HW4\_Variations on Logistic Regression

Eom SangJun

2020 10 9

#1. Latent Variables

어떤 학생의 능력을 T라고 하고, 특정 문제의 난이도를 d라고 하자. 이 때 T>d일 때만 학생은 정답을 맞출 수 있다. 이제 d를 고정하고 T를 random variable로 두고 density를 f, 그리고 distribution function을 F라고 하자. 이 때 학생이 답을 틀릴 확률은

라고 할 수 있으며, 이 때 T를 latent variable이라고 한다. 즉, 직접적으로 관찰되지 않지만, 우리가 관심이 있는 변수 또는 결과값에 영향을 미치는 변수이다.

Distribution of T를 logistic이라고 한다면,

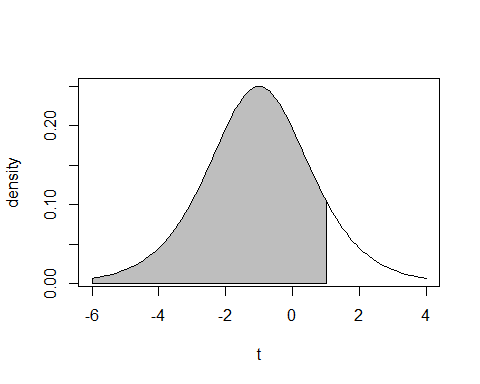
이 때, y=T, 를 의미한다.

따라서

만약 우리가 라고 둔다면 우리는 이제 logistic regression model을 갖는 것이다.

d=1, 이고 T가 -1의 평균 값을 갖는다고 하자. 그 때 T의 그래프를 그려보면 다음과 같다.

x<-seq(-6,4,0.1)  
y<-dlogis(x,location = -1)  
plot(x,y,type='l', ylab='density', xlab='t')  
ii <- (x<=1)  
polygon(c(x[ii], 1, -6), c(y[ii],0,0), col='gray')



* 그래프의 모양이 Normal Distribution과 흡사하다.
* 회색으로 칠해진 부분이 F(d)의 부분으로 이 학생이 문제를 틀릴 확률을 나타낸다.
* 문제의 난이도는 1인 반면에 T의 평균값은 -1이기 때문에 틀릴 확률이 절반 이상임을 알 수 있다.

#2. Link Functions

우리는 지금까지 logit link function만을 probability와 linear predictor를 잇는 데에 사용하였다. 하지만 다른 선택지도 존재한다.

Latent Variable formulation은 몇 개의 가능한 link function들을 제시한다. 다음의 것들은 모두 glm function 내에 내장된 것들.

1. Probit: 이 때 는 normal cumulative distribution function. 🡪 Normally distributed latent variable에서 비롯.

2. Complementary log-log: 🡪 Gumbel-distributed latent variable에서 비롯.

3. Cauchit: 🡪 Cauchy-distributed latent variable에서 비롯.

이제 각기 다른 insecticide concentrate level에 따른 insects 죽음 여부에 대한 데이터를 가지고 각 link function의 효과를 살펴보자.

data(bliss, package='faraway')  
bliss

## dead alive conc  
## 1 2 28 0  
## 2 8 22 1  
## 3 15 15 2  
## 4 23 7 3  
## 5 27 3 4

🡪 Concentrate level이 총 5개이며 level이 높아질수록 죽는 비율이 높아진다.

mlogit <- glm(cbind(dead, alive) ~ conc, family=binomial, data=bliss)  
mprobit <- glm(cbind(dead, alive) ~ conc, family=binomial(link=probit), data=bliss)  
mcloglog <- glm(cbind(dead, alive) ~ conc, family=binomial(link=cloglog), data=bliss)  
mcauchit <- glm(cbind(dead, alive) ~ conc, family=binomial(link=cauchit), data=bliss)

🡪 Link function을 달리해서 model 을 만들어보자.  
  
fitted(mlogit)

## 1 2 3 4 5   
## 0.08917177 0.23832314 0.50000000 0.76167686 0.91082823

predict(mlogit, type='response')

## 1 2 3 4 5   
## 0.08917177 0.23832314 0.50000000 0.76167686 0.91082823

coef(mlogit)[1] + coef(mlogit)[2]\*bliss$conc

## [1] -2.323790e+00 -1.161895e+00 1.332268e-15 1.161895e+00 2.323790e+00

predict(mlogit)

