In [1]:
```python
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
```

In [2]:
```python
# Input vector X and output vector y
X = np.array([[1, 1], [1, 2], [2, 2], [2, 3], [2, 4], [3, 4], [3, 5]])
y = np.array([4, 6, 7, 8, 9, 10, 12])
```

In [3]:
```python
# Split the 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 [4]:
```python
# Initialize and fit the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[4]:
```
▼ LinearRegression
LinearRegression()
```

In [5]:
```python
# Make predictions on the testing set
y_pred = model.predict(X_test)
```

In [6]:
```python
# Calculate mean squared error
mse = mean_squared_error(y_test, y_pred)
```

In [7]:
```python
# Calculate R-squared
r_squared = r2_score(y_test, y_pred)
```

In [8]:
```python
# Calculate adjusted R-squared
n = len(y_test)
p = X_test.shape[1]
adjusted_r_squared = 1 - (1 - r_squared) * ((n - 1) / (n - p - 1))
```

In [9]:
```python
# Print the results
print("Coefficients (beta):", model.coef_)
print("Intercept (beta_0):", model.intercept_)
print("Mean squared error (MSE):", mse)
print("R-squared:", r_squared)
print("Adjusted R-squared:", adjusted_r_squared)
```

```
Coefficients (beta): [1.2 1.2]
Intercept (beta_0): 1.9999999999999982
Mean squared error (MSE): 0.15999999999999992
R-squared: 0.8400000000000001
Adjusted R-squared: 1.16
```

In [ ]: