# Chapter 06.
# Regression

# 목차

# 1. 회귀분석 개요

- **회귀분석(Regression Analysis)**
  - 특정 변수(독립변수)가 다른 변수(종속변수)에 어떠한 영향을 미치는가 (인과관계 분석)
  - 예) 가격은 제품 만족도에 영향을 미치는가?
  - 한 변수의 값으로 다른 변수의 값 예언

**[참고] 인과관계(因果關係) : 변수A가 변수B의 값이 변하는 원인이 되는 관계(변수A : 독립변수, 변수B : 종속변수)**

- 상관관계분석 : 변수 간의 관련성 분석
- 회귀분석 : 변수 간의 인과관계 분석

# 1. 회귀분석 개요

## 【회귀분석 중요사항】

- '통계분석의 **꽃**' ➔ 가장 강력하고, 많이 이용
- 종속변수에 영향을 미치는 변수를 규명(변수 선형 관계 분석)
- 독립변수와 종속변수의 관련성 강도
- 독립변수의 변화에 따른 종속변수 변화 예측
- **회귀 방정식**($Y=a+\beta X$ → Y:종속변수, a:상수, β:회귀계수, X:독립변수)을 도출하여 회귀선 추정
- 독립변수와 종속변수가 모두 <u>등간척도 또는 비율척도</u> 구성

# 1. 회귀분석 개요

## 【회귀분석 중요사항】

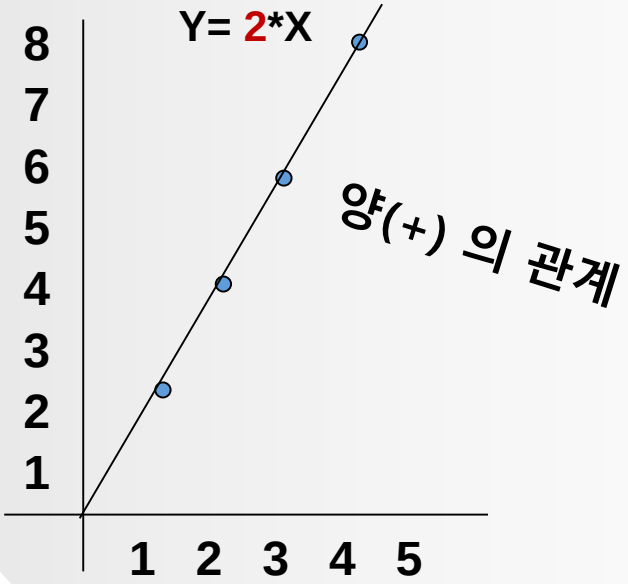
- '통계분석의 **꽃**' ➔ 가장 강력하고, 많이 이용
- 종속변수에 영향을 미치는 변수를 규명(변수 선형 관계 분석)
- 독립변수와 종속변수의 관련성 강도
- 독립변수의 변화에 따른 종속변수 변화 예측
- **회귀 방정식**(Y=a+βX → Y:종속변수, a:상수, β:회귀계수, X:독립변수)을 도출하여 회귀선 추정
- 독립변수와 종속변수가 모두 등간척도 또는 비율척도 구성

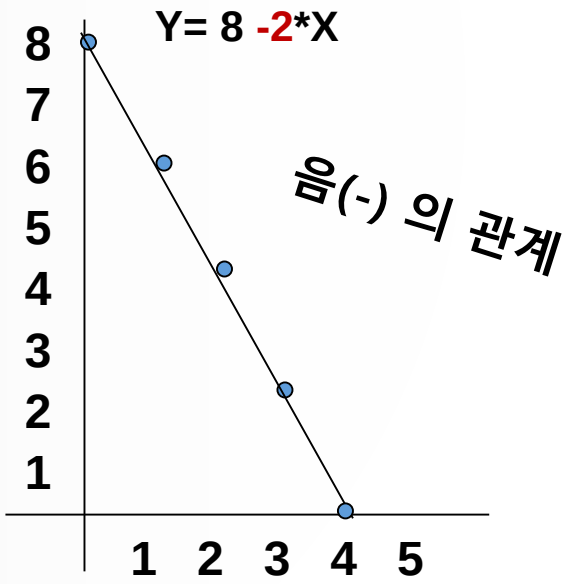
# 1. 회귀분석 개요

- **선형 회귀 방정식(1차 함수) : 회귀선 추정**

  Y= a + b*X

  (Y : 종속변수, a : 절편, b : 기울기, X : 독립변수)



