5. Worksheet: Alpha Diversity

Joy O'Brien; Z620: Quantitative Biodiversity, Indiana University

25 January, 2023

OVERVIEW

In this exercise, we will explore aspects of local or site-specific diversity, also known as alpha (α) diversity. First we will quantify two of the fundamental components of (α) diversity: **richness** and **evenness**. From there, we will then discuss ways to integrate richness and evenness, which will include univariate metrics of diversity along with an investigation of the **species abundance distribution (SAD)**.

Directions:

1. In the Markdown version of this document in your cloned repo, change "Student Name" on line 3 (above) to your name.
2. Complete as much of the worksheet as possible during class.
3. Use the handout as a guide; it contains a more complete description of data sets along with the proper scripting needed to carry out the exercise.
4. Answer questions in the worksheet. Space for your answer is provided in this document and indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio (color may vary if you changed the editor theme).
5. Before you leave the classroom, **push** this file to your GitHub repo.
6. For the assignment portion of the worksheet, follow the directions at the bottom of this file.
7. When you are done, **Knit** the text and code into a PDF file.
8. After Knitting, submit the completed exercise by creating a **pull request** via GitHub. Your pull request should include this file `AlphaDiversity_Worskheet.Rmd` and the PDF output of `Knitr` (`AlphaDiversity_Worskheet.pdf`).

1) R SETUP

In the R code chunk below, please provide the code to: 1) Clear your R environment, 2) Print your current working directory, 3) Set your working directory to your `5.AlphaDiversity` folder, and 4) Load the `vegan` R package (be sure to install first if you haven't already).

```
rm(list = ls())
getwd()
```

```
## [1] "/Users/joyobrien/GitHub/QB2023_OBrien/2.Worksheets/5.AlphaDiversity"
```

```
setwd("~/GitHub/QB2023_OBrien/2.Worksheets/5.AlphaDiversity")
# install.packages("vegan")
require("vegan")
```

```
## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-4
```

2) LOADING DATA

In the R code chunk below, do the following: 1) Load the BCI dataset, and 2) Display the structure of the dataset (if the structure is long, use the `max.level = 0` argument to show the basic information).

```
data(BCI)
str(BCI)
```

```
## 'data.frame':    50 obs. of  225 variables:
##  $ Abarema.macradenia              : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Vachellia.melanoceras           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acalypha.diversifolia           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acalypha.macrostachya           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Adelia.triloba                  : int  0 0 0 3 1 0 0 0 5 0 ...
##  $ Aegiphila.panamensis            : int  0 0 0 0 1 0 1 0 0 1 ...
##  $ Alchornea.costaricensis         : int  2 1 2 18 3 2 0 2 2 2 ...
##  $ Alchornea.latifolia             : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Alibertia.edulis                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Allophylus.psilospermus         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Alseis.blackiana                : int  25 26 18 23 16 14 18 14 16 14 ...
##  $ Amaioua.corymbosa               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Anacardium.excelsum             : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Andira.inermis                  : int  0 0 0 0 1 1 0 0 1 0 ...
##  $ Annona.spraguei                 : int  1 0 1 0 0 0 0 1 1 0 ...
##  $ Apeiba.glabra                   : int  13 12 6 3 4 10 5 4 5 5 ...
##  $ Apeiba.tibourbou                : int  2 0 1 1 0 0 0 1 0 0 ...
##  $ Aspidosperma.desmanthum         : int  0 0 0 1 1 1 0 0 0 1 ...
##  $ Astrocaryum.standleyanum        : int  0 2 1 5 6 2 2 0 2 1 ...
##  $ Astronium.graveolens            : int  6 0 1 3 0 1 2 2 0 0 ...
##  $ Attalea.butyracea               : int  0 1 0 0 0 1 1 0 0 0 ...
##  $ Banara.guianensis               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Beilschmiedia.pendula           : int  4 5 7 5 8 6 5 9 11 14 ...
##  $ Brosimum.alicastrum             : int  5 2 4 3 2 2 6 4 3 6 ...
##  $ Brosimum.guianense              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Calophyllum.longifolium         : int  0 2 0 2 1 2 2 2 2 0 ...
##  $ Casearia.aculeata               : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Casearia.arborea                : int  1 1 3 2 4 1 2 3 9 7 ...
##  $ Casearia.commersoniana          : int  0 0 1 0 1 0 0 0 1 0 ...
##  $ Casearia.guianensis             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Casearia.sylvestris             : int  2 1 0 0 0 3 1 0 1 1 ...
##  $ Cassipourea.guianensis          : int  2 0 1 1 3 4 4 0 2 1 ...
##  $ Cavanillesia.platanifolia       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cecropia.insignis               : int  12 5 7 17 21 4 0 7 2 16 ...
##  $ Cecropia.obtusifolia            : int  0 0 0 0 1 0 0 2 0 2 ...
```

```
##  $ Cedrela.odorata               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ceiba.pentandra               : int  0 1 1 0 1 0 0 1 0 1 ...
##  $ Celtis.schippii               : int  0 0 0 2 2 0 1 0 0 0 ...
##  $ Cespedesia.spathulata         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chamguava.schippii            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chimarrhis.parviflora         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Maclura.tinctoria             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chrysochlamys.eclipes         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chrysophyllum.argenteum       : int  4 1 2 2 6 2 3 2 4 2 ...
##  $ Chrysophyllum.cainito         : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Coccoloba.coronata            : int  0 0 0 1 2 0 0 1 2 1 ...
##  $ Coccoloba.manzinellensis      : int  0 0 0 0 0 0 0 2 0 0 ...
##  $ Colubrina.glandulosa          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cordia.alliodora              : int  2 3 3 7 1 1 2 0 0 2 ...
##  $ Cordia.bicolor                : int  12 14 35 23 13 7 5 10 7 13 ...
##  $ Cordia.lasiocalyx             : int  8 6 6 11 7 6 6 3 0 4 ...
##  $ Coussarea.curvigemma          : int  0 0 0 1 0 2 1 0 1 1 ...
##  $ Croton.billbergianus          : int  2 2 0 11 6 0 0 4 2 0 ...
##  $ Cupania.cinerea               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cupania.latifolia             : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Cupania.rufescens             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cupania.seemannii             : int  2 2 1 0 3 0 1 2 2 0 ...
##  $ Dendropanax.arboreus          : int  0 3 6 0 5 2 1 6 1 3 ...
##  $ Desmopsis.panamensis          : int  0 0 4 0 0 0 0 0 0 1 ...
##  $ Diospyros.artanthifolia       : int  1 1 1 1 0 0 0 0 0 1 ...
##  $ Dipteryx.oleifera             : int  1 1 3 0 0 0 0 2 1 2 ...
##  $ Drypetes.standleyi            : int  2 1 2 0 0 0 0 0 0 0 ...
##  $ Elaeis.oleifera               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Enterolobium.schomburgkii     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Erythrina.costaricensis       : int  0 0 0 0 0 3 0 0 1 0 ...
##  $ Erythroxylum.macrophyllum     : int  0 1 0 0 0 0 0 1 1 1 ...
##  $ Eugenia.florida               : int  0 1 0 7 2 0 0 1 1 3 ...
##  $ Eugenia.galalonensis          : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Eugenia.nesiotica             : int  0 0 1 0 0 0 5 4 3 0 ...
##  $ Eugenia.oerstediana           : int  3 2 5 1 5 2 2 3 3 3 ...
##  $ Faramea.occidentalis          : int  14 36 39 39 22 16 38 41 33 42 ...
##  $ Ficus.colubrinae              : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Ficus.costaricana             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.insipida                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.maxima                  : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.obtusifolia             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.popenoei                : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Ficus.tonduzii                : int  0 0 1 2 1 0 0 0 0 0 ...
##  $ Ficus.trigonata               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.yoponensis              : int  1 0 0 0 0 1 1 0 0 0 ...
##  $ Garcinia.intermedia           : int  0 1 1 3 2 1 2 2 1 0 ...
##  $ Garcinia.madruno              : int  4 0 0 0 1 0 0 0 0 1 ...
##  $ Genipa.americana              : int  0 0 1 0 0 0 1 0 1 1 ...
##  $ Guapira.myrtiflora            : int  3 1 0 1 1 7 3 1 1 1 ...
##  $ Guarea.fuzzy                  : int  1 1 0 1 3 0 0 2 0 3 ...
##  $ Guarea.grandifolia            : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Guarea.guidonia               : int  2 6 2 5 3 4 4 0 1 5 ...
##  $ Guatteria.dumetorum           : int  6 16 6 3 9 7 8 6 2 2 ...
##  $ Guazuma.ulmifolia             : int  0 0 0 1 0 0 0 0 0 0 ...
```

```
##  $ Guettarda.foliacea             : int  1 5 1 2 1 0 0 4 1 3 ...
##  $ Gustavia.superba               : int  10 5 0 1 3 1 8 4 4 4 ...
##  $ Hampea.appendiculata           : int  0 0 1 0 0 0 0 0 2 1 ...
##  $ Hasseltia.floribunda           : int  5 9 4 11 9 2 7 6 3 4 ...
##  $ Heisteria.acuminata            : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ Heisteria.concinna             : int  4 5 4 6 4 8 2 5 1 5 ...
##  $ Hirtella.americana             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Hirtella.triandra              : int  21 14 5 4 6 6 7 14 8 7 ...
##  $ Hura.crepitans                 : int  0 0 0 0 0 2 1 1 0 0 ...
##  $ Hieronyma.alchorneoides        : int  0 2 0 0 0 0 0 0 1 0 ...
##   [list output truncated]
##  - attr(*, "original.names")= chr [1:225] "Abarema.macradenium" "Acacia.melanoceras" "Acalypha.divers
```

