流行病學與生物統計計算Homework 8

學號：b07401048 系級：醫學五 姓名：賴柏瑞

**# Homework8**

**# EX 20-1**

**Q1 : use Newton-Raphson method to find MLE of the regression coefficient og logistic regression**

*# EX 20-1*

**setwd**("/Users/raymond/Desktop/")

resp <- **read.csv**("/Users/raymond/Desktop/resp.csv", header = TRUE)

**head**(resp)

**dim**(resp)

*# Q1 : use NR to find maximum likelihood estimate of beta*

*# convert treatment to numeric, setting "P" be 1, "A" be 0*

respma <- **as.data.frame**(resp)

for (i in 1 : **length**(resp$treatment)) {

if (respma[i, 3] == "P") {

respma[i, 3] <- 1

} else {

respma[i, 3] <- 0

}

}

**head**(respma)

(treatment <- **as.numeric**(respma[, 3]))

*# constructing matrix*

X <- **cbind**(**rep**(1, **length**(resp$outcome)), treatment, resp$age, resp$baseline) *# nolint*

Y <- resp$outcome *# nolint*

*# ftn*

**ftn** <- function(betacoef) {

pi1 <- **exp**(X %\*% betacoef) / (1 + **exp**(X %\*% betacoef))

gradient <- **t**(X) %\*% (Y - pi1)

hessian <- - **t**(X) %\*% **diag**(**c**(pi1 \* (1 - pi1)), **length**(resp$outcome)) %\*% X *# nolint*

return(list(gradient, hessian))

}

*# highorder Newton-Raphson method*

**highnew** <- function(ftn, x0, tol, maxiter) {

x <- x0

fx <- ftn(x)

iter <- 0

while ((**max**(**abs**(fx[[1]])) > tol) && (iter < maxiter)) {

x <- x - (**solve**(fx[[2]]) %\*% fx[[1]])

fx <- ftn(x)

iter <- iter + 1

}

if (**max**(**abs**(fx[[1]])) > tol) {

**cat**("algorithm failed to converge\n")

return(NULL)

} else {

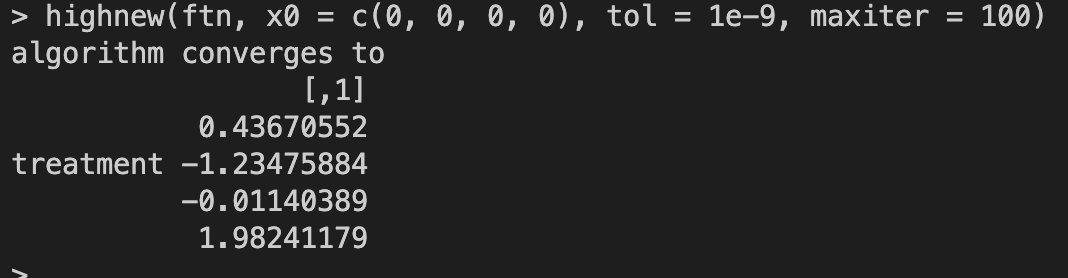
**cat**("algorithm converges to\n")

return(x)

}

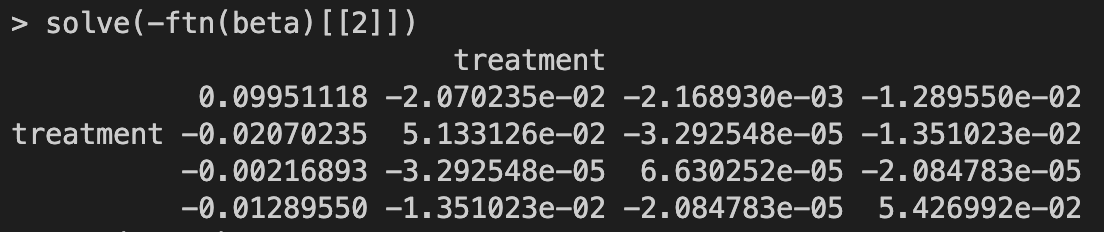
}

highnew(ftn, x0 = **c**(0, 0, 0,0), tol = 1e-9, maxiter = 100)



**# Q2：find the variance-covariance matrix**

(beta <- highnew(ftn, **c**(0, 0, 0, 0), tol = 1e-9, maxiter = 100))

 **solve**(-ftn(beta)[[2]])

**# Q3：find the log likelihood**

X <- **cbind**(**rep**(1, **length**(resp$outcome)), treatment, resp$age, resp$baseline)

Y <- resp$outcome

**ftn** <- function(betacoef) {

pi1 <- **exp**(X %\*% betacoef) / (1 + **exp**(X %\*% betacoef))

gradient <- **t**(X) %\*% (Y - pi1)

hessian <- - **t**(X) %\*% **diag**(**c**(pi1 \* (1 - pi1)), **length**(resp$outcome)) %\*% X *# nolint*

loglike <- **sum**(Y \* **log**(pi1 / (1 - pi1)) + **log**(1 - pi1))

return(list(gradient, hessian, loglike))

}

ftn(beta)[[3]]

