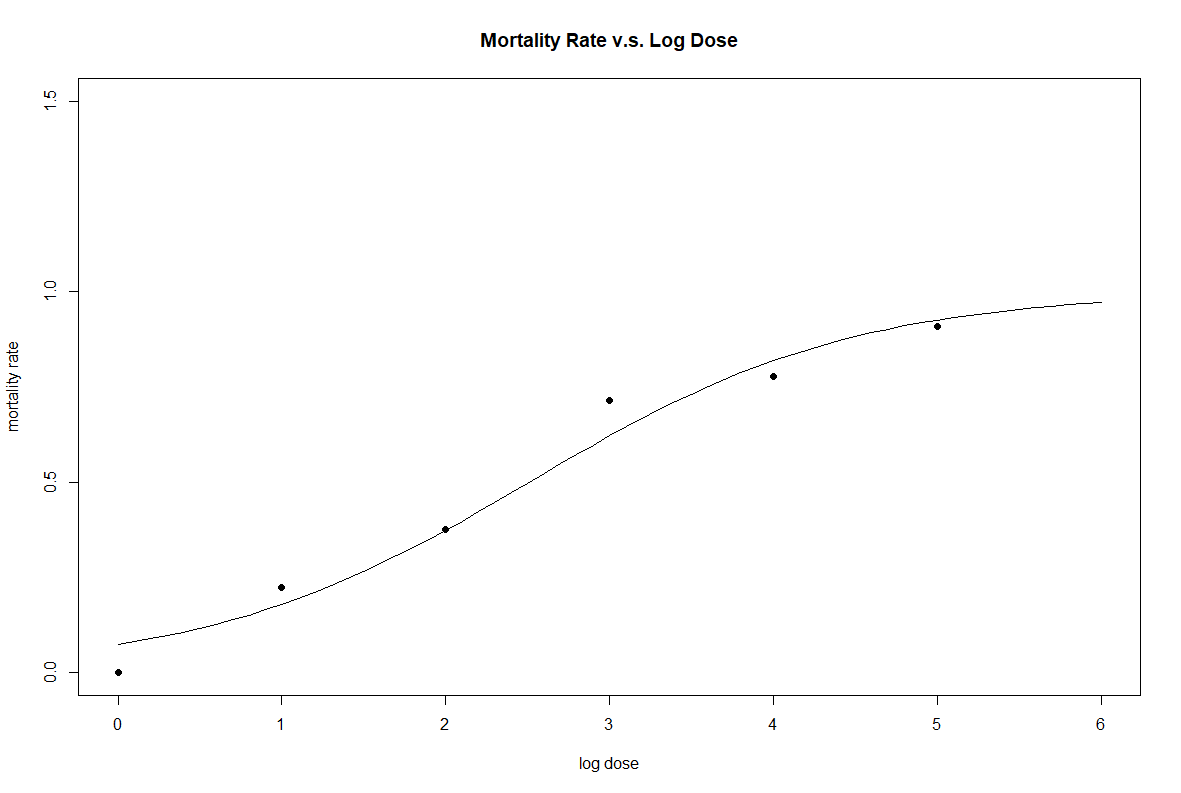
**STAT34700 HW1**

**Problem 3:**

**Model: Logistic Regression Model**

**(a)**

**The plot of mortality rate v.s. log dose:**



The points in the plot are original data points.

The curve is fitted mortality rate by fitting the linear logistic model.

**Output:**

> sumary(glmfit)

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.54114 0.77185 -3.2923 0.0009939

logdose 1.01208 0.26261 3.8540 0.0001162

n = 6 p = 2

Deviance = 1.57815 Null Deviance = 25.79224 (Difference = 24.21408)

**Fitted Model:**

**(b)**

If the predictor is then the mortality rate p is 0.5.

Using delta method to calculate the 95% CI of :

The 95% CI of is [1.798170, 3.223461].

**Output:**

> LD50

(Intercept)

2.510815

> se

[1] 0.3635947

> c(LD50-1.96\*se,LD50+1.96\*se)

(Intercept) (Intercept)

1.798170 3.223461

**(c)**

The 1-α CI for ρ is

It is equivalent to

To calculate the 1-α CI for ρ, we need to solve the equation:

Assume α=0.05, then

It is equivalent to

Hence, .

The 95% CI for is [-3.313109, -1.656959].

**Output:**

> cm

(Intercept) logdose

(Intercept) 0.5957559 -0.17824759

logdose -0.1782476 0.06896221

> CI

(Intercept) (Intercept)

-3.313109 -1.656959

**(d)**

**Parametric Bootstrap:**

The 95% CI forρ is [-3.211831, -1.747074]

**Output:**

> apply(phodata,2,function(x) quantile(x,c(0.025,0.975)))

pho

2.5% -3.211831

97.5% -1.747074

**Non-parametric Bootstrap:**

The 95% CI forρ is [-3.320073, -1.755771]

**Output:**

> apply(phonondata,2,function(x) quantile(x,c(0.025,0.975),na.rm=TRUE))

phonon

2.5% -3.320073

97.5% -1.755771

**(e)**

If the predictor is then the mortality rate p is 0.5.

Hence,

In the restricted model, the.

We can use as the predictor and no intercept to fit the logistic regression model.