3) SPECIES RICHNESS

Species richness (S) refers to the number of species in a system or the number of species observed in a sample.

Observed richness

In the R code chunk below, do the following:

1. Write a function called `S.obs` to calculate observed richness
2. Use your function to determine the number of species in `site1` of the BCI data set, and
3. Compare the output of your function to the output of the `specnumber()` function in `vegan`.

```
# Writing function
S.obs <- function(x = ""){
  rowSums(x > 0     ) * 1
}
# Determine species in site1
S.obs(BCI[1, ])
```

```
##  1
## 93
```

```
# Creating site1
site1 <- BCI[1, ]
#Comparing to vegan function
specnumber(BCI[1, ]) # gives the same output as the function we wrote
```

```
##  1
## 93
```

```
# Calculating species richness for the first four sites of the data
specnumber(BCI[1:4, ])
```

```
##  1  2  3  4
## 93 84 90 94
```

Question 1: Does `specnumber()` from `vegan` return the same value for observed richness in `site1` as our function `S.obs`? What is the species richness of the first four sites (i.e., rows) of the BCI matrix?

> ***Answer 1***: Yes, the 'specnumber()' function does return the same value for the observed richness function we wrote, S.obs. The species richness of the first four sites of the BCI matrix is 93, 84, 90, and 94.

Coverage: How well did you sample your site?

In the R code chunk below, do the following:

1. Write a function to calculate Good's Coverage, and
2. Use that function to calculate coverage for all sites in the BCI matrix.

```
# Function for Good's coverage
C <- function(x = ""){
  1 - rowSums(x == 1) / rowSums(x)
}
# Using the function to calculate coverage for BCI data
C(BCI)
```

```
##         1         2         3         4         5         6         7         8
## 0.9308036 0.9287356 0.9200864 0.9468504 0.9287129 0.9174757 0.9326923 0.9443155
##         9        10        11        12        13        14        15        16
## 0.9095355 0.9275362 0.9152120 0.9071038 0.9242054 0.9132420 0.9350649 0.9267735
##        17        18        19        20        21        22        23        24
## 0.8950131 0.9193084 0.8891455 0.9114219 0.8946078 0.9066986 0.8705882 0.9030612
##        25        26        27        28        29        30        31        32
## 0.9095023 0.9115479 0.9088729 0.9198966 0.8983516 0.9221053 0.9382423 0.9411765
##        33        34        35        36        37        38        39        40
## 0.9220183 0.9239374 0.9267887 0.9186047 0.9379310 0.9306488 0.9268868 0.9386503
##        41        42        43        44        45        46        47        48
## 0.8880597 0.9299517 0.9140049 0.9168704 0.9234234 0.9348837 0.8847059 0.9228916
##        49        50
## 0.9086651 0.9143519
```

Question 2: Answer the following questions about coverage:

a. What is the range of values that can be generated by Good's Coverage?
b. What would we conclude from Good's Coverage if n_i equaled N?
c. What portion of taxa in `site1` was represented by singletons?
d. Make some observations about coverage at the BCI plots.

> ***Answer 2a***: 0 to 1
>
> ***Answer 2b***: We would conclude that the Good's Coverage is zero and that each species is a singleton.
>
> ***Answer 2c***: Approximately 7%
>
> ***Answer 2d***: The coverage of the BCI plots are high which means that most species found within the plots were found > more than once. I think we can say that these sites were sampled well given the Good's coverage values.

