Bonnie Turek

Eco 634 – Lab 7

10/20/21

*****I worked with Matthew F on this lab

Penguins – Parametric CI

Review the bootstrap and parametric confidence interval materials in the lab walkthrough.

Calculate a parametric 95% CI for mean bill length (in mm) for the Gentoo penguins in the penguins dataset from package palmerpenguins using your SSE function. For this calculation you should use Student's t-distribution to calculate the critical values.

Q1. What is the sample size, n? Show the code you used for the calculation and remember to check for missing data.

The sample size, n, is the number of rows or observations in the input data, so in this case it is the number of observations of bill length in Gentoo penguins (except we must check for missing or NA data). That value of **n is 123**, after one NA value was removed. I calculated this by using the length() and na.omit() functions. See code below:

```
#subsetting Gentoo penguins and bill length from total penguins data

peng = subset(penguins, species == "Gentoo")              #selecting only Gentoo species

gen_dat = peng[,1:3]              #we don't need any other data columns besides bill length

gentoo_dat = gen_dat[,-2]         #get rid of unnecessary island column


#determine the value of n (the number of observations/rows)

length(na.omit(gentoo_dat$bill_length_mm))

# can also use

length(gentoo_dat$bill_length_mm)-sum(is.na(gentoo_dat$bill_length_mm))

>123
```

n = 123

Q2. What is the sample standard deviation? Show the code you used for the calculation.

```
ssd = sd(gentoo_dat$bill_length_mm, na.rm = TRUE)
```

ssd = 3.081857

Q3. What are the critical t-values? Show the R code you used for the calculation.

The crititcal t values (since we are using the student t-distribution and NOT the normal distribution as shown in class) are -1.98 and 1.98

<u>CODE:</u>

```
# 95% confidence, Student t-dist, 5 DF, two-sided

t_lower = qt(alpha / 2, df = n-1)

t_upper = qt(1 - (alpha/2), df = n-1)

t_crit = abs(qt(alpha / 2, df = n-1))

print(c(`Critical T: lower tail` = t_lower,

      `Critical T: upper tail` = t_upper),

    digits = 3)

#check 95% density is within the upper and lower critical values using the cumulative probability function

pt(t_upper, df = n-1)

pt(t_lower, df = n-1)

pt(t_upper, df = n-1) - pt(t_lower, df = n-1)
```

<u>OUTPUT:</u>

```
Critical T: lower tail Critical T: upper tail

                 -1.98                  1.98

pt(t_upper, df = n-1)

 0.975

pt(t_lower, df = n-1)

 0.025

pt(t_upper, df = n-1) - pt(t_lower, df = n-1)

0.95
```

This all checks out!

Q4. What is the sample standard error? Show the R code you used for the calculation.

```
#define function for SSE

sse_mean = function(x)

{

  sse = sd(x,na.rm=TRUE)/sqrt(length(x)-sum(is.na(x)))

  return(sse)

}

sse = sse_mean(gentoo_dat$bill_length_mm)
```

SSE = 0.2778817

Q5. Finally, construct the CI and show the R code you used for the calculation.

CI = $x$bar ± SSE × T$crit$

```
#variables needed from above to calc CI

n = length(gentoo_dat$bill_length_mm)-sum(is.na(gentoo_dat$bill_length_mm))

sse = sse_mean(gentoo_dat$bill_length_mm)

t_crit = abs(qt(alpha / 2, df = n-1))

#construct Confidence interval

ci_radius = sse * t_crit

ci_radius

sample_mean - ci_radius

sample_mean + ci_radius
```

OUTPUT:

```
> sample_mean - ci_radius

46.95478

> sample_mean + ci_radius

48.05497
```

CI : 46.95 – 48.05 mm (bill length in observed Gentoo penguins)

Penguins – Bootstrap CI

Review the bootstrap confidence interval materials in the lab walkthrough.

Calculate a bootstrap 95% CI for mean bill length (in mm) for the Gentoo penguins in penguins dataset from package palmerpenguins.

Use the boot() function from package boot()

Q6. What is the CI?

The bootstrap confidence interval is:

```
   2.5%        97.5%
46.96475     48.04919
```

Q7. Show the r code you used to call the boot() function.

```
#Bootstrap CI
m = 10000
# numeric() creates an vector of length m with all values initialized to zero
result = numeric(m)
head(result)
#perform the bootstrap
for(i in 1:m)
{
  result[i] = mean(sample(gentoo_dat$bill_length_mm, replace=TRUE), na.rm = TRUE)
}
mean(result)
quantile(result,c(0.025,0.975))
#call boot function
require(boot)
boot_mean = function(x, i)
{
  return(mean(x[i], na.rm = TRUE))
}
```

```
pengboot =
  boot(
    data = gentoo_dat$bill_length_mm,
    statistic = boot_mean,
    R = 10000)
print(pengboot)
```

OUTPUT:

```
ORDINARY NONPARAMETRIC BOOTSTRAP
Call:
boot(data = gentoo_dat$bill_length_mm, statistic = boot_mean,
    R = 10000)
Bootstrap Statistics :
    original            bias              std. error
t1* 47.50488      -0.001643284          0.2739598
```

Q8. Show the r code you used to calculate the upper and lower 2.5% quantiles.

```
quantile(
  pengboot$t,
  c(0.025, 0.975))
```

OUTPUT:

```
2.5%          97.5%
46.96694      48.06505
```

Moths - Rarefaction Sampler

Q9. Show your completed rarefaction_sampler() function.

```
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. What did you find most difficult about building the function?

Just the fact that there is so much to the function and so many lines. It was difficult keeping track of i, j, m, and n. Especially, when all of these values are not clearly defined within the function. I went through to try and find where ‘n’ was in the function but not actually defined in order to fix or de-bug the code. I ended up having to replace the undefined ‘n’ with n_input_rows which is the number of rows of the input dataset. This then becomes the number of columns of the output matrix.

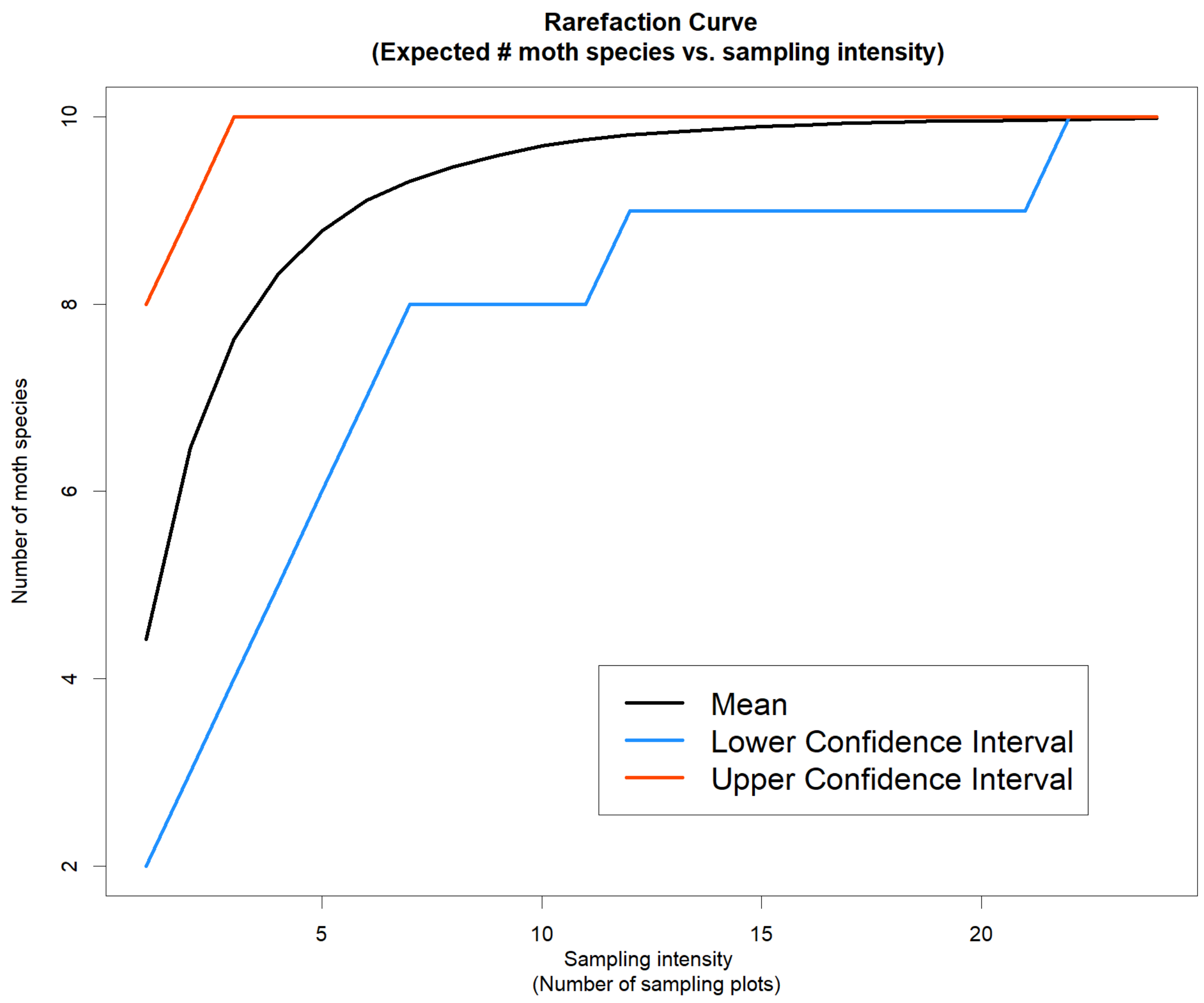
Q11. Show the code you used to perform the simulations and construct the curve.

