

# +·\*~ 2021 SMARCLE 겨울방학 인공지능 스터디 ~·\*

## 둘째마당 - 3장 선형회귀

1주차 1팀 2021.1.4.

김건우 박지하 송혜원





# 목차

## 1. 개요

- (1) 우리가 사용할 책
- (2) 머신러닝, 딥러닝은 무엇일까?



## 2. 딥러닝을 하려면?

- (1) 머신러닝에 필요한 간단한 지식들(2)
- 딥러닝의 원리

## 3. 선형회귀란?

- (1) 선형회귀의 정의에 대하여
- (2) 가장 훌륭한 예측선 찾기



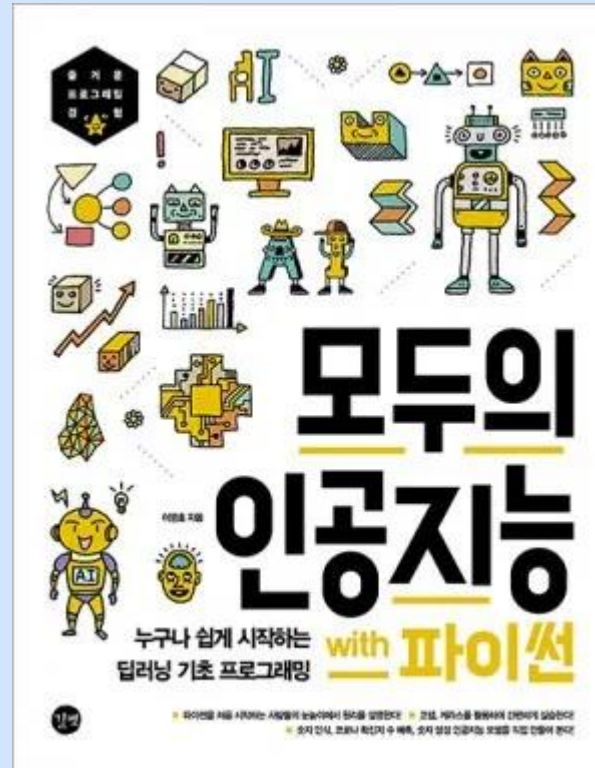
## 4. 기울기와 절편 구하기

- (1) 최소 제곱법 – 실습1
- (2) 평균 제곱 오차 – 실습2

# 우리의 책 : 모두의 딥러닝

두둥..!

두둥..!



두둥..!

두둥..!