Estimated richness

In the R code chunk below, do the following:

1. Load the microbial dataset (located in the `5.AlphaDiversity/data` folder),
2. Transform and transpose the data as needed (see handout),
3. Create a new vector (`soilbac1`) by indexing the bacterial OTU abundances of any site in the dataset,
4. Calculate the observed richness at that particular site, and
5. Calculate coverage of that site

```
# Loading data
soilbac <- as.matrix(read.table("data/soilbac.txt", sep = "\t", header = TRUE, row.names = 1))
# Transform
soilbac_t <- as.data.frame(t(soilbac))
soilbac1 <- soilbac_t[1, ]

# Calculating observed richness
specnumber(soilbac1)
```

```
## T1_1
## 1074
```

```
# Calculating Good's Coverage
C(soilbac1)
```

```
##      T1_1
## 0.6479471
```

```
rowSums(soilbac1)
```

```
## T1_1
## 2119
```

Question 3: Answer the following questions about the soil bacterial dataset.

a. How many sequences did we recover from the sample `soilbac1`, i.e. N?
b. What is the observed richness of `soilbac1`?
c. How does coverage compare between the BCI sample (`site1`) and the KBS sample (`soilbac1`)?

Answer 3a: Using the rowSums function, we recovered 2119 sequences.

Answer 3b: 1074

Answer 3c: The KBS sample has a coverage of 65 %, and the BCI sample has a coverage of 93%. The coverage of the KBS sample is lower than the coverage in the BCI sample.

Richness estimators

In the R code chunk below, do the following:

1. Write a function to calculate **Chao1**,
2. Write a function to calculate **Chao2**,
3. Write a function to calculate **ACE**, and
4. Use these functions to estimate richness at `site1` and `soilbac1`.

```
# Chao1 function
S.chao1 <- function(x = ""){
  S.obs(x) + (sum(x == 1)^2) / (2 * sum(x == 2))
}
# Chao2 function
S.chao2 <- function(site = "", SbyS = ""){
  SbyS = as.data.frame(SbyS)
  x = SbyS[site, ]
  SbyS.pa <- (SbyS > 0) * 1 #convert to presence absence
  Q1 = sum(colSums(SbyS.pa) == 1) #species observed once
  Q2 = sum(colSums(SbyS.pa) == 2) #species observed twice
  S.chao2 = S.obs(x) + (Q1^2)/(2 * Q2)
  return(S.chao2)
}
# ACE function
S.ace <- function(x = "", thresh = 10){
  x <- x[x>0]
  S.abund <- length(which(x > thresh))
  S.rare <- length(which(x <= thresh))
  singlt <- length(which(x == 1))
  N.rare <- sum(x[which(x <= thresh)])
  C.ace <- 1 - (singlt / N.rare)
  i <- c(1:thresh)
  count <- function(i, y){
    length(y[y == 1])
  }
  a.1 <- sapply(i, count, x)
  f.1 <- (i * (i-1)) * a.1
  G.ace <- (S.rare/C.ace) * (sum(f.1)/(N.rare*(N.rare-1)))
  S.ace <- S.abund + (S.rare/C.ace) + (singlt/C.ace) * max(G.ace, 0)
  return(S.ace)
}
# Estimating richness at site 1 and soil bac
# Soilbac1

S.chao1(soilbac1) #2628.5
```

```
##     T1_1
## 2628.514
```

```
S.chao2(site = "T1_1", soilbac_t) # 21055.39
```

```
##     T1_1
## 21055.39
```

```
S.ace(soilbac1) # 215604.7
```

```
## [1] 215604.7
```

```
S.chao1(site1) #119.69
```

```
##        1
## 119.6944
```

```
S.chao2("1", BCI) # 104.60
```

```
##        1
## 104.6053
```

```
S.ace(site1) #918.46
```

```
## [1] 918.4679
```

Question 4: What is the difference between ACE and the Chao estimators? Do the estimators give consistent results? Which one would you choose to use and why?

> ***Answer 4***: There is a lot of variation within the output of the Chao and ACE richness estimators. The difference between ACE and Chao is that ACE uses a threshold to look at the abundance of rare species, while the Chao estimator is based on the number of singletons and doubletons that are present. I think the choice > to use one or the other depends on the type of sample and goal of estimating richness. In my opinion, I would choose ACE for my QB project because I am interested in > the richness of rare species.

Rarefaction

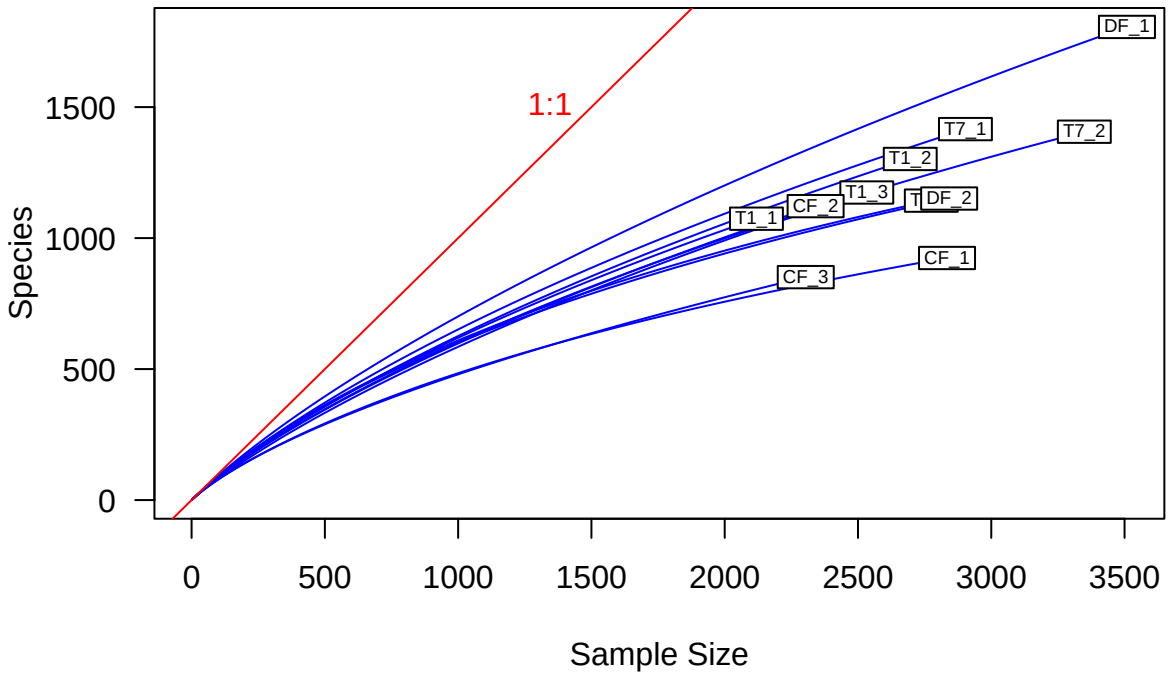
In the R code chunk below, please do the following:

