5. Worksheet: Alpha Diversity

Mark Hibbins; Z620: Quantitative Biodiversity, Indiana University

22 January, 2019

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).

```
remove(list=ls())
getwd()
```

```
## [1] "C:/Users/Marcus/Box Sync/Courses/Quantitative Biodiversity/QB2019_Hibbins/2.Worksheets/5.AlphaD
```

```
setwd('C:/Users/Marcus/Box Sync/Courses/Quantitative Biodiversity/QB2019_Hibbins/2.Worksheets/5.AlphaDi
library(vegan)
```

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

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

```
## This is vegan 2.5-3
```

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  "Abarema.macradenium" "Acacia.melanoceras" "Acalypha.diversifolia
```

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
```

```
specnumber(BCI[2,])
```

```
##  2
## 84
```

```
specnumber(BCI[3,])
```

```
##  3
## 90
```

```
specnumber(BCI[4,])
```

```
##  4
## 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: specnumber() and the S.obs function return the same value (as they should). The richness for the first four sites are 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.

```
C = function(x) {
  1 - (sum(x == 1) / sum(x))
}

apply(BCI, 1, C)
```

```
##         1         2         3         4         5         6         7
## 0.9308036 0.9287356 0.9200864 0.9468504 0.9287129 0.9174757 0.9326923
##         8         9        10        11        12        13        14
## 0.9443155 0.9095355 0.9275362 0.9152120 0.9071038 0.9242054 0.9132420
##        15        16        17        18        19        20        21
## 0.9350649 0.9267735 0.8950131 0.9193084 0.8891455 0.9114219 0.8946078
##        22        23        24        25        26        27        28
## 0.9066986 0.8705882 0.9030612 0.9095023 0.9115479 0.9088729 0.9198966
##        29        30        31        32        33        34        35
## 0.8983516 0.9221053 0.9382423 0.9411765 0.9220183 0.9239374 0.9267887
##        36        37        38        39        40        41        42
## 0.9186047 0.9379310 0.9306488 0.9268868 0.9386503 0.8880597 0.9299517
##        43        44        45        46        47        48        49
## 0.9140049 0.9168704 0.9234234 0.9348837 0.8847059 0.9228916 0.9086651
##        50
## 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: Because Good's Coverage is calculated as a proportion, it should be a value between 0 and 1.

Answer 2b: If ni equals N, then Good's coverage will equal 0, meaning every individual in the sample belongs to its own species that was only sampled once. This is not good sampling coverage.

Answer 2c: Approximately 7% (1 - 0.93)

Answer 2d: The values of coverage for this dataset are in the range of 85-95%, meaning that the majority of individuals at each site belong to a species that is sampled at least twice.

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

```
soilbac <- read.table('data/soilbac.txt', sep = '\t', header = TRUE, row.names = 1)

soilbac_transpose <- as.data.frame(t(soilbac))
soilbac3 <- soilbac_transpose[3,]

sum(soilbac3)
```

```
## [1] 2533
```

```
specnumber(soilbac3)
```

```
## T1_3
## 1174
```

```
C(soilbac3)
```

```
## [1] 0.6735097
```

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: 2533

Answer 3b: 1174

Answer 3c: The coverage for the KBS sample is much lower (67.3% vs. 93%)

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(x){
  specnumber(x) + (sum(x == 1)^2) / (2 * sum(x == 2))
}

S.chao2 <- function(site, SbyS){
  SbyS = as.data.frame(SbyS)
  x = SbyS[site,]
  SbyS_presenceabsence <- (SbyS > 0) * 1
  Q1 = sum(colSums(SbyS_presenceabsence) == 1)
  Q2 = sum(colSums(SbyS_presenceabsence) == 2)
  S.chao2 <- S.obs(x) + (Q1^2)/(2*Q2)
  return(S.chao2)
```

```
}

S.ace <- function(x, threshold){
  x <- x[x>0]
  S_abund <- length(which(x > threshold))
  S_rare <- length(which(x <= threshold))
  singletons <- length(which(x == 1))
  N_rare <- sum(x[which(x <= threshold)])
  C_ace <- 1 - (singletons / N_rare)
  i <- c(1:threshold)
  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) + (singletons/C_ace) * max(G_ace,0)
  return(S_ace)
}

site1 <- BCI[1,]

S.chao1(site1)
```

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

```
S.chao1(soilbac3)
```

```
##     T1_3
## 3042.658
```

```
S.chao2(site = 1, SbyS = BCI)
```

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

```
S.chao2(site = 3, SbyS = soilbac_transpose)
```

```
##     T1_3
## 21155.39
```

```
S.ace(site1, 10)
```

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

```
S.ace(soilbac3, 10)
```

```
## [1] 5090.28
```

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***: The ACE estimator defines a threshold for comparison of rare vs. common species, while the Chao estimators use singletons and doubletons. The Chao 1 and ACE estimators seem to give roughly similar estimates, while the Chao 2 estimator reports a much larger value for soilbac3. This is probably because the Chao 2 estimator reports richness across multiple sites, and sampling coverage for an invidual site in the soilbac data is lower. All the estimators seem to operate best under different circumstances. but the ACE estimator is less intuitive and also

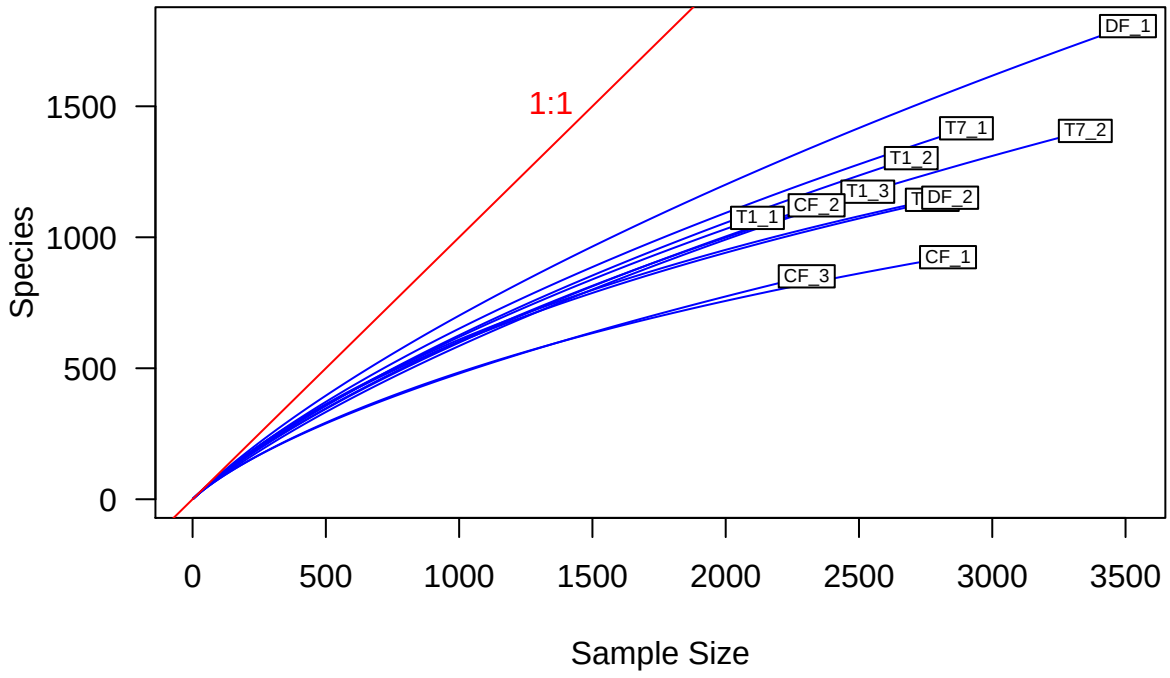
cannot be effectively used in datasets where there are many species with few samples. Therefore, for most situations I would probably use one of the Chao estimators.

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.

```
soilbac_S <- S.obs(soilbac_transpose)
min_N <- min(rowSums(soilbac_transpose))
S_rarefy <- rarefy(x = soilbac_transpose, sample = min_N, se = TRUE)
rarecurve(x = soilbac_transpose, 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.

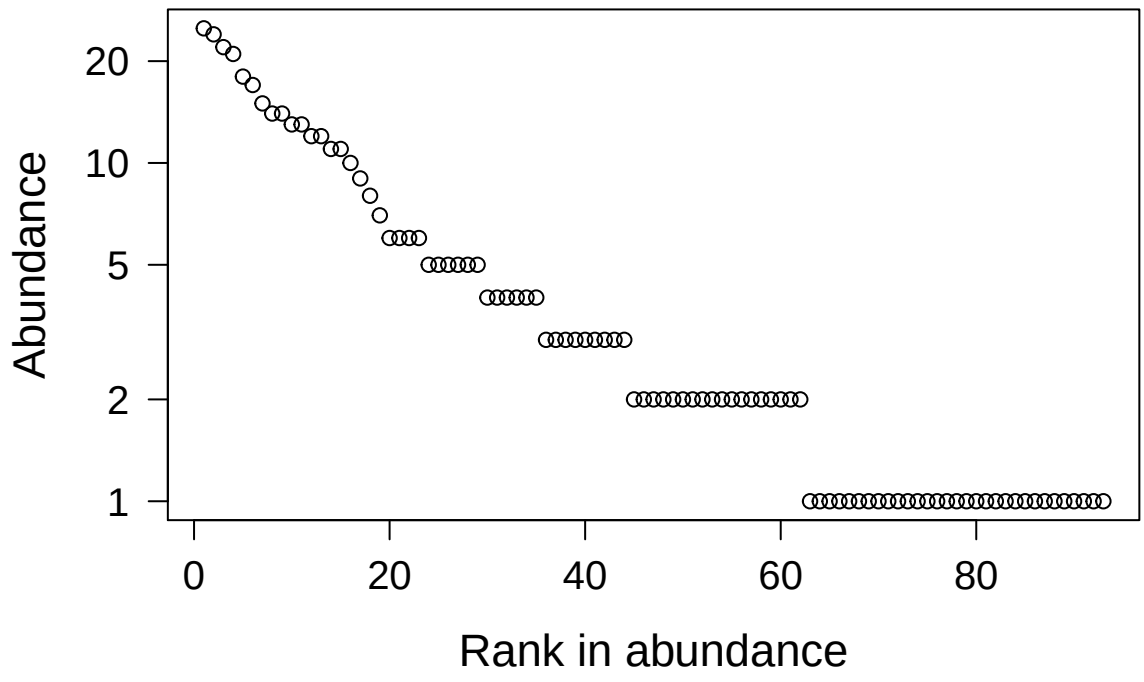
In the R code chunk below, do the following:

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()

rac <- RAC(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, ce

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***: When visualized on the log scale, it makes it look as though the decay in abundance is linear. This is compounded by the fact that the plot is made using points spread horizontally around each rank value. When interpreting figures with a log y-axis we must keep in mind that trends that look linear on a log scale are not actually linear in absolute scale.

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)
}
```

```
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 evenness estimated by E_{var} is higher than for $E_{1/D}$. Since the rank-abundance curve for this dataset appears to operate on a log-scale, E_{var} removes the bias introduced by the more abundant species. resulting in a more even metric. If we compared the figure above on a log-scale y axis to another figure on a linear axis, we would see qualitatively that the abundance curve in the log figure is more even.

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
```

```
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)
}

1/SimpD(site1)
```

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

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

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

```
diversity(site1, 'inv')
```

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

```
diversity(site1, 'simp')
```

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

Question 7: Compare estimates of evenness for `site1` of BCI using $E_{H'}$ and E_{var}. Do they agree? If so, why? If not, why? What can you infer from the results.

> ***Answer 7***: (This is a duplicate of question 6, see above)

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.

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

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

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

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

Question 8: 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 8***: Fisher's $\boldsymbol{\alpha}$ is fitted statistically from the data, it can explicitly account for the effects of sampling bias on estimates of species diversity. $E_{H'}$ and E_{var} are summary statistics that only describe the data at hand.

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.

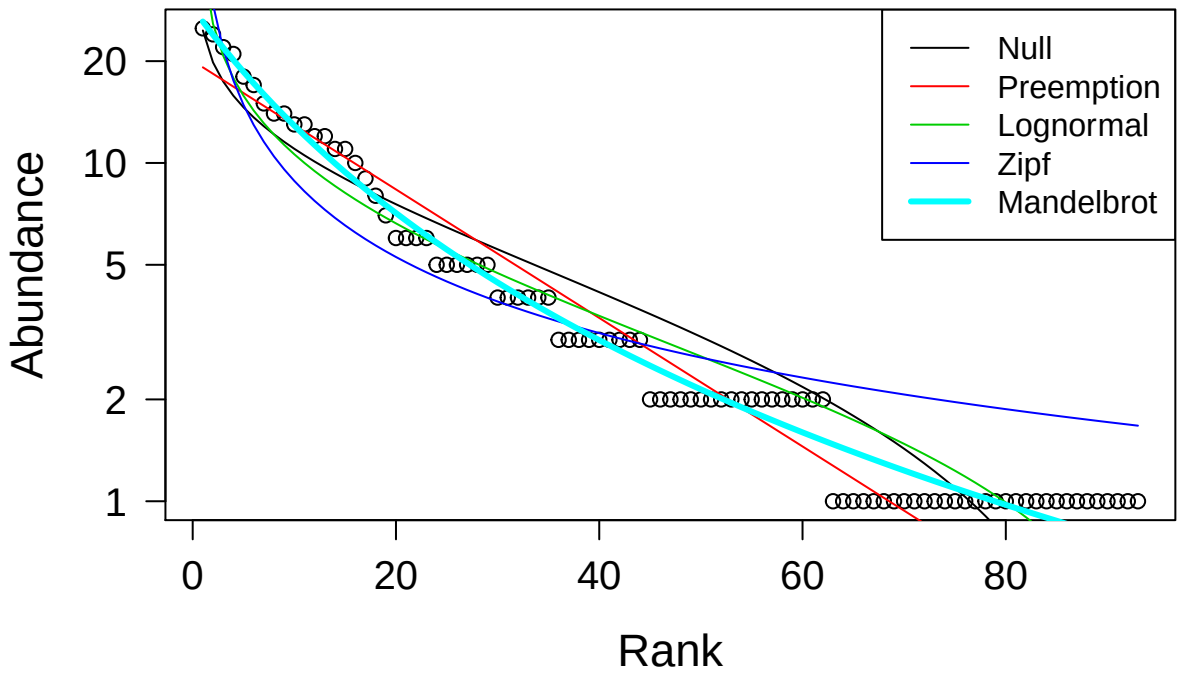
In the R code chunk below, please do the following:

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)
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.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***: All three summary statistics for model fit (deviance, AIC, and BIC), have their lowest values for the Mandelbrot model. Therefore, this model best fits the rank-abundance curve for site1.
>
> ***Answer 9b***: The Mandelbrot model is a methodological extension of the Zipf model, which includes a parameter to control the evenness of the rank abudance curve. The Zipf model is based on Zipf's law, which is an empirical description of a pattern that is consistently observed in ranked data, rather than a model of actual ecological processes. Therefore, while the Mandelbrot model fits the data best, I don't think it provides much insight into ecological processes. We would have to provide a biological explanation for why rank-abundance curves follow Zipf's law.

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 total abundance is directly proportional to total resources, because the parameter for resource consumption α causes the value of N to decay in the equation.
>
> ***Answer 10b***: In the log scale, the preemption model is described by a simple linear equation: $log(a_r) = log(N) + log(\alpha) + (r-1)log(1-\alpha)$. Therefore, when we plot the model fit on a figure with a log-scale y axis, it looks linear.

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: When fitting models, there is always a tradeoff between bias and variance. A model with less parameters introduces bias because it does not capture all the information contained in the data. However, it has fewer degrees of freedom, and therefore less variance, potentially improving its performance when introduced to new data. Models with a large number of parameters will always fit a specific dataset better, because they can capture more information and reduce bias. However, with more degrees of freedom, they become more sensitive to new variance introduced in other datasets, reducing their predictive power; this phenomenon is called overfitting. Penalizing models with a large number of parameters is desirable to prevent overfitting, even if they may explain more of the variance for a specific dataset.

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.

```
SimpD_fin <- function(x){
  D = 0
  N = sum(x)
  for (n_i in x){
    D = D + ((n_i)*(n_i - 1))/((N)*(N-1))
  }
  return(D)
}

SimpD_fin(site1)
```

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

```
1/SimpD_fin(site1)
```

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

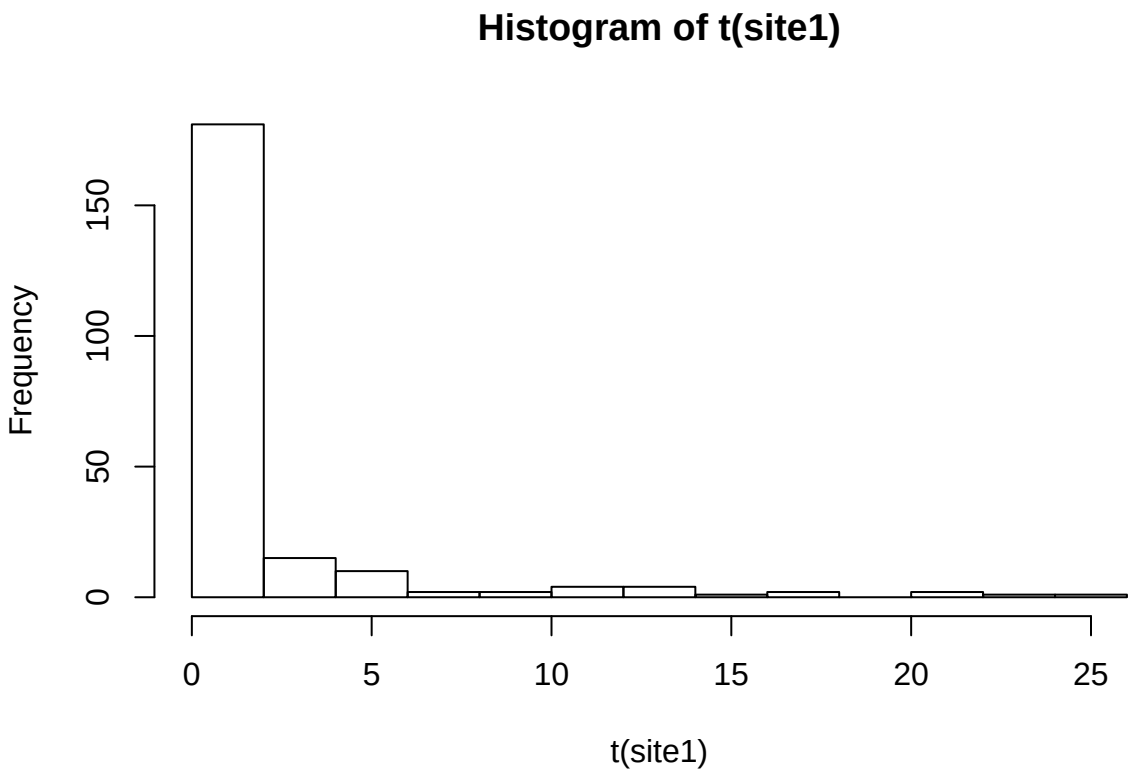
```
1-SimpD_fin(site1)
```

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

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.

   Answer: As seen in the histogram below, the vast majority of species are quite rare, with abundances between 0 and 2. Then, there are a much smaller number of more abundant species. Generally, this is the same trend observed in the rank-abundance curve from earlier.

```
hist(t(site1))
```



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?

```
library(readr)
epiphytes <- read_csv("data/epiphytes.csv")
```

```
## Warning: Missing column names filled in: 'X43' [43], 'X44' [44],
## 'X45' [45], 'X46' [46], 'X47' [47], 'X48' [48], 'X49' [49], 'X50' [50],
## 'X51' [51], 'X52' [52], 'X53' [53], 'X54' [54], 'X55' [55], 'X56' [56],
## 'X57' [57], 'X58' [58], 'X59' [59], 'X60' [60], 'X61' [61], 'X62' [62],
## 'X63' [63], 'X64' [64], 'X65' [65], 'X66' [66], 'X67' [67], 'X68' [68],
## 'X69' [69], 'X70' [70], 'X71' [71], 'X72' [72], 'X73' [73], 'X74' [74],
## 'X75' [75]
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   Origin = col_character(),
##   Phorophyte = col_character(),
##   Tree_structure = col_double(),
##   Wood_density = col_double(),
##   Deciduousness = col_double(),
##   Rugosity = col_double(),
##   X43 = col_character(),
##   X44 = col_character(),
```

```
##   X45 = col_character(),
##   X46 = col_character(),
##   X47 = col_character(),
##   X48 = col_character(),
##   X49 = col_character(),
##   X50 = col_character(),
##   X51 = col_character(),
##   X52 = col_character(),
##   X53 = col_character(),
##   X54 = col_character(),
##   X55 = col_character(),
##   X56 = col_character()
##   # ... with 19 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
epiphytes <- epiphytes[,1:42]
```

> *__Answer__: This dataset has 72 sites, with 36 species across the whole matrix. Based on examining the matrix, some of the species have much lower sampling than others, so we will probably have to do some kind of rarefaction to account for that.

SUBMITTING YOUR ASSIGNMENT

Use Knitr to create a PDF of your completed alpha_assignment.Rmd document, push it to GitHub, and create a pull request. Please make sure your updated repo include both the HTML and RMarkdown files.

Unless otherwise noted, this assignment is due on **Wednesday, January 23rd, 2017 at 12:00 PM (noon)**.