Assignment: Local (alpha) Diversity

Lana Bolin; Z620: Quantitative Biodiversity, Indiana University

23 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. 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/lana/GitHub/QB2019_Bolin/2.Worksheets/5.AlphaDiversity"
```

```
setwd("~/GitHub/QB2019_Bolin/2.Worksheets/5.AlphaDiversity/")
require("vegan")
```

```
## Loading required package: 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 `max.level=0` to show just 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) 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
```

```
# Site 1 has 93 species

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***: Yes they return the same observed richness. The richnesses of the first 4 sites are: 93, 84, 90, 94

Coverage. How Well Did You Sample Your Site?

In the R code chunk below, do the following:

1. Write a function to calculate Good's Coverage, and
2. Use that function to calculate coverage for all sites in the BCI matrix.

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

site1 <- BCI[1, ]

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
```

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***: 0 < C < 1
>
> ***Answer 2b***: Good's Coverage would equal zero, so our coverage would be terrible. We need to sample more!
>
> ***Answer 2c***: ~93%
>
> ***Answer 2d***: Coverage is generally good! Most sites have >90% coverage, with a few in the upper 80's.

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

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

# 2
soilbac.t <- as.data.frame(t(soilbac))

# 3
soilbac1 <- soilbac.t[1, ]

# 4
S.obs(soilbac1)
```

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

```
# 5
C(soilbac1)
```

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

Question 3: Answer the following questions about the soil bacterial dataset.

a. How many sequences did we recover from the sample `soilbac1`, i.e. N?
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***: 1074 sequences
>
> ***Answer 3b***: 64.8% coverage
>
> ***Answer 3c***: Coverage is about 28% lower in the KBS sample than the BCI 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 both `site1` and `soilbac1`.

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

# 2
S.chao2 <- function(site = "", SbyS = "") {
  SbyS = as.data.frame(SbyS)
  x = SbyS[site, ]
  SbyS.pa <- (SbyS > 0) * 1  # makes SbyS presence/absence
  Q1 = sum(colSums(SbyS.pa) == 1)  # number singletons
  Q2 = sum(colSums(SbyS.pa) == 2)  # number doubletons
  S.chao2 = S.obs(x) + (Q1^2)/(2 * Q2)
  return(S.chao2)
}

# 3
S.ace <- function(x = "", thresh = 10) {
  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 (proportion non-singleton rare individuals)
  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)
}

# 4
S.chao1(site1)
```

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

```
S.chao2(1, BCI)
```

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

```
S.ace(site1)

S.chao1(soilbac1)
```

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

```
S.chao2(1, soilbac.t)
```

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

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

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.

