5. Worksheet: Alpha Diversity

Xiaotian Zhou; Z620: Quantitative Biodiversity, Indiana University

2021 - 04 - 08

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).

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

```
## [1] "C:/Users/sherry/Documents/GitHub/QB2021_Zhou/2.Worksheets/5.AlphaDiversity"
```

```
setwd("C:/Users/sherry/Documents/GitHub/QB2021_Zhou/2.Worksheets/5.AlphaDiversity")
require("vegan")
```

```
## Loading required package: vegan
```

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

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

```
## This is vegan 2.5-7
```

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
```

```
obs_richness = c()
for (i in 1:4){
  obs_richness[i] <- S.obs(BCI[i,])
}
obs_richness
```

```
## [1] 93 84 90 94
```

Question 1: Does `specnumber()` from `vegan` return the same value for observed richness in `site1` as our function `S.obs`? What is the species richness of the first four sites (i.e., rows) of the BCI matrix?

> ***Answer 1***: Yes, they return the same value (S = 93). The observed species richness of the first four sites are 93, 84, 90, 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 - (rowSums(x == 1)/rowSums(x))
}
coverage_BCI = c()
for (i in 1:nrow(BCI)){
  coverage_BCI[i] <- C(BCI[i,])
}
coverage_BCI
```

```
##  [1] 0.9308036 0.9287356 0.9200864 0.9468504 0.9287129 0.9174757 0.9326923
##  [8] 0.9443155 0.9095355 0.9275362 0.9152120 0.9071038 0.9242054 0.9132420
## [15] 0.9350649 0.9267735 0.8950131 0.9193084 0.8891455 0.9114219 0.8946078
## [22] 0.9066986 0.8705882 0.9030612 0.9095023 0.9115479 0.9088729 0.9198966
## [29] 0.8983516 0.9221053 0.9382423 0.9411765 0.9220183 0.9239374 0.9267887
## [36] 0.9186047 0.9379310 0.9306488 0.9268868 0.9386503 0.8880597 0.9299517
## [43] 0.9140049 0.9168704 0.9234234 0.9348837 0.8847059 0.9228916 0.9086651
## [50] 0.9143519
```

```
mean(coverage_BCI)
```

```
## [1] 0.9182232
```

```
min(coverage_BCI)
```

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

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: The range of values that the Good's coverage can generate is [0,1].

Answer 2b: If n_i equaled N, the Good's Coverage value would be 0, which means all the species only be observed once and there would be a large sampling bias that more species and individuals really exist but go undetected.

Answer 2c: Around 7% of taxa in `site1` was represented by singletons.

Answer 2d: The BCI dataset is a well-sampled one with 91.8% of mean coveage value and 87.1% of the minimum.

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.t <- as.data.frame(t(soilbac))
soilbac1 <- soilbac.t[1,]
S.obs(soilbac1)
```

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

```
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: N = 2119

Answer 3b: S = 1074

Answer 3c: The coverage of the KBS sample (%C_{site1}$ = 64.8%) is lower than that of 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 `site1` and `soilbac1`.

```
#calculate chao1
S.chao1 <- function(x = ""){
  S.obs(x) + (sum(x == 1)^2) / (2*sum(x == 2))
}
print(S.chao1(soilbac1))
```

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

```
print(S.chao1(BCI[1,]))
```

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

```
#calculate chao2
S.chao2 <- function(site = "", SbyS = ""){
  SbyS = as.data.frame(SbyS)
  x = SbyS[site,]
  SbyS.pa <- (SbyS > 0 ) * 1 #convert to presence/absence
  Q1 = sum(colSums(SbyS.pa) == 1)
  Q2 = sum(colSums(SbyS.pa) == 2)
  S.chao2 = S.obs(x) + (Q1^2)/(2*Q2)
  return(S.chao2)
}
print(S.chao2(1,soilbac.t))
```

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

```
print(S.chao2(1,BCI[,]))
```

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

```
#calculate ACE
S.ace <- function(x = "", thresh = 10){
  x <- x[x>0] #exclude zero-abundance data
  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){
    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)
}
print(S.ace(soilbac1))
```

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

```
print(S.ace(BCI[1,]))
```

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

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?

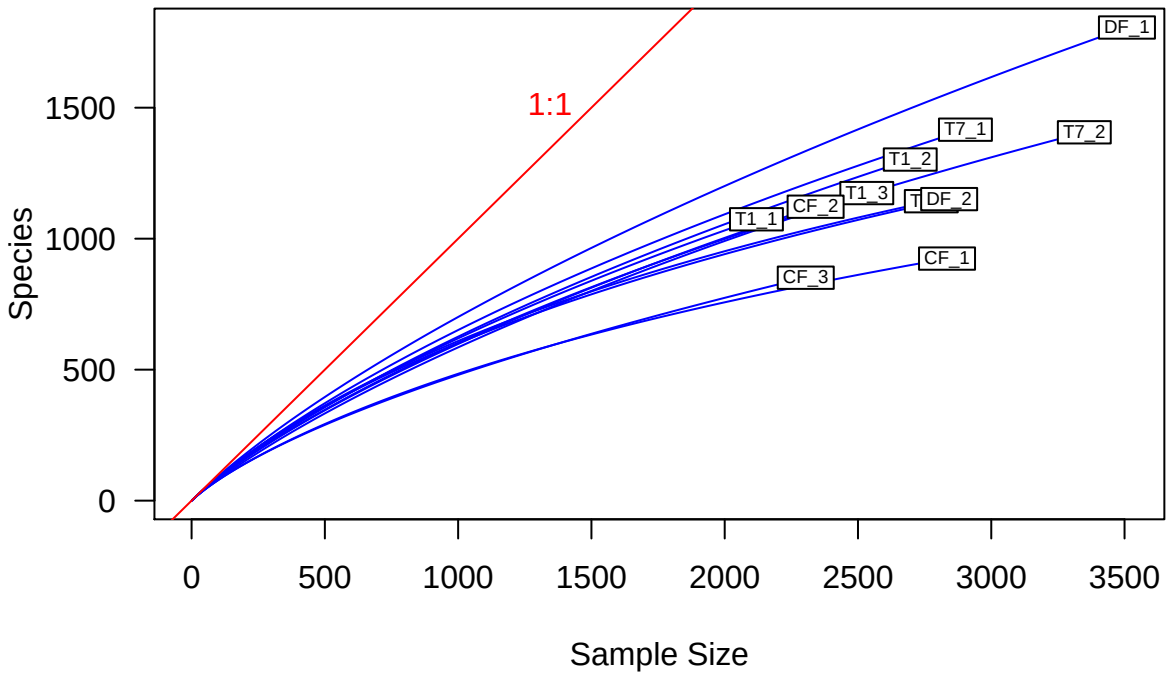
> ***Answer 4***: The difference between ACE and the Chao estimators lies in the sensitivity towards rarity. The question is that how rare the species are so that they can be viewed as the underestimated part which is derived from sampling bias: singletons, doubletons or those with indivdual number smaller than ten. The idea behind the Chao estimators is that if a community is being sampled, and rare species (singletons) are still being discovered, there is likely still more rare species not found; as soon as all species have been recovered at least twice (doubletons), there is likely no more species to be found; while the ACE requires the rare threshold up to ten. Obviously they give inconsitent results for different reasons, which depends on attributes of the dataset. For **BCI** dataset with good Good's coverage, sparse site-by-species matrix and relative low species abundance, I would like to use the Chao estimator, specifically the Chao1 because it is abundance data rather than presence/absence data; for the **soil bacterial** dataset, I also choose the Chao1 because using ACE would make the majority of sampled species ignored.

