6. Worksheet: Diversity Sampling

Emmi Mueller; Z620: Quantitative Biodiversity, Indiana University

29 January, 2019

OVERVIEW

In this worksheet, you will use the jelly bean site-by-species matrix generated from **6. Diversity Sampling**. Along with tools outlined in the **5. Local (alpha) Diversity** and **7. Control Structures** handouts, you will develop questions, visualize data, and test hypotheses related to sampling effects and its effect on estimates of within-sample biodiversity.

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. Refer to previous handout to help with developing of questions and writing of code.
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 `6.DiversitySampling_Worskheet.Rmd` and the PDF output of `Knitr` (`DiversitySampling_Worskheet.pdf`).

1) Group brainstorming

With your team partner and perhaps other students in the class, spend 15 minutes or so brainstorming questions, code, "fantasy figures", and statistical tests that could be used to test questions with the class's data represented in the site-by-species matrix that you have generated.

2) Code

Use the space below for code that is being used to analyze your data and test your hypotheses. Create one (and only one, although it can have multiple panels) *publication quality* figure. Make sure to annotate your code using # symbols so others (including instructors) understand what you have done and why you have done it.

```
#Clear workspace, set working directory and load packages
rm(list=ls())
getwd()
```

```
## [1] "C:/Users/emmim/GitHub/QB2019_Mueller/2.Worksheets/6.DiversitySampling"
```

```
setwd("~/GitHub/QB2019_Mueller/2.Worksheets/6.DiversitySampling")
require("vegan")
```

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

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

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

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

```
require("ggplot2")
```

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

```
#Load Jelly Bean dataset, declare groups and remove group row from table

jellysource <- as.data.frame(read.table("./JellyBeans.Source.Reduced.txt", sep ="\t", header = TRUE))
jellysource <- as.data.frame(t(jellysource), stringsAsFactors = FALSE)
typeof(jellysource["Count",])
```