1. Calculate observed richness for all samples in `soilbac`,
2. Determine the size of the smallest sample,
3. Use the `rarefy()` function to rarefy each sample to this level,
4. Plot the rarefaction results, and
5. Add the 1:1 line and label.

```r
soilbac.S <- S.obs(soilbac_t)

# Determining smallest sample size
min.N <- min(rowSums(soilbac_t))
# Rarifying to rarefy each sample to this level
S.rarefy <- rarefy(x = soilbac_t, sample = min.N, se = TRUE)

# Plotting rarefaction results
rarecurve(x = soilbac_t, step = 20, col = "blue", cex = 0.6, las = 1)
abline(0, 1, col = 'red')
text(1500, 1500, "1:1", pos = 2, col = 'red')
```



4) SPECIES EVNENNESS

Here, we consider how abundance varies among species, that is, **species evenness**.

Visualizing evenness: the rank abundance curve (RAC)

One of the most common ways to visualize evenness is in a **rank-abundance curve** (sometime referred to as a rank-abundance distribution or Whittaker plot). An RAC can be constructed by ranking species from the most abundant to the least abundant without respect to species labels (and hence no worries about 'ties' in abundance).

In the R code chunk below, do the following:

1. Write a function to construct a RAC,
2. Be sure your function removes species that have zero abundances,
3. Order the vector (RAC) from greatest (most abundant) to least (least abundant), and
4. Return the ranked vector

```
RAC <- function(x = ""){
  x.ab = x[x > 0]
  x.ab.ranked = x.ab[order(x.ab, decreasing = TRUE)]
  as.data.frame(lapply(x.ab.ranked, unlist))
  return(x.ab.ranked)
}
```

Now, let us examine the RAC for `site1` of the BCI data set.

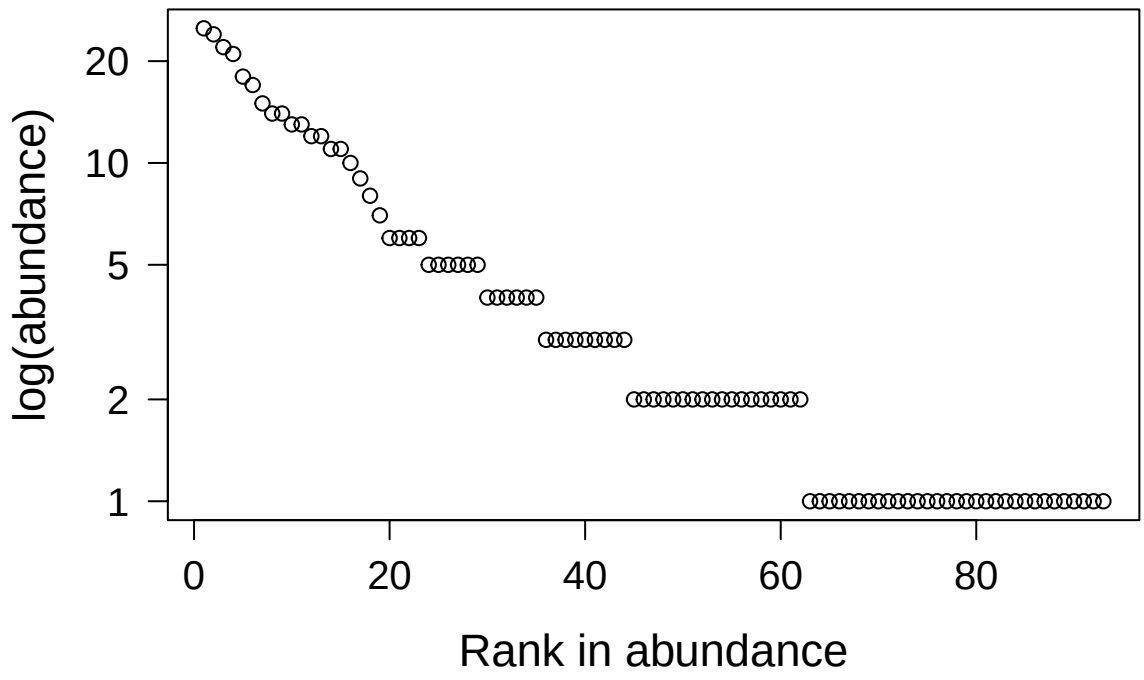
In the R code chunk below, do the following:

1. Create a sequence of ranks and plot the RAC with natural-log-transformed abundances,
2. Label the x-axis "Rank in abundance" and the y-axis "log(abundance)"

```
plot.new()
site1 <- BCI[1, ]

rac <- RAC(x = site1)
ranks <- as.vector(seq(1, length(rac)))
opar <- par(no.readonly = TRUE)
par(mar = c(5.1, 5.1, 4.1, 2.1))
plot(ranks, log(rac), type = 'p', axes = F,
     xlab = "Rank in abundance", ylab = "log(abundance)",
     las = 1, cex.lab = 1.4, cex.axis = 1.25)

box()
axis(side = 1, labels = T, cex.axis = 1.25)
axis(side = 2, las = 1, cex.axis = 1.25,
     labels = c(1, 2, 5, 10, 20), at = log(c(1, 2,5, 10, 20)))
```




```
par <- opar
```

Question 5: What effect does visualizing species abundance data on a log-scaled axis have on how we interpret evenness in the RAC?

> ***Answer 5***: Viewing the species abundance data on a log-scaled axis lessens the skewness of the data. From this we can conclude that there are high abundant species and less abundant species because abundance decreases and there is a long tail of low abundances. Therefore, we could say that the evenness is low.

Now that we have visualized unevennes, it is time to quantify it using Simpson's evenness ($E_{1/D}$) and Smith and Wilson's evenness index (E_{var}).

Simpson's evenness ($E_{1/D}$)

In the R code chunk below, do the following:

1. Write the function to calculate $E_{1/D}$, and
2. Calculate $E_{1/D}$ for `site1`.

```
SimpE <- function(x = ""){
  S <- S.obs(x)
  x = as.data.frame(x)
```

```
  D <- diversity(x, "inv")
  E <- (D)/S
  return(E)
  }
# Calculating Simpson's evenness for site1
site1 <- BCI[1, ]
SimpE(site1)
```

```
##         1
## 0.4238232
```

Smith and Wilson's evenness index (E_{var})

In the R code chunk below, please do the following:

1. Write the function to calculate E_{var},
2. Calculate E_{var} for `site1`, and
3. Compare $E_{1/D}$ and E_{var}.

```
Evar <- function(x){
  x <- as.vector(x[x > 0])
  1 - (2/pi) * atan(var(log(x)))
}

Evar(site1)
```

```
## [1] 0.5067211
```

Question 6: Compare estimates of evenness for `site1` of BCI using $E_{1/D}$ and E_{var}. Do they agree? If so, why? If not, why? What can you infer from the results.

