Assignment: Local (alpha) Diversity

Katie Beidler; Z620: Quantitative Biodiversity, Indiana University

22 January, 2017

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. Change "Student Name" on line 3 (above) with your name.
2. Complete as much of the exercise as possible during class; what you do not complete in class will need to be done on your own outside of 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. Be sure to **answer the questions** in this exercise document; they also correspond to the handout. 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 ">".
5. Before you leave the classroom, **push** this file to your GitHub repo.
6. For homework, 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, please submit the completed exercise by creating a **pull request** via GitHub. Your pull request should include this file `alpha_assignment.Rmd` and the PDF output of `Knitr` (`alpha_assignment.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 `/Week2-Alpha` folder, and 4) Load the `vegan` R package (be sure to install if needed).

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

```
## [1] "/Users/bhbeidler/GitHub/QB2017_Beidler/Week2-Alpha"
```

```
setwd("/Users/bhbeidler/GitHub/QB2017_Beidler/Week2-Alpha")
install.packages("vegan", repos = "http://cran.rstudio.com/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/65/897bx8690x5_qclp18zt0sf80000gn/T//Rtmpq9Le0L/downloaded_packages
```

```
require("vegan")
```

```
## Loading required package: vegan

## Warning: package 'vegan' was built under R version 3.3.2

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.4-2
```

2) LOADING DATA

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

```
data(BCI)
str(BCI, max.level=0)
```

```
## 'data.frame':    50 obs. of  225 variables:
##   [list output truncated]
##  - attr(*, "original.names")= chr  "Abarema.macradenium" "Acacia.melanoceras" "Acalypha.diversifolia
```

3) SPECIES RICHNESS

Species richness (S) is simply 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`, 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   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  93  84  90  94 101  85  82  88  90  94  87  84  93  98  93  93  93  89
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
## 109 100  99  91  99  95 105  91  99  85  86  97  77  88  86  92  83  92
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50
##  88  82  84  80 102  87  86  81  81  86 102  91  91  93
```

```r
# Comparing the output of S.obs to 'specnumber()' function from the vegan package
specnumber(BCI)
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  93  84  90  94 101  85  82  88  90  94  87  84  93  98  93  93  93  89
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
## 109 100  99  91  99  95 105  91  99  85  86  97  77  88  86  92  83  92
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50
##  88  82  84  80 102  87  86  81  81  86 102  91  91  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 4 sites (i.e., rows) of the BCI matrix?

> ***Answer 1***:specnumber()`from`vegan‘ returns the same value for observed richness (93) in site1. The species richness for site 1 is 93, 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.

```r
C = function(x = ""){
  1 - (rowSums(x == 1) / rowSums(x))
}

C(BCI)
```

```
##         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
```

```r
# Number of singletons for site 1
y = function(x = ""){
    (rowSums(x == 1))
}
# Portion of taxa in site 1 represented by singletons
c.por = (y(BCI[1,]))/(length(BCI[1,]))
```

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` were represented by singletons?
d. Make some observations about coverage at the BCI plots.

> ***Answer 2a***: The values for Good's Coverage estimator range from 0-1.
>
> ***Answer 2b***: We could conclude that all of the species are singletons.
>
> ***Answer 2c***: 13.8% of the taxa are singletons at site 1.
>
> ***Answer 2d***: C is close to 1 for the majority of sites, so coverage is good as there are few singleton species.

Estimated Richness

In the R code chunk below, do the following:

1. Load the microbial dataset (located in the `/Week2-Alpha/data` folder),
2. Transform and transpose the data as needed (see handout),
3. Create a vector (`soilbac1`) with the bacterial OTU abundances at any site in the dataset,
4. Calculate the observed richness at that particular site, and
5. Calculate the coverage at that particular site

```r
soilbac = read.table("./data/soilbac.txt", sep = "\t", header = TRUE, row.names = 1)
soilbac.t = as.data.frame(t(soilbac))
soilbac1 = soilbac.t[1,]
# Number of sequences for site 1
sum(soilbac1)
```

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

```r
# Richness at site 1
S.obs(soilbac.t[1,])
```

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

```r
# Coverage at site 1
C(soilbac.t[1,])
```

```
##      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?
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***: 2,119 sequences

> ***Answer 3b***: 1074

> ***Answer 3c***: Coverage is lower at site 1 for the KBS sample (C = 0.65) compared to BCI sample (C = 0.93)

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 both `site1` and `soilbac1`.

