Matt Fertakos w/ help from Bonnie

Question 1:
```
require(palmerpenguins)
dat_gentoo = subset(penguins, species == "Gentoo")
as.data.frame(dat_gentoo)->dat_gentoo
length(dat_gentoo$species)-length(sum(is.na(dat_gentoo)))
n=123
```

Question 2:
```
sd(dat_gentoo$bill_length_mm,na.rm=TRUE)
sd=3.081857
```

Question 3:
```
alpha=0.05
df=122
lower_qt = qt(0.05/2,122,lower.tail=TRUE)
upper_qt = qt(1-(0.05/2),122,lower.tail=TRUE)
lower critical t = -1.9796
upper critical t = 1.9796
```

Question 4:
```
sse_mean = function(x)
{
  sd(x,na.rm=TRUE)/sqrt(length(x)-sum(is.na(x)))
}

SSE = sse_mean(dat_gentoo$bill_length_mm)
SSE = 0.2778817
```

Question 5:
```
mean_gentoo = mean(na.omit(dat_gentoo$bill_length_mm))
mean_gentoo + (SSE*lower_qt)
mean_gentoo + (SSE*upper_qt)
CI= 47.50488 +/- 0.5500946
or 46.95 to 48.05
```

Question 6:
The CI, an abbreviation for confidence interval, is an interval found to contain the true value 95% of the time when recalculated on x repeated samplings. In this case, the CI is 46.97338 to 48.05970.

Question 7:

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

Question 8:

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

CI is 46.97338 to 48.05970

Question 9:

```
moths = read.csv(here("eco_634_2021","data","moths.csv"))
moth_dat = moths[,-1]
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)
```

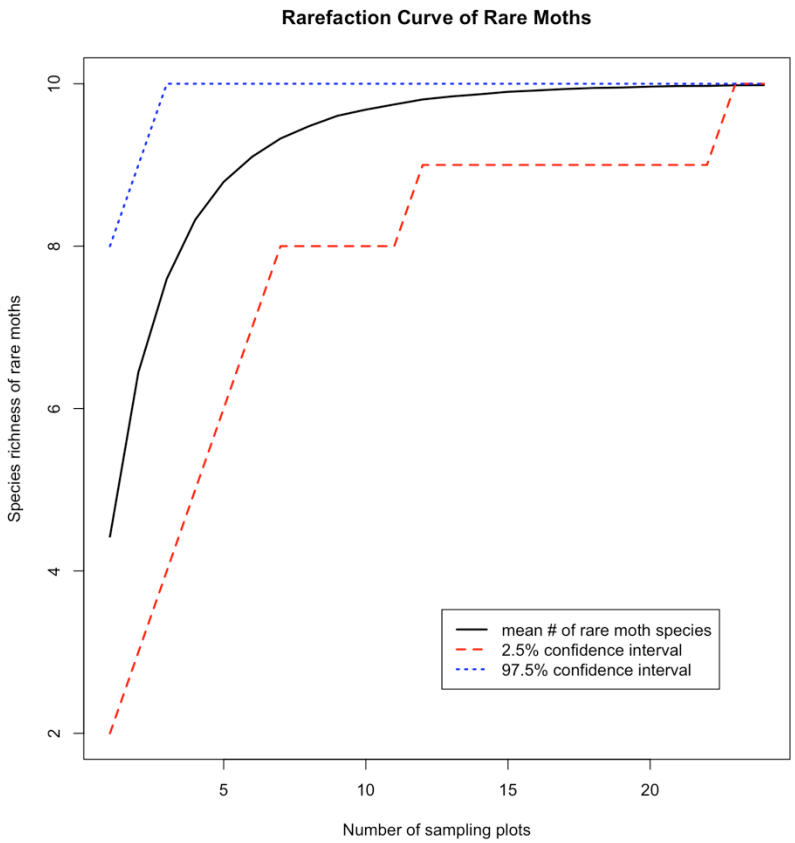
```
    }
  }
  return(results_out)
}
```

Question 10: The most difficult part of the function to build for me was figuring out how to fix the code so it knew what n was. This turn out to be the n_input_rows object which represents the number of rows in the input data.

Question 11:

```
moth_dat = moths[,-1]
rarefact = rarefaction_sampler(moth_dat, 10000)
```

Question 12:




```
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))
png(filename=here("eco_634_2021","lab_07_curve.png"),width=1500,height=1600,units="px",
res=180)
matplot(
  rare,
  type='l',
  col=c("black","red","blue"),
  lwd=c(2,2,2),
  xlab='Number of sampling plots',
  ylab='Species richness of rare moths',
  main='Rarefaction Curve of Rare Moths')

legend(
  'bottomright',
  legend=c('mean # of rare moth species','2.5% confidence interval','97.5% confidence interval'),
  lty=c(1,2,3),col=c("black","red","blue"),lwd=c(2,2,2),inset=c(.1,.1))
dev.off()
```

Question 13: At about 15 plots you have a high chance to see all 10 species of moths because the mean rarefaction curve reaches 10 at this point. You are more guaranteed to see all 10 moth species at 20 plots though, because that is the x-value where the mean rarefaction curve, as well as lower and upper confidence interval lines intersect.