## 1 2 3 4 5   
## -2.323790e+00 -1.161895e+00 1.332268e-15 1.161895e+00 2.323790e+00

library(faraway)

## Warning: package 'faraway' was built under R version 3.6.3

ilogit(mlogit$lin)

## 1 2 3 4 5   
## 0.08917177 0.23832314 0.50000000 0.76167686 0.91082823

🡪 logit link를 사용했을 때의 각 level에서의 예측 확률값은 위와 같다.

이제 같은 방식으로 logit link부터 cauchit link까지 각 level에서의 예측 확률값들을 비교해보자.

predval <- sapply(list(mlogit, mprobit, mcloglog, mcauchit), fitted)  
dimnames(predval) <- list(0:4, c('logit', 'probit', 'cloglog', 'cauchit'))  
round(predval,3)

## logit probit cloglog cauchit  
## 0 0.089 0.084 0.127 0.119  
## 1 0.238 0.245 0.250 0.213  
## 2 0.500 0.498 0.455 0.506  
## 3 0.762 0.752 0.722 0.791  
## 4 0.911 0.914 0.933 0.882

🡪 많이 차이가 나지 않는다는 것을 알 수 있다.

🡪 level의 범위를 늘려서 차이가 뚜렷이 나타나도록 해보자.

dose <- seq(-4, 8, 0.2)  
predval <- sapply(list(mlogit, mprobit, mcloglog, mcauchit), function(m)  
 predict(m, data.frame(conc=dose), type='response'))  
colnames(predval) <- c('logit', 'probit', 'cloglog', 'cauchit')  
predval <- data.frame(dose, predval)  
library(tidyr)

## Warning: package 'tidyr' was built under R version 3.6.3

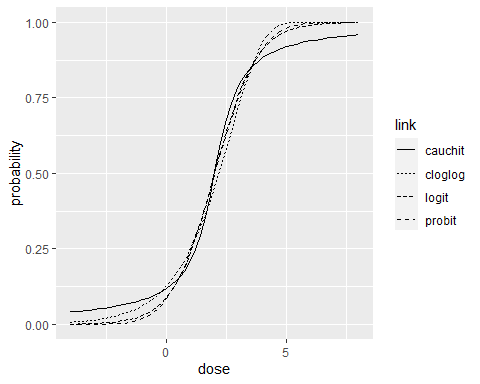
mpv <- gather(predval, link, probability, -dose)

🡪 tidyr package의 gather function은 기존의 data frame을 key값을 기준으로 새로운 형태로 정렬시켜준다.

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.3

ggplot(mpv, aes(x=dose, y=probability, linetype=link))+geom\_line()



* 0~5사이에서는 차이가 크지 않지만 양 끝으로 갈수록 차이가 많이 난다는 것을 알 수 있다.
* 그 중 Cauchit가 다른 세 개의 link function들에 비해 가장 다른 형태를 보이는데 이는 latent variable이 가장 가변적(variable)이기 때문이다. 양 끝에서 0과 1에 가장 천천히 converge 하는 모습을 보인다.
* Complementary log log도 logit과 probit에 비해서는 차이를 보이는데 logit과 probit은 거의 동일해보인다.
* 우리는 lower와 upper tail에서의 ratio of probabilities를 조사함으로써 logit과 probit의 차이를 알 수 있다.

ggplot(predval, aes(x=dose, y=probit/logit)) + geom\_line() + xlim(c(-4,0))

## Warning: Removed 40 row(s) containing missing values (geom\_path).



ggplot(predval, aes(x=dose, y=(1-probit)/(1-logit))) + geom\_line() + xlim(c(4,8))

## Warning: Removed 40 row(s) containing missing values (geom\_path).



* 만약 probit과 logit이 정말 차이가 없다면 비율은 1에 가까워야 하지만 lower과 upper tail 양 끝 쪽에 가까워질수록 1에서 멀어지는 모습을 보인다. 즉, 양 끝 쪽에서 probit과 logit은 많은 차이를 보인다.
* 이는 Complementary log log에서도 동일하게 나타난다.

