Julian Burgoff

10/31/2022

Analysis of Environmental Data Lab

Lab 7

1. require(palmerpenguins)

penguins\_gentoo= penguins[penguins$species == 'Gentoo',]

n = sum(!is.na(penguins\_gentoo$bill\_length\_mm))

print(n)

[1] 123

1. std\_dev= sd(penguins\_gentoo$bill\_length\_mm, na.rm = TRUE)

print(std\_dev)

[1] 3.081857

1. t\_crit = abs(qt(alpha / 2, df = n - 1))

print(t\_crit)

[1] 1.9796

1. sse = sd(penguins\_gentoo$bill\_length\_mm, na.rm = TRUE) / sqrt(n)

print(sse)

[1] 0.2778817

1. ci\_radius = sse \* t\_crit

gentoo\_bill\_ci = c(

lower = mean(penguins\_gentoo$bill\_length\_mm, na.rm= TRUE) - ci\_radius,

upper = mean(penguins\_gentoo$bill\_length\_mm, na.rm = TRUE) + ci\_radius)

print(round(gentoo\_bill\_ci, 4))

lower upper

46.9548 48.0550

1. NA
2. boot\_mean = function(x, i)

{

return(mean(x[i], na.rm = TRUE))

}

myboot =

boot(

data = penguins\_gentoo$bill\_length\_mm,

statistic = boot\_mean,

R = 10000)

1. quantile(

myboot$t,

c(0.025, 0.975))

2.5% 97.5%

46.97071 48.05285

1. rarefaction\_sampler = function(input\_dat, n\_iterations)

{

n = nrow(moth\_dat) #number of rows or sample observations

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)

{

# sample the input data row indices, with replacement

rows\_j = sample(n, 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)

}

1. Understanding what the inner and outer loops are doing
2. moths = read.csv(here("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))

1. matplot(

rare,

type='l',

xlab='Number of sampling plots',

ylab='Species richness',

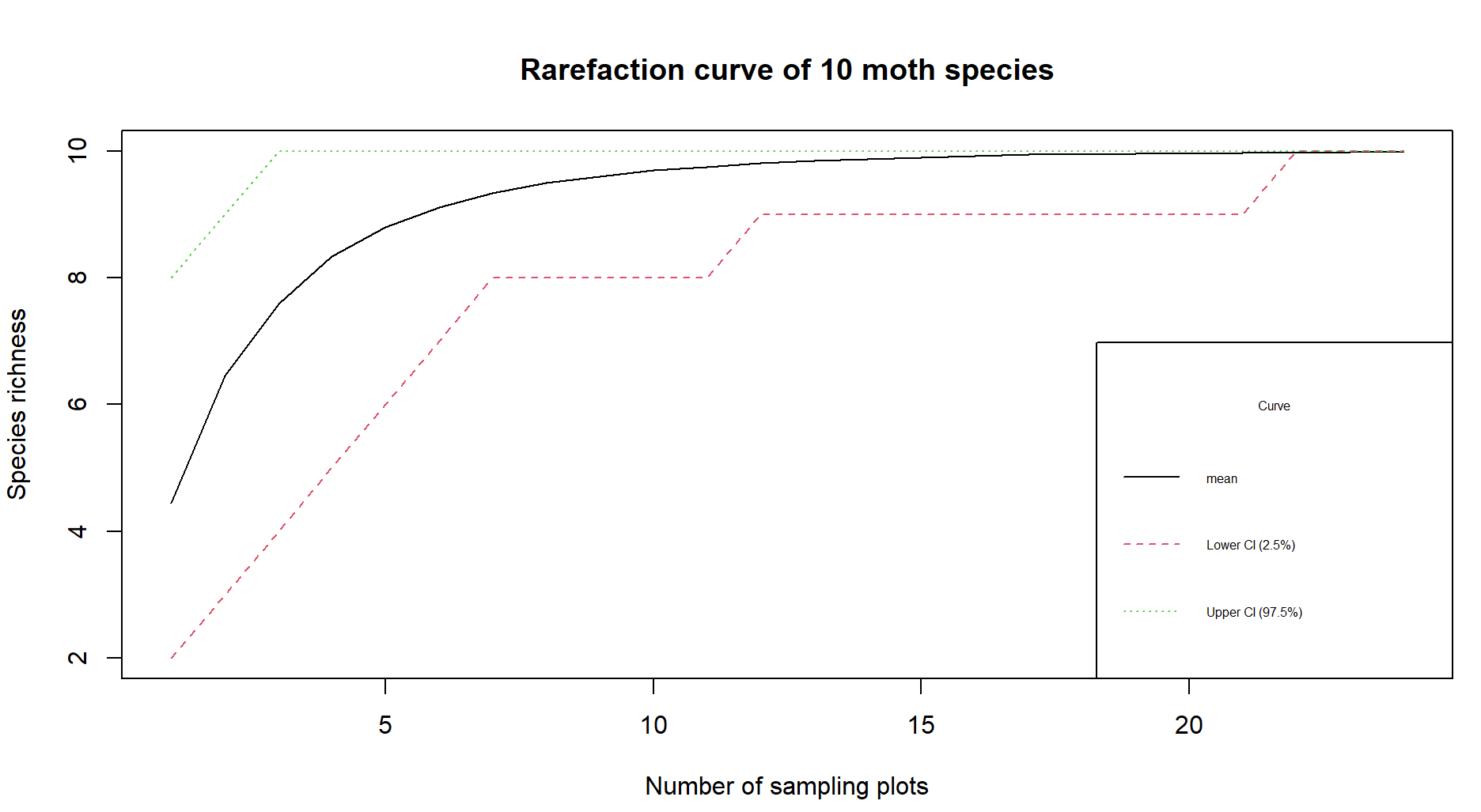
main="Rarefaction curve of 10 moth species")

legend(

'bottomright',

legend=c('mean','Lower CI (2.5%)','Upper CI (97.5%)'),

lty=c(1,2,3),col=c(1,2,3), cex= 0.5, title= "Curve")



1. Chances are pretty good that you see all ten species after visiting 10 sampling plots, the curve gets very flat after this point indicating that you don’t have a very good chance of observing more species with each additional plot.