Linear Regression & Logistic Regression

김서희 | 김세정 | 김장미 | 우현우

Contents

Linear Regression & R 실습

회귀 (Regression)

- 한 개의 수치 종속 변수(예측 값)와 한 개 이상의 수치 독립 변수(예측변수) 사이의 관계를 명시하는 것
- Why? 데이터 요소 간의 복잡한 관계를 모델링하고, 처리가 결과에 미치는 영향을 추정하며 미래의 값을 보간하기 위해서
- When? 사건과 반응 간의 인과 관계의 정량화, 미래의 행위를 예측하는데 필요한 패턴의 식별

회귀 방정식

- $y = a + bx$ 같은 기울기-절편 형식으로 데이터를 모젠퀵
- x 와 y 값의 관계를 가장 잘 나타내는 a 와 b 의 값을 찾아내는 것
- a 와 b 는 파라미터인 변수임, 즉 a 와 b 의 값을 찾는 것은 파라미터 추정치를 찾아내는 것!
- 선형 회귀란 직선을 사용하는 가장 기본적인 모델

단순 선형 회귀 (Simple linear regression)

- 독립 변수가 한 개만 있는 경우 (ex. 온도가 기계 손상에 영향을 미치는가? … 독립변수 한 개!)
- $y = a + bx$ ..... a 와 b 구하기는 p3~4

다중 선형 회귀 (Multiple linear regression)

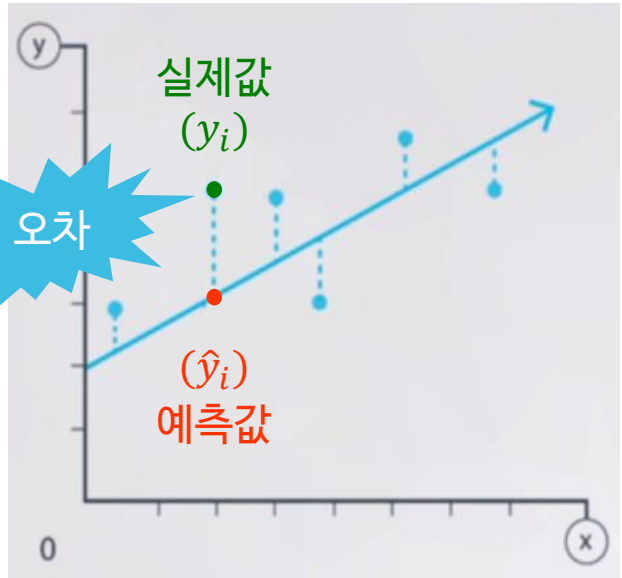
- 독립 변수가 두 개 이상 있는 경우 (ex. 온도, 압력, 연식이 기계 손상에 영향을 미치는가? … 독립변수 여러 개!)
- $y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \epsilon$ ..... β 구하기는 p5

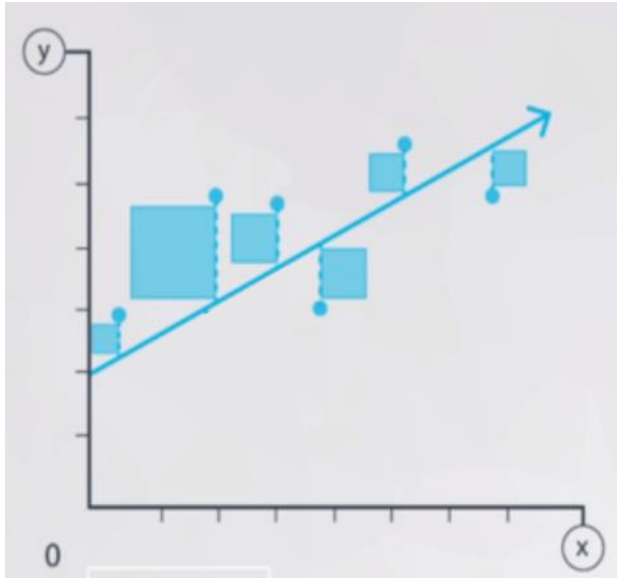
장점	단점
▪ 수치 데이터를 모델링하기 위한 가장 일반적인 방법 ▪ 어떤 모델링 작업에도 대부분 적용됨 ▪ 특징과 결과 간의 관계에 대한 강도와 크기 추정치를 제공	▪ 데이터에 대한 강한 가정 ▪ 모델 형태가 사용자에 의해 미리 지정돼야만 함 ▪ 누락 데이터를 처리하지 않음 ▪ 수치 특징을 처리하므로 범주 데이터는 추가적 처리 필요 ▪ 이상치에 민감함

보통 최소 제곱법 (OLS, Ordinary least squares)

- a(절편)와 b(기울기)의 최적 추정치를 결정하는 것
- 최적 추정치는 오차 제곱합이 최소화되도록 선택한다

- 오차제곱합이란?





- 오차 (e)
  : 실제 값과 예측 값 사이의 수직 거리
- $\sum(y_i - \hat{y}_i)^2 = \sum e_i^2$

보통 최소 제곱법 (OLS, Ordinary least squares) (Cont)

이제 원리를 알았으니 최京嶋 婆椄 彧뿷崩 큚쩮읰 a(절편)와 b(기울기)를 구해보자
먼저 b 값을 알아야 a 값을 구할 수 있음!

$$b = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}, \quad Var(x) = \text{분산} = \frac{\sum(x_i - \bar{x})^2}{n}, \quad Cov(x,y) = \text{공분산} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n}$$

$$\frac{Cov(x,y)}{Var(x)} = \frac{\frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n}}{\frac{\sum(x_i - \bar{x})^2}{n}} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$, 따라서 $\boldsymbol{b}$는 $\boldsymbol{\frac{Cov(x,y)}{Var(x)}}$

a 를 구할 때는 $y = a + bx$ 를 떠올려 보자, a의 해는 b 값에 종속되므로 $\boldsymbol{a}$는 $\boldsymbol{\bar{y} - b\bar{x}}$

다중 선형 회귀 베타값 구하기

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \varepsilon$$

(오차)

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \varepsilon$$

(오차)

(행렬)

$$Y = X\beta + \varepsilon \longrightarrow \hat{\beta} = (X^T X)^{-1} X^T Y$$

β_0	β_1	β_2

x_0	x_1	x_2
1		
1		
1		

✓ solve(): 역행렬

✓ t() : 전치행렬

- 각 특징 i에 대해 추정된 베타 값과 x 값

- α는 어떤 독립변수 x와도 연관되지 않으며, 다른 회귀 파라미터와 전혀 차이가 없기 때문에 β_0와 값이 1인 상수 항의 곱으로 생각할 수 있음

- 독립변수들은 행렬 X로 결합되며, 종속변수는 벡터 Y로 모든 예시에 대한 행을 가짐
- 절편 항에 대한 1로 된 열을 각 특징에 더해 구성
- β와 ε도 모두 벡터로 처리함

