

분석 : 요약, 정리

- 통계학
- 대수학(대수식) : 복잡한 일상현상을 하나의 수식으로 표현하고자 수학적 분야

1) 선형 대수식

2) 비선형 대수식

## 분석 : 요약, 정리

- 통계학기반
- 데이터 마이닝

숫자	문자열	시간	

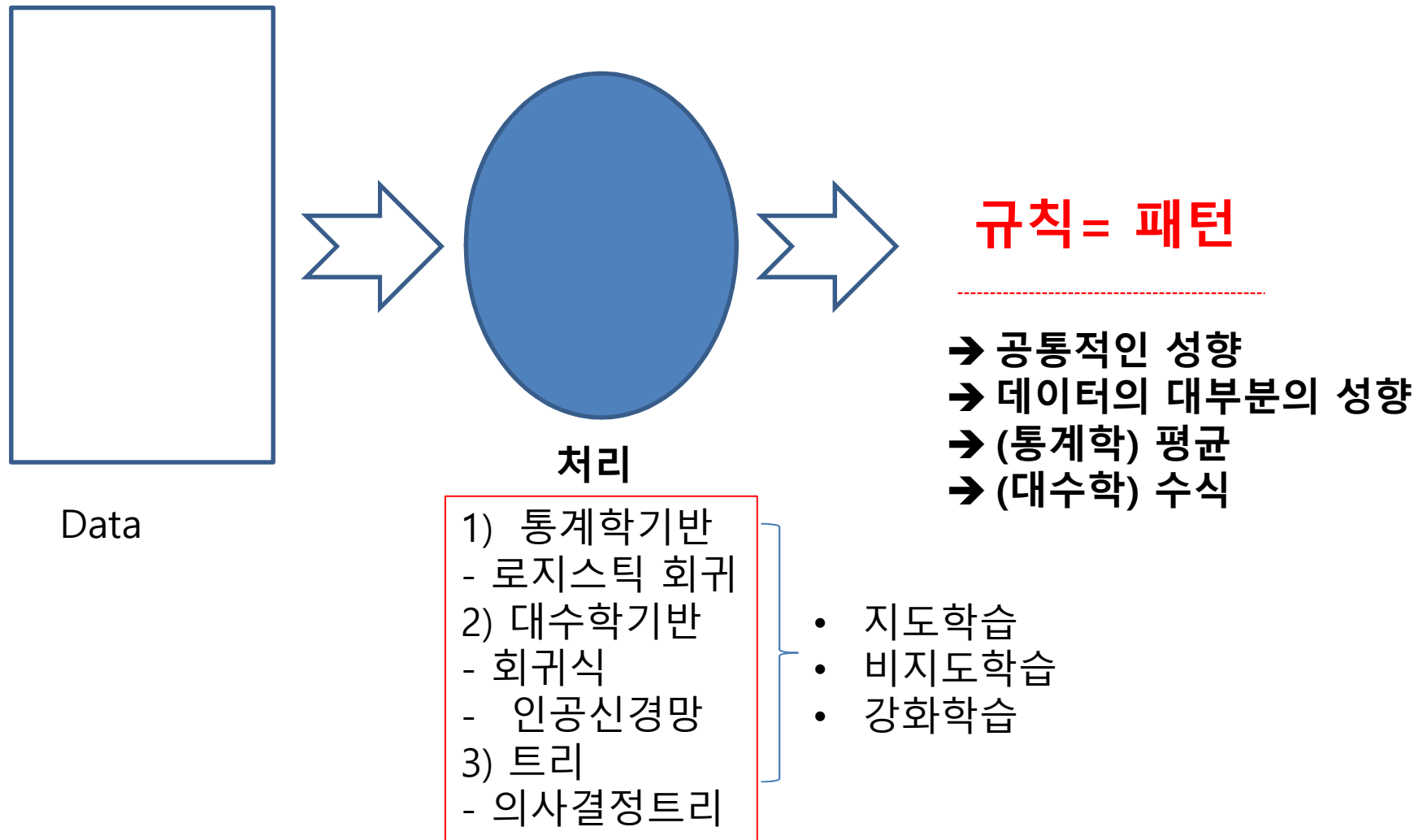
(회귀분석, 의사결정트리....)

인공지능 : 기계가 인간의 지능을 갖도록

- HOW 1) 프로그래밍(규칙기반, 지식기반)
- 2) 학습이론(경험을 누적 = 데이터)



## 머신러닝(Machine Learning)



# 회귀분석 & 로지스틱 회 귀분석

# 회귀분석

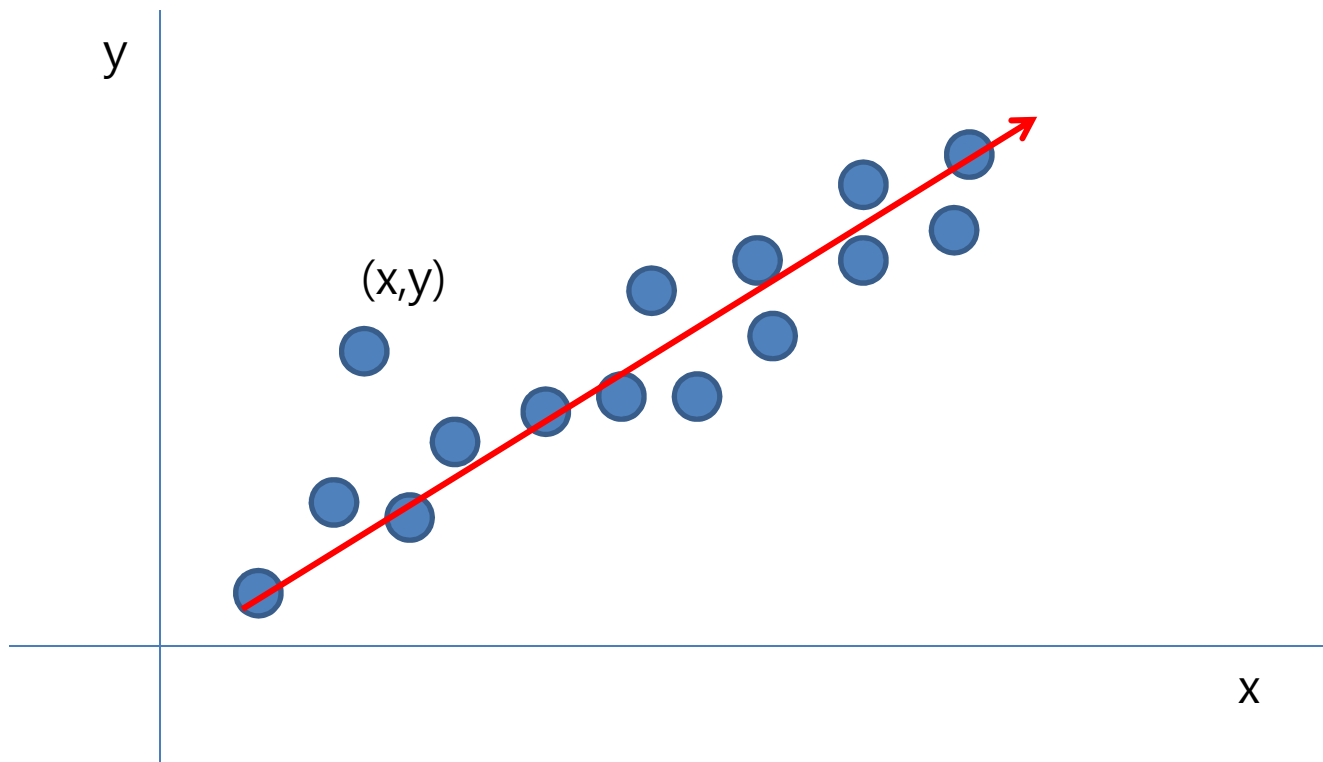
## 회귀분석(Regression) : 인과관계

→ 회귀식 → 선형대수식

$$Y(\text{종속변수}) = \frac{a * X(\text{독립변수}) + b}{}$$

다중 : 여러 개의 독립변수

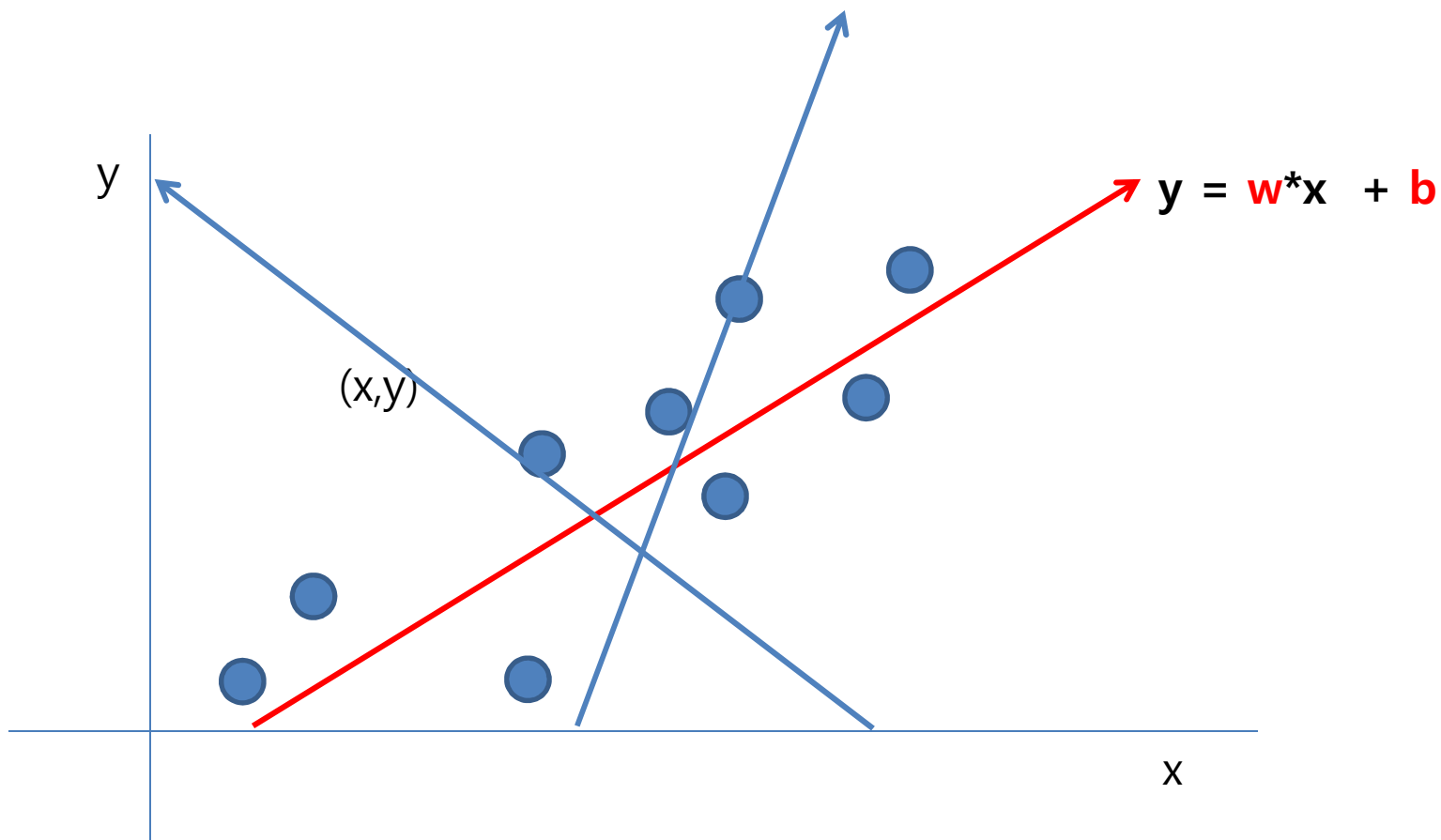
$$\text{다항} = X^2$$





회귀식  $y = \mathbf{w}^*x + \mathbf{b}$

최소 오차 제곱법





회귀식=방정식:  $y = w^*x + b$

최소 오차 제공법

•단순선형회귀식

$$\underline{y} = w^* \underline{x} + b$$

종속변수, 영향을받는변수

독립변수, 영향을주는변수

•다중회귀식

$$\underset{\text{주가}}{y} = w1^* \underset{\text{종가}}{x1} + w2^* \underset{\text{거래량.....}}{x2} \dots + b$$

•다항회귀식

$$y = w1^*x1^2 + b$$



## 회귀분석(Regression)

- 회귀분석의 가정 충족

(1) 선형성- 종속변수와 독립변수간의 선형관계이어야 한다.

(2) 잔차의 정규성 = 오차항의 기대값=0

- 회귀식이 정규분포를 이용한 개념이므로  $\sum$  잔차 = 0이어야 한다.

- $y(i) = a \cdot x(i) + b + \varepsilon_i$ : 정규성  $\sum \varepsilon_i = 0$  이어야 한다.  $\rightarrow E(\varepsilon_i) = 0$

(3) 잔차의 등분산성

- 잔차의 분산(변화량)이 독립변수에 따라 달라지면 안 된다.

- log변환, 가중최소제곱법을 사용해 이분산성을 해결

(4) 잔차의 독립성 : 오차항간의 상관관계가 없을 것

- 더빈-왓슨값

(5) 다중공선성: 독립변수간의 독립성

# 회귀분석(Regression)

## 0. 선형성(Linearity)

- 독립변수와 종속 변수가 선형적이어야 한다.
- 선형대수학에서 Linear 하다는 것은
  - superposition (additivity)와
  - Homogeneity를 만족하는 것을 의미한다

1) **superposition (additivity)**  $f(ax_1) = af(x_1)$

2) **Homogeneity** : 균질성  $f(a_1x_1 + a_2x_2) = a_1f(x_1) + a_2f(x_2)$

이 두 가지 조건을 합쳐서 표현하면 다음과 같다.