> ***Answer 6***: The estimate of evenness for site1 of BCI using Simpson's metric is 0.42 and the estimate of evenness for site1 of BCI using Smith and Wilson's Evenness index is 0.5. These two values are close but they do not agree and this may be because Simpson's evenness is sensitive to differences in the few most abundant species and the Smith and Wilson's Evenness index is less biased towards abundant organisms because the abundances are transformed (natural log). From these results, we can infer that there is a moderate amount of evenness which means that abundances of organisms do vary within site 1 of the BCI data.

5) INTEGRATING RICHNESS AND EVENNESS: DIVERSITY METRICS

So far, we have introduced two primary aspects of diversity, i.e., richness and evenness. Here, we will use popular indices to estimate diversity, which explicitly incorporate richness and evenness We will write our own diversity functions and compare them against the functions in `vegan`.

Shannon's diversity (a.k.a., Shannon's entropy)

In the R code chunk below, please do the following:

1. Provide the code for calculating H' (Shannon's diversity),
2. Compare this estimate with the output of `vegan`'s diversity function using method = "shannon".

```
ShanH <- function(x = ""){
  H = 0
  for (n_i in x){
    if(n_i > 0){
      p = n_i / sum(x)
      H = H - p*log(p)
    }
  }
  return(H)
}
ShanH(site1)
```

```
## [1] 4.018412
```

```
# Comparing to vegan
diversity(site1, index = "shannon")
```

```
## [1] 4.018412
```

Simpson's diversity (or dominance)

In the R code chunk below, please do the following:

1. Provide the code for calculating D (Simpson's diversity),
2. Calculate both the inverse (1/D) and 1 - D,
3. Compare this estimate with the output of `vegan's` diversity function using method = "simp".

```
SimpD <- function(x = ""){
  D = 0
  N = sum(x)
  for (n_i in x){
    D = D + (n_i^2) /(N^2)
  }
  return(D)
}

D.inv <- 1/SimpD(site1)
D.sub <- 1-SimpD(site1)

# Comparing to vegan
diversity(site1, "inv")
```

```
## [1] 39.41555
```

```
diversity(site1, "simp")
```

```
## [1] 0.9746293
```

Fisher's α

In the R code chunk below, please do the following:

1. Provide the code for calculating Fisher's α,
2. Calculate Fisher's α for `site1` of BCI.

```
rac <- as.vector(site1[site1 > 0])
invD <- diversity(rac, "inv")
invD
```

```
## [1] 39.41555
```

```
Fisher <- fisher.alpha(rac)
Fisher
```

```
## [1] 35.67297
```

Question 7: How is Fisher's α different from $E_{H'}$ and E_{var}? What does Fisher's α take into account that $E_{H'}$ and E_{var} do not?

> ***Answer 7***: Fisher's alpha is different from Shannon Diversity and the Smith/Wilson's evenness index because it estimates diversiversty instead of calculating a diversity metric. Therefore, Fisher's alpha accounts for sampling error because it is merely an estimation.

6) HILL NUMBERS

Remember that we have learned about the advantages of Hill Numbers to measure and compare diversity among samples. We also learned to explore the effects of rare species in a community by examining diversity for a series of exponents q.

Question 8: Using `site1` of BCI and `vegan` package, a) calculate Hill numbers for q exponent 0, 1 and 2 (richness, exponential Shannon's entropy, and inverse Simpson's diversity). b) Interpret the effect of rare species in your community based on the response of diversity to increasing exponent q.

```
# Simulate communities
C1 <- data.frame(t(rep(1, 500))); colnames(C1) <- paste("sp", 1:500)
C2 <- data.frame(t(rep(1, 250))); colnames(C2) <- paste("sp", 1:250)
specnumber(C1)
```

```
## [1] 500
```

```r
specnumber(C2)
```

```
## [1] 250
```

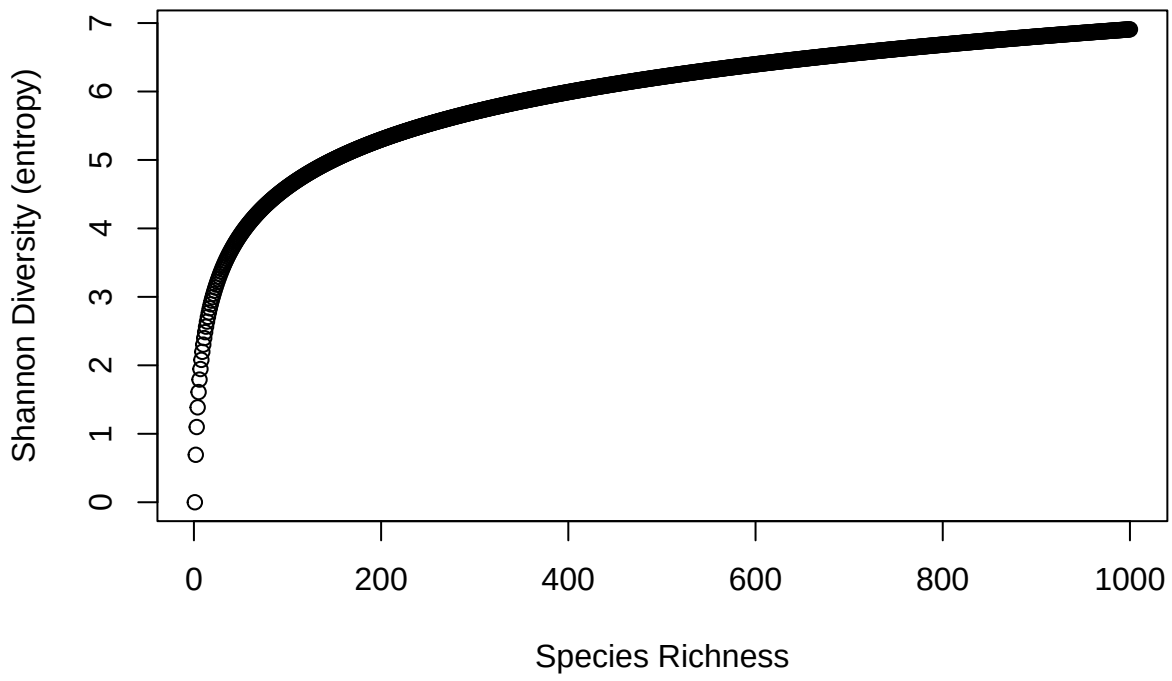
```r
# Calculate shannon diversity
H1 <- diversity(C1, index = "shannon")
H2 <- diversity(C2, index = "shannon")
H1; H2
```

```
## [1] 6.214608
```

```
## [1] 5.521461
```

```r
# Calculate shannon's entropy
H_all <- matrix(ncol = 2, nrow = 1000)
for(i in 1:1000) {
  C <- data.frame(t(rep(1, i)))
  colnames(C) = paste("sp", 1:i)
  H_all[i, 1] <- i
  H_all[i,2] <- diversity(C, index = "shannon")
}

plot(H_all[,1], H_all[,2], xlab = "Species Richness",
     ylab = "Shannon Diversity (entropy)")
```