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 = 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")
```



##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

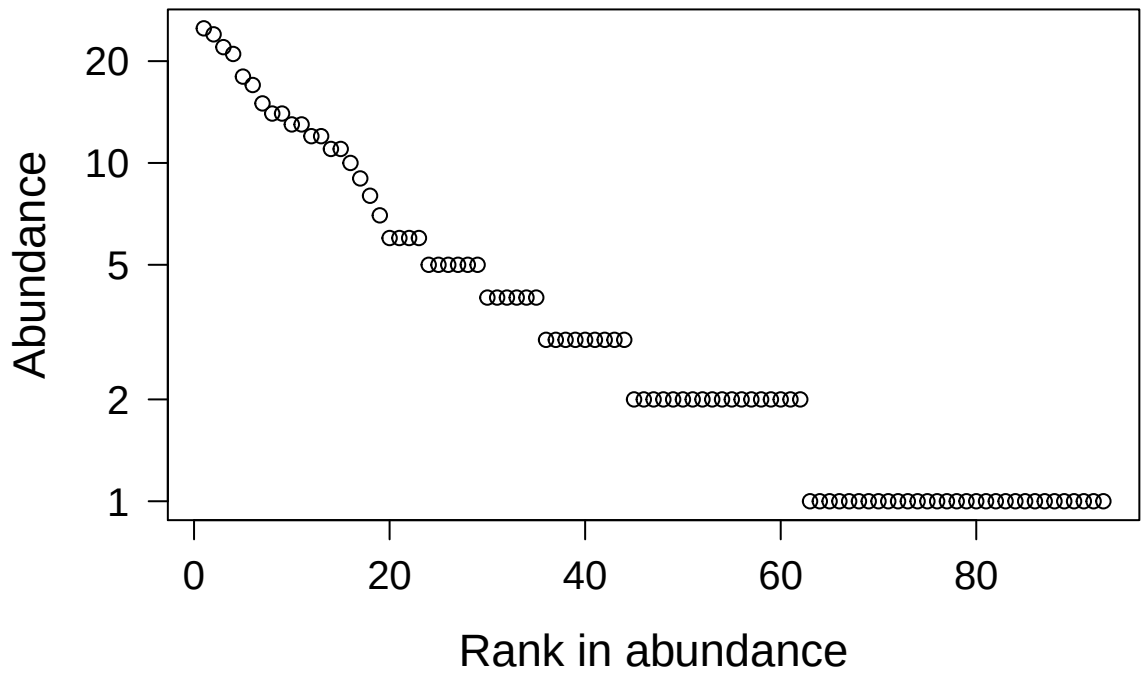
```
RAC <- function(x = ""){
  x = as.vector(x)
  x.ab = x[x > 0]
  x.ab.ranked = x.ab[order(x.ab, decreasing = TRUE)]
return(x.ab.ranked)
}
```

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)))
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() # Manually adds border
axis(side = 1, labels = T, cex.axis = 1.25) # Manually adds X-axis
axis(side = 2, las = 1, cex.axis = 1.25, # Manually adds Log-Scaled Y-axis
  labels = c(1, 2, 5, 10, 20), at = log(c(1, 2, 5, 10, 20)))
```




```
par <- opar  # Resets 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 log-scaled axis helps decrease bias towards the most abundant species.

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***: No. For site1 of BCI, the Smith and Wilson's evenness value (0.51) is higher than Simpson's evenness value (0.42), because the former use log-transformed abundances.

##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)
}
D.inv <- 1/SimpD(site1)
D.sub <- 1-SimpD(site1)
print(D.inv)
```

