로지스틱 회귀
카이제곱 검정 2

5조
강연욱, 김장미, 반민아, 임경룡

목 차

왈드 우도비 스코어

로지스틱 회귀 모형

AUC, ROC 커브


왈드 검정
가능도비 검정
스코어 검정

범주형 자료에서의 확률분포

베르누이 분포

$$f(x) = P\{X = x\} = \begin{cases} \pi, & if\ x = 1 \\ 1 - \pi, & if\ x = 0 \end{cases}$$

이항분포

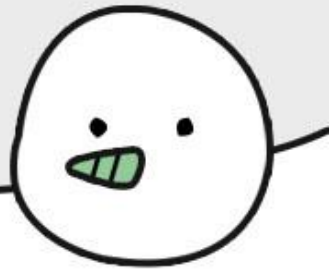
$$f(y) = P(Y = y) = \binom{n}{y}\ \pi^y (1-\pi)^{n-y},$$
$$y = 0, 1, 2, \ldots, n$$

다항분포

$$f(n_1, n_2, \ldots, n_c) = \frac{n!}{n_1!\, n_2! \cdots n_c!} {\pi_1}^{n_1} {\pi_2}^{n_2} \cdots {\pi_k}^{n_k}$$

포아송분포

$$f(x) = P(X = x) = \frac{\exp(-\mu)(\mu)^x}{x!},$$
$$x = 0, 1, 2, \ldots$$



Wald test

Wald 통계량

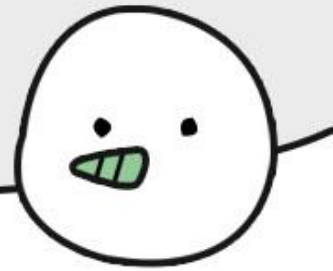
HOW

$$z = \frac{\hat{\beta} - \beta_0}{SE}, \quad z^2 \sim x_1^2$$

$$s.e. = \sqrt{\pi(1-\pi)/N}$$

최대 가능도 추정값의 표준오차 사용

π : 이항 모수 (이항분포에서 p의 부분)



Example

임신한 여인이 낙태할 권리를 찬성하는가에 대한 질문에
950명 중 400명은 "예" , 550명은 "아니오" 라고 응답.
찬성률 π 이 50%인지에 관해 유의수준 5%로 검정하여라.

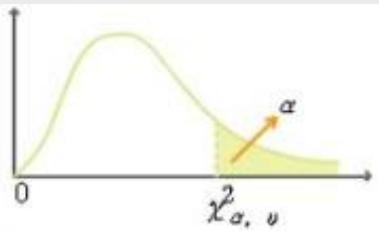
$$H_0 : \pi = 0.5$$
$$H_1 : \pi \neq 0.5$$

$$z = \frac{\hat{\beta} - \beta_0}{S.E.} = \frac{\left(\frac{400}{950}\right) - 0.5}{\sqrt{\left(\frac{400}{950}\right) * \left(\frac{550}{950}\right) / 950}}$$

Wald 신뢰구간

$$p \pm 1.96\, s.e. = p \pm 1.96 \sqrt{\frac{p(1-p)}{N}}$$

분포표



α=.05	α=.025	α=.01	α=.005	ν
3.841	5.024	6.635	7.879	1
5.991	7.378	9.210	10.597	2
7.815	9.348	11.345	12.838	3
9.488	11.143	13.277	14.860	4
11.070	12.832	15.086	16.750	5
13.592	14.449	16.812	18.548	6
14.067	16.013	18.475	20.278	7
15.507	17.535	20.090	21.955	8
16.919	19.023	21.666	23.589	9
18.307	20.483	23.209	25.188	10
19.675	21.920	24.725	26.757	11
21.026	23.337	26.217	28.300	12
22.362	24.736	27.688	29.819	13
23.685	26.119	29.141	31.319	14
24.996	27.488	30.578	32.801	15
26.296	28.845	32.000	34.267	16
27.587	30.191	33.409	35.718	17
28.869	31.526	34.805	37.516	18
30.114	32.852	36.191	38.582	19
31.410	34.170	37.566	39.997	20

Score test

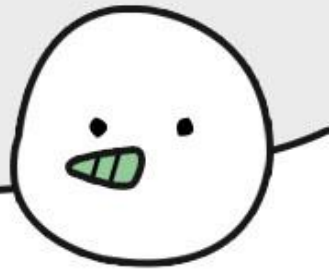
Wald test와 통계량 형태는 동일하다!

HOW

$$s.e. = \sqrt{\pi(1-\pi)/N}$$

단, 귀무가설 하에서의 표준 오차 사용

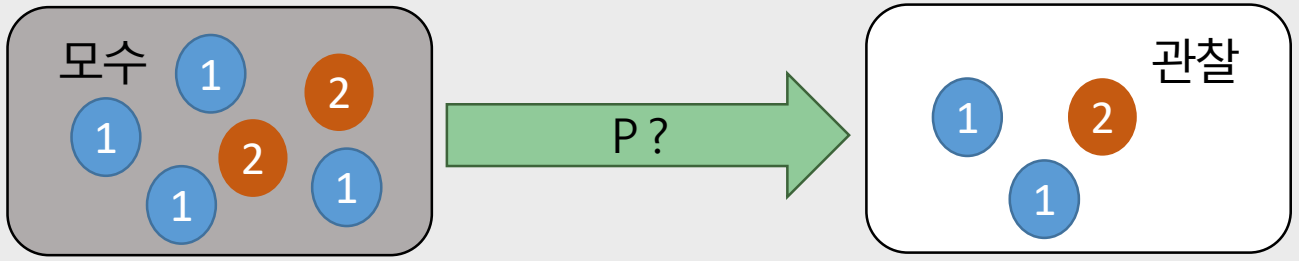
Example 의 s.e ? $\sqrt{0.5(1-0.5)/950}$



확률과 우도의 차이

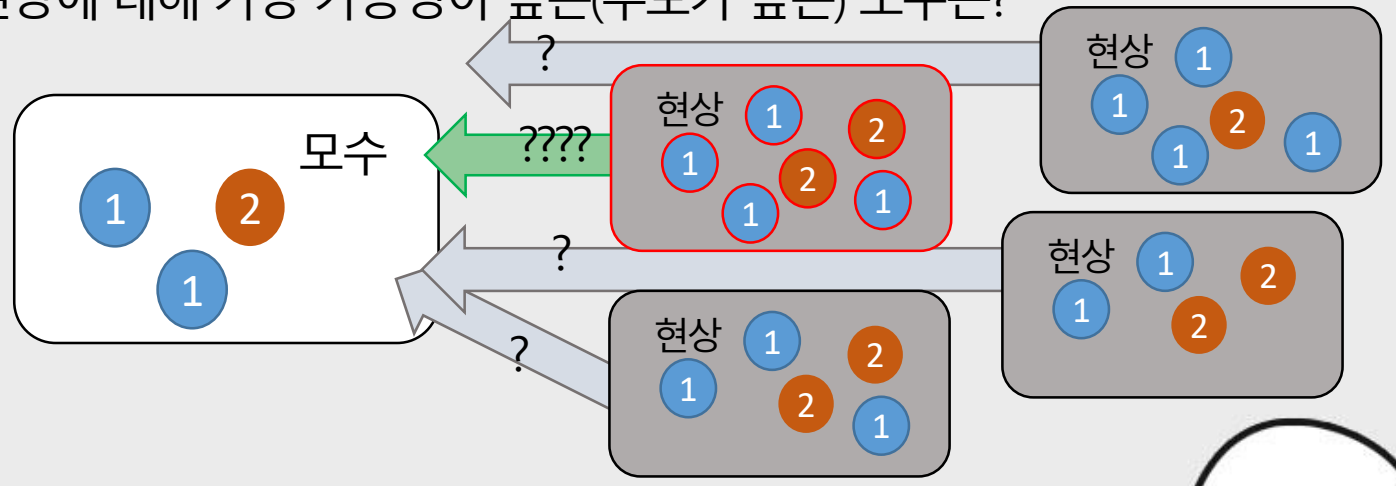
확률

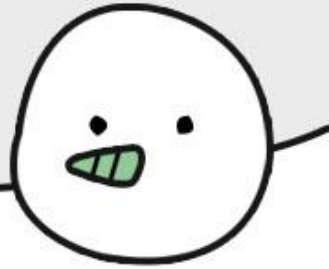
모수로부터 다음과 같이 관찰될 확률은?



우도

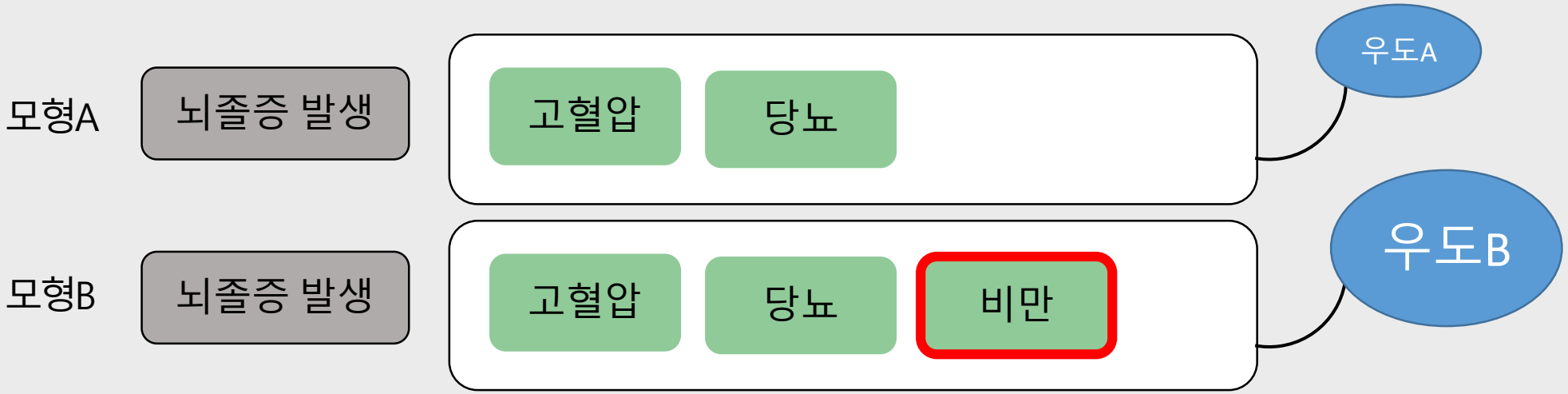
현상에 대해 가장 가능성이 높은(우도가 높은) 모수는?





우도비 검정

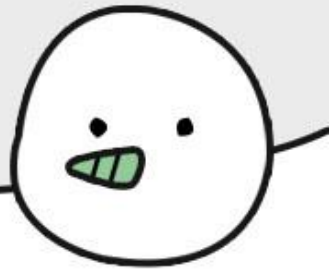
두 개의 모형의 우도의 비를 계산해서
두 모형의 우도가 유의한 차이가 나는지 비교하는 방법



☞ 모형A에서 B로 '비만'이라는 변수 추가하기!

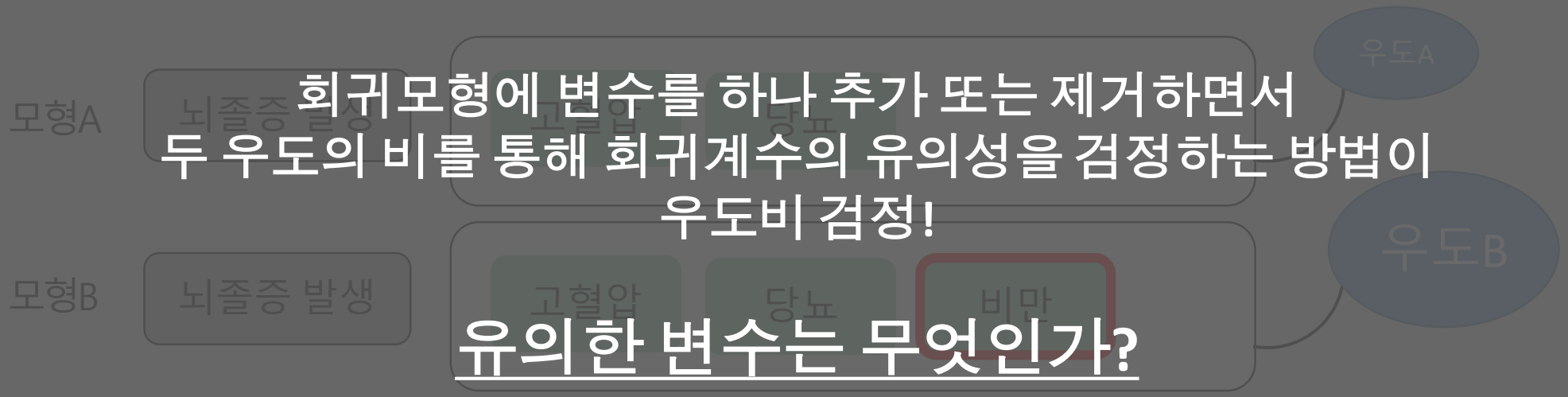
이 두 모형이 통계적으로 유의한 우도의 차이를 보인다면

비만이라는 변수는 그만큼 의미 있는 변수라고 할 수 있을 것이다



우도비 검정

두 개의 모형의 우도의 비를 계산해서
두 모형의 우도가 유의한 차이가 나는지 비교하는 방법



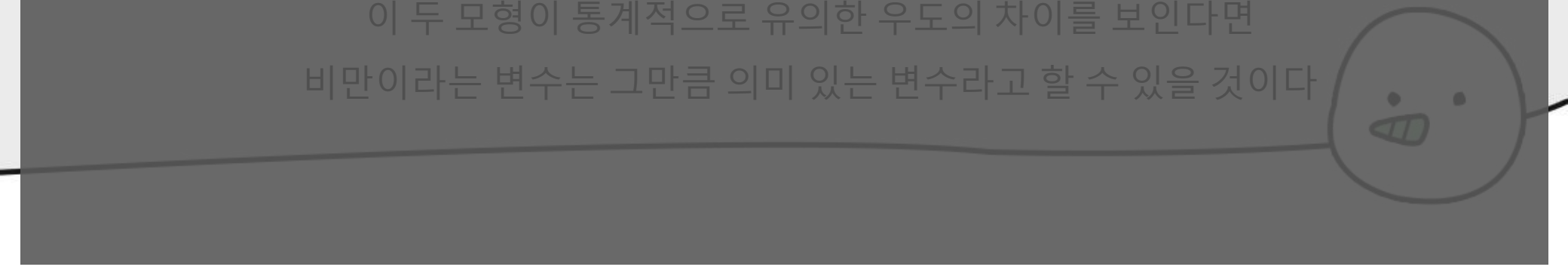
회귀모형에 변수를 하나 추가 또는 제거하면서
두 우도의 비를 통해 회귀계수의 유의성을 검정하는 방법이
우도비 검정!

유의한 변수는 무엇인가?

☞ 모형A에서 B로 '비만'이라는 변수 추가하기!

이 두 모형이 통계적으로 유의한 우도의 차이를 보인다면

비만이라는 변수는 그만큼 의미 있는 변수라고 할 수 있을 것이다



Likelihood test

Likelihood 통계량

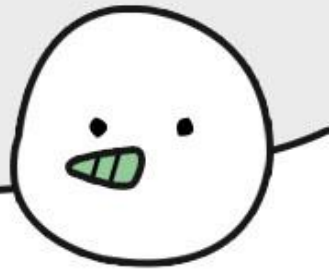
$$-2\log\left(\frac{l_0}{l_1}\right) \sim x_1^2$$

HOW

l_0 : 귀무가설 하에서 구한 가능도 함수의 최댓값

l_1 : 모든 모수 값에 대해 구한 가능도 함수의 최댓값

모형 간 설명력 검정 시 사용



- Example

새로운 치료법을 평가하기 위한 임상시험에서 10번 시행 중 9번 성공 시,

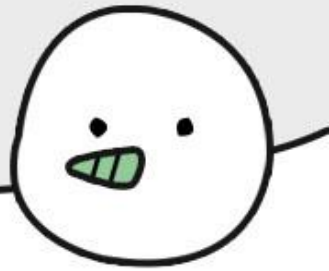
해당 치료법의 성공확률이 0.5인지에 대해 알아보자.

$$H_0 : \pi = 0.5 \quad H_1 : \pi \neq 0.5$$

통계량

$$-2 \log \frac{[\frac{10!}{9!1!}] (0.5)^9 (0.5)^1}{[\frac{10!}{9!1!}] (0.9)^9 (0.1)^1} = 7.36$$

p-value : 0.007



Wald, Score, Likelihood?

Wald

계산이 비교적 간편함
But, n=10 등의 소 표본 추정 시 신뢰도 낮음 (z 통계량)

Score

변수 진입 결정 기준으로 사용
(나머지는 변수 제거 기준)

Likelihood

모델의 변수 비교 시 주로 사용


Logistic
Regression

로지스틱 회귀(logistic regression)

영국의 통계학자인 D. R. Cox.가 1958년에 제안한 확률 모델

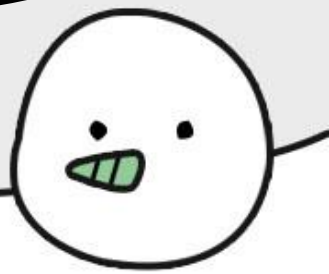
독립변수의 선형 결합을 이용하여 사건의 발생 가능성을 예측

범주형 데이터에서 정규성 가정 어려움 !
종속변수가 0 ,1 로 구분

일반 선형회귀의 정규성

이에 맞는 모델 개발 !

로지스틱 회귀

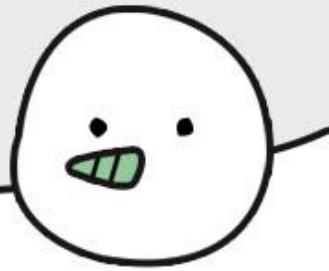


Odds : 실패에 비해 성공 확률의 비 (승산)

Odds ratio : 승산비 , 대응위험도

Example) 약 A, B의 효능 실험에서 생존, 사망연관성을 나타낼 때

B에 대한 A의 오즈비가 0.36 이다. = A의 경우 생존율이 64% 낮아진다.

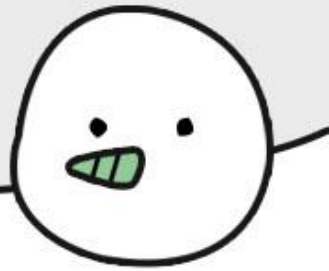


odds에 log를 취한다.

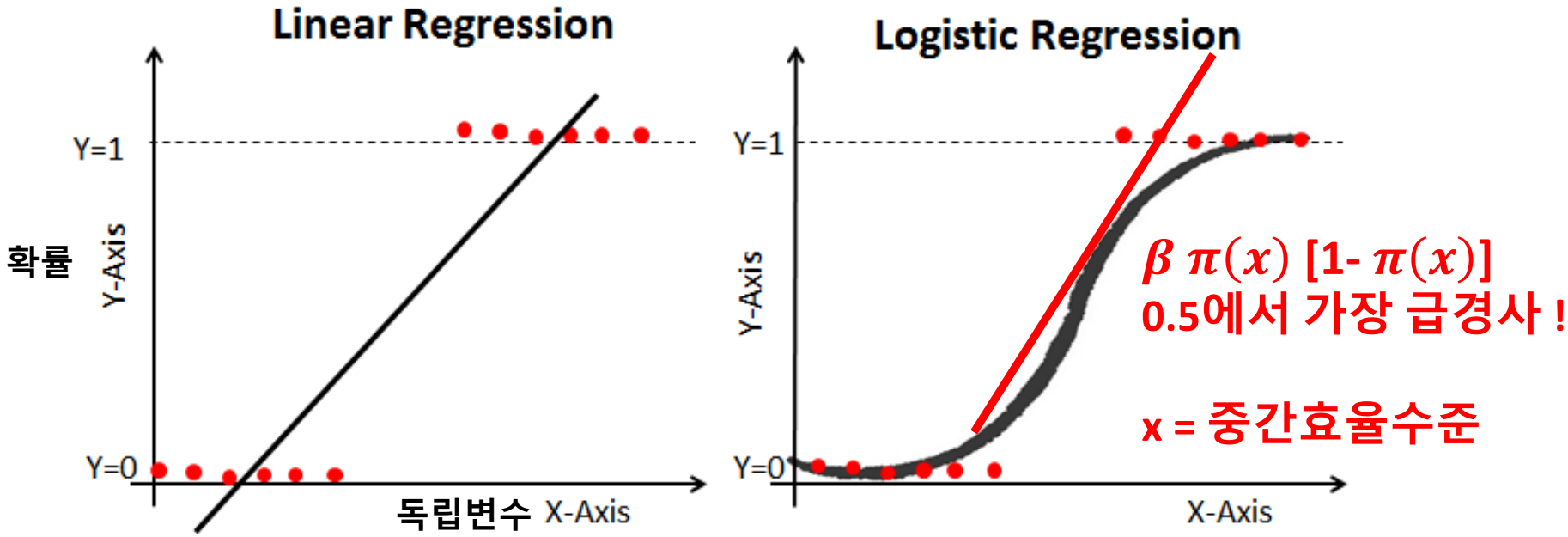
$$\text{logit}\,[\pi(x)] = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x_1 + \ \ldots + \beta_{\text{m}} x_{\text{m}}$$

$$\exp(a+\beta x) = e^{a+\beta x} \qquad \pi(x) = \frac{\exp(a+\beta x)}{1+\exp(a+\beta x)}$$

$\pi(x)$: 모형의 예측확률! $\quad 0 \le \pi(x) \le 1$



선형근사 해석



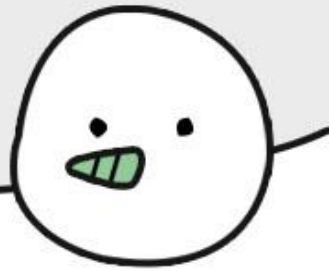
y = 0 , 1인 경우 선형회귀로 적합하는 것은 어렵다!

따라서, Logit을 이용한 곡선 적합 (x의 단위 변화에 따라 확률변화율이 다르다.)

Example

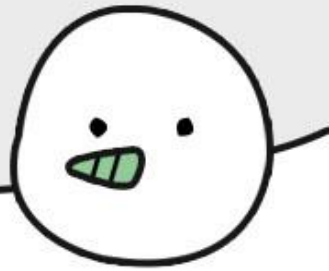
```
> data(iris)
> d <- subset(iris, Species == "setosa" | Species == "versicolor")
> str(d)
'data.frame':   100 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

예측값이 0 또는 1의 두 개로 분류되어야 하므로 임의로 “setosa”와 “versicolor”만 남긴다.



```
> d$Species <- factor(d$Species)
> str(d)
'data.frame':   100 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 2 levels "setosa","versicolor": 1 1 1 1 1 1 1 1 1 1 ...
```

다음과 같은 과정을 거쳐 2 level의 factor로 바뀌게 되었다.



```
> (model <- glm(Species ~ ., data = d, family = binomial))

Call:  glm(formula = Species ~ ., family = binomial, data = d)

Coefficients:
 (Intercept)  Sepal.Length   Sepal.Width  Petal.Length   Petal.Width
       6.556        -9.879        -7.418        19.054        25.033

Degrees of Freedom: 99 Total (i.e. Null);  95 Residual
Null Deviance:      138.6
Residual Deviance: 1.317e-09    AIC: 10
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Wald test를 통한 계수의 유의성 확인

$e^{-9.879}$: x 변수가 한 단위 증가 시, versicolor일 오즈가 약 0.00005123 배 감소한다.

이탈도 : 두 모델 비교 척도!

$-2\log\left(\frac{l_0}{l_1}\right)$: 모형 이탈도 차이

(H0 : l_0에 포함되지 않은 모수가 모두 0)

모형 적합도 : 비교적 작을수록 좋다!

(많은 모수를 가진 모형에 대한 penalty)

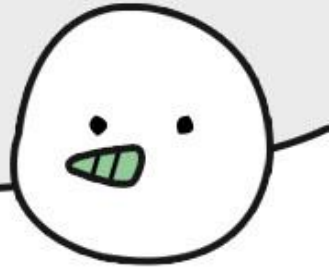
glm(Y~X , data , family="binomial")

잔차그림은 의미가 없어짐!

setosa = 0, versicolor = 1 의 확률 예측!

```
> fitted(model)[c(1:5, 51:55)]
           1            2            3            4            5           51           52           53
2.220446e-16 2.220446e-16 2.220446e-16 5.151938e-13 2.220446e-16 1.000000e+00 1.000000e+00 1.000000e+00
          54           55
1.000000e+00 1.000000e+00
```

얼마나 잘 예측되었을까??

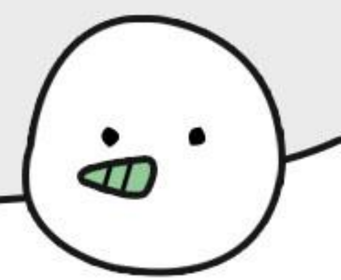


```
> f <- fitted(model)
> as.numeric(d$Species)
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [52] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

as.numeric()은 요인을 숫자로 저장한 벡터를 변환한다.
하지만 1부터 값을 부여하기 때문에, 1을 빼주어야 로지스틱 회귀분석처럼 0과 1을 갖게 된다.

```
> ifelse(f > 0.5, 1, 0) == as.numeric(d$Species) - 1
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21 
TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE 
  22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37   38   39   40   41   42 
TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE 
  43   44   45   46   47   48   49   50   51   52   53   54   55   56   57   58   59   60   61   62   63 
TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE 
  64   65   66   67   68   69   70   71   72   73   74   75   76   77   78   79   80   81   82   83   84 
TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE 
  85   86   87   88   89   90   91   92   93   94   95   96   97   98   99  100 
TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE 
```

1을 빼주게 되면 다음과 같은 결과 값을 얻을 수 있다.
이제 TRUE의 개수를 코드를 사용해 세보고 예측의 정확도를 알아보도록 하자.



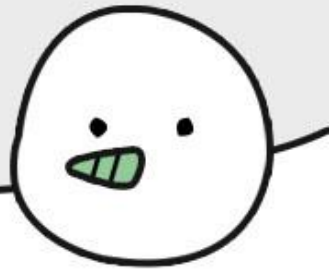
```
> is_correct <- (ifelse(f > 0.5, 1, 0) == as.numeric(d$Species) - 1)
> sum(is_correct)
[1] 100
> sum(is_correct)/NROW(is_correct)
[1] 1
```

sum() 함수는 TRUE를 1, FALSE를 0으로 취급하고, NROW()는 데이터 개수를 반환한다.

```
> predict(model, newdata = d[c(1, 10, 55), ], type = "response")
            1            10            55
2.220446e-16 2.220446e-16 1.000000e+00
```

새로운 데이터에 대한 예측은 predict() 함수를 사용한다.

type은 response로 지정하고 예측을 수행하면 0에서 1사이의 결과 값을 구해준다.



Multinomial Logistic Regression

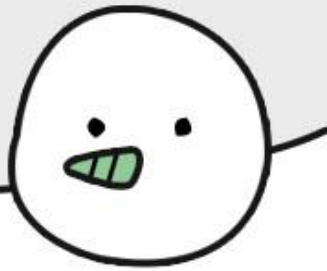
예측하고자하는 분류가 0,1 이상의 경우 사용

```
> library(nnet)
> (m <- multinom(Species ~ ., data = iris))
# weights:  18 (10 variable)
initial  value 164.791843
iter  10 value 16.177348
iter  20 value 7.111438
iter  30 value 6.182999
iter  40 value 5.984028
iter  50 value 5.961278
iter  60 value 5.954900
iter  70 value 5.951851
iter  80 value 5.950343
iter  90 value 5.949904
iter 100 value 5.949867
final  value 5.949867
stopped after 100 iterations
Call:
multinom(formula = Species ~ ., data = iris)

Coefficients:
           (Intercept) Sepal.Length Sepal.Width Petal.Length Petal.Width
versicolor    18.69037    -5.458424   -8.707401     14.24477   -3.097684
virginica    -23.83628    -7.923634  -15.370769     23.65978   15.135301

Residual Deviance: 11.89973
AIC: 31.89973
```

multinom()을 사용해 모델을 작성
nnet 패키지가 필요하다.

Setosa 를 기준점으로 계수 생성

$e^{-7.924}$ **: 해당 x 변수 한 단위 증가 시,**
Setosa 대비 virginica가 될 오즈는 약 0.00036배 감소



```
> head(fitted(m))
     setosa   versicolor    virginica
1 1.0000000 1.526406e-09 2.716417e-36
2 0.9999996 3.536476e-07 2.883729e-32
3 1.0000000 4.443506e-08 6.103424e-34
4 0.9999968 3.163905e-06 7.117010e-31
5 1.0000000 1.102983e-09 1.289946e-36
6 1.0000000 3.521573e-10 1.344907e-35
>
> apply(fitted(m), 1, max)
        1         2         3
1.0000000 0.9999996 1.0000000 0.999996
       11        12        13        1
1.0000000 0.9999997 0.9999992 0.999999
       21        22        23        2
0.9999999 1.0000000 1.0000000 0.999999
       31        32        33        3
0.9999956 1.0000000 1.0000000 1.000000
```

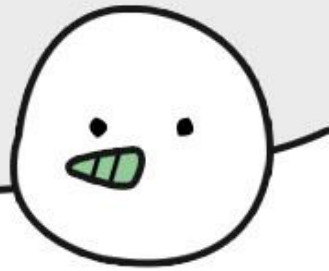
```
> a <- apply(fitted(m), 1, max)
> ifelse(a == 1, "setosa", ifelse(a == 2, "versicolor", "virginica"))
          1           2           3           4           5           6
"virginica" "virginica" "virginica" "virginica" "virginica" "virginica"
          9          10          11          12          13          14
"virginica" "virginica" "virginica" "virginica" "virginica" "virginica"
```

```
> predict(m)
  [1] setosa     setos
 [10] setosa     setos
 [19] setosa     setos
 [28] setosa     setos
 [37] setosa     setos
```

fitted()의 결과는 각 행의 데이터가 **각 분류에 속할 확률을 의미**

어떤 분류로 예측되었는지를 알아보기 위해 각 행에서 가장 큰 값이 속하는 열을 뽑을 수도 있으나, 더 간단하게 **predict()를 사용**

특히 predict()에는 newdata에 새로운 데이터를 지정할 수 있다.



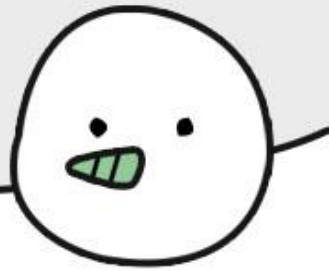
o

```
> predict(m, newdata = iris[c(1, 51, 101), ], type = "class")
[1] setosa     versicolor virginica
Levels: setosa versicolor virginica
```

default 값이 class

```
> predict(m, newdata = iris[c(1, 51, 101), ], type = "probs")
          setosa   versicolor    virginica
1   1.000000e+00 1.526406e-09 2.716417e-36
51  2.427101e-07 9.999877e-01 1.201699e-05
101 9.453717e-25 2.718072e-10 1.000000e+00
```

각 분류에 속할 **확률을 예측**하고자 한다면 **type = "probs"**를 지정

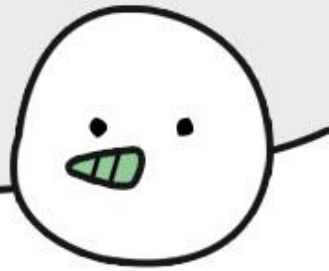


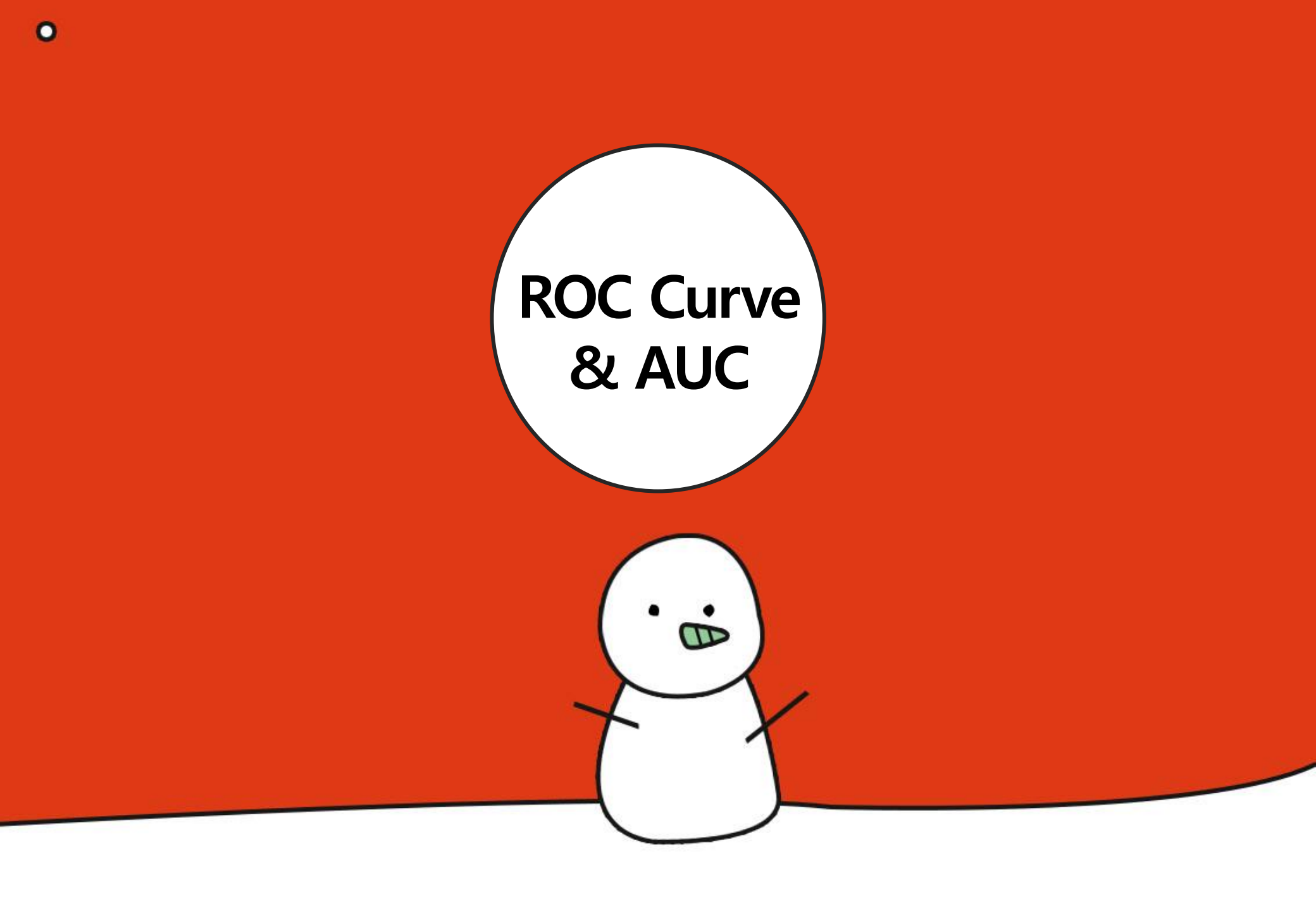
```
> predicted <- predict(m, newdata = iris)
> sum(predicted == iris$Species)/NROW(iris)
[1] 0.9866667
```

예측된 Species와 실제 Species를 비교하여 모델 정확도 판단.

```
> xtabs(~predicted + iris$Species)
            iris$Species
predicted    setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         49         1
  virginica       0          1        49
```

분류 대상이 2개 이상인 경우 분할표를 그린다.
분할표는 xtabs(도수를 나타내는 컬럼 ~ 변수 + 변수 + ...)의 형식이다.




ROC Curve
& AUC

ROC Curve

; Receiver Operating Characteristic curve

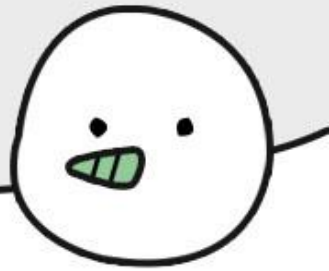
- **TPR과 FPR을 각각 x, y축으로 놓은 그래프**
- **특정 진단 방법의 민감도와 특이도가 어떤 관계를 가지고 있는지를 표현한 그래프**

(주로 역학 및 의학에서 많이 사용)

민감도 & 특이도?

민감도 (Sensitivity)
-1인 케이스에 대해 1이라고 예측한 것
ex) 진짜 환자 중 검사 방법이 환자를 얼마나 잘 분류하는가

특이도 (Specificity)
-0인 케이스에 대해 0이라고 예측한 것
ex) 진짜 비환자 중 검사 방법이 비환자를 잘 분류하는가



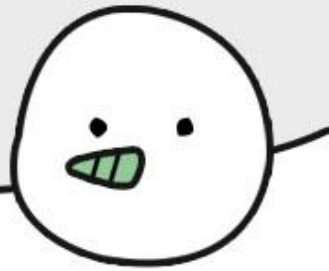
TPR & FPR?

TPR : True Positive Rate (양성율)
-1인 케이스에 대해 1로 맞게 예측한 비율
ex) 암 환자를 진찰해서 암이라고 진단 함
= 민감도 = (1 - 위음성율, true accept rate)

$$TPR = TP/(TP + FN)$$

FPR : False Positive Rate (위양성율)
- 0인 케이스에 대해 1로 잘못 예측한 비율
ex) 암 환자가 아닌데 암이라고 진단 함
= (1 – 특이도, false accept rate)

$$FPR = FP/(FP + TN)$$

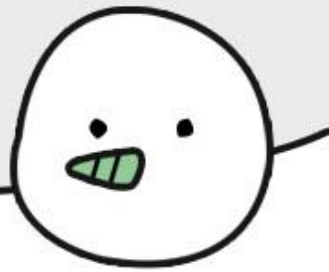


- Example : 폐암 환자 X-ray 검진

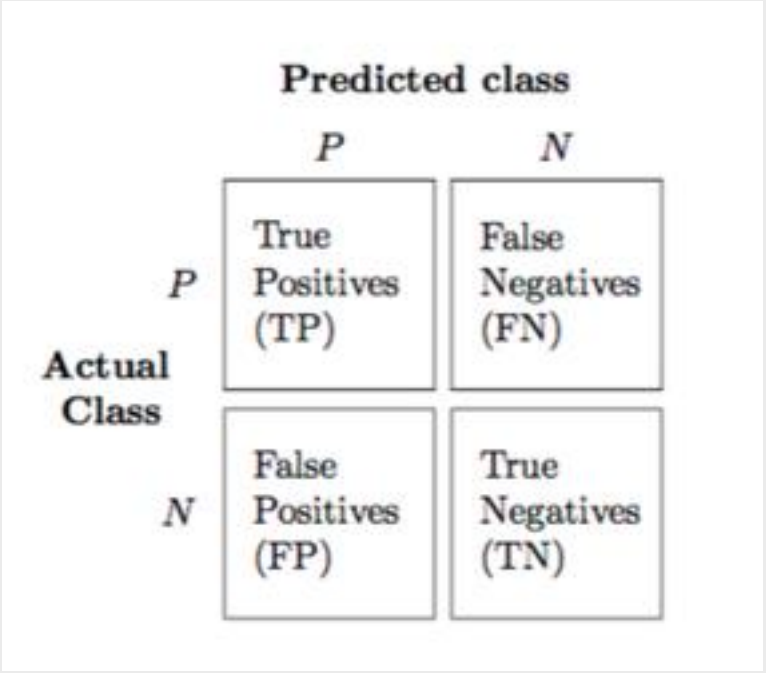
	폐암 O	폐암 X	계
X-ray 양성	90	100	190
X-ray 음성	10	800	810
계	100	900	1000

민감도 : 90/100 = 0.9 **특이도 : 800/900 = 0.89**

TPR : 0.9
FPR : 1 - 0.89 = 0.11

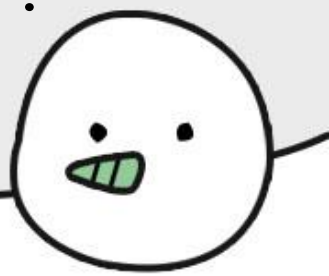


ROC Curve
어떻게 생겼으며, 왜 사용할까?

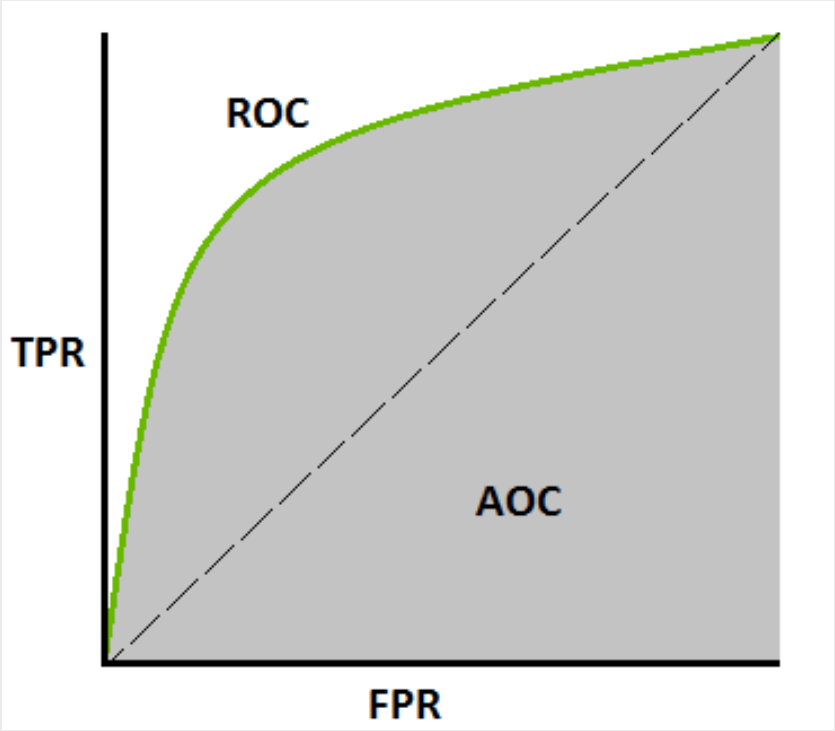


Confusion Matrix만으로는 모델의 평가 척도로 부족함 !

ROC Curve를 통한 모델의 효율성을 평가하기!

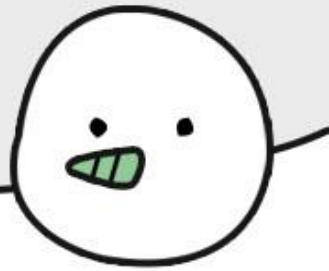


ROC Curve
어떻게 생겼으며, 왜 사용할까?



민감도(=TPR)와 1-특이도(=FPR)의 관계

녹색선의 경우가 좋은 성능으로 해석됨



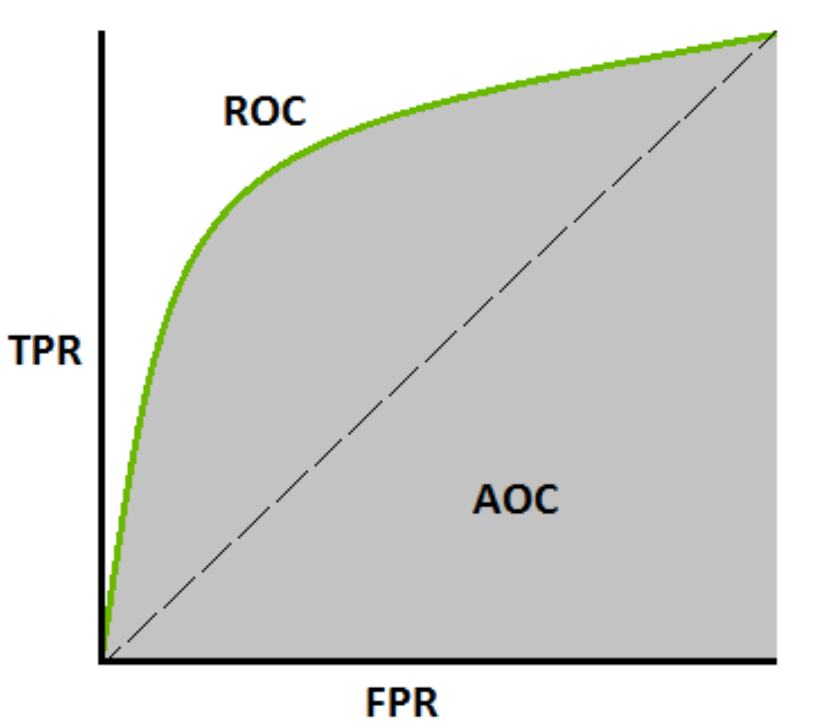
AUC

; Area Under Curve

ROC Curve와 x축이 이루고 있는 면적의 넓이

AUC

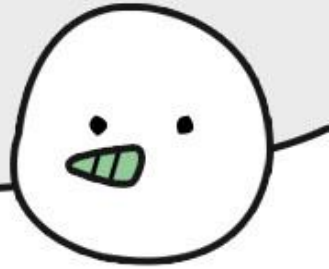
; Area Under Curve



민감도와 특이도는 반비례관계!

특이도가 주어져 있을 때, 민감도가 높을수록 더 나은 예측 검정력

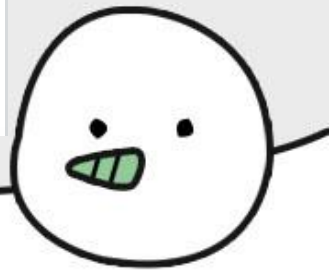
전체적인 민감도와 특이도의 상관관계를 보여줄 수 있어 편리



Example

'Chile'는 칠레의 독재자 아우구스토 피노체트가 1988년 자신의 집권 연장 찬반 여부를 묻는 투표에 대한 설문조사 데이터로써, region(지역), population(응답자 커뮤니티의 인구), sex(성별), age(연령), education(교육수준), income(수입), statusquo(현재 상황에 대한 지지도)에 따른 vote(투표경향)를 파악할 수 있다.

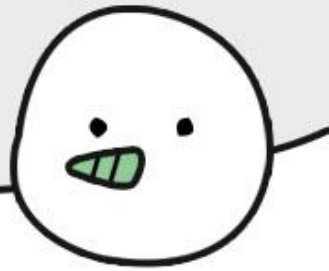
```
> library(car)
> str(Chile) #Car 패키지의 내장데이터 Chile 데이터 확인
'data.frame':   2700 obs. of  8 variables:
 $ region    : Factor w/ 5 levels "C","M","N","S",..: 3 3 3 3 3 3 3 3
 3 3 ...
 $ population: int  175000 175000 175000 175000 175000 175000 175000 1
75000 175000 175000 ...
 $ sex       : Factor w/ 2 levels "F","M": 2 2 1 1 1 1 2 1 1 2 ...
 $ age       : int  65 29 38 49 23 28 26 24 41 41 ...
 $ education : Factor w/ 3 levels "P","PS","S": 1 2 1 1 3 1 2 3 1 1
 ...
 $ income    : int  35000 7500 15000 35000 35000 7500 35000 15000 1500
0 15000 ...
 $ statusquo : num  1.01 -1.3 1.23 -1.03 -1.1 ...
 $ vote      : Factor w/ 4 levels "A","N","U","Y": 4 2 4 2 2 2 2 2 3 2
 ...
```



투표경향(vote)은 A(기권), U(보류), N(반대), Y(찬성)의 네가지 계급을 갖는다.

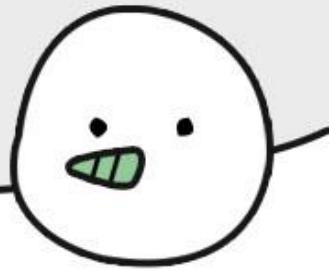
로지스틱 회귀분석을 사용하기 위해 Y 외에는 모두 N으로 바꿔준다.

```
> chile_data <- na.omit(Chile)
> chile_data$vote[chile_data$vote != 'Y'] <- 'N'
> chile_data$vote <- factor(chile_data$vote)
> head(chile_data)
  region population sex age education income statusquo vote
1      N     175000   M  65         P  35000   1.00820    Y
2      N     175000   M  29        PS   7500  -1.29617    N
3      N     175000   F  38         P  15000   1.23072    Y
4      N     175000   F  49         P  35000  -1.03163    N
5      N     175000   F  23         S  35000  -1.10496    N
6      N     175000   F  28         P   7500  -1.04685    N
```



모델의 퍼포먼스를 확인하기 위한 Train Data, Test Data의 분류

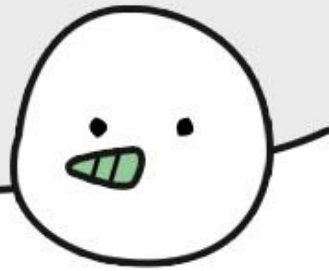
```
> chile_data_num <- nrow(chile_data)
> train_chile <- sample(1:chile_data_num) < (chile_data_num*0.8)
> test_chile <- chile_data[!train_chile,] ; head(test_chile)
   region population sex age education income statusquo vote
3       N     175000   F  38         P  15000   1.23072    Y
6       N     175000   F  28         P   7500  -1.04685    N
7       N     175000   M  26        PS  35000  -0.78626    N
12      N     175000   M  19         S  35000   1.02791    N
20      N     175000   F  50         S   2500  -1.05805    N
21      N     175000   F  38         S  35000   1.38534    Y
> train_chile <- chile_data[train_chile,] ; head(train_chile)
  region population sex age education income statusquo vote
1      N     175000   M  65         P  35000   1.00820    Y
2      N     175000   M  29        PS   7500  -1.29617    N
4      N     175000   F  49         P  35000  -1.03163    N
5      N     175000   F  23         S  35000  -1.10496    N
8      N     175000   F  24         S  15000  -1.11348    N
9      N     175000   F  41         P  15000  -1.01292    N
```



```
> outcome <- glm(vote~. ,family = binomial(), data = train_chile) ; summary(outcome)

Call:
glm(formula = vote ~ ., family = binomial(), data = train_chile)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4938  -0.4658  -0.2181   0.5491   3.0884

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.583e+00  2.948e-01  -5.369  7.9e-08 ***
regionM      5.256e-01  3.841e-01   1.369 0.171131
regionN      8.108e-01  2.417e-01   3.355 0.000794 ***
regionS      5.230e-01  1.923e-01   2.720 0.006524 **
regionSA     4.705e-01  2.445e-01   1.924 0.054336 .
population  -1.040e-06  1.008e-06  -1.032 0.302047
sexM        -1.538e-02  1.363e-01  -0.113 0.910163
age          1.653e-04  5.025e-03   0.033 0.973751
educationPS -1.803e-01  2.456e-01  -0.734 0.462844
educationS  -1.725e-01  1.627e-01  -1.060 0.288990
income       4.824e-06  1.970e-06   2.449 0.014333 *
statusquo    2.106e+00  9.240e-02  22.786  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2505.4  on 1943  degrees of freedom
Residual deviance: 1406.2  on 1932  degrees of freedom
AIC: 1430.2

Number of Fisher Scoring iterations: 5
```

Train Data를 통해 얻은 로지스틱 회귀 모형이다.

이 모델에 Test Data를 넣어 유권자들이 찬성할 확률을 계산해보자.



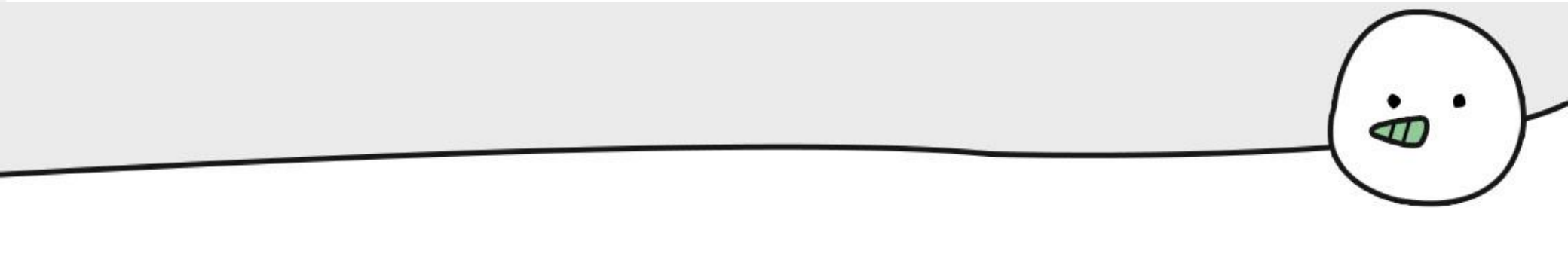
Test Data를 넣어 얻은 확률 값이다.

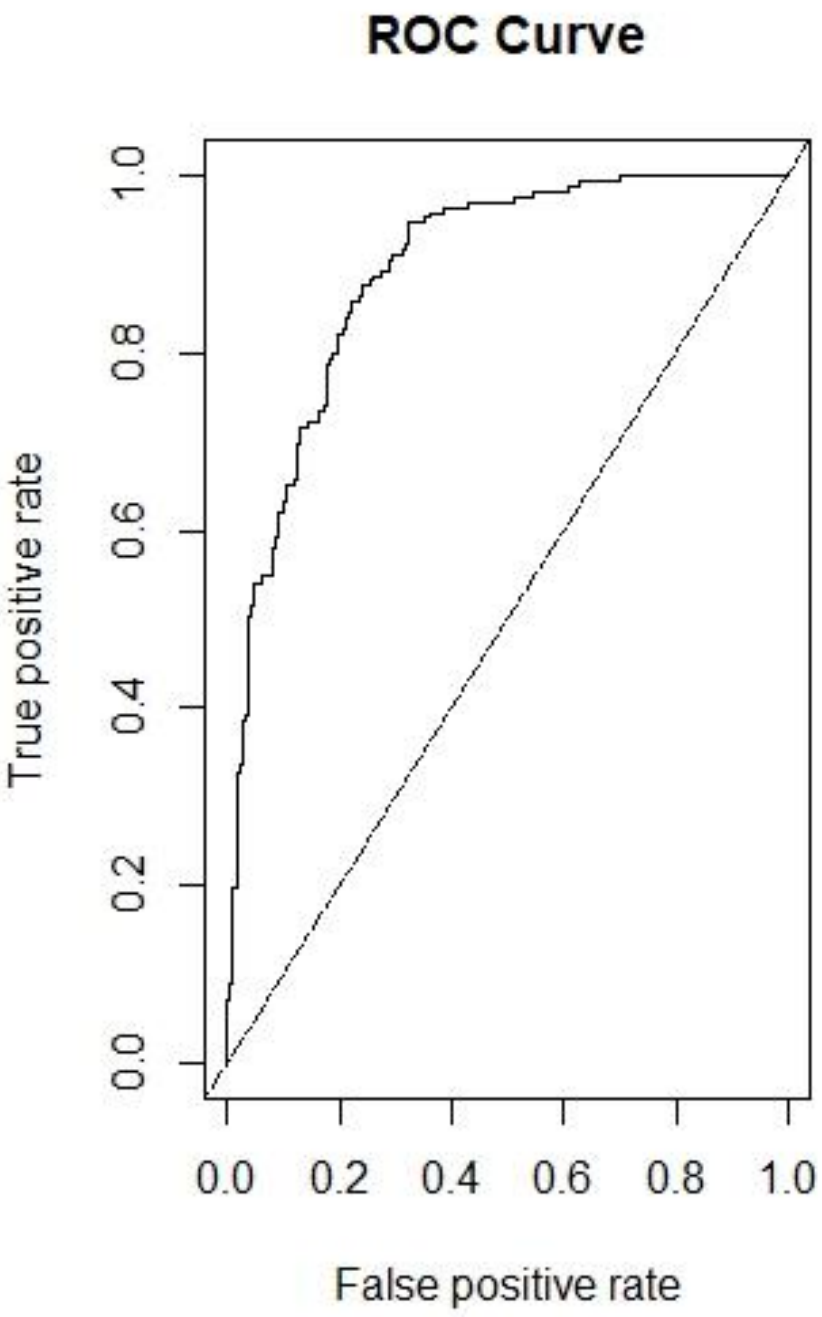
다음의 데이터를 통하여 어떤 사람이 찬성했는지 예측

```
> predict_data <- predict(outcome, newdata = test_chile, type = "response") ; predict_data
          3           6           7          12          20          21
0.847569249 0.042386451 0.067088902 0.767476027 0.034417133 0.877095184
         23          24          26          27          40          43
0.922880403 0.919680809 0.552714088 0.869991836 0.654604977 0.885716082
         59          62          77          79          85          87
0.921274234 0.669463197 0.421021963 0.272778423 0.037045448 0.206449977
         93         101         116         121         132         136
0.033616372 0.046992520 0.063139915 0.022770502 0.045544943 0.022564453
        137         138         139         144         152         155
0.039637307 0.138479429 0.117814578 0.857876890 0.040585768 0.103966287
        158         172         173         179         182         193
0.920765803 0.085014054 0.040677712 0.906542293 0.734641450 0.101543257
        196         198         217         218         225         226
0.875656326 0.885845377 0.399712493 0.050467960 0.886079155 0.368169983
```

SUMMARY

```r
outcome <- glm(vote~. ,family = binomial(), data = train_chile) ; summary(outcome)
predict_data <- predict(outcome, newdata = test_chile, type = "response") ; predict_data

install.packages("ROCR")
library(ROCR) #prediction(), performance() 함수를 사용할 수 있다.
cm <- prediction(predict_data, test_chile$vote)
#prediction() 함수는 위에서 계산한 확률 predict_data와
#실제 테스트 데이터 test_chile$vote를 비교해서 분류율을 계산해준다.
pfrf <- performance(cm, measure = "tpr", x.measure = "fpr")
#performance() 함수는 위에서 계산한 confusion matrix의
#수치 cm에서 필요한 데이터를 뽑아 plot() 함수에 넣으면 ROC 곡선을 그릴 수 있도록 하는 데이터를 반환해준다.
plot(pfrf, main = 'ROC Curve')
```

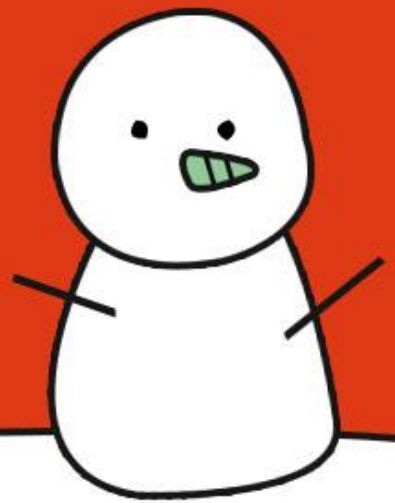




AUC 면적 구하기

```
> auc=performance(cm,measure = "auc")
> auc
An object of class "performance" Slot
"x.name":
[1] "None"

Slot "y.name":
[1] "Area under the ROC curve"

Slot "alpha.name":
[1] "none"

Slot "x.values": list()

Slot "y.values":
[[1]]
[1] 0.8931376
Slot "alpha.values":
list()
```

Thank you



아모레퍼시픽의 아리따움
폰트를 사용하여
디자인하였습니다