즉, 특정한 function/o  $f(x_1 + x_2) = f(x_1) + f(x_2)$  가진다고 말한다.

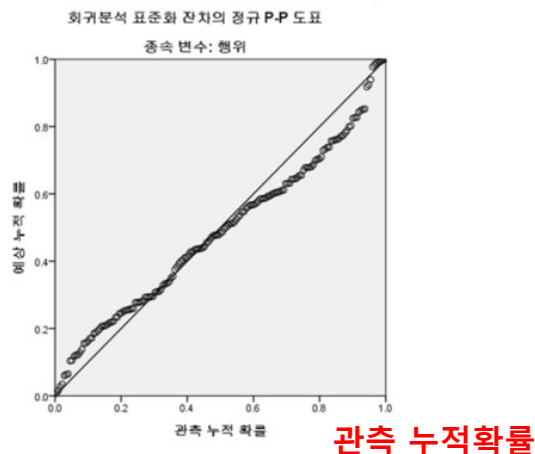
# 회귀분석(Regression)

## 1. 잔차의 정규성

오차항의 확률분포가 정규분포에서 많이 벗어나는 경우에는 수행한 결과를 해석하는데 신중해야 한다. 그 이유는 추정과 검정에서 사용되는 분포, 분포 등이 모두 정규분포로부터 파생된 확률분포들이기 때문이다.

오차항의 정규성은 잔차의 Normal Probability Plot (Q-Q plot)으로 검토할 수 있다. Normal Probability Plot은 잔차의 정규분포하에서의 기대값을 가로축으로 하고 실제 관찰된 residual을 세로축으로하여 그린 그림이다. 이 그림이 45도의 기울기를 가진 직선에 가까우면 가까울수록 오차항이 정규분포를 따른다고 볼 수 있다.

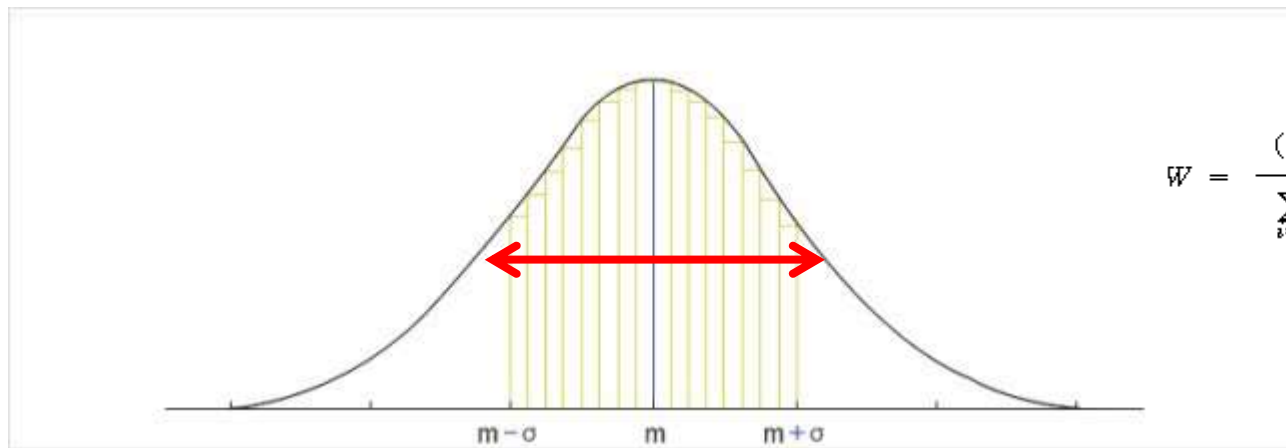
예상 누적확률



# 회귀분석(Regression)

## 1. 잔차의 정규성

- 잔차를 확률변수로 생각하자. 잔차의 기대값은 0이며 정규분포를 이루어야 한다.



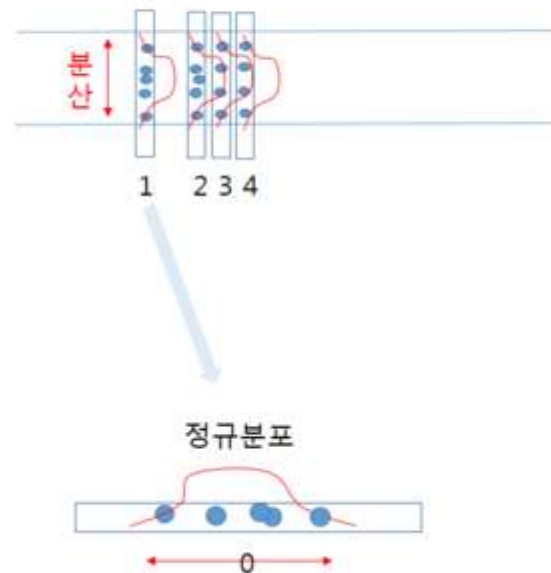
편차가 대칭이므로  $\sum$  오차항 =  $E(\text{오차항}) = 0$  되어야 한다.

- **shapiro.test()** : 단일 정규성 검정(shapiro-wilk검정방법)
  - ➔ 정규분포를 따르면 w값이 1에 가까워짐
- **qqnormal()** 함수 이용

# 회귀분석(Regression)

## 2. 잔차의 등분산성 검정과 해결

- 오차항의 분산은 모든 관찰치(독립변수)에서 일정해야 한다.



### 잔차의 등분산성

잔차값들이 0에 몰려있기는 하지만, 값이 퍼져있는 정도가 다르지 않고 비슷합니다.

### 잔차의 정규분포

1번에서 잔차들이 0에 몰려있고 파란색 선쪽으로 갈 수록 적어지는 것을 알 수 있습니다.

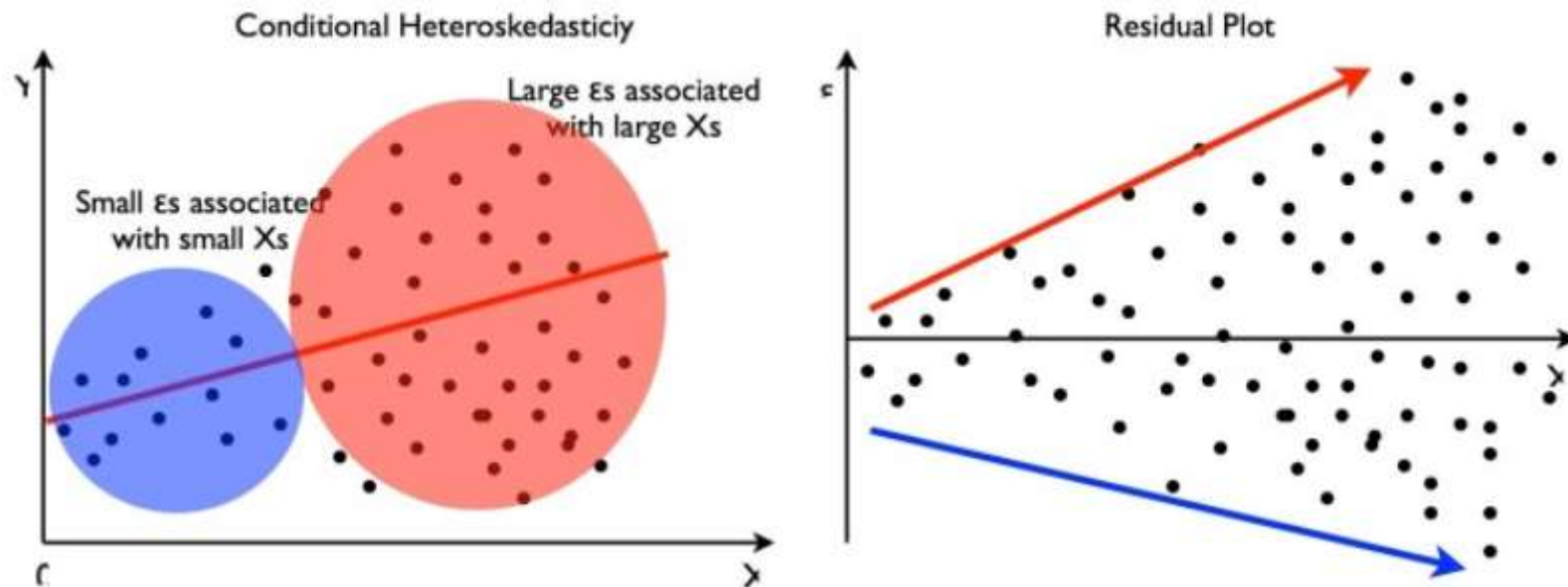
이것은 잔차가 정규분포를 따르기 때문에 0에 많은 값이 있고 0에서 멀어질 수록 값이 적어지는 것 입니다.

## 회귀분석(Regression)

### 2. 잔차의 등분산성 검정과 해결

- 오차항의 분산은 모든 관찰치(독립변수)에서 일정해야 한다.

(1) 잔차의 이분산성 : 독립변수와 오차항이 상관관계가 있다는 의미

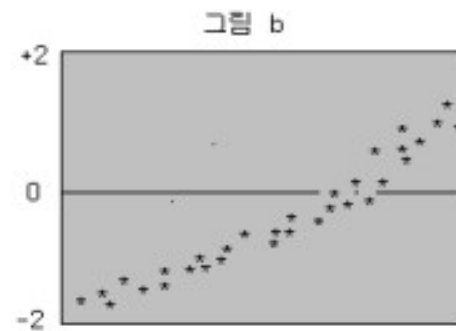
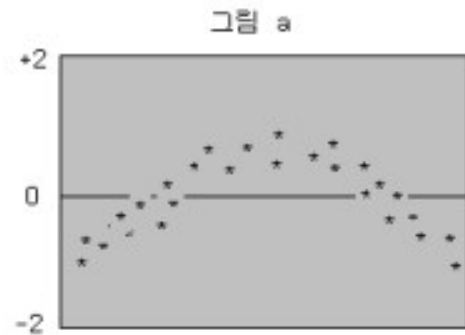


## 회귀분석(Regression)

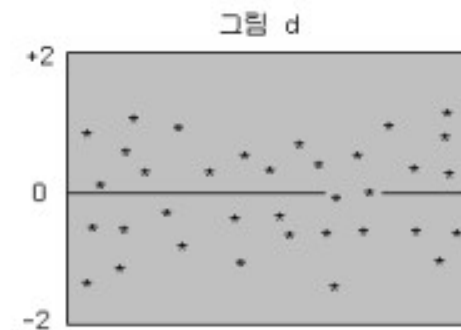
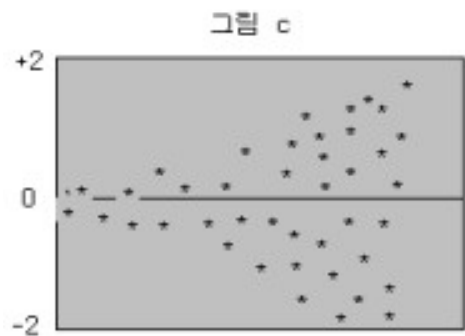
### 2. 잔차의 등분산성 검정과 해결

- 오차항의 분산은 모든 관찰치(독립변수)에서 일정해야 한다.

(1) 잔차의 이분산성 : 독립변수와 오차항이 상관관계가 있다는 의미



• 산점도 : 점들이 0을 기준으로 퍼진 정도를 확인



# 회귀분석(Regression)

## 2. 잔차의 등분산성 검정과 해결

### (1) 잔차의 이분산성 : 독립변수와 오차항이 상관관계가 있다는 의미

#### ① 그림 a

반응변수의 값이 증가함에 따라 잔차가 음의 값에서 양의 값으로, 다시 음의 값으로 변화하는 양상을 보이고 있다. 이러한 데이터는 오차항의 분산이 추정치와 멱함수 관계의 형태를 갖는 것으로서 선형성을 만족하지 못 한다고 볼 수 있다. 따라서 선형회귀직선보다는 2차식의 관계를 고려하는 것이 바람직하다.

#### ② 그림 b

수평축을 예측값이 아니고 관측된 순서로 하여 그린 것이다. 관측 순서에 따른 잔차의 변화가 이 그림에서와 같이 일정한 양상을 보이고 있으면 오차항이 서로 상관성을 갖고 있다고 할 수 있다.

#### ③ 그림 c

반응변수의 예측값이 증가함에 따라 잔차의 흩어진 폭도 넓어지는 경향을 보인다. 이는 예측값이 증가함에 따라 오차의 퍼진 정도가 증가함을 의미하므로 오차항의 등분산성에 대한 가정을 위반하는 대표적인 경우로서 이런 데이터는 분산 안정화 변환을 한 뒤 분석해야 한다. 예컨대 반응변수의 표준편차가 평균반응에 비례하는 경우에 로그변환을 사용할 수 있다.

#### ④ 그림 d

가정에 아무런 이상이 없는 경우를 나타낸다. 이 경우 특징은 대략 수평축 0을 기준으로 대칭적 분포를 보이며 잔차들이 일정한 패턴을 보여주지 않으며 랜덤하게 흩어진 모습을 하고 있다. 또한 독립변수의 값에 따른 잔차의 분포가 서로 비슷한 유형을 나타낸다.



## 회귀분석(Regression)

### 2. 잔차의 등분산성 검정과 해결

(1) 잔차의 이분산성 : 독립변수와 오차 항이 상관관계가 있다는 의미

- 오차항의 분산이 모든 관찰 값에서 동일한 상수(constant)를 가지는 것을 등분산성이라고 한다. 그런데 이분산성은 이 오차항의 분산이 관찰 값에 따라 달라지는 특징을 가진다. 수식으로 나타내면 다음과 같다.

