5. Worksheet: Alpha Diversity

Richard Hull; Z620: Quantitative Biodiversity, Indiana University

13 April, 2021

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).

```
# Clear R environment
rm(list=ls())
# Print working directory
getwd()
```

```
## [1] "C:/Users/Rich Hull/GitHub/QB2021_Hull/2.Worksheets/5.AlphaDiversity"
```

```r
# Set working directory to alpha diversity
setwd("C:/Users/Rich Hull/GitHub/QB2021_Hull/2.Worksheets/5.AlphaDiversity")
```

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).

```r
# Install vegan
#install.packages("vegan")
require("vegan")
```

```
## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-7
```

```r
# Load data
data(BCI)
# Structure data
str(BCI, max.level = 0)
```

```
## 'data.frame':    50 obs. of  225 variables:
##  - 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`.

```r
# Write function
S.obs <- function(x = ""){
  rowSums(x > 0) * 1
}
# Run function
specrichs <- data.frame(S.obs(BCI))
```

```r
# Run function on first site
site1 <- BCI[1, ]
firstsitesr <- S.obs(site1)
# Run vegan function
sperichv <- data.frame(specnumber(BCI))
# Run vegan function on first site
firstsitesrv <- specnumber(site1)
```

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 species richness for the first four sites are respectively 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.

```r
# Write function
C <- function(x = ""){
  1 - (rowSums(x ==1) / rowSums(x))
}
# Calculate Good's Coverage
goodcov <- data.frame(C(BCI))
```

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***:
> 0-1 ***Answer 2b***:
> All species would be singletons ***Answer 2c***:
> 0.07 ***Answer 2d***:
> Overall, there was good coverage as the site with the smallest proportion of singletons possessed only a proportion of about 0.13, while most sites had a proportion of less than 0.1 singletons.

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

```
# Load microbial dataset
# Load data
soilbac <- read.table("data/soilbac.txt", sep = "\t", header = TRUE)
# Transform and transpose
soilbac.t <- as.data.frame(t(soilbac))
# Index
soilbac1 <- soilbac.t[1,]
# Calculate observed richness
print(S.obs(soilbac1))
```

```
##   OTU
## 13310
```

```
# Calculate coverage
print(C(soilbac1))
```

```
## OTU
##   1
```

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:
13310 ***Answer 3b***: 13310 ***Answer 3c***:
The coverage in site one of the BCI sample was lower than that of the first sample of the KBS data ($0.93 < 1$).

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`.

```r
# Write function Chao1
S.chao1 <- function(x = ""){
  S.obs(x) + (sum(x == 1)^2) / (2 * sum(x == 2))
}
# Write function Chao2
S.chao2 <- function(site = "", SbyS = ""){
  SbyS = as.data.frame(SbyS)
  x = SbyS[site, ]
  SbyS.pa <- (SbyS > 0) * 1
  Q1 = sum(colSums(SbyS.pa) == 1)
  Q2 = sum(colSums(SbyS.pa) == 2)
  S.chao2 = S.obs(x) + (Q1^2)/(2 * Q2)
  return(S.chao2)
}
# Write function ACE
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 == i])
  }
  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)
}
# Calculate functions at site 1 (do not use chao2, used for multiple sites)
print(S.chao1(site1))
```

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

```r
print(S.ace(site1))
```

```
## [1] 159.3404
```

```r
# Calculate functions at soilbac1
print(S.chao1(soilbac1))
```

```
##     OTU
## 13310.5
```

```r
print(S.ace(soilbac1))
```

```
## [1] 13311.34
```

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***: In site1 Chao provides a much smaller estimator than ACE, but in soilbac1 Chao provides only a slightly smaller estimator than ACE. In soilbac1 I think it is OK to use ACE because there are many species with greater than 10 individuals, while in site1 I would use Chao because most species have less than 10 individuals.

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.

This code would not generate a plot even after troubleshooting and updating R, so I had to exclude it

##4) SPECIES EVENNESS 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

