5. Worksheet: Alpha Diversity

Pat Wall; Z620: Quantitative Biodiversity, Indiana University

03 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).

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

```
## [1] "/home/patgwall/Classwork/Current/QB/2.Worksheets/5.AlphaDiversity"
```

```r
setwd('~/Classwork/Current/QB/2.Worksheets/5.AlphaDiversity/')
require(vegan)
```

```
## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-7
```

```
require(dplyr)
```

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

```
##
## 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
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(purrr)
```

```
## Loading required package: purrr
```

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, 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`.

```
 s_obs <- function(x){
   rowSums(x > 0) * 1
   }
s_obs(BCI[1:4, ])
```

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

```
vegan::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***: The `vegan` function gets the same results as our function. For site 1 we have 93 species, 84 for site 2, 90 for site 3, and 94 for site 4.

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.

```
g_cov <- function(x) {
  1 - rowSums(x == 1) / rowSums(x)
}
coverages <- g_cov(BCI)
print(coverages)
```

```
##         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***: Good's coverage can produce numbers from 0 to 1 if each species if observed only as a singleton or no singletons are observed respectively.
>
> ***Answer 2b***: This would produce a Good's coverage of zero which would suggest that we've severely undersampled our sites. If each time we observed a new individual it belongs to a new species, we need to keep counting.
>
> ***Answer 2c***: About 7 % of species in site 1 are singletons.
>
> ***Answer 2d***: Coverages are in general good. In the site with the worst coverage 87% of species are represented by more than one individual.

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

```
mat <- read.table('data/soilbac.txt', header = TRUE, row.names = 1)
sb <- tibble::as_tibble(t(mat))
site1 <- sb[1,]
s <- s_obs(site1)
g <- g_cov(site1)
print(paste('Site 1: S = ', s, '; G = ', g, '; N = ', sum(site1)))
```

```
## [1] "Site 1: S =  1074 ; G =  0.64794714487966 ; N =  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***: There was a total of 2119 individuals in this site
>
> ***Answer 3b***: These individuals represented 1074 species
>
> ***Answer 3c***: The civerage in the first site of the BCI sample was much higher (almost 30 percentage points) than in the soil 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`.

```
s_chao1 <- function(df, site) {
   x <- df[site,]
   s_obs(x) + (sum(x == 1)^2) / (2 * sum(x == 2))
}

s_chao2 <- function(df, site) {
   pa <- (df > 0) * 1
   q1 <- sum(colSums(pa) == 1) # observed in one site
   q2 <- sum(colSums(pa) == 1) # observed in two sites
   s_obs(df[site,]) + (q1^2 / (2 * q2))
}

s_ace <- function(df, site, thresh = 10) {
   df <- as.data.frame(df)
   x <- df[site,]
   x <- x[x>0]
   s_abund <- sum(x > thresh) # not rare
   s_rare <- sum(x < thresh)  # rare
   singles <- sum(x == 1)     # singletons
   n_rare <- sum(x[which(x <= thresh)])
   c_ace <- 1 - (singles / s_rare) # coverage
   i <- c(1:thresh)           # counters

   count <- function(i, y){
      length(y[y == i])
   }

   a1 <- sapply(i, count, x) # number of individuals in each class singleton, doubleton etc.?
   f1 <- (i * (i - 1)) * a1 # not sure why we need this

   g_ace <- (s_rare / c_ace) * (sum(f1) / (n_rare * (n_rare - 1)))
   s_abund + (s_rare / c_ace) + (singles / c_ace) * max(g_ace, 0)
}
# first we'll look at the fisrt site in the BCI dataset
sc1 <- s_chao1(BCI, 1)
sc2 <- s_chao2(BCI, 1)
sa <- s_ace(BCI, 1)
print(paste('BCI; Chao1: ', sc1, '; Chao2: ', sc2, '; ACE: ', sa, sep=''))
```

```
## [1] "BCI; Chao1: 119.694444444444; Chao2: 103.5; ACE: 248.738088409542"
```

```
sc1 <- s_chao1(sb, 1)
sc2 <- s_chao2(sb, 1)
sa <- s_ace(sb, 1)
print(paste('SoilBac; Chao1: ', sc1, '; Chao2: ', sc2, '; ACE: ', sa, sep=''))
```

```
## [1] "SoilBac; Chao1: 2628.51396648045; Chao2: 4300.5; ACE: 13273.677251517"
```

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***: Each of these estimators compensates for potential bias in different ways by incorporating different information. They do not give consistent results at all, really. If I had prior information telling me that all of my sites ought to be very similar I would use Chao2, otherwise I would probably use ACE because it incorporates more information than Chao1.

Rarefaction

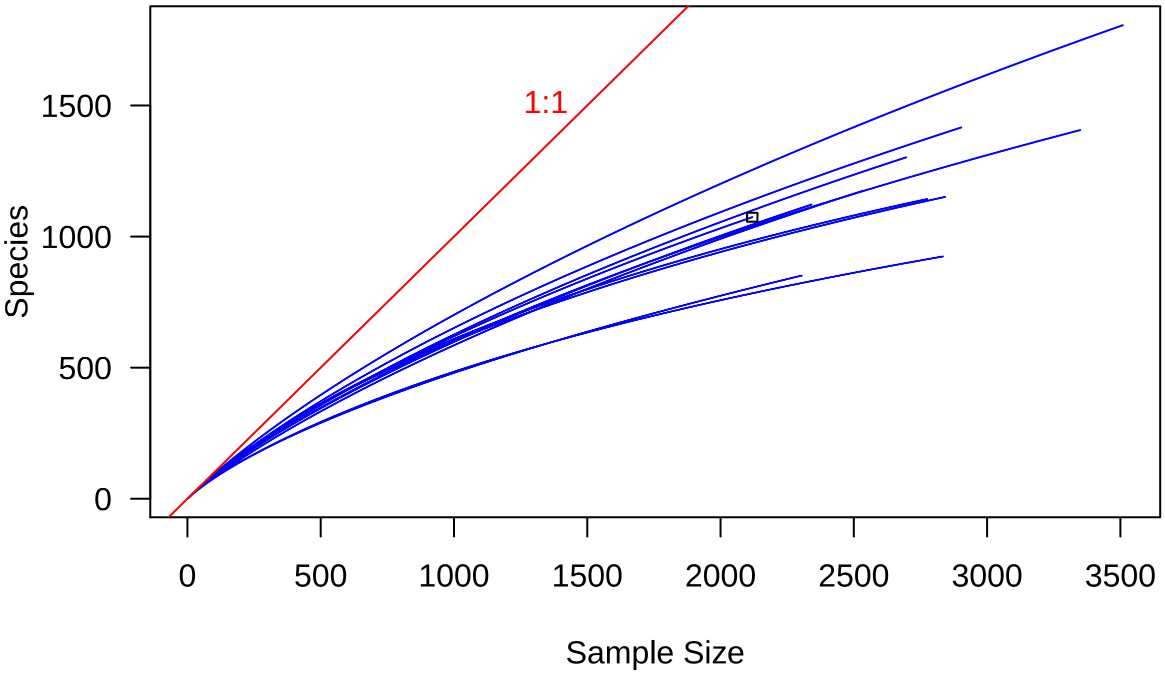
In the R code chunk below, please do the following:

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.

```
s_obs(sb)
```

```
##  [1] 1074 1302 1174 1416 1406 1143 1806 1151  924 1122  851
```

```
smallest_sample <- min(rowSums(sb))
s_rarefy <- vegan::rarefy(sb, smallest_sample, se = TRUE)
vegan::rarecurve(x = sb, step = 20, col = 'blue', cex = 0.6, las = 1, label=TRUE)
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

```
rank_abundance <- function(df, site) {
   x <- df %>% slice(1)
   x_nonzero <- x[x > 0]
   x_ranked <- sort(x_nonzero, decreasing=TRUE)
}
```

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)"

```
require(scales)
```

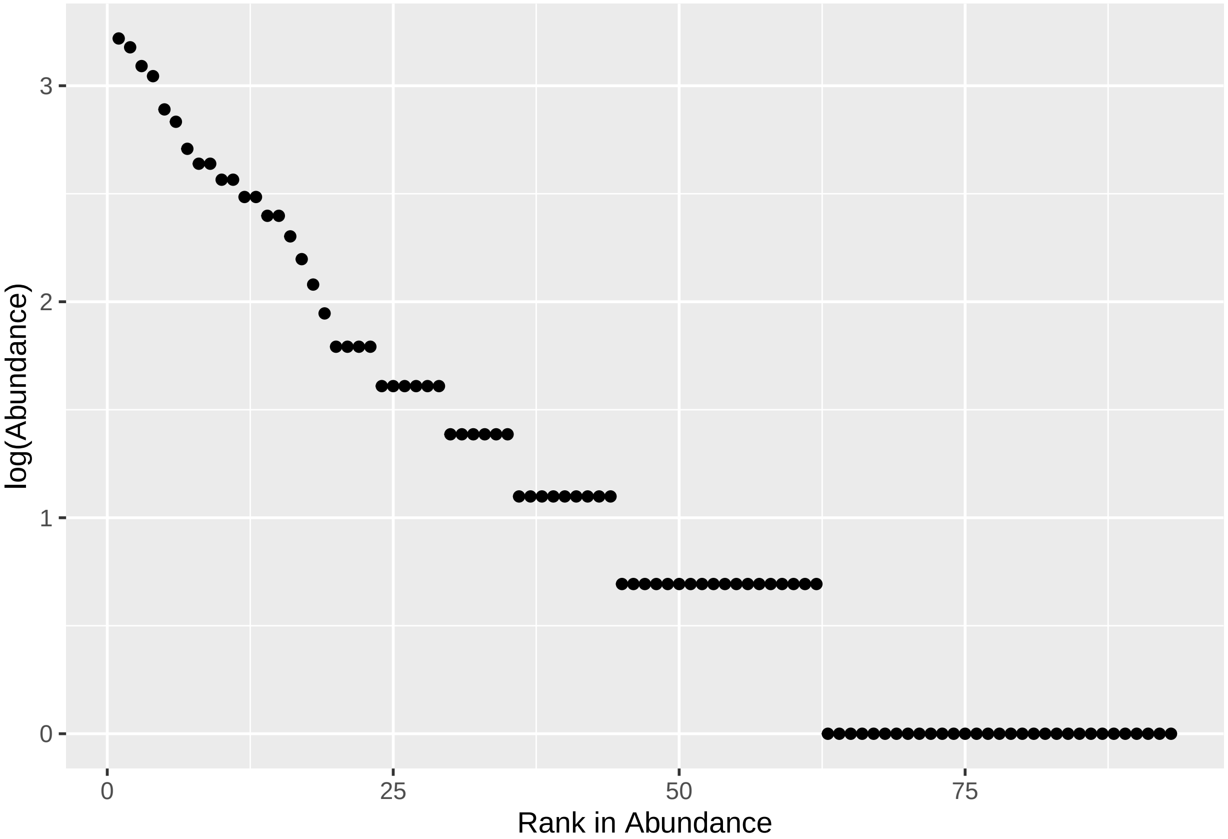
```
## Loading required package: scales
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard
```

```
ranked <- rank_abundance(BCI, 1)
ranks <- c(1:length(ranked))
plot_rac <- tibble(rank = ranks, abundance = ranked)

# make the plot
g <- ggplot(plot_rac, aes(x = rank, y = abundance)) +
   geom_point() +
   scale_y_continuous(trans = 'log',
                      breaks=scales::trans_breaks('log', function(x) exp(x)),
                      labels=scales::trans_format('log', math_format(.x))) +
   xlab('Rank in Abundance') +
   ylab('log(Abundance)')
g
```



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***: Log scales make it easier to see patterns in data with huge variation. In this case our data looks kind of linear and the slope of a best fit line tells us about evenness.

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 <- function(df, site) {
   x <- df[site,]
   S <- s_obs(x)
   D <- vegan::diversity(x, 'inv')
   D / S
}

simpson(BCI, 1)
```

```
##         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}.

```
smith_wilson <- function(df, site) {
   x <- as_vector(df[site, ])
   x_nonzero <- x[x>0]
   1 - (2 / pi) * atan(var(log(x_nonzero)))
}
smith_wilson(BCI, 1)
```

```
## [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***: These values do not match but both report moderately even communities. Because of the bias in Simpson's measure, this might indicate that the most abundant species are less even than the community as a whole.

##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".

```
entropy <- function(df, site) {
   H <- 0
   for (n in df[site,]) {
      if (n > 0) {
         p <- n / sum(df[site,])
         H <- H - p * log(p)
      }
   }
   H
}
entropy(BCI, 1)
```

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

```
vegan::diversity(BCI[1,])
```

```
## [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(df, site) {
   D <- 0
   N <- sum(df[site,])
   for (n in df[site,]) {
      D = D + (n / N)^2
   }
   D
}

d_sub <- 1 - simpD(sb, 1)
d_inv <- 1 / simpD(sb, 1)

d_sub
```

```
## [1] 0.994351
```

```
vegan::diversity(sb[1,], index='simp')
```

```
## [1] 0.994351
```

```
d_inv
```

```
## [1] 177.0219
```

```
vegan::diversity(sb[1,], index='inv')
```

```
## [1] 177.0219
```

Fisher's $\boldsymbol{\alpha}$

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.

```
x_vec <- as_vector(BCI[1,])
vegan::diversity(x_vec[x_vec > 0], 'inv')
```

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

```
vegan::fisher.alpha(x_vec[x_vec > 0])
```

```
## [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***: Fisher's $\boldsymbol{\alpha}$ takes into account a theoretical distribution. Specifically, $\boldsymbol{\alpha}$ is the coefficient in front of the terms of the logseries.

##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.

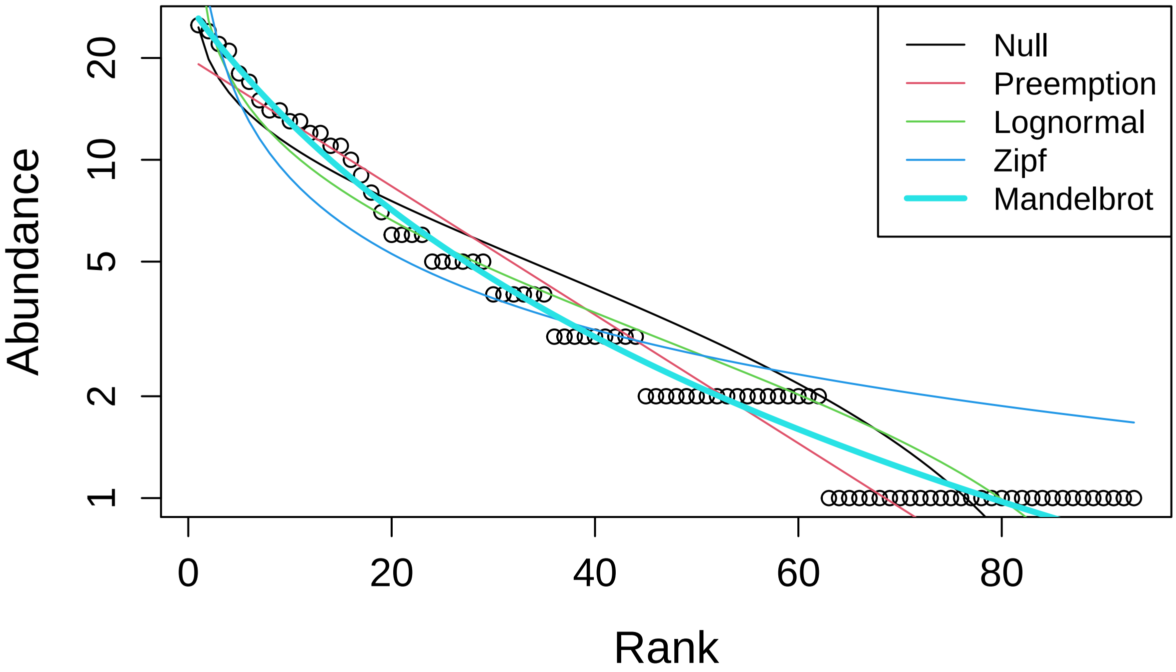
In the R code chunk below, please do the following:

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.

```
RAC <- vegan::radfit(x_vec)
RAC
```

```
##
## 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(RAC, 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***: Mendelbrot fits best according to all three goodness-of-fit colunms.
>
> ***Answer 8b***: This one is not easy. In this context, we might see this distribution as a product of cumulative advantage or preferrential where some positive feedback allows high rank species to further increase in frequency.

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 9a***: The preemption model assumes that the total abundance can be allocated to nonoverlapping niches whose size are proportional to the abundance allocated to them.
>
> ***Answer 9b***: When you take the logarithm of the expression you get $ln(a_r) = ln(N) + ln(\alpha) + (r-1)ln(1-\alpha)$ which is linear in r.

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 10***: With a parameter for every daya point we could perfectly fit our data with a completely useless model that tells us nothing that we dont already know. In order for this kind of a model to be useful it needs to generalize which we only get if we don't overfit our model.

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.

```
simpsons_D <- function(x) {
   N <- sum(x)
   sum((x * (x - 1)) / (N * (N - 1)))
}
print(paste("Simpson's D:", simpsons_D(BCI[1,])))
```

```
## [1] "Simpson's D: 0.0231903163950144"
```

```
print(paste("1 - D:", 1 - simpsons_D(BCI[1,])))
```

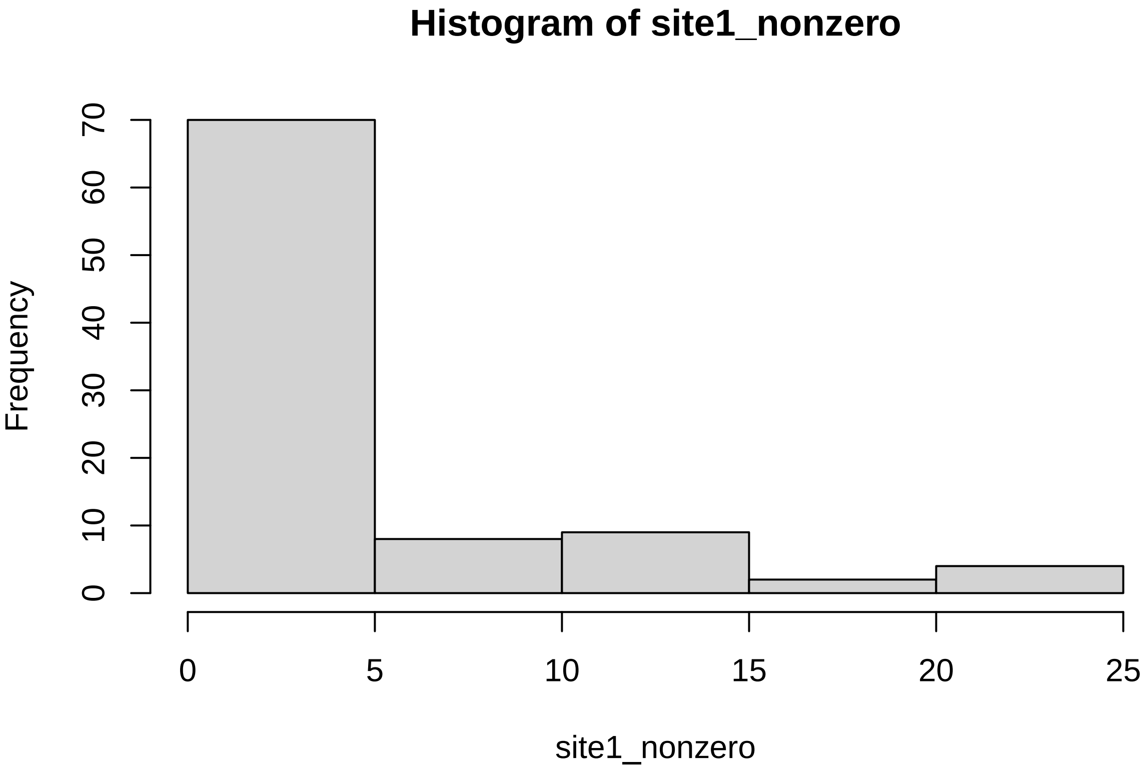
```
## [1] "1 - D: 0.976809683604986"
```

```
print(paste("1 / D:", 1 / simpsons_D(BCI[1,])))
```

```
## [1] "1 / D: 43.1214470284238"
```

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_vector(BCI[1,])
site1_nonzero <- site1[site1 > 0]
hist(site1_nonzero)
```



> The histogram is qualitatively similar to the RAC abundance curve in its shape. In both we see a sharp drop-off and a substantial tail. Despite the different view this gives us on our community, both tell us that most species are rare and a small number are common.

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?

```
require(readxl)
```

```
## Loading required package: readxl
```

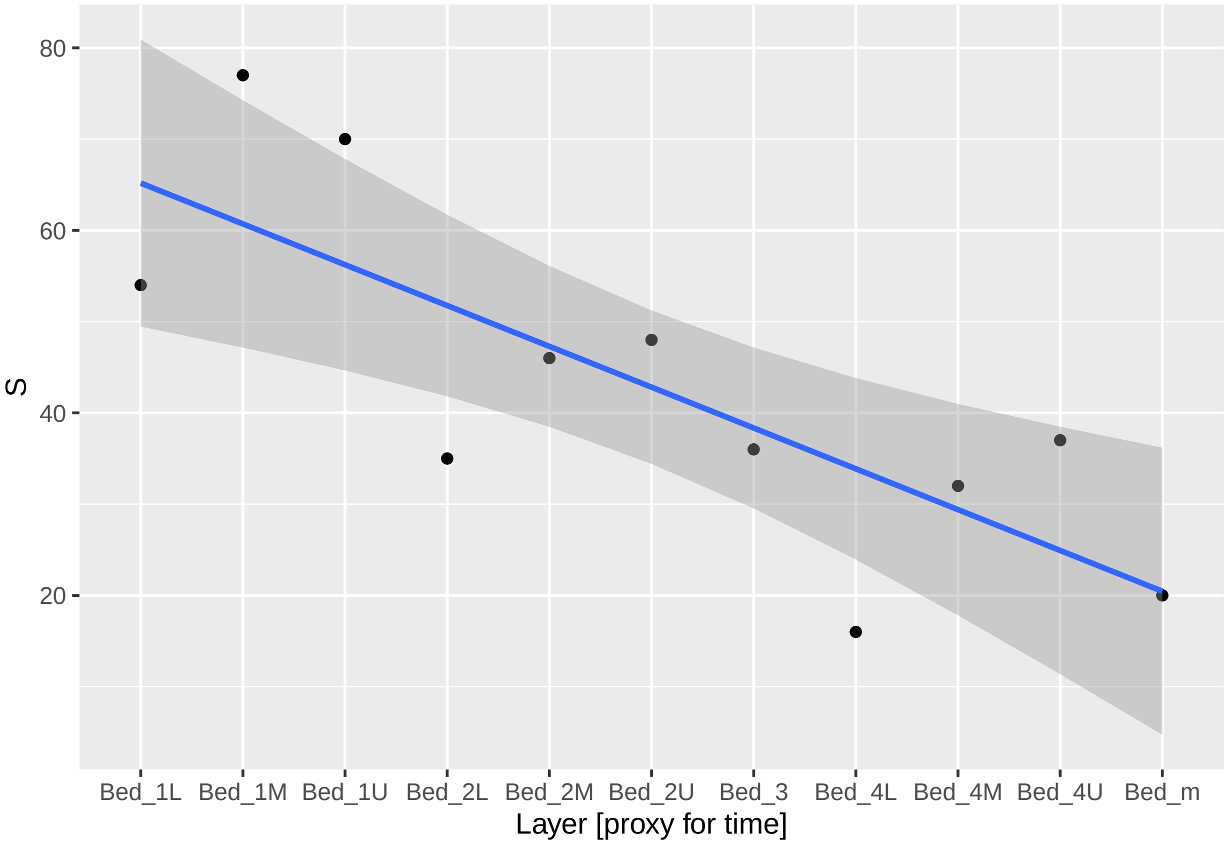
```
require(dplyr)
df <- readxl::read_excel('~/Classwork/Current/QB/Project/species.xlsx')
str(df)
```

```
## tibble[,13] [178 x 13] (S3: tbl_df/tbl/data.frame)
##  $ Genus  : chr [1:178] "Clarias" "Tilapia" "Cichlidae" "1enopus" ...
##  $ Species: chr [1:178] "sp." "sp." "Indet" "sp." ...
##  $ Bed_1L : num [1:178] 1 1 0 1 1 0 0 0 0 1 ...
##  $ Bed_1M : num [1:178] 1 1 1 1 1 0 1 1 1 1 ...
##  $ Bed_1U : num [1:178] 0 0 0 1 1 1 1 1 1 1 ...
##  $ Bed_2L : num [1:178] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Bed_2M : num [1:178] 1 1 0 0 0 0 0 0 0 0 ...
##  $ Bed_2U : num [1:178] 1 1 0 0 0 0 0 0 0 0 ...
##  $ Bed_3  : num [1:178] 1 0 0 0 0 0 0 0 0 0 ...
##  $ Bed_4L : num [1:178] 1 0 0 0 0 0 0 0 0 0 ...
##  $ Bed_4M : num [1:178] 1 0 0 0 0 0 0 0 0 0 ...
##  $ Bed_4U : num [1:178] 1 0 0 0 0 0 0 0 0 0 ...
##  $ Bed_m  : num [1:178] 1 0 0 0 0 0 0 0 0 0 ...
```

```
df_long <- df %>%
  tidyr::pivot_longer(Bed_1L:Bed_m, names_to = 'time', values_to = 'pres', names_transform = list())
time_series <- df_long %>%
  group_by(time) %>%
  summarize(time_sums = sum(pres))
time_series
```

```
## # A tibble: 11 x 2
##    time   time_sums
##    <chr>      <dbl>
##  1 Bed_1L        54
##  2 Bed_1M        77
##  3 Bed_1U        70
##  4 Bed_2L        35
##  5 Bed_2M        46
##  6 Bed_2U        48
##  7 Bed_3         36
##  8 Bed_4L        16
##  9 Bed_4M        32
## 10 Bed_4U        37
## 11 Bed_m         20
```

```
g <- ggplot(time_series, aes(x = time, y = time_sums, group=1)) +
  geom_point() +
  stat_smooth(formula = 'y ~ x', method = 'lm') +
  xlab('Layer [proxy for time]') +
  ylab('S')
g
```



>

Our dataset contains presence absence informaiton for animals in the fossil record at the same site through time. There are 178 species observed at least once, and as the figure shows, species richness tends to decrease in time.

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)**.