5. Worksheet: Alpha Diversity

Herbert Sizek; Z620: Quantitative Biodiversity, Indiana University

09 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).

```
# install.packages('vegan')
require('vegan')
```

```
## Loading required package: vegan
```

```
## Warning: package 'vegan' was built under R version 3.6.3
```

```
## Loading required package: permute
```

```
## Warning: package 'permute' was built under R version 3.6.3
```

```
## Loading required package: lattice
```

```
## This is vegan 2.5-7
```

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

```
## [1] "D:/GitHub/QB2021_Sizek/2.Worksheets/5.AlphaDiversity"
```

```
setwd("D:/GitHub/QB2021_Sizek/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).

```
require('vegan')
data(BCI)
BCI
```

Abarema.macradenia <int>	Vachellia.melanoceras <int>	Acalypha.diversifolia <int>	Acalypha.macrost
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	1	0	0

1-10 of 50 rows | 1-5 of 226 columns Previous **1** 2 3 4 5 Next

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`.

```
S.obs <- function(x=''){
  rowSums(x>0)*1
  }
S.obs(BCI[1,])
```

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

```
specnumber(BCI[1,])
```

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

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 they both return the same value.

```
S.obs(BCI[1:4,])
```

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

```
S.goodCover<- function(x=''){
  1- (rowSums(x==1)/rowSums(x))
}
BCI['C'] <- S.goodCover(BCI)
```

Question 2: Answer the following questions about coverage:

a. What is the range of values that can be generated by Good's Coverage? > ***Answer 2a***: 0 to 1

b. What would we conclude from Good's Coverage if n_i equaled *N*? > ***Answer 2b***: n is the nuber of species with one individual, while N is the number of individuals in total, so if $n = N$ then all species in the dataset only occur once or zero times.

c. What portion of taxa in `site1` was represented by singletons? > ***Answer 2c***: 13.7%

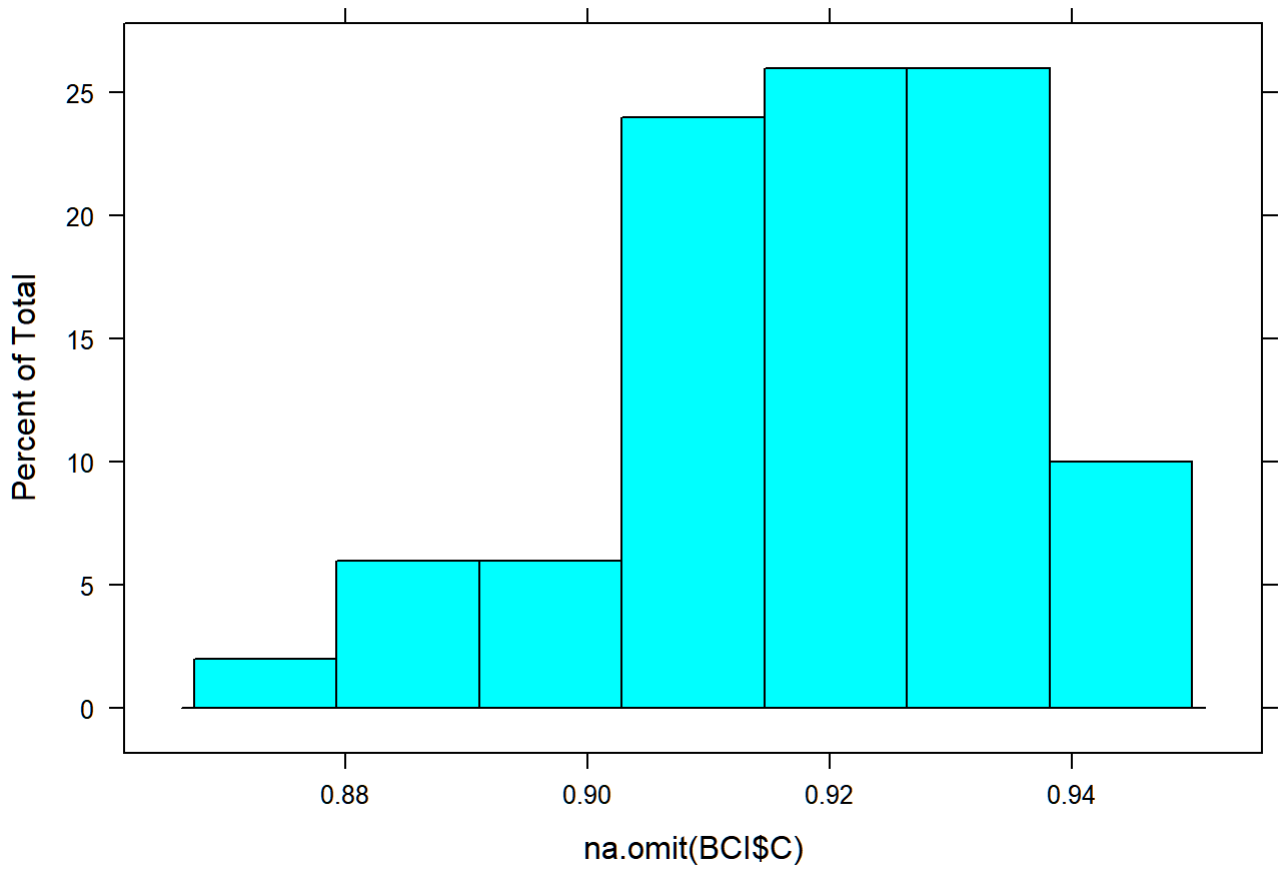
```
S.one <- function(x=''){
  rowSums(x==1)/dim(x)[2]
}
S.one(BCI[1,])
```

