HW\_3\_Template

Lina Cao , Thursdays 1:15PM

SDGB 7844; Prof. Nagaraja; Fall 2017

## Question 1

n1 <- 100 # number of individuals that are captured and tagged  
n2 <- 100 # number of individuals recaptured  
N <- 5000 # total pupulation  
m <- sample(1:N, size=n2, replace=FALSE)  
set.seed(2)  
# 1:N: we selected 1 to N numbered individual  
# size : n2=100, it is the number of individual that we randomly sampled  
# replace = FALSE, the capture is without replacement  
  
# assume that tagged individuals are numbered from 1 to 100  
# then the individual that is less than 100 should be calculated as "recaptured tagged group"  
# calculate the number of individuals to be m2  
m2 <- length(m[m<=n1])  
paste("m2=",m2,sep = "")

## [1] "m2=2"

# calculate estimate of total individual  
est.n <- n1\*n2/m2  
paste("estimated total population=",round(est.n),sep = "")

## [1] "estimated total population=5000"

From the sampling, the nuber of tagged individuals ,m2, is 3. Thereby, the result of estimated total population is around 3333.

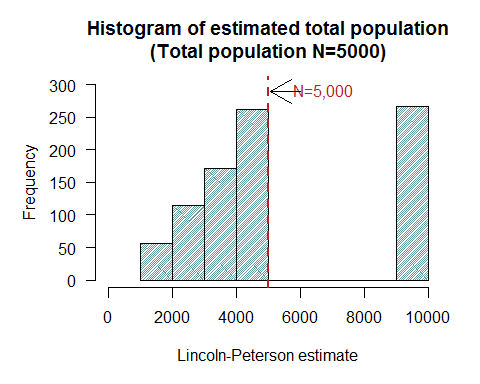
## Question 2

(Only code for this question.)

crec <- function(N=5000,n1=100,n2=100,sim.run=10){   
 # as calculated in Que.1, set default value for each parameter, which can be set specifically later  
 # set null vector to form vectors later  
 m2.vector <- rep(NA, times = sim.run)  
 est.vector <- rep(NA, times = sim.run)  
   
 for (i in 1:sim.run){  
 # get sample of "recaptured group" in each iteration   
 x <- sample(1:N, size=n2, replace=FALSE)  
 # calculate the number of tagged ones in x and save it in the vector  
 m2.vector[i] <- length(x[x<=n1])  
 # calculate the estimated number of population and save it in the vector  
 est.vector[i] <- n1\*n2/m2.vector[i]  
   
 }# end of loop  
   
 # return result of the function  
 return(list("recapture results"=data.frame(m2.vector,est.vector), "total population"=N))  
   
}# end of function  
  
# make histogram plot using crec function  
set.seed(3)  
# simulate 1000 iterations  
temp <- crec(N=5000,n1=100,n2=100,sim.run=1000)  
# check the first 6 rows of the estimated result  
head(temp$"recapture results")

## m2.vector est.vector  
## 1 3 3333.333  
## 2 0 Inf  
## 3 2 5000.000  
## 4 0 Inf  
## 5 0 Inf  
## 6 4 2500.000

# plot  
# add a wider margin to addjust plot  
par(mar=c(5, 5, 4, 2))  
hist(temp$"recapture results"$est.vector,  
 main ="Histogram of estimated total population\n(Total population N=5000)",   
 las = TRUE, col="cadetblue", density=60, angle=45, border="black",  
 xlab = "Lincoln-Peterson estimate",  
 xlim = c(0,10000),ylim = c(0,300))  
# add line of N=5000   
abline(v=5000, col="firebrick", lty=2, lwd=2)  
# N=5000 is as high as the title of plot using axis  
# axis(side=3, at=5000, label="N=5000", col="lightblue", las=TRUE)  
# use text to indicate N instead  
# use locator() to get approximate location of the label  
text(6700,290, cex = 1, labels = "N=5,000", col = "firebrick")  
arrows(x0=6000, y0=290, x1=5050, lty = 1)