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

```
print(D.sub)
```

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

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

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

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

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

Fisher's α

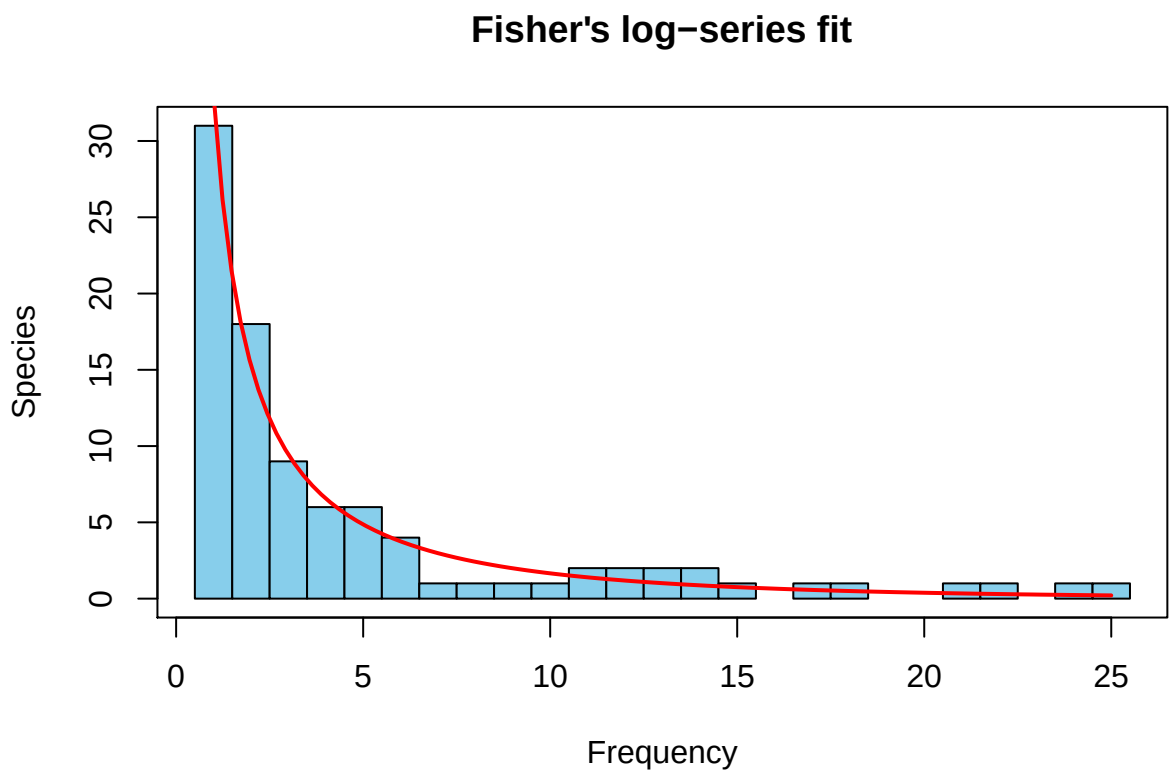
In the R code chunk below, please do the following:

1. Provide the code for calculating Fisher's α,
2. Calculate Fisher's α for `site1` of BCI.

```r
# calculate Fisher's alpha
logseries.alpha <- function(y = ""){
  S <- rowSums(y>0)*1 # total number of species
  N <- rowSums(y)  # total number of individuals
  f <- function(x, S, N) x*log(1+N/x)-S # equation of log-series model to solve 'x'(fisher alpha)
  # solve the equation with search interval from 1 to 100 and desired accuracy of 0.0001
  result <- uniroot(f,c(1,100), N=N, S=S, tol = 0.0001)
  return(result$root)
}
print(logseries.alpha(site1))
```

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

```r
# Fisher fit and calculate alpha
rac_fisher <- as.vector(site1[site1 > 0])
plot(fisherfit(rac_fisher),xlab = "Frequency", ylab = "Species", bar.col = "skyblue",line.col = "red", l
title("Fisher's log-series fit")
```




```r
fisher.alpha(rac_fisher)
```

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

Question 7: How is Fisher's α different from $E_{H'}$ and E_{var}? What does Fisher's α take into account that $E_{H'}$ and E_{var} do not?

Answer 7: The Fisher's α is an estimating diversity parameter instead of just calculating a diversity metric, which takes the whole aggregate entity into account rather than every single individual, because Fisher's α is actually based only on the number of species S and number of individuals.

##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.

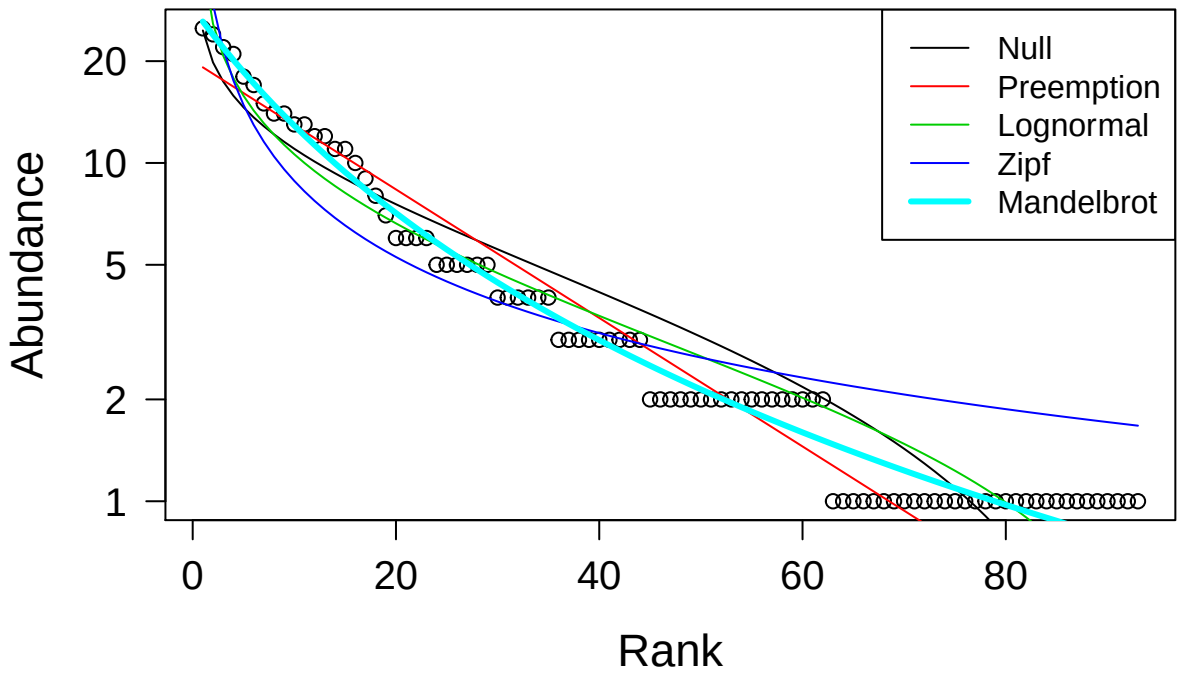
In the R code chunk below, please do the following:

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 8: Answer the following questions about the rank abundance curves: a) Based on the output of `radfit()` and plotting above, discuss which model best fits our rank-abundance curve for `site1`? b) Can we make any inferences about the forces, processes, and/or mechanisms influencing the structure of our system, e.g., an ecological community?