```
##         1
## 0.1371681
```

d. Make some observations about coverage at the BCI plots.

> ***Answer 2d***: Most sites have only a few ones in it in comparison to the total number of species. in the dataset.

```
histogram(na.omit(BCI$C))
```



Estimated richness

In the R code chunk below, do the following:

1. Load the microbial dataset (located in the `5.AlphaDiversity/data` folder),

```
soilbac <- read.table("./data/soilbac.txt",sep="\t",header=TRUE,row.names =1)
```

2. Transform and transpose the data as needed (see handout),

```
soilbac.t <- as.data.frame(t(soilbac))
```

3. Create a new vector (`soilbac1`) by indexing the bacterial OTU abundances of any site in the dataset,

```
soilbac1 <- soilbac.t[1,]
```

4. Calculate the observed richness at that particular site, and

```
S.obs(soilbac1)
```

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

5. Calculate coverage of that site

```
S.goodCover(soilbac1)
```

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

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*? > ***Answer 3a***:

```
sum(soilbac1)
```

```
## [1] 2119
```

b. What is the observed richness of `soilbac1` ?

> ***Answer 3b***:

```
S.obs(soilbac1)
```

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

c. How does coverage compare between the BCI sample (`site1`) and the KBS sample (`soilbac1`)?

> ***Answer 3c***: The coverages is less in the KBS sample, meaning that there are more samples that have only one read.

Richness estimators

In the R code chunk below, do the following:

1. Write a function to calculate **Chao1**,

```
S.chao1 <-function(x=''){
  S.obs(x) + (sum(x==1)^2)/(2*sum(x--2))
}
```

2. Write a function to calculate **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)
}
```

3. Write a function to calculate **ACE**, and

```
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)

}
```

4. Use these functions to estimate richness at `site1` and `soilbac1` .

```
site1 <- BCI[1,]
print(paste("site1 Chao1:",S.chao1(site1)))
```

```
## [1] "site1 Chao1: 94.5333372974875"
```

```
print(paste("site1 Chao2:",S.chao2(1,BCI)))
```

```
## [1] "site1 Chao2: 105.605263157895"
```

```
print(paste("site1 ACE:",S.ace(site1)))
```

```
## [1] "site1 ACE: 160.554300870267"
```

```
print(paste("soilbac1 Chao1:",S.chao1(soilbac1)))
```

```
## [1] "soilbac1 Chao1: 1083.68224364105"
```

```
print(paste("soilbac1 Chao2:",S.chao2(1,soilbac.t)))
```

```
## [1] "soilbac1 Chao2: 21055.3862763916"
```

```
print(paste("soilbac1 ACE:",  S.ace(soilbac1)))
```

```
## [1] "soilbac1 ACE: 4465.9827101141"
```

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***: ACE uses an estimator based off of how rare species are counted in comparison to species where there is one species as a correction for the nonobserved species, similar to Chao2, but with a threshold and only accounting for within site. Chao2 uses the entire site by species list, which means that if there is large variation in sites, it might not be appropriate. Tor example if some of your sites were poor habitat or degraded due to human activity, it would skew the richness estimation of other sites. I would probably choose Chao1 in most circumstances because it is the most concervative measure, but it would really depend on the dataset and the properties of the data.