• 등분산성일 경우  $E() = \text{기대값} = \text{확률적 평균}$

$$\begin{aligned} \text{var}(\varepsilon_i) &= E \left[ (\varepsilon_i - E(\varepsilon_i))^2 \right] \\ &= E(\varepsilon_i^2) = \sigma_{\varepsilon}^2 \end{aligned}$$

하나의 독립변수의 표본 추출된 값에 따라 오차의 분산 값이 변화되면 안 된다.

• 이분산성일 경우

$$E(\varepsilon_i^2) = \sigma_i^2$$

## 회귀분석(Regression)

### 2. 잔차의 등분산성 검정과 해결

#### (2) 잔차의이분산성이 회귀식에 미치는 영향

- ① 회귀추정으로 산출된 표준오차(standard error)의 신뢰성 낮음
- ② 추정 기울기계수  $\beta$  자체에 영향을 미치지 않는다.

따라서 여전히 불편 추정치를 만족한다.

- ③ 표준오차가 과소계상될 경우, 추정 회귀계수의  $t$  통계량이 과대평가되어 귀무가설을 기각하는 오류 발생(1종 오류발생)


- ④ F검정의 신뢰성도 낮아짐

$$\frac{\bar{X} - \mu}{s / \sqrt{n}}$$

$$F = \frac{\text{집단간분산}}{\text{집단내분산}}$$

표준오차가 과소되면 집단내분산은 작아지고, 집단간 분산값은 높아져서 F값이 본래의 값에 비해 커진다.

표준오차가 정상적으로 인정되지 않고, 패턴이 있는 상태가 되면 과소평가된다.



## 회귀분석(Regression)

### 3. 계열 상관성(Serial Correlation)의 검정과 해결

#### (1) 계열 상관성

- 오차간의 상관관계가 없어야 한다.
- 계열 상관성은 자기상관(autocorrelation)이라고도 하는데, 오차항들끼리 상관성이 존재하는 현상을 의미하며 횡단면 자료(Cross-sectional)가 아닌 시계열 자료(time-series)에서 주로 나타난다. 이는 완벽한 모형이라 할 수 있는 것은 없으니 오차항에서 이의 영향을 떠안을 수 밖에 없고 여러 기간에 걸쳐서 영향을 받게 되면 오차항 간의 일종의 상관성이 형성될 수 있기 때문이다.

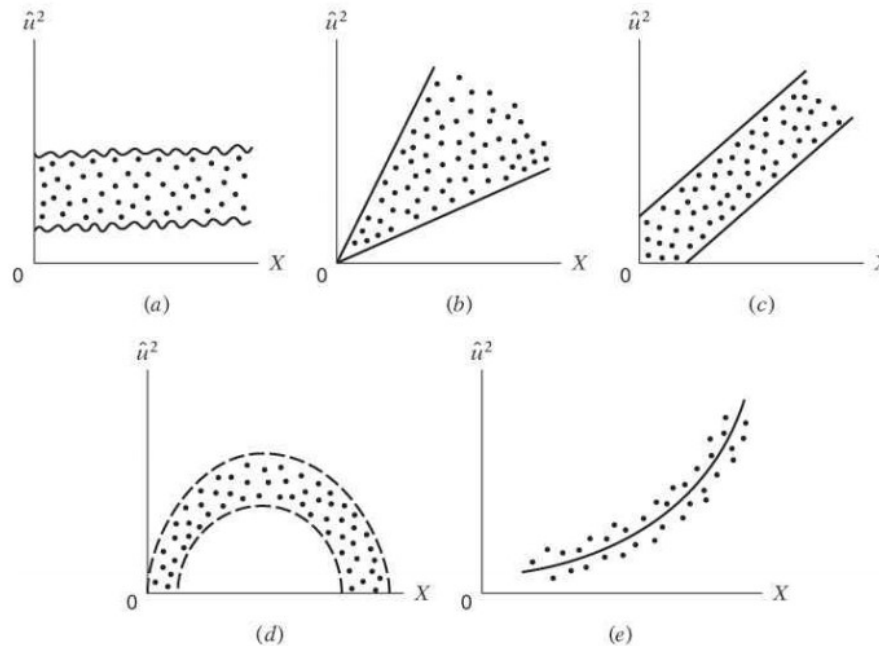
## 회귀분석(Regression)

### 3. 계열 상관성(Serial Correlation)의 검정과 해결

#### (1) 계열 상관성 = 잔차의 독립성

- 아래 (b)~(e)는 잔차들이 독립이 아닌 예이다.

'등분산이 아닌 것 or 일정 패턴'을 보인다면 잔차들끼리 독립이 아닌 것이다



## 회귀분석(Regression)

### 3. 계열 상관성(Serial Correlation)의 검정과 해결

#### (1) 계열 상관성

- 오차간의 상관관계가 없어야 한다.

$$\text{공분산 } \text{Cov}(\varepsilon_i, \varepsilon_{i+1}) = \frac{\sum_{i=1}^n (\varepsilon_i - \bar{\varepsilon})(\varepsilon_{i+1} - \bar{\varepsilon})}{n}$$

↓  
두 변량의 평균 변화량  
↓  
E(평균으로부터 떨어진 거리)

- 공분산  $\text{Cov}(x, y) = 0$  의 의미는 두 값이 평균에서 떨어지지 않았다는 의미
  - ➔ 그러므로 평균(즉 기대값=예측치)에 근접했다는 의미
  - ➔ 둘간의 상관관계가 없다는 의미
  - ➔ x를 이전 오차, y= 현재오차 라 할 수 있다.

## 회귀분석(Regression)

### 3. 계열상관성(Serial Correlation)의 검정과 해결

#### (1) 계열 상관성

- 앞 식에서 오차간의 상관관계가 없다!라는 것은

$\text{Cov}(\varepsilon_i, \varepsilon_{i+1}) = 0$  이라는 의미며,

- 반대로 오차간에 상관관계가 존재하면

$\text{Cov}(\varepsilon_i, \varepsilon_{i+1}) \neq 0$  가 된다.

- 즉, 모수에측시

$$E(\varepsilon_t \varepsilon_{t-k}) \neq 0, \text{ which } k=1, 2, 3 \dots$$

$$\varepsilon_t = \rho \varepsilon_{t-k} + \mu_t$$

상수가 아니라 계산식이 됨

## 회귀분석(Regression)

### 3. 계열 상관성(Serial Correlation)의 검정과 해결

#### (1) 계열 상관성

- 오차항간의 상관관계가 없을 경우

$$\begin{aligned} cov(\varepsilon_t, \varepsilon_{t-k}) &= E[\{\varepsilon_t - E(\varepsilon_t)\} \{\varepsilon_{t-k} - E(\varepsilon_{t-k})\}] \\ &\text{since } E(\varepsilon_t) = E(\varepsilon_{t-k}) = 0 \\ &= E(\varepsilon_t \varepsilon_{t-k}) = 0, \text{ which } k=1, 2, 3 \dots \end{aligned}$$

- 오차항간의 상관관계가 있을 경우

$$E(\varepsilon_t \varepsilon_{t-k}) \neq 0, \text{ which } k=1, 2, 3 \dots$$

$$\varepsilon_t = \rho \varepsilon_{t-k} + \mu_t$$

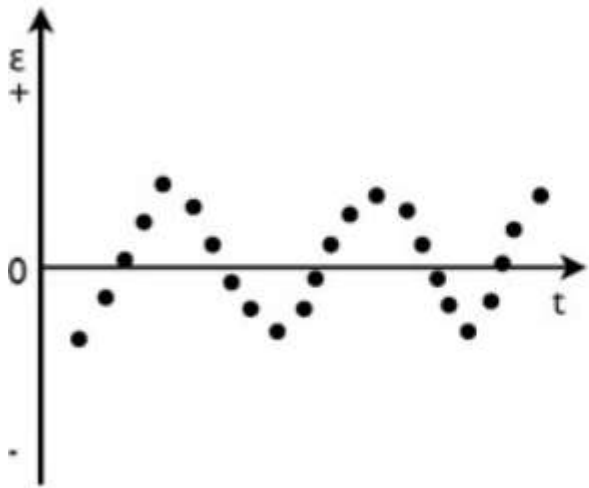
##시간이 갈수록  $0 < \rho < 1$ 의 양의 계열을 가짐

## 회귀분석(Regression)

### 3. 계열 상관성(Serial Correlation)의 검정과 해결

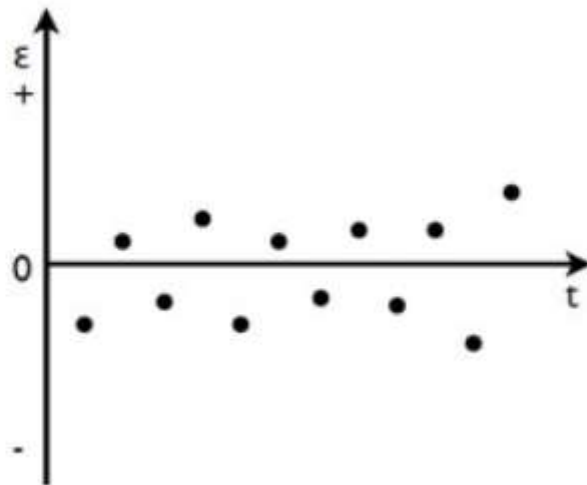
#### (2) 계열 상관성의 종류

- 양의 계열상관 & 음의 계열 상관



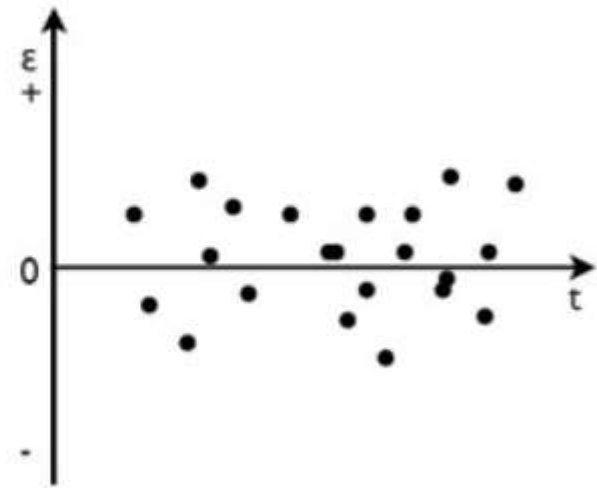
양의상관성

- 일정한 추세를 가짐
- $p(\text{상관계수}) = 1$ 이 된 상태로 오차가 전상태와 후상태와의 관계성이 등장함



음의 상관성

- (-)와 (+)를 왔다갔다
- $p(\text{상관계수}) = -1$ 이 되어 값의 부호가 전상태의 반대상태가 됨



상관 관계 없음



## 회귀분석(Regression)

### 3. 계열 상관성(Serial Correlation)의 검정과 해결

#### (3) 계열 상관성의 부작용

- ① 오차항 간의 연관성으로 표준오차의 신뢰도 낮아짐  
추정 기울기 계수의 유의성 저하됨
- ② MSE(Mean Squared Error)도 과소평가되므로, F 검정의 신뢰도 낮아짐  
 $0 < p < 1$  이 되므로

#### (4) 계열 상관성 확인 방법

- ① 오차항의 궤적(residual plot)을 통한 시각적인 방법
- ② 더빈-왓슨 검정법
- ③ LM 테스트

$$DW = \frac{\sum_{i=2}^n (e_i - e_{i-1})^2}{\sum_{i=1}^n e_i^2}$$

## 회귀분석(Regression)

### 3. 계열 상관성(Serial Correlation)의 검정과 해결

#### (4) 계열 상관성 확인


- 더빈-왓슨(Durbin-Watson)

$$d = \frac{\sum_{t=2}^t (\epsilon_t - \epsilon_{t-1})^2}{\sum_{t=1}^t \epsilon_t^2}$$

$\epsilon_t = \rho \epsilon_{t-k} + \mu_t$

계열 상관이 없는 경우, 앞선 오차  $e(t-k)$ 와  $e(t)$ 가 상관관계가 없으므로  $\rho=0$ 이 되며, 이때  $d \approx 2$ 에 근사한다.

- 양의 계열 상관  
 $d \approx 0$  ( $\because \rho=1$ )
- 음의 계열 상관  
 $d \approx 4$  ( $\because \rho=-1$ )
- 계열상관이 없을 경우  
 $d \approx 2$  ( $\because \rho=0$ )



## 회귀분석(Regression)

### 4. 다중 공선성의 검정과 해결

#### (1) 다중 공선성

- 독립변수간의 상관관계가 없어야 한다.

#### (2) 다중 공선성 확인 방법

- ① 기울기 계수의 낮은 통계적 유의성 + 유의한 F검정값 + 높은 결정계수
- ② 독립변수간 상관계수가 높은 것
- ③ 분산팽창요인(VIF : Variance Inflation Factors)

## 회귀분석(Regression)

### 4. 다중 공선성의 검정과 해결

#### (2) 다중 공선성 확인 방법

##### ① 기울기 계수의 낮은 통계적 유의성 + 유의한 F검정값 + 높은 결정계수

- 다중 공선성을 의심할 수 있는 가장 기초적이지만 효율적인 방법
- 기울기계수와 t값과 p-value는 해당 기울기계수가 얼마나 유의한지를 보여준다. 만약 해당 다중회귀모형에서 상관성 높은 독립변수들이 포함되어 있다면, 종속 변수의 변동을 설명함에 있어 그 설명력을 나눠서 설명해야 할 것이다. 즉, 설명력이 분산된다는 것이다. 따라서 개별 기울기 계수의 유의성(t값)은 낮게 나올 것이다. 하지만 전체 모형상으로는 잘 설명할 수 있으니 F 검정값 및 결정 계수 ( $R^2$ )은 높게 나오게 된다.