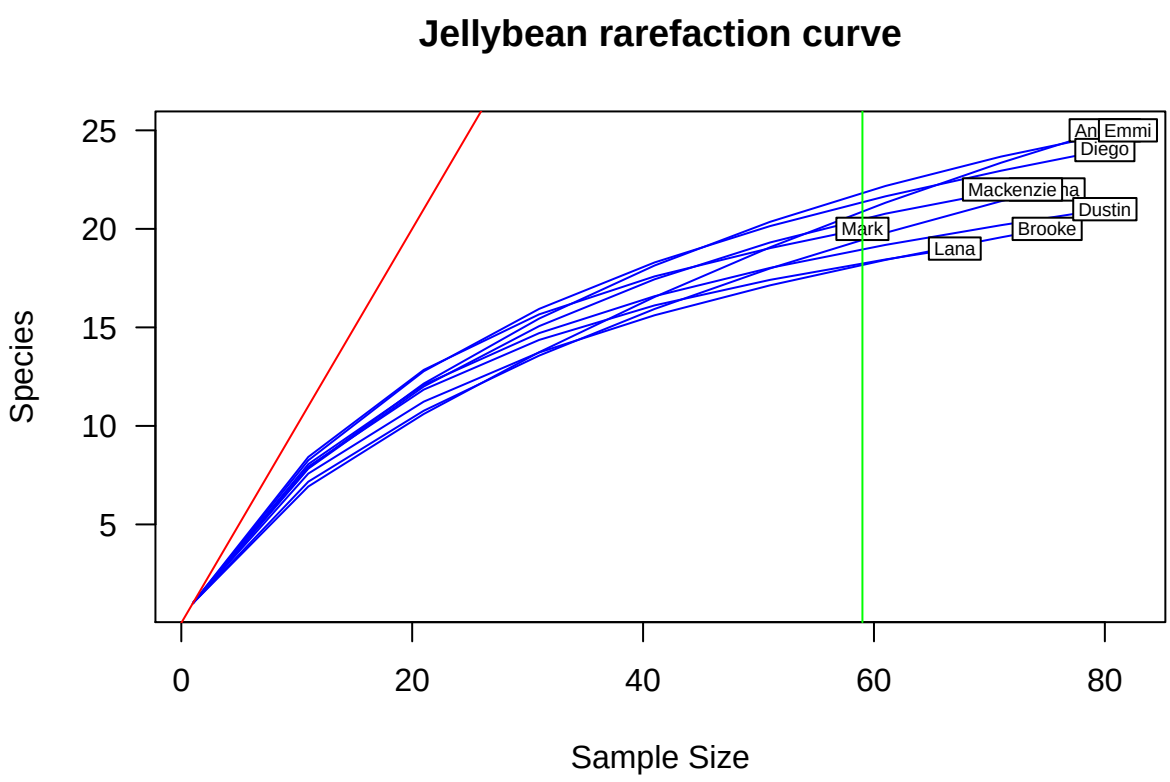
```
## [1] "list"
```

```
jellysource["Count",] <- as.numeric(jellysource["Count",])
typeof(jellysource["Count",])
```

```
## [1] "list"
```

```
jellysource <- as.data.frame(jellysource)
jellybean <- read.table("./JellyBeans_site.txt", sep = "\t", header = TRUE, row.names = 1)
jellybeangroups <- as.data.frame(jellybean["Group"])
jellybean <- jellybean[,2:29]
```

```
#Create function S.obs and C to calculate observed abundance and coverage
S.obs <- function(x = ""){
   rowSums(x > 0) * 1
}
C <- function(x = ""){
  1 - (rowSums(x == 1) / rowSums(x))
}

#Rarefaction curve
min.N <- min(rowSums(jellybean))
jelly.rarefy <- rarefy(x = jellybean, sample = min.N, se = TRUE)
rarecurve(x = jellybean, step = 10, col = "blue", cex = 0.6, las = 1, main = "Jellybean rarefaction cur
abline(0,1, col = 'red')
abline(v = min.N, col = 'green')
```




```
#Rarefy sample and generate Group samples
jelly.rarefied <- rrarefy(x = jellybean, sample = min.N)
x <- rep(0, ncol(jelly.rarefied))
names <- row.names(jelly.rarefied)
jelly.rarefied <- rbind(x, jelly.rarefied)
jelly.rarefied <- rbind(x, jelly.rarefied)
row.names(jelly.rarefied) <- c("GroupA", "GroupB", names)
for(column in colnames(jelly.rarefied)){
  for (row in rownames(jelly.rarefied)){
    if (row == "GroupA" || row == "GroupB"){}
    else if (jellybeangroups[row,] == "A"){
          jelly.rarefied["GroupA", column] <- jelly.rarefied["GroupA", column] + jelly.rarefied[row, col
    }
    else if (jellybeangroups[row,] == "B"){
          jelly.rarefied["GroupB", column] <- jelly.rarefied["GroupB", column] + jelly.rarefied[row, col
    }
  }
}

#Print rarefied sample individuals, species richness, and coverage
jelly.rarefied <- as.data.frame(jelly.rarefied)
for (row in row.names(jelly.rarefied)){
  if (row == "GroupA"){
    cat("\n", row, "\n")
    cat("Individuals:", as.numeric(rowSums(jelly.rarefied[row,])), " ")
    cat("Species:",as.numeric(S.obs(jelly.rarefied[row,])), " ")
```

```
    cat("Coverage:",as.numeric(C(jelly.rarefied[row,])),"\n")
  }
  else if (row != "GroupB" && jellybeangroups[row, ] == "A"){
    print(row)
    cat("Individuals:", as.numeric(rowSums(jelly.rarefied[row,])), " ")
    cat("Species:",as.numeric(S.obs(jelly.rarefied[row,])), " ")
    cat("Coverage:",as.numeric(C(jelly.rarefied[row,])),"\n")
  }
}
```

```
##
##  GroupA
## Individuals: 295  Species: 28  Coverage: 0.9898305
## [1] "Andrea"
## Individuals: 59  Species: 19  Coverage: 0.8135593
## [1] "Brianna"
## Individuals: 59  Species: 19  Coverage: 0.8474576
## [1] "Brooke"
## Individuals: 59  Species: 18  Coverage: 0.8813559
## [1] "Emmi"
## Individuals: 59  Species: 22  Coverage: 0.779661
## [1] "Mackenzie"
## Individuals: 59  Species: 20  Coverage: 0.8644068
```

```