```
# 1
soilbac.S <- S.obs(soilbac.t)
soilbac.S
```

```
## T1_1 T1_2 T1_3 T7_1 T7_2 T7_3 DF_1 DF_2 CF_1 CF_2 CF_3
## 1074 1302 1174 1416 1406 1143 1806 1151  924 1122  851
```

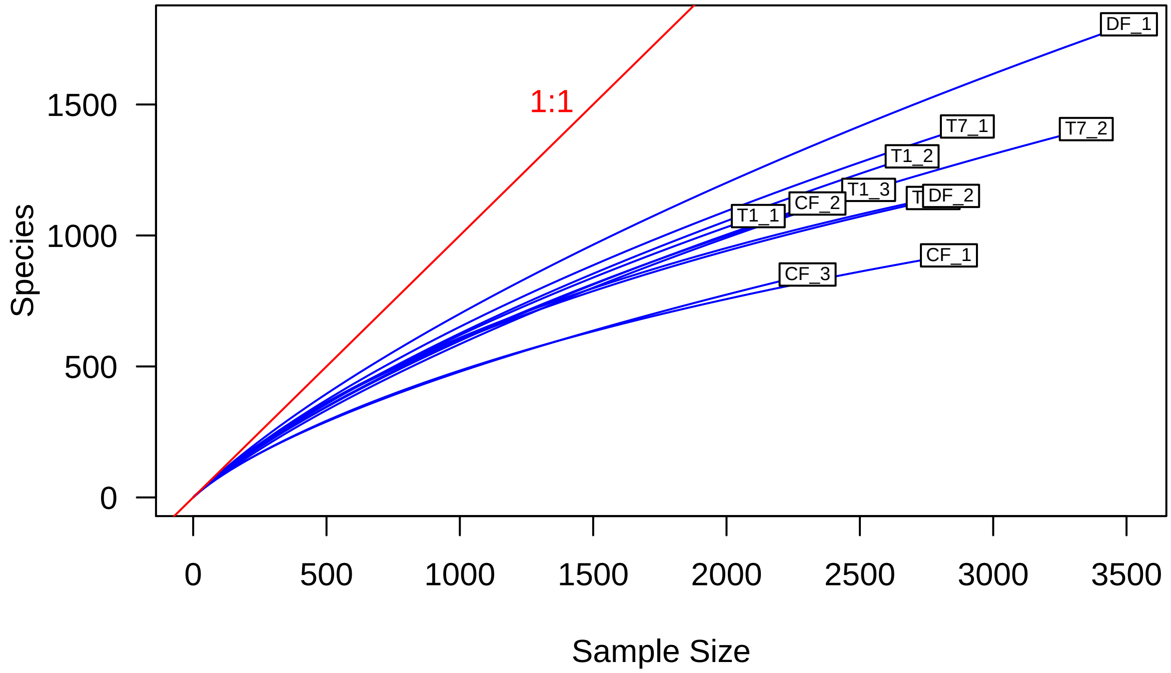
```
# 2
min.N <- min(rowSums(soilbac.t))
min.N
```

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

```
# 3
S.rarefy <- rarefy(x = soilbac.t, sample = min.N, se = TRUE)

# 4
rarecurve(x = soilbac.t,
          step = 20,
          col = "blue",
          cex = 0.6,
          las = 1)

# 5
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***: ACE looks at the abundance of other rare species besides singletons and doubletons (i.e. taxa that have fewer than 10 individuals).

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

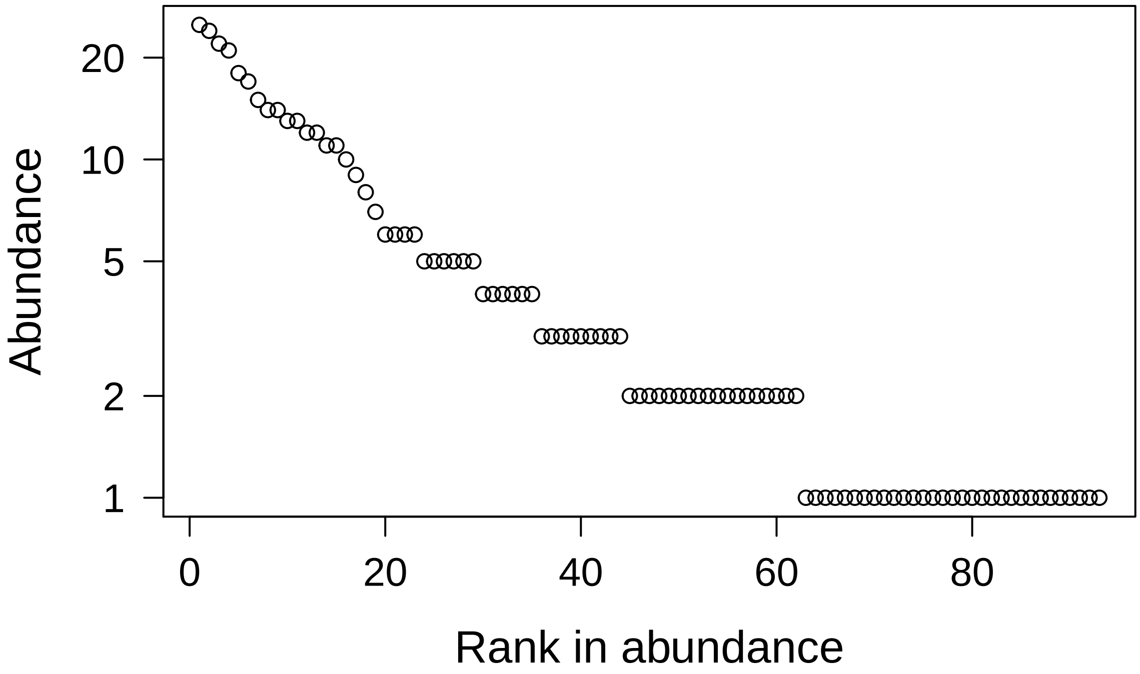
```
RAC <- function(x = "") {
  x = as.vector(x)
  x.ab = x[x > 0]
  x.ab.ranked = x.ab[order(x.ab, decreasing = TRUE)]
  return(x.ab.ranked)
}
```