## 회귀분석(Regression)

### 4. 다중 공선성의 검정과 해결

#### (2) 다중 공선성 확인 방법

##### ② 독립변수간 상관계수가 높은 것

- 높은 상관계수의 기준은 0.7 이상이면 다중 공선성을 의심

##### ③ 분산팽창요인(VIF : Variance Inflation Factors)

- VIF는 다중 공선성이 추정 기울기 계수의 표준오차를 얼마나 증가시켰는지를 측정하는 지표인데, 문자 그대로 해석하면 "분산을 증가시키는 요소"라는 의미이다.

$$\widehat{var}(\beta_j) = \frac{s^2}{(n-1) var(X_j)} \times \frac{1}{1-R_a^2}$$

$$VIF(\hat{\beta}_i) = \frac{1}{(1-R_a^2)}$$

## 회귀분석(Regression)

### 4. 다중 공선성의 검정과 해결

#### (3) VIF

$$VIF(\hat{\beta}_i) = \frac{1}{(1 - R_a^2)} \quad k=1,2,3,\dots,p(\text{설명변수의 개수})$$

- 결정 계수( $R^2$ )가 커질수록 VIF의 계수값도 커진다.  $1 - R^2$ 이기 때문이다.( $R$ 은 소수점이므로 역수의 형태로 분모에 존재하므로.)
- 전혀 관계가 없는 경우 결정 계수  $R^2=0$ 에 가까워지므로  $VIF \approx 1$ (1에 가까워진다.)
- 이에 따라  $5 < VIF < 10$ 이면 다중공선성 의심  
 $VIF > 10$ 이면 다중공선성 심각
- VIF 값이 클수록 다중 공선성의 원인이 되는 변수로 판단해 제거 대상이 된다.

## 회귀분석(Regression)

### 5. 회귀분석에서의 설명력(결정계수, $R^2$ , R-square)

#### (1) 최소 제곱법

-데이터의 잔차(residual)의 제곱의 합이 최소화되는 공식을 도출

## 회귀분석(Regression)

### 5. 회귀분석에서의 설명력(결정계수, $R^2$ , R-square)

#### (2) 결정계수

-회귀모형은 데이터를 설명할 수 있는 부분과 설명할 수 없는 부분(잔차)으로 나뉘어진다. 이를 각각 제곱합(Sum of Square)로 나타낸다.

-여기서 회귀제곱합(Regression sum of Square = SSR)을 전체 제곱합(Total Sum of Square=SST)로 나누어주면 전체 모형이 차지하는 부분에서 설명 가능한 부분의 비중을 알 수 있게 된다.

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$



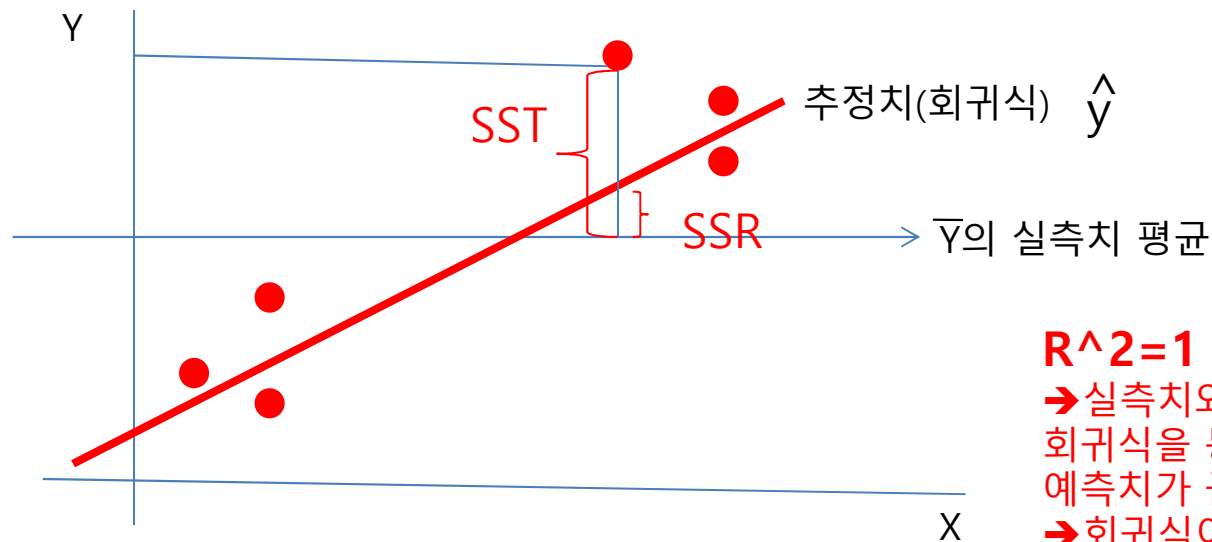
# 회귀분석(Regression)

## 5. 회귀분석에서의 설명력(결정계수, $R^2$ , R-square)

- 결정계수  $R^2$

## 총 변동량에 대비해  
회귀모형의 변동량을 설명

$$R^2 = \frac{\text{설명된변화량}}{\text{총변화량}} = \frac{SS_R}{SS_T} = \frac{\sum (\hat{Y}_i - \bar{Y})^2}{\sum (Y_i - \bar{Y})^2} = \frac{(\text{추정치편차})^2}{(\text{실제편차})^2} = \frac{SSR}{SST}$$



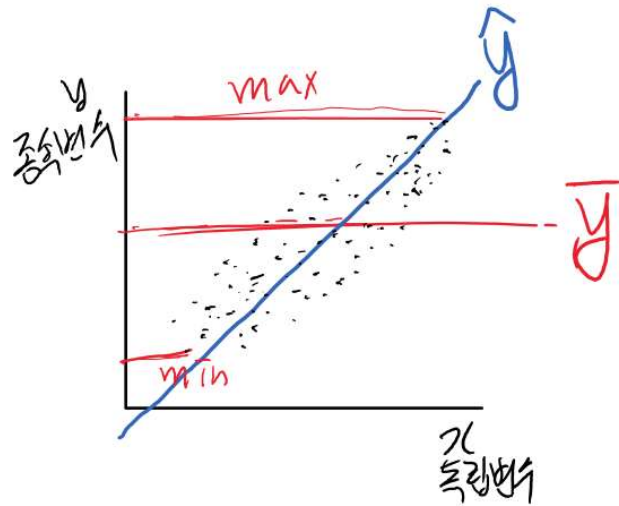
**$R^2=1$**

→ 실측치와  
회귀식을 통한  
예측치가 근접해있다.  
→ 회귀식이 실측치에 대한  
설명력이 좋다.

## 회귀분석(Regression)

### 5. 회귀분석에서의 설명력(결정계수, $R^2$ , R-square)

#### (2) 결정계수



- SST: Sum of Square Total  
- 편차의 제곱합
- SSE : Sum of Square Error  
- 회귀식과 실제 값의 차이를 의미
- SSR : Sum of Square Regression  
- 회귀식과 평균값의 차이

$$SST = \sum_{i=1}^n (y_i - \hat{y}_i + \hat{y}_i - \bar{y})^2$$

$$SSE = \sum (y_i - \hat{y}_i)^2$$

$$SSR = \sum (\hat{y}_i - \bar{y})^2$$

$$SST = SSE + SSR$$

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$



## •가변 변수(dummy variable)

0과 1로 표현된 데이터

회귀분석 시에는 범주형데이터를 → 연속형데이터

	Speciesversicolor	Speciesvirginica
Setosa	0	0
Versicolor	1	0
verginica	0	1

	Setosa		
Setosa	1	0	0
Versicolor	0	1	0
verginica	0	0	1



## • 다양한 형태의 회귀식(1함수)

$$y = x^2 + 3x + 4$$

$$y = 3 \cdot (\underline{x_1 + x_2})$$

$$\neq 2x_1 + 3x_2$$

## 회귀분석(Regression)

```
m <- lm(dist~speed, data=cars)
summary(m) #분석모형 = m
predict(m,newdata=data.frame(speed=4), interval = "confidence")
```

### **#모델평가 (분산분석)**

```
summary(m)
full <- lm(dist~speed,data=cars) #집단A
reduced <- lm(dist~1,data=cars) #집단B
anova(reduced,full)
```

### **##모델진단-가정충족여부 판단**

```
par(mfrow=c(2,2),mar=c(3,3,3,3))
plot(m)
```

### **#회귀직선 시각화**

```
plot(cars$speed,cars$dist)
abline(coef(m))
```

## 회귀분석(Regression)

### #이상치 탐색

#student 잔차(개별잔차를 잔차의 표준편차로 나눈 값)

`rstudent(m)` #t-test사용(너무 크거나, 너무 작거나)

`##install.packages("car")`

`library(car)`

`outlierTest(m)` ##0.05보다 작은 값이 이상치로 판단

### #잔차의 독립성 평가

`install.packages("lmtest")`

`library(lmtest)`

`dwtest(m)` #더빈 왓슨  $d=2$ ( $p=0$ :상관 관계 없음), 보통  $d=1\sim3$  정상판단

## 회귀분석(Regression)

### 6. 변수선택방법: 후진제거법 & 전진선택법 & 단계적 방법

#### (2) information criterion (정보지수 )

- AIC(Akaike information criterion) 나 BIC(Bayesian information Criterion)의 기준으로 평가

- ①Akaike 정보 지수 ( Akaike information criterion : AIC )
- ②Bayesian 정보 지수 ( Bayesian information criterion : BIC)
- ③내재되지 않은 두 모형의 **로그 우도 함수 값의 차이**

$$-2\ln L_A - (-2\ln L_B)$$

는  $\chi^2$ (카이제곱)검정(우도비 검정)이 불가능하다.

➔ 따라서 정보지수(information criterion)을 이용한다.

## 회귀분석(Regression)

### 6. 변수선택방법: 후진제거법 & 전진선택법 & 단계적 방법

#### (2) information criterion (정보지수)

$$AIC = -2 \ln L + 2q$$

Log(likelihood): 0~1 값을 가짐

$\ln L$

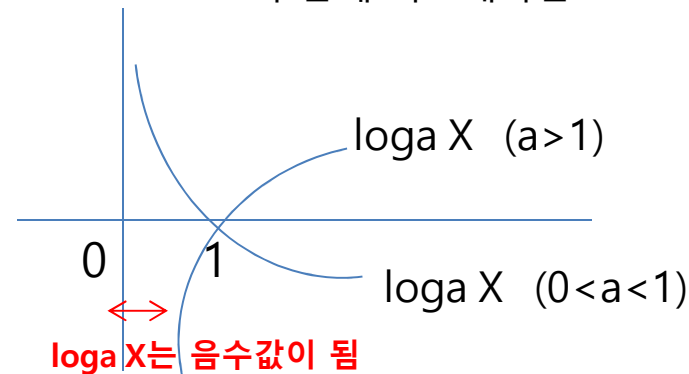
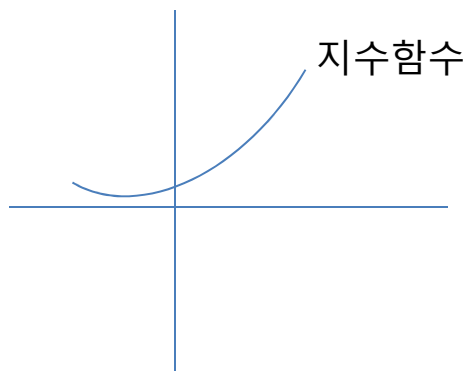
likelihood : 가능도함수  
→ 조건부확률과 유사

•  $q$ : 모형의 미지수(모수의 특성 개수)  
→ 특성변수가 많아지면 별점 부과하는 듯한 효과

• 낮을 수록 좋다.

단, 표본 크기가 커지면 AIC 값이 커져  
BIC와 함께 비교해야함

$-2 \ln L$  는 모형의 적합도를 의미 작을수록 유리함





## 회귀분석(Regression)

- **확률(probability) & 우도(likelihood)**

- 확률은 확실한 비율을 나타낸다. 동전의 앞면이 나올 확률, 주사위가 1이 나올 확률 ( 동전이나 주사위는 이상적으로 만들어졌다고 가능, 찌그러졌거나, 가공된 경우 제외)
- 통계는 데이터를 중심으로 비율(또는 사실)을 추정함. 즉, 동전이나 주사위가 이상적 (균형)으로 만들어지지 않는 경우 앞면이 나올 확률, 주사위가 1이 나올 확률은 각각  $1/2$ ,  $1/6$  이 아니고, 시행을 거친 후 확률을 추론해야 한다.
- 논리학관점으로 보면, 확률은 연역(사전에 알 수 있음), 통계는 귀납적(이후에 알 수 있음)이라고 할 수 있다.