```
# Write RAC function
RAC <- function(x = ""){
  x = as.vector(x)
  x.ab = x[x > 0]
  x.ab.ranked = x.ab[order(x.ab, decreasing = TRUE)]
  return(x.ab.ranked)
}
# Conduct RAC on site1
racsite1 <- RAC(site1)
```

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

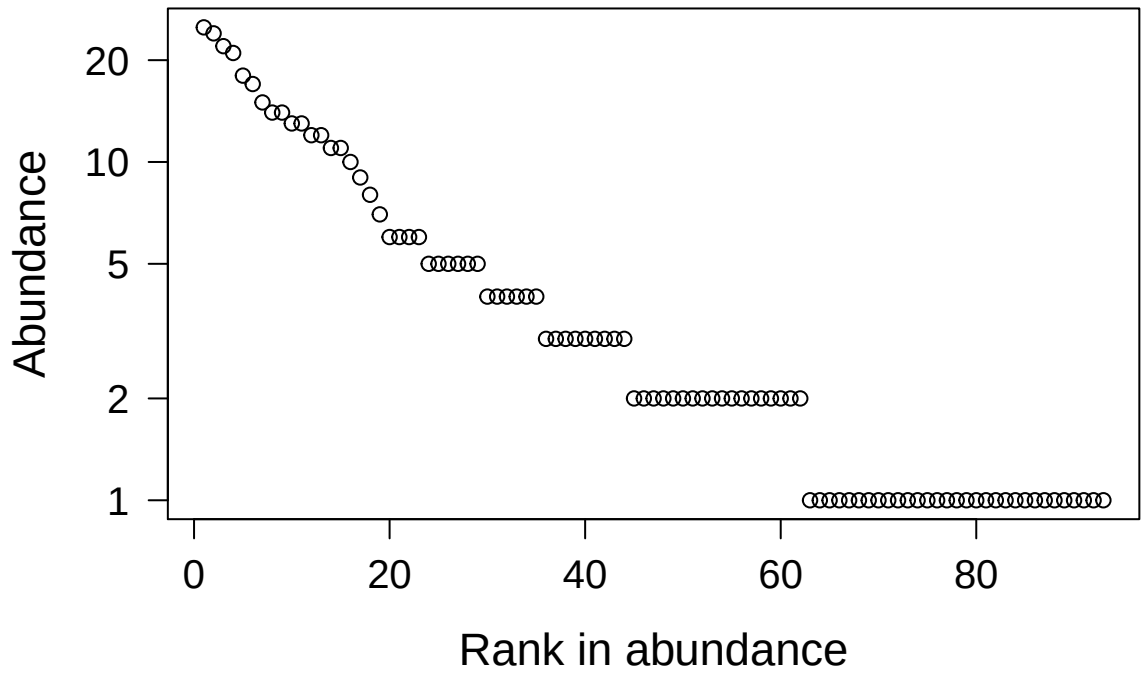
In the R code chunk below, do the following:

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
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 = "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)))
```



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: # This clearly demonstrates that abundance among species is unequally distributed

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`.

```
# Simpson's evenness function
SimpE <- function(x = ""){
  S <- S.obs(x)
  x = as.data.frame(x)
  D <- diversity(x, "inv")
  E <- (D)/S
  return(E)
}
# Calculate Simpson's evenness for site1
site1 <- BCI[1, ]
print(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}.

```
# Function for Smith and Wilson's evenness
Evar <- function(x){
  x <- as.vector(x[x > 0])
  1 - (2/pi)*atan(var(log(x)))
}
# Calculate evenness of site1
print(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***: # No, the Simpson's evenness value of 0.42 is smaller than the Smith and Wilson's evenness value of 0.51. This difference is due to the fact that Smith and Wilson's evenness accounts for the bias of the most abundant species. Lastly, an evenness of 0.51 is indicative of an intermediate level of evenness.

##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".

```
# Define function for shannon's diversity
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)
}
# Calculate shannon's diversity for bci site1
print(ShanH(site1))
```

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

```
# Caculate shannon's diversity using vegan
print(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".

```
# Define function for calculating simpson's diversity
SimpD <- function(x = ""){
  D = 0
  N = sum(x)
  for (n_i in x){
    D = D + (n_i^2)/(N^2)
  }
```

```
  return(D)
}
# Calculate 1/D
D.inv <- 1/SimpD(site1)
print(D.inv)
```

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

```
# Calculate 1-D
D.sub <- 1 - SimpD(site1)
print(D.sub)
```

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

```
# Compare with 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 $\boldsymbol{\alpha}$,
2. Calculate Fisher's $\boldsymbol{\alpha}$ for `site1` of BCI.

```
# Fisher code
Fisher <- fisher.alpha(rac)
# Calculate Fisher's alpha for site1
Fisher
```

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

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

> ***Answer 7***: # It is a smaller number (35.7) than the inverse of Simpson's diversity. It takes into account sampling error, which Simpson's diversity does not account for.