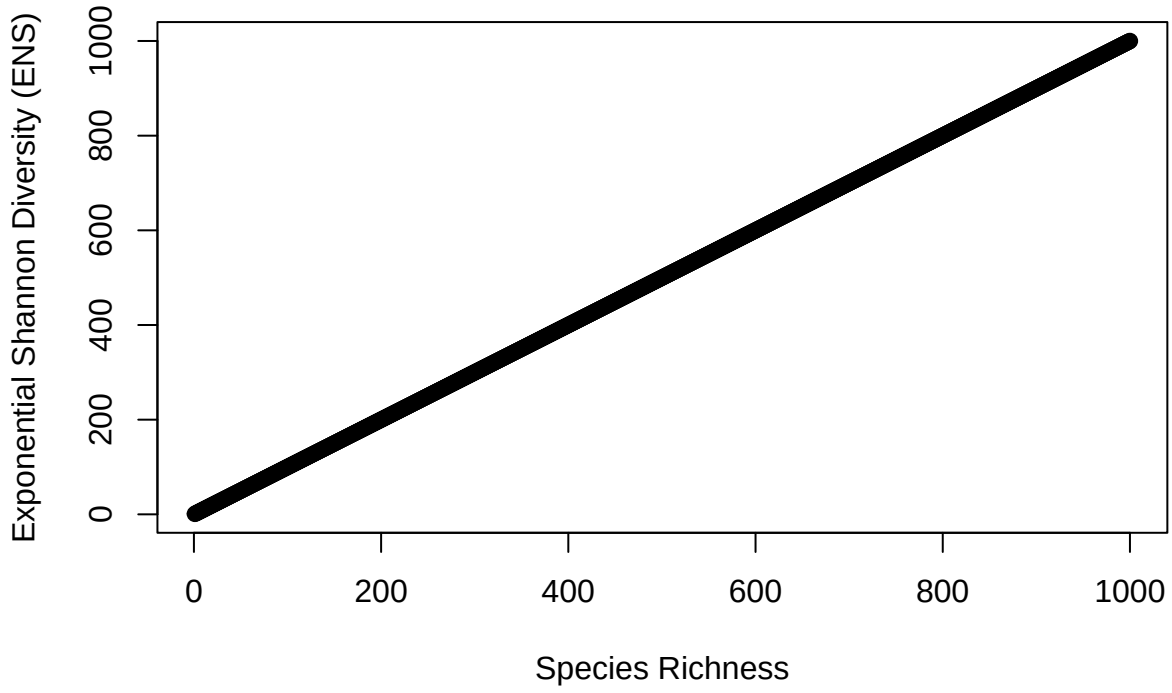


```r
# Calculate exponential Shannon's entropy (equal to the Hill number q = 1)
H1_Hill <- exp(diversity(C1, index = "shannon"))
H2_Hill <- exp(diversity(C2, index = "shannon"))
H1_Hill; H2_Hill
```

```
## [1] 500
```

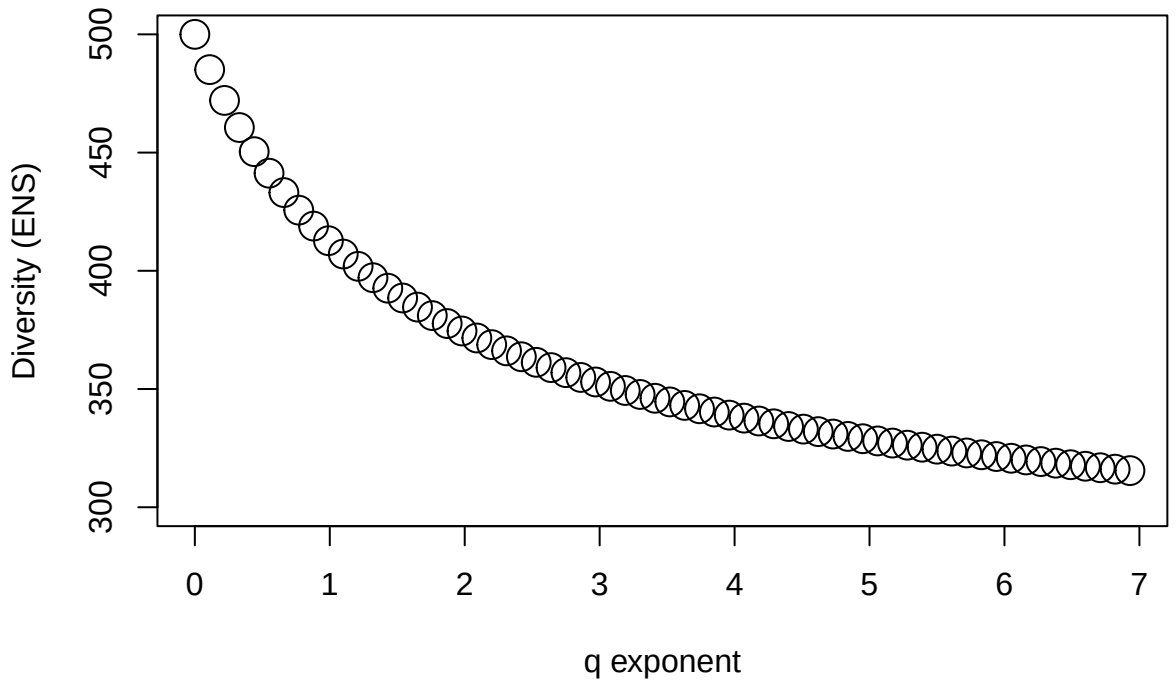
```
## [1] 250
```

```r
# Calculate for each richness level to compare Shannon entropy with Hill number 1 (exponential)
H_all_Hill <- matrix(ncol = 2, nrow = 1000)
for(i in 1:1000) {
  C = data.frame(t(rep(1, i)))
  H_all_Hill[i, 1] = i
  H_all_Hill[i, 2] = exp(diversity(C, index = "shannon"))
}
plot(H_all_Hill[,1], H_all_Hill[,2], xlab = "Species Richness",
     ylab = "Exponential Shannon Diversity (ENS)")
```




```r
# Writing the 'profile' function to calculate diversity from the equation along the continuum of q valu
profile <- function(C) {
  cbind(seq(0, 7, by = 0.11),
        unlist(lapply(seq(0, 7, by = 0.11), function(q) sum(apply(C, 1, function(x)
          (x/sum(x))^q))^(1/(1 - q))))) }
```

```
set.seed(42)
C3 <- data.frame(t(sample(1:1000, 500))); colnames(C3) = paste("sp", 1:500)
C3_profile <- profile(C3)
plot(C3_profile[,1], C3_profile[,2], ylim=c(300,500), cex = 2,
     xlab = "q exponent", ylab = "Diversity (ENS)")
```