> ***Answer 8a***: The Mandelbrot model makes the best fits with lowest deviance, AIC and BIC value. ***Answer 8b***: The Zipf-Mandelbrot model predicts a power-law like distribution with fewer abundant species and more rare species, indicating an uneven, non-stochastic background of community assembly process. Evenness is the key factor to shaping community structure.

Question 9: Answer the following questions about the preemption model: a. What does the preemption model assume about the relationship between total abundance (N) and total resources that can be preempted? b. Why does the niche preemption model look like a straight line in the RAD plot?

> ***Answer 10a***: The preemption model assumes a fixed proportion between total resources and total number of richness (S), but no relatioship between total resources and total abundance (N). ***Answer 10b***: The niche preemption model look like a straight line in the RAD plot because it is linear in the logarithmic scale.

Question 10: Why is it important to account for the number of parameters a model uses when judging how well it explains a given set of data?

> ***Answer 11***: More parameters mean more unexplainable mutual effects and make the underlying mechanism obscurer.

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.

```
SimpD2 <- function(x = ""){
  D = 0
  N = sum(x)
  for (n_i in x){
    D = D + n_i * (n_i - 1) /( N * (N - 1))
  }
  return(D)
}
D2.inv <- 1 / SimpD2(site1)
D2.sub <- 1 - SimpD2(site1)
print(D2.inv)
```

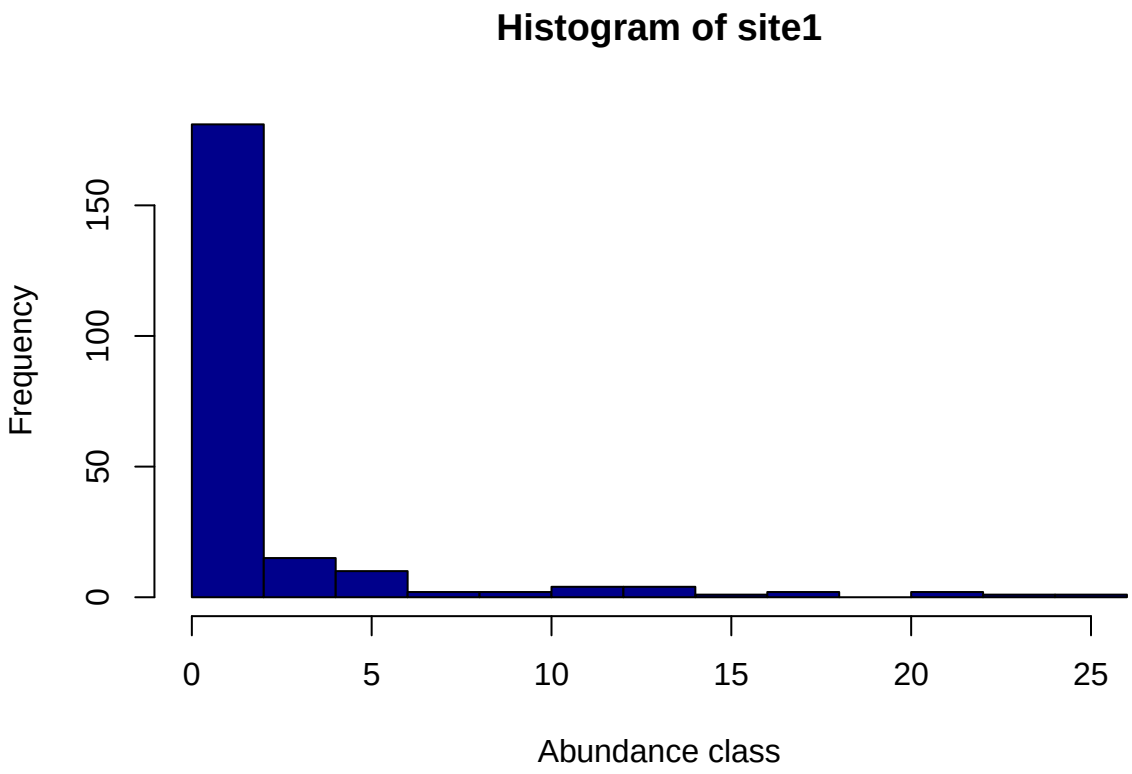
```
## [1] 43.12145
```

```
print(D2.sub)
```

