lab7.R

Feipeng Huang

2022-10-26

#Calculate a parametric 95% CI for mean bill length (in mm) for the Gentoo penguins  
#Q1  
require(palmerpenguins)

## Loading required package: palmerpenguins

alpha = 0.05  
dat\_gentoo = subset(penguins, species == "Gentoo")  
n = length(na.omit(dat\_gentoo$bill\_length\_mm))  
#n = 123  
  
#Q2  
ssd = sd(dat\_gentoo$bill\_length\_mm, na.rm = TRUE)  
#ssd = 3.081857  
  
#Q3  
alpha = 0.05  
t\_crit = abs(qt(alpha / 2, df = n - 1))  
#t\_crit = 1.9796  
  
#Q4  
sse = ssd / sqrt(n)  
~~#sse = 0.08847361~~

#sse = 0.2778817

#I had the correct code but copied the wrong number.  
  
#Q5  
ci\_radius = sse \* t\_crit  
ci = c(  
 lower = mean(dat\_gentoo$bill\_length\_mm, na.rm = TRUE) - ci\_radius,  
 upper = mean(dat\_gentoo$bill\_length\_mm, na.rm = TRUE) + ci\_radius)  
print(round(ci, 4))

## lower upper   
## 46.9548 48.0550

#Bootstrap (Q6-9)  
#install.packages("boot")  
require(boot)

## Loading required package: boot

boot\_mean = function(x, i)  
{  
 return(mean(x[i], na.rm = TRUE))  
}  
  
myboot =   
 boot(  
 data = dat\_gentoo$bill\_length\_mm,  
 statistic = boot\_mean,  
 R = 10000)  
print(myboot)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = dat\_gentoo$bill\_length\_mm, statistic = boot\_mean,   
## R = 10000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 47.50488 -0.002236324 0.2775417

quantile(  
 myboot$t,  
 c(0.025, 0.975))

## 2.5% 97.5%   
## 46.96748 48.05611

##########  
  
rm(list = ls())  
  
moths = read.csv("/Users/stonehuang/Documents/environmental\_data/data/moths.csv")  
#Q9  
rarefaction\_sampler = function(input\_dat, n\_iterations)  
{  
 n\_input\_rows = nrow(input\_dat)  
   
 results\_out = matrix(  
 nrow = n\_iterations,  
 ncol = n\_input\_rows)  
   
 # The outer loop: runs once for each bootstrap iteration. index variable is i  
 for(i in 1:n\_iterations)  
 {  
 # The inner loop: simulates increasing sampling intensity  
 # Sampling intensity ranges from 1 site to the complete count of  
 # sites in the input data (n)  
 for(j in 1:n\_input\_rows)  
 {  
 # sample the input data row indices, with replacement  
 rows\_j = sample(n\_input\_rows, size = j, replace=TRUE)  
   
 # Creates a new data matrix  
 t1 = input\_dat[rows\_j, ]  
   
 # Calculates the column sums  
 t2 = apply(t1, 2, sum)  
   
 # Counts the number of columns in which any moths were observed  
 results\_out[i, j] = sum(t2 > 0)  
 }  
 }  
 return(results\_out)  
}  
#Q10  
#Using the double loop while keeping track of what row and column should contain is the most difficult part about building the function.   
#Q11  
# Re-read my data:  
moths = read.csv("/Users/stonehuang/Documents/environmental\_data/data/moths.csv")  
rarefact = rarefaction\_sampler(moths[,-1], 10000)  
  
rare\_mean = apply(rarefact, 2, mean)  
rare\_quant = apply(rarefact, 2, quantile, probs=c(0.025, 0.975))  
rare = t(rbind(rare\_mean, rare\_quant))  
#Q12  
matplot(  
 rare,  
 type='l',  
 xlab='Number of sampling plots',  
 ylab='Number of moth species',  
 main="Bootstrap rarefaction curve of 10 rare MA moth species")  
  
legend(  
 'bottomright',  
 legend=c('mean','2.5% (lower CI)','97.5% (upper CI)'),  
 lty=c(1,2,3),col=c(1,2,3), inset=c(.1,.1))

Chart

Description automatically generated

#Q13  
#I would visit 22 sites if I want to see all of the moth species because both curves of both upper and lower confidence intervals reach 10 at 22 plots.