그렇다면 어떤 link function을 사용해야 하는가?

우선, 비용상의 문제 등 여러 한계로 인해 데이터만 가지고 link function을 고르기란 쉽지 않다. 따라서 우리는 일반적으로 physical knowledge 등으로부터 가정된 사실들을 바탕으로 link function을 고르거나 아니면 편의상 logit link를 고른다.

Logit link의 장점은

1. probit보다 수학적으로 계산이 쉽다.

2. odds를 사용하기 때문에 해석이 쉽다.

3. retrospectively sampled data의 분석을 쉽게 해준다.

#3. Prospective and Retrospective Sampling

아기들에게 음식을 주는 방식과 respiratory disease간의 관계에 대한 데이터를 통해 prospective sampling과 retrospective sampling 간의 차이를 알아보자.

xtabs(disease/(disease+nondisease) ~ sex + food, babyfood)

## food  
## sex Bottle Breast Suppl  
## Boy 0.16812227 0.09514170 0.12925170  
## Girl 0.12500000 0.06681034 0.12598425

성별과 음식을 주는 방식에 따른 질병 발생 비율은 위와 같다.

1.prospective sampling에서는 predictor들이 고정되어 있고 outcome들이 관찰된다. 이를 cohort study라고 한다. 위의 예시에서는 음식을 주는 방식을 고정시키고 질병 발생의 비율이 어떻게 되는 지를 관찰했다면 이는 prospective sampling이다.

2. retrospective sampling에서는 반대로 outcome들이 고정되어 있고 predictor들이 관찰된다. 이를 case-control study라고 한다. 위의 예시에서는 병이 걸린 아기들과 그렇지 않은 아이들의 sample을 구한 뒤 그들의 정보를 조사했다면 이것은 retrospective sampling이다.

우리는 predictor들이 response에 어떻게 영향을 준 것인지가 궁금한 것이므로 prospective sampling이 요구될 것으로 보인다. 우선 남아이고 bottle과 breast feeding 경우만 살펴보자.

babyfood[c(1,3),]

## disease nondisease sex food  
## 1 77 381 Boy Bottle  
## 3 47 447 Boy Breast

log-odds를 사용한다면, outcome과 predictor간의 연관성이 얼마나 강한지를 알 수 있을 것이다.

1. Breast Feeding이 주어졌을 때, respiratory disease를 가질 log-odds는: log(47/447)=-2.25

2. Bottle Feeding이 주어졌을 때, respiratory disease를 가질 log-odds는: log(77/381)=-1.60

🡪 두 log-odds의 차이 = log-odds ratio = -1.6-(-2.25)=0.65

즉, bottle fed의 경우가 breast에 비해 disease에 대한 위험성이 더 크다.

그럼 retrospective sampling 관점에서는 어떠할까?

신기하게도, disease가 주어졌을 때를 가정해도 log-odds는 동일하다.

=log77/47-log381/447=log77/381-log47/447=0.65

이는 retrospective design이 prospective design만큼 log-odds ratio를 추정하는 데에 효과적이라는 것을 보여준다. 다만, probit과 같은 다른 link에 대해서는 불가능하고 오직 logit link에서만 가능하다.

Retrospective design의 장점

1. cohort는 observation들을 추적해야 해서 시간이 오래 걸리고, 비용이 많이 들지만, retrospective는 그렇지 않다.

2. predictor가 될 수 있는 것들을 많이 조사할 수 있다.

3. cohort는 rare outcome을 얻기 위해서는 매우 많은 양의 데이터를 필요로 할 수 있는 반면, retrospective는 그렇지 않다.

Prospective design의 장점

1. sample을 고르는 데에 있어서 bias가 개입할 가능성이 적다.

2. case-control에서는 주로 historical records를 참조하는데 이는 부정확하거나 불완전할 수 있다. Prospective는 이런 문제가 발생할 가능성이 적다.

3. 하나보다 더 많은 outcome의 study를 가능하게 해준다.

4. outcome의 확률을 계산할 수 있게 해준다.

어째서 prospective design에서만 outcome의 확률을 예측할 수 있는 지에 대해 알아보자.