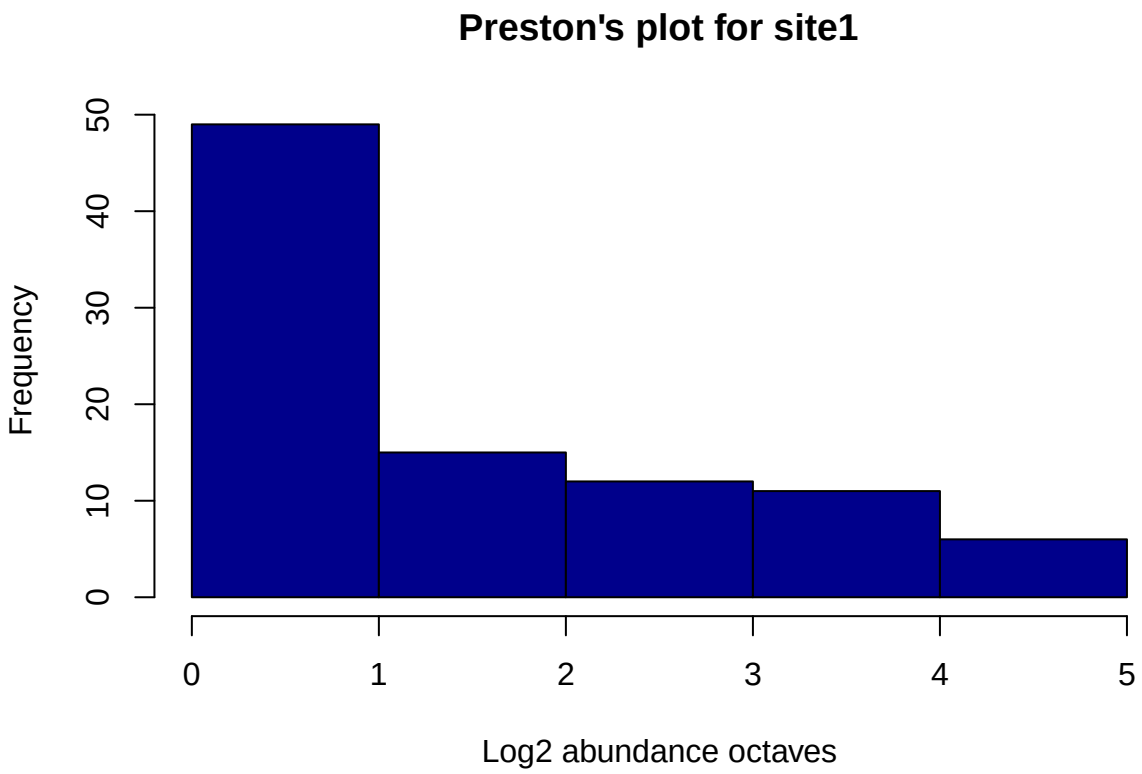
```
## [1] 0.9768097
```

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.

```
# histogram of SAD
hist(t(site1), xlab = "Abundance class", col = "darkblue", main = "Histogram of site1")
```




```
# histogram plot illustrating Preston octaves by transforming species abundance using log2
hist(log(t(site1),2), breaks = 5, xlab = "Log2 abundance octaves",
     col = "darkblue", main = "Preston's plot for site1")
```



Description: We can observe the frequency distribution with or without abundance log-transformed are both rightskewed, where there are many rare species and few abundant ones.

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?

```
## diversity estimation
# load original viral-cluster-sites matrix
vcs.o <-read.csv("C:/Users/sherry/Documents/GitHub/QBVirus/data/vcs.csv", row.names = 1)
dim(vcs.o)
```

```
## [1] 868  91
```

```
vcs <- as.data.frame(t(vcs.o))
# remove unclustered populations
vcs.rm.uncluster <- vcs.o[-868,]
# sites X species
vcs.rm.uncluster.t <- as.data.frame(t(vcs.rm.uncluster))
# calculate number of species (S), Shannon index (H) and Smith & Wilson's Evenness (E) of each site
vcs.S <- c()
vcs.E <- c()
```

```
vcs.H <- c()
for (i in 1:91) {
  vcs.S[i] = S.obs(vcs[i,])
  vcs.E[i] = Evar(vcs[i,])
  vcs.H[i] = ShanH(vcs[i,])
}
vcs.div <- cbind(vcs.S, vcs.E, vcs.H)
write.csv("C:/Users/sherry/Documents/GitHub/QBVirus/data/vcs-div.csv")
```

```
## "","x"
## "1","C:/Users/sherry/Documents/GitHub/QBVirus/data/vcs-div.csv"
```

```
## stacked barplot for top 20 viral clusters
require(ggplot2)
```

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

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
require(reshape2)
```

```
## Loading required package: reshape2
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
#convert to relative abundance table
vcs.rel <- vcs/rowSums(vcs)
top <- names(head(sort(colSums(vcs),decreasing = T),20))
top <- as.vector(top)
vcs.rel2 <- vcs.rel
colnames(vcs.rel2)[!colnames(vcs.rel2)%in%top] <- "Others"
```

```
others <- rowSums(cbind(vcs.rel2[colnames(vcs.rel2)=="Others"]))
vcs.top <- cbind(vcs.rel2[colnames(vcs.rel2)!="Others"],others)
vcs.top <- vcs.top[,order(colSums(vcs.top))] # ranking
#load group data
env.class.geo.div <- read.table("C:/Users/sherry/Documents/GitHub/QBVirus/data/env-geo-div-class.txt", s
vcs.top2 <- as.data.frame(cbind(vcs.top,env.class.geo.div$Layer,env.class.geo.div$Region))
vcs.top2$sample <- rownames(vcs.top2)
vcs.top20 <- melt(vcs.top2,ID="names")
```

```
## Using env.class.geo.div$Layer, env.class.geo.div$Region, sample as id variables
```

```
colnames(vcs.top20)[names(vcs.top20)=="variable"]<-"Taxa"
# group by layer (zonation)
vcs.top.layer <- aggregate(vcs.top2[,1:21], by = list(Layer = vcs.top2$`env.class.geo.div$Layer`),FUN =
vcs.top.layer$sample <- rownames(vcs.top.layer)
vcs.group.layer <- melt(vcs.top.layer, ID="names")
```

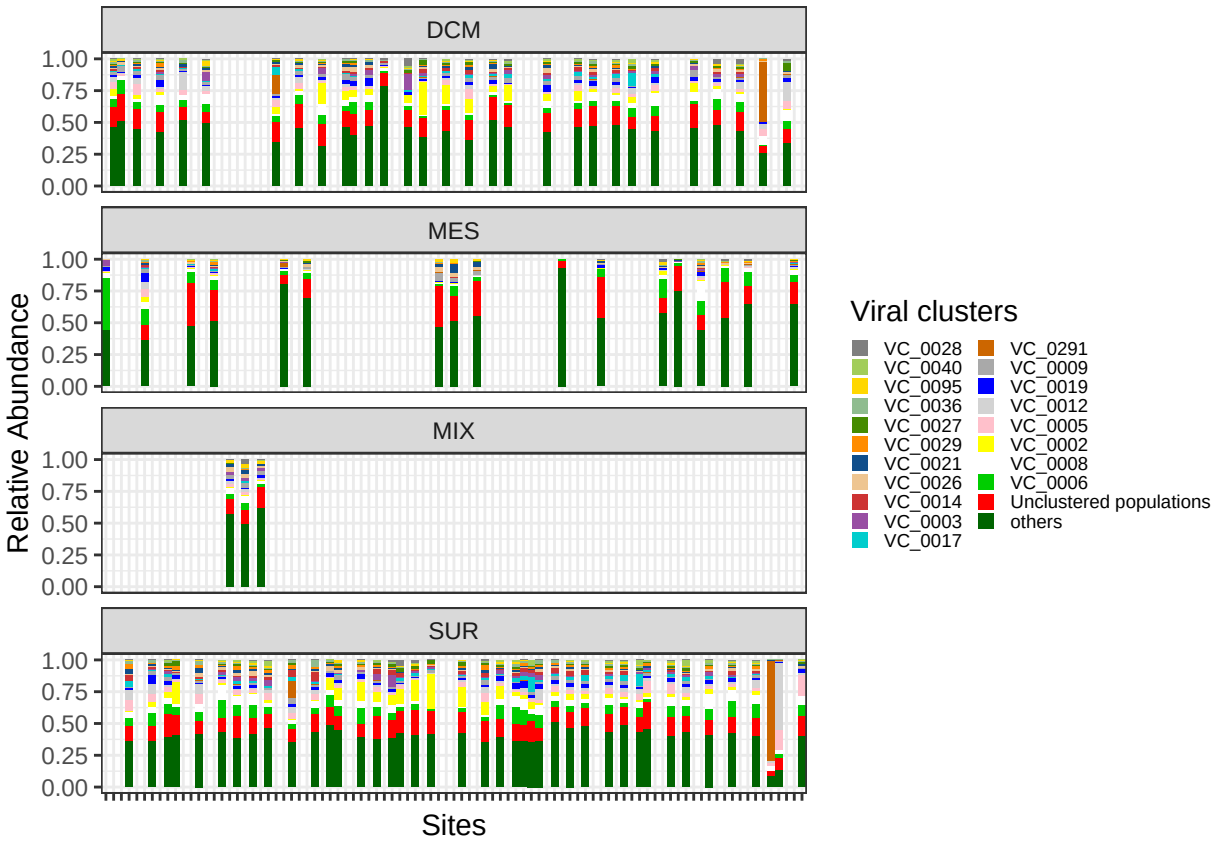
```
## Using Layer, sample as id variables
```

```
colnames(vcs.group.layer)[names(vcs.group.layer)=="variable"]<-"Taxa"
vcs.group.layer$Layer <- factor(vcs.group.layer$Layer, levels = c("SUR","DCM","MIX","MES")) #set order
# group by region
vcs.top.region <- aggregate(vcs.top2[,1:21], by = list(Layer = vcs.top2$`env.class.geo.div$Region`),FUN
vcs.top.region$sample <- rownames(vcs.top.region)
vcs.group.region <- melt(vcs.top.region, ID="names")
```

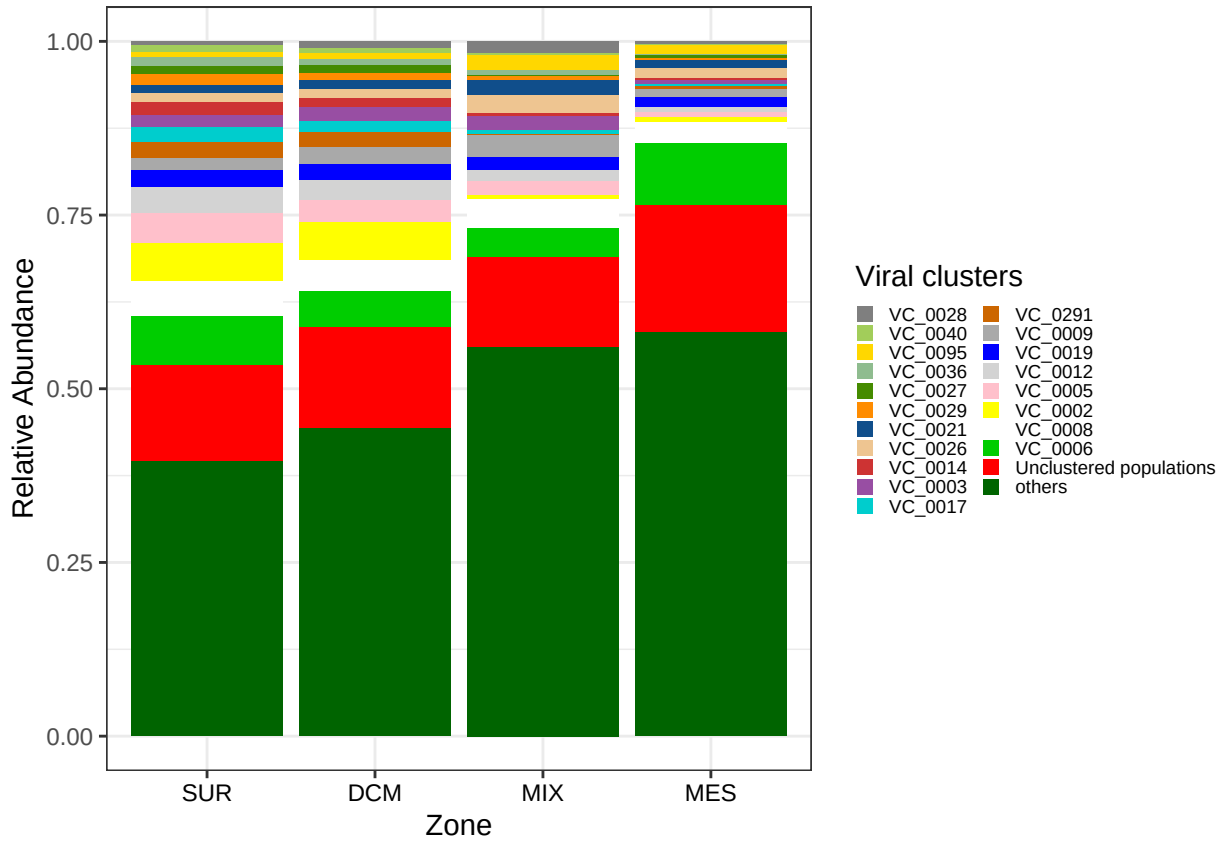
```
## Using Layer, sample as id variables
```