for (row in row.names(jelly.rarefied)){
  if (row == "GroupB"){
    cat("\n",row, "\n")
    cat("Individuals:", as.numeric(rowSums(jelly.rarefied[row,])), " ")
    cat("Species:",as.numeric(S.obs(jelly.rarefied[row,])), " ")
    cat("Coverage:",as.numeric(C(jelly.rarefied[row,])),"\n")
  }
  else if (row != "GroupA" && jellybeangroups[row,] == "B"){
    print(row)
    cat("Individuals:", as.numeric(rowSums(jelly.rarefied[row,])), " ")
    cat("Species:",as.numeric(S.obs(jelly.rarefied[row,])), " ")
    cat("Coverage:",as.numeric(C(jelly.rarefied[row,])),"\n")
  }
}
```

```
##
##  GroupB
## Individuals: 236  Species: 27  Coverage: 0.9915254
## [1] "Diego"
## Individuals: 59  Species: 21  Coverage: 0.8813559
## [1] "Dustin"
## Individuals: 59  Species: 18  Coverage: 0.9322034
## [1] "Lana"
## Individuals: 59  Species: 18  Coverage: 0.9152542
## [1] "Mark"
## Individuals: 59  Species: 20  Coverage: 0.8813559
```

```
#Create ACE estimator function
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)
}

#Estimate ACE for the Source, Group A, and Group B

Ace <- c(S.ace(as.numeric(jellysource["Count",])),S.ace(jellyrarefied["GroupA",]), S.ace(jellyrarefie
Ace <- as.data.frame(Ace, row.names = c("Source","GroupA", "GroupB"))
print(Ace)
```

```
##             Ace
## Source 26.00000
## GroupA 32.02012
## GroupB 29.59811
```

```
#Test RAD model fits to determine best fit
RACresultsA <- radfit(jellyrarefied["GroupA",])
cat("Group A RAD model fits", "\n")
```

```
## Group A RAD model fits
```

```
RACresultsA
```

```
##
## RAD models, family poisson
## No. of species 28, total abundance 295
##
##            par1     par2    par3    Deviance AIC     BIC
## Null                                 59.604  161.709 161.709
## Preemption  0.14055                  63.075  167.180 168.513
## Lognormal   1.7316   1.1666          32.535  138.640 141.305
## Zipf        0.25966 -1.0139          32.630  138.735 141.399
## Mandelbrot  0.75886 -1.3894  1.3664  27.479  135.584 139.581
```

```
RACresultsB <- radfit(jellyrarefied["GroupB",])
cat("\n", "Group B RAD model fits", "\n")
```

```
##
##  Group B RAD model fits
```

```
RACresultsB
```

```
##
## RAD models, family poisson
## No. of species 27, total abundance 236
##
```

```
##            par1     par2     par3    Deviance AIC      BIC
## Null                                   8.8044 107.7346 107.7346
## Preemption  0.11221                   13.2071 114.1373 115.4331
## Lognormal   1.7913   0.90027           3.2850 106.2152 108.8068
## Zipf        0.19275 -0.80894          11.0271 113.9573 116.5489
## Mandelbrot  1.4825  -1.465    3.6388   5.5910 110.5212 114.4087
```

```
RACresultsfull <- radfit(as.numeric(jellysource["Count",]))
cat("\n", "Source RAD model fits", "\n")
```

```
##
##  Source RAD model fits
```