## Question 3

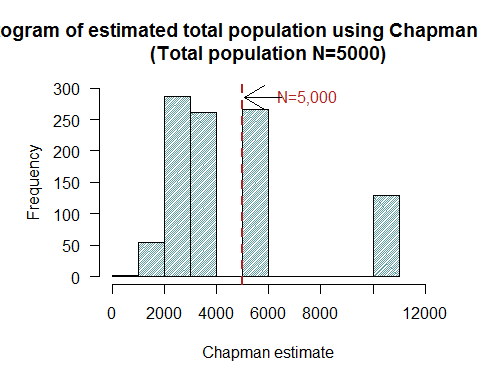
# derive vector that includes all infinite value in estimated total population  
nInf <- temp$`recapture results`$est.vector[temp$`recapture results`$est.vector==Inf]  
# calculate the proporation  
percent <- length(nInf)/length(temp$`recapture results`$est.vector)  
paste("The percent of infinite value in estimated population values is ", percent, sep="")

## [1] "The percent of infinite value in estimated population values is 0.129"

The percent of infinite value in estimated population values is 0.129. The reason of its occurrence is that the ramdomly recapture will receive no tagged individuals, i.e. m2=0. Therefore the result of equation 1 has denominator of 0, which leads to an infinite result.

## Question 4

# generate the saved m2 values first  
m2.sim <- temp$`recapture results`$m2.vector  
#calculate Chapman estimator  
est.chapman <- (n1+1)\*(n2+1)/(m2.sim+1)-1  
# plot  
# addjust plot  
par(mar=c(5, 5, 4, 2)+0.2)  
hist(est.chapman,  
 main ="Histogram of estimated total population using Chapman estimator\n(Total population N=5000)",   
 las = TRUE, col="cadetblue", density=60, angle=45, border="black",  
 xlab = "Chapman estimate",  
 xlim = c(0,12000), ylim = c(0,300))  
# add line of N=5000   
abline(v=5000, col="firebrick", lty=2, lwd=2)  
# N=5000 is as high as the title of plot using axis  
# axis(side=1, at=5000, label="N=5000", las=TRUE)  
## use text to indicate N instead  
## use locator() to get approximate location of the label  
text(7500,285, cex = 1, labels = "N=5,000", col = "firebrick")  
arrows(x0=6500, y0=285, x1=5050, lty = 1)



## Question 5

##########################################  
## use estimates from former simulation ##  
##########################################  
N = 5000  
bias.LP <- mean(temp$"recapture results"$est.vector)-N  
bias.C <- mean(est.chapman)-N  
bias.LP

## [1] Inf

bias.C

## [1] -673.8366

########################################################################  
## use function to try more times and get a more closer value of bias ##  
########################################################################  
  
# write a function that can rotate several times within one command  
bias<- function(n1=100, n2=100, N=5000, run=10, sim=100){  
   
 # n1: number of individuals that are captured and tagged  
 # n2: number of individuals recaptured  
 # N: total pupulation   
 # sim: iritation times  
   
# set null vector for each variable   
 m2.vector <- rep(NA, times = run)  
 est.LP <- rep(NA, times = run)  
 est.C <- rep(NA, times = run)  
 ave.LP <- rep(NA, times = sim)  
 ave.C <- rep(NA, times = sim)  
   
   
 for(i in 1:sim){ # how many times of the iritation  
   
 for(j in 1:run){ # how many estimates will be generated in one iritation  
   
 # get sample of "recaptured group" in each iteration   
 x <- sample(1:N, size=n2, replace=FALSE)  
 # calculate the number of tagged ones in x and save it in the vector  
 m2.vector[j] <- length(x[x<=n1])  
 # calculate the LP estimates and save it in the vector  
 est.LP[j] <- n1\*n2/m2.vector[j]   
 # calculate the chapman estimates and save it in the vector  
 est.C[j] <- (n1+1)\*(n2+1)/(m2.vector[j]+1)-1   
   