Rarefaction

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

1. Calculate observed richness for all samples in `soilbac`,

```
soilbac.S <-S.obs(soilbac.t)
```

2. Determine the size of the smallest sample,

```
min.N <- min(rowSums(soilbac.t))
```

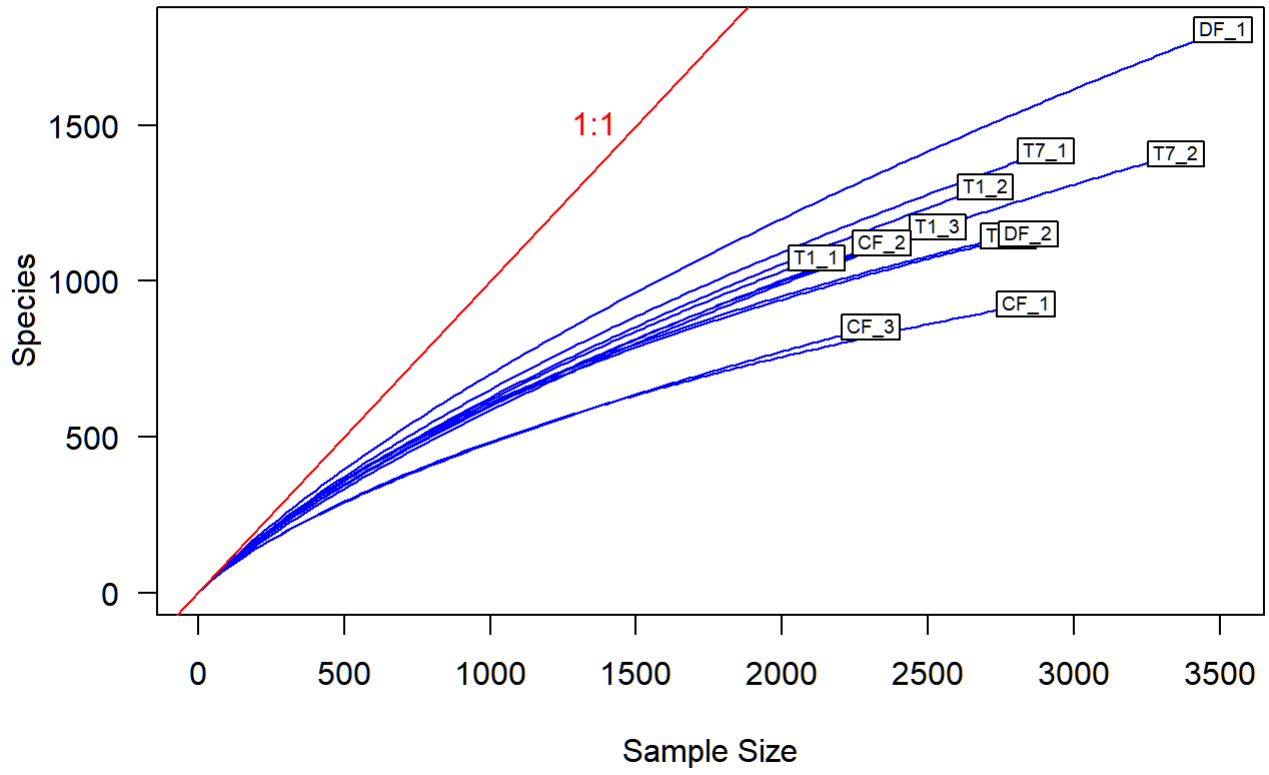
3. Use the `rarefy()` function to rarefy each sample to this level,

```
S.rarefy <- rarefy(x=soilbac.t, sample =min.N,se=TRUE)
```

