Assignment: Local (alpha) Diversity

Matt Gibson; Z620: Quantitative Biodiversity, Indiana University

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).

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

```
## [1] "C:/Users/matth/Documents/bin/QB2017_Gibson/Week2-Alpha"
```

```r
setwd("c:/Users/matth/Documents/bin/QB2017_Gibson/Week2-Alpha")
require('vegan')
```

```
## Loading required package: vegan

## Warning: package 'vegan' was built under R version 3.2.5

## Loading required package: permute

## Warning: package 'permute' was built under R version 3.2.5
## 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).

```r
data(BCI)
str(BCI)
```

```
## 'data.frame':    50 obs. of  225 variables:
##  $ Abarema.macradenia             : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Vachellia.melanoceras          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acalypha.diversifolia          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acalypha.macrostachya          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Adelia.triloba                 : int  0 0 0 3 1 0 0 0 5 0 ...
##  $ Aegiphila.panamensis           : int  0 0 0 0 1 0 1 0 0 1 ...
##  $ Alchornea.costaricensis        : int  2 1 2 18 3 2 0 2 2 2 ...
##  $ Alchornea.latifolia            : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Alibertia.edulis               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Allophylus.psilospermus        : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Alseis.blackiana               : int  25 26 18 23 16 14 18 14 16 14 ...
##  $ Amaioua.corymbosa              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Anacardium.excelsum            : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Andira.inermis                 : int  0 0 0 0 1 1 0 0 1 0 ...
##  $ Annona.spraguei                : int  1 0 1 0 0 0 0 1 1 0 ...
##  $ Apeiba.glabra                  : int  13 12 6 3 4 10 5 4 5 5 ...
##  $ Apeiba.tibourbou               : int  2 0 1 1 0 0 0 1 0 0 ...
##  $ Aspidosperma.desmanthum        : int  0 0 0 1 1 1 0 0 0 1 ...
##  $ Astrocaryum.standleyanum       : int  0 2 1 5 6 2 2 0 2 1 ...
##  $ Astronium.graveolens           : int  6 0 1 3 0 1 2 2 0 0 ...
##  $ Attalea.butyracea              : int  0 1 0 0 0 1 1 0 0 0 ...
##  $ Banara.guianensis              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Beilschmiedia.pendula          : int  4 5 7 5 8 6 5 9 11 14 ...
##  $ Brosimum.alicastrum            : int  5 2 4 3 2 2 6 4 3 6 ...
##  $ Brosimum.guianense             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Calophyllum.longifolium        : int  0 2 0 2 1 2 2 2 2 0 ...
##  $ Casearia.aculeata              : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Casearia.arborea               : int  1 1 3 2 4 1 2 3 9 7 ...
##  $ Casearia.commersoniana         : int  0 0 1 0 1 0 0 0 1 0 ...
##  $ Casearia.guianensis            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Casearia.sylvestris            : int  2 1 0 0 0 3 1 0 1 1 ...
##  $ Cassipourea.guianensis         : int  2 0 1 1 3 4 4 0 2 1 ...
##  $ Cavanillesia.platanifolia      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cecropia.insignis              : int  12 5 7 17 21 4 0 7 2 16 ...
##  $ Cecropia.obtusifolia           : int  0 0 0 0 1 0 0 2 0 2 ...
##  $ Cedrela.odorata                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ceiba.pentandra                : int  0 1 1 0 1 0 0 1 0 1 ...
##  $ Celtis.schippii                : int  0 0 0 2 2 0 1 0 0 0 ...
##  $ Cespedesia.spathulata          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chamguava.schippii             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chimarrhis.parviflora          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Maclura.tinctoria              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chrysochlamys.eclipes          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chrysophyllum.argenteum        : int  4 1 2 2 6 2 3 2 4 2 ...
##  $ Chrysophyllum.cainito          : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Coccoloba.coronata             : int  0 0 0 1 2 0 0 1 2 1 ...
##  $ Coccoloba.manzinellensis       : int  0 0 0 0 0 0 0 2 0 0 ...
