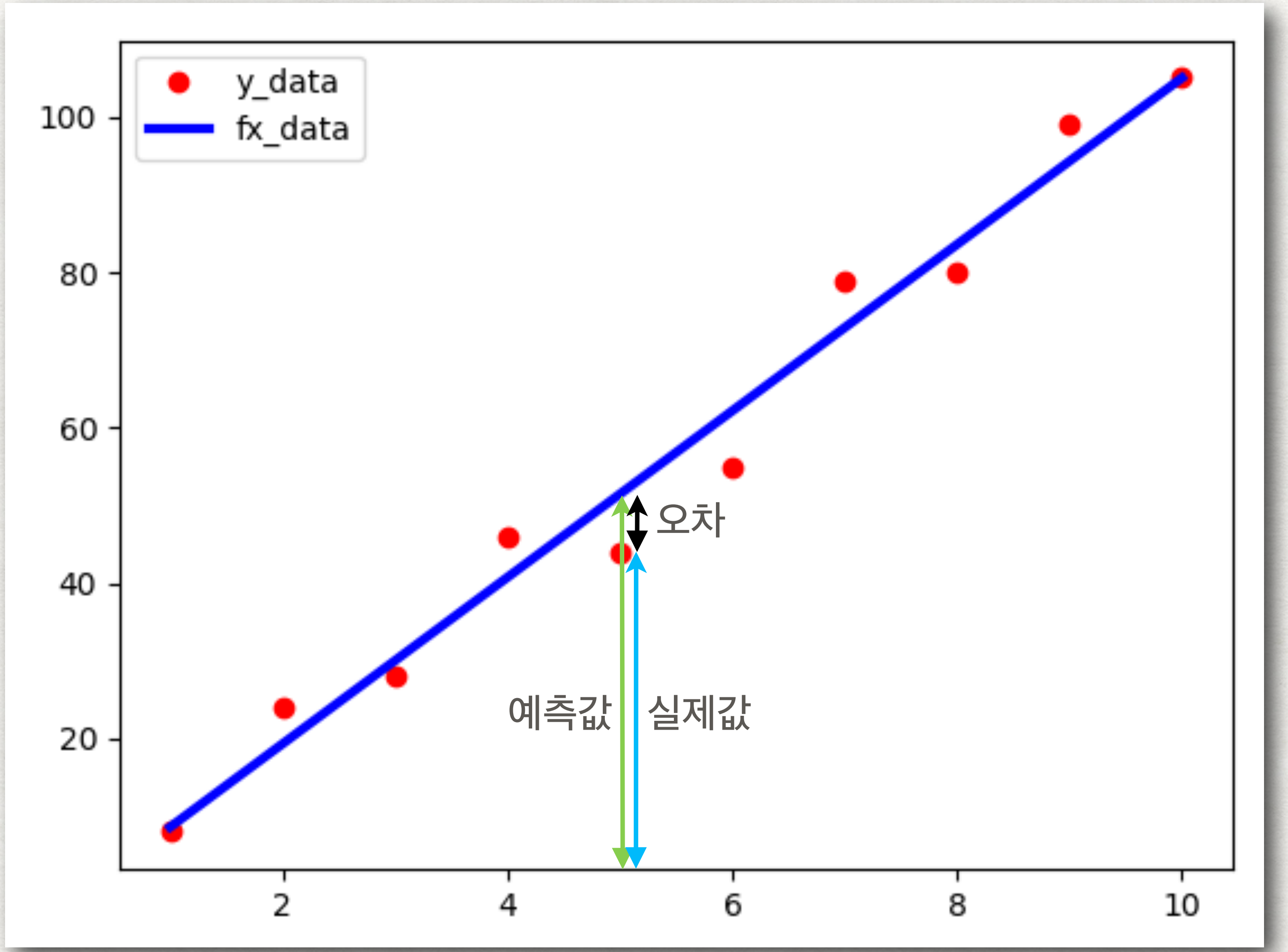
최소 제곱법

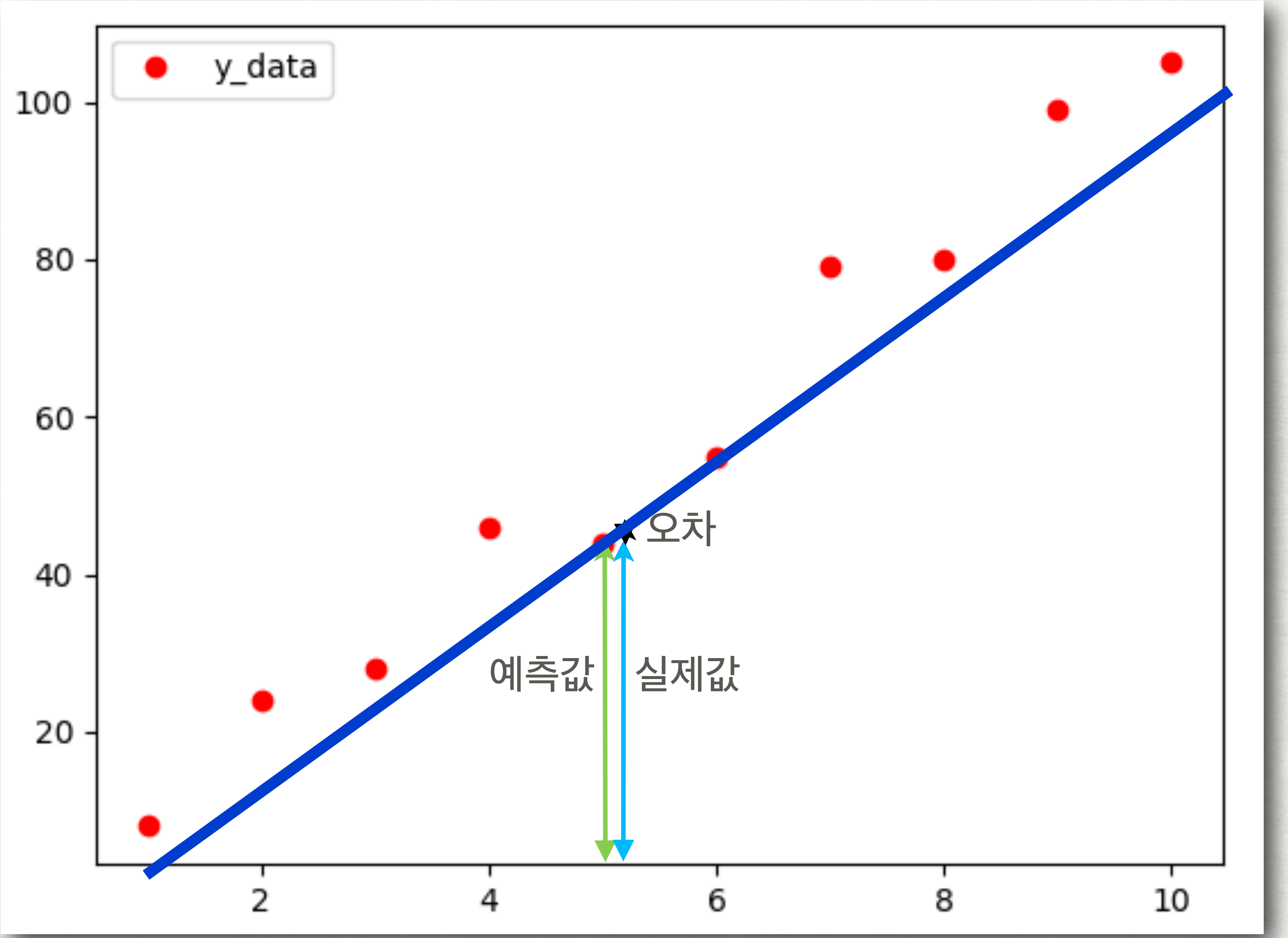
오차



회귀직선을 통해 예측한 값과 실제 데이터의 값의 차이

최소의 오차를 가진 직선을 찾는 것이 선형회귀의 목표

오차

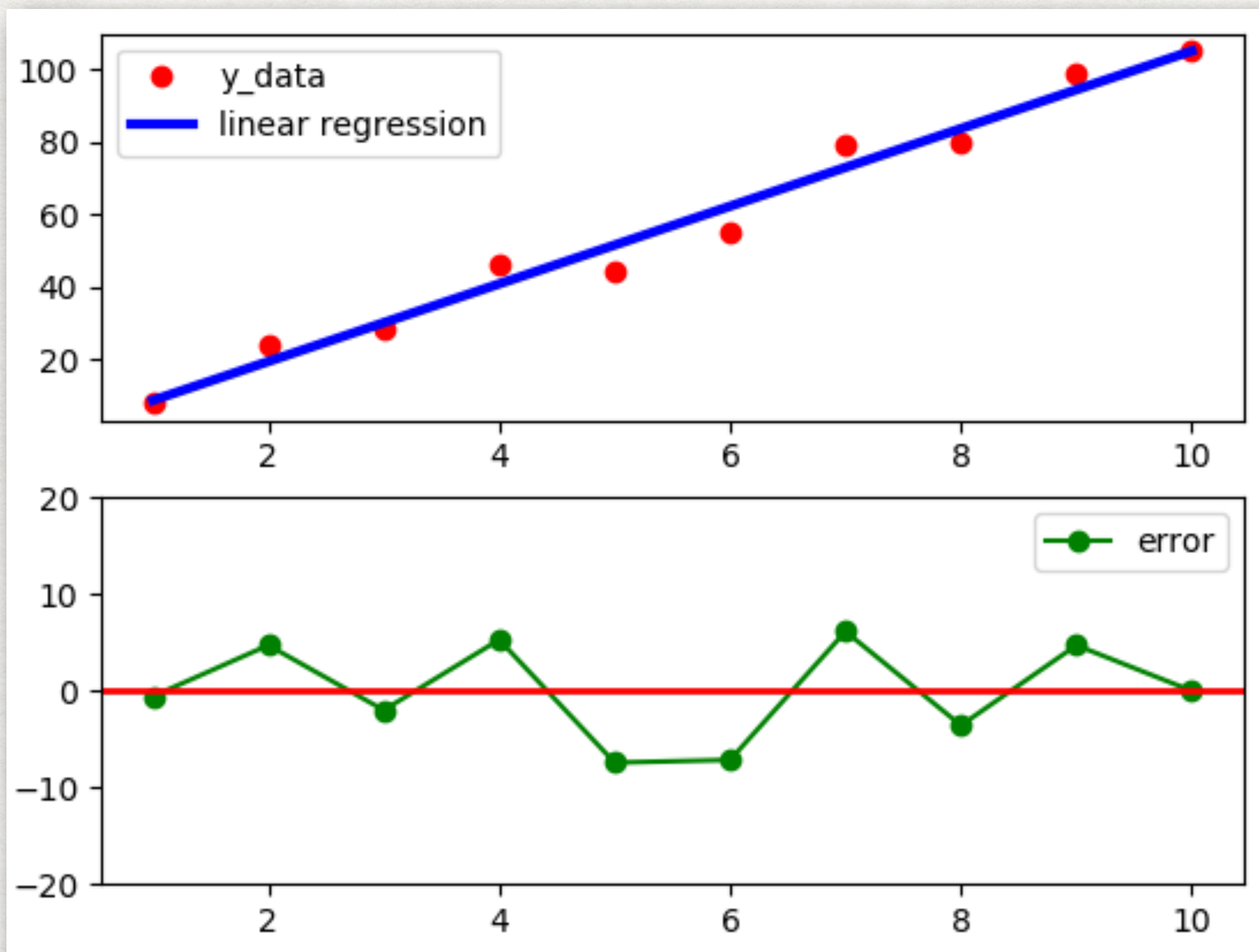