4. Plot the rarefaction results, and

5. Add the 1:1 line and label.

```
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 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

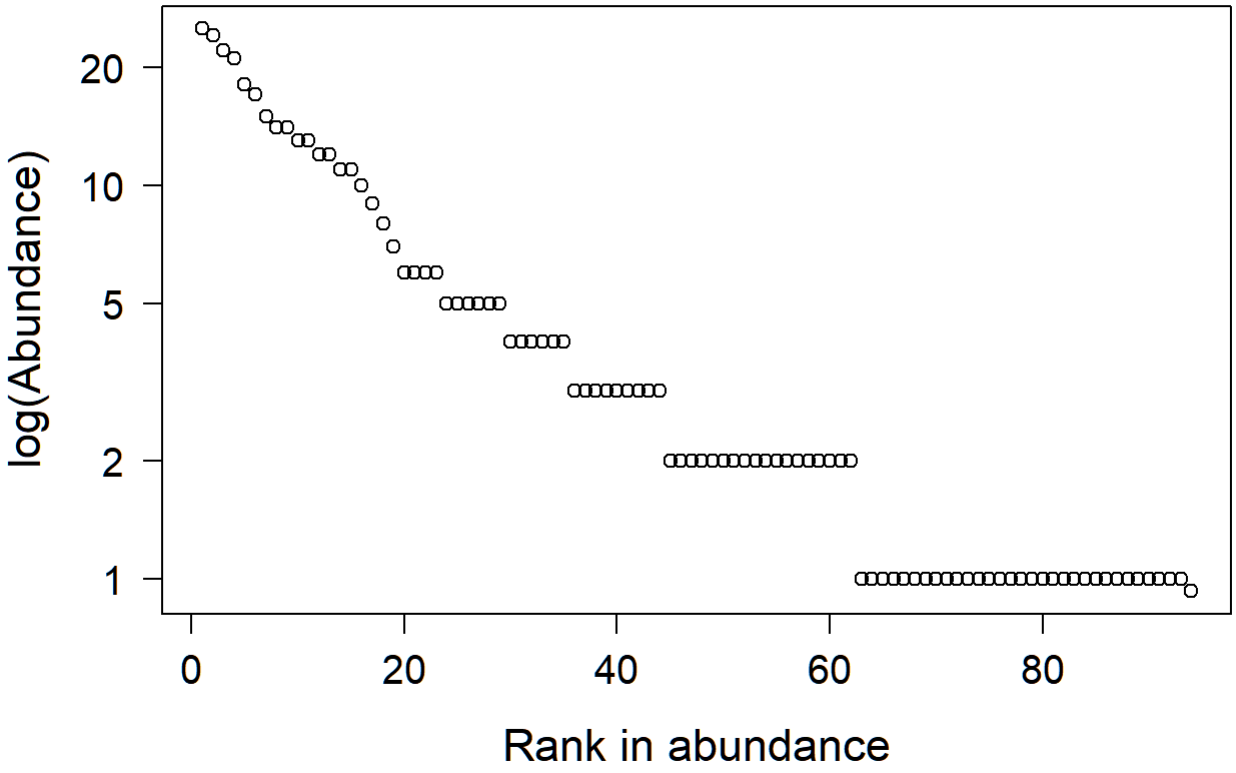
```
RAC <- function(x=''){
  x=as.vector(x)
  x.ab = x[x>0]
  x.ab.ranked = x.ab[order(x.ab,decreasing=TRUE)]
  return(x.ab.ranked)
}
```

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.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=TRUE,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***: Different distributions have different shapes that are easier to analize by sight by putting distributions on log-normal and log-log scales (also sometimes cumulative distributions are use). In terms of eveness, because we don't expect a heavy tailed distribution of species (variance is greater than the mean), it can be more informative to understand the distribuion of occurance species rather than just the frequencies of the species.

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

```
SimpE <- function(x = ''){
  S <-S.obs(x)
  x= as.data.frame(x)
  D<- diversity(x,'inv')
  E <-(D)/S
  return(E)
}
```

2. Calculate $E_{1/D}$ for `site1` .