After fitting the restricted model, we can calculate its log likelihood by R.

In addition, we calculate the log likelihood of unrestricted model we get in part(a).

Under the null hypothesis, LR(4) approximately follows the Chi-square distribution with df = 1

so the p-value is 0.0006622652.

**Output:**

> pchisq(11.59227,df=1,lower.tail = FALSE)

[1] 0.0006622652

**R Code**

**# Problem 2 ---------------------------------------------------------------**

install.packages("faraway")

library(faraway)

**#a**

logdose = c(0,1,2,3,4,5)

mortality = c(0,2/9,3/8,5/7,7/9,10/11)

mort = c(0,2,3,5,7,10)

unmort = c(7,7,5,2,2,1)

dose = data.frame(mort,unmort,mortality,logdose)

dose$total = dose$mort + dose$unmort

#plot the mortality rate vs log dose

plot(c(0,6),c(0,1.5),type="n",xlab="log dose",ylab="mortality rate",main="Mortality Rate v.s. Log Dose")

points(dose$logdose,dose$mortality,pch=16)

#fit the logistic model

glmfit = glm(cbind(mort,unmort) ~ logdose, family = binomial(), data=dose)

sumary(glmfit)

#superimpose the fitted values

x = seq(0,6,0.1)

pre=predict(glmfit,data.frame(logdose=x),type="response",se.fit=TRUE)

lines(x,pre$fit)

**#b**

#estimate LD50

LD50=-glmfit$coef[1]/glmfit$coef[2]

glmfitsum=summary(glmfit)

#use delta method to calculate the CI for LD50

dr <- c(-1/glmfit$coef[2],glmfit$coef[1]/glmfit$coef[2]^2)

se = sqrt(dr %\*% glmfitsum$cov.un %\*% dr)[,]

c(LD50-1.96\*se,LD50+1.96\*se)

**#c**

#get the covariance matrix of coefficients

cm = vcov(glmfit)

#solve the function

a = glmfit$coef[2]^2-1.96^2\*cm[2,2]

b = 2\*(1.96^2\*cm[1,2]-glmfit$coef[1]\*glmfit$coef[2])

c = glmfit$coef[1]^2-1.96^2\*cm[1,1]

ro2 = (-b + sqrt(b^2-4\*a\*c))/(2\*a)

ro1 = (-b - sqrt(b^2-4\*a\*c))/(2\*a)

#get the 95% CI for pho

CI = c(ro1, ro2)

ro = glmfit$coef[1]/glmfit$coef[2]

**#d**

**#nonparametric bootstrap**

nrows=dim(dose)[1]

aa=sapply(1:nrows,pr<-function(x) rep(dose$logdose[x],dose$total[x]),simplify="array")

aaa=unlist(aa)

# Insert affected 1's and unaffected 0's for each concentration level

bb=sapply(1:nrows,pr<-function(x) c(rep(1,dose$mort[x]),rep(0,dose$unmort[x])),simplify="array")

bbb=unlist(bb)

# Combine concentration and binary vector of affected/unaffected.

casedose=cbind(aaa,bbb)

# Create a list that can be sampled.

data=split(casedose,row(casedose))

# Now 500 times resample from the cases and estimate a model

set.seed(123)

sampsize=500

cfsnon=matrix(rep(0,sampsize\*2),c(sampsize,2))

for (j in seq(sampsize))

{

n=length(data)

ss=sample(data,n,replace=TRUE)

# Make the sample into a nx2 matrix

dd=t(matrix(unlist(ss),nrow=2,ncol=n))

# Make the sampled list into a data frame with counts for each concentration

# level found in the sample

ee=table(dd[,1],dd[,2])

# Reestimate the model from the resampled data.

glmsam2=glm(ee[,1:2]~dose$logdose,family=binomial())

cfsnon[j,]=glmsam2$coefficients

}

phonon=cfsnon[,1]/cfsnon[,2]

phonondata=data.frame(phonon)

#Calculate 95% CI for pho

apply(phonondata,2,function(x) quantile(x,c(0.025,0.975),na.rm=TRUE))

**#parametric bootstrap**

preds=matrix(glmfit$fitted.values)

nums=dose$total

set.seed(123)

# Number of simulations.

sampsiz=500

# Matrix of coeficients of each sample. sampsiz x 2

cfs=matrix(rep(0,sampsiz\*2),c(sampsiz,2))

for (j in seq(sampsiz))

{

# Simulate the data for each log-concentration level

aff=mapply(rbinom,rep(1,length(nums)),nums,preds)

unaff=nums-aff

# Fit model to simulated data

glmsam1=glm(cbind(aff,unaff)~dose$logdose, family=binomial())

# Store coefficients.

cfs[j,]=glmsam1$coefficients

}

pho=cfs[,1]/cfs[,2]

phodata=data.frame(pho)

#Calculate 95% CI for pho

apply(phodata,2,function(x) quantile(x,c(0.025,0.975)))

**#e**

#transform the predictor

dose$logdosenull = dose$logdose - 4

#fit the model

nullfit=glm(cbind(mort,unmort) ~ logdosenull - 1, family = binomial,data = dose)

#get LR(4)

LR4=2\*(logLik(glmfit)-logLik(nullfit))

#calculate the p-value

pchisq(11.59227,df=1,lower.tail = FALSE)