```
s.chao1 = function(x  = ""){
  S.obs(x) + (sum(x == 1)^2) / (2 * sum(x == 2))
}

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)
}

S.ace = function(x = "", thresh = ""){
x = x[x>0] # excludes zero-abundance taxa
S.abund = length(which(x > thresh)) # richness of abundant taxa
S.rare = length(which(x <= thresh)) # richness of rare taxa
singlt = length(which(x == 1)) # number of singleton taxa
N.rare = sum(x[which(x <= thresh)]) # abundance of rare individuals
C.ace = 1 - (singlt / N.rare) # coverage (prop non-singlt rare inds)
i = c(1:thresh) # threshold abundance range
count = function(i, y){ # counter to go through i range
length(y[y == i])
}
a.1 = sapply(i, count, x) # number of individuals in richness i richness classes
f.1 = (i * (i - 1)) * a.1 # k(k-1)kf sensu Gotelli
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)
}

site1 = BCI[1,]
soilbac1 = soilbac.t[1,]

#Chao 1 Estimator
s.chao1(site1)
```

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

```
s.chao1(soilbac1)
```

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

```
# S site 1 = 119.7
# S soilbac1 = 2628.5

#Chao 2 Estimator
S.chao2(1,BCI)
```

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

```
S.chao2("T1_1",soilbac.t)
```

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

```
# S site 1 = 104.6
# S soilbac1 = 21055.4

# Abundance Based Coverage Estimator (ACE)
S.ace(site1,10)
```

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

```
S.ace(soilbac1,10)
```

```
## [1] 4465.983
```

```
# S site 1 = 159.3
# S soilbac1 = 4466
```

Rarefaction

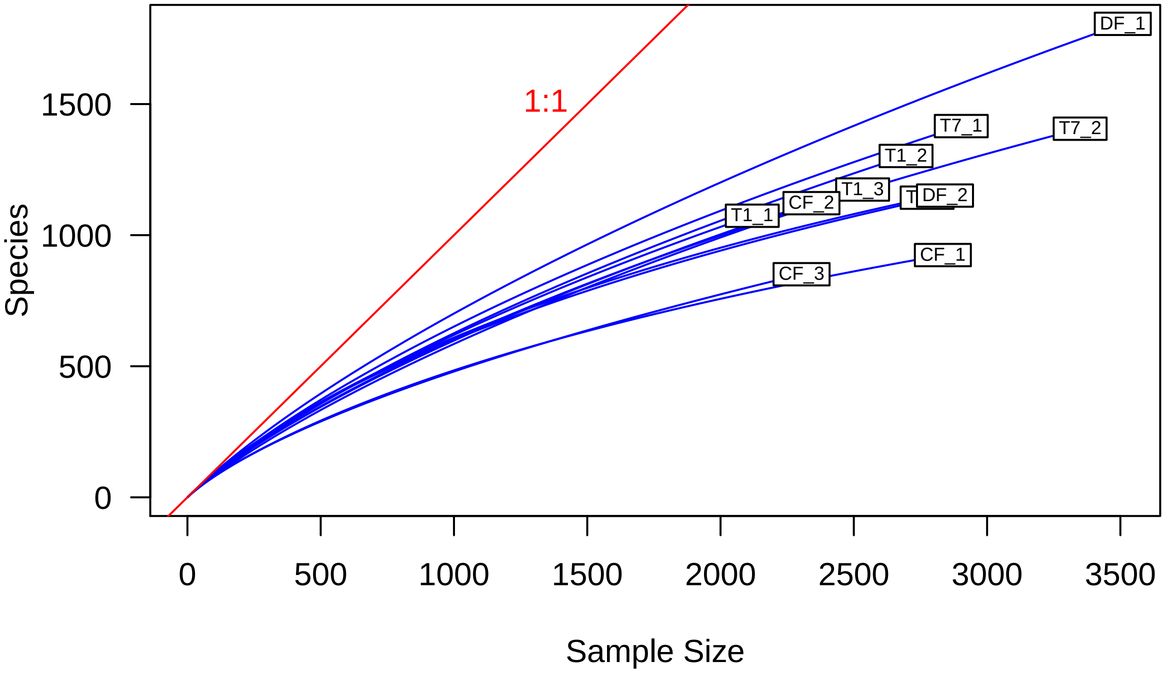
In the R code chunk below, please do the following:

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