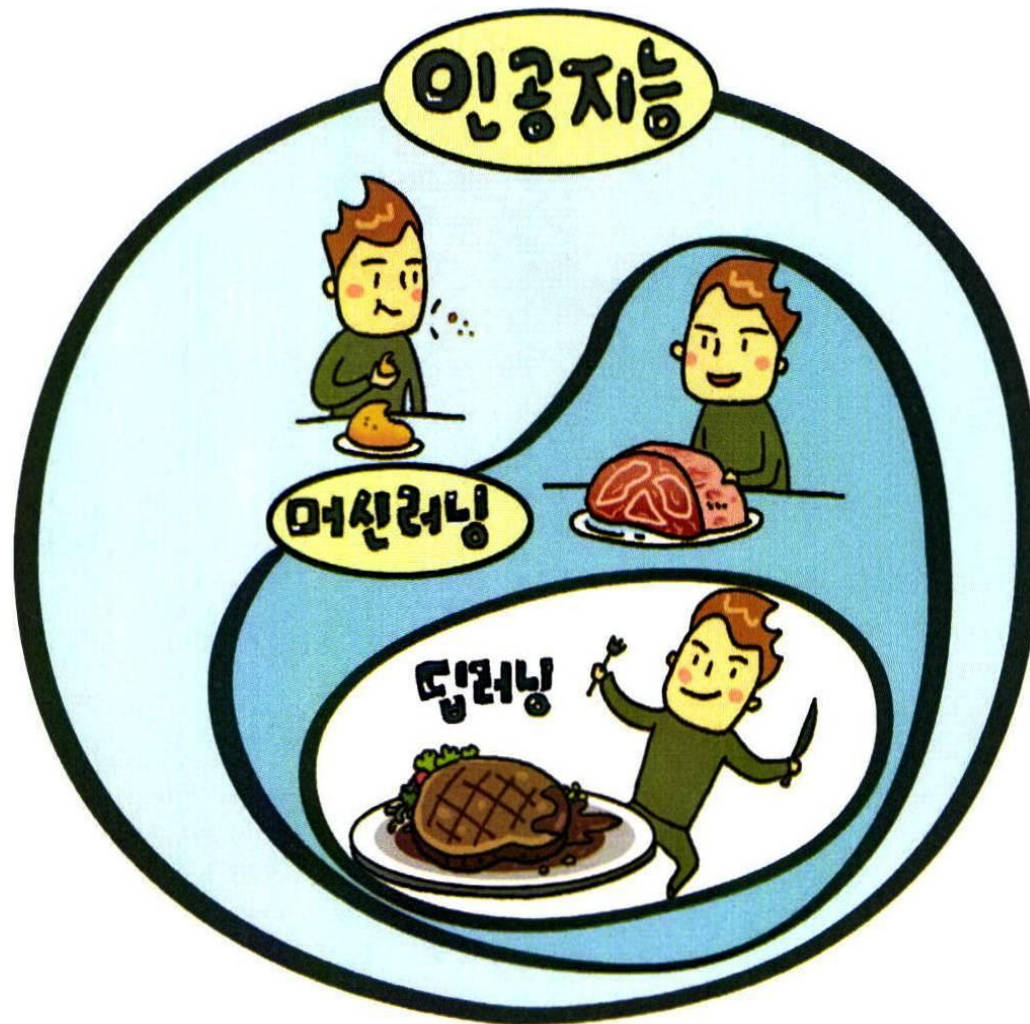




## 머신러닝, 딥러닝?



# 딥러닝이란?





# 머신러닝에 필요한 간단한 지식들! - ① 작업 환경



	장점	단점	결론
주피터 노트북	<ul style="list-style-type: none"> <li>• 아나콘다만 설치한 후 간단하게 이용 가능</li> <li>• 대화형 프로그래밍 가능</li> <li>• 문서화 작업 용이</li> </ul>	<ul style="list-style-type: none"> <li>• 컴퓨터 사양에 종속</li> </ul>	설치가 비교적 쉽고 CPU 환경과 GPU 환경 모두에서 무난히 사용할 수 있다.
구글 코랩	<ul style="list-style-type: none"> <li>• 설치 필요 없음</li> <li>• 구글의 GPU, TPU 활용 가능</li> <li>• 구글 드라이브와 연동 가능</li> </ul>	<ul style="list-style-type: none"> <li>• 로그아웃 시 설치한 환경, 샘플 파일, 가중치 값 등이 유실될 수 있음</li> <li>• 여러 사람 동시 작업 불가</li> <li>• 동시에 실행할 수 있는 작업 수 제한</li> </ul>	간단한 설정만으로 구글의 GPU 및 TPU 환경을 이용할 수 있다. 컴퓨터에 GPU가 설치되어 있지 않을 경우 사용하기 좋다.
파이참	<ul style="list-style-type: none"> <li>• 파이썬 전문 프로그래밍 통합 패키지</li> <li>• 디버깅, 코드 자동 완성, 모듈 관리 가능</li> </ul>	<ul style="list-style-type: none"> <li>• 컴퓨터 사양에 종속</li> <li>• 프로그램 설치 및 인터프리터 연결 필요</li> <li>• 시작 시 대기 시간 소요</li> </ul>	컴퓨터에 이미 성능 좋은 GPU가 설치되어 있을 경우 추천하는 툴. 파이썬 통합 패키지 툴의 여러 기능을 활용해 전문적인 프로그래밍을 할 수 있다.



## 머신러닝에 필요한 간단한 지식들! - ② 데이터



강아지

의욕을 가지세요



고양이



<딥러닝 - 지도학습>

<딥러닝 - 비지도학습>

이렇게 딥러닝은 두 종류의 데이터로 구분할 수 있다 !



