In [63]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

X=np.array([6,7,8,9,10,11,4])
Y=np.array([4,5,6,7,8,9,10])

X_mean= np.mean(X)
Y_mean= np.mean(Y)


a= sum((X-X_mean)*(Y-Y_mean))
b=sum((X-X_mean)**2)


# now calculate the value of B1 slope...

B1= a/b

print("The slope of the linear regession line is:", B1)

B0= Y_mean-B1*X_mean

print("The intercept of the linear regession line is:" ,B0)


y_pred= B0+B1*X
print("The predicted value is:")
print(y_pred)

plt.scatter(X,Y, color="blue", label=" Data points")
plt.plot(X,y_pred, color="red", label="Linear regession line")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Linear regession")
plt.legend()
plt.show()

```

In [ ]:

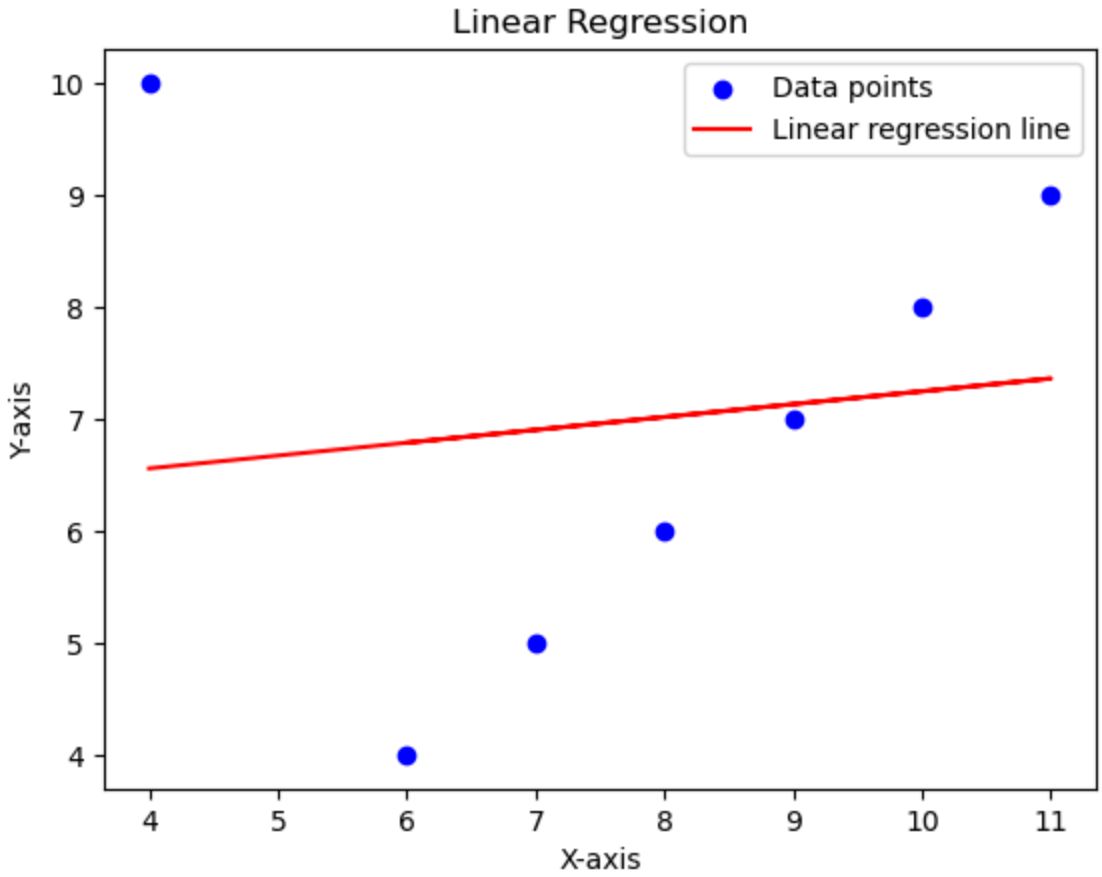
```
1
```

same code using function

In [65]:

```
import numpy as np
import matplotlib.pyplot as plt


X = np.array([6, 7, 8, 9, 10, 11, 4])
Y = np.array([4, 5, 6, 7, 8, 9, 10])


def linear_regression(X, Y):
    X_mean = np.mean(X)
    Y_mean = np.mean(Y)

    a = sum((X - X_mean) * (Y - Y_mean))
    b = sum((X - X_mean) ** 2)

    B1 = a / b
    print("The slope of the linear regression line is:",B1)

    B0 = Y_mean - B1 * X_mean
    print("The intercept of the linear regression line is:",B0)

    y_pred = B0 + B1 * X
    print("The predicted values are:",y_pred)


    # Plot the results
    plt.scatter(X, Y, color="blue", label="Data points")
    plt.plot(X, y_pred, color="red", label="Linear regression line")
    plt.xlabel("X-axis")
    plt.ylabel("Y-axis")
    plt.title("Linear Regression")
    plt.legend()
    plt.show()


linear_regression(X,Y)
```

```
The slope of the linear regression line is: 0.11475409836065578
The intercept of the linear regression line is: 6.098360655737705
The predicted values are: [6.78688525 6.90163934 7.01639344 7.13114754 7.245
90164 7.36065574
 6.55737705]
```


Linear Regression
Data points
Linear regression line
Y-axis
X-axis
10
9
8
7
6
5
4
4
5
6
7
8
9
10
11

linear regression using "iris dataset"

In [66]:
```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5
6 data= pd.read_csv("iris")
7 data
```

Out[66]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [67]:
```
1 data.shape
```

Out[67]:
```
(150, 5)
```

In [68]:
```
1 data.count()
```

Out[68]:
```
sepal_length    150
sepal_width     150
petal_length    150
petal_width     150
species         150
dtype: int64
```

In [69]:
```
1 print(data.isnull().sum())
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [70]:
```
1 print(data.duplicated().sum())
```

```
1
```