한 데이터의 오차를 0으로 만들었지만 다른 데이터의 오차가 더 커지는 문제 발생

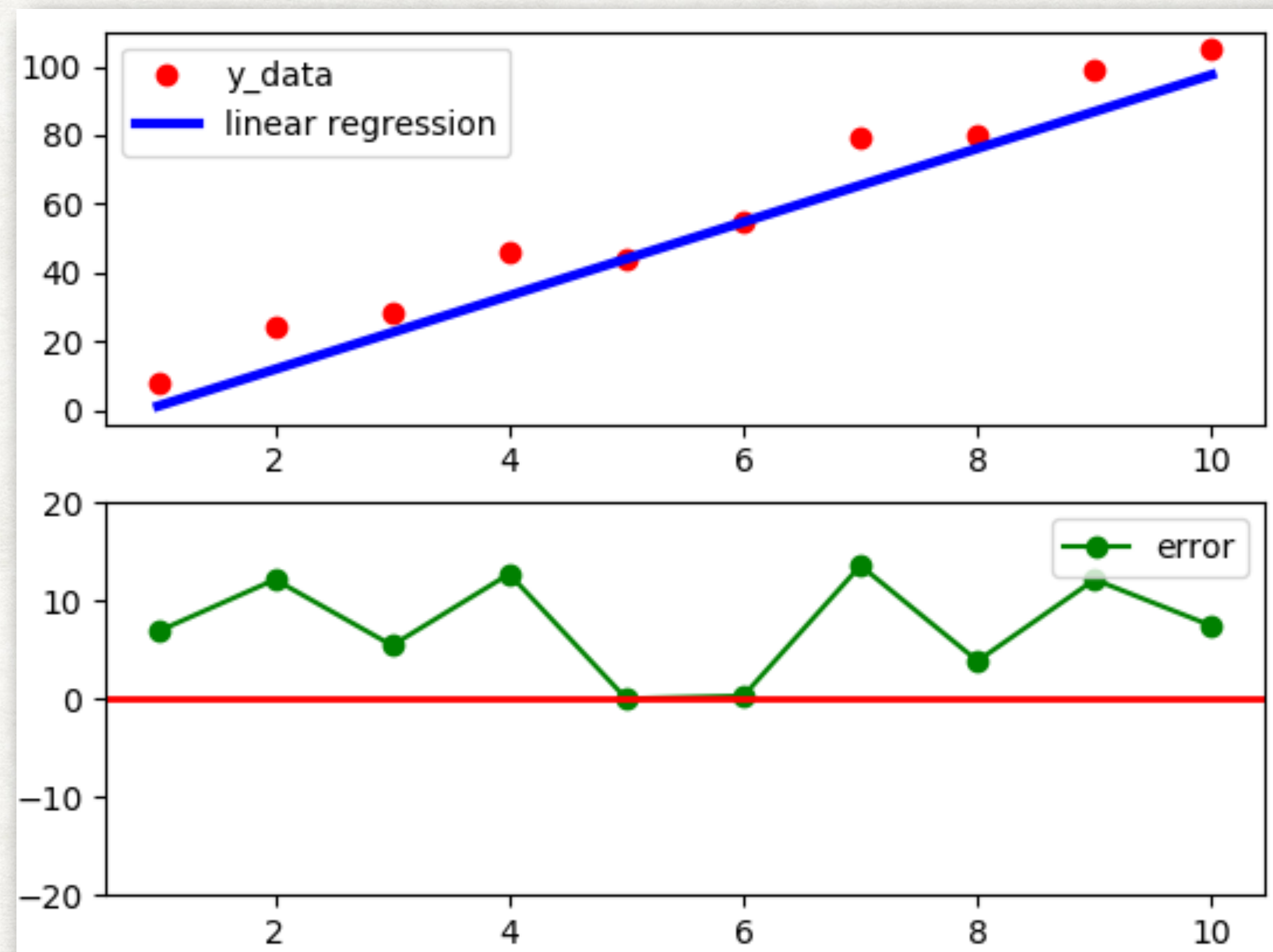
최소 제곱법을 이용해서 문제 해결

lab-06.py, lab-07.py

오차 비교

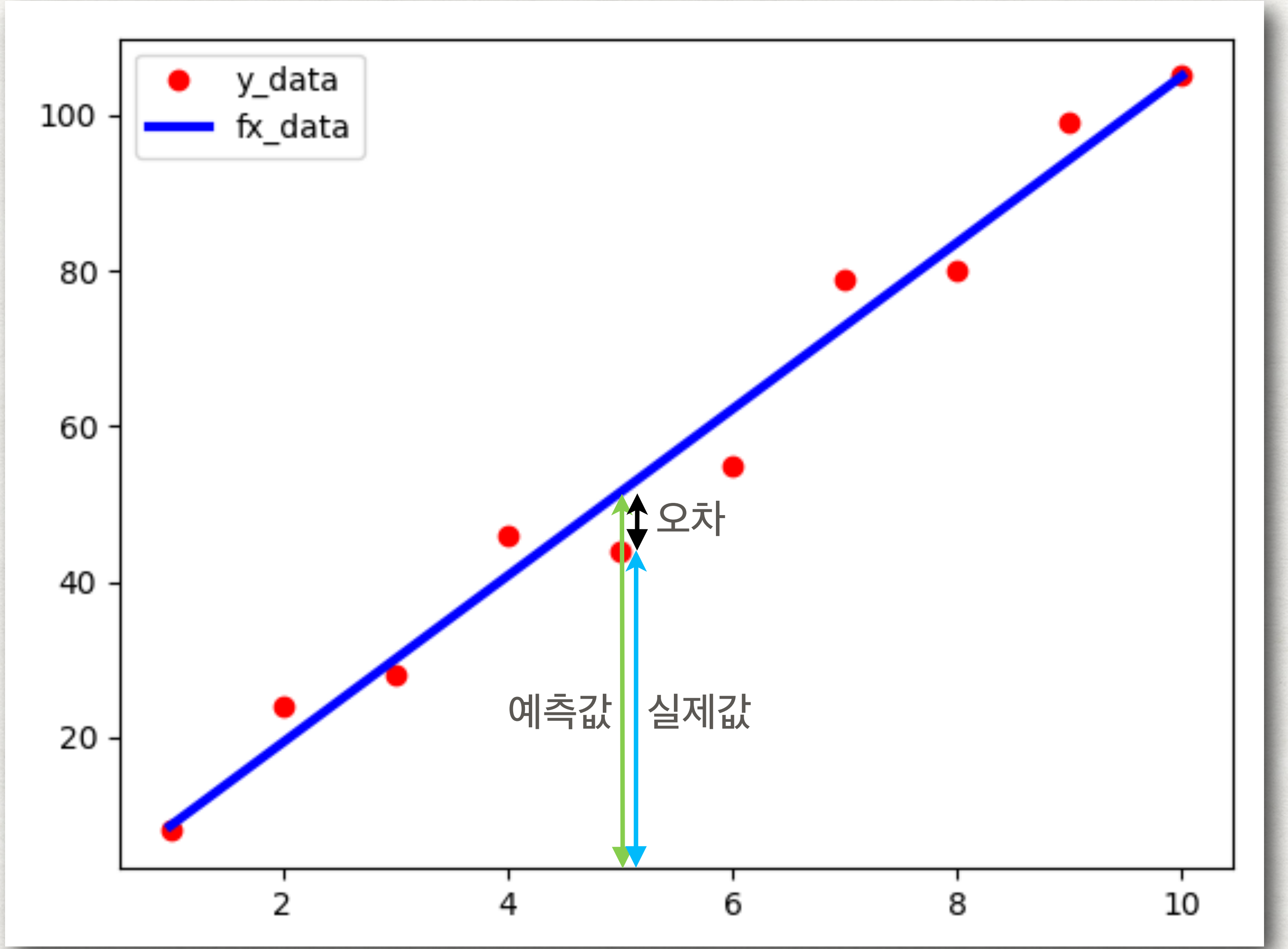