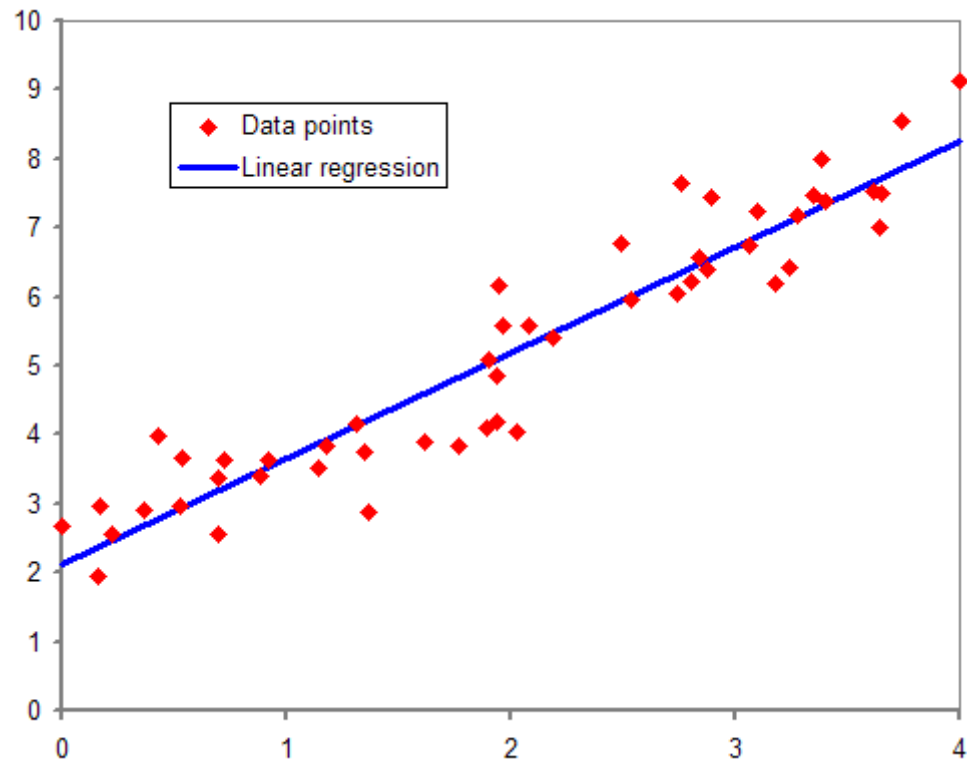
## 머신러닝에 필요한 간단한 지식들! - ③ 파이썬

\* 사고 방법이나 틀은 C언어와 비슷 \*

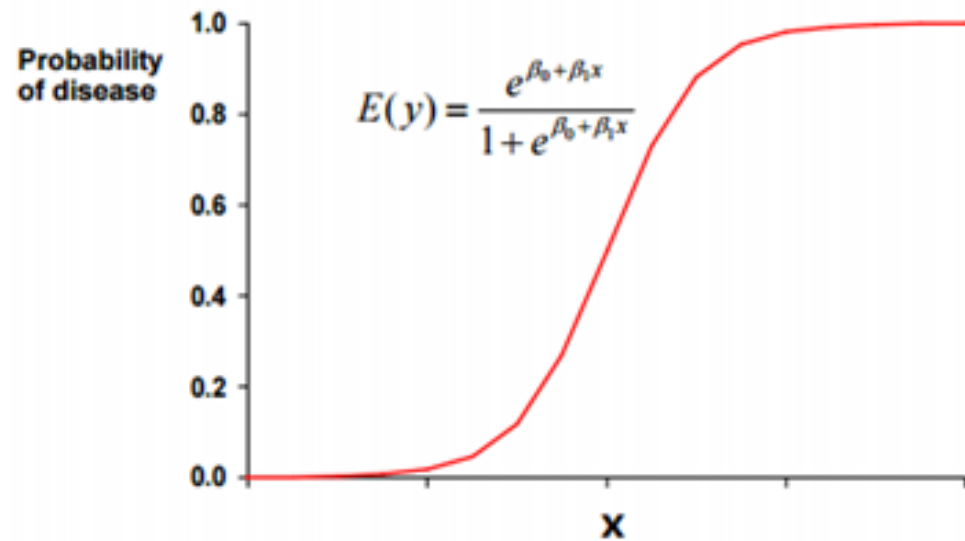
- (파이썬) 함수
- (파이썬) 라이브러리
- numpy 라이브러리



## 딥러닝의 동작 원리 : 선형회귀, 로지스틱 회귀



〈선형 회귀〉