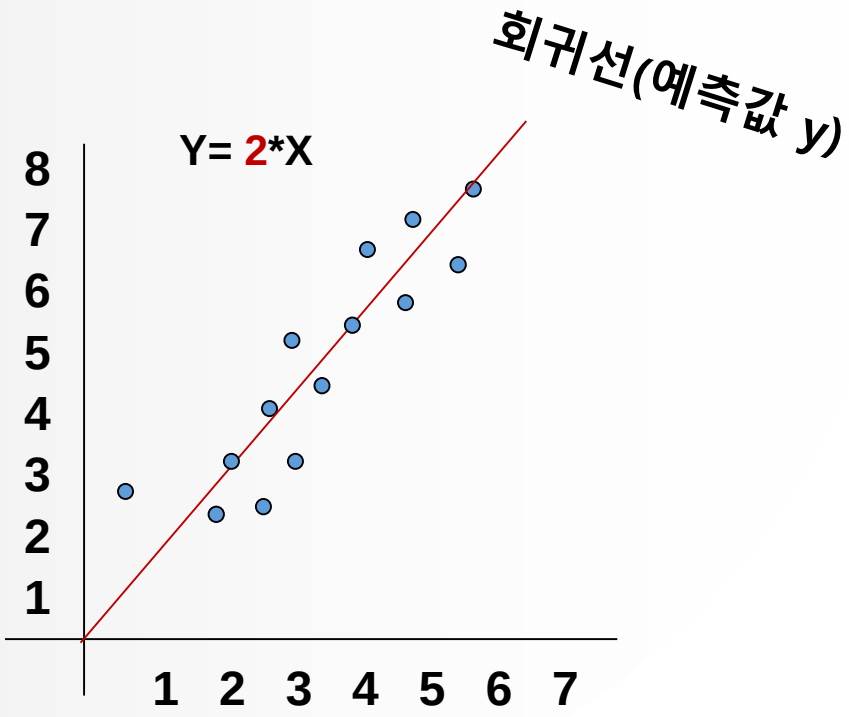

회귀방정식에 의해서 x가 10일 때 y는 20 예측 -> 회귀분석은 **수치 예측**

# 1. 회귀분석 개요

- **최소자승법 적용 회귀선**

  회귀방정식에 의해서 그려진 y의 추세선

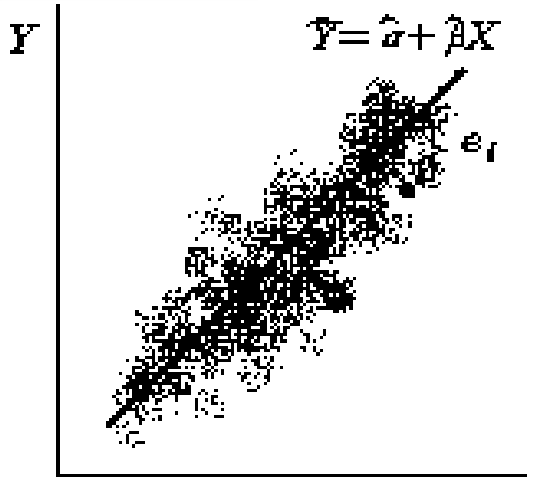
  산포도 각 점의 위치를 기준으로 정중앙 통과하는 회귀선 추정 방법



# 1. 회귀분석 개요

## 【회귀방정식】

- 회귀 방정식 -> 회귀선 추정
  - ✓ Y=a+βX : Y:종속변수, a:상수, β:회귀계수, X:독립변수
- 회귀계수(β) : 단위시간에 따라 변하는 양(기울기)이며, 회귀선을 추정함에 있어 최소자승법 이용
- 최소자승법 : 산포도에 위치한 각 점에서 회귀선에 수직으로 이르는 값의 제곱의 합 최소가 되는 선(정중앙을 통과하는 직선)을 최적의 회귀선으로 추정




산포도와 회귀선

# 2. 단순 회귀분석

## ● 단순 회귀분석

- ➢ 독립변수와 종속변수 각각 1개
- ➢ 독립변수가 종속변수에 미치는 인과관계 분석

### 【연구 가설】

**단순 회귀분석을 수행하기 위한 연구 가설은 다음과 같다.**

- 연구가설($H_1$) : 음료수 제품의 당도와 가격수준을 결정하는 제품 적절성(독립변수)은 제품 만족도(종속변수)에 **정(正)**의 영향을 미친다.
- 귀무가설($H_0$) : 음료수 제품의 당도와 가격수준을 결정하는 제품 적절성은 제품의 만족도에 영향을 미치지 않는다.