를 병을 가지고 있지 않은 사람이 study에 포함될 확률, 를 병을 가진 사람이 study에 포함될 확률이라고 하자. Prospective design에서는 outcome에 대한 지식이 없기 때문에 이라고 본다. 반면, retrospective의 경우 일반적으로 이 보다 훨씬 낮다.

이번에는 를 어떤 사람이 study에 포함되었을 때(given), 그 사람이 병을 가지고 있을 조건부 확률이라고 하자. 또한 를 어떤 사람이 병을 가지고 있을 marginal probability 또는 비조건부 확률이라 하자.

Bayes Theorem에 의해

따라서

즉, retrospective와 prospective의 차이는 오직 라는 것을 알 수 있다. 그런데 일반적으로 는 알려져 있지 않다. 따라서 retrospective에서는 covariates의 relative effect를 알 수는 있지만, 절대적인 effect값을 알지는 못한다. 반면 prospective는 이므로 절대적인 값을 계산할 수 있다.

#4. Prediction and Effective Doses

우리는 covariates값이 주어졌을 때, outcome을 예측하고 싶을 수 있다. 예를 들어, binomial case에서는 성공확률을 예측하고 싶을 수 있다. 이 때 normal approximation을 통해 confidence interval을 구할 수도 있고, linear predictor를 inverse of the link function에 집어넣어서 확률 값을 점 추정할 수도 있다.

앞서 보았던 Insect data를 이용하여 이를 살펴보자.

lmod <- glm(cbind(dead, alive) ~ conc, family=binomial, data=bliss)  
lmodsum <- summary(lmod)

dose가 2.5일 때 insect가 죽을 확률을 구해보자.  
x0 <- c(1,2.5)  
eta0 <- sum(x0\*coef(lmod))  
ilogit(eta0)

## [1] 0.6412854

🡪 64% 확률로 죽을 것이다.

이번에는 Confidence Interval을 구해보자. 우선, 이를 위해서는 variance matrix를 구해야 한다.

(cm <- lmodsum$cov.unscaled)

## (Intercept) conc  
## (Intercept) 0.17463024 -0.06582336  
## conc -0.06582336 0.03291168

따라서 logit scale(linear predictor)의 standard error는 다음과 같다.

se <- sqrt(t(x0) %\*% cm %\*% x0) #%\*%은 행렬곱, 내적임.

이에 따라 probability scale 상의 CI를 구하면,  
ilogit(c(eta0-1.96\*se, eta0+1.96\*se))

## [1] 0.5342962 0.7358471

이렇게 하는 것이 번거로울 경우 predict command를 이용하면 쉽게 점 추정 값과 standard error값을 구할 수 있다.

predict(lmod, newdata=data.frame(conc=2.5), se=T)

## $fit  
## 1   
## 0.5809475   
##   
## $se.fit  
## [1] 0.2262995  
##   
## $residual.scale  
## [1] 1

ilogit(c(0.58095-1.96\*0.2263, 0.58095+1.96\*0.2263))

## [1] 0.5342966 0.7358478

🡪 위에서 계산한 것과 거의 동일한 결과 값을 얻을 수 있었다.

Linear Regression 상황과 달리 binomial에서는 future observation의 CI와 mean response의 CI간 차이가 없다.

우리는 이제 dose가 -5일 때를 살펴보자.

x0 <- c(1,-5)  
se <- sqrt(t(x0) %\*% cm %\*% x0)  
eta0 <- sum(x0\*lmod$coef)  
ilogit(c(eta0-1.96\*se, eta0+1.96\*se))

## [1] 2.357639e-05 3.643038e-03

절대적인 CI interval은 매우 작아보이지만, upper limit이 lower limit의 100배가 넘는다. 즉 상대적으로 봤을 때 넓다고 할 수 있다.

이번에는 p값이 고정되어 있고 그에 해당하는 covariates x 값을 찾고 싶다고 하자. 예를 들어서 p=1/2일 때의 dose값을 구하고 싶다고 하자. 이 때 그러한 dose는 ED50으로 표현한다(Effective Dose). 또는 어떤 대상을 죽이거나 하는 상황에서는LD50으로 표현하기도 한다(Lethal Dose).