### **likelihood vs probability**

- 사전적으로는 likelihood와 probability는 같다
- 단, 확률/통계학에서만 그 차이를 구분한다고 한다.
- likelihood는 통계에 의한 추론(데이터 중심)
- probability는 확률(모수 중심)

## 회귀분석(Regression)

### 6. 변수선택방법: 후진제거법 & 전진제거법 & 단계적 방법

- **BIC** : 여러 가지 경쟁 모형중에서 BIC의 절대 값이 작은 모형 선택

범위	모형차이
$0 \leq BIC < 2$	작은 차이
$2 \leq BIC < 6$	보통 차이
$6 \leq BIC < 10$	큰 차이
$10 \leq BIC$	매우 큰 차이

$$BIC = -G + (df)(\ln N)$$

변수의 수에 대해 벌점에 가중치를 높게 준다

## 회귀분석(Regression)

6. 변수선택방법: 후진제거법 & 전진제거법 & 단계적 방법

예)

`m<-lm(medv ~. , data=BostonHousing)` #medv : 보스턴 집값

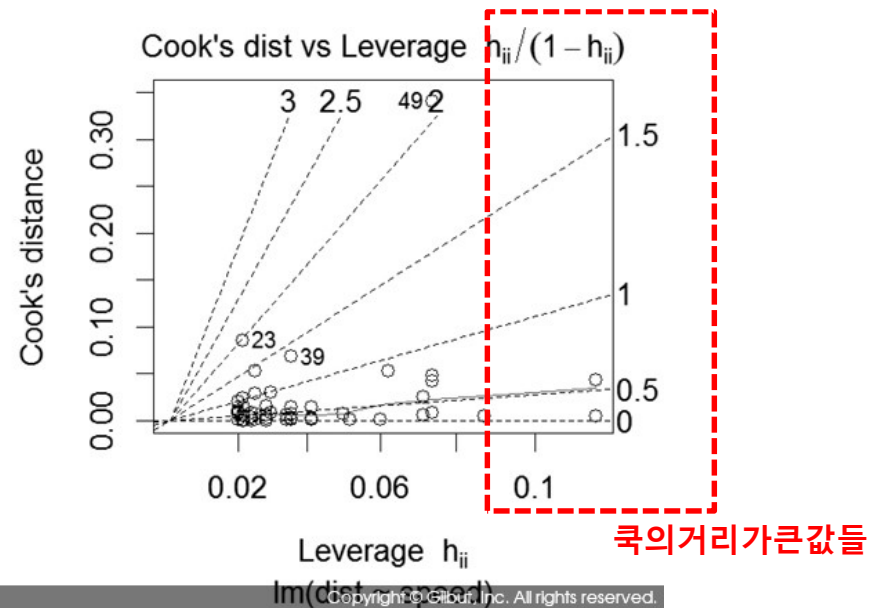
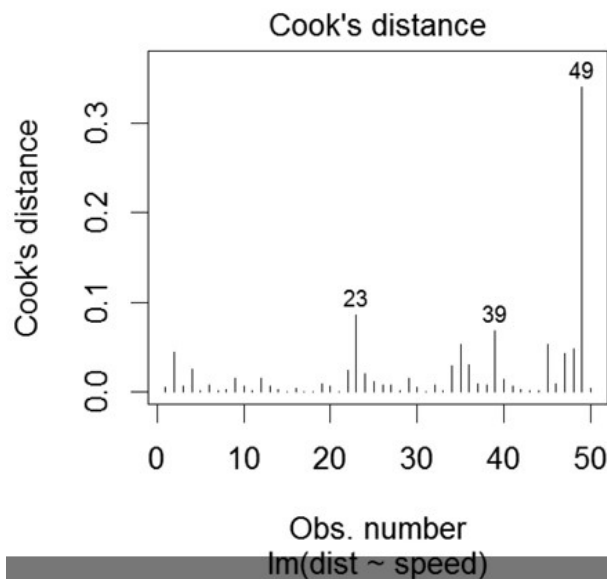
`step(m, direction="both")` ###forward, backward

$$\text{수정 결정계수(Adjusted } R^2\text{)} : R^2_{\text{adj}} = 1 - \left[ \frac{n-1}{n-(p+1)} \right] \frac{SSE}{SST} \leq 1 - \frac{SSE}{SST} = R^2$$

## 회귀분석(Regression)

### 7. 쿡의 거리

- 회귀 직선의 모양(기울기나 절편 등)에 크게 영향을 끼치는 점들을 찾는 방법으로 쿡의 거리는 레버리지와 잔차에 비례하므로 두 값이 큰 우측 상단과 우측 하단에 쿡의 거리가 큰 값들이 위치한다. → 평균에 가까운 데이터는 영향이 없으며, 영향(Leverage)이 큰 것은 이상치일 가능성이 있다. 극단치(잔차가 큰것)를 찾을 수 있도록 제공하는 것이 쿡의 거리이다.



# 로지스틱 회귀

# Linear Regression

선형 회귀분석은 **최소제곱법**을 기준으로 회귀선을 찾는다. 최소제곱법을 설명하기 위해서는 **잔차**라는 개념을 먼저 알아야 한다.

**잔차(residual)**란 관측값의  $y$ 와 예측값의  $y$  간의 차이를 말하며, 보통  $e$ 로 표기한다.

예를 들어  $A(1, 4)$ 과  $B(2, 3)$ 라는 2개의 점이 있다고 하자. 그리고 회귀식이  $y = 2x + 1$  이라면 점  $A$ 의 관측값  $y$ 는 4, 예측값  $y$ 는 3이고 점  $B$ 의 관측값  $y$ 는 3, 예측값  $y$ 는 5이다. 이때  $A$ 의 잔차는  $4 - 3 = 1$ 이고  $B$ 의 잔차는  $3 - 5 = -2$ 이다.

최소제곱법은 잔차의 제곱의 합이 최소가 되도록 하는 직선을 회귀선으로 한다는 것을 의미한다. 회귀선과 실제 관측값 사이의 제곱, 즉 잔차의 제곱의 합이 최소가 되도록 회귀계수를 구하는 것이다. 그렇다면 위의 예에서 잔차의 제곱의 합은 5이다.

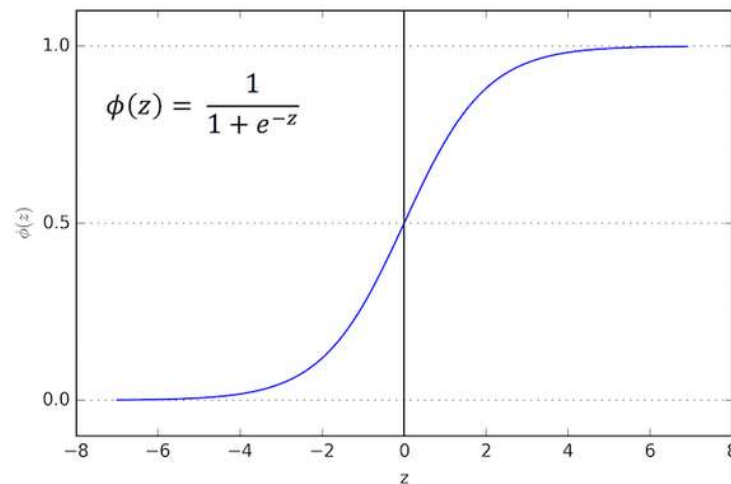


# Logistic Regression

$x[i, ]$  가 입력 열일때 로지스틱 회귀는 다음과 같은 적합한 함수를 찾는다.

$$P(y[j] \text{ in class}) = f(x[i, ]) = s(a + b[1]x[i,1] + \dots + b[n]x[i,n])$$

여기서  $s(Z)$ 는 시그모이드 함수로  $S(Z) = 1/(1 + \exp(-Z))$  로 정의



만약  $y[j]$ 가 확률이고,  $x[i,]$ 가 관심영역에 속한다면(예:비행기가 지연될 것인지)  
 $f(x[i,])$ 가  $y[j]$ 의 최상의 추정치가 되도록  $b[1].....[n]$ 을 찾아낼 것이다.



## Logistic Regression

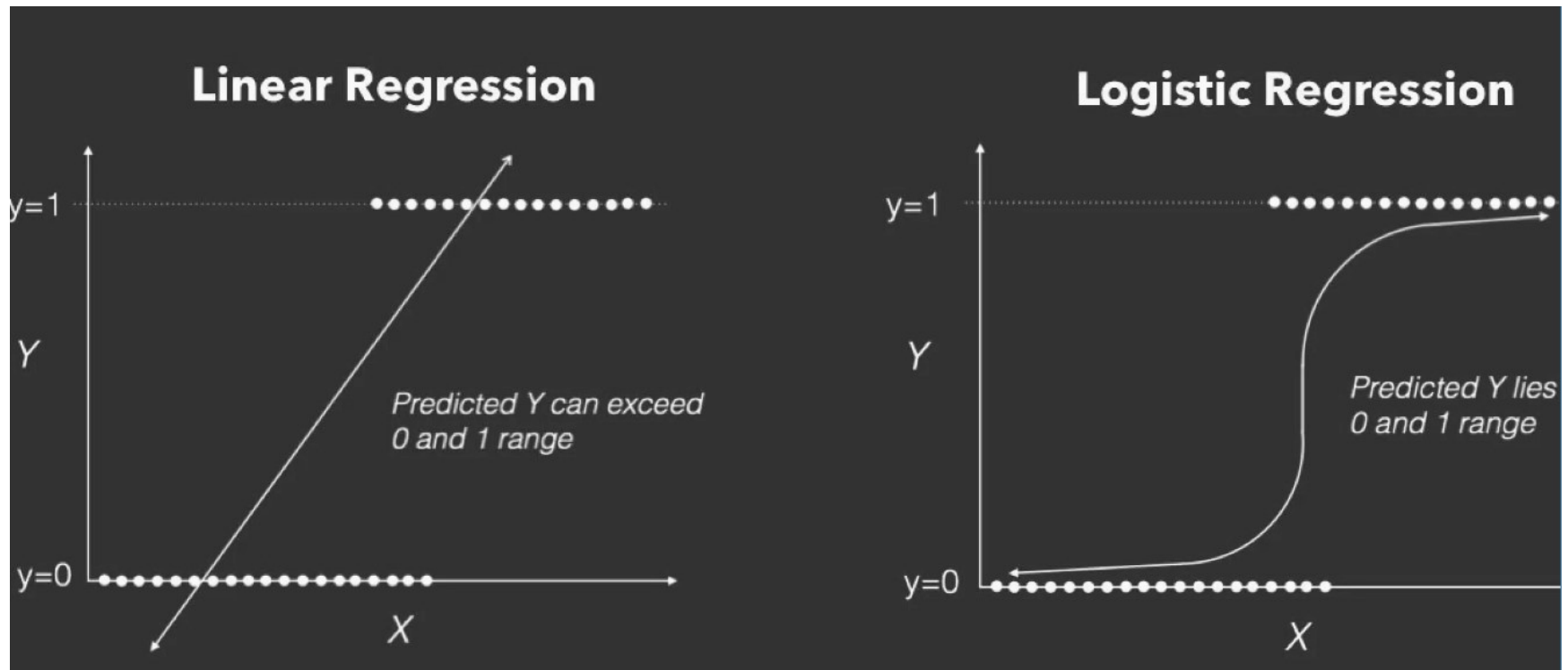
$$\log \frac{p}{1-p} = a + b[1] * x[i,1] + \dots + b[n] * x[i,n]$$

$$\mathbf{z} = \log \frac{p}{1-p} \quad (\text{로짓함수라고 한다})$$



# Logistic Regression

선형회귀와 로지스틱회귀



# Logistic Regression 실습

- 예제 코드

## (1) 데이터 탐색

GRE, GPA(내신), RANK이 입학(admission)에 어떤 영향을 주는지 로지스틱 회귀분석을 통해 분석한다.

```
library(aod)
library(ggplot2)

# view the first few rows of the data
mydata <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
head(mydata) # 데이터의 대략적인 분포 확인
summary(mydata) # 데이터 구조 확인
str(mydata) # 변수별 표준편차 확인
sapply(mydata, sd) # contingency table : xtabs(~ x + y, data)
xtabs(~admit+rank, data=mydata)
```

# Logistic Regression 실습

- 예제 코드

## (1) 데이터 탐색

```
> head(mydata) admit gre gpa rank 1 0 380 3.61 3 2 1 660 3.67 3 3 1 800 4.00 1 4 1
640 3.19 4 5 0 520 2.93 4 6 1 760 3.00 2

> summary(mydata) admit gre gpa rank Min. :0.0000 Min. :220.0 Min. :2.260 1: 61
1st Qu.:0.0000 1st Qu.:520.0 1st Qu.:3.130 2:151 Median :0.0000 Median :580.0
Median :3.395 3:121 Mean :0.3175 Mean :587.7 Mean :3.390 4: 67 3rd Qu.:1.0000
3rd Qu.:660.0 3rd Qu.:3.670 Max. :1.0000 Max. :800.0 Max. :4.000

> str(mydata) 'data.frame': 400 obs. of 4 variables: $ admit: int 0 1 1 1 0 1 1 0 1 0 ...
$ gre : int 380 660 800 640 520 760 560 400 540 700 ... $ gpa : num 3.61 3.67 4 3.19
2.93 3 2.98 3.08 3.39 3.92 ... $ rank : Factor w/ 4 levels "1","2","3","4": 3 3 1 4 4 2 1 2
3 2 ...

> sapply(mydata, sd) admit gre gpa rank 0.4660867 115.5165364 0.3805668
0.9444602

> xtabs(~admit+rank, data=mydata) rank admit 1 2 3 4 0 28 97 93 55 1 33 54 28 12
```

# Logistic Regression 실습

- 예제 코드

## (2) glm()을 통한 로지스틱 회귀모형 구축

- rank 변수를 factor 타입으로 변경시킨다.
- glm 을통해 로지스틱회귀모형을 fitting시킨다. **family='binomial'** 인자를 통해, glm으로 로지스틱 회귀모형을 쓸 수 있다. (link function이 logit function이라는 의미)

```
> mydata$rank <- factor(mydata$rank)
➤ mylogit <- glm(admit ~ gre + gpa + rank, data = mydata, family =
  "binomial")
➤ summary(mylogit)
```

# Logistic Regression 실습

- 예제 코드

## (2) glm()을 통한 로지스틱 회귀모형 구축

```
> summary(mylogit)
Call:
glm(formula = admit ~ gre + gpa + rank, family = "binomial",
    data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max 
-1.6268  -0.8662  -0.6388   1.1490   2.0790 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979    1.139951  -3.500  0.000465 ***
gre           0.002264    0.001094   2.070  0.038465 *
gpa           0.804038    0.331819   2.423  0.015388 *
rank2        -0.675443    0.316490  -2.134  0.032829 *
rank3        -1.340204    0.345306  -3.881  0.000104 ***
rank4        -1.551464    0.417832  -3.713  0.000205 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.52  on 394  degrees of freedom
AIC: 470.52

Number of Fisher Scoring iterations: 4
```

# Logistic Regression 실습

- 예제 코드

## (3) 로지스틱 회귀모형 결과 해석

- **Call** : 구축한 모형에 대해 다시 상기시켜준다.
- **Deviance Residuals** : Deviance residual에 대한 정보를 알려주는데, model fitting이 잘 되었는지에 대한 measure이다. 이를 통해 모형이 잘 적합됐는지를 평가할 수 있다.
- **Coefficient** : 회귀계수와 그것들의 표준편차, z-statistics(wals's z-statistics), p-value를 나타낸다. 위 결과에서는 모든 변수가 유의한 것을 알 수 있다. 로지스틱 회귀모형에서는 회귀계수가 변수가 한 단위 증가했을 때  $\log(\text{odds})$ 의 증가량으로 해석할 수 있다

# Logistic Regression 실습

- 예제 코드

## (3) 로지스틱 회귀모형 결과 해석

- Coefficients 해석

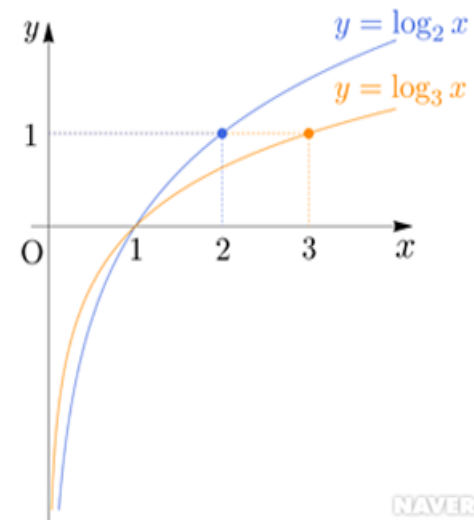
기준컬럼을 대상으로 승산비율에 대한 log 값을 출력한 것이다.

