LINEAR REGRESSION

In [2]:
```
import warnings
import sys
if not sys.warnoptions:
    warnings.simplefilter("ignore")
```

In [10]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

In [24]:
```
# Load the dataset
data=pd.read_excel("LR_Sample1.xlsx")
data
```

Out[24]:

	X	Y
0	1	2
1	2	4
2	3	6
3	4	8
4	5	10
5	6	12
6	7	14
7	8	16
8	9	18
9	10	20
10	11	22
11	12	24
12	13	26
13	14	28
14	15	30
15	16	32
16	17	34
17	18	36
18	19	38
19	20	40

In [18]:
```
from sklearn.linear_model import LinearRegression
```

In [ ]:
```
# If mean square error is approx. equal to zero, the model performance is good.
```

In [20]:
```
from sklearn.metrics import mean_squared_error, r2_score
```

In [26]:
```
X = data[['X']].values # X variable should 2D dataframe
Y = data['Y'].values
```

In [28]:
```
model = LinearRegression()
```

In [32]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

In [34]:
```
X_train
```

Out[34]:
```
array([[ 9],
       [ 6],
       [12],
       [ 4],
       [19],
       [17],
       [14],
       [ 3],
       [10],
       [20],
       [ 5],
       [13],
       [ 8],
       [11],
       [15],
       [ 7]], dtype=int64)
```

In [36]:
```
X_test
```

Out[36]:
```
array([[ 1],
       [18],
       [16],
       [ 2]], dtype=int64)
```

In [38]:
```
y_train
```

Out[38]:
```
array([18, 12, 24,  8, 38, 34, 28,  6, 20, 40, 10, 26, 16, 22, 30, 14],
      dtype=int64)
```

In [40]:
```
y_test
```

Out[40]:
```
array([ 2, 36, 32,  4], dtype=int64)
```

In [42]:
```
model.fit(X_train,y_train)
```

Out[42]:
```
LinearRegression
LinearRegression()
```

In [46]:
```
Y_predict = model.predict(X_test)
```

In [48]:
```
Y_predict
```

Out[48]:
```
array([ 2., 36., 32.,  4.])
```

In [52]:
```
y_test
```

Out[52]:
```
array([ 2, 36, 32,  4], dtype=int64)
```

Model Performance

Mean Square Error

In [71]:
```
print("Mean Squared Error (MSE):", mean_squared_error(y_test, Y_predict))
```

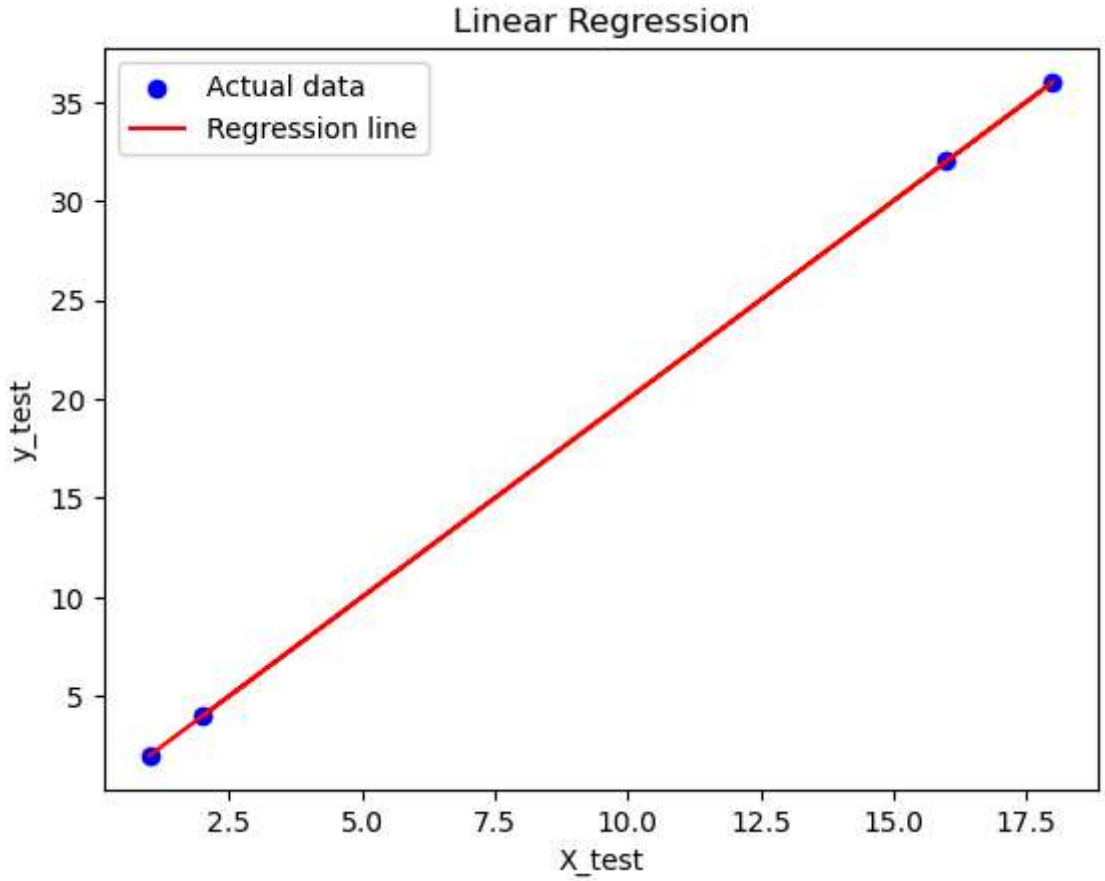
```
Mean Squared Error (MSE): 3.2540512340366737e-29
```

R-Square Error

In [59]:
```
print("R-Squared Error (RSE):", r2_score(y_test, Y_predict)) # If RSE = 1, the model is perfect (highly unlikely)
```

```
R-Squared Error (RSE): 1.0
```

In [73]:
```
plt.scatter(X_test, y_test, color='blue', label='Actual data')
plt.plot(X_test, Y_predict, color='red', label='Regression line')
plt.title('Linear Regression')
plt.xlabel('X_test')
plt.ylabel('y_test')
plt.legend()
plt.show()
```



Mean Square Error is also referred as Cost Function.

The best learning rate (alpha) should be 0.001 to ensure best convergence.