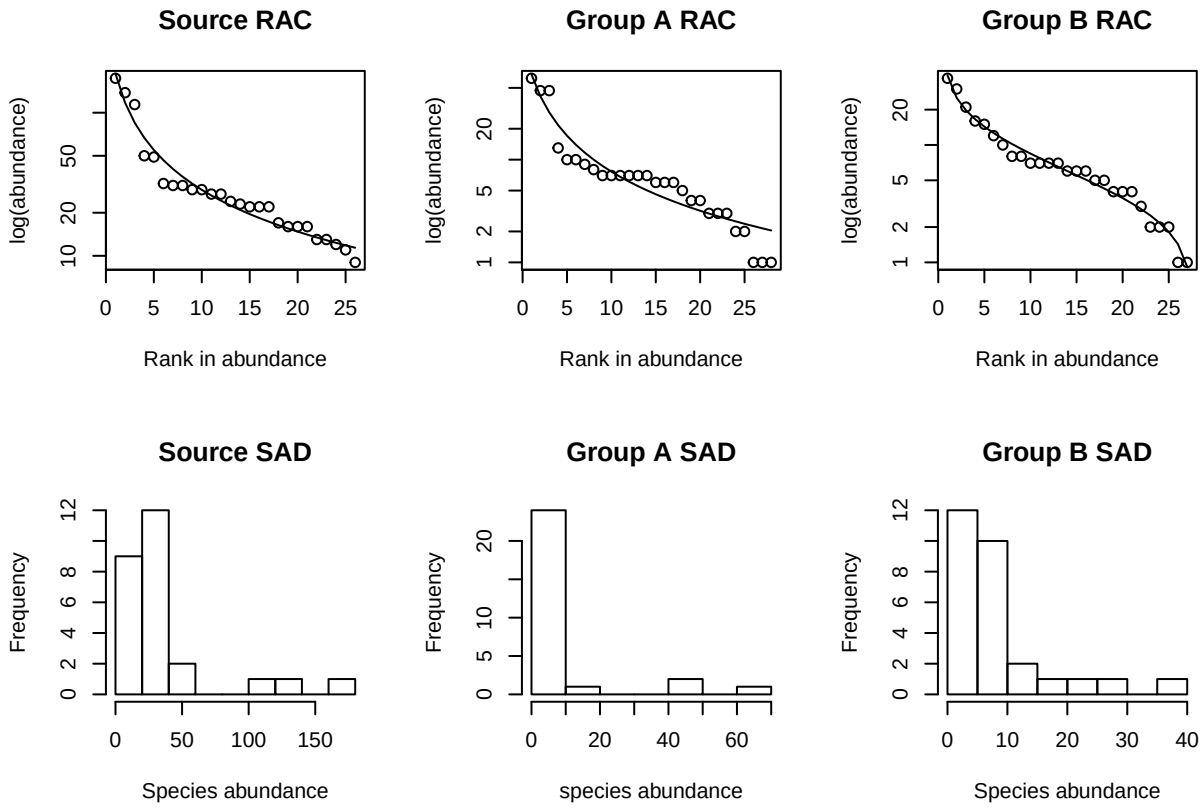
```
RACresultsfull
```

```
##
## RAD models, family poisson
## No. of species 26, total abundance 967
##
##            par1     par2     par3    Deviance AIC     BIC
## Null                                  132.735  265.955 265.955
## Preemption  0.11395                   134.732  269.952 271.210
## Lognormal   3.2358   0.90574          64.712  201.932 204.448
## Zipf        0.20875 -0.85112          35.213  172.433 174.949
## Mandelbrot  0.33614 -1.0201   0.7163  30.738  169.958 173.732
```

```
#Generate figure of RAD and SAD for Source, Group A and Group B

par(mfrow = c(2,3), main = "SAD and RAD for Vicariance event source and sampled populations")
```

```
## Warning in par(mfrow = c(2, 3), main = "SAD and RAD for Vicariance event
## source and sampled populations"): "main" is not a graphical parameter
```

```
plot(rad.zipfbrot(as.numeric(jellysource["Count",])), main = "Source RAC", xlab = "Rank in abundance", y
plot(rad.zipfbrot(jelly.rarefied["GroupA",]), main = "Group A RAC", xlab = "Rank in abundance", ylab = "
plot(rad.lognormal(jelly.rarefied["GroupB",]), main = "Group B RAC", xlab = "Rank in abundance", ylab =


jelly.full.df <-unlist(jellysource["Count",])
jelly.full.df <- as.vector(as.numeric(jelly.full.df))

hist(jelly.full.df, main = "Source SAD", xlab = "Species abundance")

jelly.A.df <- unlist(jelly.rarefied["GroupA",])
jelly.A.df <- as.vector(jelly.A.df)

hist(jelly.A.df, main ="Group A SAD", xlab = "species abundance")

jelly.B.df <- unlist(jelly.rarefied["GroupB",])
jelly.B.df <- as.vector(jelly.B.df)

hist(jelly.B.df, main = "Group B SAD", xlab = "Species abundance")
```



3) Figure caption

Write an informative yet succicnt (~5 sentences) caption that creates a "stand-alone" figure. Take a peek at figures and figure captions in a paper published in your favorite journal for inspiration.

> Figure 1: Rank abundance curves for the source and divided communities of a vicariance event show a Mantelbrot best fit models for the Source and Group A community and a Lognormal best fit model for the Group B community. Group B is more even than the source community likely due random division of the source community at the time of the vicariance event. Group A shows a best fit model and RAC most similar to that of the Source community. Species abundance distributions for the Source, A and B communities show a higher abundance of the most abundance spcies in the Group A community and higher frequency of low abundance species.

SUBMITTING YOUR ASSIGNMENT

Use Knitr to create a PDF of your completed 6.DiversisitySampling_Worksheet.Rmd document, push it to GitHub, and create a pull request. Please make sure your updated repo includes both the pdf and RMarkdown files.

Unless otherwise noted, this assignment is due on **Wednesday, January 30th, 2017 at 12:00 PM (noon)**.