P=1/2 일 때, logit(p) 값은 0이 되고 이에 따라

(ld50 <- -lmod$coef[1]/lmod$coef[2])

## (Intercept)   
## 2

🡪 우리의 데이터에서는 dose가 2일 때, 50% 확률 값을 가진다.

Standard Error를 구하기 위해서 delta method를 이용해보자.

Multivariate 에 관하여 variance of g()의 일반적인 표현은 다음과 같다.

이를 이용했을 때 standard error값은 다음과 같다.

dr <- c(-1/lmod$coef[2], lmod$coef[1]/lmod$coef[2]^2)  
sqrt(dr %\*% lmodsum$cov.unscaled %\*% dr)[,]

## [1] 0.1784367

따라서 95% CI값은,

c(2-1.96\*0.178, 2+1.96\*0.178)

## [1] 1.65112 2.34888

50%가 아니라 다른 level에 대해서도 궁금할 수 있다.

이 때 effective dose값은

90%일 때의 effective dose값을 구해보자.

(ed90 <- (logit(0.9) - lmod$coef[1])/lmod$coef[2])

## (Intercept)   
## 3.89107

MASS Package에는 편리하게도 effective dose값을 구해주는 dose.p function이 있다.

library(MASS)  
dose.p(lmod, p=c(0.5, 0.9))

## Dose SE  
## p = 0.5: 2.00000 0.1784367  
## p = 0.9: 3.89107 0.3449965

#5. Matched Case-Control Studies

Case-Control Study에서 우리는 outcome에 대한 특정 risk factor의 영향력(effect)를 알아내고자 한다. 그런데, 우리는 outcome에 다른 confounding variable들이 영향을 줄 수 있다는 것을 알고 있다. 따라서 이를 해결할 수 있는 한 가지 방법은 그 confounding variable을 찾아서 모델에 집어넣는 것이다. 그러나 confounding variable의 형태가 모델에 적합하지 않는 등의 문제가 있을 수 있다.

Matched case-control study는 이러한 문제를 보완하기 위한 방법이다. 이는 각각의 case를 하나 또는 더 많은 control과 match 시키는 것인데 이 때 control은 case와 비교했을 때 potential confounding variable의 관점에서 유사하거나 동일해야 한다. 그리고 이렇게 match한 그룹을 우리는 matched-set이라고 부른다. 이러한 matching의 효과는 우리가 측정하기 어려운 confounder들을 조정해주는 것이다.

당연히 confounding variable을 더 많이 특정할수록 case와 control을 match 시키는 것은 더욱 어려워진다. 따라서 matching requirements를 적절하게 조절할 필요가 있다.

다만 matched case-control study에도 문제점들이 있는데, 우선 matched set을 구성하는 것이 쉽지 않으며 match를 하는 데에 사용한 variable의 effect를 측정할 수 없다는 단점이 있다. 또한 이렇게 match를 시키면, random sampling의 효과가 사라지기 때문에 relative effects를 찾더라도 population group으로 확장할 수 없게 된다.

때때로 case는 rare한데, control은 쉽게 구할 수 있는 경우가 있다. 1:M design은 각각의 case에 대해 M개의 control이 있는 상황이다. 이 때 M은 일반적으로 작지만, matched set에 따라 그 크기가 매우 다양할 수 있다. 각 추가적인 control은 risk factor를 estimate하는 데에 있어서 increased efficiency를 오히려 감소시키는 결과를 가져오므로 M=5를 초과하는 것은 좋지 않다.

이제 logistic regression model을 세워보자.

i를 individual, j-th matched set에 대해 covariate vector 를 생각해보자. 는 우리가 관심있는 risk factor뿐만 아니라 우리가 adjust하고 싶지만, 모종의 이유로 matched set을 만들 때 criteria로 만들지는 못한 variable까지 포함할 것이다. Matched set은 총 n개가 있고 i=0을 case로, i=1,…,M을 control이라고 하자. 그럼 이 때 logistic regression model의 form은 다음과 같다.

는 j 번째 matched set 안에서 confounding variables의 effect를 의미한다. 그리고 이 때 conditional probability of the observed outcome은 다음과 같다.

🡪 가 사라진다는 것을 알 수 있다.

Conditional likelihood for the model은 다음과 같다.