```
SimpE(site1)
```

```
##        1
## 0.420987
```

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},

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

2. Calculate E_{var} for `site1` , and

```
Evar(site1)
```

```
## [1] 0.5058337
```

3. Compare $E_{1/D}$ and E_{var}.

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***: Yes the agree. Having simson's be smaller than smith and wilson suggests that the distribution is skewed towards the larger species, though a large number of less frequent species were in the dataset.

##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),

```
ShanH <-function(x='',base = exp(1)){
  x <- as.vector(x[x>0])
  x <- x/sum(x)
  return(-sum(x*log(x,base=base)))
}
```

2. Compare this estimate with the output of `vegan` 's diversity function using method = "shannon".

```
ShanH(site1)
```

```
## [1] 4.024962
```

```
diversity(site1,index="shannon")
```

```
## [1] 4.024962
```

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),

```
SimpD <-function(x=""){
  N=sum(x)
  sum((x^2)/(N^2))
}
```

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(site1)
```

```
## [1] 0.0252699
```

```
1/SimpD(site1)
```

```
## [1] 39.57278
```

```
1-SimpD(site1)
```

```
## [1] 0.9747301
```

```
diversity(site1,"inv")
```

```
## [1] 39.57278
```

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

```
## [1] 0.9747301
```

Fisher's α

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

1. Provide the code for calculating Fisher's α,

```
cFisher <-function(x=''){
  fisher.alpha(as.integer(as.vector(x[x>0])))
}
```

2. Calculate Fisher's α for `site1` of BCI.

```
cFisher(site1)
```

```
## [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***: It accounts for fluxtuations in the sampling of the data, not just the observations.

##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.

```
RACresults <-radfit(site1)
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

```
## Warning in nlm(canfun, p = p, x = x, rnk = rnk, logJ = logJ, wt = wt, hessian =
## TRUE, : NA/Inf replaced by maximum positive value
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

```
## Warning in nlm(canfun, p = p, x = x, rnk = rnk, logJ = logJ, wt = wt, hessian =
## TRUE, : NA/Inf replaced by maximum positive value
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

```
## Warning in nlm(canfun, p = p, x = x, rnk = rnk, logJ = logJ, wt = wt, hessian =
## TRUE, : NA/Inf replaced by maximum positive value
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

```
## Warning in nlm(canfun, p = p, x = x, rnk = rnk, logJ = logJ, wt = wt, hessian =
## TRUE, : NA/Inf replaced by maximum positive value
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

```
## Warning in nlm(mandelfun, p = p, x = x, rnk = rnk, off = off, family = fam, :
## NA/Inf replaced by maximum positive value
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

```
## Warning in nlm(mandelfun, p = p, x = x, rnk = rnk, off = off, family = fam, :
## NA/Inf replaced by maximum positive value
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

