7. Worksheet: Diversity Synthesis

Carter Stancil; Z620: Quantitative Biodiversity, Indiana University

24 February, 2026

OVERVIEW

In this worksheet, you will conduct exercises that reinforce fundamental concepts of biodiversity. First, you will construct a site-by-species matrix by sampling confectionery taxa from a source community. Second, you will make a preference-profile matrix, reflecting each student's favorite confectionery taxa. With this primary data structure, you will then answer questions and generate figures using tools from previous weeks, along with wrangling techniques that we learned about in class.

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 handouts 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 `7.DiversitySynthesis_Worskheet.Rmd` and the PDF output of `Knitr` (`DiversitySynthesis_Worskheet.pdf`).

QUANTITATIVE CONFECTIONOLOGY

We will construct a site-by-species matrix using confectionery taxa (i.e, jelly beans). The instructors have created a **source community** with known abundance (N) and richness (S). Like a real biological community, the species abundances are unevenly distributed such that a few jelly bean types are common while most are rare. Each student will sample the source community and bin their jelly beans into operational taxonomic units (OTUs).

SAMPLING PROTOCOL: SITE-BY-SPECIES MATRIX

1. From the well-mixed source community, each student should take one Dixie Cup full of individuals.

2. At your desk, sort the jelly beans into different types (i.e., OTUs), and quantify the abundance of each OTU.

3. Working with other students, merge data into a site-by-species matrix with dimensions equal to the number of students (rows) and taxa (columns)