 }# end of loop1--an entire iteration to derive an estimate  
   
 # calculate average value of each estimator  
 ave.LP[i] <- mean(est.LP)  
 ave.C[i] <- mean(est.C)  
   
 } # end for loop2--calculate the mean of result from total times of simulation  
  
 return(list("average of LP estimator"=mean(ave.LP),   
 "average of C estimator"=mean(ave.C),  
 "bias of LP estimator"=mean(ave.LP)-N,  
 "bias of C estimator"=mean(ave.C)-N))  
   
} # end function  
  
# trial   
set.seed(4)  
list("A"=bias(),  
 "B"=bias(run=1000,sim=500),  
 "C"=bias(run=5000,sim=500),  
 "D"=bias(run=5000,sim=1000))

## $A  
## $A$`average of LP estimator`  
## [1] Inf  
##   
## $A$`average of C estimator`  
## [1] 4347.67  
##   
## $A$`bias of LP estimator`  
## [1] Inf  
##   
## $A$`bias of C estimator`  
## [1] -652.3299  
##   
##   
## $B  
## $B$`average of LP estimator`  
## [1] Inf  
##   
## $B$`average of C estimator`  
## [1] 4376.205  
##   
## $B$`bias of LP estimator`  
## [1] Inf  
##   
## $B$`bias of C estimator`  
## [1] -623.7949  
##   
##   
## $C  
## $C$`average of LP estimator`  
## [1] Inf  
##   
## $C$`average of C estimator`  
## [1] 4376.256  
##   
## $C$`bias of LP estimator`  
## [1] Inf  
##   
## $C$`bias of C estimator`  
## [1] -623.7445  
##   
##   
## $D  
## $D$`average of LP estimator`  
## [1] Inf  
##   
## $D$`average of C estimator`  
## [1] 4377.052  
##   
## $D$`bias of LP estimator`  
## [1] Inf  
##   
## $D$`bias of C estimator`  
## [1] -622.9483

I think neither of the estimator is unbiased when n1,n2=100.

1. Use the estimated values of total population in my former simulation.

From the result, the bias of Lincoln-Pearson estimator is Inf, while that of Chapman estimator is -673.84. Neither of them is equal to 0, so I think both estimators are biased.

1. Use function to rotate several times to check (Is it correct to do so?)

This is the case when I simulate the sample average for hunderd times, will the bias converge to a certain small range. Due to the infinite situation of estimate in Lincoln-Pearson estimator, the bias is steadily "infinite". According to four cases simulation, bias of Chapman estimator converges to around -623, or which is a little far from zero, therefore, I don't think it can be regard as unbiased.

## Question 6

I think Chapman estimator is better. Because the denominator is the number of tagged individual plus 1, the answer of the equation will never be infinite.

According to the two histogram above, the "real" number of total population 5000 is among 1000 times of Chapman estimates, in another word, Chapman estimates are around "real" population and has a similar magnitude of the true value, rather than far away from it. Also, compared to LC estimator, the data below and above true value N is at a similar amount level.

On the contray, the data below true value N in Lincoln-Pearson estimator take up much more than that above N. Therefore it is easier for Lincoln-Pearson estimates to underestimate the ture value.

In addition, the bias of Chapman estimator is less than that of LC estimator, which is infinite.

Hence I think Chapman estimates are closer to true value, which makes it a better estimator.

## Question 7

1. each individual is independently captured.

Some individuals can be captured dependent on other individuals. In real life, many relationships can cause denpendency. In the "total fish population in a pond" case, fish are gregarious and can be clustered, while in sociology census sampling case, family or neighbours will be more likely to be chosen than people totally with no relationship. Therefore in real life independently capture will be difficult.

1. each individual is equally likely to be captured.

Continue considering the reason in (a), a cluster of individual will affect the capture result. Imagine that when we meet a group of people in the sociology survey, if one of them accept to help with the questionnaire, the rest will be more likely to do the same survey. Thus the probablity that each individual to be captured is not equal.

1. there are no births, deaths, immigration, or emigration of individuals

In real life capture-recapture experiment, the time period of the whole process will always lies in a period of time, like a month, or even years. It is not realistic to regard no births, deaths and other population migration in real life.