```r
# Calculate S.obs() for all samples in soilbac
soilbac.S = S.obs(soilbac.t)

# Determining the size of the smallest sample
min.N = min(rowSums(soilbac.t))
min.N
```

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

```r
# Rarefaction
S.rarefy = rarefy(x = soilbac.t, sample = min.N, se = TRUE)
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')
```



Question 4: What is the difference between ACE and the Chao estimators?

> ***Answer 4***: Chao estimators make inferences based on the number of singletons and doubletons. ACE does not use singletons or doubletons instead it implements a threshold which defines rare speceis as taxa having abundances below the threshold.

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

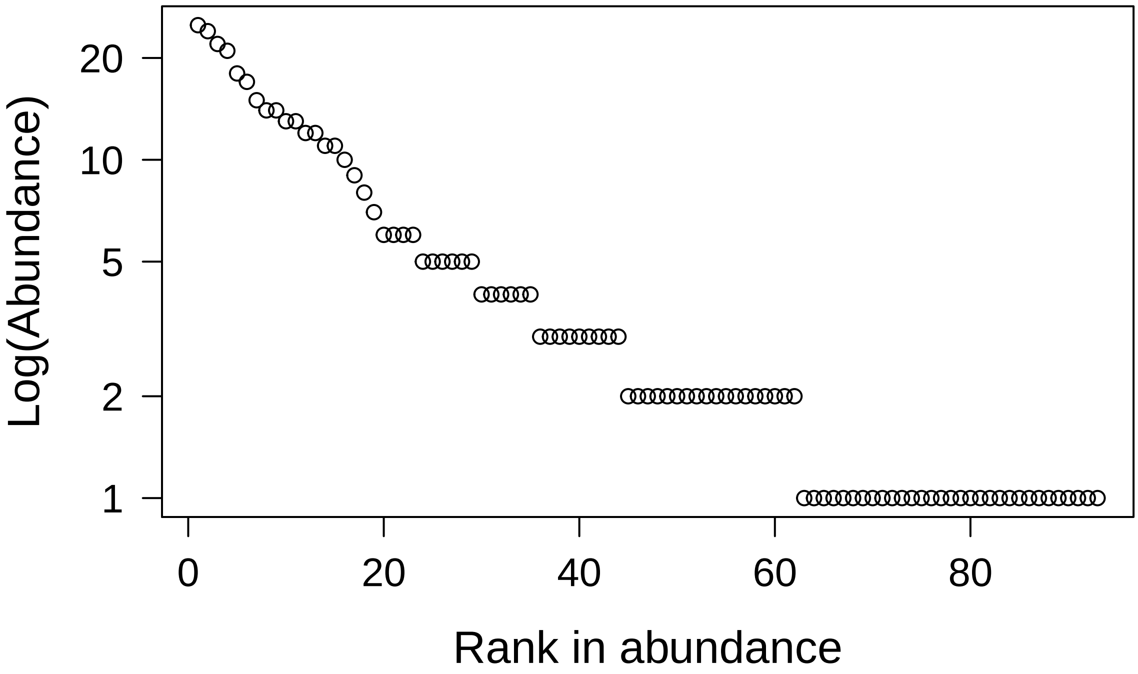
```
RAC = function(x=""){
  x = as.vector(x)
  x.ab = x[x > 0]
  x.ab.ranked = x.ab[order(x.ab, decreasing = TRUE)]
return(x.ab.ranked)
}
```

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)))

# Plot the RAC with natural-log-transformed abundances
opar = par(no.readonly = TRUE)
par(mar = c(5.1, 5.1, 4.1, 2.1))
plot(ranks, log(rac), type = 'p', axes = F,
     xlab = "Rank in abundance", ylab = "Log(Abundance)",
     las = 1, cex.lab = 1.4, cex.axis = 1.25)
box()
axis(side = 1, labels = T, cex.axis = 1.25)
axis(side = 2, las = 1, cex.axis = 1.25,labels = c(1, 2, 5, 10, 20), at = log(c(1, 2, 5, 10, 20)))
```



```
par = opar
```

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

> ***Answer 5***: The log scaled axis makes it visually easier to compare abundances among species to dect how abundance varies among species (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`.

```
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
```

```
# Calculate $E_{1/D}$ for `site1`
# 0.423
```