0을 기준으로 고르게 퍼져있는 오차



큰 양수 오차들이 많이 발생

최소 제곱법

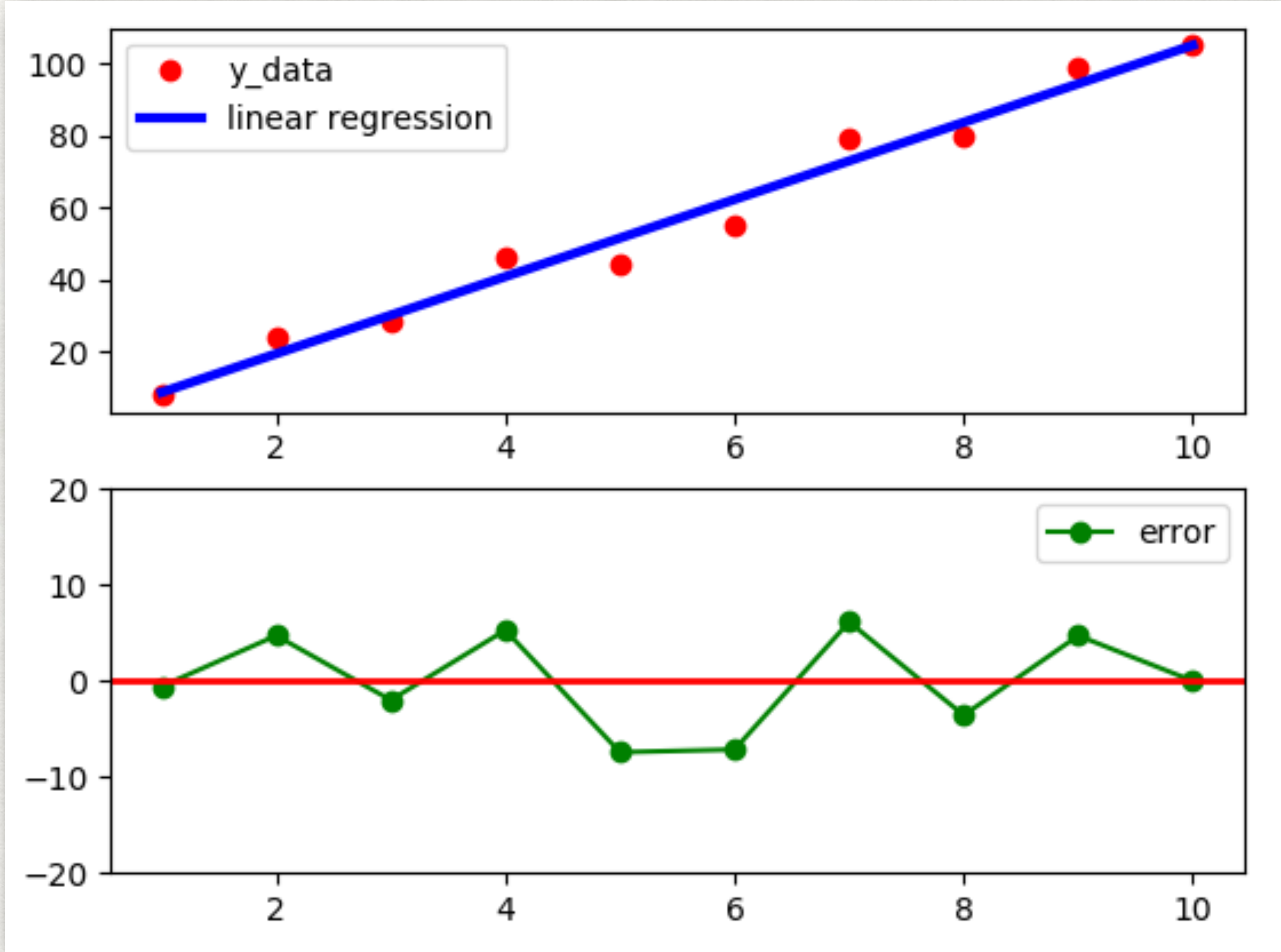


회귀직선을 찾는 방법

오차를 **제곱한 값**들의 평균을 최소화 시키기

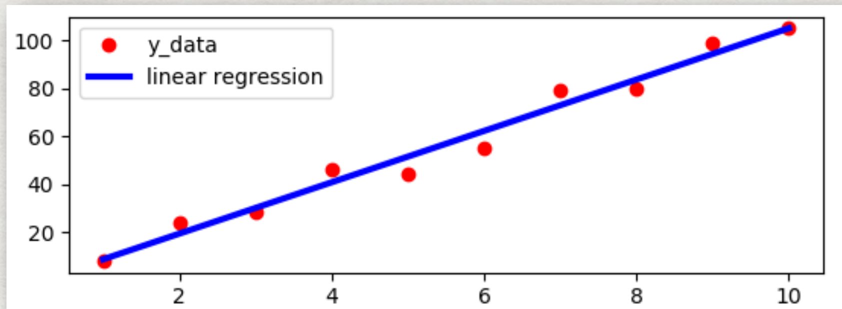
오차를 제곱해서 평균을 구하는 이유는?

