

코딩으로 확인하는 최소 제곱법(Method of Least Squares)

일차 함수의 기울기 a와 절편 b를 최소 제곱법으로 구하기

- 기울기 a 는
 $(x-x\text{평균})(y-y\text{평균})\text{의 합} / (x-x\text{평균})\text{제곱의 합}$

- 절편 b 는
 $y\text{평균} - (x\text{평균} * \text{기울기 } a)$

In [1]:

```
1 import numpy as np
2
3 # x 값과 y값
4 x=[2, 4, 6, 8]
5 y=[81, 93, 91, 97]
```

In [2]:

```
1 print("x : ", x)
2 print("y : ", y)
```

x : [2, 4, 6, 8]
y : [81, 93, 91, 97]

In [3]:

```
1 # x와 y의 평균값
2 mx = np.mean(x)
3 my = np.mean(y)
4 print("x의 평균값:", mx)
5 print("y의 평균값:", my)
```

x의 평균값: 5.0
y의 평균값: 90.5

최소제곱법 공식 : $(x-x\text{평균})(y-y\text{평균})\text{의 합} / (x-x\text{평균})\text{제곱의 합}$

- 최소 제곱법에서 분자에 해당하는 식 : $(x-x\text{평균})(y-y\text{평균})\text{의 합} \rightarrow$ 함수 dividend 로 정의

In [4]:

```
1 # Method of Least Squares 함수 정의
2 # x : [2, 4, 6, 8] , x의 평균 mx : 5.0
3 # y : [81, 93, 91, 97] , y의 평균 my : 90.5
4
5 def dividend(x, mx, y, my):
6     gradient = 0
7     for i in range(len(x)): # len(x)는 4이므로, i는 0~3
8         gradient += (x[i] - mx) * (y[i] - my)
9     return gradient
```

- 최소 제곱법에서 분자에 해당하는 부분은 위에서 정의된 dividend 함수를 호출해서 실행

In [5]:

```
1 dividend_result = dividend(x, mx, y, my) #분자 즉, 피제수(dividend)에 해당
2 print("분자:", dividend_result) # 기울기 공식의 분자
```

분자: 46.0

- 최소 제곱법에서 분모에 해당하는 식 : $(x-x\text{평균})^2$ 의 합

In [6]:

```
1 # x : [2, 4, 6, 8] , x의 평균 mx : 5.0
2
3 divisor = sum([(i - mx)**2 for i in x]) #분모 즉, 제수(divisor)에 해당
4 print("분모:", divisor) # 기울기 공식의 분모
```

분모: 20.0

- 최소 제곱법에서 기울기 a, 절편 b를 계산

기울기 a 는

$(y-y\text{평균})$ 의 합 / $(x-x\text{평균})$ 의 합

절편 b 는

$y\text{평균} - (x\text{평균} * \text{기울기 } a)$

In [7]:

```
1 # 기울기 a : (x-x평균) (y-y평균)의 합 / (x-x평균)제곱의 합
2 # 즉, 분자 dividend_result / 분모 divisor
3
4 a = dividend_result / divisor
5 print("기울기 a =", a)
```

기울기 a = 2.3

In [8]:

```
1 # 절편 b : y평균 - (x평균 * 기울기 a)
2 # 즉, 90.5 - (5.0 * 2.3)
3
4 b = my - (mx*a)
5 print("y 절편 b =", b)
```

y 절편 b = 79.0

In []:

```
1
```