+ : 기준 컬럼에 비해 해당 컬럼이 승산비율이 높다고 해석

- : 기준 컬럼에 비해 해당 컬럼이 승산비율이 낮다고 해석

- LOG함수

**$\log(0 < x < 1)$  : - 가 되고,  
 $\log(x \geq 1)$  : +가 된다.**



# Logistic Regression 실습

- 예제 코드

## (4) example

- gre가 1증가할 때, admission의 log odds(non-admission에 대한)가 0.0022 증가한다.

```
-----
Coefficients:
(Intercept) -3.989979  1.139951  -3.500  0.000465 ***
gre          0.002264  0.001094   2.070  0.038465 *
gpa          0.804038  0.331819   2.423  0.015388 *
rank2       -0.675443  0.316490  -2.134  0.032829 *
rank3       -1.340204  0.345306  -3.881  0.000104 ***
rank4       -1.551464  0.417832  -3.713  0.000205 ***
-----
```

- gpu의 경우도 마찬가지로 해석한다.
- 더미변수인 rank의 경우 해석이 약간 다른데, 예를 들어 rank2의 회귀계수 -0.67은 rank1에서 rank2로 바뀌었을 때, log(odds)의 변화량이다. 즉, rank1에 비해 rank2가 admission에 안좋은 영향을 준다는 것을 알 수 있다.



## Logistic Regression 실습

- 예제 코드

### (4) example

- **OR(Odds ratio) 과 회귀계수의 관계**

앞서 로지스틱 회귀분석에서의 회귀계수는 log odds의 증가량이라고 언급하였다. 그러면 회귀계수에 exponential을 취하면, OR의 증분이 된다.

```
> exp(coef(mylogit))
(Intercept) gre gpa rank2 rank3
0.0185001 1.0022670 2.2345448 0.5089310 0.2617923
0.2119375
```

해석 ex) gpa가 1증가하면 admission의 non-admission에 대한 OR이 2.23배 증가한다.

# Logistic Regression 실습

- 예제 코드

## (5) 회귀계수들의 신뢰구간 얻기

Log-likelihood를 통해 구하는 법

```
confint(mylogit)
> confint(mylogit)
Waiting for profiling to be done.
                2.5 %      97.5 %
(Intercept) -6.2716202334 -1.792547080
gre          0.0001375921  0.004435874
gpa          0.1602959439  1.464142727
rank2        -1.3008888002 -0.056745722
rank3        -2.0276713127 -0.670372346
rank4        -2.4000265384 -0.753542605
```

주어진 회귀계수의 표준편차를 이용해 신뢰구간을 구하는 법

```
## CIs using standard errors
confint.default(mylogit)
> confint.default(mylogit)
                2.5 %      97.5 %
(Intercept) -6.2242418514 -1.755716295
gre          0.0001202298  0.004408622
gpa          0.1536836760  1.454391423
rank2        -1.2957512650 -0.055134591
rank3        -2.0169920597 -0.663415773
rank4        -2.3703986294 -0.732528724
```

이 신뢰구간은 이렇게도 계산 가능하다. 예를 들어, gre의 95% 신뢰구간은 아래와 같이 구한다.

```
> 0.002264-1.96*0.001094
[1] 0.00011976
> 0.002264+1.96*0.001094
[1] 0.00440824
```

## Logistic Regression 실습

- 예제 코드

### (6) Wald Test를 통해 범주형 변수의 overall effect 파악하기

rank 변수를 dummy 변수로 만들어, reference인 rank1과 비교해 어떤 effect가 있는지 분석하였다. 하지만 aod 패키지의 wald.test를 이용하면, rank의 overall effect를 파악할 수 있다. [b:회귀계수, Sigma:error term의 공분산행렬, Terms:rank변수가 있는 열] wald.test는 순서를 통해 해당 범주형 변수의 위치를 파악하기 때문에 순서를 잘 신경 써야 한다.

```
wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 4:6)
```

```
> wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 4:6)
Wald test:
-----
Chi-squared test:
X2 = 20.9, df = 3, P(> X2) = 0.00011
```

카이스퀘어 테스트의 p-value가 0.00011이므로 rank 변수는 유의하다.

## Logistic Regression 실습

- 예제 코드

### (7) 변수간의 회귀계수가 같은지 검정하기

wald test를 통해 rank가 admission에 유의한 효과가 있다는 것을 파악하였고, rank1이라는 reference에 비해 rank2와 rank3가 유의한 영향을 준다는 것도 확인하였다. 한가지 의문은 rank2, rank3, rank4 의 회귀계수가 동일한지에 대한 것이다. 회귀 계수가 동일하다면, rank1일 때에 비해 rank2, rank3, rank4의 효과가 동일하다는 것을 의미한다.

이 구문은 rank2의 효과와 rank3의 효과가 동일한지를 검정한다.

```
> l <- cbind(0, 0, 0, 1, -1, 0)
```

```
> wald.test(b = coef(mylogit), Sigma = vcov(mylogit), L = l)
```

```
> l <- cbind(0, 0, 0, 1, -1, 0)
> wald.test(b = coef(mylogit), sigma = vcov(mylogit), L =
1)
wald test:
-----

Chi-squared test:
X2 = 5.5, df = 1, P(> X2) = 0.019
```

검정 결과, rank2, rank3의 효과는 유의하다 다른 것이라 나타났다.

# Logistic Regression 실습

- 예제 코드

## (8) 예측 하기

```
newdata1 <-  
  with(mydata, data.frame(gre = mean(gre), gpa = mean(gpa), rank = factor(1:4)))
```

### 1. 다른 변수들을 고정하고 rank가 변할 때 예측값의 변화를 보기

```
newdata1 <- with(mydata, data.frame(gre = mean(gre), gpa = mean(gpa), rank = factor(1:4)))
```

```
> newdata1  
  gre gpa rank  
1 587.7 3.3899 1  
2 587.7 3.3899 2  
3 587.7 3.3899 3  
4 587.7 3.3899 4
```

```
newdata1$rankP <- predict(mylogit, newdata = newdata1, type = "response") # probability 점추정값
```

```
newdata1  
> newdata1  
  gre gpa rank rankP  
1 587.7 3.3899 1 0.5166016  
2 587.7 3.3899 2 0.3522846  
3 587.7 3.3899 3 0.2186120  
4 587.7 3.3899 4 0.1846684
```

# Logistic Regression 실습

- 예제 코드

## (8) 예측 하기

### 2. rank, gpa를 고정한 후, gre의 효과 보기

```
newdata2 <- with(mydata, data.frame(gre = rep(seq(from = 200, to = 800,  
length.out = 100), 4), gpa = mean(gpa), rank = factor(rep(1:4, each = 100))))
```

```
> newdata2
```

	gre	gpa	rank
1	200.0000	3.3899	1
2	206.0606	3.3899	1
3	212.1212	3.3899	1
4	218.1818	3.3899	1
5	224.2424	3.3899	1
6	230.3030	3.3899	1
7	236.3636	3.3899	1
8	242.4242	3.3899	1
9	248.4848	3.3899	1
10	254.5455	3.3899	1

# Logistic Regression 실습

- 예제 코드

## (8) 예측 하기

### 2. rank, gpa를 고정한 후, gre의 효과 보기

```
newdata3 <- cbind(newdata2, predict(mylogit, newdata = newdata2, type = "link",  
se = TRUE))
```

```
newdata3 <- within(newdata3, { PredictedProb <- plogis(fit) LL <- plogis(fit -  
(1.96 * se.fit))
```

```
UL <- plogis(fit + (1.96 * se.fit)) })
```

```
> head(newdata3)
```

	gre	gpa	rank	fit	se.fit	residual.scale	UL	LL	PredictedProb
1	200.0000	3.3899	1	-0.8114870	0.5147714	1	0.5492064	0.1393812	0.3075737
2	206.0606	3.3899	1	-0.7977632	0.5090986	1	0.5498513	0.1423880	0.3105042
3	212.1212	3.3899	1	-0.7840394	0.5034491	1	0.5505074	0.1454429	0.3134499
4	218.1818	3.3899	1	-0.7703156	0.4978239	1	0.5511750	0.1485460	0.3164108
5	224.2424	3.3899	1	-0.7565919	0.4922237	1	0.5518545	0.1516973	0.3193867
6	230.3030	3.3899	1	-0.7428681	0.4866494	1	0.5525464	0.1548966	0.3223773

# Logistic Regression 실습

- 예제 코드

## (8) 예측 하기

### 2. rank, gpa를 고정한 후, gre의 효과 보기

• 이 때 type="link" 옵션을 주면, link scale로 예측값을 내준다. 이는 logit 변환을 하기전의 예측값이다. **이를 통해 예측한 확률의 신뢰구간을 구할 수 있다.** 왜냐하면 이 link scale의 예측값 (fit 변수)의 표준편차를 쉽게 알 수 있기 때문이다. 따라서 link scale의 예측값의 신뢰구간을 구한 후 이를 다시 logit 변환하여 예측 확률의 신뢰구간을 얻을 수 있다. 이를 구현한 것이 바로 위의 코드이다.

• plogis 함수를 link scale의 예측값에서 확률 추정값 p를 구해준다. 즉  $\log(p/1-p) = Xb$  의 방정식을 풀어서 p를 구한다. 예를 들어 첫번째 라인의 경우,  $\log(p/(1-p)) = -0.811$  의 방정식을 양변에 exponential을 취해 풀면, 대략  $p=0.308$ 이 나온다.



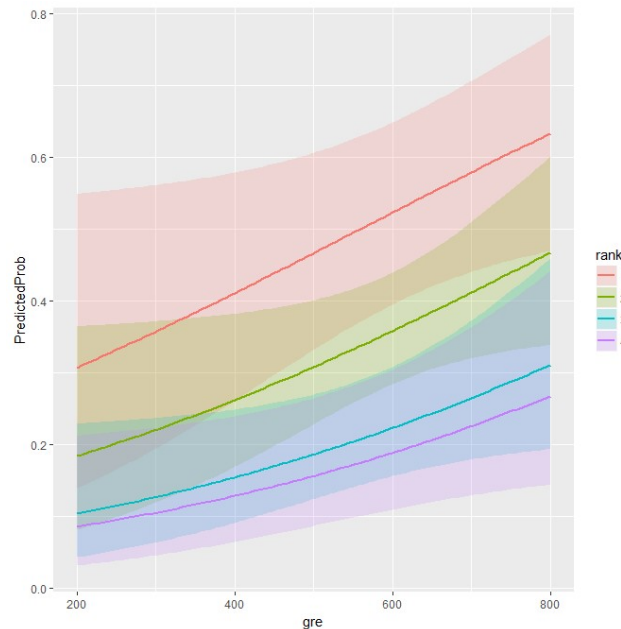
# Logistic Regression 실습

- 예제 코드

## (9) newdata3의 예측값 시각화하기

rank를 고정시켜놓고 gre가 admission 확률에 어떤 영향을 주는지를 시각화

```
ggplot(newdata3, aes(x = gre, y = PredictedProb)) + geom_ribbon(aes(ymin = LL, ymax = UL, fill = rank), alpha = 0.2) + geom_line(aes(colour = rank), si
```





## Multinomial Logistic Regression

• Multinomial Logistic Regression이란 y의 범주가 3개 이상(multi)이며 명목형(nomial)일 때 사용하는 로지스틱 회귀분석이다.