최소 제곱법



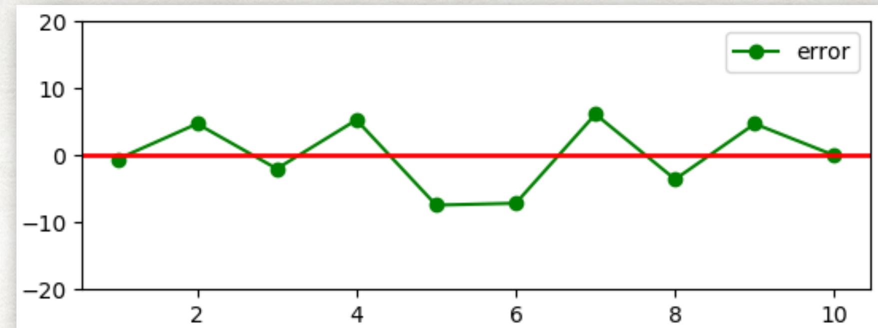
회귀직선의 윗쪽에 분포된 데이터는 양수 오차를 가지고
회귀직선의 아랫쪽에 분포된 데이터는 음수 오차를 가진다.

동일한 부호를 가지게 하기 위해서 제곱 연산 수행



lab-08.py

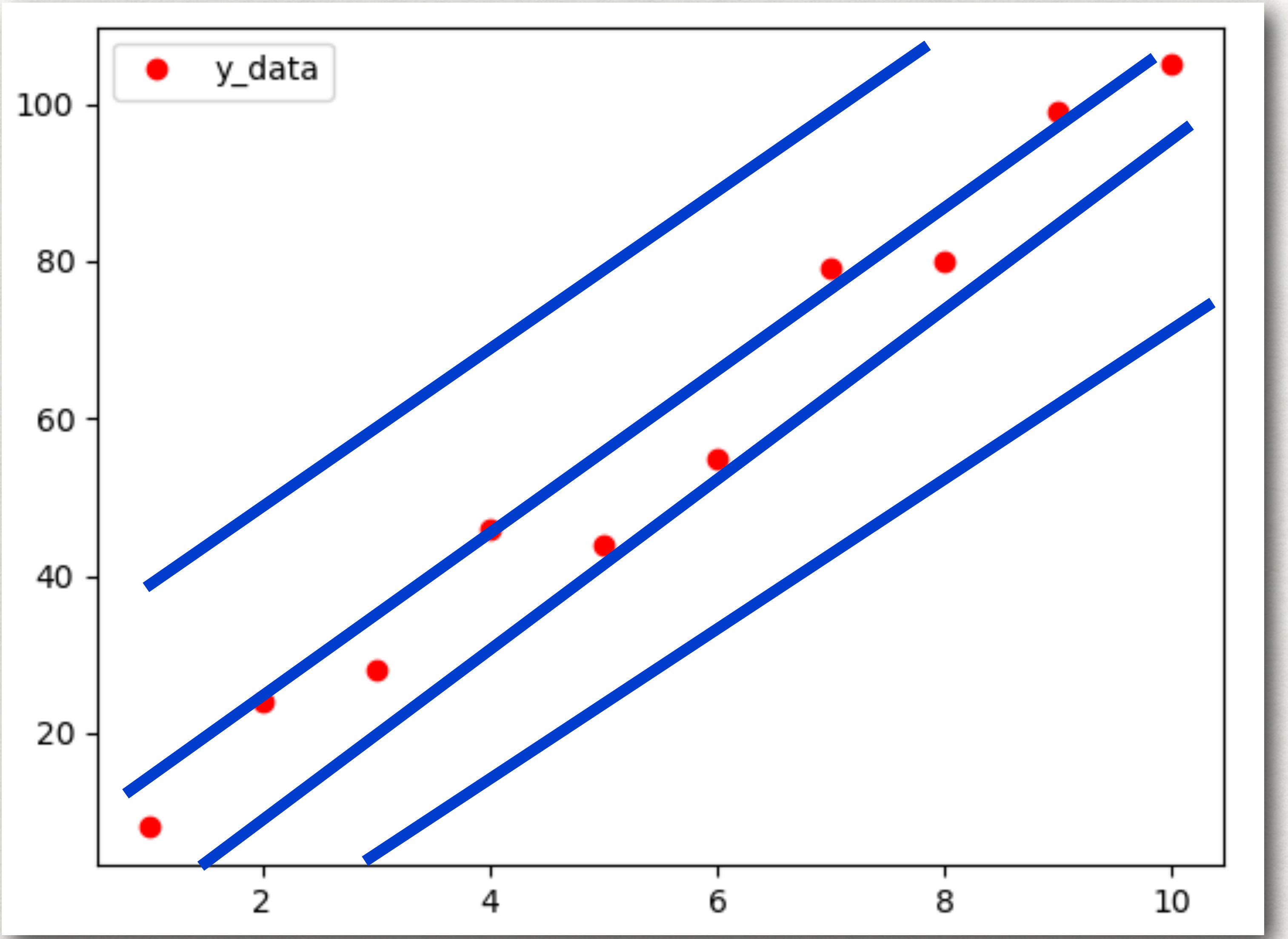
최소 제곱법



x값	1	2	3	4	5	6	7	8	9	10
실제 데이터값	8	24	28	46	44	55	79	80	99	105
회귀직선 예측값	8.58	19.29	30.01	40.72	51.44	62.15	72.87	83.58	94.30	105.01
오차 (실제값 - 예측값)	-0.58	4.70	-2.01	5.27	-7.44	-7.15	6.12	-3.58	4.69	-0.01
오차 제곱	0.33	22.11	4.04	27.80	55.38	51.23	37.54	12.87	22.06	0.00