##  $ Colubrina.glandulosa           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cordia.alliodora               : int  2 3 3 7 1 1 2 0 0 2 ...
##  $ Cordia.bicolor                 : int  12 14 35 23 13 7 5 10 7 13 ...
##  $ Cordia.lasiocalyx              : int  8 6 6 11 7 6 6 3 0 4 ...
##  $ Coussarea.curvigemma           : int  0 0 0 1 0 2 1 0 1 1 ...
##  $ Croton.billbergianus           : int  2 2 0 11 6 0 0 4 2 0 ...
##  $ Cupania.cinerea                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cupania.latifolia              : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Cupania.rufescens              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cupania.seemannii              : int  2 2 1 0 3 0 1 2 2 0 ...
##  $ Dendropanax.arboreus           : int  0 3 6 0 5 2 1 6 1 3 ...
##  $ Desmopsis.panamensis           : int  0 0 4 0 0 0 0 0 0 1 ...
##  $ Diospyros.artanthifolia        : int  1 1 1 1 0 0 0 0 0 1 ...
##  $ Dipteryx.oleifera              : int  1 1 3 0 0 0 0 2 1 2 ...
##  $ Drypetes.standleyi             : int  2 1 2 0 0 0 0 0 0 0 ...
##  $ Elaeis.oleifera                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Enterolobium.schomburgkii      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Erythrina.costaricensis        : int  0 0 0 0 0 3 0 0 1 0 ...
##  $ Erythroxylum.macrophyllum      : int  0 1 0 0 0 0 0 1 1 1 ...
##  $ Eugenia.florida                : int  0 1 0 7 2 0 0 1 1 3 ...
##  $ Eugenia.galalonensis           : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Eugenia.nesiotica              : int  0 0 1 0 0 0 5 4 3 0 ...
##  $ Eugenia.oerstediana            : int  3 2 5 1 5 2 2 3 3 3 ...
##  $ Faramea.occidentalis           : int  14 36 39 39 22 16 38 41 33 42 ...
##  $ Ficus.colubrinae               : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Ficus.costaricana              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.insipida                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.maxima                   : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.obtusifolia              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.popenoei                 : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Ficus.tonduzii                 : int  0 0 1 2 1 0 0 0 0 0 ...
##  $ Ficus.trigonata                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.yoponensis               : int  1 0 0 0 0 1 1 0 0 0 ...
##  $ Garcinia.intermedia            : int  0 1 1 3 2 1 2 2 1 0 ...
##  $ Garcinia.madruno               : int  4 0 0 0 1 0 0 0 0 1 ...
##  $ Genipa.americana               : int  0 0 1 0 0 0 1 0 1 1 ...
##  $ Guapira.myrtiflora             : int  3 1 0 1 1 7 3 1 1 1 ...
##  $ Guarea.fuzzy                   : int  1 1 0 1 3 0 0 2 0 3 ...
##  $ Guarea.grandifolia             : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Guarea.guidonia                : int  2 6 2 5 3 4 4 0 1 5 ...
##  $ Guatteria.dumetorum            : int  6 16 6 3 9 7 8 6 2 2 ...
##  $ Guazuma.ulmifolia              : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Guettarda.foliacea             : int  1 5 1 2 1 0 0 4 1 3 ...
##  $ Gustavia.superba               : int  10 5 0 1 3 1 8 4 4 4 ...
##  $ Hampea.appendiculata           : int  0 0 1 0 0 0 0 0 2 1 ...
##  $ Hasseltia.floribunda           : int  5 9 4 11 9 2 7 6 3 4 ...
##  $ Heisteria.acuminata            : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ Heisteria.concinna              : int  4 5 4 6 4 8 2 5 1 5 ...
##  $ Hirtella.americana              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Hirtella.triandra               : int  21 14 5 4 6 6 7 14 8 7 ...
##  $ Hura.crepitans                  : int  0 0 0 0 0 2 1 1 0 0 ...
##  $ Hieronyma.alchorneoides         : int  0 2 0 0 0 0 0 0 1 0 ...
##   [list output truncated]
##  - attr(*, "original.names")= chr  "Abarema.macradenium" "Acacia.melanoceras" "Acalypha.diversifolia
```

```
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
    }

#For site 1 using `S.obs`
print(S.obs(BCI[1,]))
```

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

```
#For site 1 using `specnumber()`
print(specnumber(BCI[1,]))
```

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

```
#For first 4 sites:
print(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 4 sites (i.e., rows) of the BCI matrix?

