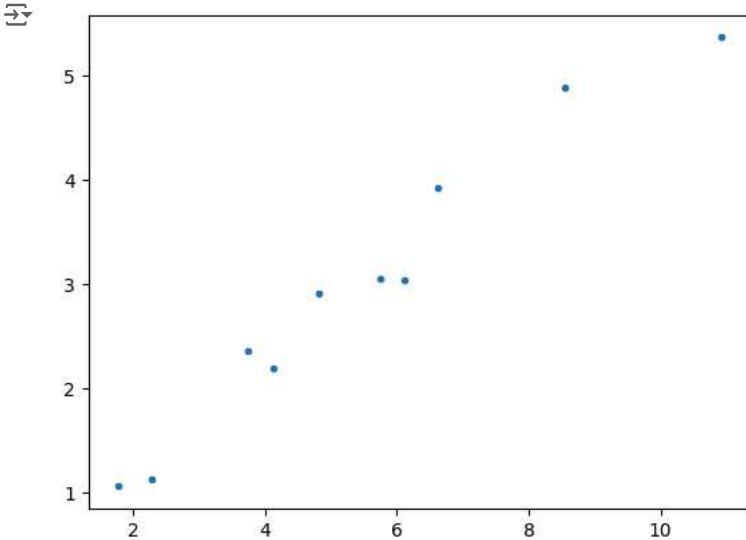


▼ create dataset

- $x = [1.78, 6.12, 6.63, 4.13, 8.55, 10.92, 4.81, 3.75, 2.28, 5.74]$
- $y = [1.06, 3.04, 3.93, 2.19, 4.89, 5.37, 2.91, 2.36, 1.13, 3.05]$



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.array([1.78, 6.12, 6.63, 4.13, 8.55, 10.92, 4.81, 3.75, 2.28, 5.74])
5 y = np.array([1.06, 3.04, 3.93, 2.19, 4.89, 5.37, 2.91, 2.36, 1.13, 3.05])
6
7 plt.plot(x, y, '.')
```



▼ linear regression using simple code

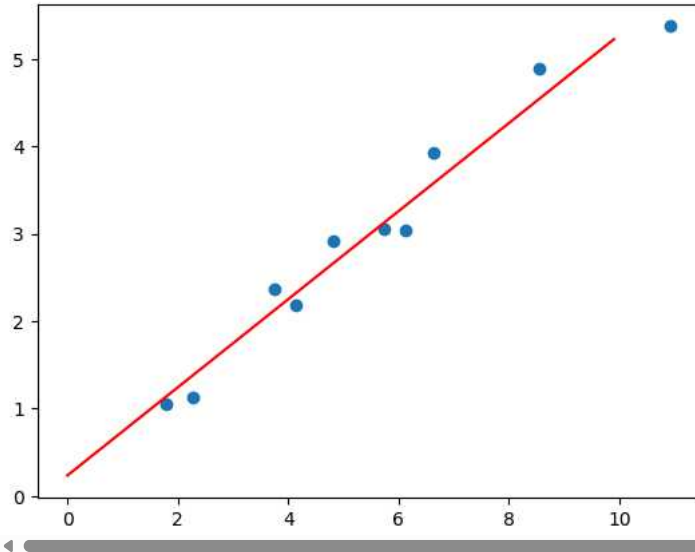
```
1 def ols_loss(params, x, y):
2     w, b = params
3     y_hat = w * x + b
4     mse = np.mean((y_hat - y) ** 2)
5
6     return mse
```

1 코딩을 시작하거나 AI로 코드를 생성하세요.

```
1 def compute_gradient(params, x, y):
2     w, b = params
3     n = len(x)
4     grad_w = (2/n) * np.sum((w*x + b - y) * x)
5     grad_b = (2/n) * np.sum(w*x + b - y)
6     return np.array([grad_w, grad_b])
7
8 def minimize(x, y, learning_rate = 0.01, num_iterations = 1000):
9     w, b = 0, 0
10    for _ in range(num_iterations):
11        gradient = compute_gradient([w, b], x, y)
12        w = w - learning_rate * gradient[0]
13        b = b - learning_rate * gradient[1]
14    return w, b
15
```

```
1 w_opt, b_opt = minimize(x, y)
2 print(w_opt, b_opt)
3
4 x_test = np.arange(0, 10, 0.1)
5 y_hat = w_opt * x_test + b_opt
6
7 plt.scatter(x, y)
8 plt.plot(x_test, y_hat, c='red')
9 plt.show()
```

↔ 0.5037244251723141 0.23639361345270662



✓ linear regression using scikit_learn

```
1
2 def ols_loss(params, x, y):
3     w, b = params
4     y_hat = w * x + b
5     mse = np.mean((y_hat - y) ** 2)
6
7     return mse
8

1 from scipy import optimize
2 result = optimize.minimize(ols_loss, [1.0, 1.0], args=(x,y))
3
4 print(result)
5 w_opt, b_opt = result.x
```

↔ message: Optimization terminated successfully.
success: True
status: 0
fun: 0.06684729140805669
x: [5.031e-01 2.403e-01]
nit: 6
jac: [8.382e-09 -9.313e-10]
hess_inv: [[7.141e-02 -3.907e-01]
[-3.907e-01 2.637e+00]]
nfev: 21
njev: 7

```
1 x_test = np.arange(0, 10, 0.1)
2 y_hat = w_opt * x_test + b_opt
3
4 plt.scatter(x, y)
5 plt.plot(x_test, y_hat, c='red')
6 plt.show()
```