```
#TRY WITH 10,000 iterations

# Re-read my data:

moths = read.csv(here("data", "moths.csv"))

rarefact = rarefaction_sampler(moths[,-1], 10000)          #perform 10,000 simulations


rare_mean = apply(rarefact, 2, mean)                #gets the mean value for construction of curve

rare_quant = apply(rarefact, 2, quantile, probs=c(0.025, 0.975))  #gets quantiles for curve

rare = t(rbind(rare_mean, rare_quant))              #combines mean and quantiles to plot
```

Q12. Include your rarefaction curve plot in your report. Show the R-code you used to create your plot.

```
#plotting the curve

image_file = "lab_07rarefaction.png"

png(here("images", image_file), width = 1800, height = 1500, res = 180)

matplot(

  rare,

  type='l',

  xlab='Sampling intensity

  (Number of sampling plots)',

  ylab='Number of moth species',

  col = c("black","dodgerblue","orangered"),

  lty = c(1,1,1),lwd = 3, cex = 1.5,

  main='Rarefaction Curve
```

```
 (Expected # moth species vs. sampling intensity)')
legend(
 'bottomright',
 legend=c('Mean','Lower Confidence Interval','Upper Confidence Interval' ),
 lty = c(1,1,1),col=c("black","dodgerblue","orangered"),lwd = 3,
 cex = 1.5, inset=c(.1,.1))
dev.off()
```



Q13. About how many sites should you visit if you want to see all of the moth species? Explain your reasoning using your rarefaction curve figure.

According to the rarefaction curve above, if you wanted to see all of the moth species (10 in total) you should visit at least 20 sites. The mean of the curve approaches 10 species when you have about 10-15 sampling sites, but it doesn't actually reach 10 species until you have 20 sampling sites. Although the mean curve appears to approach 10 very closely between 15-20 sampling plots, the lower confidence interval tells us there is a chance you might only observe 9 species under those conditions. You would be safest sampling at least 20 or more sites to observe all species, since the graph shows all 3 curves (mean, upper and lower confidence) reaches the 10 total species.