우리는 이제 inference를 위해 standard likelihood methods를 사용할 것이다. 위의 Likelihood form은 생존분석에서 사용되는 proportional hazards model에 대한 likelihood와 형태가 동일하다. 따라서 우리는 이를 이용할 것이다.

참고로 에 대한 추정이 안 되기 때문에, 개개인의 prediction값은 구할 수 없다. 단지, s에 의해서 측정되는 relative risk만 구할 수 있다.

X-ray와 childhood acute myeloid leukemia 간의 관계에 대한 데이터로 앞서 언급했던 내용들을 살펴보자.

head(amlxray)

## ID disease Sex downs age Mray MupRay MlowRay Fray Cray CnRay  
## 1 7004 1 F no 0 no no no no no 1  
## 2 7004 0 F no 0 no no no no no 1  
## 3 7006 1 M no 6 no no no no yes 3  
## 4 7006 0 M no 6 no no no no yes 2  
## 5 7009 1 F no 8 no no no no no 1  
## 6 7009 0 F no 8 no no no no no 1

🡪 Case 한 개와 Control 한 개로 짝 지어진 matched set을 볼 수 있다. 앞에 나온 것이 case, 뒤에 나온 것이 control

🡪 여기서 나이는 단지 matching variable로서 사용되었다. 그리고 나머지 변수들은 모두 risk factor로 사용되었다.

🡪 그런데 Downs syndrome은 risk factor로 이미 알려져 있다. 그리고 subjects 중 오직 7개만 down syndrome에 해당한다는 것을 알 수 있다.

amlxray[amlxray$downs=='yes', 1:4]

## ID disease Sex downs  
## 7 7010 1 M yes  
## 17 7018 1 F yes  
## 78 7066 1 F yes  
## 88 7077 1 M yes  
## 173 7146 1 F yes  
## 196 7176 1 F yes  
## 210 7189 1 F yes

🡪 Down syndrome을 가지고 있는 경우 모두 case에 해당했다. 따라서 만약 이를 variable로 넣는다면, coefficient값이 무한대로 발산할 것이다. 따라서 이를 제외해주자.

(ii <- which(amlxray$downs=='yes'))

## [1] 7 17 78 88 173 196 210

ramlxray <- amlxray[-c(ii,ii+1),]

추가적으로 Mray, MupRay, MlowRay의 경우 각각 아이의 어머니가 한 번이라도 X-ray를 촬영했거나 upper body X-ray를 촬영했거나, lower body x-ray를 촬영했거나를 나타낸다. 이 변수들은 모두 상당히 관련이 있기 때문에 일단은 그냥 Mray만을 선택하기로 한다.

CnRay와 Cray의 경우도 CnRay는 아이가 X-ray를 정확히 몇 번을 찍었는 지를 알려주는 반면, Cray는 단지 찍은 경험이 있는 지만을 알려주므로 CnRay만을 선택한다.

Survival Package에는 proportional hazards model을 만들 수 있는 함수가 존재한다.  
  
library(survival)

##   
## Attaching package: 'survival'

## The following objects are masked from 'package:faraway':  
##   
## rats, solder

cmod <- clogit(disease ~ Sex+Mray+Fray+CnRay+strata(ID),ramlxray)

🡪 conditional logit model을 fit하기 위해서는 clogit function을 사용해야 한다.

🡪 matched set의 경우 반드시 독립변수 쪽에 strata function을 사용해서 표시를 해주어야 한다.

summary(cmod)

## Call:  
## coxph(formula = Surv(rep(1, 224L), disease) ~ Sex + Mray + Fray +   
## CnRay + strata(ID), data = ramlxray, method = "exact")  
##   
## n= 224, number of events= 104   
##   
## coef exp(coef) se(coef) z Pr(>|z|)   
## SexM 0.1563 1.1691 0.3861 0.405 0.68566   
## Mrayyes 0.2276 1.2556 0.5821 0.391 0.69573   
## Frayyes 0.6933 2.0003 0.3512 1.974 0.04839 \*   
## CnRay.L 1.9408 6.9641 0.6207 3.127 0.00177 \*\*  
## CnRay.Q -0.2480 0.7803 0.5819 -0.426 0.66993   
## CnRay.C -0.5801 0.5599 0.5906 -0.982 0.32598   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## exp(coef) exp(-coef) lower .95 upper .95  
## SexM 1.1691 0.8553 0.5486 2.492  
## Mrayyes 1.2556 0.7964 0.4013 3.929  
## Frayyes 2.0003 0.4999 1.0049 3.982  
## CnRay.L 6.9641 0.1436 2.0631 23.507  
## CnRay.Q 0.7803 1.2815 0.2495 2.441  
## CnRay.C 0.5599 1.7862 0.1759 1.781  
##   
## Concordance= 0.662 (se = 0.056 )  
## Likelihood ratio test= 20.89 on 6 df, p=0.002  
## Wald test = 14.49 on 6 df, p=0.02  
## Score (logrank) test = 18.6 on 6 df, p=0.005