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)"

```
rac <- RAC(x = site1)
ranks <- as.vector(seq(1, length(rac)))
opar <- par(no.readonly = TRUE)           # saves default plot parameters
par(mar = c(5.1, 5.1, 4.1, 2.1))         # new settings for par
plot(ranks, log(rac),
     type = "p",
     axes = F,                            # plots w/o axes
     xlab = "Rank in abundance", ylab = "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)))   # adds log-scale y-axis
```



```
par <- opar          # reset plotting parameters
```

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 evenness is much lower than it appears on a log scale, because low rank sites have orders of magnitude higher abundance than high rank sites

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`.

```
# 1
SimpE <- function(x = "") {
  S <- S.obs(x)
  x = as.data.frame(x)
  D <- diversity(x, "inv")
  E <- (D)/S
  return(E)
}
```

```
# 2
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}.

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

# 2
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***: Smith and Wilson's evenness index is higher than Simpson's evenness by ~0.08, so we can infer that bias in the most abundant species was artificially reducing our evenness estimate.

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".

```
# 1
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, "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”.

```
# 1
SimpD <- function(x) {
  D = 0
  N = sum(x)
  for (n_i in x) {
    D = D + (n_i^2)/(N^2)
  }
  return(D)
}

# 2
1/(SimpD(site1))
```

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

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

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

```
# 3
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***: I don’t see Shannon’s Diversity “H”’ called $E_{H'}$ in the handout, but I’m assuming these are synonyms and will answer this questions under that assumption. $E_{H'}$ = 4.02, and E_{var} = 0.51. They don’t agree because they are scaled differently (?) - E_{var} is scaled to be between zero and one, while $E_{H'}$ is not.

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.

```
rac <- as.vector(site1[site1 > 0])
fisher.alpha(rac)
```

```
## [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 an estimate of diversity, rather than simply a diversity metric. It takes into account sampling errror, while $E_{H'}$ 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:

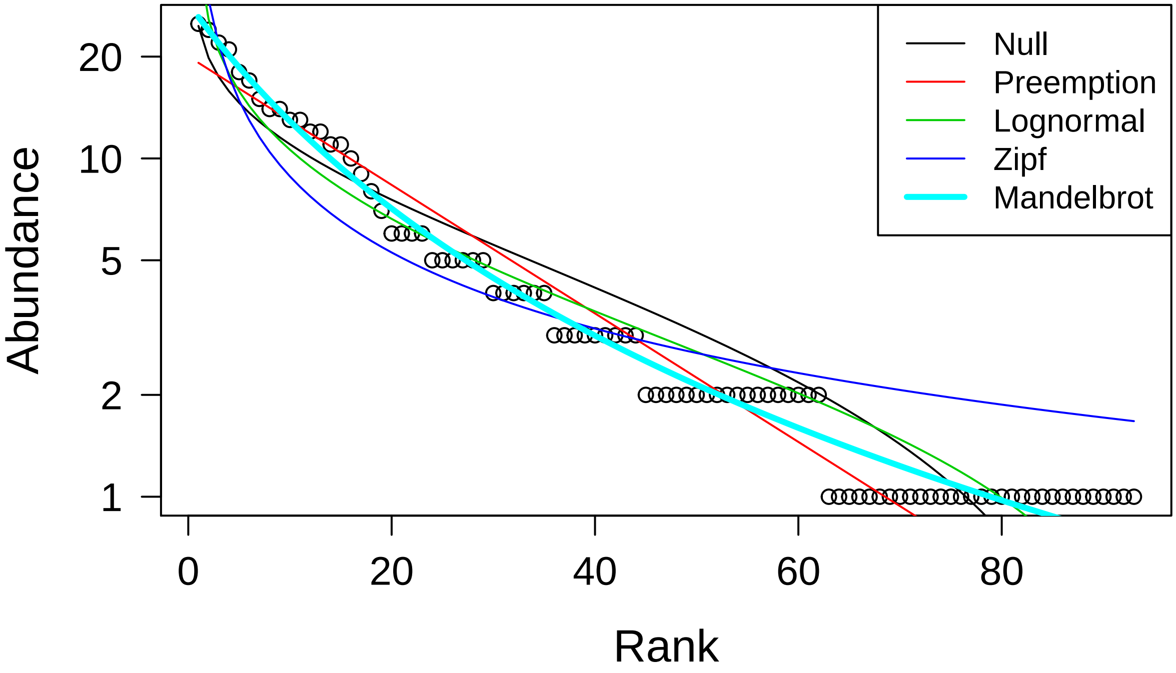
1. Use the `radfit()` function in the `vegan` package to fit the predictions of various species abundance models to the RAC of `site1` in BCI,
2. Display the results of the `radfit()` function, and
3. Plot the results of the `radfit()` function using the code provided in the handout.

```
# 1
RACresults <- radfit(site1)

# 2
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
```

```
# 3
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 best - it has the lowest AIC and BIC, and the curve seems to match our data better than the other curves. ***Answer 9b***: I'm not sure how we can do that with only the information we have…

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 individuals use a proportion of the total resources, rather than a set quantity of resources. ***Answer 10b***: Because the y-axis is log-scale.

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***: As you add parameters your model is always going to fit your data better, but it may not become more predictive (which is often what we want to use models for). So penalizing additional parameters is a way to avoid overfitting.

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.fin = 0
  N = sum(x)
  for (n_i in x) {
    D.fin = D.fin + (n_i * n_i - 1)/(N * (N - 1))
  }
  return(D.fin)
}

SimpD.fin(site1)
```

```
## [1] 0.02430389
```

```
1/(SimpD.fin(site1))
```