오차 제곱의 평균 : 23.340606061393394

최소 제곱법



y = ax + b

x : 입력 데이터
y : 출력 데이터
a : 기울기
b : y절편

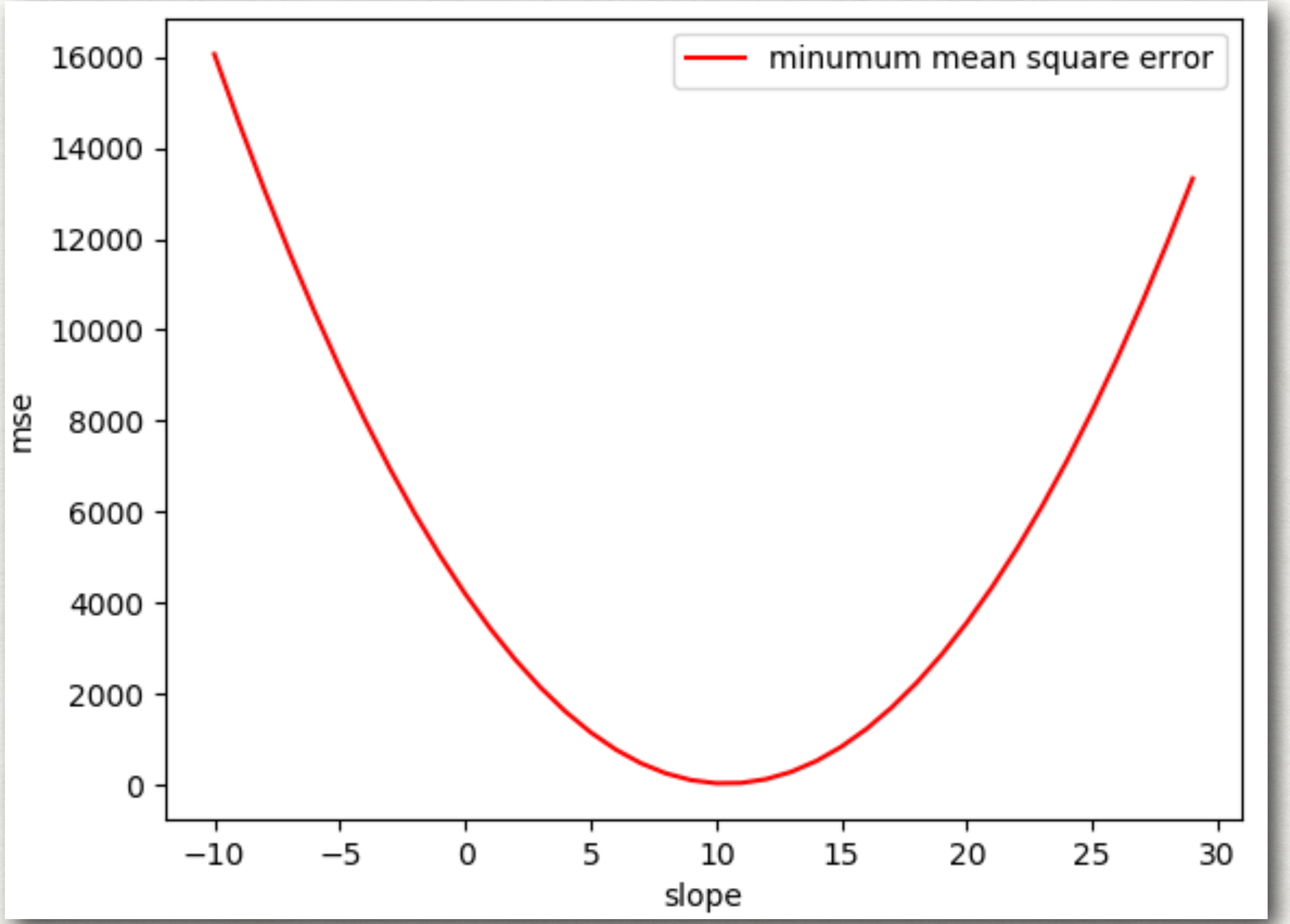
a, b 값에 따라서
선의 모양이 결정된다

회귀직선을 찾는 방법

오차를 **제곱한 값**들의 평균을 최소화 시키기

오차 제곱의 평균이 최소가 되는 a, b 값을 찾는 것

최소 제곱법



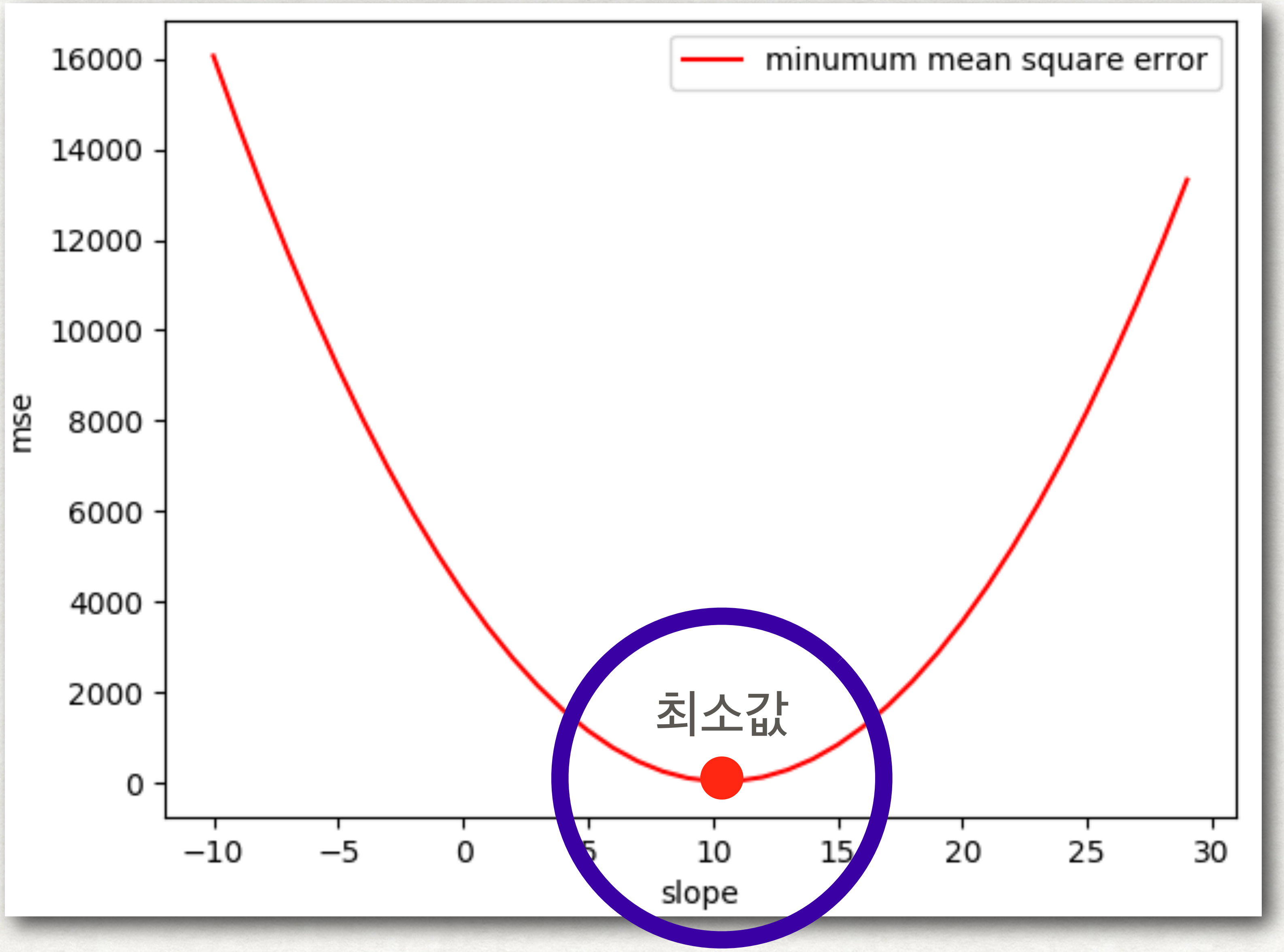
y = ax

b(y절편)는 임시로 삭제

a(slope) 값에 따른
오차 제곱의 평균값(mse)
그래프

오차 제곱의 평균(mse)이 최소가 되는 a(slope)값은?

최소 제곱법



y = ax

b(y절편)는 임시로 삭제

a(slope) 값에 따른
오차 제곱의 평균값(mse)
그래프

오차 제곱의 평균(mse)이 최소가 되는 a(slope)값은?