🡪 성별과 Mray는 통계적으로 유의하지 않은 것으로 보인다.

🡪 Overall Test의 경우 적어도 어떤 한 variable은 통계적으로 유의함을 나타내주는 것이다. P-value들이 전부 0.05보다 낮기 때문에 적어도 하나의 variable은 통계적으로 유의하다.

🡪 Fray와 CnRay는 통계적으로 유의한데, 그 중 CnRay.L이 가장 명확하게 통계적으로 유의하다고 드러난다.

🡪 CnRay는 ordered Factor로, linear, quadratic, cubic contrast를 사용하는데, 이 중 오직 linear effect만 significant하다.

🡪 CnRay의 linear effect만 significant하기 때문에 ordered factor인 CnRay를 numeric data로 변환해서 분석을 다시 해보자. 그리고 유의하지 않았던 변수들도 빼서 진행해보자.

cmodr <- clogit(disease ~ Fray + unclass(CnRay)+strata(ID), ramlxray)  
summary(cmodr)

## Call:  
## coxph(formula = Surv(rep(1, 224L), disease) ~ Fray + unclass(CnRay) +   
## strata(ID), data = ramlxray, method = "exact")  
##   
## n= 224, number of events= 104   
##   
## coef exp(coef) se(coef) z Pr(>|z|)   
## Frayyes 0.6704 1.9550 0.3441 1.948 0.051394 .   
## unclass(CnRay) 0.8145 2.2580 0.2368 3.439 0.000584 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## exp(coef) exp(-coef) lower .95 upper .95  
## Frayyes 1.955 0.5115 0.996 3.838  
## unclass(CnRay) 2.258 0.4429 1.419 3.592  
##   
## Concordance= 0.654 (se = 0.052 )  
## Likelihood ratio test= 19.55 on 2 df, p=6e-05  
## Wald test = 14.12 on 2 df, p=9e-04  
## Score (logrank) test = 17.56 on 2 df, p=2e-04

🡪 CnRay의 값은 그대로 숫자로 대응되는 것이 아니라, 1=none, 2=1 or 2 x-rays, 3=3 or 4 x-rays, 4=5 or more x-rays라는 것을 유의하자.

🡪 그럼 이 때 인접한 category로 이동할 때(이 때 이동은 1에서 2, 2에서 3 등 올라가는 것) odds of the disease는 2.26상승한다.

🡪 유의할 것은 Fray가 이번에는 통계적으로 유의하지 않다고 나왔다는 점이다.

gmod <- glm(disease ~ Fray + unclass(CnRay), family=binomial, ramlxray)  
sumary(gmod)

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -1.16228 0.30105 -3.8607 0.0001131  
## Frayyes 0.50035 0.30780 1.6255 0.1040461  
## unclass(CnRay) 0.60054 0.17739 3.3855 0.0007106  
##   
## n = 224 p = 3  
## Deviance = 293.26338 Null Deviance = 309.38611 (Difference = 16.12272)

🡪 위의 분석은 흔히 실수할 수 있는 분석방식이다. 다른 결과를 보여준다.

우리가 비록 child가 x-ray를 찍는 것에 대한 영향을 찾아냈지만, 우리는 x-ray가 disease의 원인이라고 단정할 수 없다. 왜냐하면 무언가 문제가 있는 사람들만 보통 x-ray를 찍기 때문에, x-ray는 모종의 다른 원인적 변수와 관련이 있을 가능성이 높다.