1. 데이터 탐색 단계

‘Boston’ 데이터 불러오기

```
library(MASS)
data("Boston", package = "MASS")
data<-Boston

str(data)
```

```
> str(data)
'data.frame':   506 obs. of  14 variables:
 $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
 $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
 $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 .
 $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.5
 $ rm     : num  6.58 6.42 7.18 7 7.15 ...
 $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ..
 $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
 $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
 $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
 $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 .
 $ black  : num  397 397 393 395 397 ...
 $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
 $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

변수명	속성	변수 설명
crim	수치형(numeric)	per capita crime rate by town 타운별 1인당 범죄율
zn	수치형(numeric)	proportion of residential land zoned for lots over 25,00 25,000평방피트를 초과하는 거주지역 비율
indus	수치형(numeric)	proportion of non-retail business acres per town. 비소매 사업지역의 토지 비율
chas	범주형(integer)	Charles River dummy variable (= 1 if tract bounds river; otherwise). 찰스강 더비 변수 (강의 경계에 위치 = 1, 아니면 = 0)
nox	수치형(numeric)	nitrogen oxides concentration (parts per 10 million). 10ppm당 농축 일산화질소
rm	수치형(numeric)	average number of rooms per dwelling. 주택 1가구등 방의 평균 개수
age	수치형(numeric)	proportion of owner-occupied units built prior to 1940. 1940년 이전에 건축된 소유자 주택 비율
dis	수치형(numeric)	weighted mean of distances to five Boston employment cen 5개의 보스턴 고용센터까지의 접근성 지수
rad	범주형(integer)	index of accessibility to radial highways. 방사형 도로까지의 접근성 지수
tax	수치형(numeric)	full-value property-tax rate per \$10,000. 10,000달러당 재산세율
ptratio	수치형(numeric)	pupil-teacher ratio by town. 타운별 학생/교사 비율
black	수치형(numeric)	*1000(Bk - 0.63)^2* where *Bk* is the proportion of blacks b 타운별 흑인의 비율
lstat	수치형(numeric)	lower status of the population (percent). 모집단의 하위계층의 비율
medv (목표변수)	수치형(numeric)	median value of owner-occupied homes in \$1000s. 본인 소유의 주택가격(중앙값)

1. 데이터 탐색 단계

```
stem(data$medv)
```

```
> stem(data$medv)

  The decimal point is at the |

   4 | 006
   6 | 30022245
   8 | 1334455788567
  10 | 2224455899035778899
  12 | 013567778011112333444455668888899
  14 | 011123344555668999000122344666667
  16 | 011122345566778801112223444555678888889
  18 | 012223344455556677788999900111122333334444445555666667788899999
  20 | 000001111122333344445556666667788899000112222244444556677777778899
  22 | 000000012222233445556666677888899990000111111122223333445667777888889
  24 | 001112333444455566777888800000000123
  26 | 24456667011555599
  28 | 01244567770011466889
  30 | 111357801255667
  32 | 0024579011223448
  34 | 679991244
  36 | 01224502369
  38 | 78
  40 | 37
  42 | 38158
  44 | 084
  46 | 07
  48 | 358
  50 | 0000000000000000
```

```
i=which(data$medv==50)
boston<- data[-i,]
boston$chas <- factor(boston$chas)
boston$rad <- factor(boston$rad)
```

① 주택 최대값 50인 자료 빼기

② Chas와 rad 변수 factor 형식으로 바꾸기

1. 데이터 탐색 단계

```
> cor(boston[c("zn","nox","rm","age","dis","tax","ptratio","black","crim","indus","medv")])
                zn        nox         rm        age        dis        tax    ptratio      black       crim      indus       medv
zn       1.0000000 -0.5121366  0.3105064 -0.5631835  0.6732275 -0.3028968 -0.3818151  0.1761175 -0.1990746 -0.5271206  0.4046076
nox     -0.5121366  1.0000000 -0.3226090  0.7276714 -0.7681220  0.6673801  0.1883808 -0.3830868  0.4204761  0.7651551 -0.5244510
rm       0.3105064 -0.3226090  1.0000000 -0.2684636  0.2457893 -0.2819552 -0.2932991  0.1192044 -0.2193066 -0.4124130  0.6866343
age     -0.5631835  0.7276714 -0.2684636  1.0000000 -0.7430434  0.4996824  0.2684593 -0.2790018  0.3537511  0.6379705 -0.4929152
dis      0.6732275 -0.7681220  0.2457893 -0.7430434  1.0000000 -0.5320248 -0.2467726  0.2994261 -0.3822309 -0.7102844  0.3688132
tax     -0.3028968  0.6673801 -0.2819552  0.4996824 -0.5320248  1.0000000  0.4522520 -0.4482115  0.5837111  0.7176777 -0.5724417
ptratio -0.3818151  0.1883808 -0.2932991  0.2684593 -0.2467726  0.4522520  1.0000000 -0.1736361  0.2870789  0.3876564 -0.5186410
black    0.1761175 -0.3830868  0.1192044 -0.2790018  0.2994261 -0.4482115 -0.1736361  1.0000000 -0.3844599 -0.3633936  0.3649280
crim    -0.1990746  0.4204761 -0.2193066  0.3537511 -0.3822309  0.5837111  0.2870789 -0.3844599  1.0000000  0.4080530 -0.4501152
indus   -0.5271206  0.7651551 -0.4124130  0.6379705 -0.7102844  0.7176777  0.3876564 -0.3633936  0.4080530  1.0000000 -0.6000052
medv     0.4046076 -0.5244510  0.6866343 -0.4929152  0.3688132 -0.5724417 -0.5186410  0.3649280 -0.4501152 -0.6000052  1.0000000
```

Cor을 통해 변수와의 관계를 보았습니다.
종속변수 medv와 독립변수 rm의 상관관계가 커보여서
단순회귀모형을 만들어보았습니다!

2. 모델 훈련 단계 – 단순선형회귀

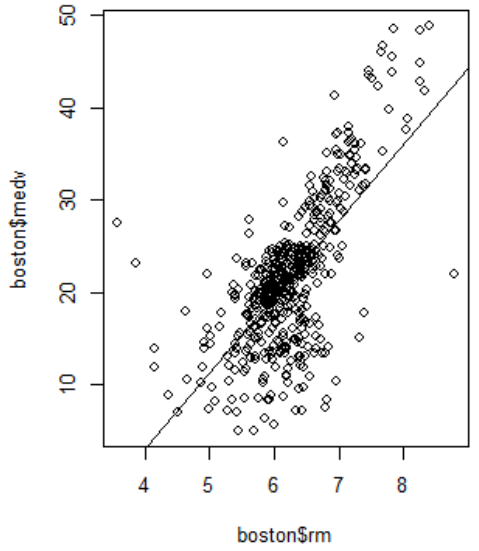
```
m = lm( dv ~ iv , data = mydata )
```

종속변수 / 독립변수 / 종속변수, 독립변수를 포함하는 데이터

```
m = lm(medv~rm, data=boston)
summary(m)
```

```
> summary(m)

Call:
lm(formula = medv ~ rm, data = boston)

Residuals:
     Min       1Q   Median       3Q      Max
-20.6928  -2.2840   0.4704   3.1676  28.0608