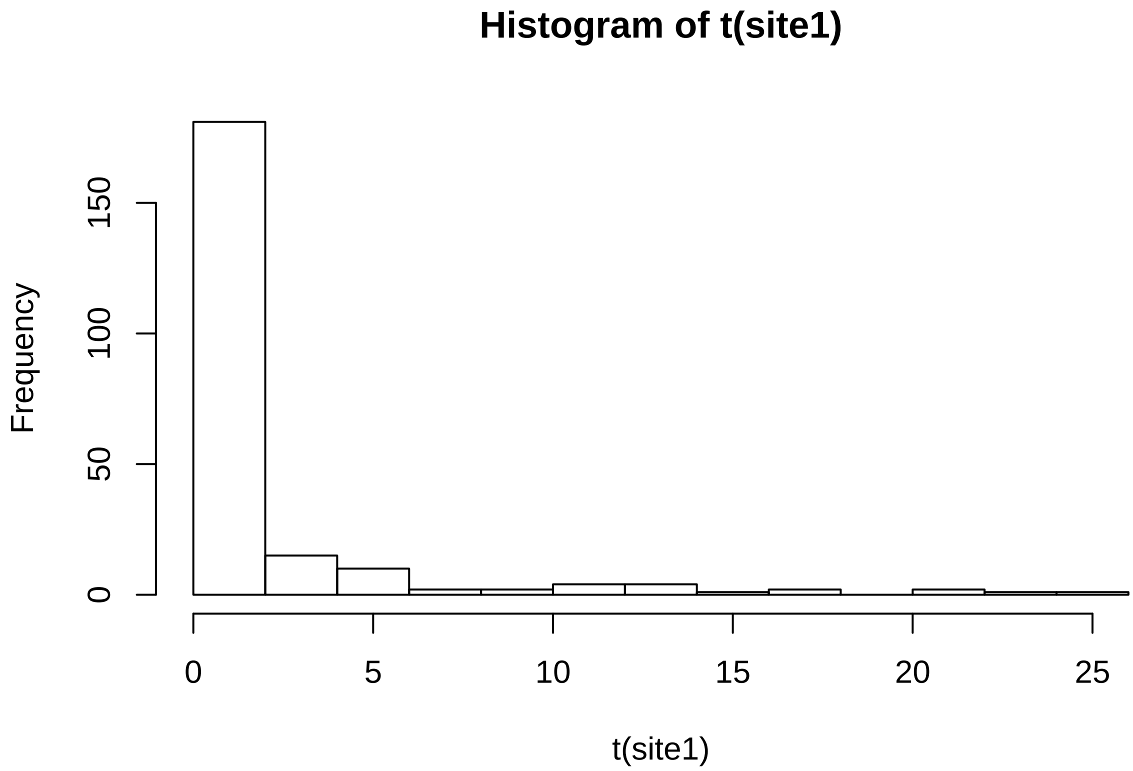
```
## [1] 41.14567
```

```
1-(SimpD.fin(site1))
```

```
## [1] 0.9756961
```

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(t(site1))
```



There is a huge spike in rare taxa (1 or 2 individuals), and a low frequency of more common taxa. In other words, the distribution is right skewed.

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?

```
crawley <- read.csv("data/Crawley_data.csv", header = TRUE, row.names = 1)
crawley <- crawley[, 1:20]
str(crawley)
```

```
## 'data.frame':    302 obs. of  20 variables:
##  $ Athenaeum                 : int  0 1 0 0 0 0 1 0 0 0 ...
##  $ Bauman.Park               : int  1 0 0 0 0 0 1 0 0 1 ...
##  $ Center.for.Inquiry        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chinqpin.Oak.Park         : int  0 1 0 0 0 0 1 0 0 0 ...
##  $ Community.Caring...Sharing: int  1 0 0 0 0 0 1 0 0 0 ...
##  $ Cottage.Home              : int  1 0 0 0 1 0 0 1 0 0 ...
##  $ Englewood                 : int  0 1 0 1 0 1 0 0 0 0 ...
##  $ Fletcher.Gateway          : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Fletcher.Park             : int  1 0 0 0 0 0 0 1 0 0 ...
##  $ Historic.Meridian.Park    : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Jonathan.Jennings         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Lynhurst                  : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Merrill.Street            : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Paramount.Charter         : int  1 0 0 0 0 0 1 0 0 0 ...
##  $ Purpose.Park              : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Ransom.Place              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Skiles.Test               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ University.Park           : int  0 1 0 0 0 0 1 0 0 0 ...
##  $ Westminster               : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Willard.Park              : int  0 0 0 0 0 0 0 0 0 1 ...
```

```
sum(S.obs(crawley) > 0, na.rm = T)
```

```
## [1] 270
```

> There are 20 sites. There are 302 species in the matrix, but only 270 of them were observed at at least one site. These are presence/absence data, rather than counts, so I don't believe we have the tools yet to deal with this data structure…

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)**.