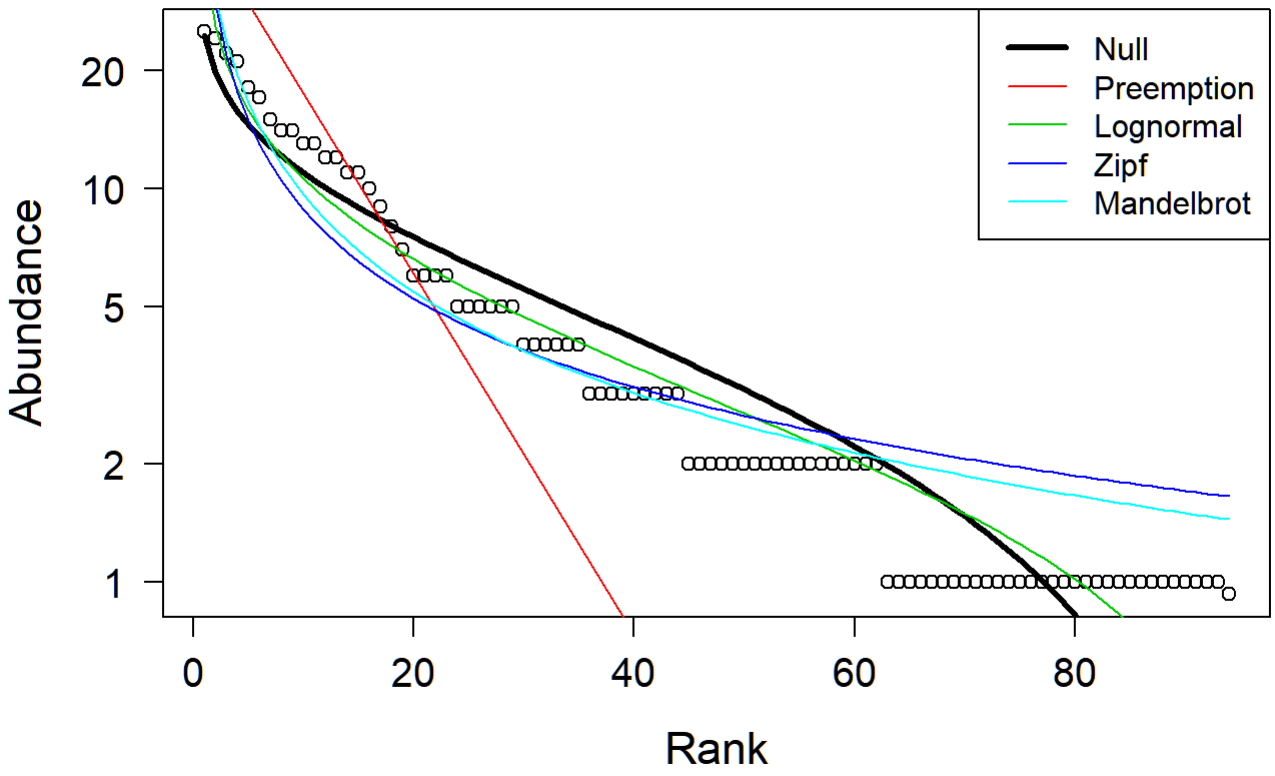
```
## Warning in nlm(mandelfun, p = p, x = x, rnk = rnk, off = off, family = fam, :
## NA/Inf replaced by maximum positive value
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

```
## Warning in nlm(mandelfun, p = p, x = x, rnk = rnk, off = off, family = fam, :
## NA/Inf replaced by maximum positive value
```

```
## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804

## Warning in dpois(y, mu, log = TRUE): non-integer x = 0.930804
```

```
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` ? > ***Answer 8a***: Looks pretty log normal or Null. I would probably want to look at a cumulative distribution to see if it is Zipf or derivative (the tails are too succeptible to noise), but probably we would need more data/analysis to actually make a statement about which model it is outside of Preemption.

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 8b***: Meh. I don't like doing this. These models do imply a bit about some processes, but the amount of generating models there are to get these distributions is so vast that you could argue it many ways. It is more of something that should be used as support for mechanisms/forces, not to suggest mechanisms/forces. We could causually toss that there might possibly be niche partitioning.

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? > ***Answer 10a***: As I said above, nothing. But under the resource partitioning theory, the resource allocation is by dominating species that then compete with each other for resorce share, with the assumption that one player will always be less able to compete for the resource, but still be able to secure parts of the resource. If this happens on a continual basis (and is the only mechanism present) you get a geometric curve.

b. Why does the niche preemption model look like a straight line in the RAD plot?

> ***Answer 10b***: Because we are on a log scale: $a_r = N * \alpha(1 - alpha)^(r - 1)$ $log(a_r) = log(N *(1-alpha)^(r-1))= log(N) + log() + (r-1)log(1-alpha))

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***: More parameters=more likely to get a more presise fit. But when we are dealing with these models that have very sensitive tails, it is probably even more important to do other things like bootstrapping and KS statistics. See Clauset, Shalizi, and Newmand 2009 SIAM.