식 유도

$$MSE(a) = \frac{1}{k}\sum_{i=1}^{k}(y_i - a \cdot x_i)^2$$

$$\frac{d}{da}MSE(a) = \frac{2}{k}\sum_{i=1}^{k}(y_i - a \cdot x_i) \cdot (-x_i)$$

$$\frac{d}{da}MSE(a) = \frac{2}{k} \cdot (a\sum_{i=1}^{k}x_i^2 - \sum_{i=1}^{k}x_i \cdot y_i))$$

x_i = 입력 데이터
y_i = 출력 데이터
k = 데이터 개수

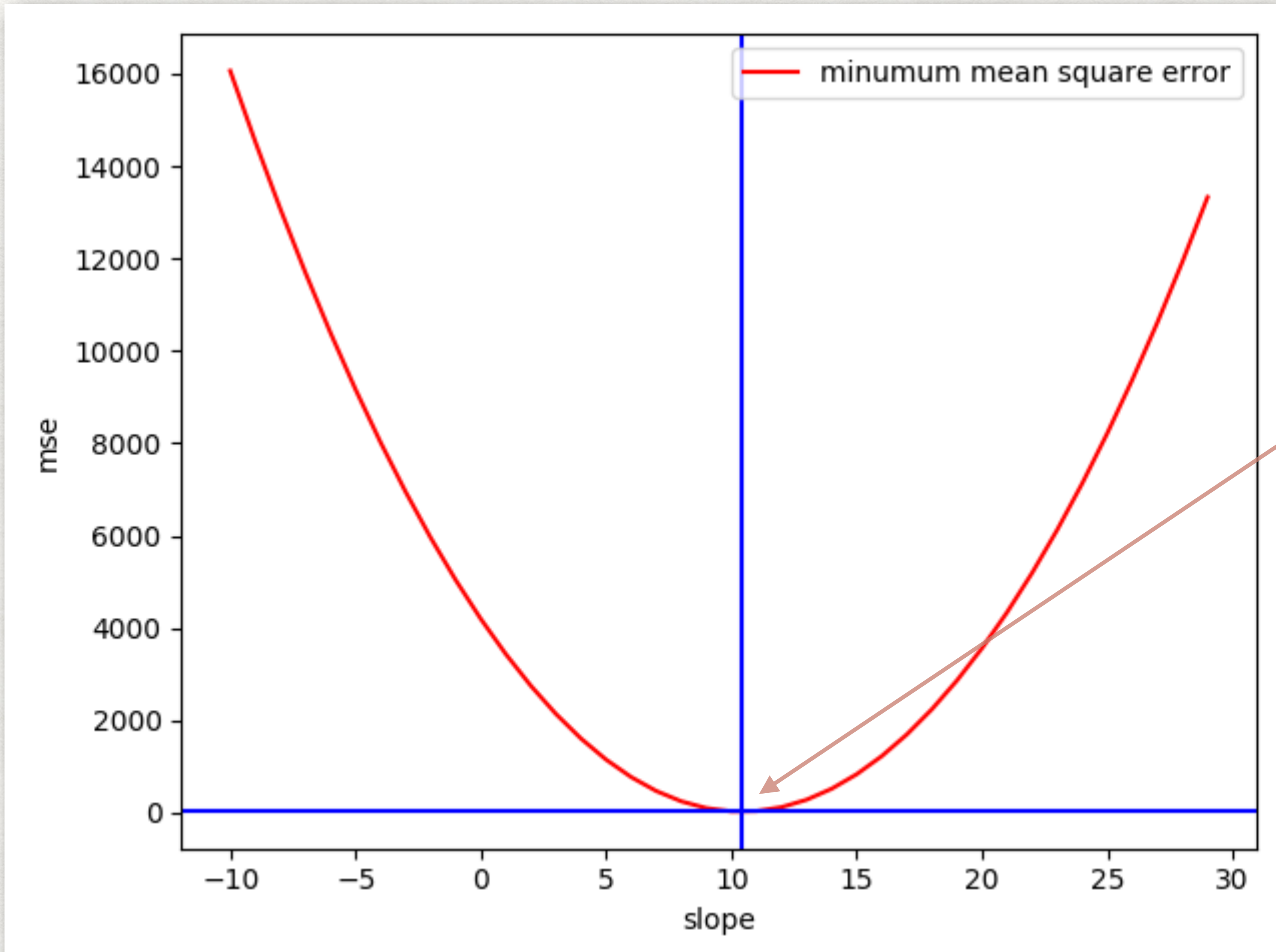
$\boxed{y = ax}$

y절편(b)은 없다고 가정

이차함수 MSE(a)의 최소가 되는 a값은? **a로 미분한 식이 0이 되는 a값**

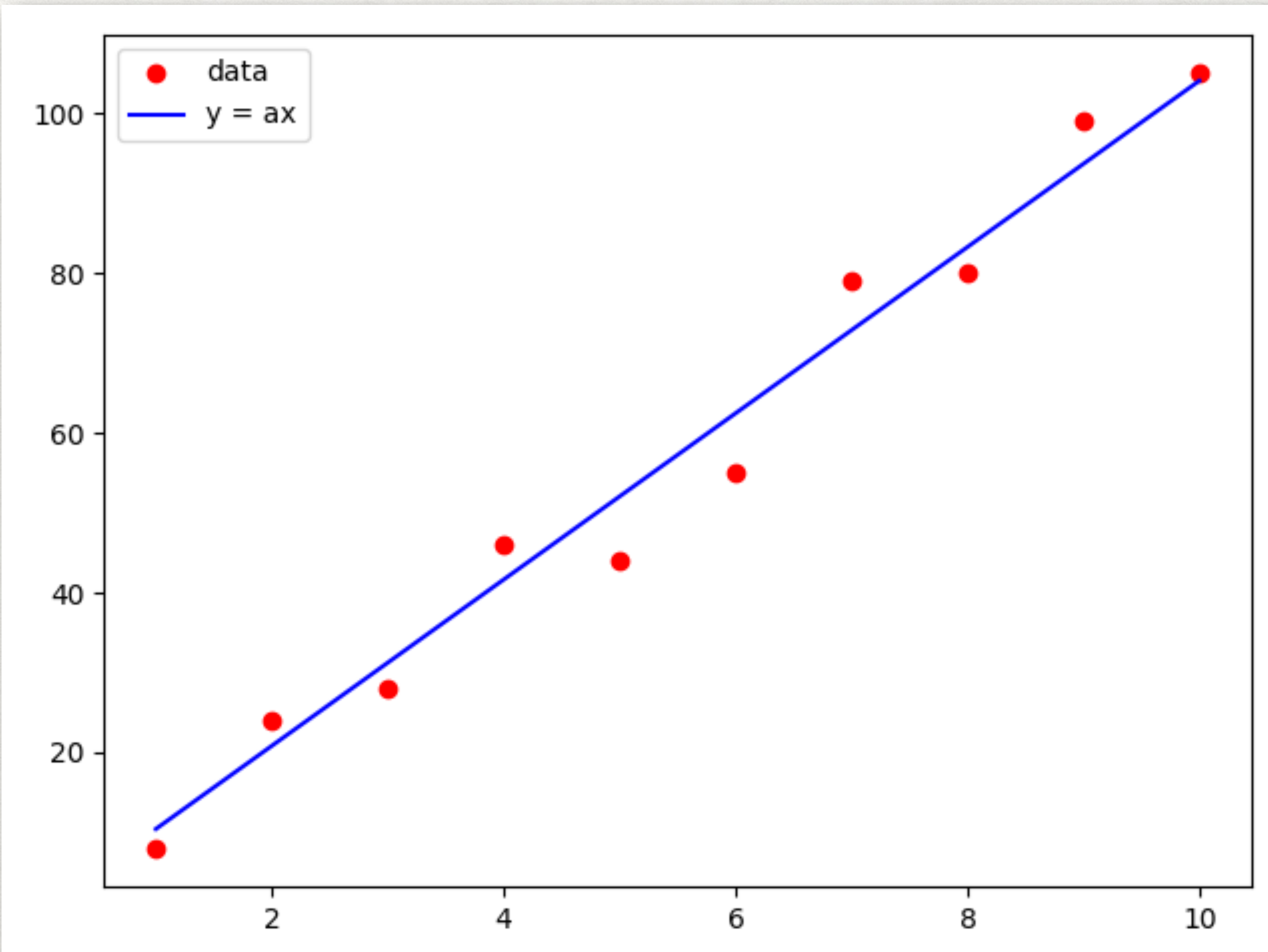
$$\therefore a = \frac{\sum_{i=1}^{k} x_i \cdot y_i}{\sum_{i=1}^{k} x_i^2}$$