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -30.0051     2.4886  -12.06   <2e-16 ***
rm            8.2686     0.3963   20.86   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.724 on 488 degrees of freedom
Multiple R-squared:  0.4715,    Adjusted R-squared:  0.4704
F-statistic: 435.3 on 1 and 488 DF,  p-value: < 2.2e-16
```

```
plot(boston$rm, boston$medv)
abline(m)
```



2. 모델 훈련 단계 – 단순선형회귀

```
> summary(m)

Call:
lm(formula = medv ~ rm, data = boston)

Residuals:
     Min       1Q   Median       3Q      Max
-20.6928  -2.2840   0.4704   3.1676  28.0608

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -30.0051     2.4886  -12.06   <2e-16 ***
rm            8.2686     0.3963   20.86   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.724 on 488 degrees of freedom
Multiple R-squared:  0.4715,    Adjusted R-squared:  0.4704
F-statistic: 435.3 on 1 and 488 DF,  p-value: < 2.2e-16
```

`Residuals:`

오차에 대한 요약 통계로 '잔차'를 나타내준다.

`Pr(>|t|)`

통계적 유의성 확인
유의 수준보다 낮은 p값이 (보통 0.05)
통계적으로 유의한 것으로 간주된다.

`Adjusted R-squared`

모델이 전체적으로 종속 변수 값을 얼마나 잘 설명하는 지를 알려주는 역할 ! 변수가 늘어날수록 증가하는 R-squared와는 달리 변수의 개수만큼 패널티를 준 값이다.

모델이 종속 변수 변화량의 약 47% 정도 설명하고 있다.

3. 예측해보기

만든 회귀모델을 가지고 예측 값을 산출해보겠습니다.

```
lm_result <- lm(medv~rm, data=boston)

#예측할 독립변수
room <- c(6, 7, 8, 9, 10)
df_input <- data.frame(rm=room)

#예측
predict_medv <- predict(lm_result, df_input, interval = "confidence", level=0.95)

#결과
cbind(df_input, predict_medv)
```

```
> #결과
> cbind(df_input, predict_medv)
  rm      fit      lwr      upr
1  6 19.60622 19.06339 20.14906
2  7 27.87478 27.09804 28.65153
3  8 36.14334 34.68571 37.60096
4  9 44.41190 42.20766 46.61614
5 10 52.68045 49.71307 55.64784
```

쨔잔!.

2. 모델 훈련 단계 – 다중선형회귀

```
m = lm( dv ~ iv + iv +.. , data = mydata )
```

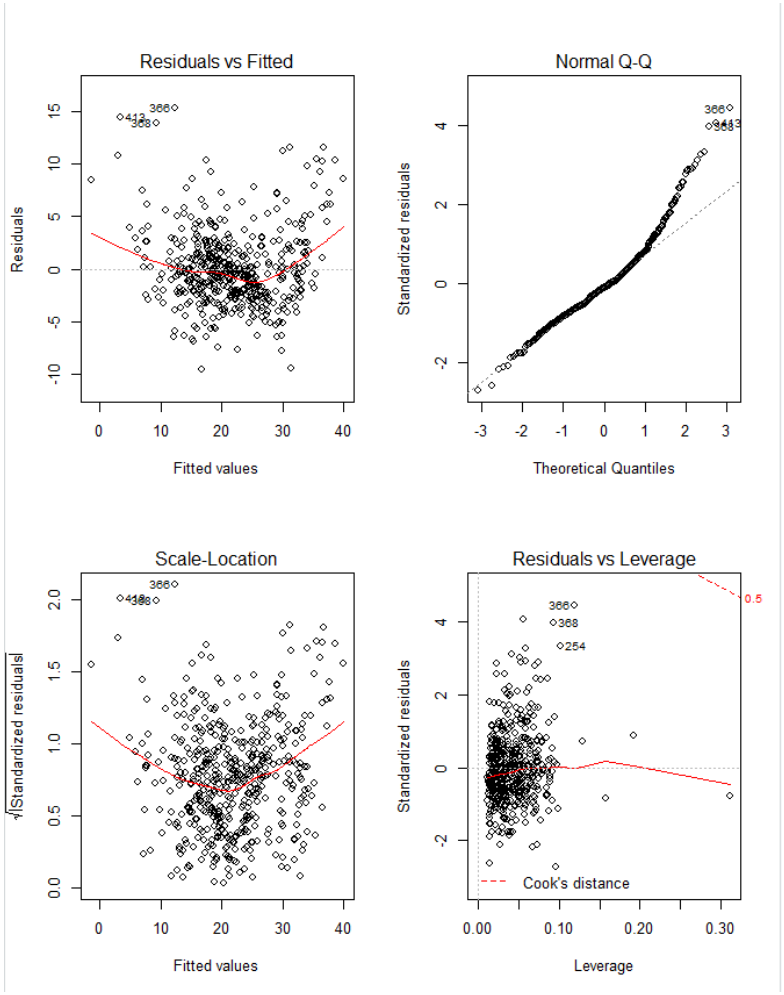
종속변수 — 독립변수들 — 종속변수, 독립변수들을 포함하는 데이터

```
m2 <- lm(medv~., data=boston)
summary(m2)

par(mfrow=c(2,2))
plot(m2)
```

```
Call:
lm(formula = medv ~ ., data = boston)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5220 -2.2592 -0.4275  1.6778 15.2894

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  30.120918   4.338656   6.942 1.29e-11 ***
crim         -0.105648   0.025640  -4.120 4.47e-05 ***
zn            0.044104   0.011352   3.885 0.000117 ***
indus        -0.046743   0.051044  -0.916 0.360274
chas1         0.158802   0.736742   0.216 0.829435
nox         -11.576589   3.084187  -3.754 0.000196 ***
rm            3.543733   0.356605   9.937  < 2e-16 ***
age          -0.026082   0.010531  -2.477 0.013613 *
dis          -1.282095   0.160452  -7.991 1.05e-14 ***
rad2          2.548109   1.175012   2.169 0.030616 *
rad3          4.605849   1.064492   4.327 1.85e-05 ***
rad4          2.663393   0.950747   2.801 0.005299 **
rad5          3.077800   0.962725   3.197 0.001483 **
rad6          1.314892   1.157689   1.136 0.256624
rad7          4.864208   1.241760   3.917 0.000103 ***
rad8          5.772296   1.194221   4.834 1.82e-06 ***
rad24         6.195415   1.417826   4.370 1.53e-05 ***
tax          -0.009396   0.003070  -3.061 0.002333 **
ptratio      -0.828498   0.114436  -7.240 1.85e-12 ***
black         0.007875   0.002084   3.779 0.000178 ***
lstat        -0.354606   0.041901  -8.463 3.36e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.671 on 469 degrees of freedom
Multiple R-squared:  0.7911,    Adjusted R-squared:  0.7821
F-statistic: 88.78 on 20 and 469 DF,  p-value: < 2.2e-16
```

2. 모델 훈련 단계