> ***Answer 1***: Yes, `specnumber()` returns the same value for observed richness in site1 as foes `S.obs`. The species richness for the first 4 sites is 93, 84, 90, and 94, respectively.

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) / rowSums(x))
}

#ADD LOOP HERE
d <- dim(BCI)
s <- seq(2, d[1])
coverage <- C(BCI[1, ])
for (i in s) {
  coverage <- cbind(coverage, C(BCI[i, ]))
}
print(coverage)
```

```
##    coverage
## 1 0.9308036 0.9287356 0.9200864 0.9468504 0.9287129 0.9174757 0.9326923
##
## 1 0.9443155 0.9095355 0.9275362 0.915212 0.9071038 0.9242054 0.913242
##
## 1 0.9350649 0.9267735 0.8950131 0.9193084 0.8891455 0.9114219 0.8946078
##
## 1 0.9066986 0.8705882 0.9030612 0.9095023 0.9115479 0.9088729 0.9198966
##
## 1 0.8983516 0.9221053 0.9382423 0.9411765 0.9220183 0.9239374 0.9267887
##
## 1 0.9186047 0.937931 0.9306488 0.9268868 0.9386503 0.8880597 0.9299517
##
## 1 0.9140049 0.9168704 0.9234234 0.9348837 0.8847059 0.9228916 0.9086651
##
## 1 0.9143519
```

```
range(coverage)
```

```
## [1] 0.8705882 0.9468504
```

```
# Number of singletons
sum(BCI[1,] == 1)/S.obs(BCI[1,])
```

```
##         1
## 0.3333333
```

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***: Between 0 and 1.
>
> ***Answer 2b***: It would be 0.
>
> ***Answer 2c***: 31 singleton taxa. Proportion of singleton taxa = .3333 (# of singletons/S.obs for site 1)
>
> ***Answer 2d***: Coverage is around .9 for all sites. Values for Good's Coverage range from 0.8705882 0.9468504. The lowest was site 23. The highest was site 4.

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=T, row.names = 1)
soilbac.t <- as.data.frame(t(soilBac))
soilbac1 <- soilbac.t[1,]

#Obs richness
print(S.obs(soilbac1))
```

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

```r
#Coverage
print(C(soilbac1))
```

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

```r
# Number sequences recovered
sum(soilbac1)
```

```
## [1] 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***: We recovered 2119 sequences from the sample soilbac1.
>
> ***Answer 3b***: The observed richness was 1074
>
> ***Answer 3c***: The coverage for the BCI site1 (0.9308036) was higher than the KBS sample site1 (0.6479471).

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`.

```r
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 = 10){
  x <- x[x>0]
  S.abund <- length(which(x > thresh))
  S.rare <- length(which(x <= thresh))
  singlt <- length(which(x == 1))
  N.rare <- sum(x[which(x <= thresh)])
  C.ace <- 1 - (singlt / N.rare)
  i <- c(1:thresh)
  count <- function(i, y){
    length(y[y == i])
  }

  a.1 <- sapply(i, count, x)
  f.1 <- (i * (i -1)) * a.1
  G.ace <- (S.rare/C.ace)*(sum(f.1)/(N.rare*(N.rare - 1)))
  S.ace <- S.abund + (S.rare/C.ace) + (singlt/C.ace) * max(G.ace,0)
  return(S.ace)
}

#Chao1
S.chao1(BCI[1,])
```

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

```r
S.chao1(soilbac1)
```

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

```r
#Chao2
S.chao2(1, BCI)
```

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

```r
S.chao2('T1_1', soilbac.t)
```

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