> ***Answer 8a***: When q is 0 (species richness), the ENS value is approximately 500. When q is 1 (exponential Shannon's entropy), the ENS value is approxumately 400. When q is 2 (inverse Simpson's diversity), the ENS value is approximately 375. > ***Answer 8b***: As q increases, diversity decreases. However, a lower q exponent may be biased towards rare species and give a similar weight to the abundant species. Therefore, diversity would increase if rare species are favored.

##7) MOVING BEYOND UNIVARIATE METRICS OF α DIVERSITY

The diversity metrics that we just learned about attempt to integrate richness and evenness into a single, univariate metric. Although useful, information is invariably lost in this process. If we go back to the rank-abundance curve, we can retrieve additional information – and in some cases – make inferences about the processes influencing the structure of an ecological system.

Species abundance models

The RAC is a simple data structure that is both a vector of abundances. It is also a row in the site-by-species matrix (minus the zeros, i.e., absences).

Predicting the form of the RAC is the first test that any biodiversity theory must pass and there are no less than 20 models that have attempted to explain the uneven form of the RAC across ecological systems.

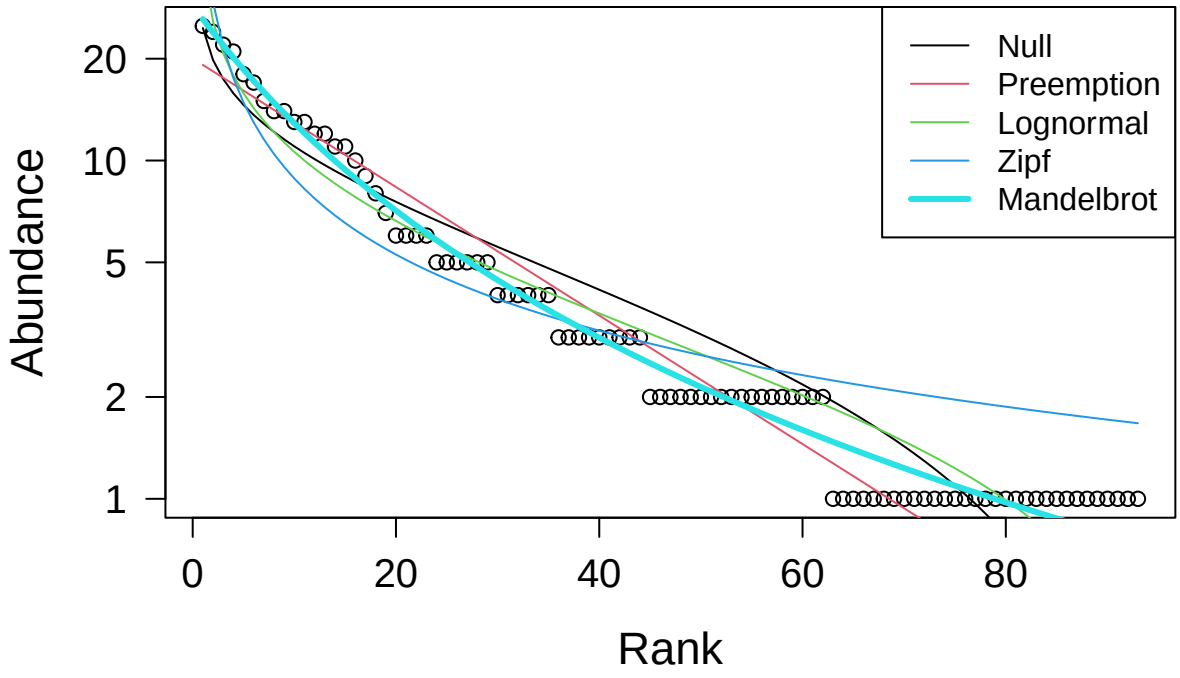
In the R code chunk below, please do the following:

1. Use the `radfit()` function in the `vegan` package to fit the predictions of various species abundance models to the RAC of `site1` in BCI,
2. Display the results of the `radfit()` function, and
3. Plot the results of the `radfit()` function using the code provided in the handout.

```
RACresults <- radfit(site1)
# Viewing the results
RACresults
```

```
##
## RAD models, family poisson
## No. of species 93, total abundance 448
##
##             par1      par2     par3    Deviance AIC      BIC
## Null                                    39.5261 315.4362 315.4362
## Preemption  0.042797                    21.8939 299.8041 302.3367
## Lognormal   1.0687     1.0186           25.1528 305.0629 310.1281
## Zipf        0.11033   -0.74705          61.0465 340.9567 346.0219
## Mandelbrot  100.52    -2.312    24.084   4.2271 286.1372 293.7350
```

```
# Plot results of function
plot.new()
plot(RACresults, las = 1, cex.lab = 1.4, cex.axis = 1.25)
```



Question 9: Answer the following questions about the rank abundance curves: a) Based on the output of `radfit()` and plotting above, discuss which model best fits our rank-abundance curve for `site1`? b) Can we make any inferences about the forces, processes, and/or mechanisms influencing the structure of our system, e.g., an ecological community?

> ***Answer 9a***: I would say that the Mandelbrot model best fits the rank-abundance curve for site1 because it has the lowest AIC value (286.13) which means that our data has the lowest amount of penalty scores associated with the model. ***Answer 9b***: I think based on the plot and 'radfit()' output that we can assume that null processes are not taking place within the ecological community because the null model does not best fit the data according to the AIC and BIC values.

Question 10: Answer the following questions about the preemption model: a. What does the preemption model assume about the relationship between total abundance (N) and total resources that can be preempted? b. Why does the niche preemption model look like a straight line in the RAD plot?

