lab7.R

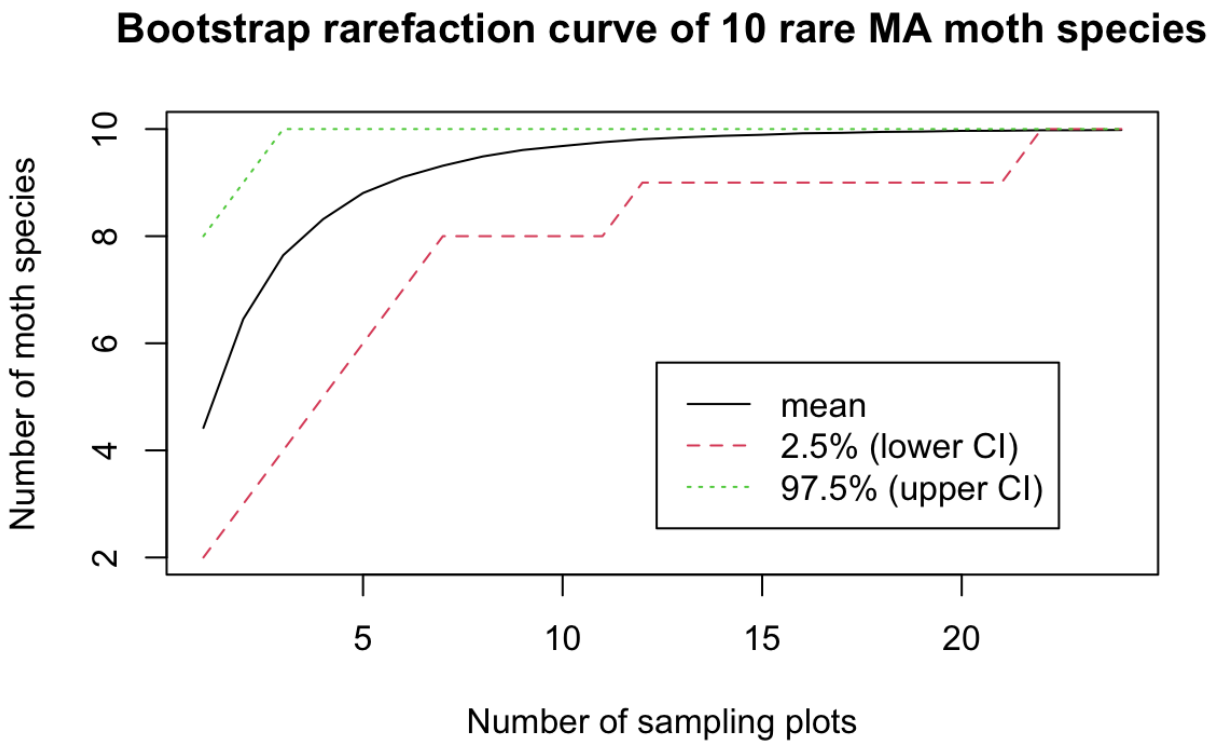
Feipeng Huang

2022-10-26

```
#Calculate a parametric 95% CI for mean bill length (in mm) for the Gentoo pe
nguins
#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.c
sv")
#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 i
```

```
s 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 cont
ain is the most difficult part about building the function.
#Q11
# Re-read my data:
moths = read.csv("/Users/stonehuang/Documents/environmental_data/data/moths.c
sv")
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))
```




```
#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.
```