```r
#Chao3
S.ace(BCI[1,])
```

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

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

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

Rarefaction

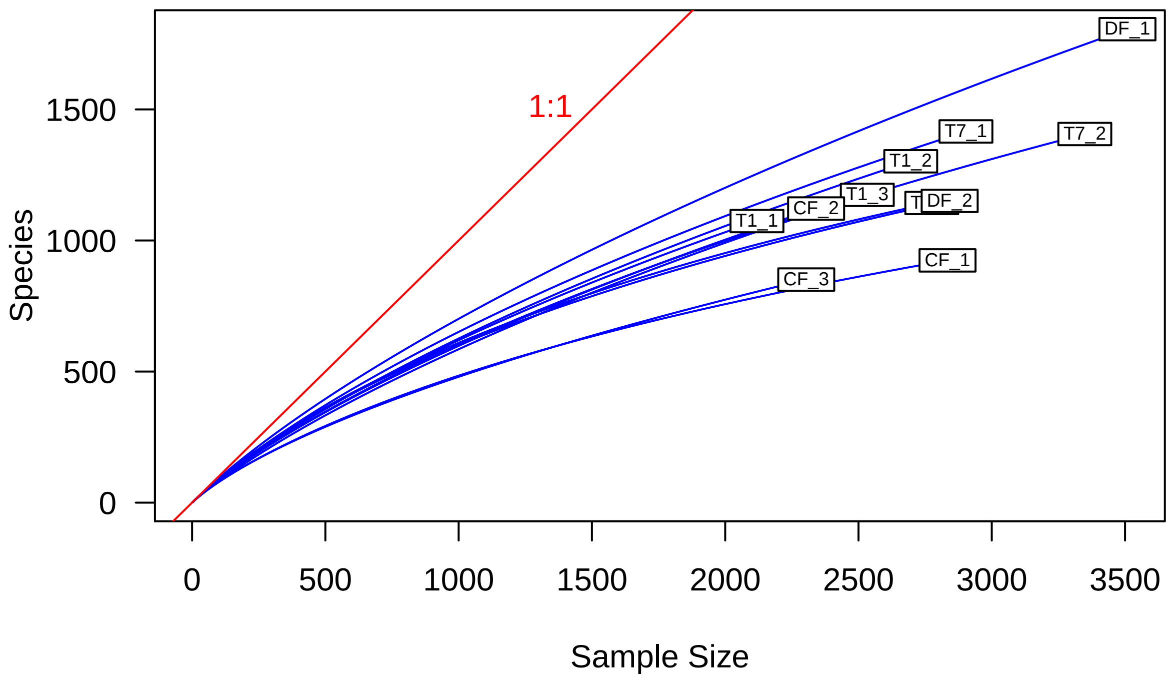
In the R code chunk below, please do the following:

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


S.rarefy <- rarefy(x = soilbac.t, sample = min.N, se = T)
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***: The Chao estimators both use the counts of singletons and doubletons (either for a single site [Chao1] or among sites [Chao2]) as well as observed richness to attempt to extrapolate richness estimates beyond the observed data. They try to use the numbers of rare species to determine how likely it is that there are more unobserved species. ACE does the same thing except a threshold is set for defining what a "rare" species is (i.e. not just singletons or doubletons)

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

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

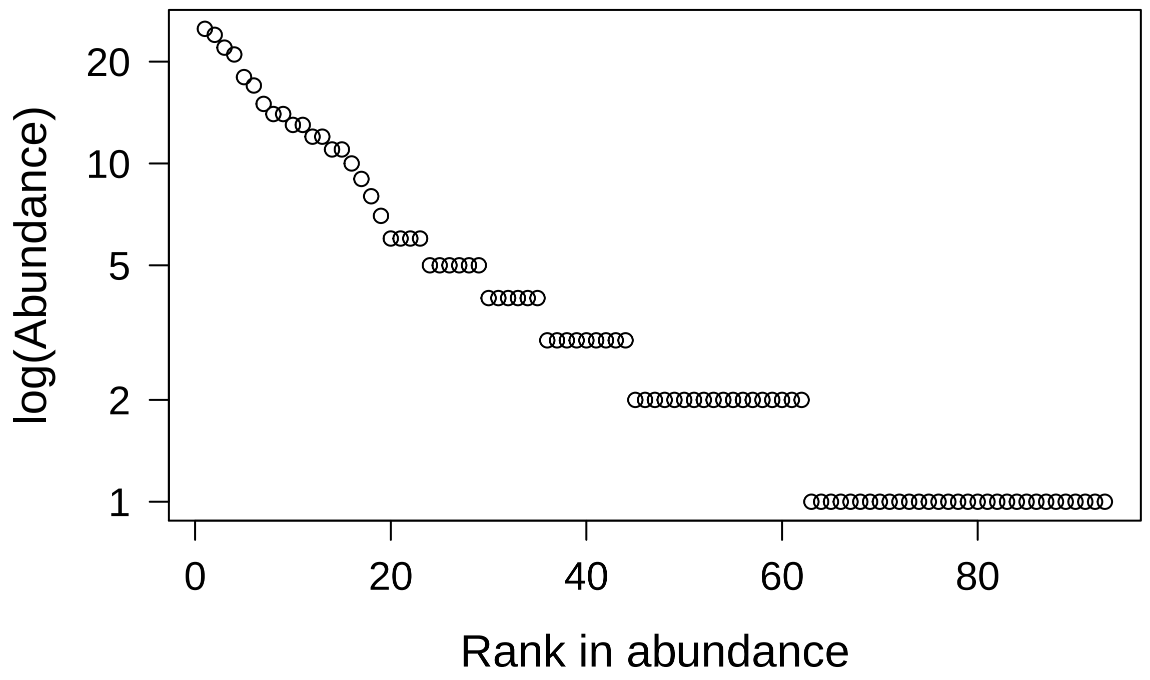
Now, let's examine the RAC for `site1` of the BCI data set.

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)"

```r
plot.new()
site1 <- BCI[1, ]

rac <- RAC(x = site1)
ranks <- as.vector(seq(1, length(rac)))
opar <- par(no.readonly = T)
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***: We use the log-scaled abundances because there is an excess of rare species and visualizing in the log-scale makes reading the RAC easier. If we instead viewed the graph with an untransformed y-axis, we would see that the few very abundant species would take up the majority of the plot. We must keep in mind how we are visualizing the data. In the log-scale, the most abundant species appear less abundant, and the least abundant species appear more abundant than they actually are.

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`.

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


site1 <- BCI[1, ]
print(simpe <- 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}.

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

print(evar <- Evar(site1))
```

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

```r
print(simpe)
```

```
##         1
## 0.4238232
```

```r
print(evar)
```

```
## [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 estimate for Simpson's evenness (0.424) was lower than the estimate for Smith and Wilson's Evenness (0.507) though this difference is not extreme. But since the estimate was different between the two methods, one might infer that the $E_{1/D}$ estimate may have been biased by the most abundant species and that the E_{var} estimate (which involves natural log transformations to reduce bias, among other things) is a better estimate of evenness.

5) INTEGRATING RICHNESS AND EVENNESS: DIVERSITY METRICS

So far, we have introduced two primary aspects of diversity, i.e., richness and evenness. Here, we will use popular indices to estimate diversity, which explicitly incorporate richness and evenness We will write our own diversity functions and compare them against the functions in `vegan`.

Shannon's diversity (a.k.a., Shannon's entropy)

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

1. Provide the code for calculating H' (Shannon's diversity),
2. Compare this estimate with the output of `vegan`'s diversity function using method = "shannon".

```
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)
}


#Both methods give the exact same result.
print(H <- ShanH(site1))
```

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

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

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

```
S <- S.obs(site1)
#Hmax = ln(S)
Hmax <- log(S)
```

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)
}


#We get same results using both methods.
print(D.inv <- 1/SimpD(site1))
```

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

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

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

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

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

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

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

```
J <- H/Hmax
J
```

```
##         1
## 0.8865579
```

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's diversity is not a direct estimate of species evenness, so making a comparison between $E_{H'}$ and E_{var} might not reveal much. Regardless, my estimate for E_{var} at site 1 was 0.5067 and my estimate for $E_{H'}$ at site 1 was 4.018.
> We can use Pielou's evenness index (J) which is defined as H'/max(H') as an estimate of evenness using Shannon's index. Using Pielou's index, the evenness estimate was 0.8865 which is considerably higher than E_{var}.

Fisher's α

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

1. Provide the code for calculating Fisher's α,
2. Calculate Fisher's α 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 $\boldsymbol{\alpha}$ involves the fitting of the parameter α to the observed data rather than the calculation of a diversity metric. In other words, it is an estimated parameter. Fisher's $\boldsymbol{\alpha}$ accounts for sampling error in the data, 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.

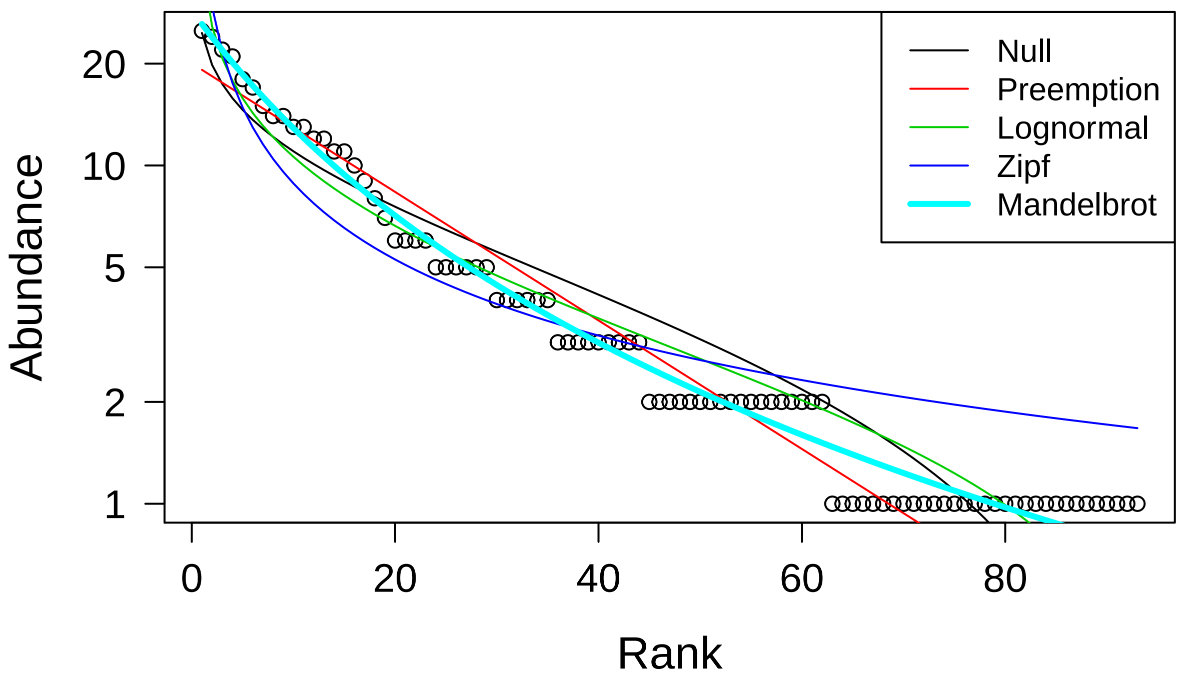
In the R code chunk below, please do the following:

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.

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

```
##
## 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
```

```r
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 Mandelbrot model fits the data for site1 most cloesly. Deviance, AIC, and BIC are all minimized in this model. This is somewhat expected since this model incorporates the most parameters (3). A close second to the Mandelbrot model is the much simpler Preemption model which, despite the higher deviance, has rather similar AIC and BIC values to the Mandelbrot model. For simplicity, we might opt to use the Preemption model.

> ***Answer 9b***: If we go with the Preemption model, we could consider the inference that the community in site1 was assembled successively (e.g one species after another).

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***: That total abundance will decay by a constant amount (α) per rank.

> ***Answer 10b***: Because it involves only a single parameter.

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***: Because we could explain more variation by adding additional parameters to any model. The model may be a better fit with more parameters, but we begin to lose the ability to make inferences from 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.

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

print(D.inv.fi <- 1/SimpD.fi(site1))
```

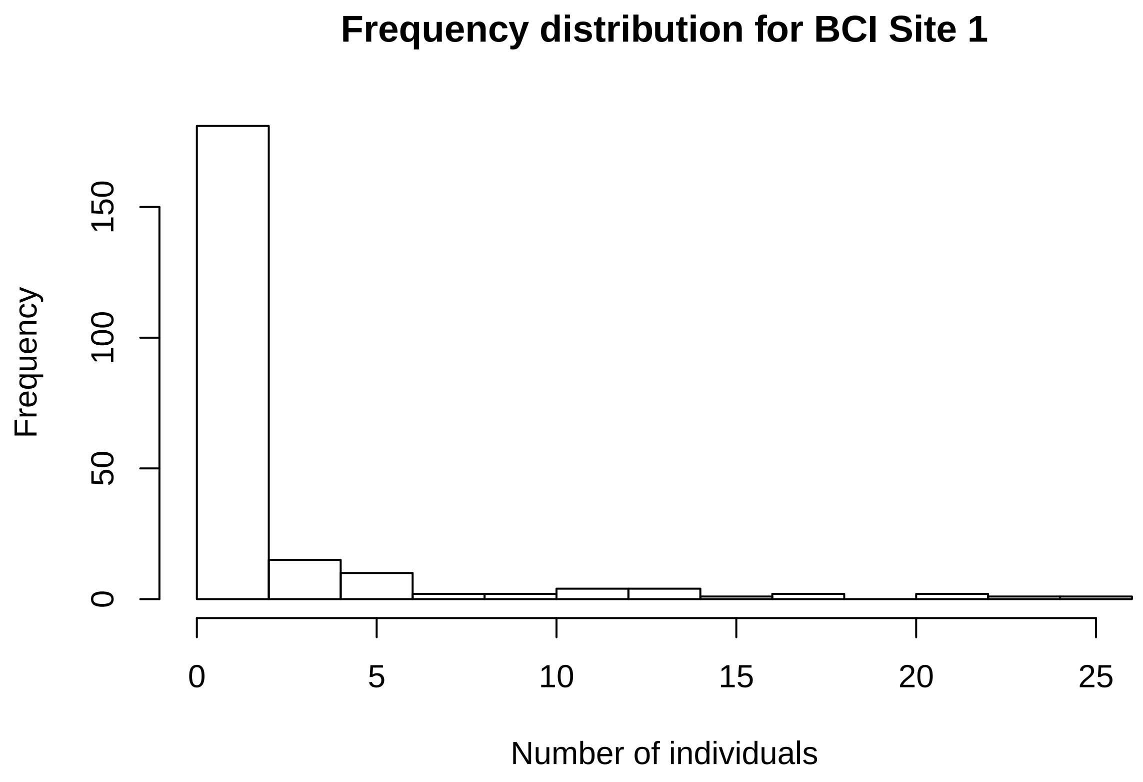
```
## [1] 43.12145
```

```
print(D.sub.fi <- 1-SimpD.fi(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.

```
hist(as.numeric(site1), main="Frequency distribution for BCI Site 1", ylab="Frequency", xlab="Number of
```



In this histogram we see that most species are observed at very low frequency (i.e. most species have between 0 and ~2 individuals).

3. We asked you to find a biodiversity dataset with your partner. This data could be one of your own or it could be something that you obtained from the literature. Load that dataset. How many sites are there? How many species are there in the entire site-by-species matrix? Any other interesting observations based on what you learned this week?

```r
myData <- read.table("data/speciesdata_clean.csv", sep=",", header=T, row.names = 1)
dim(myData)
```

```
## [1] 153 400
```

```r
#There are 153 sites.

#There are 400 species
```

There are 153 sites and 400 species in our dataset.

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)**.