HW3-Darshan Patel-3:30-5:30PM

Darshan Patel

10/18/2018

In the beginning of the 17th century, John Graunt wanted to determine the effect of the plague ont he population of England; two hundred years later, Pierre-Simon Laplace wanted to estimate the population of France. Both Graunt and Laplace implemented what is now called the capture-recapture methood. This technique is used to not only count human populations (such as the homeless) but also animals in the wild.

In its simplest form, individuals are “captured,” “tagged,”" and released. A while later, individuals are “captured” and the number of “tagged” individuals, is counted. If is the true total population size, we can estimate it with as follows:

using the relation

This is called the Lincoln-Peterson estimator.

We make several strong assumptions when we use this method: (a) each individual is independently captured, (b) each individual is equally likely to be captured, (c) there are no births, deaths, immigration, or emigration of individuals (i.e., a closed population), and (d) the tags do not wear off (if it is a physical mark) and no tag goes unnoticed by a researcher.

*Goal:* In this assignment, you will develop a Monte-Carlo simulation of the capture-recapture method and investigate the statistical properties of the Lincoln-Peterson and Chapman estimators of population size . (Since you are simulating your own data, you know the true value of the population size allowing you to study how well these estimators work.)

# Import libraries  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 2.2.1 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.6  
## ✔ tidyr 0.8.1 ✔ stringr 1.3.0  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## Warning: package 'tidyr' was built under R version 3.4.4

## Warning: package 'purrr' was built under R version 3.4.4

## Warning: package 'dplyr' was built under R version 3.4.4

## ── Conflicts ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

## Question 1:

Simulate the capture-recapture method for a population of size when and using the sample() function (assume that each individual is equally likely to be “captured”). Determine and calculate .

Answer:

# Create variables  
n <- 5000  
n\_1 <- 100  
n\_2 <- 100  
  
# Sample the captured points  
cap\_one <- sample(1:n, n\_1, replace = FALSE)  
cap\_two <- sample(1:n, n\_2, replace = FALSE)  
  
# Calculate m\_2 and N\_LP  
m\_2 <- length(intersect(cap\_one, cap\_two))  
N\_LP <- n\_1 \* n\_2 / m\_2  
paste("m\_2:", m\_2, sep = ' ')

## [1] "m\_2: 1"

paste("N\_LP:", N\_LP, sep = ' ')

## [1] "N\_LP: 10000"

## Question 2:

Write a function to simulate the capture-recapture procedure using the inputs: , . and the number of simulation runs. The function should output in list form (a) a data frame with two columns: the values of and for each iteration and (b) . Run your simulation for iterations for a population of size where and make a histogram of the resulting vector. Indicate on the plot.

Answer:

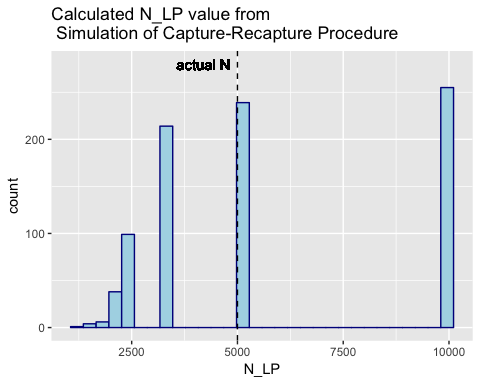
# Function to simulate the capture-recapture procedure using the inputs: N, n\_1, n\_2 and number of simulation runs  
capture\_recapture = function(N, n\_1, n\_2, runs){  
   
 # Create vector to store m\_2s and N\_LPs  
 m\_2 <- c()  
 N\_LP <- c()  
   
 # For each run, perform the capture-recapture procedure   
 for(i in 1:runs){  
   
 # Sample the captured points   
 cap\_one <- sample(1:N, n\_1, replace = FALSE)  
 cap\_two <- sample(1:N, n\_2, replace = FALSE)  
   
 # Finds the points that are recaptured   
 recaptured <- length(intersect(cap\_one, cap\_two))  
   
 # Append to m\_2 vector  
 m\_2 <- c(m\_2, recaptured)  
   
 # Calculate N\_LP and append to its vector   
 N\_LP <- c(N\_LP, (n\_1 \* n\_2 / recaptured))  
 }  
   
 # Return a list where the first element is a dataframe of m\_2 and N\_LP   
 # and the second element is the actual N  
 return(list(df = data.frame(m\_2, N\_LP), N = N))  
}

Simulation:

# Perform a simulation using 1000 runs  
simulation <- capture\_recapture(5000, 100, 100, 1000)  
df <- simulation$df  
  
# Plot a histogram of the calculated N\_LP values  
ggplot(df, aes(x = df$N\_LP)) + geom\_histogram(color = "darkblue", fill = "lightblue") + ggtitle("Calculated N\_LP value from \n Simulation of Capture-Recapture Procedure") + geom\_vline(aes(xintercept = 5000), color = "black", linetype = "dashed") + geom\_text(aes(x = simulation$N - 800, y = 280, label = "actual N")) + labs(x = "N\_LP")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 144 rows containing non-finite values (stat\_bin).



