Chloe Lang
Professor Nelson
Lab 7 Report
October 31

1. The sample size (n) is 123.

```
dat_gentoo = subset(penguins, species =="Gentoo")
gentoo_sample = sum(!is.na(dat_gentoo$bill_length_mm))
gentoo_sample
```

2. The sample standard deviation is 3.08157

```
gentoo_SD = sd(dat_gentoo$bill_length_mm, na.rm = TRUE)
gentoo_SD
```

3. The critical t-values for 95% are -1.9796 and 1.9796.

```
gentoo_tvalue = qt(c(0.025, 0.975), df = gentoo_sample - 1, lower.tail = TRUE)
gentoo_tvalue
```

4. The sample standard error is 0.2778817.

```
gentoo_SSE = gentoo_SD / sqrt(gentoo_sample)
gentoo_SSE
```

5. The CI for the mean bill length (in mm) is 46.95478 and 48.05497.

```
mean_bill = mean(dat_gentoo$bill_length_mm, na.rm = TRUE)
mean_bill
gentoo_CI = mean_bill + gentoo_tvalue * gentoo_SSE
gentoo_CI
```

6. The CI is 46.97072 and 48.05887.

7. R code with boot () function:

```
require(boot)
boot_mean = function(x, i)
{
  return(mean(x[i], na.rm = TRUE))
}
gentoo_boot = boot(
  data = dat_gentoo$bill_length_mm, boot_mean, 10000)
gentoo_boot

alpha=0.05
gentoo_mean = mean(gentoo_boot$t)
gentoo_SSEEE = gentoo_SSE(gentoo_boot$t)
gentoo_Confidence = quantile(gentoo_boot$t, c((alpha / 2), (1 - alpha /2)))
gentoo_Confidence
```

8. R code to calculate upper and lower 2.5% quantiles:

```
quantile(gentoo_boot$t,c(0.025, 0.975))
   2.5%    97.5%
46.97072 48.05887
```

9. 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)

  for(i in 1:n_iterations)
  {
    for(j in 1:n_input_rows)
    {
      rows_j = sample(n_input_rows, size = j, replace = TRUE)
      t1 = input_dat[rows_j, ]
      t2 = apply(t1, 2, sum)
      results_out[i, j] = sum(t2 > 0)
    }
  }
  return(results_out)
}

rarefact = rarefaction_sampler(moths[,-1], 100)
head(rarefact)
```

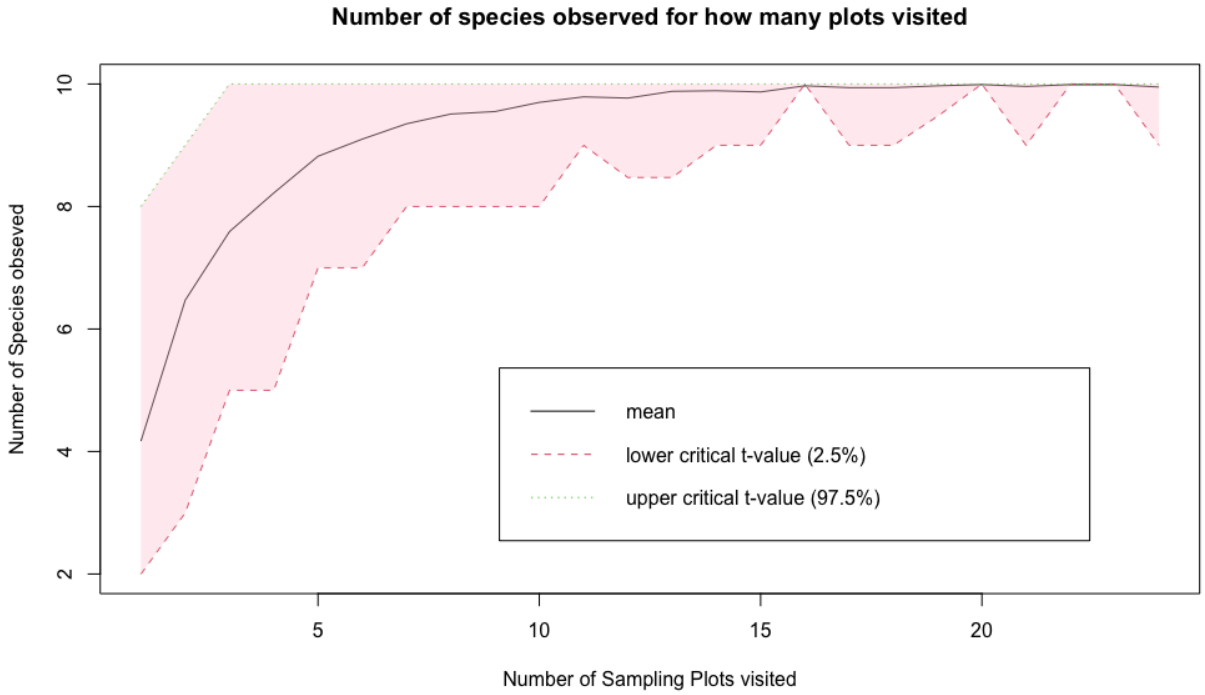
10. The most difficulty I had with building the function was making sure I stayed consistent with the walkthrough function example and making sure I matched it up correctly so there were no errors in my code. It took a couple tries with extra commas or missing parentheses.

11. Code for simulations and the curve:

```
#Simulation
rarefact = rarefaction_sampler(moths[,-1], 100)
head(rarefact)

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))

#curve & plot
x = c(1:24)
y1 = rare[,2]
y2 = rare[,3]

matplot(
```

```
  rare,
  type='l',
  xlab='Number of Sampling Plots visited',
  ylab='Number of Species obseved',
  main='Number of species observed for how many plots visited')

legend(
  'bottomright',
  legend=c('mean','lower critical t-value (2.5%)','upper critical t-value (97.5%)'),
  lty=c(1,2,3),col=c(1,2,3), inset=c(.1,.1))

polygon(c(x, rev(x)) , c(y1, rev(y2)),
        col = rgb(255, 209, 220, 100, maxColor = 255), lty = 0)
```

12. Rarefaction Curve plot:



```
matplot(
  rare,
  type='l',
  xlab='Number of Sampling Plots visited',
  ylab='Number of Species obseved',
  main='Number of species observed for how many plots visited')

legend(
  'bottomright',
  legend=c('mean','lower critical t-value (2.5%)','upper critical t-value (97.5%)'),
  lty=c(1,2,3),col=c(1,2,3), inset=c(.1,.1))

polygon(c(x, rev(x)) , c(y1, rev(y2)),
        col = rgb(255, 209, 220, 100, maxColor = 255), lty = 0)
```

13. You should visit about 23 sites if you want to see all of the moth species. This is because from the curve, the maximum value of species observed maxs out at about 23 sites as seen in the mean, black curved line.