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.

```
SimpDf <- function(x=''){
  return(sum(x*(x-1)/( sum(x)*( sum(x)-1))))
}
```

```
SimpDf(site1)
```

```
## [1] 0.02309382
```

```
1-SimpDf(site1)
```

```
## [1] 0.9769062
```

```
1/SimpDf(site1)
```

```
## [1] 43.30162
```

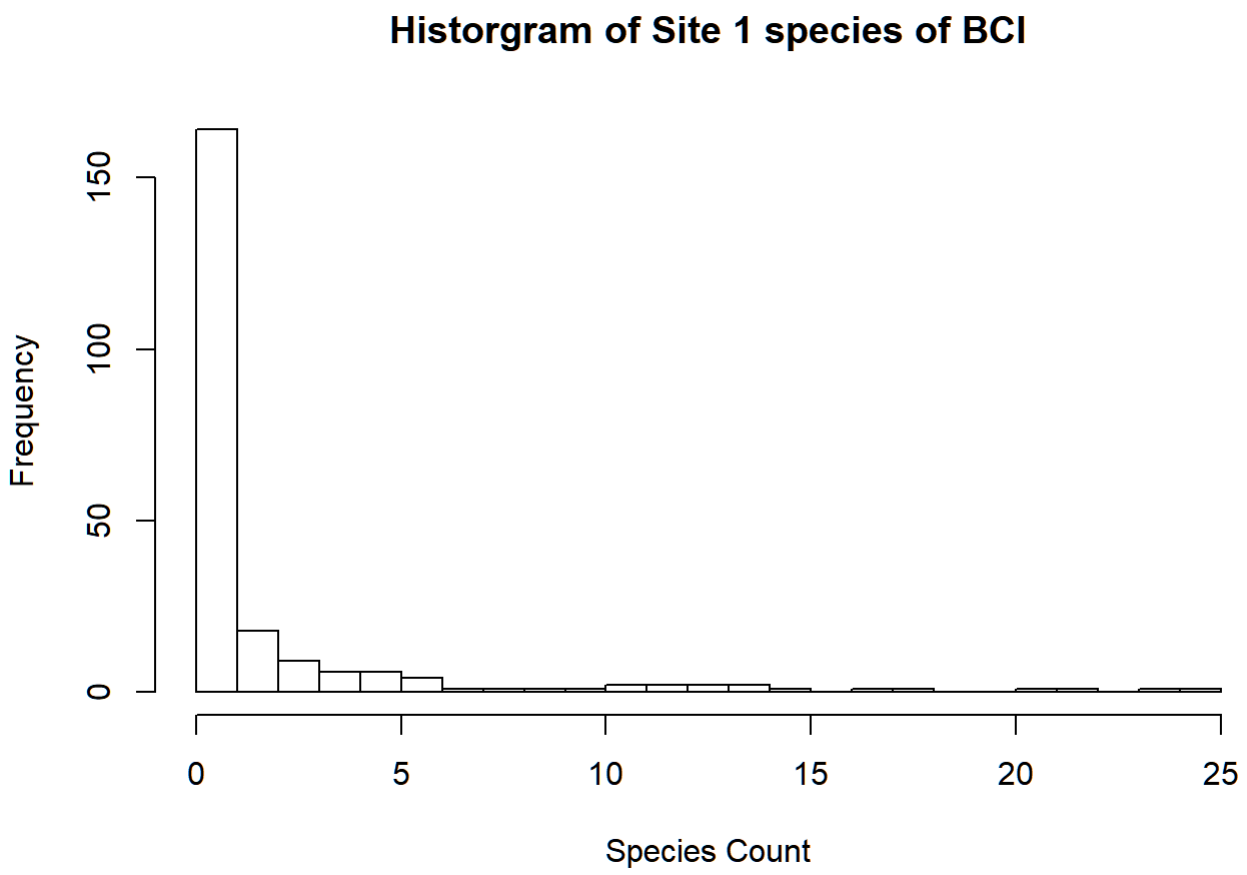
2. Along with the rank-abundance curve (RAC), another way to visualize the distribution of abundance among species is with a histogram (a.k.a., frequency distribution) that shows the frequency of different abundance classes.

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.

```
hist(as.numeric(as.vector(site1)),xlab='Species Count',main='Historgram of Site 1 species of BC
I',breaks=20)
```



Most species only occur once at Site 1, while there are more that occur later on, similar to what is seen in the RAC.

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.

```
require(tidyr)
```

```
## Loading required package: tidyr
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
lfdata <- read.table("data/Deam_Collection_Data_Filtered_3-27-2021.csv",sep=',',header=TRUE)
```

How many sites are there? > First we need to define the site, we'll use the county definition

```
gbcounty <- lfdata %>% count(county, speciesName)
countydata <- as.data.frame(pivot_wider(gbcounty, names_from = county, values_from = n))
countydata.t <- as.data.frame(t(countydata))
print(dim(countydata.t))
```

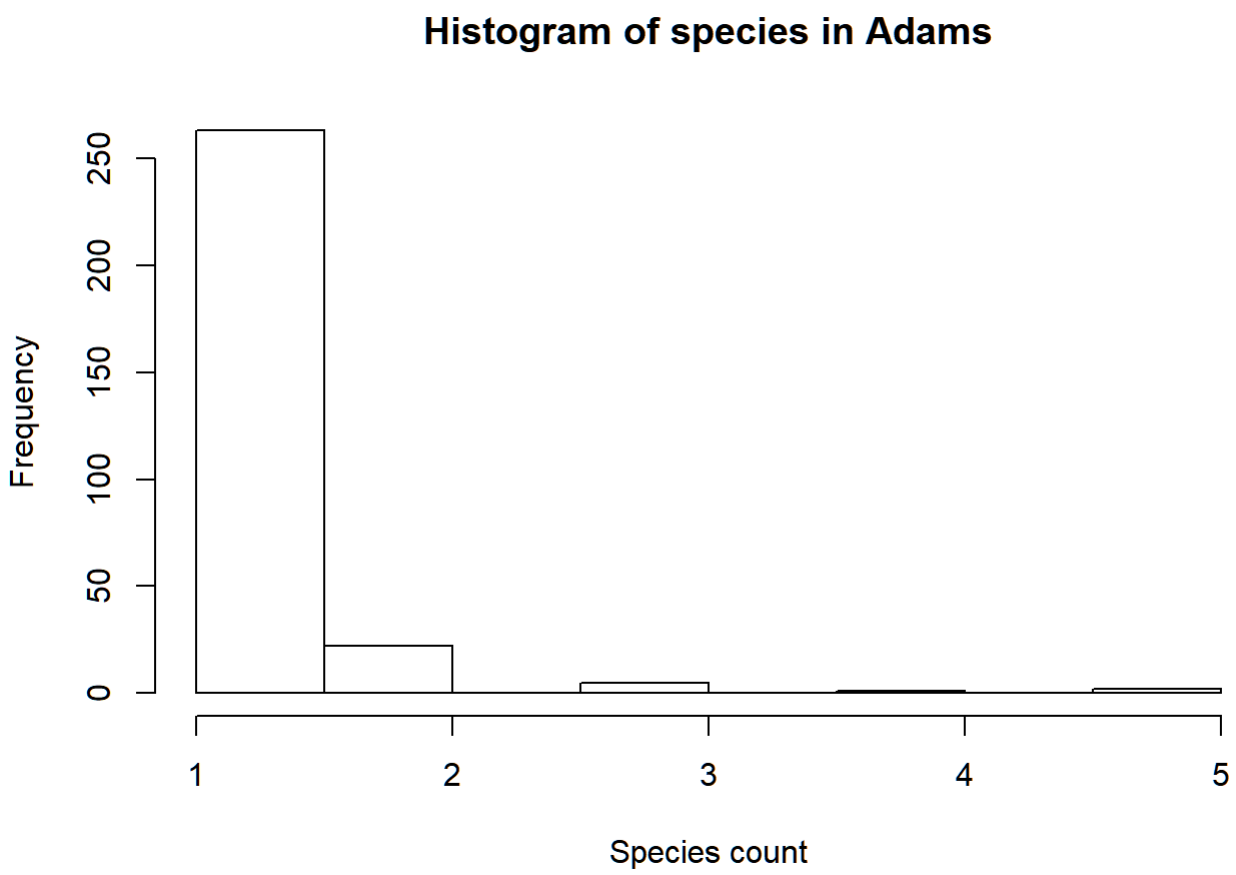
```
## [1]   93 2095
```

92 counties

How many species are there in the entire site-by-species matrix? > 2095

Any other interesting observations based on what you learned this week? > The Chao corrections will not work on this dataset because of the low sampling per county, occurance vs full samplings, and the expressed purpose to sample in that manner.

```
hist(as.numeric(countydata.t[2,]), main='Histogram of species in Adams',xlab='Species count')
```



##

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)**.