• 분류가 여러 개로 분류가 K기준이라면

$$\ln \frac{\Pr(Y_i = 1)}{\Pr(Y_i = K)} = \beta_1 \cdot \mathbf{X}_i$$

$$\ln \frac{\Pr(Y_i = 2)}{\Pr(Y_i = K)} = \beta_2 \cdot \mathbf{X}_i$$

.....

$$\ln \frac{\Pr(Y_i = K - 1)}{\Pr(Y_i = K)} = \beta_{K-1} \cdot \mathbf{X}_i$$

• 그 후 , 양변을 e의 지수로 하고 정리를 한다.

$$\Pr(Y_i = 1) = \Pr(Y_i = K) e^{\beta_1 \cdot \mathbf{X}_i}$$

$$\Pr(Y_i = 2) = \Pr(Y_i = K) e^{\beta_2 \cdot \mathbf{X}_i}$$

.....

$$\Pr(Y_i = K - 1) = \Pr(Y_i = K) e^{\beta_{K-1} \cdot \mathbf{X}_i}$$

# Multinomial Logistic Regression

확률의 합은 1이므로, 위 식의 좌우 변을 모두 합한다

$$\Pr(Y_i = K) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot \mathbf{X}_i}}$$

•최종적으로 정리하면 이런 식이 도출 되는 것이다.

$$\Pr(Y_i = 1) = \frac{e^{\beta_1 \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot \mathbf{X}_i}}$$

$$\Pr(Y_i = 2) = \frac{e^{\beta_2 \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot \mathbf{X}_i}}$$

.....

$$\Pr(Y_i = K - 1) = \frac{e^{\beta_{K-1} \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot \mathbf{X}_i}}$$

# Multinomial Logistic Regression

- 예제 코드

## (1) 데이터 설명

해당 데이터셋은 hsbdemo 데이터셋으로 총 n은 200이다.

```
ml <- read.dta("https://stats.idre.ucla.edu/stat/data/hsbdemo.dta")
```

```
> head(ml)
  id female   ses schtyp      prog read write math science socst honors awards cid
1  45 female low public vocation  34   35  41    29    26 not enro|led    0    1
2 108 male middle public general  34   33  41    36    36 not enro|led    0    1
3  15 male high public vocation  39   39  44    26    42 not enro|led    0    1
4  67 male low public vocation  37   37  42    33    32 not enro|led    0    1
5 153 male middle public vocation  39   31  40    39    51 not enro|led    0    1
6  51 female high public general  42   36  42    31    39 not enro|led    0    1
```

Y : prog (프로그램 타입)

X : ses(social economic status), write (writing score)

다중회귀분석을 통해 X가 Y에 어떤 영향을 미치는지 알아본다.

# Multinomial Logistic Regression

- 예제 코드

## (2) 빈도 테이블과 그룹 평균을 통해 X와 Y의 관계 알아보기

```
with(ml, table(ses, prog))  
# 또는 table(ml$ses, ml$prog)  
# 또는 xtabs(~ses+prog, data=ml)
```

```
> with(ml, table(ses, prog))
```

	prog		
ses	general	academic	vocation
low	16	19	12
middle	20	44	31
high	9	42	7

```
with(ml, do.call(rbind, tapply(write, prog, function(x) c(M = mean(x), SD = sd(x)))))
```

```
> with(ml, do.call(rbind, tapply(write, prog, function(x) c(M = mean(x), SD = sd(x)))))
```

	M	SD
general	51.33333	9.397775
academic	56.25714	7.943343
vocation	46.76000	9.318754

# Multinomial Logistic Regression

- 예제 코드

- (3) nnet 패키지의 multinom 함수를 통해 다중 로지스틱 회귀분석 실시

```
ml$prog2 <- relevel(ml$prog, ref = "academic")  
test <- multinom(prog2 ~ ses + write, data = ml)
```

multinom을 실행하기 전에 relevel이라는 함수를 통해 reference를 지정해 준다. 이 reference를 기준으로 결과를 해석할 수 있다. 따라서 이 reference는 어떤 baseline이 되어야한다. 이 경우에 academic이 baseline이며 이를 기준으로 하여 분석한다.

multinom 함수를 실행하면 모델이 fitting되어 계수가 결정되고 이 정보는 test에 담긴다.

# Multinomial Logistic Regression

- 예제 코드

## (3) nnet 패키지의 multinom 함수를 통해 다중 로지스틱 회귀분석 실시

```
summary(test)
> summary(test)
Call:
multinom(formula = prog2 ~ ses + write, data = ml)

Coefficients:
(Intercept)  sesmiddle  seshigh  write
general      2.852198 -0.5332810 -1.1628226 -0.0579287
vocation     5.218260  0.2913859 -0.9826649 -0.1136037

Std. Errors:
(Intercept)  sesmiddle  seshigh  write
general      1.166441  0.4437323  0.5142196  0.02141097
vocation     1.163552  0.4763739  0.5955665  0.02221996

Residual Deviance: 359.9635
AIC: 375.9635
```

$$\ln \left( \frac{P(\text{prog} = \text{general})}{P(\text{prog} = \text{academic})} \right) = b_{10} + b_{11}(\text{ses} = 2) + b_{12}(\text{ses} = 3) + b_{13}\text{write}$$

$$\ln \left( \frac{P(\text{prog} = \text{vocation})}{P(\text{prog} = \text{academic})} \right) = b_{20} + b_{21}(\text{ses} = 2) + b_{22}(\text{ses} = 3) + b_{23}\text{write}$$



# Multinomial Logistic Regression

- 예제 코드

## (4) 결과해석

이 예제의 경우 다중 로지스틱 회귀분석의 계수는 2세트가 나오게 된다.

- ① academic vs general 의 log odds에 관한 계수
- ② academic vs vocation의 log odds에 관한 계수

$$\ln \left( \frac{P(\text{prog} = \text{general})}{P(\text{prog} = \text{academic})} \right) = b_{10} + b_{11}(\text{ses} = 2) + b_{12}(\text{ses} = 3) + b_{13}\text{write}$$

$$\ln \left( \frac{P(\text{prog} = \text{vocation})}{P(\text{prog} = \text{academic})} \right) = b_{20} + b_{21}(\text{ses} = 2) + b_{22}(\text{ses} = 3) + b_{23}\text{write}$$

## [결과해석]

```
Coefficients:
              (Intercept)  sesmiddle  seshigh  write
general      2.852198    -0.5332810  -1.162826  -0.0579287
vocation      5.218260     0.2913859  -0.9826649  -0.1136037
```

- ① write가 1단위 증가할 때, academic이 general이 될 log odds가 -0.058이 된다.write가 1단위 증가할 때, academic이 vocation이 될 log odds가 -0.1136037이 된다.ses:row에서 ses:high로 변할 때, academic이 general이 될 log odds가 -1.1628이 된다.
- ② ses:row에서 ses:middle로 변할 때, academic이 general이 될 log odds가 -0.5332가 된다.
- ③ ses:row에서 ses:high로 변할 때, academic이 vocation이 될 log odds가 -0.9826이 된다.
- ④ ses:row에서 ses:middle로 변할 때, academic이 vocation이 될 log odds가 +0.29가 된다.



# Multinomial Logistic Regression

- 예제 코드

## (5) 변수의 유의성 판단 & OR 추정값

### 변수의 유의성 판단

coefficient가 0라는 귀무가설 하에서  $\text{coef} \sim N(0, \text{sd}^2)$ 라는걸 이용하여 z-test 한다.

```
z <- summary(test)$coefficients/summary(test)$standard.errors
z
> z <- summary(test)$coefficients/summary(test)$standard.errors
> z
              (Intercept)  sesmiddle  seshigh  write
general      2.445214    -1.2018081  -2.261334  -2.705562
vocation     4.484769     0.6116747  -1.649967  -5.112689
# 2-tailed z test
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
> p <- (1 - pnorm(abs(z), 0, 1)) * 2
> p
              (Intercept)  sesmiddle  seshigh  write
general  0.0144766100  0.2294379  0.02373856  6.818902e-03
vocation 0.0000072993  0.5407530  0.09894976  3.176045e-07
```

### OR 추정값

```
exp(coef(test))
> exp(coef(test))
              (Intercept)  sesmiddle  seshigh  write
general      17.32582  0.5866769  0.3126026  0.9437172
vocation     184.61262  1.3382809  0.3743123  0.8926116
```

회귀 계수들에 exponential을 취하면, 이는 OR에 대한 추정값이 된다.

# Multinomial Logistic Regression

- 예제 코드

## (6) 예측하기

- Write가 평균일 때, ses에 따라 예측값이 어떻게 달라지는가?

```
dses <- data.frame(ses = c("low", "middle", "high"), write = mean(ml$write))
dses
predict(test, newdata = dses, "probs")
```

```
> dses <- data.frame(ses = c("low", "middle", "high"), write = mean(ml$write))
> dses
  ses write
1 low  52.775
2 middle 52.775
3 high  52.775

> predict(test, newdata = dses, "probs")
  academic general vocation
1 0.4396845 0.3581917 0.2021238
2 0.4777488 0.2283353 0.2939159
3 0.7009007 0.1784939 0.1206054
```



## Multinomial Logistic Regression

- 예제 코드

### (6) 예측하기

- Write를 변화시켜가면서 예측값의 변화 관찰하기

```
dwrite <- data.frame(ses = rep(c("low", "middle", "high"), each = 41), write = rep(c(30:70), 3))

## store the predicted probabilities for each value of ses and write
pp.write <- cbind(dwrite, predict(test, newdata = dwrite, type = "probs", se = TRUE))

## calculate the mean probabilities within each level of ses
by(pp.write[, 3:5], pp.write$ses, colMeans)

## melt data set to long for ggplot2
lpp <- melt(pp.write, id.vars = c("ses", "write"), value.name = "probability")
head(lpp) # view first few rows
> head(lpp) # view first few rows
  ses write variable probability
1 low   30 academic 0.09843588
2 low   31 academic 0.10716868
3 low   32 academic 0.11650390
4 low   33 academic 0.12645834
5 low   34 academic 0.13704576
6 low   35 academic 0.14827643
```



# Multinomial Logistic Regression

- 예제 코드

## (7) 그래프 그리기

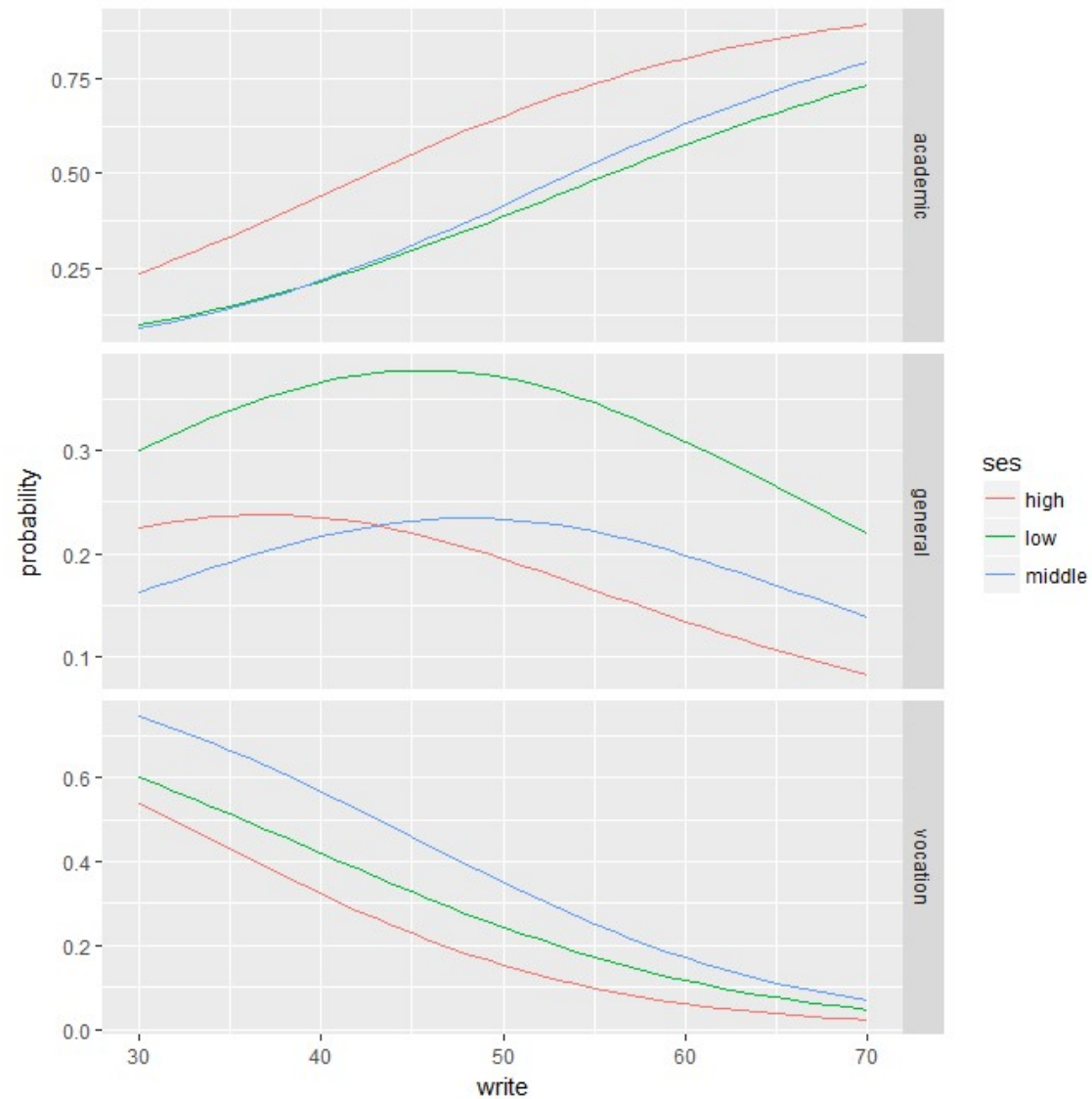
이 그래프는 식을 전개하여 y를 sigmoid 함수 형태로 만든 것이다. 2개의 적합식에서 3개의 함수를 만들어낼 수 있다.

```
## plot predicted probabilities across write values for each level of ses  
## faceted by program type  
ggplot(lpp, aes(x = write, y = probability, colour = ses)) + geom_line()  
+ facet_grid(variable ~ ., scales = "free")
```