4. Create a worksheet (e.g., Google sheet) and share the site-by-species matrix with the class.

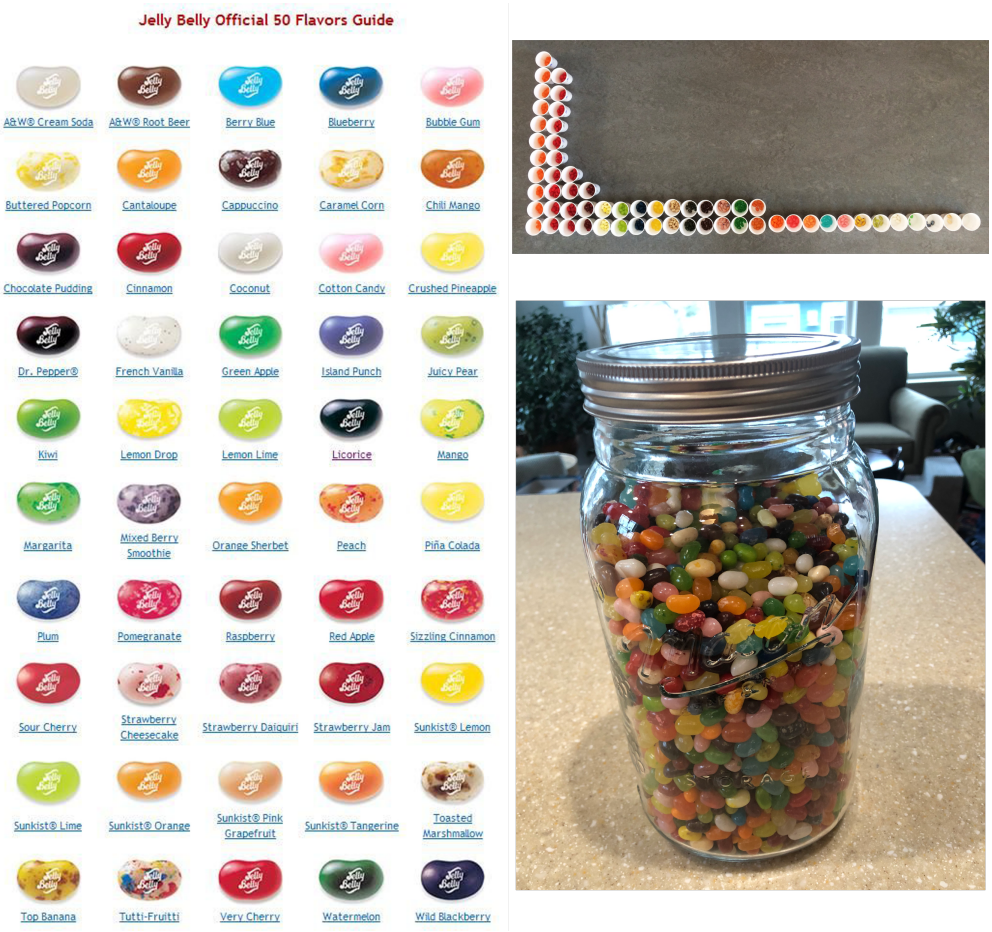


Figure 1: **Left**: taxonomic key, **Top right**: rank abundance distribution, **Bottom right**: source community

SAMPLING PROTOCOL: PREFERENCE-PROFILE MATRIX

1. With your individual sample only, each student should choose their top 5-10 preferred taxa based on flavor, color, sheen, etc.

2. Working with other students, merge data into preference-profile incidence matrix where 1 = preferred and 0 = non-preferred taxa.

3. Create a worksheet (e.g., Google sheet) and share the preference-profile matrix with the class.

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 `Week5-Confection/` folder, and 4) Load the `vegan` R package (be sure to install first if you have not already).

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

```
## [1] "C:/github/QB2026_Stancil/QB2026/Week5-DataWrangling"
```

```
setwd("../Week5-DataWrangling/")

package.list <- c('vegan','tidyverse','ggplot2','dplyr','broom')
for (package in package.list) {
  if (!require(package, character.only = TRUE, quietly = TRUE)) {
    install.packages(package)
  }
  library(c(package), character.only = TRUE)
}
```

```
## Warning: package 'vegan' was built under R version 4.5.2
```

```
## Warning: package 'permute' was built under R version 4.5.2
```

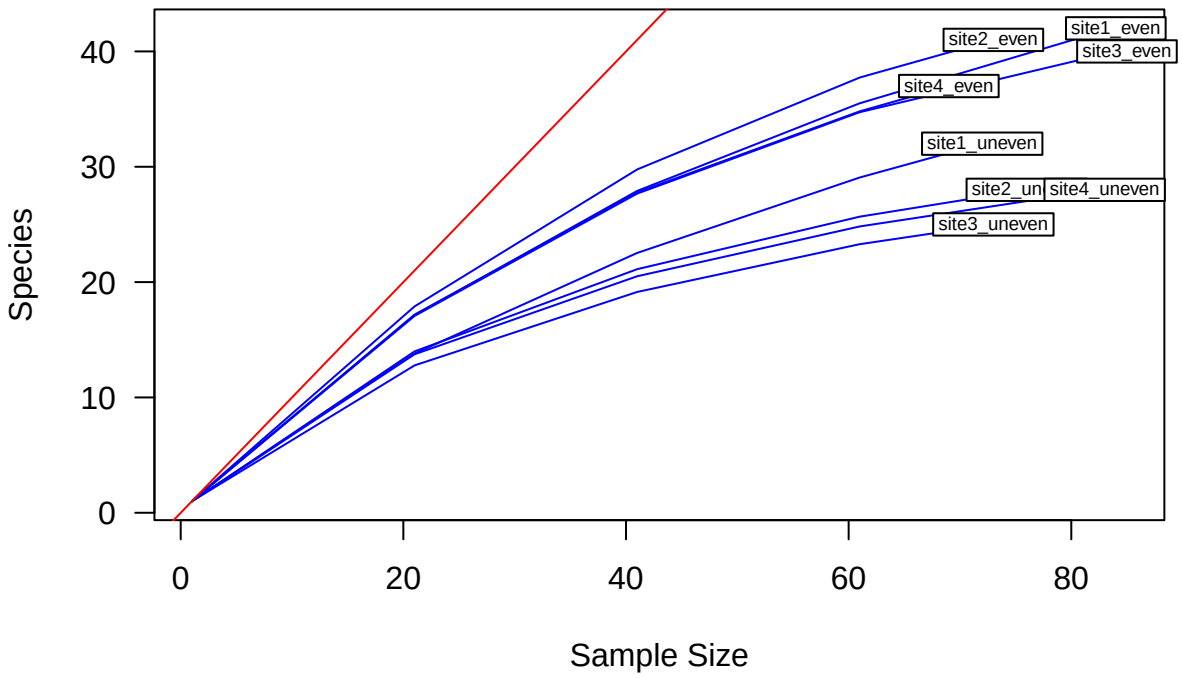
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
dat <- read.csv(file = "./simulated.csv", header = TRUE, row.names = 1)
```

DATA ANALYSIS

Question 1: In the space below, generate a rarefaction plot for all samples of the source community. Based on these results, discuss how individual vs. collective sampling efforts capture the diversity of the source community.