최소 제곱법



$$slope = \frac{\sum_{i=1}^{k} x_i \cdot y_i}{\sum_{i=1}^{k} x_i^2}$$

최소 제곱법



$$a = \frac{\sum_{i=1}^{k} x_i \cdot y_i}{\sum_{i=1}^{k} x_i^2}$$

$$y = ax$$

x_i = 입력 데이터

y_i = 출력 데이터

k = 데이터 개수

식 유도 2

$\boxed{y = ax + b}$

$$MSE(a, b) = \frac{1}{k} \sum_{i=1}^{k} (y_i - a \cdot x_i - b)^2$$

$$\frac{d}{da} MSE(a, b) = \frac{2}{k} \sum_{i=1}^{k} (y_i - a \cdot x_i - b) \cdot (-x_i)$$

$$\frac{d}{db} MSE(a, b) = \frac{2}{k} \sum_{i=1}^{k} (y_i - a \cdot x_i - b) \cdot (-1)$$

$$\frac{d}{da} MSE(a, b) = \frac{2}{k} \cdot (a \sum_{i=1}^{k} x_i^2 + b \sum_{i=1}^{k} x_i - \sum_{i=1}^{k} x_i \cdot y_i))$$

$$\frac{d}{da} MSE(a, b) = \frac{2}{k} \cdot (bk + a \sum_{i=1}^{k} x_i - \sum_{i=1}^{k} y_i))$$

이차함수 MSE(a, b)의 최소가 되는 a값은?

a로 미분한 식도 0이 되고,
b로 미분한 식도 0이 되는 a, b 값

$$\therefore a = \frac{\sum_{i=1}^{k} x_i \cdot y_i - b \sum_{i=1}^{k} x_i}{\sum_{i=1}^{k} x_i^2}$$

$$\therefore b = \frac{\sum_{i=1}^{k} y_i - a \sum_{i=1}^{k} x_i}{k}$$

식 유도 2 - 1

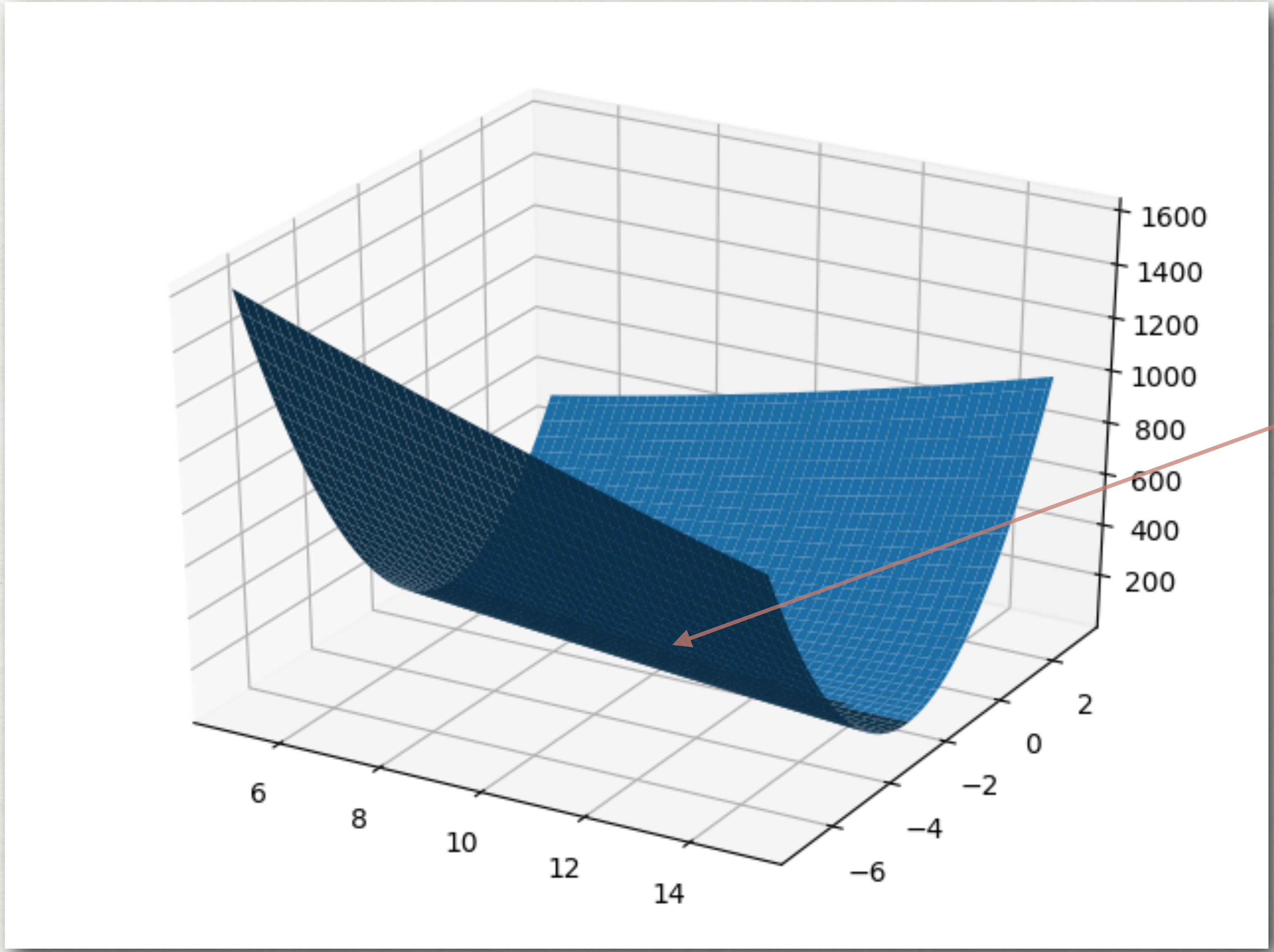
$$\therefore a = \frac{\sum_{i=1}^{k} x_i \cdot y_i - b\sum_{i=1}^{k} x_i}{\sum_{i=1}^{k} x_i^2} \qquad \therefore b = \frac{\sum_{i=1}^{k} y_i - a\sum_{i=1}^{k} x_i}{k}$$

연립

$$A = \sum_{i=1}^{k} x_i \quad B = \sum_{i=1}^{k} y_i \quad C = \sum_{i=1}^{k} x_i^2 \quad D = \sum_{i=1}^{k} x_i \cdot y_i$$

$$\boxed{\therefore a = \frac{kD - AB}{kC - A^2} \qquad \therefore b = \frac{B - aA}{k}}$$