##6) 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

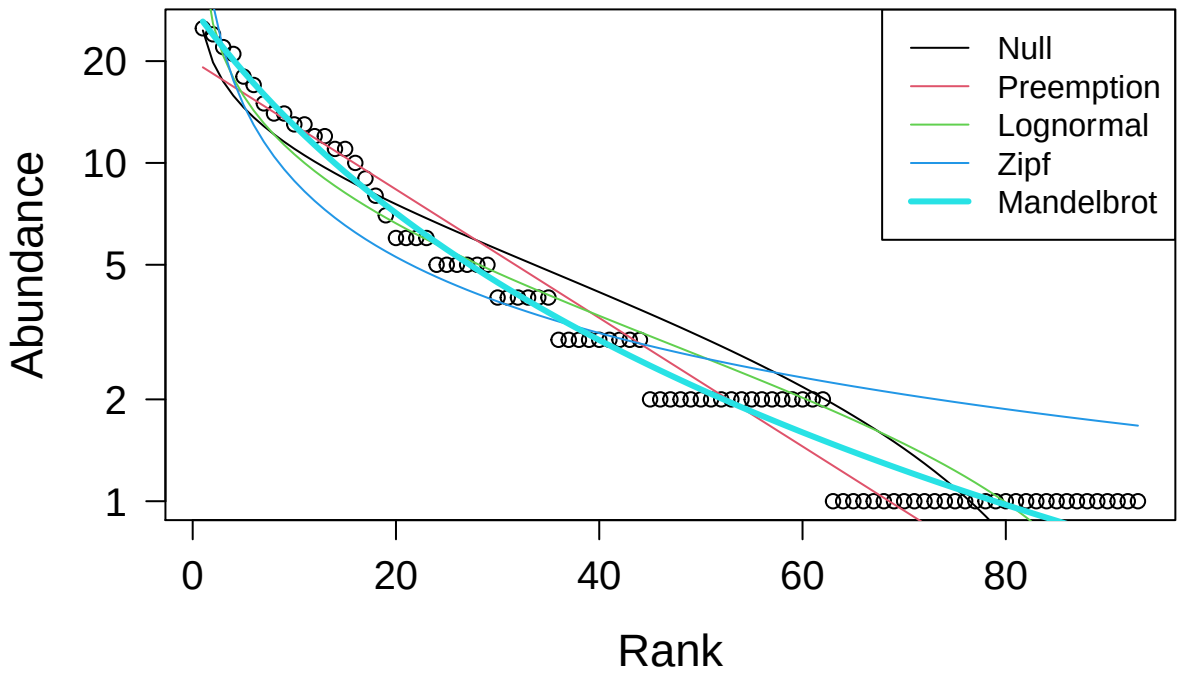
The RAC is a simple data structure that is both a vector of abundances. It is also a row in the site-by-species matrix (minus the zeros, i.e., absences).

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.

```
# Fit predictions
RACresults <- radfit(site1)
# Display and visualize results
plot.new()
plot(RACresults, las = 1, cex.lab = 1.4, cex.axis = 1.25)
```



Question 8: 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 8a:
The Mandelbrot model best fits the data.

Answer 8b: # This seems to represent what we would typically see in an ecological community (think rank abundance curve)

Question 9: 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:
First, resources are limited; second, N and r are related to each other exponentially.

Answer 10b: The preemption model is described via a linear equation.

Question 10: 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 difficult to obtain more parameters, so it is important to quantify how well a model does with minimal information.

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.

```
# Define function for calculating simpson's diversity for noninfinite community
SimpDnoninf <- function(x = ""){
  D = 0
  N = sum(x)
  for (n_i in x){
    D = D + (n_i*(n_i - 1))/(N*(N - 1))
  }
  return(D)
}
# Calculate Simpson's D for site1
site1 <- BCI[1, ]
print(SimpDnoninf(site1))
```

```
## [1] 0.02319032
```

```
# Define and calculate 1 - D
OneminusSimpD <- (1 - SimpDnoninf(site1))
print(OneminusSimpD)
```