In [71]:
```
1 data=data.drop_duplicates()
```

In [72]:
```
1 data
```

Out[72]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

149 rows × 5 columns

In [73]:
```
1 data.describe()
```

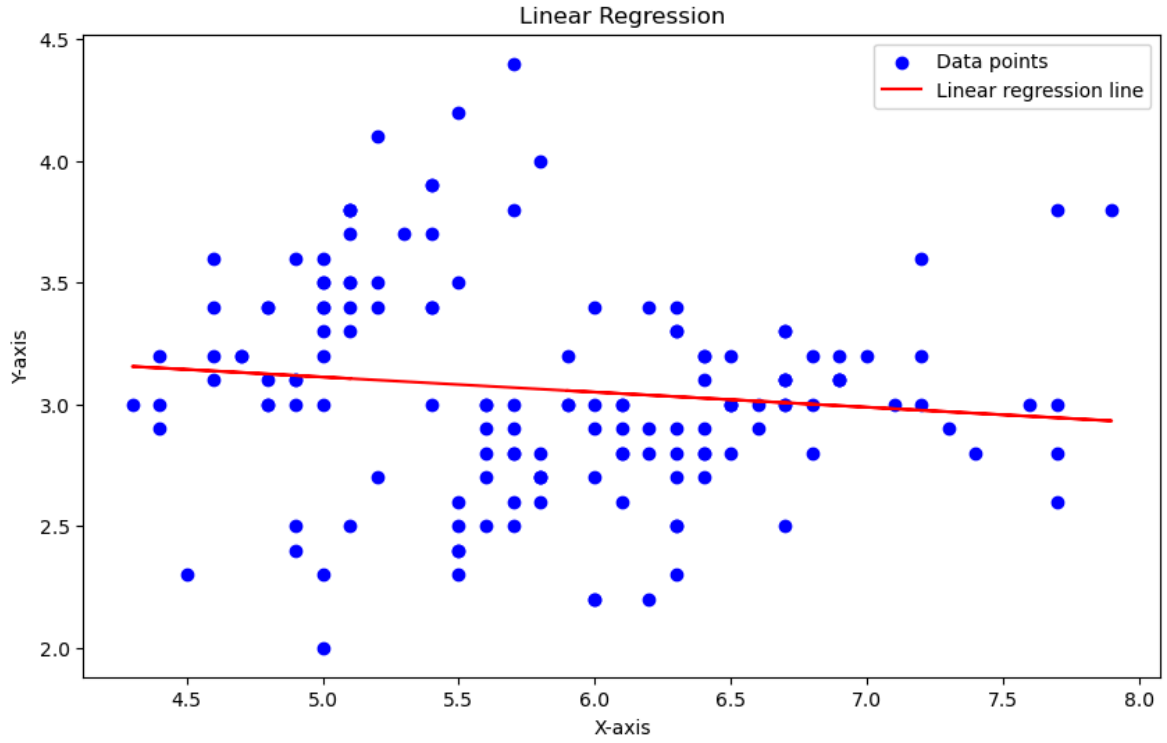
Out[73]:

	sepal_length	sepal_width	petal_length	petal_width
count	149.000000	149.000000	149.000000	149.000000
mean	5.843624	3.059732	3.748993	1.194631
std	0.830851	0.436342	1.767791	0.762622
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.300000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [74]:

```
# just use sepal_lenght and sepal_widht as a data

X=data["sepal_length"]
Y=data["sepal_width"]



def linear_regression(X, Y):
    X_mean = np.mean(X)
    Y_mean = np.mean(Y)

    a = sum((X - X_mean) * (Y - Y_mean))
    b = sum((X - X_mean) ** 2)

    B1 = a / b
    print("The slope of the linear regression line is:",B1)

    B0 = Y_mean - B1 * X_mean
    print("The intercept of the linear regression line is:",B0)

    y_pred = B0 + B1 * X
    print("The predicted values are:",y_pred)


    # Plot the results
    plt.figure(figsize=(10,6))
    plt.scatter(X, Y, color="blue", label="Data points")
    plt.plot(X,y_pred, color="red", label="Linear regression line")
    plt.xlabel("X-axis")
    plt.ylabel("Y-axis")
    plt.title("Linear Regression")
    plt.legend()
    plt.show()


linear_regression(X,Y)
```

```
The slope of the linear regression line is: -0.06203852116561995
The intercept of the linear regression line is: 3.4222613448248675
The predicted values are: 0      3.105865
1      3.118273
2      3.130680
3      3.136884
4      3.112069
         ...
145    3.006603
146    3.031419
147    3.019011
148    3.037623
149    3.056234
Name: sepal_length, Length: 149, dtype: float64
```


Linear Regression
Data points
Linear regression line
Y-axis
X-axis
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

In [ ]:
```
1
```