```
colnames(vcs.group.region)[names(vcs.group.region)=="variable"]<-"Taxa"
```

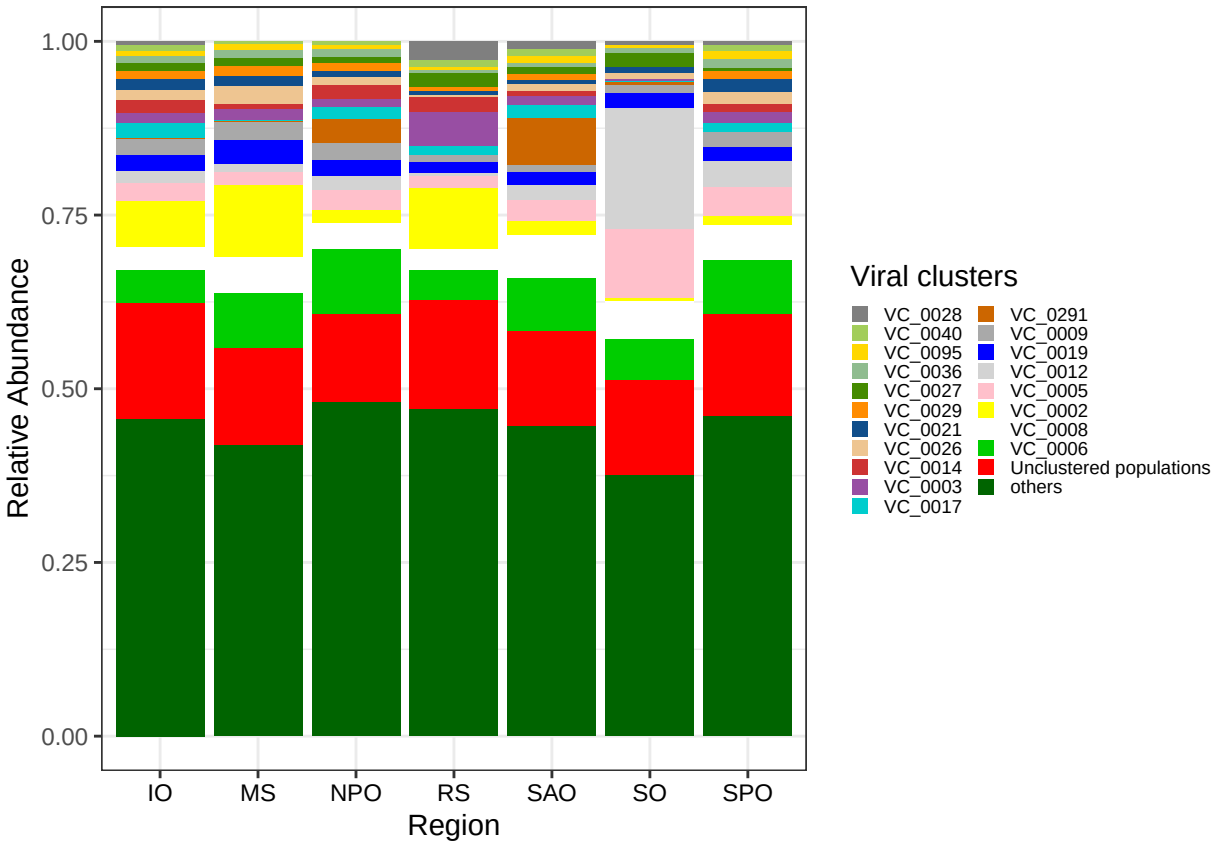
```
#stacked barplot for top 20 viral clusters
colors<-c("grey50","darkolivegreen3","gold","darkseagreen","chartreuse4","darkorange","dodgerblue4","bu
ggplot(vcs.top20,aes(x = sample, y = value, fill = Taxa))+
  geom_bar(position = "fill", stat = "identity", width = 1)+
  theme_bw()+
  scale_fill_manual(values=colors)+
  facet_wrap(~env.class.geo.div$Layer,nrow = 4)+
  labs(x = "Sites",y = "Relative Abundance", fill = "Viral clusters")+
  theme(axis.text.x = element_text(size = 0, color = "transparent"),
        legend.text = element_text(size = 7))+
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5))
```




```
# group by water layer
ggplot(vcs.group.layer,aes(x = Layer, y = value, fill = Taxa))+
  geom_bar(position = "fill", stat = "identity")+
  theme_bw()+
  scale_fill_manual(values=colors)+
  labs(x = "Zone",y = "Relative Abundance", fill = "Viral clusters")+
  theme(axis.text.x = element_text(size = 9, color = "black"),
        legend.text = element_text(size = 7))+
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5))
```




```
# group by region
ggplot(vcs.group.region,aes(x = Layer, y = value, fill = Taxa))+
  geom_bar(position = "fill", stat = "identity")+
  theme_bw()+
  scale_fill_manual(values=colors)+
  labs(x = "Region",y = "Relative Abundance", fill = "Viral clusters")+
  theme(axis.text.x = element_text(size = 9, color = "black"),
        legend.text = element_text(size = 7))+
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5))
```




```
# seek relationships with diversity and water depth
summary(lm(env.class.geo.div$ShannonH ~ env.class.geo.div$Depth))
```

```
##
## Call:
## lm(formula = env.class.geo.div$ShannonH ~ env.class.geo.div$Depth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6401 -0.0946  0.1002  0.3175  0.8970
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.8214247  0.0687869  55.555  < 2e-16 ***
## env.class.geo.div$Depth -0.0008932  0.0002381  -3.752 0.000313 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5694 on 89 degrees of freedom
## Multiple R-squared:  0.1365, Adjusted R-squared:  0.1268
## F-statistic: 14.07 on 1 and 89 DF,  p-value: 0.0003125
```

```
summary(lm(env.class.geo.div$Richness ~ env.class.geo.div$Depth))
```

```
##
```

```
## Call:
## lm(formula = env.class.geo.div$Richness ~ env.class.geo.div$Depth)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -159.428  -36.404   -2.052   37.572  191.185
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              221.09826    8.64048  25.589  < 2e-16 ***
## env.class.geo.div$Depth  -0.13410    0.02991  -4.484 2.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71.53 on 89 degrees of freedom
## Multiple R-squared:  0.1843, Adjusted R-squared:  0.1751
## F-statistic: 20.11 on 1 and 89 DF,  p-value: 2.175e-05
```

```
summary(lm(env.class.geo.div$Evenness ~ env.class.geo.div$Depth))
```

```
##
## Call:
## lm(formula = env.class.geo.div$Evenness ~ env.class.geo.div$Depth)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.143448 -0.021850 -0.001426  0.015939  0.124838
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.267e-01  4.535e-03   49.99  < 2e-16 ***
## env.class.geo.div$Depth 8.367e-05  1.570e-05    5.33 7.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03755 on 89 degrees of freedom
## Multiple R-squared:  0.242,  Adjusted R-squared:  0.2334
## F-statistic: 28.41 on 1 and 89 DF,  p-value: 7.355e-07
```

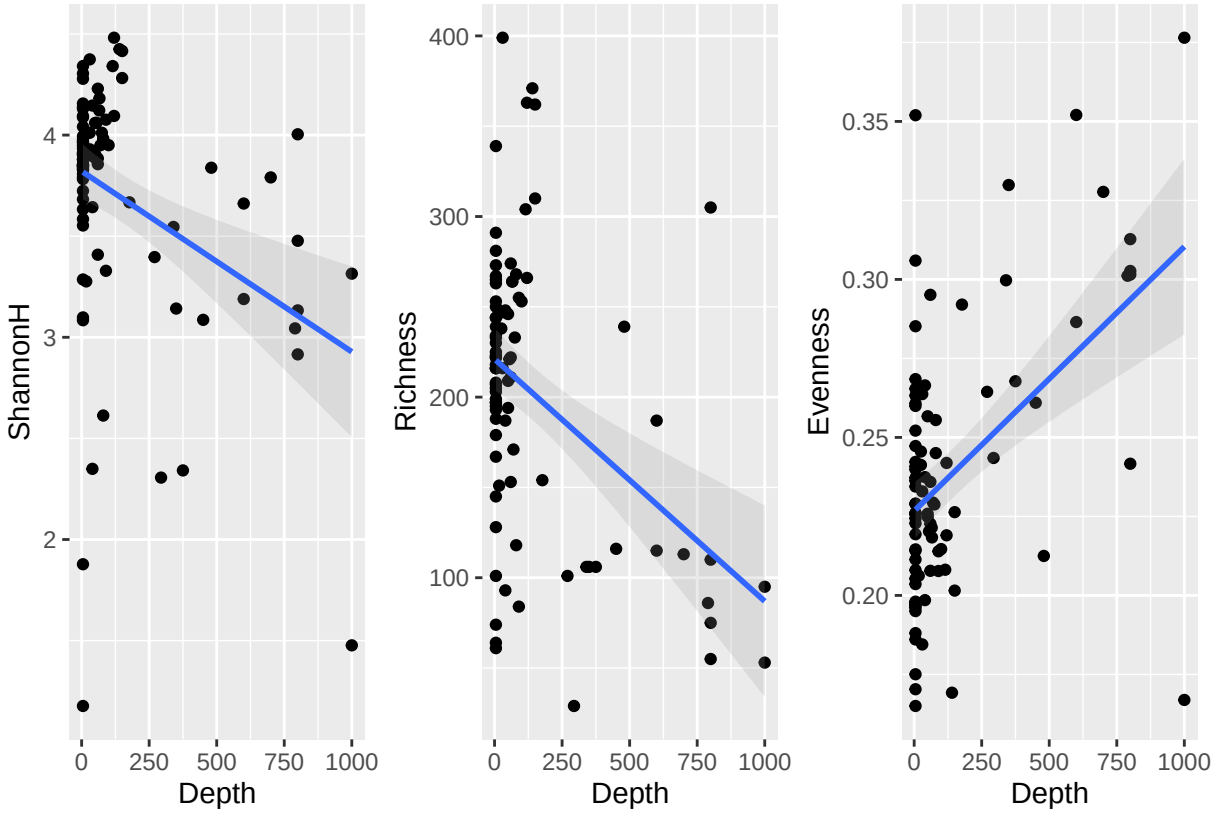
```
require(patchwork)
```

```
## Loading required package: patchwork
```

```
HD <- ggplot(env.class.geo.div, aes(x = Depth, y = ShannonH))+
  geom_point()+
  geom_smooth(method = "lm", alpha = 0.2)
SD <- ggplot(env.class.geo.div, aes(x = Depth, y = Richness))+
  geom_point()+
  geom_smooth(method = "lm", alpha = 0.2)
ED <- ggplot(env.class.geo.div, aes(x = Depth, y = Evenness))+
  geom_point()+
  geom_smooth(method = "lm", alpha = 0.2)
HD + SD + ED
```

```
## `geom_smooth()` using formula 'y ~ x'

## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



Description: I classify 91 sites(samples) using two grouping methods: water zonations and sea areas/regions. [**zonation**] SUR, surface water layer targeting a discrete depth at the top of the photic zone; DCM, deep chlorophyll maximum layer targeting a discrete depth within the photic zone; MIX, marine epipelagic wind mixed layer targeting a discrete depth within the photic zone, bottom of the selected environmental feature; MES, marine water layer within the mesopelagic zone. [**region**] IO, Indian Ocean; MS, Mediterranean Sea; NPO, North Pacific Ocean; RS, Red Sea; SAO, South Atlantic Ocean; SO, Southern Ocean; SPO, South Pacific Ocean We can observe the viral community structure varies with seawater layer. Specifically, the percentage of Top20 viral clusters decreases with depth. Among different sea regions, viral structure in the Southern Ocean is different from the rest. Furthermore, there are significant relationships between alpha diversity and water depth.

SUBMITTING YOUR ASSIGNMENT

Use Knitr to create a PDF of your completed 5.AlphaDiversity_Worksheet.Rmd document, push it to GitHub, and create a pull request. Please make sure your updated repo include both the pdf and RMarkdown files.

Unless otherwise noted, this assignment is due on **Wednesday, April 7th, 2021 at 12:00 PM (noon)**.