```
min.N <- min(rowSums(dat))
S.rarefy <- rarefy(x = dat, sample = min.N, se = TRUE)
rarecurve(x = dat, step = 20, col = "blue", cex = 0.6, las = 1)
abline(0, 1, col = 'red')
text(1500, 1500, "1:1", pos = 2, col = 'red')
```




```
#I originally transposed the data because we did last time, but I don't think I need to because species
```

Answer 1: On the rarefaction curve, you can clearly see that the sites differ in the amount of diversity captured (measured as # of species). Collective sampling efforts are always going to be better at estimating true diversity of a population or community.

Question 2: Starting with the site-by-species matrix, visualize beta diversity. In the code chunk below, conduct principal coordinates analyses (PCoA) using both an abundance- and incidence-based resemblance matrix. Plot the sample scores in species space using different colors, symbols, or labels. Which "species" are contributing the patterns in the ordinations? How does the choice of resemblance matrix affect your interpretation?

```
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 4.5.2
```

```
library(viridis)
```

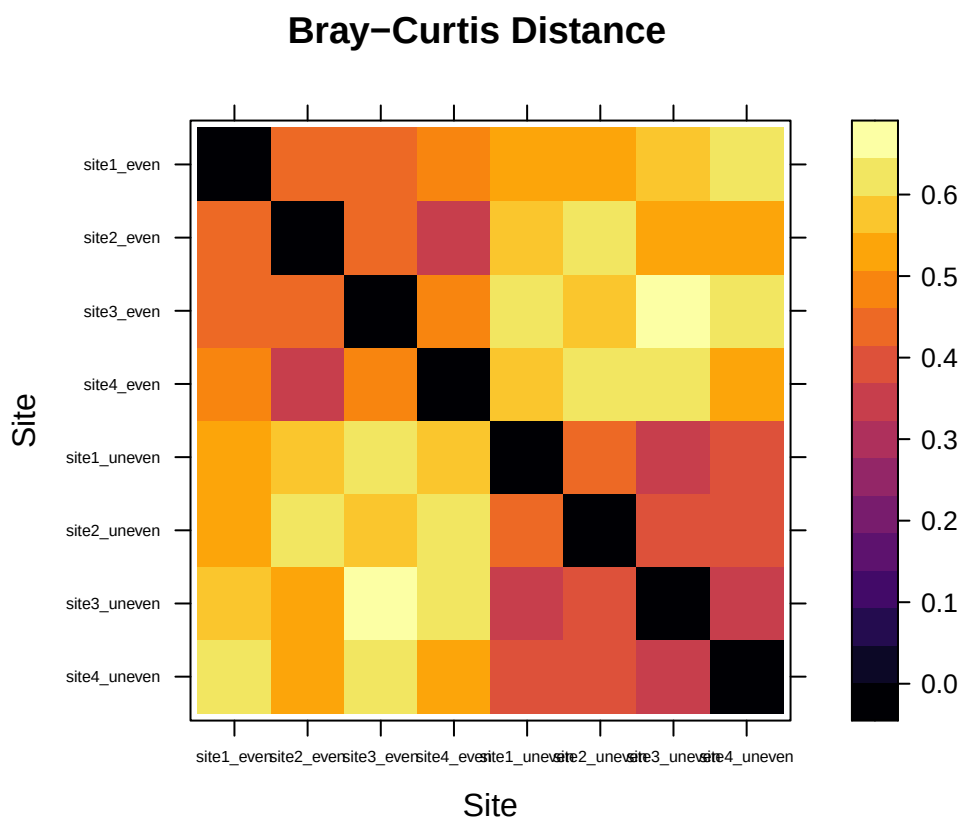
```
## Loading required package: viridisLite
```

```
library(BiodiversityR)
```

```
## Warning: package 'BiodiversityR' was built under R version 4.5.2
```

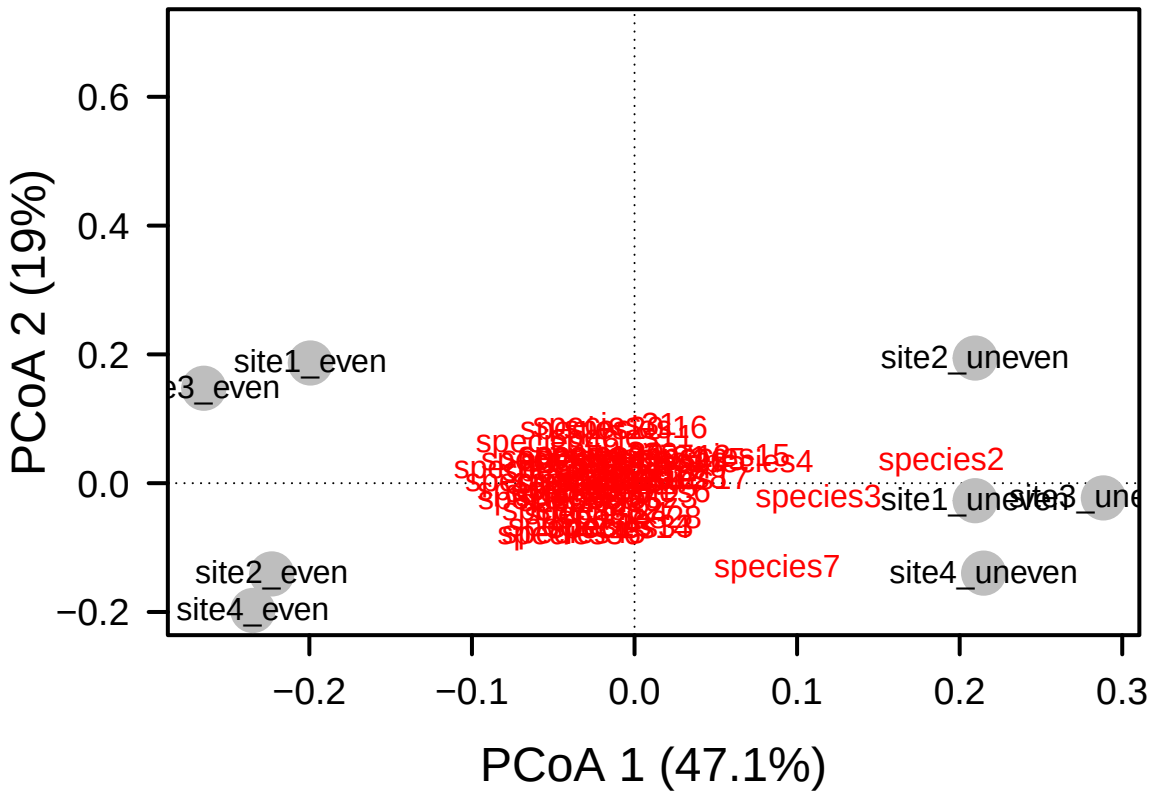
```
## Loading required package: tcltk

## BiodiversityR 2.17-4: Use command BiodiversityRGUI() to launch the Graphical User Interface;
## to see changes use BiodiversityRGUI(changeLog=TRUE, backward.compatibility.messages=TRUE)
```

```r
# Calculate Jaccard
species.j <- vegdist(dat, method = "jaccard", binary = TRUE)
# Calculate Bray-Curtis
species.db <- vegdist(dat, method = "bray")
# Calculate Sørensen
species.so <- vegdist(dat, method = "bray", binary = TRUE)

# Define Order of Sites
order <- rev(attr(species.db, "Labels"))

# Plot Heatmap
levelplot(as.matrix(species.db)[, order], aspect = "iso", col.regions = inferno,
          xlab = "Site", ylab = "Site", scales = list(cex = 0.5),
          main = "Bray-Curtis Distance")
```




```r
datREL <- dat
  for(i in 1:nrow(dat)){
    datREL[i, ] = dat[i, ] / sum(dat[i, ])
    }
```

```r
species.pcoa <- cmdscale(species.db, eig = TRUE, k = 3)
explainvar1 <- round(species.pcoa$eig[1] / sum(species.pcoa$eig), 3) * 100
explainvar2 <- round(species.pcoa$eig[2] / sum(species.pcoa$eig), 3) * 100
explainvar3 <- round(species.pcoa$eig[3] / sum(species.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

# Define Plot Parameters
par(mar = c(5, 5, 1, 2) + 0.1)

# Initiate Plot
plot(species.pcoa$points[ ,1], species.pcoa$points[ ,2], ylim = c(-0.2, 0.7),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5,
     cex.axis = 1.2, axes = FALSE)

# Add Axes
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

# Add Points & Labels
points(species.pcoa$points[ ,1], species.pcoa$points[ ,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(species.pcoa$points[ ,1], species.pcoa$points[ ,2],
     labels = row.names(species.pcoa$points))


# Now, we use this information to calculate and add species scores
species.pcoa <- add.spec.scores(species.pcoa,datREL,method = "pcoa.scores")
text(species.pcoa$cproj[ ,1], species.pcoa$cproj[ ,2],
     labels = row.names(species.pcoa$cproj), col = "red")
```




```
spe.corr <- add.spec.scores(species.pcoa, datREL, method = "cor.scores")$cproj
corrcut  <- 0.7        # user defined cutoff
imp.spp  <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]

# Permutation Test for Species Abundances Across Axes
fit <- envfit(species.pcoa, datREL, perm = 999) # compare to imp.spp list

# Subset the even community
com_e_H <- diversity(dat[grepl("_even",row.names(dat)),], index = "shannon")
print(com_e_H)
```

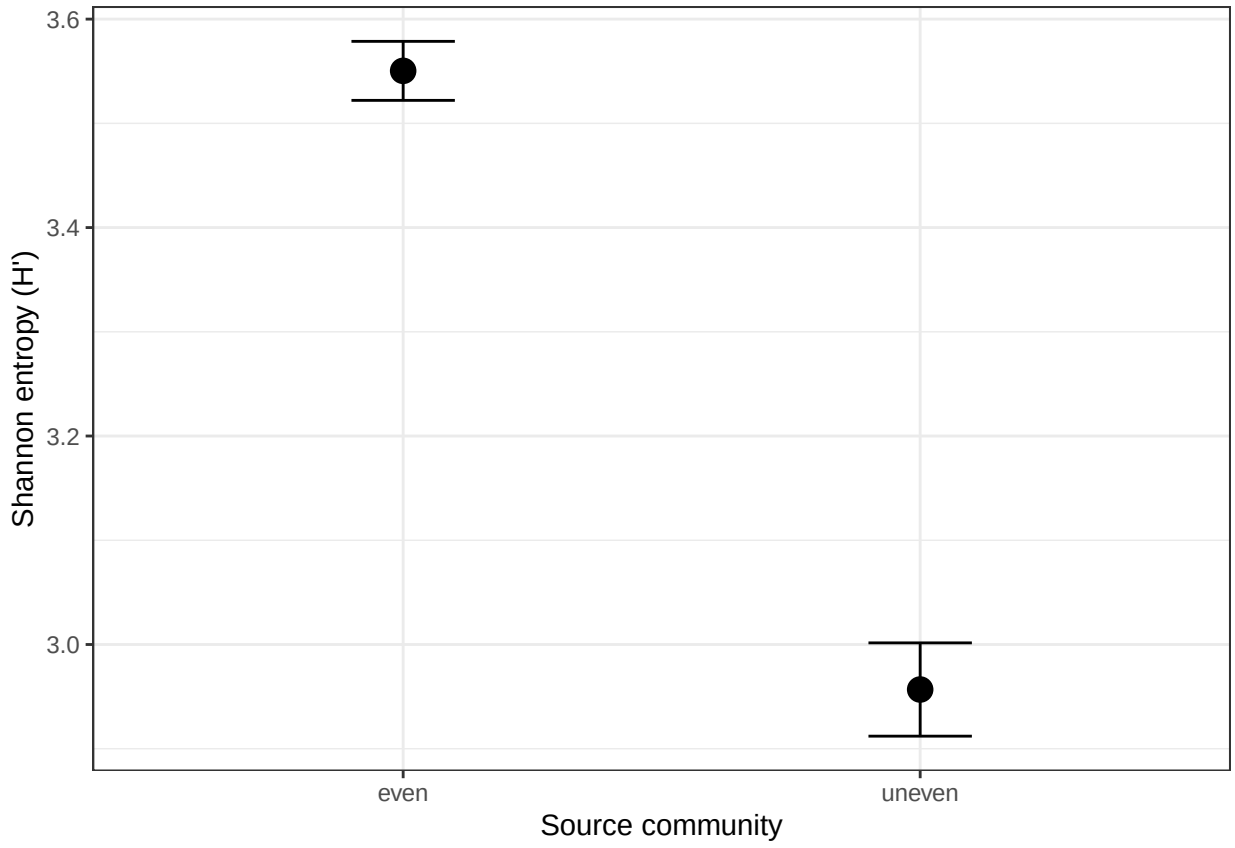
```
## site1_even site2_even site3_even site4_even
##   3.563075   3.616117   3.543160   3.479093
```

```
# Subset the uneven community
com_u_H <- diversity(dat[grepl("_uneven",row.names(dat)),], index = "shannon")
print(com_u_H)
```

```
## site1_uneven site2_uneven site3_uneven site4_uneven
##     2.964376     3.026223     2.828338     3.008247
```

```
# write function for SEM
SEM <- function(x) {
  return(sd(x)/sqrt(length(x)))
```

```
}

# The following code binds the two vectors as columns in a data frame
com_div_all <- t(cbind.data.frame(com_e_H, com_u_H))

# Rename the row names to be more intelligibl
row.names(com_div_all) <- c("even", "uneven")

# Create an empty data frame to fill with summary statistics
com_div_sum <- as.data.frame(matrix(ncol = 2 , nrow = 2))
colnames(com_div_sum) <- c("mean", 'sem')

# The following for loop will iteratively calculate
# the mean and standard error for each row, one at a time.
for(i in 1:2) {
  x = as.numeric(com_div_all[i,])
  com_div_sum[i,1] <- mean(x) # Calculate mean, save to diversity indices data
  com_div_sum[i,2] <- SEM(x)   # Calculate SEM, save to diversity indices data
}

# Let us add a grouping column to the data
com_div_sum$community <- c("even", "uneven")

ggplot(data = com_div_sum, aes(x = community, y = mean))+
  geom_point(size = 4)+
  geom_errorbar(aes(ymin = mean-sem, ymax = mean+sem), width = 0.2)+
  ylab("Shannon entropy (H')")+
  xlab("Source community")+
  theme_bw()
```




```
# if only interested in mean, can just write `mean` at end of function
# for both mean and sem, here, we specify as follows:
com_div_sum_apply <- t(apply(com_div_all, 1, FUN = function(x)
      {c(mean = mean(x), SEM = sd(x)/sqrt(length(x)))}))
print(com_div_sum_apply)
```

```
##            mean        SEM
## even   3.550361 0.02830846
## uneven 2.956796 0.04474587
```

> ***Answer 2***: I believe that species 3, 15, and 31 are driving a lot of the observed variation. Bray-curtis emphasizes abundance differences, while Sorenson emphasizes species turnover. So a PCA based on Sorenson's resemblance matrix would show which sites share species, while our PCA shows how communities are structured.

Question 3 Using the preference-profile matrix, determine the most popular jelly bean in the class using a control structure (e.g., for loop, if statement, function, etc).

```
library(dplyr)

# This doesn't make sense so we decided to interpret JellyBeans Subsampled as a preference matrix (i.e.

jelly <- as.data.frame(read.table("JellyBeans.Subsampled.txt", header = TRUE))

jelly <- jelly[,2:30]
```

```r
col_sums <- jelly %>%
  summarise(across(-Site, sum))

max_species <- names(col_sums)[which.max(col_sums)] # the most popular favor (most eaten)

#FOR LOOP OPTION
# Remove the Site column so only numeric columns remain
mat <- jelly[,-1]