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)))
}
# Calculate $E_{var}$
Evar(site1)
```

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

```
# 0.506
```

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 they do not agree. $E_{1/D}$ equals 0.42 while $E_{var} = 0.51$. The estimates of evenness differ, because Simpson's evenness metric is sensitive to differences in the few most abundant species. The Evar index is log transformed to decrease bias towards the most abundant species. We can infer that the Evar index may be a better measure of evenness in this instance.

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
```

```
# vegan estimate for Shannon's Index
diversity(site1, index = "shannon")
```

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

```
# The estimates for H are the same 4.018!
```

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".

```
# Calculating Simpsons diversity index
SimpD = function(x = ""){ D=0
N = sum(x)
for (n_i in x){
    D = D + (n_i^2)/(N^2)
  }
return(D) }
SimpD(site1)
```

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

```
# Calculate both the inverse (1/D) and 1 - D
D.inv = 1/SimpD(site1)
D.sub = 1-SimpD(site1)
D.inv
```

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

```
D.sub
```

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

```r
# Using Vegan function for Simpsons Index
diversity(site1, index = "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***: Shannon integrates richness and evenness into a diversity metric while Evar is a measure of evenness. They are different measures and thus it is hard to compare the two.

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.

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

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

```r
Fisher = fisher.alpha(rac)
Fisher
```

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

Question 8: How is Fisher's $\boldsymbol{\alpha}$ different from $E_{H'}$ and E_{var}? E_{var} What does Fisher's $\boldsymbol{\alpha}$ take into account that $E_{H'}$ and E_{var} do not?

> ***Answer 8***: Fisher's $\boldsymbol{\alpha}$ estimates diversity instead of calculating a diversity metric. $E_{H'}$ is a diversity metric and E_{var} is a measure of evenness. Fisher's $\boldsymbol{\alpha}$ also takes into acount sampling error, which $E_{H'}$ and E_{var} do not.

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.

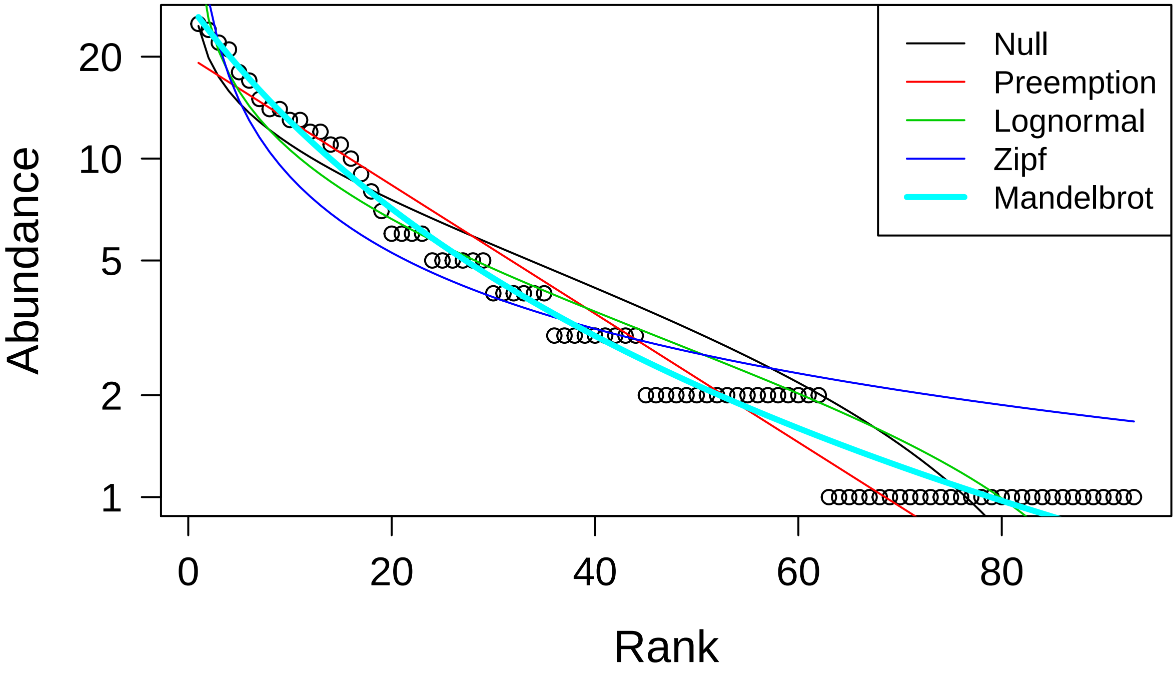
In the R code chunk below, please do the following:

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.

