
plt.figure(figsize=(8,6))
plt.scatter(X_test_tv, y_test_tv, color='blue', label='Actual Sales')
plt.plot(X_test_tv, y_pred_tv, color='red', linewidth=2, label='Regression Line'
plt.xlabel('TV Advertising Budget')
plt.ylabel('Sales')
plt.title('Simple Linear Regression: TV vs Sales')
plt.legend()
plt.show()
```



INTERPRET THE COEFFICIENTS

In [11]:
```
print("Intercept:", lr.intercept_)
print("Coefficient:", lr.coef_)

coef_df = pd.DataFrame({
    'Feature': ['TV','Radio','Newspaper'],
    'Coefficient': lr.coef_
})

coef_df
```

```
Intercept: 2.9790673381226256
Coefficient: [0.04472952 0.18919505 0.00276111]
```

Out[11]:

	Feature	Coefficient
0	TV	0.044730
1	Radio	0.189195
2	Newspaper	0.002761

In [ ]: