Hwk#12

library(binom)

betagpa <- c(0,0.5,0.8)

sig <- c(0.5,0.8) # significance level

n <- 1000

Y <- c()

tol= 1e-9

no.rep <- 100

n.max = 100

rej.rate <- matrix(NA,length(betagpa),length(sig)) # rejection rate

for(n.betaloop in 1:length(n)){ # the 1st loop

pvalue <- c()

pvalueglm <- c()

for(i in 1:no.rep){ # the 2nd loop

set.seed(i)

gpa <- rnorm(n,3.1,0.3)

gre <- rnorm(n,580,80)

pii <- exp(-6+betagpa[n.betaloop]\*gpa+0.005\*gre)/(1+exp(-6+betagpa[n.betaloop]\*gpa+0.005\*gre))

for(j in 1:n[n.betaloop])########

Y[j]<- sample(c(0,1),1,c(1-pii[j],pii[j]),replace = F)

# above: data generation process

# below: data analysis process

X <- cbind(rep(1,length(Y)),gpa,gre)

betaco <- c(0,0,0)

pii <- exp(X%\*%betaco)/ (1+exp(X%\*%betaco))

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

hessian<- -t(X)%\*%diag(c(pii\*(1-pii)), length(Y))%\*%X

n.iter <-0

while ((max(abs(gradient)) > tol) & (n.iter < n.max)) {

betaco <- betaco - solve(hessian) %\*% gradient

pii <- exp(X%\*%betaco)/ (1+exp(X%\*%betaco))

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

hessian<- -t(X)%\*%diag(c(pii\*(1-pii)), length(Y))%\*%X

n.iter <- n.iter + 1

}

if (n.iter == n.max) {

cat('Algorithm failed to converge\n')

break

} else { # max(abs(fx[[1]])) <= tol

MLE <- betaco

vcovMLE <- diag(solve(-hessian))

} #((1-pt(abs(MLE/seMLE),n-3))\*2)[3]

pvalue[i] <-(1-pt(abs(MLE/vcovMLE),n-3)\*2)[2] # Wald’s test statistic = -1.6522370, gpa 0.5852453, gre 1268.0905623

pvalueglm <- summary(glm(Y~gpa+gre))$coef[2,4] #pvalue = 0.739059

}

for(k in 1:length(sig)){

rej.rate[n.betaloop,k] <- sum(pvalue<sig[k])/no.rep# calculate rejection rate

}

}

rej.rate

matplot(sig,t(rej.rate),col=c(1:length(betagpa)),pch=c(1:length(betagpa)),lty=c(1:length(betagpa)),type="b",frame=F,xlab="Significance level",ylab="Rejection rate")

abline(a=0,b=1,col=8)

legend(0.04,rej.rate[1,2]+0.05,expression(paste(beta,'=0')),bty="n")

legend(0.04,rej.rate[2,2]+0.05,expression(paste(beta,'=0.1')),bty="n")

legend(0.04,rej.rate[3,2]+0.05,expression(paste(beta,'=0.2')),bty="n")