```
# Fitting the predictions of various species abundance models to the  RAC of site 1 in BC1
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
```

```
# Plotting the empirical RAC and the predicted RAC
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***: The model that best fits the RAC for site 1 is the Zipf-Mandelbrot curve as it has the lowest AIC & BIC values and appears to visually fit for the data.
>
> ***Answer 9b***: SAD models can help make inferences about the processes that influence the structure of communities. However, it is important to be cautious when interpreting the biology of a system based from the fit of a model. Frontier (1985) attempted to describe the mechanisms behind the Zipf–Mandelbrot model by suggesting that specialist species are only able to establish after generalist species; this may result in a few abundant species along with many rare species with similar abundances. I hesitate to make inferences about the forces structuring this community, because we have only looked at one site and many ecological communities have a few abundant species with the majority of species being rare.
>
> 1. Frontier S.1985. Diversity and structure in aquatic ecosystems, pp. 253–312. In Barnes M. (eds.), Oceanography and marine biology, an annual review. Aberdeen University Press, Aberdeen, United Kingdom.

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***: The preemption model assumes that total abundances are proportional to the total resources that can be preempted (species preempt some proportion of the total limiting resource). In other words, the abundance of each species is proportional to the niche they occupy or their share of the resource.
>
> ***Answer 10b***: The preemption model looks like a straight line on the plot of abundance against rank because the system is dominated by a few species and characterized by low diversity which results in a rapid decrease in species abundances with rank.

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***: The more parameters a model has the, the better it will fit the data set or the more variance it will explain. Overfitting will reduce the predictive power of the model. To avoid having an over parameterized model, AIC and BIC criterion assign penalties that correspond to the number of parameters in a 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.

```
SimpD(site1)
```

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

```
D.inv = 1/SimpD(site1)
D.sub = 1-SimpD(site1)
D.inv
```

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

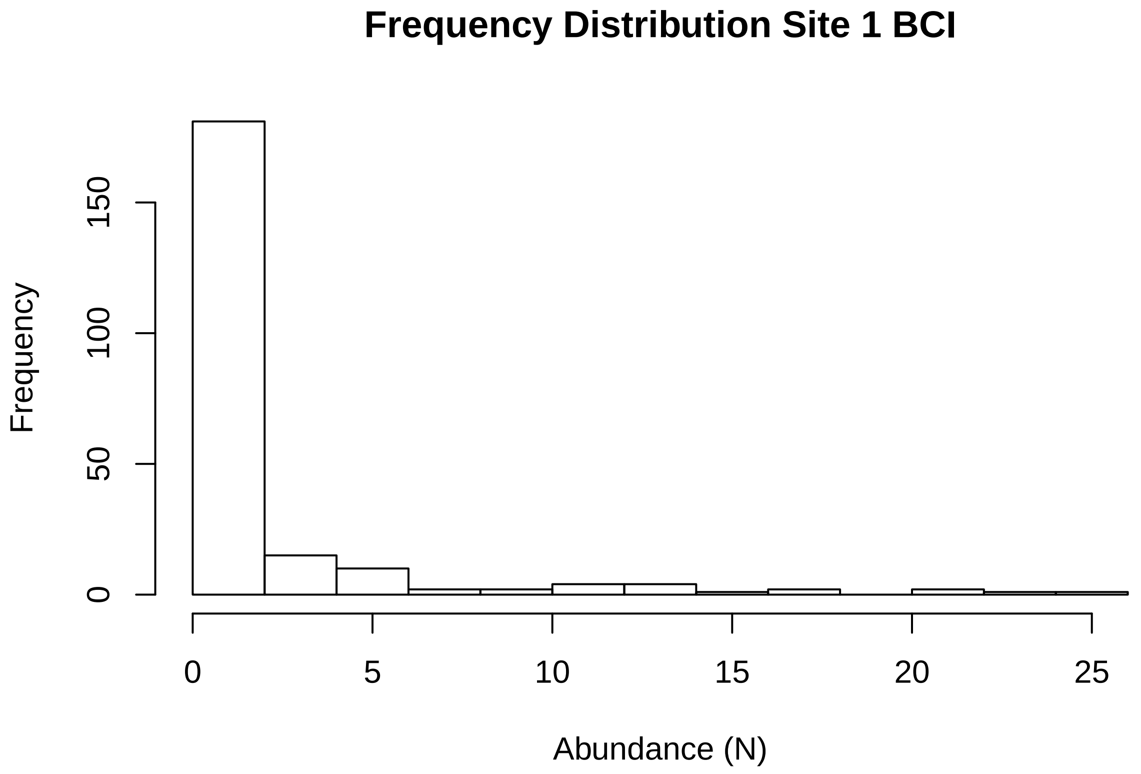
```
D.sub
```

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

Answer Synthesis1: Simpson's D = 0.25, 1 - D = 0.97, and Simpson's inverse (i.e. 1/D) = 39.4

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.t = t(site1)
hist(site1.t, xlab ="Abundance (N)", ylab = "Frequency", main = "Frequency Distribution Site 1 BCI")
```



```
# x is the abundance
# Y is the number of species
# so there are more species with low abundance
```

Answer Synthesis2: The histogram shows that most species are rare and only a few species are abundant.

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?

```
# Reading in the data
plant = read.csv("/Users/bhbeidler/GitHub/QB2017_DivPro/Data/HF_plants.csv")
str(plant)
```

```
## 'data.frame':    96 obs. of  43 variables:
##  $ year     : int  2006 2006 2006 2006 2006 2006 2006 2006 2006 2006 ...
##  $ plot     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ treatment: int  1 4 2 2 4 3 2 4 3 2 ...
##  $ acersp   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ acepen   : int  2 2 5 6 3 2 5 3 8 4 ...
##  $ acerub   : int  3 5 12 6 5 2 3 10 4 3 ...
##  $ aranud   : int  14 12 0 14 5 14 42 36 25 6 ...
##  $ aritri   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ betale   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ betlen   : int  1 1 0 0 0 0 4 7 0 0 ...
##  $ betspp   : int  0 0 0 0 0 0 1 3 1 0 ...
##  $ carpen   : int  2 52 15 0 1 172 161 20 12 3 ...
##  $ casden   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ clibor   : int  0 0 0 0 0 0 0 0 0 34 ...
##  $ coptri   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ craspp   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ denobs   : int  0 8 10 47 63 70 101 84 194 61 ...
##  $ denpun   : int  0 2 102 0 3 0 0 1 47 81 ...
##  $ dipcom   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ dryspp   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ faggra   : int  1 2 1 0 1 0 0 2 1 1 ...
##  $ gaupro   : int  117 80 61 74 109 149 189 85 63 69 ...
##  $ goopub   : int  0 11 0 0 0 0 0 0 0 0 ...
##  $ hupluc   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ lyolig   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ maican   : int  170 336 143 236 265 248 185 19 24 18 ...
##  $ medvir   : int  3 9 2 3 1 0 2 2 0 1 ...
##  $ mitrep   : int  50 303 10 0 0 0 0 0 0 256 ...
##  $ monuni   : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ pinstr   : int  0 3 0 0 1 1 0 0 0 0 ...
##  $ pruser   : int  0 0 3 3 2 0 1 0 0 0 ...
##  $ quealb   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ querub   : int  1 0 1 1 6 2 7 3 0 2 ...
##  $ smirac   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ snag     : int  0 1 1 0 2 0 0 0 0 0 ...
##  $ tribor   : int  14 55 19 9 30 4 33 24 30 11 ...
##  $ tsucan   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ unkspp   : int  0 1 1 0 1 0 0 4 0 3 ...
##  $ uvuses   : int  27 32 39 60 55 20 30 3 6 21 ...
##  $ vaccspp  : int  56 48 0 128 29 133 401 57 58 29 ...
##  $ vibace   : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ vibden   : int  0 0 0 0 0 0 0 0 0 2 ...
##  $ viblen   : int  0 0 0 0 6 0 1 3 3 0 ...
```

```
# There are 40 tree species. For each year and each treatment there are 6 sites. Treatments include: 1=

# To look at alpha diversity in the control plots for the last year of the study 2009- we need to subse

install.packages("dplyr", repos = "http://cran.rstudio.com/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/65/897bx8690x5_qclp18zt0sf80000gn/T//Rtmpq9Le0L/downloaded_packages
```

```
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
```

```
# Control plots- 2009
C_2009_plant = filter(plant, treatment == 1,year == 2009)

# Turn C_2009_plant into a site by species matrix
C_2009_plant = C_2009_plant[, 4:43]

# Calculating the Observed Richness for our six sites
S.obs(C_2009_plant)
```

```
## [1] 15 11 20 16 19 14
```

```
# Richness ranges between 11 and 20
# Calculating Good's coverage for our six sites
C(C_2009_plant)
```

```
## [1] 0.9871795 0.9991863 0.9801136 0.9976905 0.9891599 0.9824561
```

```
# The sites seemed to have well sampled as coverage is between 0.98 and 0.99

# Calculating Chao2 for site 1
S.chao2(1,C_2009_plant)
```

```
##  1
## 16
```

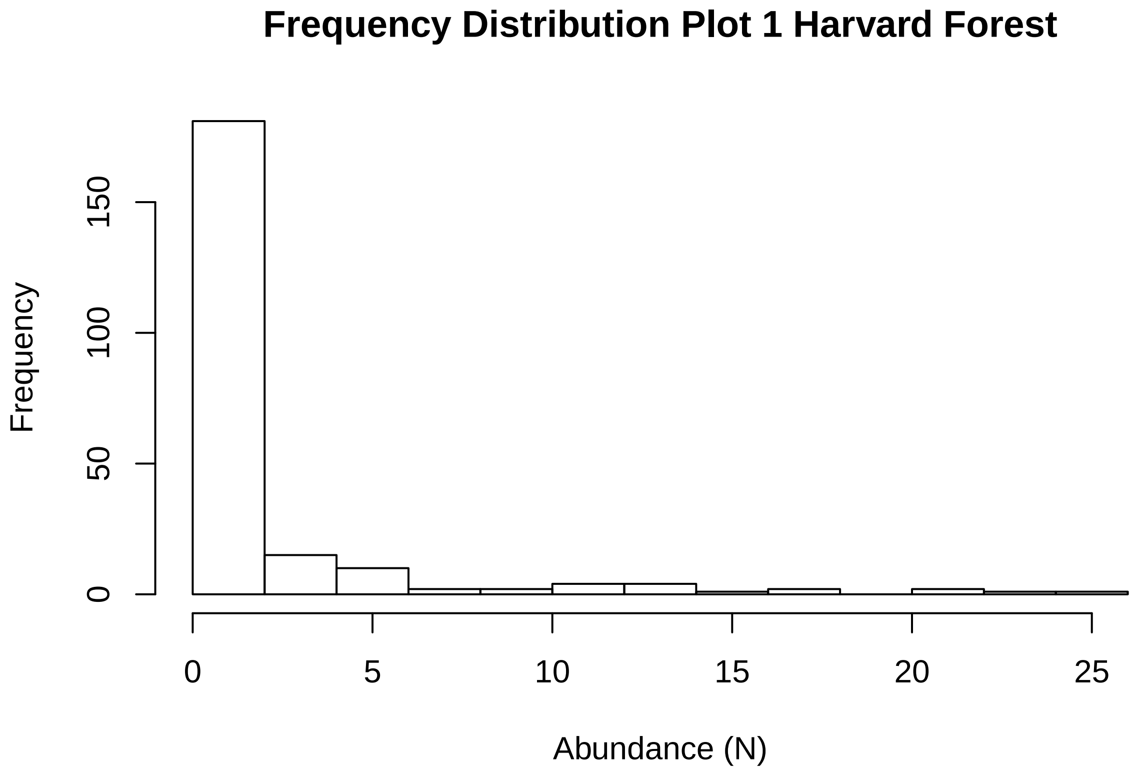
```
# Chao2 = 16, similar to obs. richness

# Comparing the different evenness estimates
SimpE(C_2009_plant[1,])
```

```
##         1
## 0.3140838
```

```
Evar(C_2009_plant[1,])
```

```
## [1] 0.2302884
```

```
# The discrepency between evenness estimates may have to do with the number of abundant species- lookin
hist(site1.t, xlab ="Abundance (N)", ylab = "Frequency", main = "Frequency Distribution Plot 1 Harvard F
```



```
# Diversity metrics for plot 1
diversity(C_2009_plant[1,], index = "shannon")
```

```
## [1] 1.898484
```

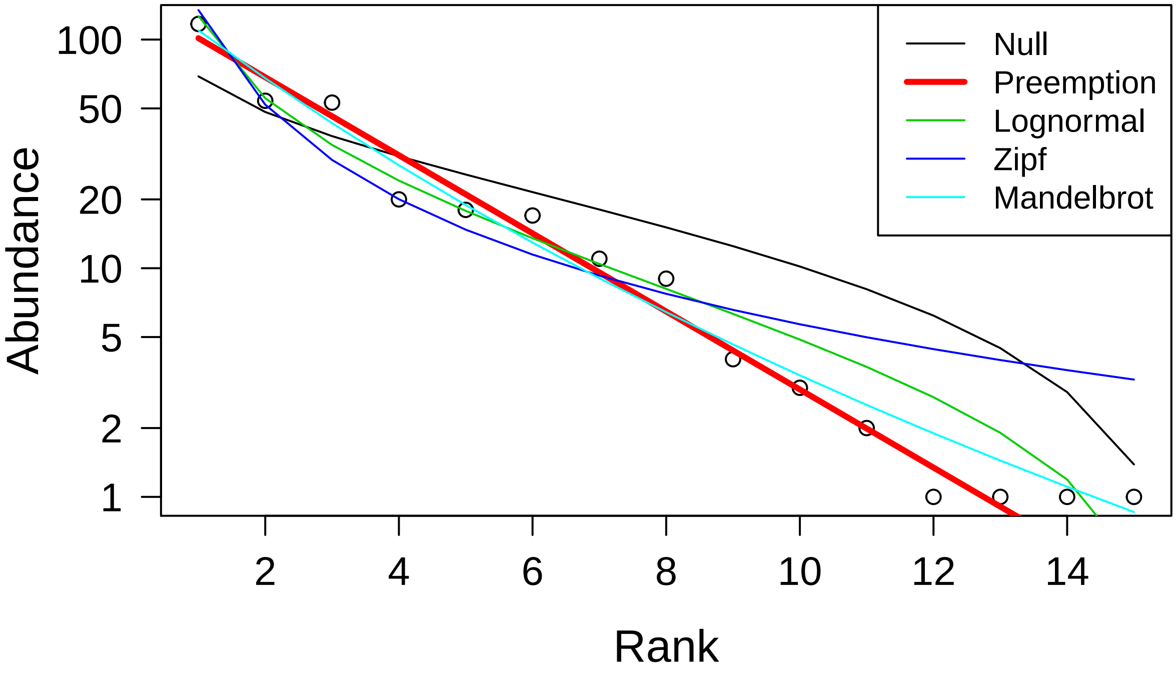
```
diversity(C_2009_plant[1,], index = "simp")
```

```
## [1] 0.7877424
```

```
# Fitting the predictions of various species abundance models to the  RAC of plot1 for HF data
RACresults_HF = radfit(C_2009_plant[1,])
RACresults_HF
```

```
##
## RAD models, family poisson
## No. of species 15, total abundance 312
##
##             par1        par2    par3    Deviance AIC     BIC
## Null                                     81.648  139.327 139.327
## Preemption  0.32505                      14.126   73.805  74.513
## Lognormal   2.0945      1.4981           16.020   77.698  79.115
## Zipf        0.43136    -1.3741           37.544   99.223 100.639
## Mandelbrot  3.5684e+07 -6.9899  12.976   11.694   75.373  77.497
```

```
# Plotting the empirical RAC and the predicted RAC
plot.new()
plot(RACresults_HF, las = 1, cex.lab = 1.4, cex.axis = 1.25)
```



```
#The Niche Preemption Model seems to fit best in this case but is not that different from the Mandelbro
```

Answer Synthesis3: Because our data set includes treatments and multiple years worth of data, I thought it would be a good idea to look at alpha diversity in the control plots (six sites) for 1 time period (2009). The sampling effort for the sites appears to be sufficient as Good's Coverage values range between 0.98 and 0.99. Observed richness ranged between 11 and 20. Observed richness at site 1 (15) was similar to the Chao2 estimate for richness (16). Similar to the BCI data, there appears to be a few species with high abundances while the majority of species have low abundances.

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 25th, 2015 at 12:00 PM (noon)**.