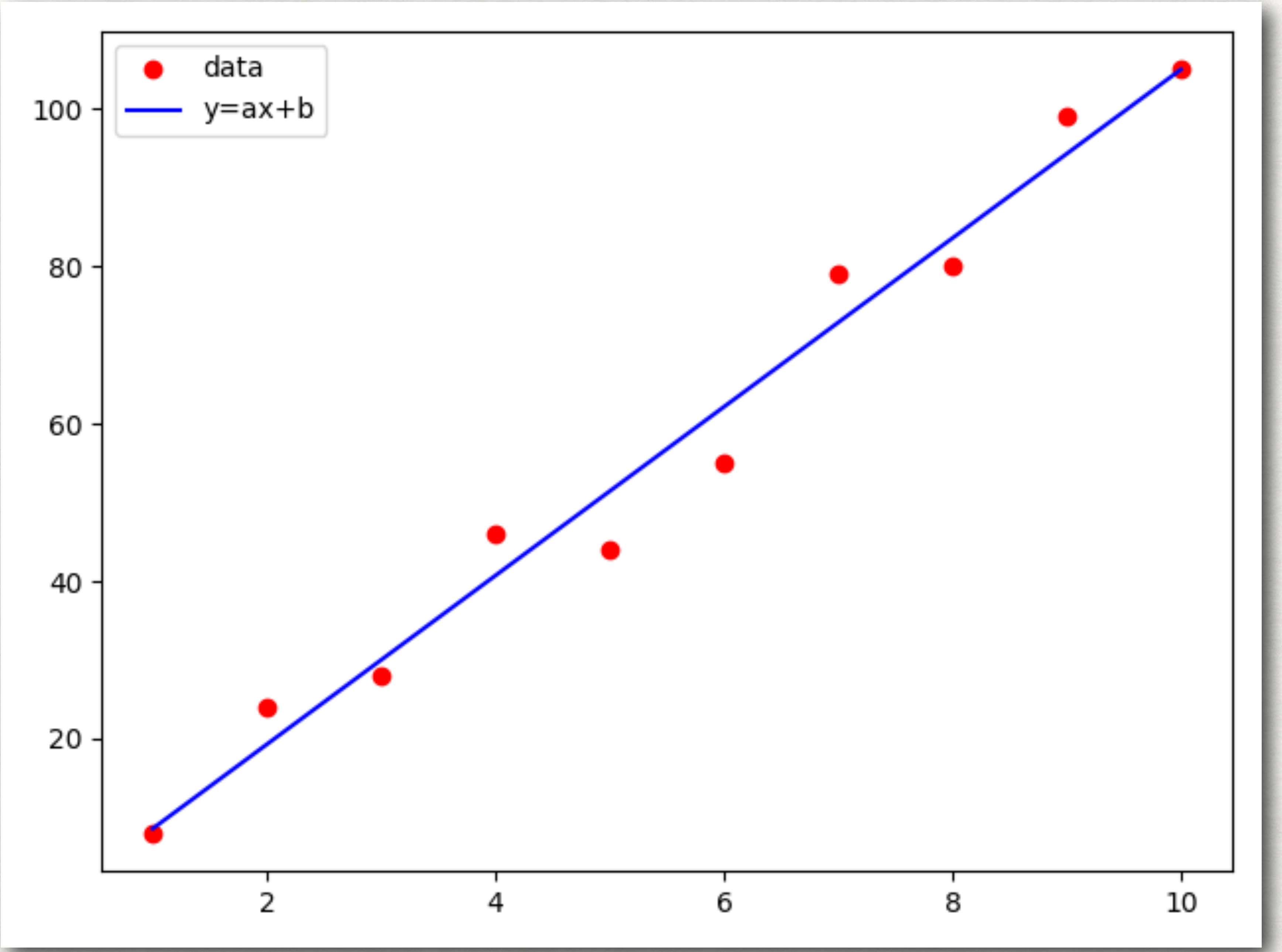
최소 제곱법



$$\therefore a = \frac{kD - AB}{kC - A^2}$$

$$\therefore b = \frac{B - aA}{k}$$

최소 제곱법

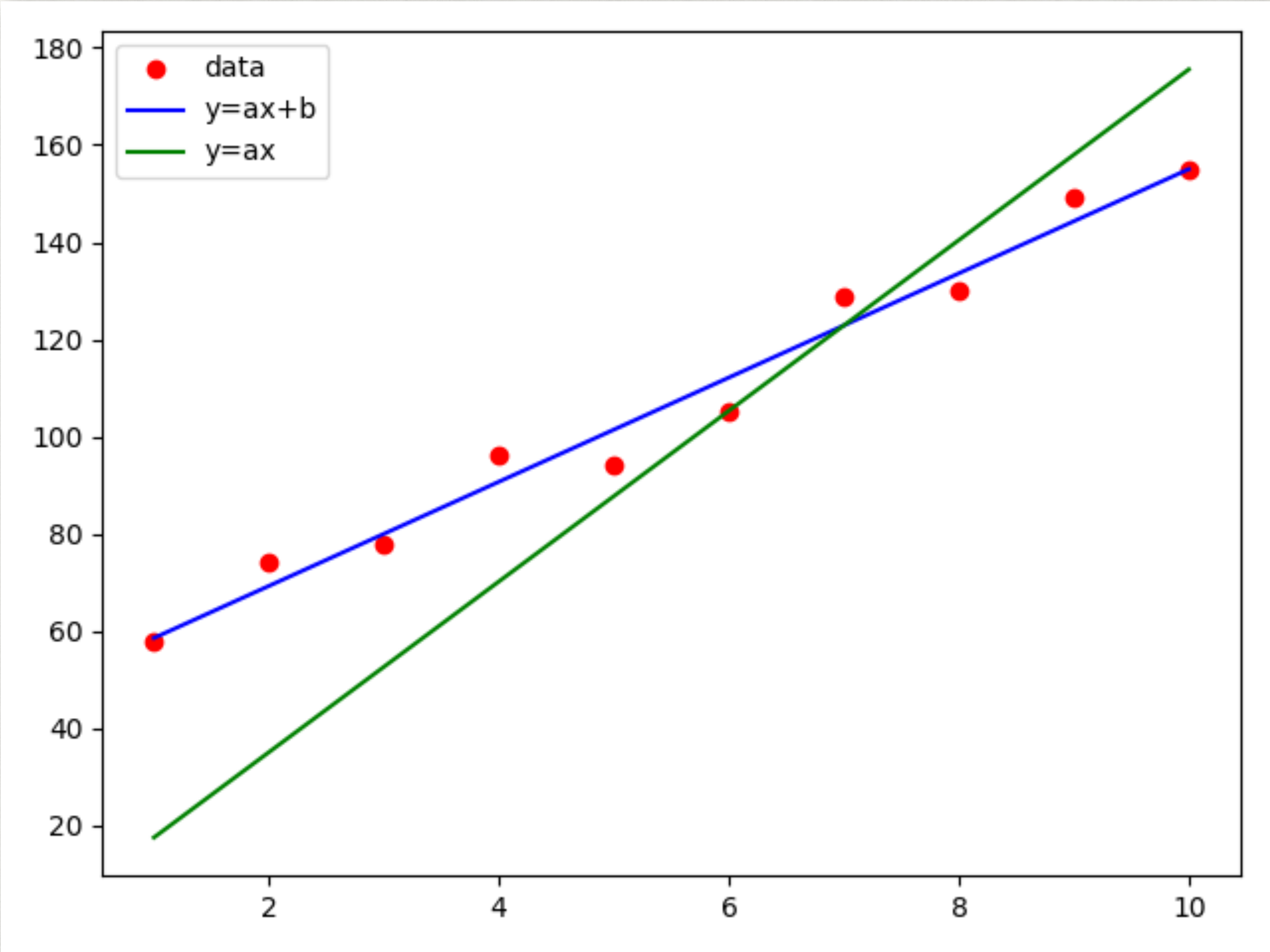


$$a = \frac{kD - AB}{kC - A^2}$$

$$b = \frac{B - aA}{k}$$

$$y = ax + b$$

차이 비교



파일에서 데이터 불러오기

- import numpy as np
    - 파이썬에서 다양한 수치연산을 쉽고 편하게 구현하기 위한 라이브러리
- data = np.loadtxt(파일명, dtype=데이터타입, delimiter=구분문자)
    - 파일에서 데이터 불러오기
- x_data = data[:, 0]
- y_data = data[:, 1]
    - 행, 열 기준으로 데이터를 적절히 잘라서 사용
- **주의할 점!** 행, 열 기준으로 자르는 연산은 numpy 배열만 가능(리스트 불가)

lab-16.py

행, 열 기준으로 데이터 가공하기

data[행범위, 열범위]

data[1:7, 2:7]

2:7

1:7

$$
data = \begin{bmatrix}
00 & 01 & 02 & 03 & 04 & 05 & 06 & 07 & 08 & 09 \\
10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 \\
20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 \\
30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 \\
40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 \\
50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 \\
60 & 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 \\
70 & 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 \\
80 & 81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 \\
90 & 91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99
\end{bmatrix}
$$