```
plot(m2)
```



실제 값과 예측 값을 나타내는 그래프
빨간 선이 약간 2차곡선의 형태를 보이긴 하나 전체적으로 점들이 골고루 퍼져 있어 등분산성을 만족하는 것으로 보임.
*빨간선 기울기가 0일 수록 좋은 모델

실제 값과 예측 값의 차이(잔차)가 정규분포를 따르는가?
45 ° 에 가까울 수록 좋은 모델

실제 값과 예측 값의 차이(잔차)가 등분산성을 따르는가?
위 그림 해석과 마찬가지.

*빨간선 기울기가 0일 수록 좋은 모델

이상치를 표현하는 그래프.

빨간 점선 안에 점들이 들어 있지 않을 수록 좋은 그래프

2. 모델 훈련 단계

다중회귀분석에서의 변수선택 방법들입니다.

- AIC나 BIC 등등의 값을 기준으로 모델의 성능이 향상되는 변수를 선택하게 됩니다.

전진 선택(forward)	상수항만 포함시킨 회귀모형에서 설명변수를 하나씩 추가하는 방법
후진 소거(backward)	모든 변수를 포함시킨 모형에서 하나씩 제거해 나가는 방법
단계별 선택법(stepwise)	전진선택과 후진소거를 왔다갔다하며 모두 쓰는 방법 전진 선택법과 후진 제거법의 장점을 더한만큼 꼼꼼하게 계산하며 계산 소요가 적고 속도면에서 부담되지 않는다면 가장 좋은 방법

2. 모델 훈련 단계

```
m2.both <- step(m2,direction="both")
m2.both
summary(m2.both)
```

```
> m2.slm

Call:
lm(formula = medv ~ crim + zn + nox + rm + age + dis + rad +
    tax + ptratio + black + lstat, data = boston)

Coefficients:
(Intercept)         crim           zn          nox           rm          age          dis         rad2         rad3         rad4
  30.252522    -0.104568     0.045510   -12.366882     3.583130    -0.025822    -1.253903     2.387130     4.644091     2.608777
       rad5         rad6         rad7         rad8        rad24          tax      ptratio        black        lstat
   3.116933     1.422890     4.868388     5.872144     6.420553    -0.010571    -0.837356     0.007949    -0.357576
```

단계적 선택법에 대한 기준으로 AIC를 사용한 결과,
필요한 변수는 21개 -> 19개 !

```
> m2.slm<-step(m2,direction = "both")
Start:  AIC=1295.03
medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
    tax + ptratio + black + lstat

          Df Sum of Sq    RSS    AIC
- chas     1      0.63 6321.5 1293.1
- indus    1     11.30 6332.2 1293.9
<none>                6320.9 1295.0
- age      1     82.67 6403.5 1299.4
- tax      1    126.28 6447.1 1302.7
- nox      1    189.88 6510.7 1307.5
- black    1    192.42 6513.3 1307.7
- zn       1    203.44 6524.3 1308.5
- crim     1    228.82 6549.7 1310.5
- rad      8    721.85 7042.7 1332.0
- ptratio  1    706.41 7027.3 1344.9
- dis      1    860.51 7181.4 1355.6
- lstat    1    965.26 7286.1 1362.7
- rm       1   1330.92 7651.8 1386.7

Step:  AIC=1293.08
medv ~ crim + zn + indus + nox + rm + age + dis + rad + tax +
    ptratio + black + lstat

          Df Sum of Sq    RSS    AIC
- indus    1     11.00 6332.5 1291.9
<none>                6321.5 1293.1
+ chas     1      0.63 6320.9 1295.0
- age      1     82.48 6404.0 1297.4
- tax      1    130.45 6451.9 1301.1
- nox      1    189.27 6510.8 1305.5
- black    1    193.59 6515.1 1305.9
- zn       1    203.76 6525.2 1306.6
- crim     1    230.58 6552.1 1308.6
- rad      8    738.26 7059.8 1331.2
- ptratio  1    719.40 7040.9 1343.9
- dis      1    861.64 7183.1 1353.7
- lstat    1    965.11 7286.6 1360.7
- rm       1   1333.37 7654.9 1384.9

Step:  AIC=1291.93
medv ~ crim + zn + nox + rm + age + dis + rad + tax + ptratio +
    black + lstat

          Df Sum of Sq    RSS    AIC
<none>                6332.5 1291.9
+ indus    1     11.00 6321.5 1293.1
+ chas     1      0.32 6332.2 1293.9
- age      1     81.09 6413.6 1296.2
- tax      1    192.78 6525.3 1304.6
- black    1    196.55 6529.0 1304.9
- zn       1    220.63 6553.1 1306.7
- crim     1    225.50 6558.0 1307.1
- nox      1    239.09 6571.6 1308.1
- rad      8    791.09 7123.6 1333.6
- ptratio  1    732.81 7065.3 1343.6
- dis      1    857.27 7189.8 1352.1
- lstat    1    987.73 7320.2 1361.0
- rm       1   1380.21 7712.7 1386.5
```

☆ 다중공선성 확인하기

다중회귀분석에서 x변수(설명변수, 독립변수)들끼리 상관관계가 존재할 경우
회귀 계수의 분산을 크게 하여, 회귀분석 시 추정 회귀 계수를 믿을 수 없게 되는 문제

```
> library(car)
> vif(m2.slm)
              GVIF Df GVIF^(1/(2*Df))
crim      1.803909  1        1.343097
zn        2.395738  1        1.547817
nox       4.259189  1        2.063780
rm        1.940485  1        1.393013
age       3.189498  1        1.785917
dis       3.992649  1        1.998161
rad      15.061518  8        1.184723
tax       8.006599  1        2.829593
ptratio   2.084007  1        1.443609
black     1.349024  1        1.161475
lstat     3.175899  1        1.782105
```

Df가 1인 수치형 변수의 경우는 좌측의GVIF

Df가 1이 아닌 범주형 변수의 경우는 우측의 GVIF^(1/2*Df))로 봐야함

이유 : VIF값이 R-Squared를 통해서 나오므로 자유도에 의해 범주형 변수의 경우는 값이 커지므로 자유도가 반영된 값으로 봐야함.

GVIF의 경우 10 이상. GVIF^(1/2*Df))의 경우 2 이상이면 다중공선성을 의심해볼 수 있음.

2. 최종 모형

```
> summary(m2.slm)

Call:
lm(formula = medv ~ crim + zn + nox + rm + age + dis + rad +
    tax + ptratio + black + lstat, data = boston)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5200 -2.2850 -0.4688  1.7535 15.3972