# Create an empty vector to store sums
col_sums <- numeric(ncol(mat))
names(col_sums) <- colnames(mat)

# Loop through each column
for (i in seq_along(mat)) {
  col_sums[i] <- sum(mat[[i]], na.rm = TRUE)
}

col_sums
```

```
##           Red     RedShinny     OrangeSpe     OrangeDrk    OrangeBrgt
##            20             8            90             7            24
##     OrangeShn   OrangeLight OrangSuperShn     YellowSld   YellowTrans
##            17            24            18            15            29
##   Watermellon    GreenTrans   GreenTrans2    GreenSoild GreenSuperShn
##            84            31             2            19             8
##  GreenMintCho  AquaSuperShn    BlueDrkShn       BlueDrk     BlueMuted
##             9            17            11           128            21
##     PurpleShn    PurpleSpot       PinkShn      PinkSpot       Rainbow
##             8            16            14            15            12
##  WhiteBrownSP WhiteSuperShn    WhiteSolid
##            14             6             3
```

Answer 3: Blue Dark is the most popular jelly bean flavor.

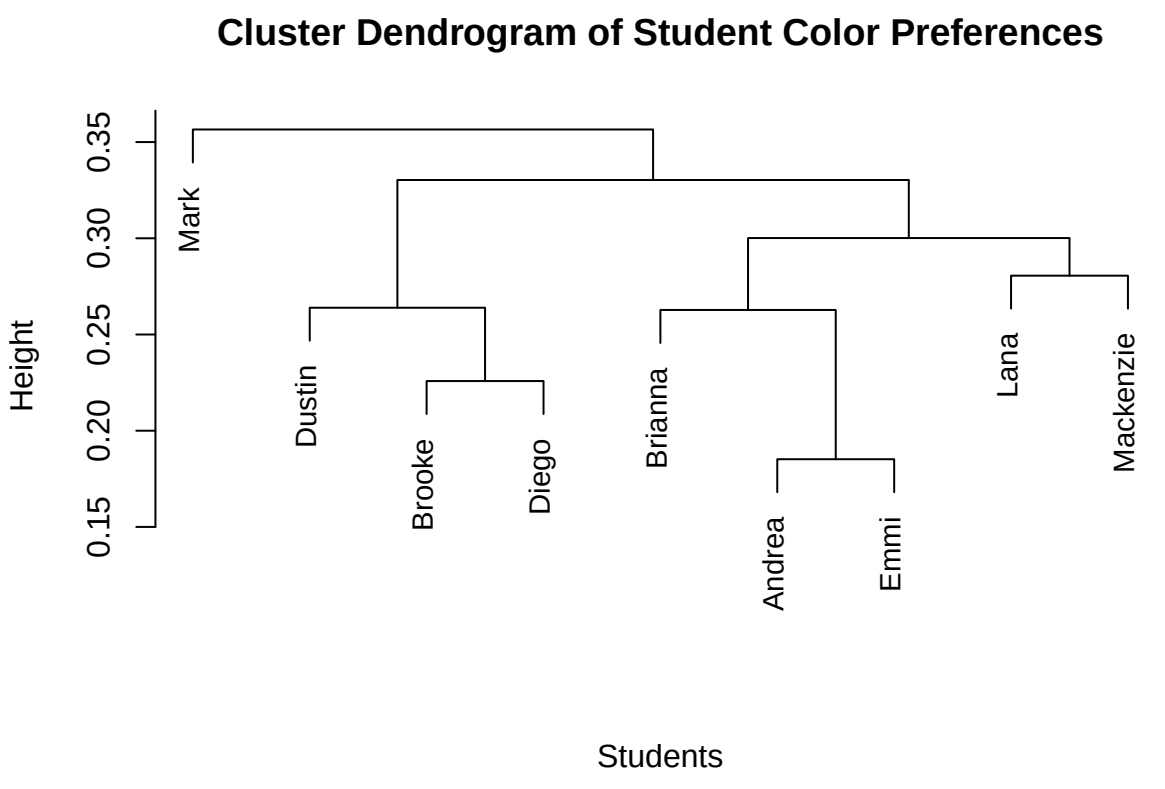
Question 4 In the code chunk below, identify the student in QB who has a preference-profile that is most like yours. Quantitatively, how similar are you to your "jelly buddy"? Visualize the preference profiles of the class by creating a cluster dendogram. Label each terminal node (a.k.a., tip or "leaf") with the student's name or initials. Make some observations about the preference-profiles of the class.

```r
#Identify color (species) columns
color_cols <- setdiff(colnames(jelly), c("Site"))
# Create a new row
new_row <- data.frame(
Group = "C",
Site = "Carter",
as.list(rep(1, length(color_cols)))
)

names(new_row)[3:ncol(new_row)] <- color_cols
# Add to bottom of jelly.sub
```

```r
new_row <- new_row %>%
  select(,2:30)

jelly.sub <- rbind(jelly, new_row)
# compare
jellies.c <- jelly.sub[, color_cols]
bray_9_10 <- vegdist(jellies.c[c(9, 10), ], method = "bray")
bray_9_10
```

```
##           9
## 10 0.5402299
```

```r
color_cols <- setdiff(colnames(jelly.sub), c("Group", "Site"))
jellies <- jelly.sub[, color_cols]
bc_dist <- vegdist(jellies[1:9,], method = "bray")
hc <- hclust(bc_dist, method = "average")
hc$labels <- as.character(jelly.sub$Site[1:9])
plot( hc, main = "Cluster Dendrogram of Student Color Preferences", xlab = "Students", sub = "", cex = 
```



Answer 4: Even though I "ate" one of every flavor of jelly bean (because I will try anything once), my preference data was as dissimilar as I anticipated from Mark's. Compared to the rest of the class, Mark was an outlier.

SUBMITTING YOUR ASSIGNMENT

Use Knitr to create a PDF of your completed `7.DiversitySynthesis_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, February 19th, 2025 at 12:00 PM (noon)**.