# Multinomial Logistic Regression

- 예제 코드

## (7) 그래프 그리기





# Multinomial Logistic Regression

- 예제 코드

## (8) $Y \sim \text{write}$ 의 결과랑 무엇이 다른가?

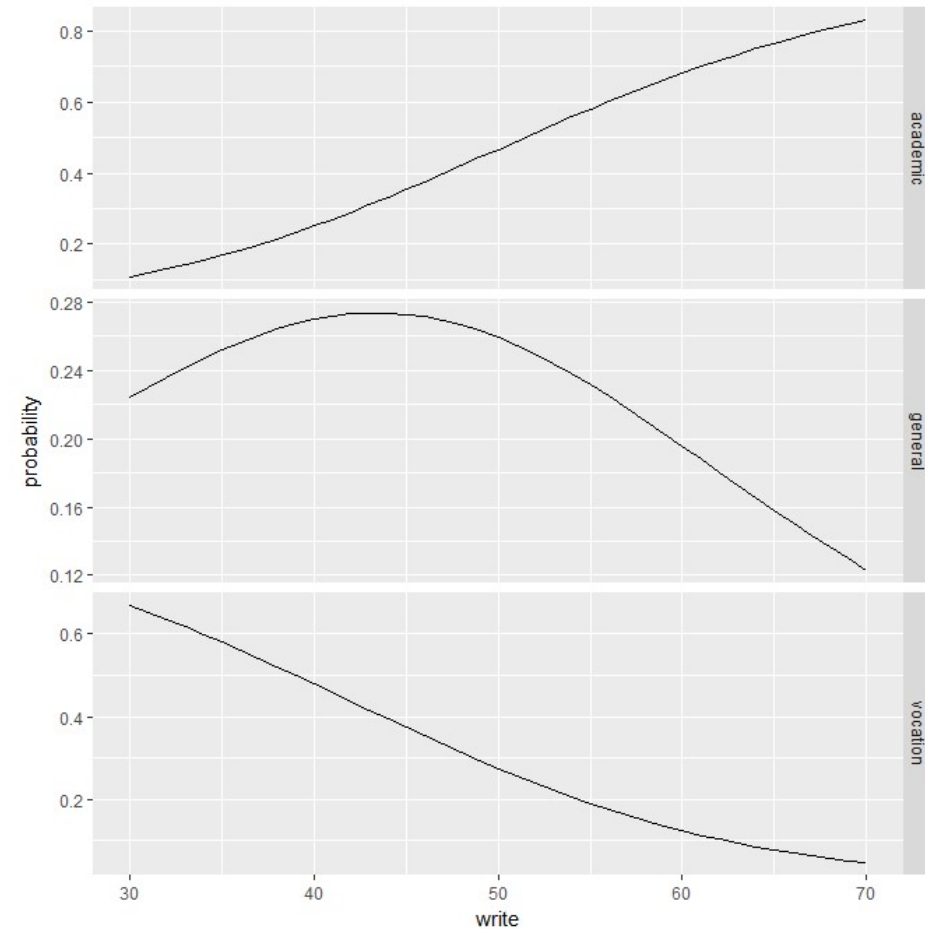
예측의 관점에서 보면 ses를 빼고 write만을 통해  $Y$ 를 예측하면 예측을 세분화하지 못하여 예측력이 떨어진다고 볼 수 있다. 이는 ses를 고려하지 않고, write와  $Y$ 의 관계만을 보기 때문이다. 따라서 ses가 추가됨으로써 write의 계수가 바뀌는 것을 알 수 있다. 이 과정을 ses를 **보정**한다고도 말한다.

# Multinomial Logistic Regression

- 예제 코드

(8)  $Y \sim \text{write}$ 의 결과랑 무엇이 다른가?

<https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/>





# Multinomial Logistic Regression

다항 로지스틱 회귀분석에서는 nnet패키지를 사용한다.

```
install.packages("nnet")#multinom 함수를 사용하기 위한 패키지이다.  
library(nnet)  
str(iris)  
m=multinom(Species~.,data=iris)  
m
```





## 종속변수가 순서 척도인 로지스틱 회귀분석

다항 로지스틱 회귀분석에서 종속변수가 명목형이 아닌 순서형인 경우 분석하는 방법은 다음과 같다.

- 예제 코드

### (1) 라이브러리 로드

```
>require(foreign)
>require(ggplot2)
>require(MASS)
>require(Hmisc)
>require(reshape2)
```

## 종속변수가 순서 척도인 로지스틱 회귀분석

### •예제 코드

### (2) 데이터 읽기

```
> dat <- read.dta("https://stats.idre.ucla.edu/stat/data/ologit.dta")
> head(dat)
```

```
> head(dat)
  1 very likely 0 0 3.26
  2 somewhat likely 1 0 3.21
  3 unlikel 1 1 3.94
  4 somewhat likely 0 0 2.81
  5 somewhat likely 0 0 2.53
  6 unlikel 0 1 2.59
```

- X : pared(부모중 한명이 대학 학위가 있는지 여부), public(졸업한 학교가 공립인지), gpa(학점)
- Y : apply ("unlikely", "somewhat likely", "very likely", 각각 1, 2, 3으로 코딩되어있음) => 대학원에 진학할 가능성

## 종속변수가 순서 척도인 로지스틱 회귀분석

•예제 코드

### (3) 범주형 빈도 확인

lapply 함수를 이용해 범주형 변수의 빈도표를 확인

```
## one at a time, table apply, pared, and public  
> lapply(dat[, c("apply", "pared", "public")], table)
```

```
> lapply(dat[, c("apply", "pared", "public")], table)  
$apply  
      unlikely somewhat likely      very likely  
      220      140      40  
  
$pared  
      0      1  
337  63  
  
$public  
      0      1  
343  57
```

## 종속변수가 순서 척도인 로지스틱 회귀분석

•예제 코드

### (3) 범주형 빈도 확인

변수 빈도표를 통해 로지스틱 회귀분석을 하기에 앞서 예상되는 결과를 미리 확인한다.

```
## three way cross tabs (xtabs) and flatten the table  
ftable(xtabs(~ public + apply + pared, data = dat))
```

```
> ftable(xtabs(~ public + apply + pared, data = dat))  
      pared 0  1  
public apply  
0      unlikely 175 14  
      somewhat 98 26  
      very likely 20 10  
1      unlikely 25 6  
      somewhat 12 4  
      very likely 7 3
```



## 종속변수가 순서 척도인 로지스틱 회귀분석

•예제 코드

### (4) 연속형 변수 분포 확인

```
summary(dat$gpa)
> summary(dat$gpa)
   Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
 1.900   2.720   2.990   2.999   3.270   4.000

sd(dat$gpa)
> sd(dat$gpa)
[1] 0.3979409
```

## 종속변수가 순서 척도인 로지스틱 회귀분석

•예제 코드

### (5) 연속형 독립 변수와 종속 변수와의 관계 파악

Apply에 따라 gpa가 어떻게 분포하여있는지 상자그림을 통해 확인한다. 이러한 경향을 수치화하는 것이 로지스틱 회귀분석의 목표이다.

```
ggplot(dat, aes(x = apply, y = gpa))  
  + geom_boxplot(size = .75)  
  + geom_jitter(alpha = .5)  
  + facet_grid(pared ~ public, margins = TRUE)  
  + theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```

순서 척도가 종속변수인 로지스틱 회귀분석을 어떻게 하는지 알아본다.

## 종속변수가 순서 척도인 로지스틱 회귀분석

### •예제 코드

#### (5) 연속형 독립 변수와 종속 변수와의 관계 파악

이 때 가능한 분석방법은 꼭 순서형 척도만은 아닐 것이다. GPA만을 독립변수로 하여 ANOVA를 할 수도 있고,  $y$ 를 1,2,3으로 놓고 다중회귀분석을 할 수도 있다. 또  $y$ 를 순서가 없는 명목형 척도로 보고 명목형 다항 로지스틱 분석(multinomial logistic regression)을 할 수도 있을 것이다. 하지만 해당 데이터에 대해 위 분석 방법들은 가정에 맞지 않는다. 따라서 순서형 척도를 종속변수로 하는 로지스틱 회귀분석(ordinal logistic regression)이 적합하다.

순서형 척도를 종속 변수로 하는 로지스틱 회귀분석의 중요한 가정은 회귀계수가 같다고 가정하는 것이다.  $y=1,2,3$ 인 이 데이터의 경우 어떠한 변수의 변화가 1 vs 2,3의 OR에 미치는 영향과 1,2 vs 3의 OR에 미치는 영향이 같다고 가정한다.  $y$ 의 범주가 3개인 경우 2개의 식이 나오면 이 두식은 회귀계수가 같고, 단지 절편만 다르다.

MASS 패키지의 polr 함수를(proportional odds logistic regression) 이용하여 순서형 변수를 종속변수로 하는 다항 로지스틱 회귀분석을 해보자.



## 종속변수가 순서 척도인 로지스틱 회귀분석

### •예제 코드

#### (5) 연속형 독립 변수와 종속 변수와의 관계 파악

회귀식을 적어주고 summary를 통해 적합된 결과를 본다. Hess=TRUE는 optimization을 할 때 사용하는 Hessian matrix를 의미한다.

([https://en.wikipedia.org/wiki/Hessian\\_matrix](https://en.wikipedia.org/wiki/Hessian_matrix))

```
# fit ordered logit model and store results 'm'  
m <- polr(apply ~ pared + public + gpa, data = dat, Hess=TRUE)  
## view a summary of the model summary(m)
```





## 종속변수가 순서 척도인 로지스틱 회귀분석

•예제 코드

### (5) 연속형 독립 변수와 종속 변수와의 관계 파악

```
> summary(m)
Call:
polr(formula = apply ~ pared + public + gpa, data = dat, Hess = TRUE)

Coefficients:
                Value Std. Error t value
pared      1.04769      0.2658  3.9418
public    -0.05879      0.2979  -0.1974
gpa        0.61594      0.2606  2.3632

Intercepts:
                Value Std. Error t value
unlikely|somewhat likely  2.2039  0.7795  2.8272
somewhat likely|very likely  4.2994  0.8043  5.3453

Residual Deviance: 717.0249
AIC: 727.0249
```

## ■ 종속변수가 순서 척도인 로지스틱 회귀분석

•예제 코드

### (6) 결과해석

회귀계수에 exponential을 취하면 OR(Odds ratio)이 되며 이를 통해 결과를 해석해본다.

```
# odds ratios  
exp(coef(m))
```

```
> exp(coef(m))  
pared public gpa  
2.8510579 0.9429088 1.8513972
```

## 종속변수가 순서 척도인 로지스틱 회귀분석

- 예제 코드

```
> exp(coef(m))  
pared    public    gpa  
2.8510579  0.9429088  1.8513972
```

### (6) 결과해석

순서척도 로지스틱 회귀분석에서 OR은 *proportional OR* 이라고도 부른다.

- 다른 변수가 고정되어있을 때, pared=0일 때에 비하여 pared=1일 때, "very likely" **vs** "somewhat likely" or "unlikely"의 OR이 2.851이다.

- 다른 변수가 고정되어있을 때, pared=0일 때에 비하여 pared=1일 때, "very likely" or "somewhat likely" **vs** "unlikely"의 OR이 2.851이다.

- 이는 즉, pared=1일 때, 대학원 진학(apply)을 할 가능성이 높다는 것을 뜻한다.

- 같은 binary 변수인 public도 위와 마찬가지로 해석한다.

- 다른 변수가 고정되어있을 때, gpa가 1 증가할 때, "very likely" **vs** "somewhat likely" or "unlikely" Odds가 1.85배가 된다.

- 다른 변수가 고정되어있을 때, gpa가 1 증가할 때, "very likely" or "somewhat likely" **vs** "unlikely" Odds가 1.85배가 된다



## 종속변수가 순서 척도인 로지스틱 회귀분석

•예제 코드

### (6) 결과해석

이를 식으로 표현하면,

$$\log \text{Odds}(Y=\text{very}) = 2.2039 + 1.04\text{pared} - 0.05879\text{public} + 0.61594\text{gpa}$$

$$\log \text{Odds}(Y=\text{very or somewhat}) = 4.2994 + 1.04\text{pared} - 0.05879\text{public} + 0.61594\text{gpa}$$

이 때,  $\text{Odds}(Y=1) = P(P1/(1-P1))$ ,  $\text{Odds}(Y=1,2) = P((P1+P2)/(1-P1-P2))$   
log Odds 는 logit function 이라고도 부른다.

위 두 방정식을 풀면  $P(Y=1)$ ,  $P(Y=2)$ ,  $P(Y=3)$  3개의 식을 얻을 수 있다. 이 식은 X가 주어졌을 때, Y의 예측값을 구한다.



# Multinomial Logistic Regression

## 다항 로지스틱 회귀분석 요약/정리

- 결과 변수가 연속형인 경우는 다중 선형회귀분석(multiple linear regression analysis)를 실시한다.
- 결과 변수가 범주형인 경우는 두 범주 일때와 두 범주 이상 일 때(명목형) , 두 범주 이상 일 때(순서형)로 등으로 나누어지고
- 범주가 하나 일때 **단순 로지스틱 회귀분석 (logistic regression)**
- 두범주 일때 **다중 로지스틱 회귀분석(multiple logistic regression)**
- 두범주 이상 명목형인 경우는 **다항 로지스틱 회귀분석(polychotomous logistic regression analysis)**
- 두범주 이상 순서형인 경우는 **순서형 로지스틱 회귀분석(ordinal logistic regression analysis)**를 시행한다