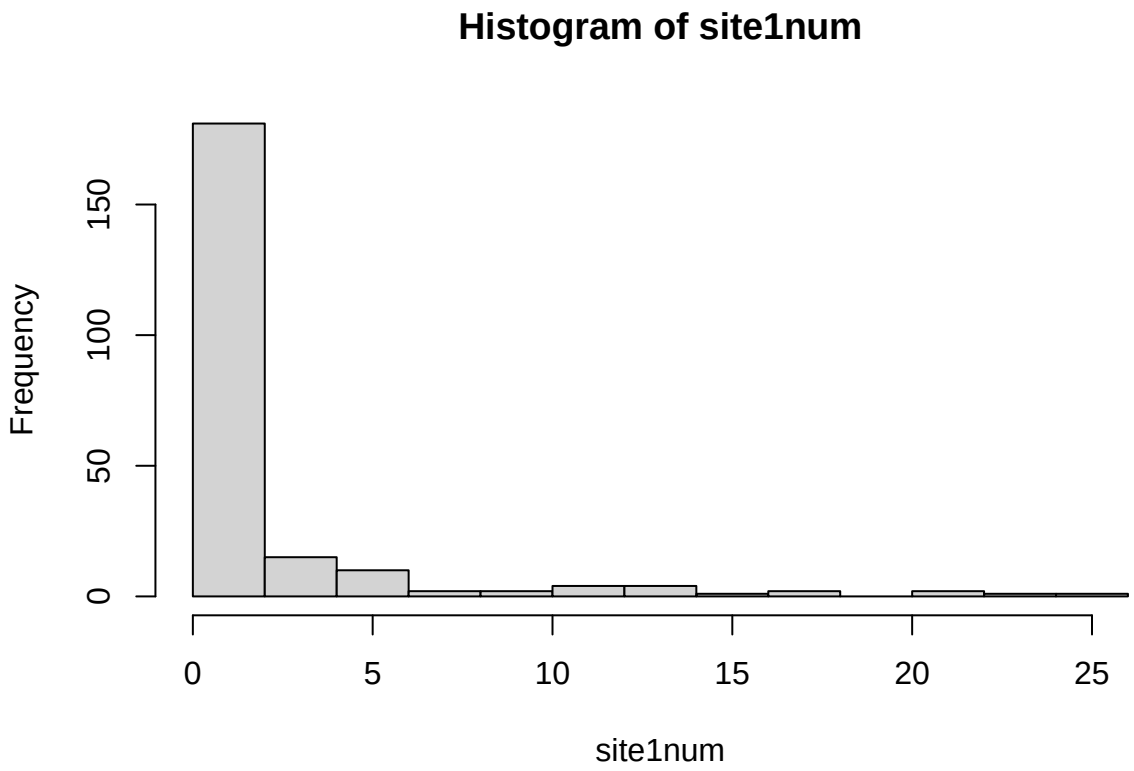
```
## [1] 0.9768097
```

```r
# Define and calculate inverse D
OneminusSimpD <- (1 / SimpDnoninf(site1))
print(OneminusSimpD)
```

```
## [1] 43.12145
```

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.

```r
# Plot histogram of site 1
site1num <- as.numeric(site1)
hist(site1num)
```




```r
# There is a very steep rank abundance curve, with the most dominant species being much more common tha
```

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? How many species are there in the entire site-by-species matrix? Any other interesting observations based on what you learned this week?

```r
# load data
library(readr)
mydata <- read_csv(file = "DeamDataInitial.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   id = col_double(),
##   collectionCode = col_logical(),
##   ownerInstitutionCode = col_logical(),
##   otherCatalogNumbers = col_double(),
##   taxonID = col_double(),
##   subgenus = col_logical(),
##   identificationReferences = col_logical(),
##   identificationRemarks = col_logical(),
##   taxonRemarks = col_logical(),
##   identificationQualifier = col_logical(),
##   year = col_double(),
##   month = col_double(),
##   day = col_double(),
##   startDayOfYear = col_double(),
##   endDayOfYear = col_double(),
##   fieldNumber = col_logical(),
##   informationWithheld = col_logical(),
##   dataGeneralizations = col_logical(),
##   dynamicProperties = col_logical(),
##   establishmentMeans = col_logical()
##   # ... with 19 more columns
## )
## i Use `spec()` for the full column specifications.
```

```
## Warning: 329 parsing failures.
##  row                       col           expected                    actual                  file
## 1132 verbatimCoordinates      1/0/T/F/TRUE/FALSE ,                          'DeamDataInitial.csv'
## 1574 identificationRemarks    1/0/T/F/TRUE/FALSE Nomenclatural adjustment 'DeamDataInitial.csv'
## 1579 identificationReferences 1/0/T/F/TRUE/FALSE FNA                      'DeamDataInitial.csv'
## 1741 identificationReferences 1/0/T/F/TRUE/FALSE FNA                      'DeamDataInitial.csv'
## 1887 identificationReferences 1/0/T/F/TRUE/FALSE FNA                      'DeamDataInitial.csv'
## .... ......................... .................. ......................... ....................
## See problems(...) for more details.
```

```r
# We are counting each Indiana as a site, so here I count the number of counties and provide the number
countycount <- setNames(as.data.frame(table(mydata$county)), c("county", "n"))
print(length(countycount$county))
```

```
## [1] 102
```

```r
# This returns 102 sites (counties), but we know there are only 92 counties in the state. These data er
# We transformed this data into a site by species matrix not provided here and also corrected species n
speciescount <- setNames(as.data.frame(table(mydata$scientificName)), c("scientificName", "n"))
print(length(speciescount$scientificName))
```

```
## [1] 2366
```

```
# Here we get that there are 2366 species, which is slightly off the corrected number due to type speci
```

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, April 7th, 2021 at 12:00 PM (noon)**.