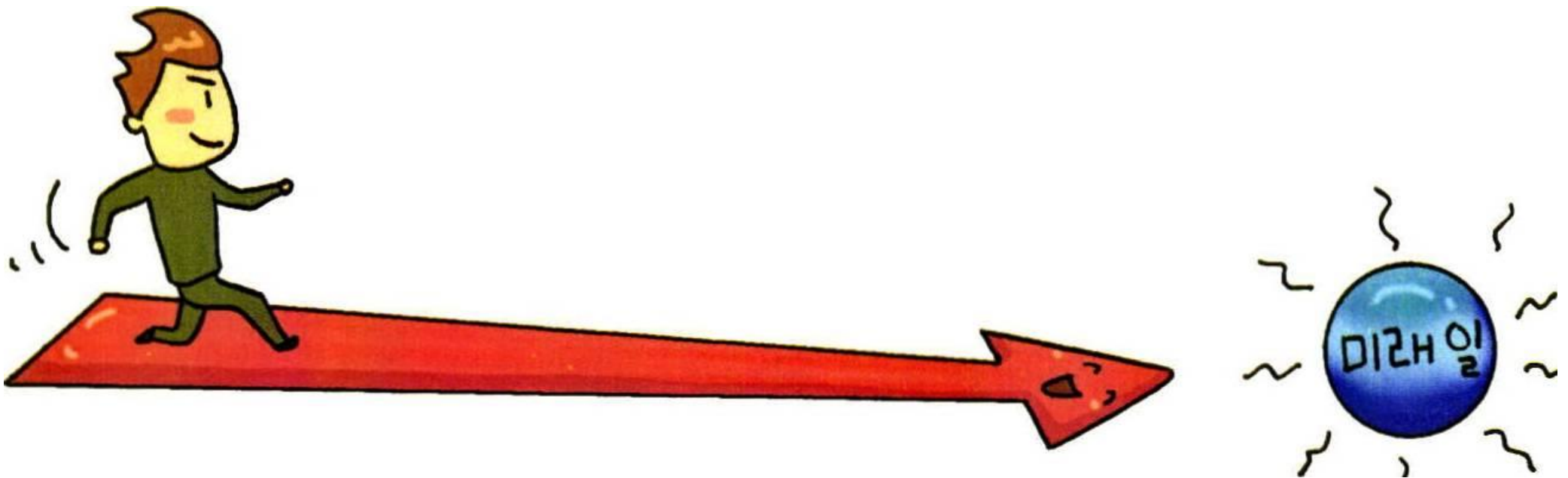


〈로지스틱 회귀〉

~1부 끝 쉬는시간~



## 가장 훌륭한 예측선 긋기 : 선형회귀





## 선형 회귀의 정의

“학생들의 중간고사 성적이 다 다르다.”

“학생들의 중간고사 성적이 [ **‘정보’** ]에 따라 다 다르다.”

$y$ (종속변수)

$x$ (독립변수)