※ 논문에서는 **연구가설을 제시하고**, 귀무가설을 토대로 **가설 채택 또는 기각** 결정

# 3. 다중 회귀분석

## ● 다중 회귀분석

➢ 여러 개 독립변수가 1개의 종속변수에 미치는 영향 분석

## 【연구 가설】

**다중 회귀분석을 수행하기 위한 연구 가설은 다음과 같다.**

- 연구가설1($H_1$) : 음료수 제품의 적절성(**독립변수1**)은 제품 만족도(종속변수)에 정(正)의 영향을 미친다.
- 연구가설2($H_1$) : 음료수 제품의 친밀도(**독립변수2**)는 제품 만족도(종속변수)에 정(正)의 영향을 미친다.

# 4. 회귀분석 모델 생성

1. 데이터 셋 가져오기(보스턴 시 주택 가격 데이터 셋)

```
bosten = load_boston() # Load the data
print(bosten)
bosten_x = bosten.data # 4개 columns
bosten_y = bosten.target # Species
print(np.shape(bosten_x)) # (506, 13) : matrix
print(np.shape(bosten_y)) # (506,): vector

# x,y 모두 연속형 변수
print(bosten_x)
print(bosten_y)
```

# 4. 회귀분석 모델 생성

## 2. 훈련/검정 데이터 셋 생성

```
# 7:3 비율 train/test data set 구성
x_train,x_test, y_train, y_test =
train_test_split(
     bosten_x, bosten_y, test_size=0.3,
random_state=123) # seed값=123

print(x_train.shape) # (105, 4)
print(x_test.shape) # (45, 4)
```

# 4. 회귀분석 모델 생성

### 3. 모델 생성과 학습

```
# train data 학습
iris_model = LinearRegression()
iris_model.fit(x_train, y_train)

# The coefficients
print('Coefficients: ', iris_model.coef_)
#Coefficients:  [ 0.6473215   0.76805423 -
0.6799173 ] - x변수 기울기
```

# 4. 회귀분석 모델 생성

## 4. 모델 평가

```
# model 평가 방법
pred = iris_model.predict(x_test) # 예측치
Y = y_test # 관측치(정답)

# 1) 평균제곱근오차(mean square error)
print('MSE : %.3f'%mean_squared_error(Y, pred))

# 2) 상관계수 : 예측치와 관측치 이용 DF 생성
df = pd.DataFrame({'pred':pred, 'Y': Y})

# 상관계수 : 모델 평가
corr = df['pred'].corr(df['Y'])
print('corr : ', corr) #  0.881042498967
```

# 5. Logistic Regression

1) Logit 변환

2) Sigmoid Function

3) 이항 로지스틱 회귀모형

4) 다항 로지스틱 회귀모형

5) Sigmoid 활성 함수

# 1) 1. Logistic Regression

- **오즈비 vs 로짓변환**

  - ## 1. 오즈비(Odds ratio) : 0(실패)에 대한 1(성공)의 비율(0:no, 1:yes)
  - # no인 상태와 비교하여 yes가 얼마나 높은지 or 낮은지 정량화한 것
  - # odds_ratio = p(success) / 1-p(fail)
  - # p : y(반응변수)=1 이 나올 확률, 1-P : y(반응변수)=1의 여사건

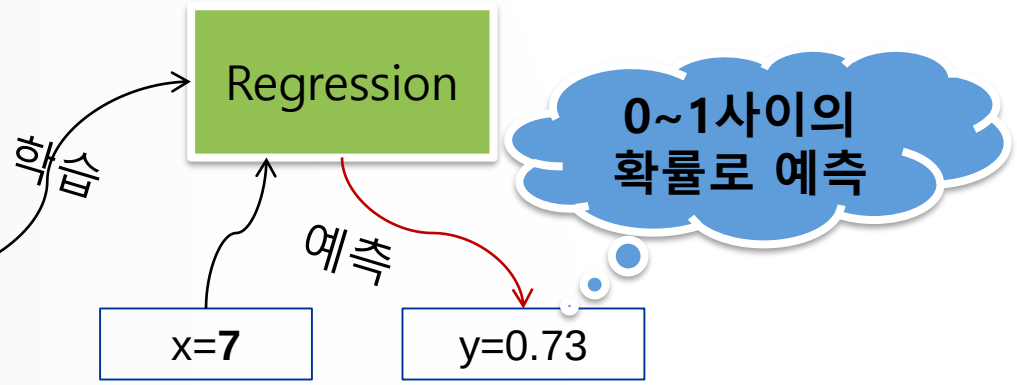
  - ## 2. 로짓변환 : 오즈비에 log 함수 적용
  - # logit = log(p / 1-p)

