**Homework** (8 points, please pay attention to all the words in this orange box)

Ex 21: Please generate random numbers from a Binomial distribution with *n*=20 (number of trials) and *p*=0.15 (probability of success), and compare the coverage and length of 95% asymptotic confidence intervals and 95% exact confidence intervals. Number of replications = 1000, and seed numbers from 1 to 1000, respectively.

Hint: rbinom(1,size=20,prob=0.15)

**1** class, **20** students enrolled in this class, the probability of passing the final exam is **0.15**.

Requirements:

1. Please use the Newton-Raphson method to calculate exact C.I., but you don’t need to present your plots when deciding initial values. (You may plot by yourself. To make this homework simpler, we will not score on them.)

2. Please calculate 95% asymptotic confidence interval based on  (asymptotic: based on the central limit theorem). If the lower bound of a C.I. is less than 0, please regard it as 0, because the probability of success will always range from 0 to 1.

**Please note: Using the “binom” package to answer this homework will be scored as 0, although you may use that package to check your own answers.**

95% exact confidence intervals

Coverage : 97%



Length: 0.3332868

> mean(re95-le95,na.rm=T)

[1] 0.3332868

95% asymptotic confidence intervals

Coverage: 80.3%



Length: 0.2765599

> mean(r95-l95,na.rm=T)

[1] 0.2765599

Code:

p <- 0.15

n <- 20

no.rep <- 1000

l95 <- rep(NA,no.rep)

r95 <- rep(NA,no.rep)

for(i in 1:no.rep){

#print(i)

set.seed(i)

x <- rbinom(1,size=n,prob=p)

phat <- x/n

l95[i] <- max(0,phat-qnorm(0.975)\*sqrt(phat\*(1-phat)/n))

r95[i] <- phat+qnorm(0.975)\*sqrt(phat\*(1-phat)/n)

}

mean((l95<=p) & (p<=r95))

newtonraphson <- function(ftn, x0, tol = 1e-9, max.iter = 100){

x <- x0 #先把xassign為起始值x0

fx <- ftn(x)

iter <- 0

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

x <- x - fx[1]/fx[2]

fx <- ftn(x)

iter <- iter + 1

#cat("At iteration", iter, "value of x isะ", x, "\n")

}

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

cat("Algorithm failed to converge\n")

return(NULL)

} else{

#cat("Algorithm converged\n")

return(x)

}

}

findECI <- function(datapoint,coverage){

ftn7 <- function(p){

fp <- (1-coverage)/2-1

dfp <- 0

for(k in 0:(datapoint-1)){ ### the datapoint should be >= 1

fp <- fp+choose(20,k)\*(p^k)\*((1-p)^(20-k))

dfp <- dfp+choose(20,k)\*((k\*p^(k-1))\*((1-p)^(20-k))-(p^k)\*((20-k)\*(1-p)^(19-k)))

}

return(c(fp, dfp))

}

if(datapoint<5){

pl <- newtonraphson(ftn7, 0.03, 1e-09)

}

if(datapoint>=5 & datapoint<8){

pl <- newtonraphson(ftn7, 0.10, 1e-09)

}

if(datapoint>=8){

pl <- newtonraphson(ftn7, 0.20, 1e-09)

}

ftn8 <- function(p){

fp <- -(1-coverage)/2

dfp <- 0

for(k in 0:datapoint){

fp <- fp+choose(20,k)\*(p^k)\*((1-p)^(20-k))

dfp <- dfp+choose(20,k)\*((k\*p^(k-1))\*((1-p)^(20-k))-(p^k)\*((20-k)\*(1-p)^(19-k)))

}

return(c(fp, dfp))

}

if(datapoint<5){

pu <- newtonraphson(ftn8, 0.3, 1e-09)

}

if(datapoint>=5 & datapoint<8){

pu <- newtonraphson(ftn8, 0.5, 1e-09)

}

if(datapoint>=8){

pu <- newtonraphson(ftn8, 0.6, 1e-09)

}

return(c(pl,pu))

}

le95 <- rep(NA,no.rep)

re95 <- rep(NA,no.rep)

for(i in 1:no.rep){

#print(i)

set.seed(i)

x <- rbinom(1,size=n,prob=p)

le95[i]<- findECI(x,0.95)[1]

re95[i]<- findECI(x,0.95)[2]

if(x==0){

le95[i] <- 0

}

}

mean((le95<=p)&(p<=re95))

mean(r95-l95,na.rm=T)

mean(re95-le95,na.rm=T)

x <- c()

for(i in 1:no.rep){

#print(i)

set.seed(i)

x[i]<- rbinom(1,size=n,prob=p)

}

aa <- cbind(x,l95,r95,le95, re95)

xx<-9

aa[aa[,1]==xx,]