## 가장 훌륭한 예측선이란?

ex) 공부한 시간에 따른 중간고사 성적 데이터

**표 3-1**

공부한 시간과 중간고사  
성적 데이터

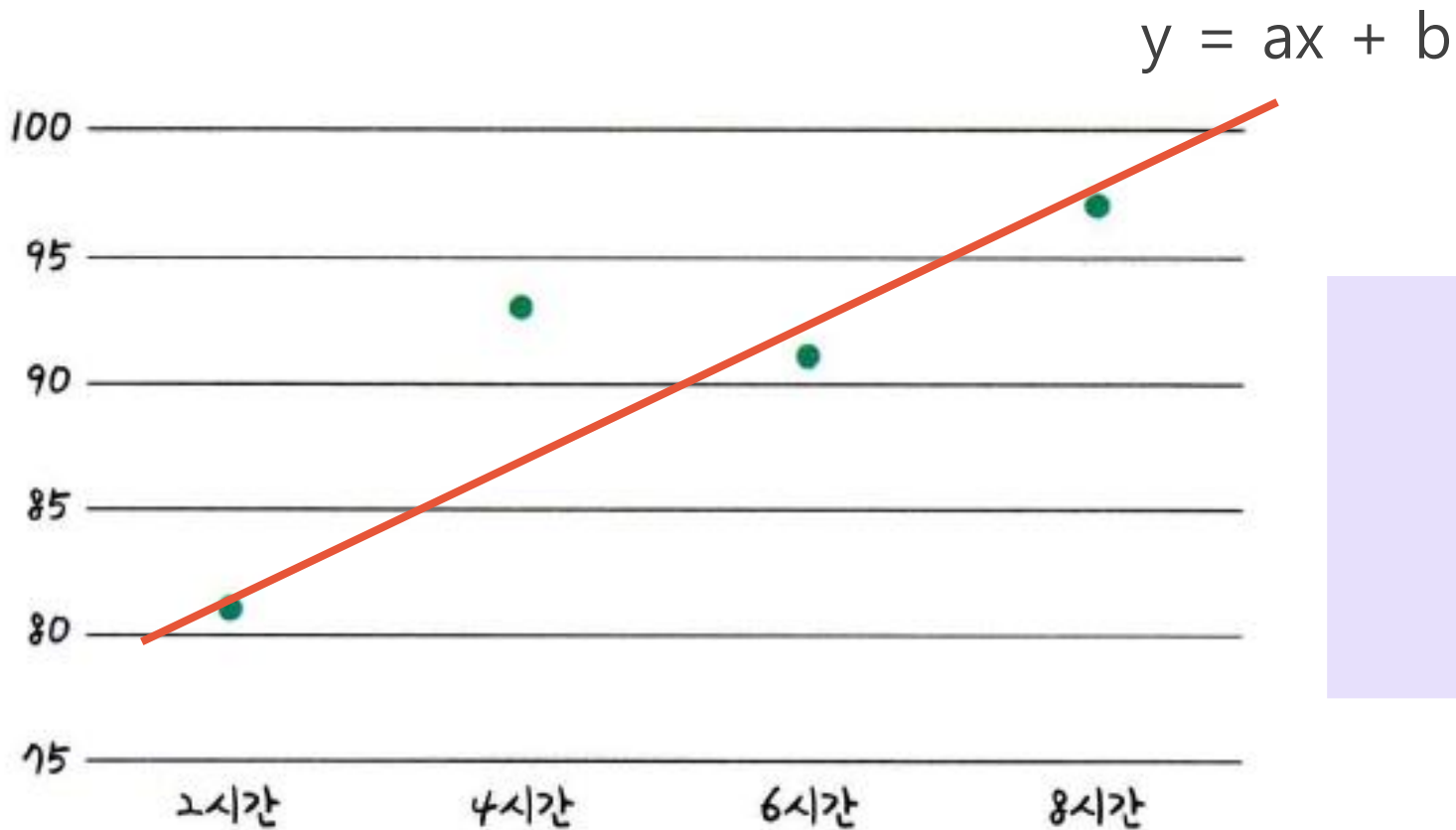
공부한 시간	2시간	4시간	6시간	8시간
성적	81점	93점	91점	97점

$$X = \{2, 4, 6, 8\}$$

$$Y = \{81, 93, 91, 97\}$$



# 가장 훌륭한 예측선이란?



기울기인  $a$  값과  
 $y$  절편  $b$  값을  
정확히 예측!

그림 3-1

공부한 시간과 성적을  
좌표로 표현



## 최소 제곱법 (method of least squares)

$$a = \frac{(x - x \text{ 평균})(y - y \text{ 평균}) \text{의 합}}{(x - x \text{ 평균})^2 \text{의 합}}$$