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.252522   4.329907   6.987 9.64e-12 ***
crim        -0.104568   0.025533  -4.095 4.96e-05 ***
zn           0.045510   0.011235   4.051 5.97e-05 ***
nox        -12.366882   2.932651  -4.217 2.97e-05 ***
rm           3.583130   0.353644  10.132  < 2e-16 ***
age         -0.025822   0.010514  -2.456 0.014412 *
dis         -1.253903   0.157029  -7.985 1.08e-14 ***
rad2         2.387130   1.160735   2.057 0.040278 *
rad3         4.644091   1.062157   4.372 1.51e-05 ***
rad4         2.608777   0.944668   2.762 0.005977 **
rad5         3.116933   0.960550   3.245 0.001258 **
rad6         1.422890   1.150280   1.237 0.216705
rad7         4.868388   1.240114   3.926 9.94e-05 ***
rad8         5.872144   1.180865   4.973 9.26e-07 ***
rad24        6.420553   1.393304   4.608 5.24e-06 ***
tax         -0.010571   0.002792  -3.787 0.000172 ***
ptratio     -0.837356   0.113420  -7.383 7.08e-13 ***
black        0.007949   0.002079   3.823 0.000149 ***
lstat       -0.357576   0.041718  -8.571  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.667 on 471 degrees of freedom
Multiple R-squared:  0.7907,    Adjusted R-squared:  0.7827
F-statistic: 98.83 on 18 and 471 DF,  p-value: < 2.2e-16
```

수치형 변수 해석

Crim : 타운 별 1인당 범죄율

타운 별 1인당 범죄율이 한 단위 증가할 때 본인소유의 주택가격이 -0.104568배 만큼 감소한다.

범주형 변수 해석

rad : 방사형 도로까지의 접근성 지수

방사형 도로까지의 접근성 지수가 2일 때가 1일 때 보다 본인소유의 주택가격이 2.387130배 만큼 증가한다.

3. 예측해보기

```
pre_medv <- predict(m2.slm,boston,interval="confidence")
pre_medv <- as.data.frame(pre_medv)
pre_medv$actual <- boston$medv
head(pre_medv)
```

Interval = "confidence" 를 넣어주면 예측값에 대한 신뢰구간을 구할 수 있습니다.

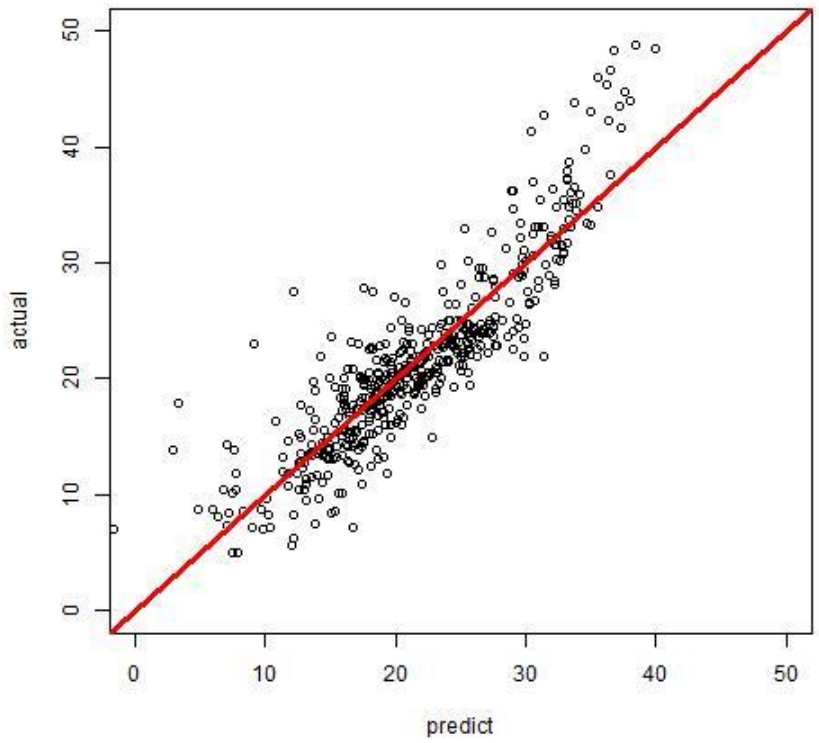
```
> head(pre_medv)
       fit      lwr      upr actual
1 26.59831 24.76235 28.43427   24.0
2 24.00195 22.37138 25.63251   21.6
3 28.99396 27.34450 30.64341   34.7
4 29.60018 28.20495 30.99540   33.4
5 29.07676 27.66851 30.48501   36.2
6 26.41636 25.04038 27.79235   28.7
```

```
> summary(pre_medv$actual)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   16.70   20.90   21.64   24.68   48.80
> with(pre_medv,sqrt(sum((actual-fit)^2)/nrow(boston)))
[1] 3.594919
```

실제값과의 차이를 보여주는 RMSE가 3.59로 비교적 좋은 예측력을 보인다고도 할 수 있다.

3. 예측해보기

```
plot(pre_medv$fit,pre_medv$actual,xlim=c(0,50),ylim=c(0,50),xlab="predict",ylab="actual")
abline(a=0,b=1,col="red",lwd=3)
```



- 예측값과 실제값의 산점도가 x=y 즉, 45도 직선에 모여있는 정도로 모형의 성능을 어느정도 예상해 볼 수 있다.
- 직선에 주로 점들이 모여있는것으로 보아 비교적 나쁘지 않은 성능을 보일 것으로 예상된다.

로지스틱 회귀 (Logistic Regression)

- **Y (Target Variable)이 범주형 (Categorical)**
- 예를 들어, Y가 0 또는 1인 경우 → Y=1이 될 확률을 $p(\boldsymbol{x})$ 라고 해보자.
- 즉, $\Pr[Y=1|X=\boldsymbol{x}]=E[Y|X=\boldsymbol{x}]$ → $p(\boldsymbol{x})=\Pr[Y=1|X=\boldsymbol{x}]$

- **이제.. 선형회귀 (Linear Regression)를 배운 학생들의 기본적인 접근은 다음과 같다.**
- 학생: $p(\boldsymbol{x})$가 x에 대한 Linear function으로 표현해볼까?

  그렇다면, $p(\boldsymbol{x})=\beta_0+\beta_1 x_1+\cdots+\beta_p x_p$ 이겠군?

- **But 이 접근에는 문제점이 있다!**

좌변의 확률 $p(\boldsymbol{x})$의 범위= [0,1]인데.. 우측의 Linear function은 unbounded (범위 제약이 없다)

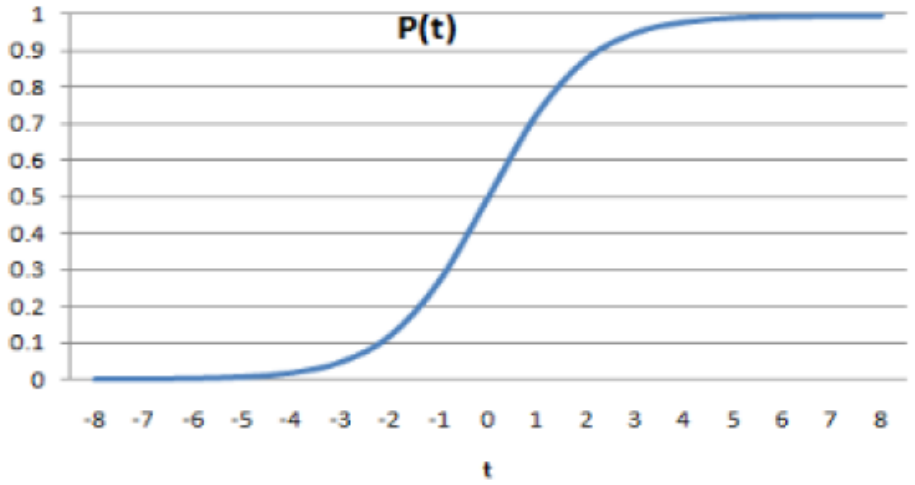
→ 이러면 계수(β)를 구해도, 그 값들이 너무 불안정해진다!

***참고: 로지스틱회귀를 R에서 돌릴때**

glm(종속변수~독립변수1+독립변수2+…독립변수n, data=데이터명, family="binomial")

로지스틱 회귀 (Logistic Regression)

- 이 문제를 해결하기 위해, 나온 것이 Logistic Function



- Linear function을 Logistic Function꼴에 집어 넣는다.
- 이를 통해, 좌변과 우변의 범위를 통일 시킨다! (양 변 다 [0,1])

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}$$

로지스틱 회귀 (Logistic Regression)

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}$$

- 이 식을, 아래와 같이 변형할 수 있다.

$$\frac{p(x)}{1-p(x)} = exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)$$

이게 **오즈비 (Odds Ratio)**;

(Y=1일 확률/Y=0일 확률)

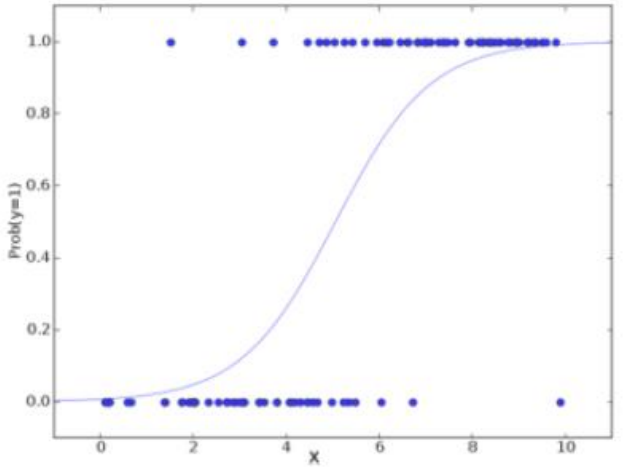
=(성공확률/실패확률)

=실패할 확률 대비 성공할 확률이 어느 정도인가?

- 양변에 로그를 씌우면, 베타 β 에 대한 Linear function형태로 표현 가능

$$ln\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

이게 Log odds = Logit



로지스틱 회귀 (Logistic Regression)

$$ln\left(\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

이게 Log odds = Logit

- 확률 $p(\boldsymbol{x})$ 가 0.5를 기준으로 할 경우..
- 아래의 기준에 따라 **최종 예측 y (predict y)**를 설정한다.

  $p(\boldsymbol{x})$ 〉= 0.5 이면 **Y=1**

  $p(\boldsymbol{x})$ 〈0.5 이면 **Y=0**

$$p(\boldsymbol{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}$$

로지스틱 회귀 (Logistic Regression)

$$ln\left(\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

이게 Log odds = Logit

- **베타 β** 의 해석 (계수의 해석)
- (1) 특정 독립변수(x)가 한 단위 변할 때, **logit이 얼만큼 변하는지**를 나타냄
  (단, 나머지 변수들 고정이라는 가정)
- (2) 특정 독립변수 x가 한 단위 변할 때, **exp(β)만큼 odds ratio(오즈비)에 영향**을 미친다.
  (단, 나머지 변수들 고정이라는 가정)

$$\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})} = exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)$$

로지스틱 회귀 (Logistic Regression)

- **Maximum Likelihood Estimation (최대 우도 추정, MLE)**

- Y=1 (성공) 또는 0 (실패) 일때, 각각의 경우에 대한 확률 (probability) 은 아래 2가지 경우밖에 없다.

  Y=1인 사람 → 성공확률 "$p(\boldsymbol{x})$"을 가진다. (획득)

  Y=0인 사람 → 실패확률 "$1-p(\boldsymbol{x})$"을 가진다. (획득)

- 따라서 각각의 사람들마다 가지는 (획득하는) 확률을 모두 고려한, **Likelihood (어떤 일이 발생할 가능성)** 는..

  (즉, n명의 사람들로 이루어진 **Sample**에서 각 사람마다 성공 또는 실패할 확률을 모두 곱한 것)

Likelihood (어떤 일이 발생할 가능성)

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \Pr[Y = y_i | X = \boldsymbol{x}_i] = \prod_{i=1}^{n} p(\boldsymbol{x}_i)^{y_i} \{1 - p(\boldsymbol{x}_i)\}^{1-y_i}$$

$$p(\boldsymbol{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}$$

로지스틱 회귀 (Logistic Regression)

- **Maximum Likelihood Estimation (최대 우도 추정, MLE)**

Likelihood (어떤 일이 발생할 가능성)

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \Pr[Y = y_i | X = \boldsymbol{x}_i] = \prod_{i=1}^{n} p(\boldsymbol{x}_i)^{y_i} \{1 - p(\boldsymbol{x}_i)\}^{1-y_i}$$

그러면, 이 **L(β)**을 최대(Maximum)로 하는 경우가, 가장 합리적!
Why? Model을 세우는 데 있어서, 어떤 일이 발생할 가능성(Likelihood)을 가장 높게 하는 것이 적절한 것이니까.
(내가 가진 한정된 자원인 sample을 이용해 가장 그럴듯한 model을 세워야하니까!)

→ Likelihood L(β)는 "β"에 따라 값이 달라지니, L을 최대화하는 β을 선택한다!
(이것이 로지스틱회귀에서 β를 선택하는 방법 = Maximum Likelihood)

$$p(\boldsymbol{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}$$

다항 로지스틱 회귀 (Multinomial Logistic Regression)

- **Y (Target Variable)이 범주형 (Categorical)**
- 하지만 , **Y가 가지는 범주의 종류가 3개 이상**인 것 !
- ex) 혈액형 (A, B, AB, O …) → 총 4 종류의 혈액형이 존재하기 때문에 이는 4개의 범주이므로 multinomial

Instead of having one set of parameters $\beta = (\beta_0, ..., \beta_p)$, each class j will have $\beta^{(j)} = (\beta_0^{(j)}, ..., \beta_p^{(j)})$

종속 변수가 2개의 범주를 가지는 logistic regression의 경우,

$$ln\left(\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

multinomial logistic regression의 경우,

$$\log\left(\frac{p(Y=j|x)}{p(Y=J|x)}\right) = \beta_0^{(j)} + \beta_1^{(j)} x_1 + \cdots + \beta_p^{(j)} x_p$$

consider J as a reference

$$p(Y=j|x) = \frac{e^{\beta_0^{(j)} + \beta_1^{(j)} x_1 + \cdots + \beta_p^{(j)} x_p}}{1 + \sum_{j=1}^{J-1} e^{\beta_0^{(j)} + \beta_1^{(j)} x_1 + \cdots + \beta_p^{(j)} x_p}}$$

다항 로지스틱 회귀 (Multinomial Logistic Regression)

종속 변수가 2개의 범주를 가지는 logistic regression의 경우,

Odds ratio between $x_k = 1$ and $x_k = 0$ (fix others): $\dfrac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k+\cdots+\beta_p x_p}}{e^{\beta_0+\beta_1 x_1+\cdots+\beta_k 0+\cdots+\beta_p x_p}}$

Interpretation of β_k:

- If we increase x_k 1, odds is increased by e^{β_k}
- $\beta_k > 0$: $p(Y = 1|x_k)$ becomes increased as x_k large
- $\beta_k < 0$: $p(Y = 1|x_k)$ becomes decreased as x_k large

그렇다면 종속변수가 **3개 이상의 범주**를 가지는 **multinomial logistic regression**의 경우에는 어떻게 비교해 ?

비교를 위한 기준으로 존재하는 것이 바로 **reference** !

$$p(Y = j|x) = \frac{e^{\beta_0^{(j)}+\beta_1^{(j)}x_1+\cdots+\beta_p^{(j)}x_p}}{1+\sum_{j=1}^{J-1} e^{\beta_0^{(j)}+\beta_1^{(j)}x_1+\cdots+\beta_p^{(j)}x_p}}$$

다항 로지스틱 회귀 (Multinomial Logistic Regression) 실습

1. 'president.csv' 데이터 불러오기

```
> #####데이터 불러오기######
> president <- read.csv("C:/Users/sseve/Desktop/president.csv",header=T)
> dim(president)
[1] 1847    6
> str(president)
'data.frame':   1847 obs. of  6 variables:
 $ 대선92: Factor w/ 3 levels "부시","클링턴",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ 나이   : int  79 32 50 56 51 48 29 40 46 37 ...
 $ 나이대: Factor w/ 4 levels "35-44세","35세 미만",..: 4 2 3 3 3 3 2 1 3 1 ...
 $ 피교육: int  12 17 6 8 17 12 13 13 13 19 ...
 $ 학력   : Factor w/ 5 levels "고등학교","고등학교 미만",..: 1 3 2 2 3 1 1 1 1 4 ...
 $ 성별   : Factor w/ 2 levels "남자","여자": 1 1 2 2 2 1 2 2 2 2 ...
```

: 1992년 미국의 대통령 선거에 투표한 사람들을 조사한 데이터

총 1847개의 관측치와 6개의 변수로 이루어져 있음 !
그리고 그 중 4개의 변수는 factor형 변수 (**범주형 변수**)

```
> apply(president[,c(1,3,5,6)],2,unique)
$대선92
[1] "클링턴" "부시"   "페롯"

$나이대
[1] "65세 이상" "35세 미만" "45-64세"   "35-44세"

$학력
[1] "고등학교"      "대학교"        "고등학교 미만" "대학원"        "초대학교"

$성별
[1] "남자" "여자"
```

종속변수는 대선92이기 때문에
3개 이상의 범주를 가지므로
multinomial logistic regression

다항 로지스틱 회귀 (Multinomial Logistic Regression) 실습

2. reference group 설정하기

```
> ####reference group(기준그룹)#######
> levels(president$대선92)
[1] "부시"   "클링턴" "페롯"
> president$대선92 <- relevel(president$대선92,ref="클링턴")
> levels(president$대선92)
[1] "클링턴" "부시"   "페롯"
> president$성별 <- relevel(president$성별,ref="여자")
> president$학력 <- relevel(president$학력,ref="대학원")
```

첫번째, levels() 함수를 통해 종속 변수의 level을 확인한다.
(factor형 변수는 level을 가짐)

여기서, 가장 먼저 제시되는 level이 multinomial logistic regression 시 **reference**로 자동 선정됨. (default reference = 가장 앞에 있는 level)
즉, 왼쪽의 코드에서 확인하면 여기서의 reference는 "부시"가 되는 것!

하지만, reference를 다른 값으로 바꿀래 ! "부시" 말고 "클링턴"을 기준으로 할래 !
그럴 때에는 **relevel() 함수**를 사용하여 **ref 인자 값**에 원하는 reference 값 입력 !

3. 다항 로지스틱 회귀 적합하기

```
> #####multinom model 구축#########
> #install.packages("nnet")
> library(nnet)
> presi.multi <- multinom(대선92~학력+성별+학력:성별,data=president) #fullmodel
# weights:  33 (20 variable)
initial  value 2029.136897
iter  10 value 1820.380884
iter  20 value 1809.843952
final  value 1809.799451
converged
> step.multi <- step(presi.multi,direction="both",trace = F) #단계선택법
trying - 학력:성별
# weights:  21 (12 variable)
initial  value 2029.136897
iter  10 value 1820.666011
final  value 1812.986476
converged
trying - 학력
trying - 성별
trying + 학력:성별
```

다항 로지스틱 회귀 (Multinomial Logistic Regression) 실습

3. 다항 로지스틱 회귀 적합하기

첫번째, multinomial logistic regression을 코드로 구현하기 위해서는 **'nnet'**라는 package가 필요하기 때문에 패키지부터 설치 ! 설치 뒤 library(nnet) 통해 불러오는 것 잊지 말기 !

두번째, nnet 패키지 안에 있는 multinom() 함수를 통해 다항 로지스틱 회귀 모델 ! 현재 설정된 독립 변수들을 살펴보면, 학력 + 성별 + 학력 : 성별 이라고 되어 있는 것을 확인할 수 있음 ! 여기서 ':' 는 교호작용을 나타내고, 교호작용이란 학력과 성별 간 존재하는 연관성을 고려한다는 것임 !

세번째, 유의미한 변수만 남기고 싶다면 **step** 함수 통하여 변수 선택 학력과 성별 변수만 남은 것을 할 수 있음 !

또한, 현재 step 함수를 사용할 때 direction의 인자를 "both"로 설정했지만 "forward", "backward"를 통해 전진선택법이나 후진제거법을 선택할 수도 있음

```
> #####multinom model 구축#########
> #install.packages("nnet")
> library(nnet)
> presi.multi <- multinom(대선92~학력+성별+학력:성별,data=president) #fullmodel
# weights:  33 (20 variable)
initial  value 2029.136897
iter  10 value 1820.380884
iter  20 value 1809.843952
final  value 1809.799451
converged
> step.multi <- step(presi.multi,direction="both",trace = F) #단계선택법
trying - 학력:성별
# weights:  21 (12 variable)
initial  value 2029.136897
iter  10 value 1820.666011
final  value 1812.986476
converged
trying - 학력
trying - 성별
trying + 학력:성별
```

다항 로지스틱 회귀 (Multinomial Logistic Regression) 실습

4. 모델 확인하기

```
> multi.sum <- summary(step.multi)
> multi.sum
Call:
multinom(formula = 대선92 ~ 학력 + 성별, data = president)

Coefficients:
     (Intercept) 학력고등학교 학력고등학교 미만 학력대학교 학력초대학교  성별남자
부시  -0.8045755    0.3871767         -0.198487  0.4243521    0.4311707 0.4581939
페롯  -2.1882810    0.8325771         -0.501696  0.8035101    1.0522364 0.7601298

Std. Errors:
     (Intercept) 학력고등학교 학력고등학교 미만 학력대학교 학력초대학교  성별남자
부시   0.1682093    0.1746745         0.2277261  0.1948072    0.2525802 0.1047093
페롯   0.2643463    0.2671955         0.3933644  0.2913039    0.3457376 0.1403827

Residual Deviance: 3625.973
AIC: 3649.973
```

Y		β	표준오차
부시	절편	-0.805	0.168
	성별=남자	0.458	0.105
	학력=초대학교	0.431	0.253
	학력=대학교	0.424	0.195
	학력=고등학교	0.387	0.175
	학력=고등학교미만	-0.198	0.228
페롯	절편	-2.188	0.264
	성별=남자	0.760	0.140
	학력=초대학교	1.052	0.346
	학력=대학교	0.804	0.291
	학력=고등학교	-0.502	0.267
	학력=고등학교미만	0.833	0.393

여기서, reference를 “클링턴”, “여자”, “대학원”으로 설정해놓았기 때문에
이 세개의 범주에 대해서는 추정 값이 부여되지 않은 것을 알 수 있음 !

어라 ? 근데 p-value가 없네 … ?

다항 로지스틱 회귀 (Multinomial Logistic Regression) 실습

5. p-value 직접 구해주기

*multinom 함수*는 유의확률(p-value)을 제공해주지 않음 ! 그렇기 때문에 직접 구해주어야 합니다.

$$Z = \frac{\hat{\beta} - 0}{\hat{\sigma}_{\hat{\beta}}} = \hat{\beta} / \hat{\sigma}_{\hat{\beta}}$$

```
> ######pvalue 구하기#######
> z <- multi.sum$coefficients/multi.sum$standard.errors
> p <- round((1 - pnorm(abs(z), 0, 1))*2,3)
> p
    (Intercept) 학력고등학교 학력고등학교 미만 학력대학교 학력초대학교 성별남자
부시           0        0.027             0.383      0.029        0.088        0
페롯           0        0.002             0.202      0.006        0.002        0
```

Y		β	표준오차	p-val
부시	절편	-0.805	0.168	0
	성별=남자	0.458	0.105	0
	학력=초대학교	0.431	0.253	0.088
	학력=대학교	0.424	0.195	0.029
	학력=고등학교	0.387	0.175	0.027
	학력=고등학교미만	-0.198	0.228	0.383
페롯	절편	-2.188	0.264	0
	성별=남자	0.760	0.140	0
	학력=초대학교	1.052	0.346	0.002
	학력=대학교	0.804	0.291	0.006
	학력=고등학교	-0.502	0.267	0.002
	학력=고등학교미만	0.833	0.393	0.202

$H_0 : \beta_i = 0$ 즉, i 번째 독립변수는 종속변수과 관계 없다.
$H_1 : \beta_i \neq 0$ 즉, i 번째 독립변수에 대한 회귀계수가 유의하다.

〈*wald test* 통하여 beta 유의성 검정〉

$z = \frac{\hat{\beta}_i - \beta_i}{SE}$는 근사적으로 N(0,1)을 따른다. 혹은 $z^2 = \left(\frac{\hat{\beta}_i - \beta_i}{SE}\right)^2$는 근사적으로 자유도1인 χ^2 분포를 따른다.

***여기서*, 학력이 고등학교 미만**일 때 유의확률은 0.383, 0.202로 0.05보다 큰 것을 확인!
즉 , **귀무 가설을 기각할 수 없음 ! →** *대학원을 졸업한* 유권자 (참조그룹) 와 *고등학교 미만의* 유권자는 지지하는 대통령후보가 다르지 않다 !

다항 로지스틱 회귀 (Multinomial Logistic Regression) 실습

6. exp(beta) 값 구하기

```
> ######exp(b)값 구하기#######
> round(exp(multi.sum$coefficients),3)
     (Intercept) 학력고등학교 학력고등학교 미만 학력대학교 학력초대학교 성별남자
부시       0.447        1.473             0.820      1.529        1.539    1.581
페롯       0.112        2.299             0.606      2.233        2.864    2.139
```

Y		β	표준오차	p-val	exp(β)
부시	절편	-0.805	0.168	0	-
	성별=남자	0.458	0.105	0	1.581
	학력=초대학교	0.431	0.253	0.088	1.539
	학력=대학교	0.424	0.195	0.029	1.529
	학력=고등학교	0.387	0.175	0.027	1.473
	학력=고등학교미만	-0.198	0.228	0.383	0.820
페롯	절편	-2.188	0.264	0	-
	성별=남자	0.760	0.140	0	2.139
	학력=초대학교	1.052	0.346	0.002	2.864
	학력=대학교	0.804	0.291	0.006	2.233
	학력=고등학교	-0.502	0.267	0.002	2.299
	학력=고등학교미만	0.833	0.393	0.202	0.606

- 고등학교, 초대학교(전문대학), 대학교 학력을 가진 유권자들의 *b의 값이 비슷*하기 때문에 세 그룹은 투표성이 비슷하다고 할 수 있다.
- 고등학교 학력의 유권자는 대학원 학력(참조그룹) 의 유권자들에 비해 클링턴보다 부시를 지지할 *odds가 약1.5배 정도* 높다.
- 남성이 여성에 비해 클링턴보다 페롯을 지지할 *odds가 약 2.1 배 정도* 높다.

다항 로지스틱 회귀 (Multinomial Logistic Regression) 실습

7. 예측하기

```
> #####model 예측#######
> head(fitted(step.multi)) #각 범주에 속할 확률
     클링턴      부시       페롯
1 0.3856708 0.4017296 0.21259952
2 0.3821835 0.4131750 0.20464150
3 0.6970404 0.2556427 0.04731685
4 0.6970404 0.2556427 0.04731685
5 0.5170391 0.3535041 0.12945683
6 0.3856708 0.4017296 0.21259952
> # predict(step.multi,president,type="probs") # 같은방법
> pred <- predict(step.multi,president)
```

- 모델 step.multi 로 예측된 각 행이 3 개의 Y범주에 속할 확률 (*가장 큰 확률을 가지는 범주*가 예측 class이다 !)
- 예측결과를 class 로 보고싶다면 predict 의 default 값 사용 (predict type을 “probs”로 설정하면 각 범주에 속할 확률)
- 예측 결과와 실제 값을 비교하여 모델의 성능을 평가해준다. confusionMatrix 함수를 사용하면 각 값을 비교할 수 있음 !

```
> library(caret)
필요한 패키지를 로딩중입니다: lattice
필요한 패키지를 로딩중입니다: ggplot2
> confusionMatrix(pred,president$대선92)
Confusion Matrix and Statistics

          Reference
Prediction 클링턴 부시 페롯
    클링턴    671  410  145
    부시      237  251  133
    페롯        0    0    0

Overall Statistics

               Accuracy : 0.4992
                 95% CI : (0.4761, 0.5222)
    No Information Rate : 0.4916
    P-Value [Acc > NIR] : 0.2649

                  Kappa : 0.095

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: 클링턴 Class: 부시 Class: 페롯
Sensitivity                0.7390      0.3797      0.0000
Specificity                0.4089      0.6880      1.0000
Pos Pred Value             0.5473      0.4042         NaN
Neg Pred Value             0.6184      0.6656      0.8495
Prevalence                 0.4916      0.3579      0.1505
Detection Rate             0.3633      0.1359      0.0000
Detection Prevalence       0.6638      0.3362      0.0000
Balanced Accuracy          0.5740      0.5339      0.5000
```

Linear Regression & Logistic Regression

감사합니다 !

김서희 | 김세정 | 김장미 | 우현우