> ***Answer 10a***: It assumes that its a linear relationship. ***Answer 10b***: The niche preemption model is a straight line because it assumes a linear relationship between species rank and abundance.

Question 11: Why is it important to account for the number of parameters a model uses when judging how well it explains a given set of data?

> ***Answer 11***: It is important to account for the parameters of a model because the more parameters a model has, the better the model will fit the data.

SYNTHESIS

1. As stated by Magurran (2004) the $D = \sum p_i^2$ derivation of Simpson's Diversity only applies to communities of infinite size. For anything but an infinitely large community, Simpson's Diversity index is calculated as $D = \sum \frac{n_i(n_i-1)}{N(N-1)}$. Assuming a finite community, calculate Simpson's D, 1 - D, and Simpson's inverse (i.e. 1/D) for `site 1` of the BCI site-by-species matrix.

```
SimpD1 <- SimpD(site1) # 0.0253
SimpD1
```

```
## [1] 0.0253707
```

```
SimpD2 <- 1 - SimpD(site1) # 0.9746
SimpD2
```

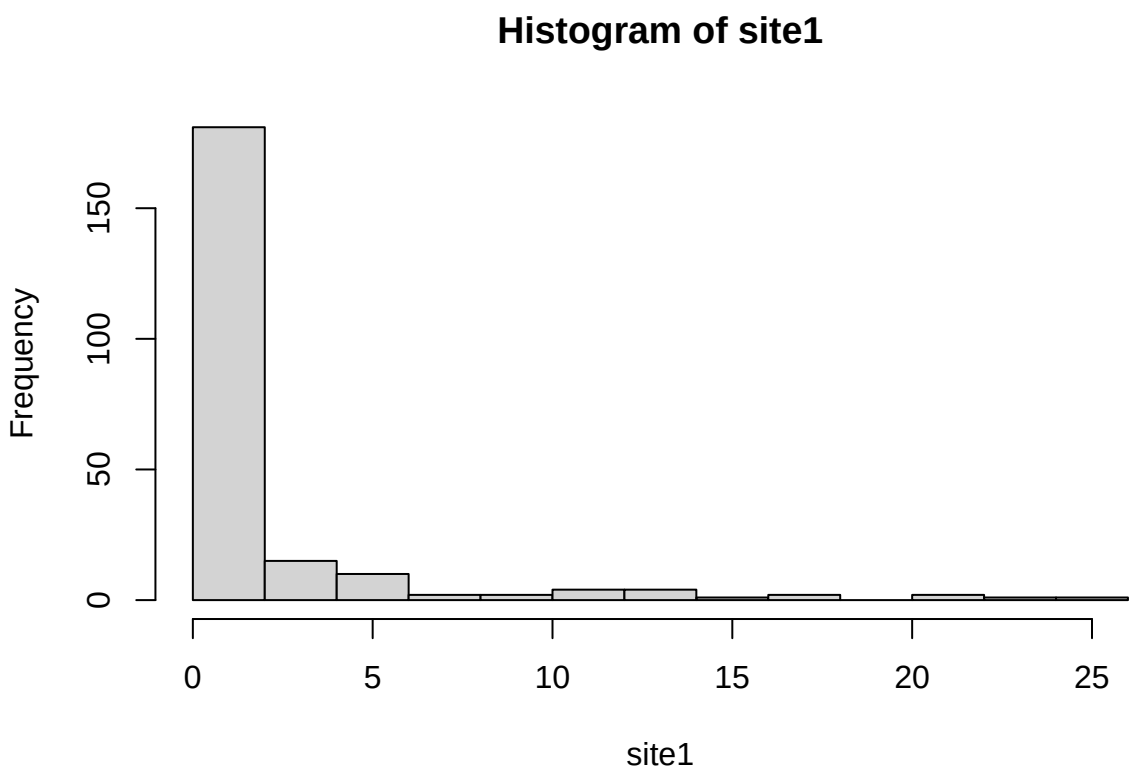
```
## [1] 0.9746293
```

```
SimpD3 <- 1/SimpD(site1) # 39.42
SimpD3
```

```
## [1] 39.41555
```

2. Along with the rank-abundance curve (RAC), another way to visualize the distribution of abundance among species is with a histogram (a.k.a., frequency distribution) that shows the frequency of different abundance classes. For example, in a given sample, there may be 10 species represented by a single individual, 8 species with two individuals, 4 species with three individuals, and so on. In fact, the rank-abundance curve and the frequency distribution are the two most common ways to visualize the species-abundance distribution (SAD) and to test species abundance models and biodiversity theories. To address this homework question, use the R function **hist()** to plot the frequency distribution for `site 1` of the BCI site-by-species matrix, and describe the general pattern you see.

```
site1 <- as.data.frame(site1)
site1 <- t(site1)
hist(site1)
```



Answer: Based on the histogram for site1 of the BCI data, we can see a long-tailed distribution such that there are a lot of species found only a few times within the site. This allows us to infer that there are rare species in site 1.

3. We asked you to find a biodiversity dataset with your partner. This data could be one of your own or it could be something that you obtained from the literature. Load that dataset. How many sites are there? > Answer: We have a total of 58 sites

How many species are there in the entire site-by-species matrix? > Answer: 34059 species

Any other interesting observations based on what you learned this week? > Answer: Based on the rank abundance curve shown below for the pond data that we are using for our project, we can see that our data follows the same distribution as the ecological pattern described in the handout.

```
# Making a RAC with pond data

#ponds.rac <- as.numeric(RAC(Pond97[1, ]))
#length(ponds.rac)
#max(ponds.rac)
#min(ponds.rac)
#plot.new()
#pond.ranks <- as.vector(seq(1, length(ponds.rac)))
#opar <- par(no.readonly = TRUE)
#par(mar = c(5.1, 5.1, 4.1, 2.1))
#plot(pond.ranks, log(ponds.rac), type = "p", axes =F,
#     xlab = "Rank in abundance", ylab = "Abundance",
#     las = 1, cex.lab = 1.4, cex.axis = 1.25);

#box()
#axis(side = 1, labels = T, cex.axis = 1.25)
#axis(side = 2, las = 1, cex.axis = 1.25,
 #    labels = c(1, 10, 100, 1000, 10000), at = log(c(1, 10, 100, 1000, 10000)))
```

"‘ ## SUBMITTING YOUR ASSIGNMENT Use Knitr to create a PDF of your completed 5.AlphaDiversity_Worksheet.Rmd document, push it to GitHub, and create a pull request. Please make sure your updated repo include both the pdf and RMarkdown files.

Unless otherwise noted, this assignment is due on **Wednesday, January 25th, 2023 at 12:00 PM (noon)**.