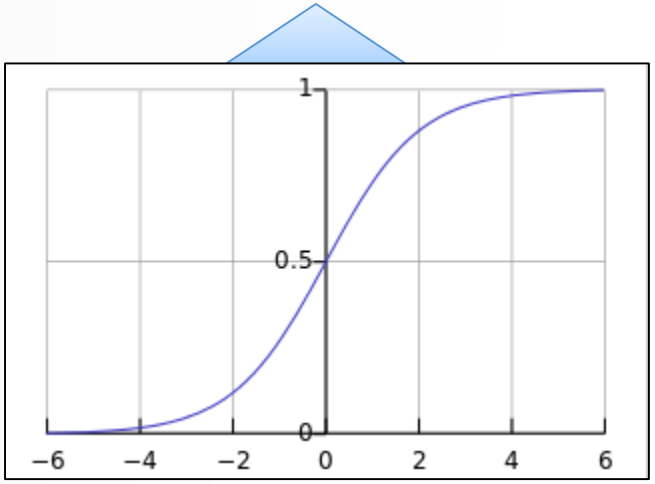
# 2) Logistic Regression

- **Sigmoid Function**
  - ➢ 합격/불합격 분류

| hours | score |
|---|---|
| 10 | pass |
| 9 | pass |
| 5 | fail |
| 3 | fail |

Training data set




**Sigmoid function**

# 3) Logistic Regression

- **이항 로지스틱 회귀모형**

```
#  로지스틱  회귀모델 생성 : 학습데이터
weater_model <- glm(RainTomorrow ~ ., data = train, family = 'binomial')
weater_model
summary(weater_model)

# 로지스틱  회귀모델 예측치 생성 : 검정데이터
# newdata=test : 새로운 데이터 셋, type="response" : 0~1 확률값으로 예측
pred <- predict(weater_model, newdata=test, type="response")
Pred
```

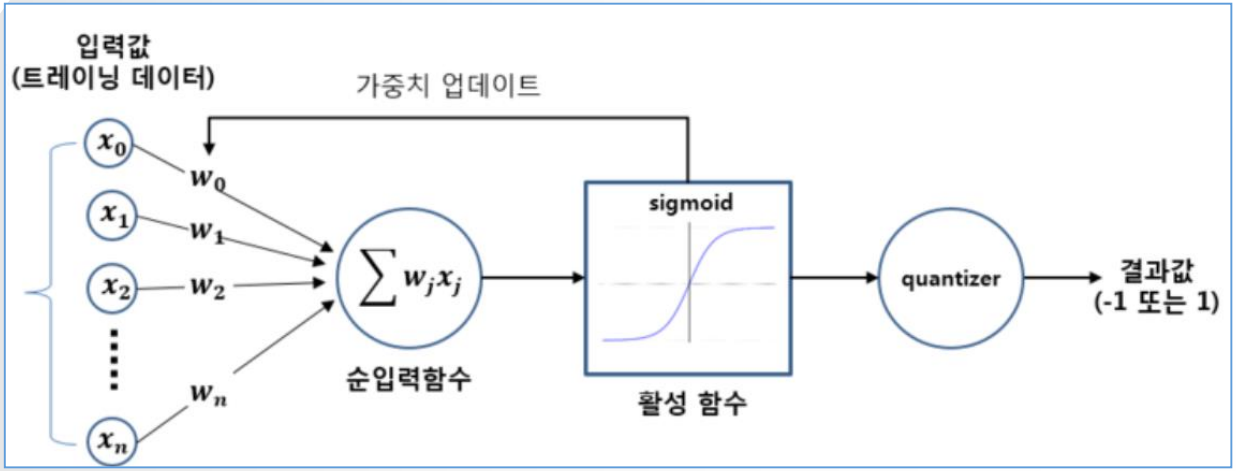
# 4) Logistic Regression

- **다항 로지스틱 회귀모형**

```
model <- multinom(Species ~ ., data = train)
fit <- model$fitted.values


# type='response' : 0~1 확률 예측 -> sigmoid 함수(yes/no)
# type='probs' : 0~1 확률 예측 -> softmax 함수(a, b, c)
pred_prob <- predict(model, newdata=test, type="probs")
pred_prob
```

# 5) Sigmoid 활성함수




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

def sigmoid(x):
    return 1 / (1 +np.exp(-x))

x = np.arange(-5.0, 5.0, 0.1)
y = sigmoid(x)
plt.plot(x, y)
plt.ylim(-0.1, 1.1)
plt.show()
```