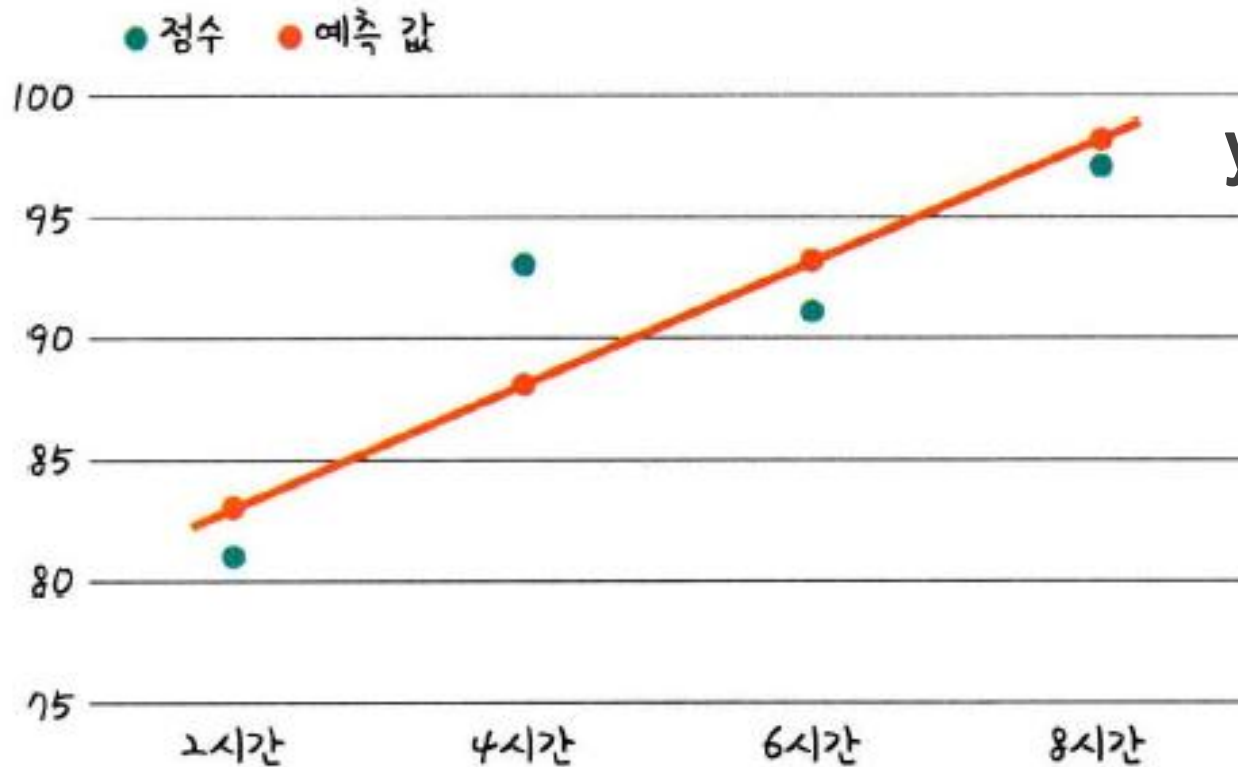
$$b = y \text{의 평균} - (x \text{의 평균} \times \text{기울기 } a)$$

# 최소 제곱법 (method of least squares)

**표 3-2**

최소 제곱법 공식으로  
구한 성적 예측 값

공부한 시간	2	4	6	8
성적	81	93	91	97
예측 값	83.6	88.2	92.8	97.4



$$y = 2.3x + 79$$

**그림 3-3**

오차가 최저가 되는  
직선의 완성

# ~실습 타임~

\* Google colab을 실행해주세요.



## 평균 제곱 오차(MSE)

◆ 임의의 직선 그려보기 ( $y = 3x + 76$ )

