In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import matplotlib.pyplot as plt
```

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

In [3]:
```
df.head()
```

Out[3]:

	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

In [5]:
```
#Define features and target
X = df[['TV', 'Radio', 'Newspaper']]   # Independent variables
y = df['Sales']                        # Dependent variable
```

In [6]:
```
#4: Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)
```

In [7]:
```
#5: Train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[7]:

LinearRegression

▶ Parameters

In [8]:
```
print("Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
```

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

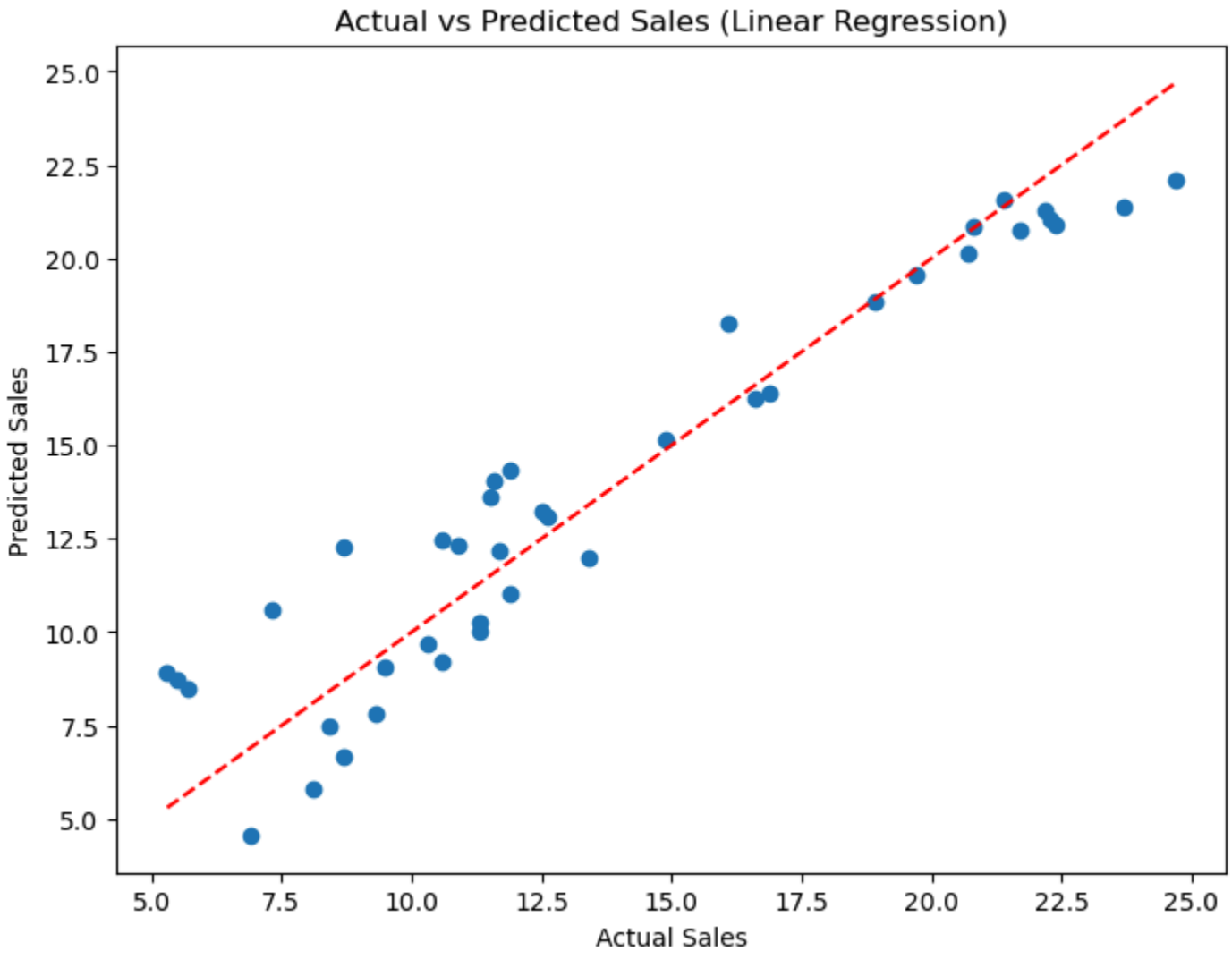
In [9]:
```
y_pred = model.predict(X_test)
```

In [10]:
```
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("R-Squared:", r2)
```

```
Mean Squared Error (MSE): 3.174097353976104
Mean Absolute Error (MAE): 1.4607567168117606
R-Squared: 0.899438024100912
```

In [11]:
```
plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Sales")
plt.ylabel("Predicted Sales")
plt.title("Actual vs Predicted Sales (Linear Regression)")
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.show()
```


Actual vs Predicted Sales (Linear Regression)
Predicted Sales
Actual Sales
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0

In [ ]: