R\_HW3\_Binomial and Proportion Responses

Eom SangJun

2020 9 26

#1. Binomial Regression Model

Chapter 2에서는 outcome이 0또는 1인 경우를 다뤘다면 이번에는 Response가 Bernoulli distributed된 것이 아니라 Binomial인 경우를 다룰 것이다. 이 때 Resoponse Y\_i의 확률은 다음과 같이 표시할 수 있다.

이 때 for 은 Response Variable은 binomially distributed

각 Response Variable 들은 독립적이라고 가정한다.

Response Variable을 구성하는 각각의 개별의 outcomes 또는 시도들(trials)은 모두 같은 q predictors( 에 종속되는데, 이 때 trials의 그룹을 covariate class라고 부른다.

Binary case에서 linear predictor를 만들면 다음과 같다.

그리고 logistic link function을 사용한다고 했을 때 log-likelihood는 다음과 같다.

이제 예시를 통해 살펴보자.

이번에 사용할 데이터는 온도와 orings에 관한 것이다. 온도가 일정 수준 이상으로 낮아지면 orings는 손상될 위험이 있고, 특정 온도에서 총 6개 중에 몇 개의 orings가 손상되었는지를 나타내는 데이터를 사용할 것이다.  
data(orings, package = 'faraway')

그리고 특정 온도에서 손상된 orings의 비율을 나타내는 그래프를 그려보자.

plot(damage/6 ~ temp, orings, xlim=c(25,85), ylim=c(0,1), xlab='Temperature',  
 ylab='Prob of damage')

우리는 특정 온도에서 oring이 몇 개나 손상될 것인지를 예측하고자 하므로 Binomial Regression을 이용해서 예측하도록 하자. 이 때 glm function에서 Response에 넣어주어야 하는 것은 두 가지이다. 하나는 success의 개수, 하나는 failure의 개수이다. 따라서 두 개의 정보를 n\*2 짜리 matrix로 만들어서 Response 자리에 넣어주어야 한다. 이렇게 했을 때 glm command를 이용해서 Binomial Regression을 적용하면 다음과 같다.  
lmod <- glm(cbind(damage, 6-damage) ~ temp, family=binomial, orings)

결과는 다음과 같다.  
sumary(lmod)

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 11.662990 3.296263 3.5382 0.0004028  
## temp -0.216234 0.053177 -4.0663 4.777e-05  
##   
## n = 23 p = 2  
## Deviance = 16.91228 Null Deviance = 38.89766 (Difference = 21.98538)

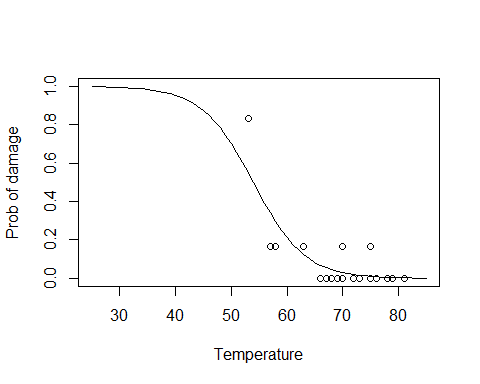
온도의 영향이 유의하다고 나온다.

Binomial Regression을 ilogit command를 이용하여 그래프를 그리면 다음과 같다.

x <- seq(25,85,1)

여기서 x는 온도로, 화씨 25도에서 85도 사이로 충분히 넓게 잡는다.

lines(x, ilogit(11.6630-0.2162\*x))



* Point 들은 온도에 따른 손상된 orings의 비율을 나타낸다. 높은 온도에서 0에 가까운 점들이 많다는 것을 알 수 있다.
* Binomial Regression을 이용해서 line을 그렸을 때 40~60 사이에서 확률이 급격히 낮아지는 것을 볼 수 있다.

31도에서의 확률을 구해보면 다음과 같다.

ilogit(11.6630-0.2162\*31)

## [1] 0.9930414

매우 높다는 것을 알 수 있다.

#2. Inference

Chapter 2에서 우리는 binomial deviance를 구하기 위해서 likelihood를 이용하였는데, 이번에도 마찬가지이다.

이 때 는 fitted value from the model

만약 model이 맞는다면, Y가 정말로 binomial이고 m\_i 가 비교적 크다면 deviance는 근사적으로 카이제곱 분포를 따른다. 이 때 자유도는 n-q-1

따라서 우리는 deviance를 model이 적절한 fit을 가졌는 지를 test하는 데에 사용할 수 있다.

p-value를 구해서 test를 진행해보자.  
pchisq(deviance(lmod), df.residual(lmod), lower=FALSE)

## [1] 0.7164099

0.05를 넘기 때문에 current model에 대한 가설을 채택한다. 즉, 현재 모델이 충분히 data에 fit한다고 판단한다. 물론 이는 우리의 모델이 정확히 정답이거나 더 간단한 모델이 틀리다는 것을 말해주지는 않는다.

만약 Null Model일 때는 어떨까? 앞선 sumary에서 Null Deviance값은 약 38.9였고 Null Model에서 q=0이므로 n-q-1=23-0-1=22. 똑같이 카이제곱 검정을 해보면,

pchisq(38.9, 22, lower=FALSE)

## [1] 0.01448877

보다시피 0.05미만이다. 따라서 적절하게 fit하지 않는다. 즉, Response가 어떠한 predictor에도 의존하지 않는 simple variation이라고는 볼 수 없다.

d의 자유도를 갖는 카이제곱 분포의 평균은 d이고 표준편차는 라는 것을 생각하면 p-value를 계산하지않아도 deviance가 큰지 작은지를 대략적으로 판단할 수 있다.

만약 deviance가 자유도보다 훨씬 크다면, 귀무가설은 기각될 수 있다.

카이제곱 분포 근사는 말그대로 근사이기 때문에 m\_i가 커질수록 정확해지고 작아질수록 부정확해진다. 따라서 m\_i=1인 경우는 완전히 실패하게 된다.

비록 m\_i가 어느정도 커야 되는 지에 대한 것은 없지만, 대략적으로 모든 i에 대해 5가 넘어가는 것이 종종 제시된다. Permutation이나 bootstrap 방법이 대안으로 제시될 수도 있다.

deviance는 두 모델을 비교하는 데에도 사용될 수 있다. 방식은 Chapter2에서 나왔던 방식과 동일하다.

Null deviance에서 current model deviance를 뺀 것과 Null model의 자유도에서 current 모델의 자유도를 뺀 것을 이용하여 카이제곱 검정을 하면,

pchisq(38.9-16.9,1,lower=FALSE)

## [1] 2.726505e-06

0.05보다 작기 때문에 온도 predictor는 유의하다는 것을 알 수 있다.

만약 어떤 covariate class에서 나온 모든 cases들을 group을 안 지으면 어떻게 될까?

erings <- with(orings,   
 data.frame(temp=rep(temp, each=6),  
 damage=as.vector(sapply(orings$damage, function(x) rep(c(0,1), times=c(6-x,x))))))  
head(erings)

## temp damage  
## 1 53 0  
## 2 53 1  
## 3 53 1  
## 4 53 1  
## 5 53 1  
## 6 53 1

결과에서 볼 수 있듯이, 특정 온도에서 각각의 시행에 대한 결과가 합쳐지는 것이 아니라 따로 나온다는 것을 알 수 있다.

emod <- glm(damage ~ temp, family=binomial, erings)  
sumary(emod)

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 11.662990 3.296157 3.5384 0.0004026  
## temp -0.216234 0.053175 -4.0665 4.773e-05  
##   
## n = 138 p = 2  
## Deviance = 54.75942 Null Deviance = 76.74480 (Difference = 21.98538)

이를 통해 binomial regression을 하고 결과를 확인해보면, parameter estimates, standard errors, deviance difference는 같게 나와서 결론은 똑같게 할 수 있다. 그러나 deviance의 절대값이 다르고 n의 개수도 차이가 난다. 우리는 이 버전으로는 residual deviance를 이용한 goodness of fit test를 할 수 없다.

각 coefficient의 신뢰구간을 구하면 다음과 같다.

confint(lmod)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 5.575195 18.737598  
## temp -0.332657 -0.120179

#3. Pearson's Chi-square Statistic  
deviance 말고도 model fit을 측정할 수 있는 대안으로는 Pearson’s Chi-square statistic이 있다.

Pearson’s Chi-square statistic의 일반적인 form은 다음과 같다.

이 때 O\_i는 관측된 수이고 E\_i는 기대값이다.

Binomial case에 적용하면, for successes, for failures이다. 따라서

만약 우리가 Pearson residual을 다음과 같이 정의한다면,

꼴이며, 이는 일종의 standard residual이라고 볼 수 있다. 그러면 이다. 따라서 Pearson’s chi-square statistic은 normal linear model에서의 residual sum of squares와 대응된다고 볼 수 있다. 우리는 deviance 때와 마찬가지로, 똑같은 null distribution에서 deviance 대신에 을 사용하여 test를 진행할 수도 있다. 다만, 이 때 주의해야 하는 것은 우리의 모델이 이 아닌 deviance를 최소화하는 것에 맞춰진 model이라는 것이다. 즉, 은 predictor가 추가될수록 증가할 수 있다.

우리의 모델에서 의 값은 28.067이다. 이를 deviance와 비교해보자.

deviance(lmod)

## [1] 16.91228

28.067과는 좀 차이가 있는 것을 알 수 있다.

그렇다면 을 이용해서 test를 진행한 결과는 어떨까?

1-pchisq(28.067, 21)

## [1] 0.1382613

다행히 p-value가 0.05이상이므로 deviance를 이용하여 test한 결과와 같은 결론을 내릴 수 있

다.

#4. Overdispersion

우리는 가끔, 모델이 맞다고 가정했을 때보다 더 큰 deviance를 관찰할 때가 있다. 이 때 우리는모델 가정에서 무엇이 잘못되었는 지를 확인해볼 필요가 있다.

우선 가장 일반적인 설명은 model의 구조적인 형태(structural form)이 잘못되었다는 것이다. 적절하지 않은 predictor를 넣었을 수도 있고, predictor들을 적당한 형태로 변형 또는 결합하지 않았을 수 있다.

또 다른 일반적인 설명은, 적은 수의 outlier들의 존재이다. 이는 diagnostic methods를 이용해서 간단하게 확인할 수 있다. 만약 더 많은 수의 outlier들이 발견된다면, 그들은 예외적인 것이 아니게 되며, 따라서 error distribution에 잘못된 것이 있다고 결론지을 수 있다.

Group size가 너무 작은 경우에도 큰 deviance를 초래할 수 있다.

이 모든 경우들을 제외하고 나서 또 다른 경우의 수는 model의 random part에 결함이 있다는 것이다.

Binomial case에서 var Y = mp(1-p)여야 한다. 이 때 m은 group의 size. 그런데 종종 직접 계산한 variance 값이 mp(1-p)값보다 훨씬 클 때가 있다. 이를 overdispersion이라고 부른다.

Overdispersion이 일어나는 주요한 이유는 두 가지이다. 우선 우리는 그룹 안에서 success또는 failure가 발생하는 case또는 probability가 independent하고 identical 하다고 가정한다. 하지만 이 가정이 틀린 경우 일어날 수 있다. 우선 constant p assumption을 살펴보자. 우리는 각 그룹에서 p값이 동일하다고 생각한다. 그러나 그룹별로 설명되지 않은 이질성이 있을 수 있고, 이는 p의 변동을 불러올 수 있다. 또한 이러한 이질성은 clustering에서 발생될 수 있다.

Sample size를 m, cluster size를 k, 그리고 cluster의 개수를 l=m/k 라고 하자. 그러면 number of success in cluster i를 라고 정의할 수 있다. 이제 가 constant가 아니라 random variable이라고 해보자. 라고 하자. 그러면 total number of success = Y = 이라고 할 때

즉, 평균은 standard case와 동일하다.

그러나, 분산은 조금 다르다.

이기 때문에 Y는 overdispersed 되었다.

Overdispersion은 trial간의 dependency 때문에 일어날 수도 있다.만약 response가 동일한 cause를 가진다면, response들은 positively correlated될 수 있다.

Overdispersion을 modeling하는 가장 간단한 방법은 추가적인 dispersion parameter()를 도입하는 것이다. 즉, Standard case에서는

는 Pearson Chi-square statistic을 이용해서 추정할 수 있다.

이 때 대신 deviance를 쓰는 것은 추천하지 않는다. 왜냐하면 consistent하지 않을 수 있기 때문. Beta의 추정은 가 response mean을 바꾸지는 않기 때문에 변화 없다. 그러나 분산의 추정은 다르다.

따라서 우리는 standard error를 의 요소만큼 늘려주어야 한다.

모델 비교에서도 deviance의 차이를 그대로 사용할 수는 없다. 왜냐하면 test statistic이 분포를 따르기 때문이다. 대신 F statistic을 사용해야 한다.

이 통계량은 근사적으로 F분포를 따른다.

이제 예시를 통해 Overdispersion case를 살펴보자.

Troutegg 데이터는 지역과 period 별로 송어 알의 생존 정도를 나타내주는 데이터이다. Total은 묻힌 송어알의 총 개수이고 survive는 그 중 몇 개가 살아남았는 지를 의미한다.

Data의 형태는 다음과 같다.  
data(troutegg, package='faraway')  
ftable(xtabs(cbind(survive, total) ~ location+period, troutegg))

## survive total  
## location period   
## 1 4 89 94  
## 7 94 98  
## 8 77 86  
## 11 141 155  
## 2 4 106 108  
## 7 91 106  
## 8 87 96  
## 11 104 122  
## 3 4 119 123  
## 7 100 130  
## 8 88 119  
## 11 91 125  
## 4 4 104 104  
## 7 80 97  
## 8 67 99  
## 11 111 132  
## 5 4 49 93  
## 7 11 113  
## 8 18 88  
## 11 0 138

이를 이용하여 binomial regression을 진행하면 다음과 같다.

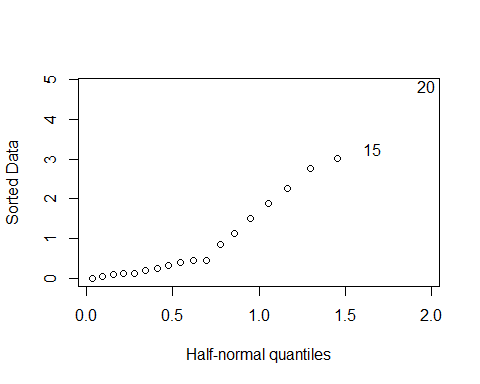
bmod <- glm(cbind(survive, total-survive) ~ location+period, family=binomial, troutegg)  
sumary(bmod)

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 4.63582 0.28132 16.4790 < 2.2e-16  
## location2 -0.41678 0.24610 -1.6936 0.09035  
## location3 -1.24208 0.21944 -5.6603 1.511e-08  
## location4 -0.95086 0.22876 -4.1566 3.230e-05  
## location5 -4.61381 0.25021 -18.4394 < 2.2e-16  
## period7 -2.17018 0.23840 -9.1031 < 2.2e-16  
## period8 -2.32562 0.24295 -9.5726 < 2.2e-16  
## period11 -2.44995 0.23410 -10.4656 < 2.2e-16  
##   
## n = 20 p = 8  
## Deviance = 64.49512 Null Deviance = 1021.46868 (Difference = 956.97356)

Deviance가 64.5 on 12 degrees of freedom 인 것을 봤을 때, model이 fit하지 않다는 것을 알 수 있다. Overdispersion임을 결론짓기 전에 다른 가능성들을 살펴보자.

우선 Outlier를 살펴보자. Outlier는 halfnorm command를 이용하여 살펴볼 수 있다.

halfnorm(residuals(bmod))



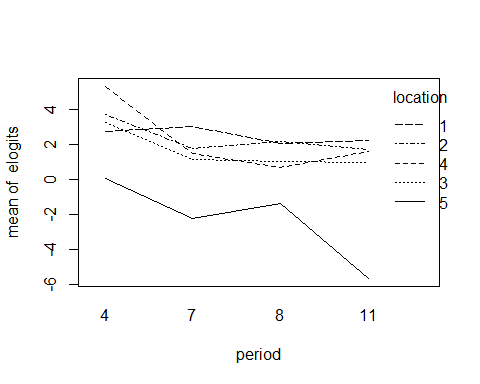
* 명확한 single outlier는 없는 것 같다.

우리는 또한 predictor들이 제대로 표현되었는 지를 empirical logits을 그려보고 확인할 수 있다.

Empirical logits =

이 때 1/2은 group이 모두 성공이거나 실패여서 infinite값이 생기는 것을 방지하기 위해서 넣어준다. 이제 empirical logits의 interaction plot을 그려보면 다음과 같다.

elogits <- with(troutegg, log((survive+0.5)/(total-survive+0.5)))  
with(troutegg, interaction.plot(period, location, elogits))



* 확실하게 뚜렷한 interaction은 보이지 않는다.
* 따라서 linear model을 설정한 것이 부적절한 것 같지 않다.

다른 가능성들을 배재했으니 이제 overdispersion에 대해 고려해보자. Overdispersion이 일어날 수 있는 가능성들은 다양하다. 송어 알들의 이질성이 있을 수 있고, 실험 과정에서의 variation이 있을 수 있다는 점도 overdispersion을 야기할 수 있다.

Overdispersion parameter를 추정하면 다음과 같다.

(sigma2 <- sum(residuals(bmod, type='pearson')^2)/12)

## [1] 5.330322

이는 standard binomial GLM에서의 것보다 상당히 크다는 것을 알 수 있다.

이제 predictor에 대한 F-test를 해보자.

Overdispersion의 경우 scale argument를 추가해주어야 한다.

drop1(bmod, scale=sigma2, test='F')

## Warning in drop1.glm(bmod, scale = sigma2, test = "F"): F test assumes  
## 'quasibinomial' family

## Single term deletions  
##   
## Model:  
## cbind(survive, total - survive) ~ location + period  
##   
## scale: 5.330322   
##   
## Df Deviance AIC F value Pr(>F)   
## <none> 64.50 157.03   
## location 4 913.56 308.32 39.494 8.142e-07 \*\*\*  
## period 3 228.57 181.81 10.176 0.001288 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Location과 period 모두 유의하다는 것을 알 수 있다.

또한 warning message에서 볼 수 있듯이, quasi-binomial GLM을 사용하였다.

Free dispersion parameter를 가졌기 때문에 이제 goodness of fit test는 불가능하다.

우리는 이제 dispersion parameter를 이용하여 standard error 추정치를 증가시킬 수 있다.

sumary(bmod, dispersion=sigma2)

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 4.63582 0.64949 7.1376 9.494e-13  
## location2 -0.41678 0.56817 -0.7335 0.46323  
## location3 -1.24208 0.50663 -2.4517 0.01422  
## location4 -0.95086 0.52814 -1.8004 0.07180  
## location5 -4.61381 0.57768 -7.9868 1.385e-15  
## period7 -2.17018 0.55040 -3.9429 8.051e-05  
## period8 -2.32562 0.56090 -4.1462 3.380e-05  
## period11 -2.44995 0.54047 -4.5330 5.815e-06  
##   
## Dispersion parameter = 5.33032  
## n = 20 p = 8  
## Deviance = 64.49512 Null Deviance = 1021.46868 (Difference = 956.97356)

이전 결과와는 다르게 오직 5번째 location만 유의한 것으로 나타났다.

이 dispersion parameter method는 covariate class들의 사이즈가 거의 동일할 때만 적절하게 사용 가능하다. 만약 그렇지 않은 경우 더 정교한 방법이 필요한데, R에서는 dispmod package 안에 있는 glm.binomial.disp command를 사용하여 적용할 수 있다.

library(dispmod)

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

dmod <- glm.binomial.disp(bmod)

##   
## Binomial overdispersed logit model fitting...  
## Iter. 1 phi: 0.03983754   
## Iter. 2 phi: 0.03813596   
## Iter. 3 phi: 0.03814806   
## Iter. 4 phi: 0.03814797   
## Iter. 5 phi: 0.03814797   
## Converged after 5 iterations.   
## Estimated dispersion parameter: 0.03814797   
##   
## Call:  
## glm(formula = cbind(survive, total - survive) ~ location + period,   
## family = binomial, data = troutegg, weights = disp.weights)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.04625 -0.21394 0.01708 0.28386 1.36990   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.5183 0.6206 7.281 3.32e-13 \*\*\*  
## location2 -0.3769 0.5603 -0.673 0.5011   
## location3 -1.2099 0.5066 -2.388 0.0169 \*   
## location4 -0.9562 0.5199 -1.839 0.0659 .   
## location5 -4.4679 0.5586 -7.999 1.25e-15 \*\*\*  
## period7 -2.0858 0.5201 -4.011 6.06e-05 \*\*\*  
## period8 -2.2273 0.5225 -4.263 2.02e-05 \*\*\*  
## period11 -2.3623 0.5186 -4.555 5.23e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 190.19 on 19 degrees of freedom  
## Residual deviance: 12.40 on 12 degrees of freedom  
## AIC: 43.393  
##   
## Number of Fisher Scoring iterations: 5

sumary(dmod)

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 4.51827 0.62059 7.2806 3.322e-13  
## location2 -0.37693 0.56029 -0.6727 0.50111  
## location3 -1.20993 0.50657 -2.3885 0.01692  
## location4 -0.95622 0.51993 -1.8392 0.06589  
## location5 -4.46793 0.55856 -7.9990 1.254e-15  
## period7 -2.08576 0.52006 -4.0106 6.055e-05  
## period8 -2.22727 0.52251 -4.2626 2.020e-05  
## period11 -2.36234 0.51857 -4.5555 5.226e-06  
##   
## n = 20 p = 8  
## Deviance = 12.40002 Null Deviance = 190.18576 (Difference = 177.78574)

Disp를 사용하지 않았을 때와 큰 차이가 없는 것을 알 수 있는데, 이는 covariate class의 size가 거의 동일하기 때문이다.

#5. Quasi-Binomial

Quasi-binomial model은 extra-binomial variation을 가능하게 해주는 방법 중 하나이다.

기본적인 아이디어는 response의 mean과 variance가 어떻게 linear predictor와 연결되어있는 지를 구체화하는 것이다.

일반적인 binomial model에서는 binomial distribution에서 추가적인 정보 말고 오직 mean과 variance information만을 활용한다. 따라서 beta와standard error를 추정할 때 full binomial assumption은 필요하지 않다.

다만 문제는 우리가 추론을 할 때 일어난다. 신뢰구간을 찾거나 또는 가설 검정을 할 때, 우리는 몇 개의 분포적 가정이 필요하다. 이전에 우리는 deviance를 사용했지만 이를 위해서 우리는 likelihood가 필요하며 또 likelihood를 계산하기 위해서 distribution이 필요하다. 이제 우리는 분포를 가정하지 않고도 계산될 수 있는 likelihood의 적절한 대체재가 필요하다.

의 평균이 이고 분산이 라고 하자. 각 는 독립적이라고 가정한다. 이 때 우리는 score 를 다음과 같이 정의하자.

그러면

이러한 특징들은 log-likelihood의 미분()에 의해 공유된다. 이는 우리가 U를 대신에 사용하는 것을 제시한다. 따라서 우리는

라고 정의할 수 있다.

목적은 Q가 log-likelihood처럼 행동해야 한다는 것이다. 우리는 log quasi-likelihood for all n observations를 다음과 같이 정의할 수 있다.

maximum likelihood에서 기대되는 일반적인 asymptotic 성질은 여기에도 똑같이 적용될 수 있다.

Quasi-likelihood는 오직 variance function에 직접적으로 의존하고 분포의 선택이 오직 variance function을 결정한다는 것에 유의하자. 따라서 variance function의 선택은 model에서 random structure와 연관되고 link function은 model의 systematic part와의 관계를 결정한다.

Standard linear model에서 quasi-likelihood는 정확히 log-likelihood와 대응된다. 여기서 dispersion parameter 는 이다. 따라서 이 접근으로는 아무것도 얻을 것이 없다.

그러나 binomial model에서는 도입은 model에 추가적인 dimension of flexibility를 제공한다. 이는 overdispersion을 modeling하는 데에 도움이 된다. 한 가지 흥미로운 가능성은 어떤 의 선택들은 어떠한 알려져 있는 분포와도 일치하지 않을 수 있다는 점이다.

베타의 추정치는 Q를 최대화함으로써 얻어진다. 모든 절차는 overdispersion parameter를 넣지 않았을 때와 동일하게 진행되지만, 추정은 다르다. 왜냐하면 likelihood approach는 여기서는 믿음직하지 않기 때문이다. 대신

를 추천한다.

비록 quasi-likelihood estimator가 더 적은 가정을 요하기 때문에 매력적일 수 있지만 regular likelihood based estimator에 대응하는 것에 비해서는 일반적으로 덜 efficient하다(분산이 큼). 따라서 만약 분포에 대한 정보를 가지고 있다면 그것을 쓰는 것이 낫다.

Regular deviance for a model은 현재 model의 log-likelihood와 saturated model의 log-likelihood의 차이로 형성된다.

그리고 유사성에 의해 quasi-deviance는 이다. 왜냐하면 saturated model로부터의 기여도가 0이기 때문이다. 가 지워지고, quasi-deviance는 다음과 같이 표현될 수 있다.

이제 data를 이용하여 위의 내용을 직접 확인해보자.

우선 우리가 이용하려는 데이터는 포유동물의 잠과 관련한 데이터이다. 우리는 포유동물이 잠에 들 때 꿈을 꾸는 시간의 비율을 종속 변수로 두고 체중, 뇌의 무게 등을 predictor로 하여 살펴보고자 한다.

data(mammalsleep, package='faraway')  
mammalsleep$pdr <- with(mammalsleep, dream/sleep)  
summary(mammalsleep$pdr)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.0000 0.1180 0.1755 0.1865 0.2427 0.4615 14

Summary를 통해 살펴보았을 때 꿈꾸는 시간의 비율은 0에서부터 거의 절반까지 나타난다는 것을 알 수 있다. 어떤 데이터셋에서는 비율의 범위가 0또는 1에 절대 가까워지지 않는 것이 있는데, 그러한 데이터셋은 normal gaussian model을 사용하는 것이 더욱 적절하다. 그런데 이 경우에는 그렇지 않고 매우 적은 비율들을 response value로 갖고 있으니, Gaussian model은 적절하지 않다. 따라서 우리는 비율 response를 그대로 modeling할 것이다. Logit link를 사용하는 것은 response가 0과1 사이의 값이기 때문에 합리적으로 보인다. 더 나아가 우리는 측정의 본질적 성질에 의해, 분산이 proportion 가 적당할 경우 더 크고, 만약 가 0또는 1에 가까워진다면 더 작아질 것이라고 예상할 수 있다. 이는 variance function의 근사적인 형태가 임을 제시한다. 이 함수는 0또는 1에 가까워지면 작아지고 1/2에 가까워질수록 커진다. 이런 것들을 봤을 때 canonical logit link를 가진 binomial GLM과 대응하지만 response가 binomial은 아니다. 따라서 우리는 quasi-binomial을 사용한다.

우선 모델에 직접적으로 사용하기에 앞서 skewed된 predictor들은 log function을 이용해서 해결해주었다.

mod1 <- glm(pdr ~ log(body) + log(brain) + log(lifespan) +   
 log(gestation) + predation + exposure + danger,  
 family=quasibinomial, mammalsleep)

이제 우리는 free dispersion parameter를 가졌기 때문에 모델 비교에 있어서 F test를 사용해야 한다.  
  
drop1(mod1, test='F')

## Single term deletions  
##   
## Model:  
## pdr ~ log(body) + log(brain) + log(lifespan) + log(gestation) +   
## predation + exposure + danger  
## Df Deviance F value Pr(>F)   
## <none> 1.5703   
## log(body) 1 1.7786 4.5107 0.04104 \*  
## log(brain) 1 1.5856 0.3320 0.56827   
## log(lifespan) 1 1.6532 1.7949 0.18922   
## log(gestation) 1 1.6232 1.1466 0.29181   
## predation 1 1.5751 0.1037 0.74946   
## exposure 1 1.5851 0.3202 0.57523   
## danger 1 1.5848 0.3146 0.57857   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Predation의 p-value 값이 제일 높으므로(least significant) 이를 제한다. 비슷한 방법으로 backward elimination을 진행하면 log(brain)과 log(gestation), exposure를 제할 수 있다.

제외된 predictor들을 빼고 모델을 다시 돌려보면,

mod1 <- glm(pdr ~ log(body) + log(lifespan) + danger, family=quasibinomial, mammalsleep)  
sumary(mod1)

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -0.493233 0.291261 -1.6934 0.0979582  
## log(body) 0.146307 0.038425 3.8076 0.0004611  
## log(lifespan) -0.286605 0.107975 -2.6544 0.0112584  
## danger -0.173190 0.059952 -2.8888 0.0061512  
##   
## Dispersion parameter = 0.04065  
## n = 45 p = 4  
## Deviance = 1.73211 Null Deviance = 2.50881 (Difference = 0.77671)

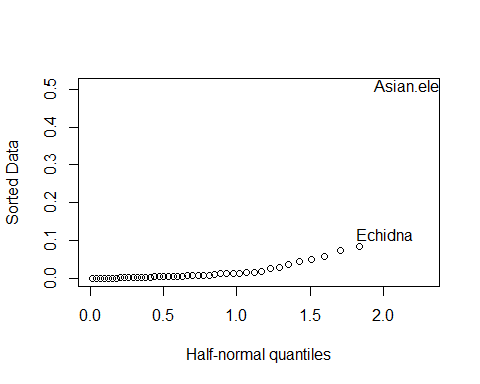
우리가 일반적으로 binomial에서 보던 기본 값보다 훨씬 더 작은 Dispersion Parameter 값을 확인할 수 있다.

결과를 해석하면, 몸집이 더 크고 수명이 더 짧으며 덜 위험한 환경 속에서 사는 포유동물일수록 꿈을 길게 꾼다.

비교적 큰 residual deviance값은 이 model이 잘 fit하지는 않는다는 것을 알려준다.

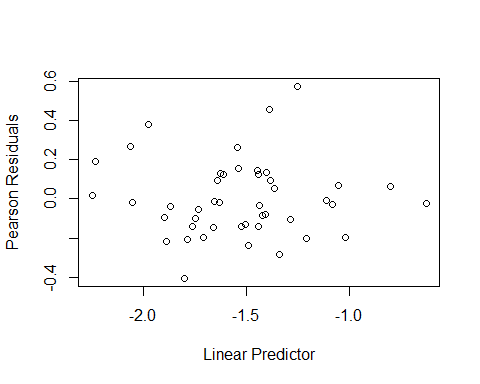
Diagnostic을 진행해보자.

l1 <- row.names(na.omit(mammalsleep[,c(1,6,10,11)]))  
halfnorm(cooks.distance(mod1), labs=l1)



* Asian elephant는 상당히 influential 하며, 이 case를 제외한 fit을 고려해보아야 한다.

plot(predict(mod1), residuals(mod1, type='pearson'), xlab='Linear Predictor', ylab='Pearson Residuals')



* Constant variation 형태를 보인다.
* 즉, 우리가 선택한 variance function이 합리적이라는 것을 나타낸다.
* Pearson Residual을 사용한 이유는 이것이 variance function을 이용해 raw residual을 더욱 명확히 normalize하여 더 명확히 check할 수 있도록 만들어주기 때문이다.

#6. Beta Regression

Response가 위의 사례처럼 0과 1사이의 값을 가질 때, 또는 scaling을 해서 0과 1사이에 둘 수 있을 때 유용한 것이 Beta Regression이다.

우선 Beta-distributed random variable Y의 density는 다음과 같다.

이 때 라고 하자.

그러면 이다.

그리고 linear predictor와 평균을 link 할 수 있는데, 이 때 사용되는 link function은 binomial regression에서 사용되는 것이라면 어느 것이라도 적절하다.

R에서는 mgcv package를 사용해서 Beta Regression을 진행할 수 있다.

data(mammalsleep, package='faraway')  
mammalsleep$pdr <- with(mammalsleep, dream/sleep)  
library(mgcv)

## Loading required package: nlme

## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.

modb <- gam(pdr ~ log(body) + log(lifespan), family=betar(), mammalsleep)

## Warning in family$saturated.ll(y, prior.weights, theta): saturated likelihood  
## may be inaccurate

summary(modb)

##   
## Family: Beta regression(8.927)   
## Link function: logit   
##   
## Formula:  
## pdr ~ log(body) + log(lifespan)  
##   
## Parametric coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.37795 0.37322 1.013 0.311   
## log(body) 0.26796 0.05513 4.860 1.17e-06 \*\*\*  
## log(lifespan) -0.92266 0.16585 -5.563 2.65e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## R-sq.(adj) = -0.178 Deviance explained = 73.5%  
## -REML = -47.801 Scale est. = 1 n = 45

결과를 확인해보면 값은 Family: Beta Regression 옆에 괄호 안에 있는 값으로 8.927이다. 그리고 앞의 장에서 quasi를 이용한 regression값과 비교해보면 크게 차이가 나지 않는다는 것을 알 수 있다. 다만 Beta-based model의 장점은 full-distribution model로, 단순히 point estimate나 standard error가 아니라 full predictive distribution을 만들 수 있다는 것이다.