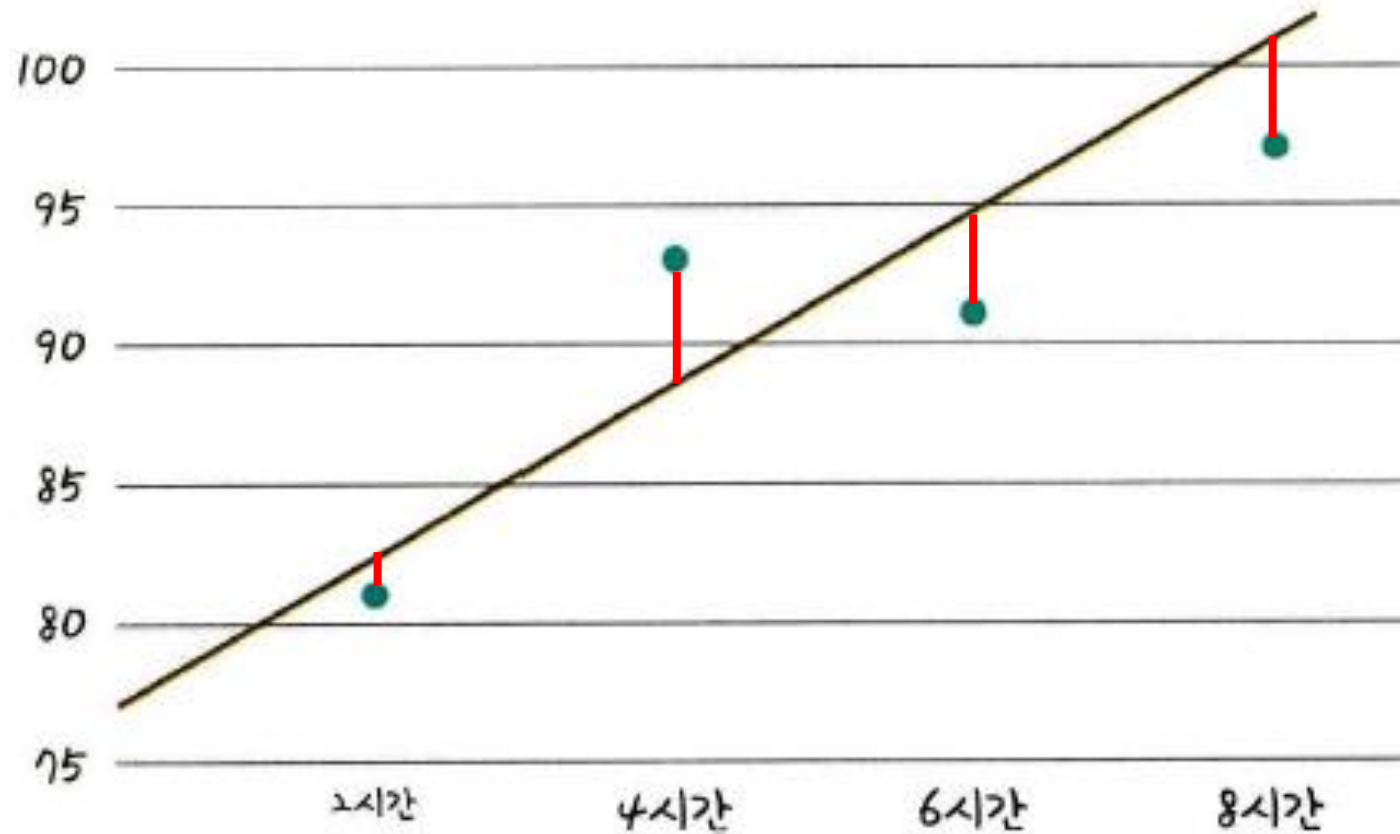


그림 3-5

임의의 직선 그려보기



## 평균 제곱 오차(MSE)

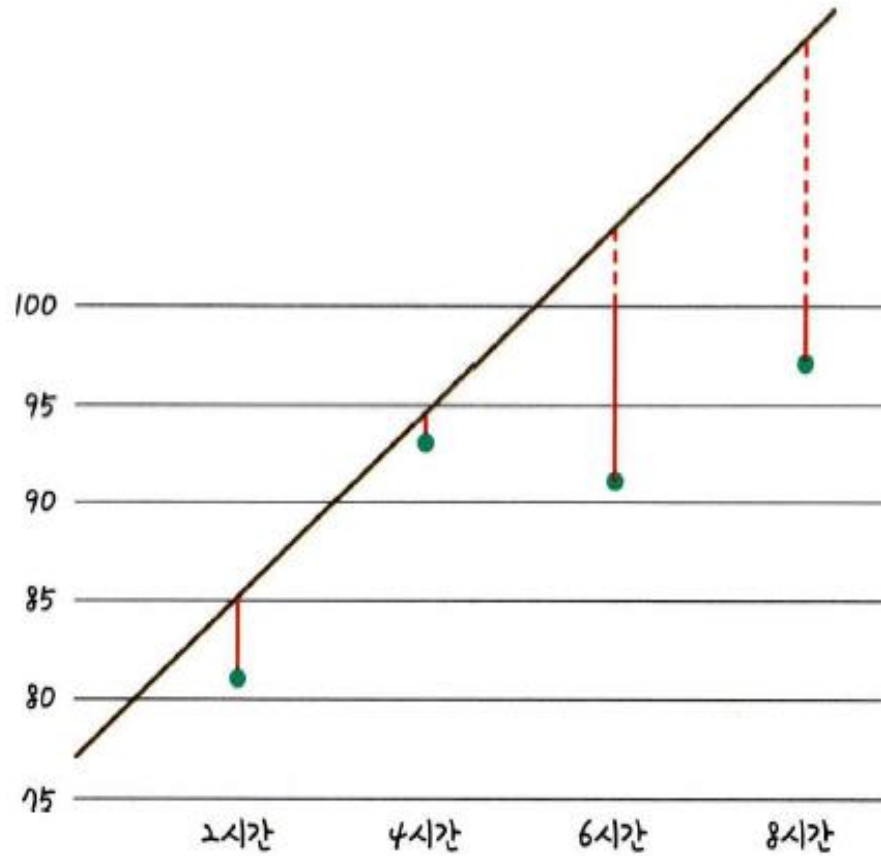
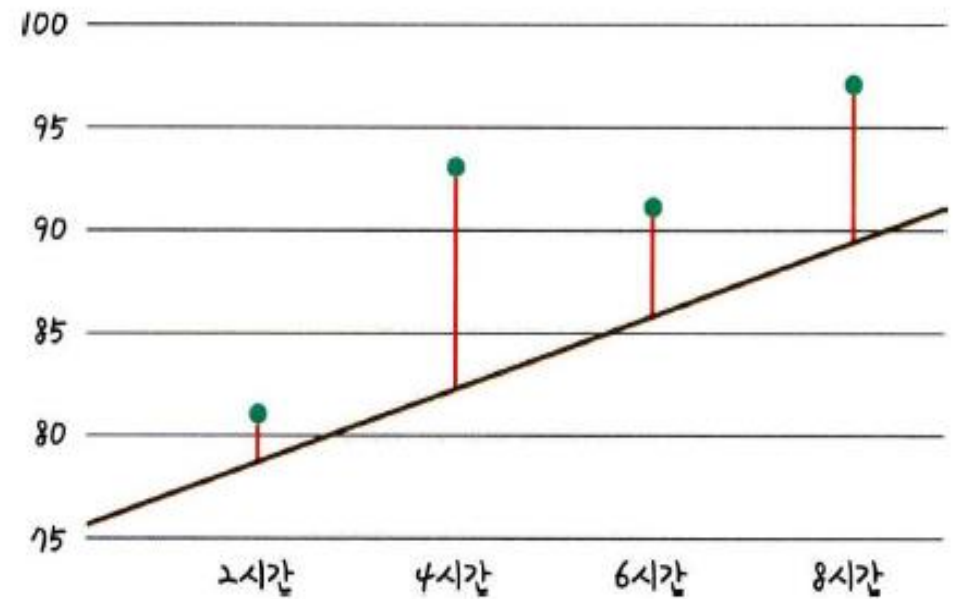


그림 3-7

기울기를 너무 크게  
잡았을 때의 오차

그림 3-8

기울기를 너무 작게  
잡았을 때의 오차



# 평균 제곱 오차(MSE)

**표 3-3**

주어진 데이터에서  
오차 구하기

공부한 시간(x)	2	4	6	8
성적(실제 값. y)	81	93	91	97
예측 값	82	88	94	100
오차	1	-5	3	3

$$\text{오차의 합} = \sum_i^n (y_i - \hat{y}_i)^2$$

$$\text{평균 제곱 오차(MSE)} = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$$

# ~실습 타임~

\* Google colab을 실행해주세요.



# ~질문 타임~

\*질문 환영



**1팀이었습시다.**  
**잘 들어주셔서 감사합니당!!**