## Question 3:

What percent of the estimated population values in Question 2 were infinite? Why can this occur?

Answer:

per\_inf <- paste(100 \* length(which(is.infinite(simulation$df$N\_LP))) / nrow(simulation$df), "%", sep = ' ')  
per\_inf

## [1] "14.4 %"

The percentage of infinite calculated is 14.4 %. This occurs because in some rare cases, there was no recapture of any points, meaning . Thus in the calculation of , there is a in the denominator, causing R to give an infinite result.

## Question 4:

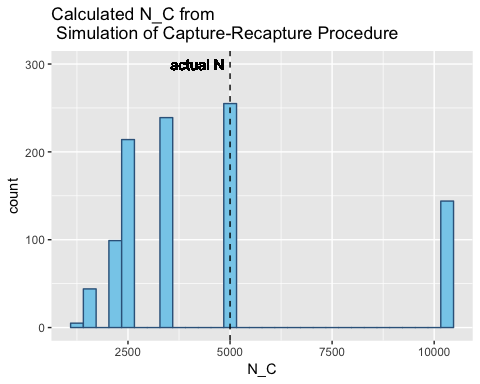
An alternative to the Lincoln-Peterson estimator is the Chapman estimator:

Use the saved values from question 2 to compute the corresponding Chapman estimates for each iteration of your simulation. Construct a histogram of the resulting estimates, indicating on the plot.

Answer:

# Calculate N\_C from the m\_2 values from the previous simulation runs   
N\_C <- data.frame(chapman = (((n\_1 + 1)\*(n\_2 + 1))/(simulation$df$m\_2 + 1)) - 1)  
  
# Plot a histogram of the calculated N\_C values  
ggplot(N\_C, aes(x = N\_C$chapman)) + geom\_histogram(color = "steelblue4", fill = "skyblue") + ggtitle("Calculated N\_C from \n Simulation of Capture-Recapture Procedure") + geom\_vline(aes(xintercept = 5000), color = "black", linetype = "dashed") + geom\_text(aes(x = simulation$N - 800, y = 300, label = "actual N")) + labs(x = "N\_C")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Question 5:

An estimator is considered *unbiased* if, on average, the estimator equals the true population value. For example, the sample mean is unbiased because on average the sample mean equals the population mean (i.e., the sampling distribution is centered around ). This is a desirable property for an estimator to have because it means our estimator is not systematically wrong. To show that an estimator is an unbiased estimator of the true value , we would need to mathematically prove that where is the expectation (i.e., theoretical average). Instead, we will investigate this property empirically by replacing the theoretical average with the sample average of the values from our simulation (i.e., where is the number of simulation runs; is in this case and is either or as both are ways to estimate ).

Estimate the bias of the Lincoln-Peterson and Chapman estimators based on the results of your simulation. Is either estimator unbiased when ?

Answer: Note that to compute the mean of , the infinites should be ignored and should be properly instated.

# Isolate the finite N\_LP values  
N\_LP\_finite <- simulation$df$N\_LP[is.finite(simulation$df$N\_LP)]  
  
# Calculate the bias of N\_LP  
bias\_LP <- mean(N\_LP\_finite) / length(N\_LP\_finite)  
  
# Calculate the bias of N\_C  
bias\_chapman <- mean(N\_C[,1]) / length(N\_C[,1])  
paste("The bias of the Lincoln-Peterson estimator is", bias\_LP, sep = ' ')

## [1] "The bias of the Lincoln-Peterson estimator is 6.54914951503354"

paste("The bias of the Chapman estimator is", bias\_chapman, sep = ' ')

## [1] "The bias of the Chapman estimator is 4.40856845873016"

Neither of the estimators are unbiased when .

## Question 6:

Based on your findings, is the Lincoln-Peterson or Chapman estimator better? Explain your answer.

Answer: Based on the findings, the Chapman estimator does better to estimate than the Lincoln-Peterson estimator. One reason is that the bias of the estimator calculated was lower than the one calculated using the Lincoln-Peterson estimator. Another reason is that the Lincoln-Peterson estimator was accustomed to calculating infinities which is theoretically not possible in real life scenarios.

## Question 7:

Explain why the assumptions (a), (b) and (c) used to make the Lincoln-Peterson estimator are unrealisitic.

Answer: Unrealistic Assumptions:

* A: When one individual is captured, it is possible that another individual is captured in a nearby vicinity as well
* B: Each individual may have factors that influence its chances of being captured, such as health or agility
* C: Births and deaths events cannot be held constant in the real world as they are natural events.