

# 7. Worksheet: Diversity Synthesis

Maddy Spencer; Z620: Quantitative Biodiversity, Indiana University

18 February, 2025

## 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.

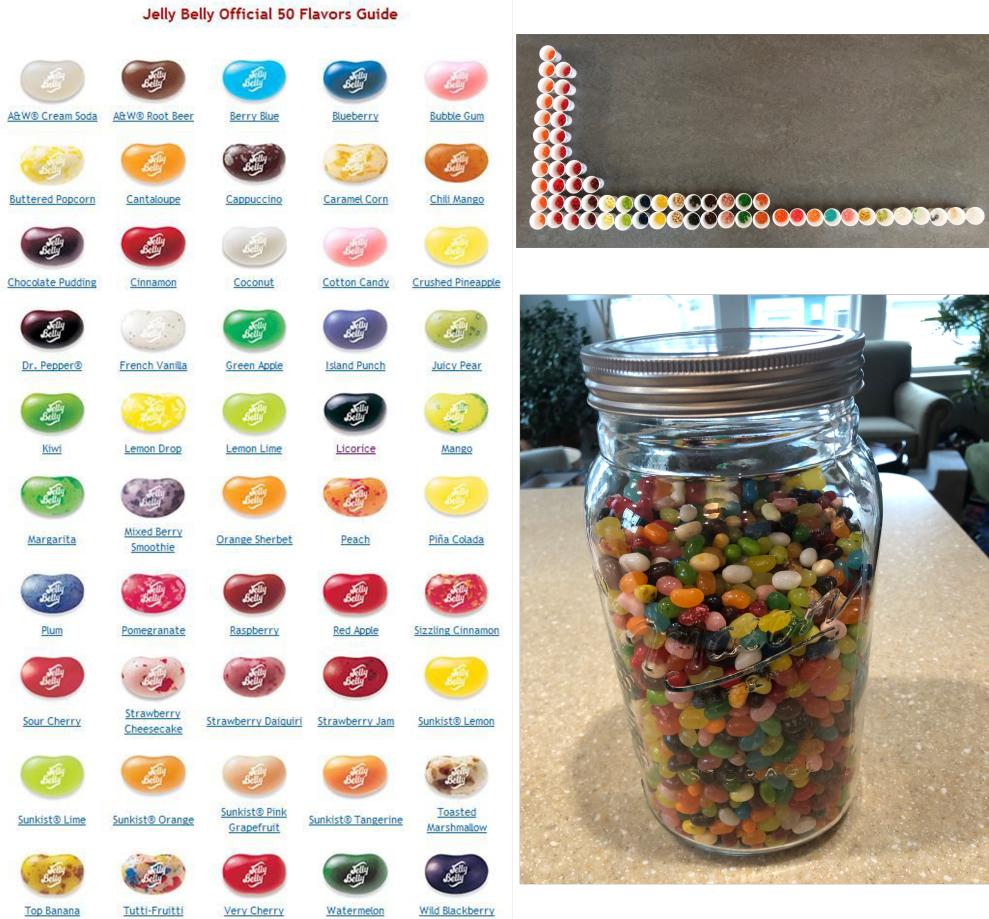


Figure 1: **Left:** taxonomic key, **Top right:** rank abundance distribution, **Bottom right:** source community

## 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] "/cloud/project/QB2025_Spencer/Week5-Confection"
require(vegan)

## Loading required package: vegan
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.6-8
require(dplyr)

## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##   filter, lag
## The following objects are masked from 'package:base':
##   intersect, setdiff, setequal, union
```

## 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.

```
jellydata <- read.table("data/Jellybelly_data.txt", sep = "\t", header = TRUE, row.names = 1)
jellydata <- as.data.frame(jellydata)
jellydata <- jellydata[, -which(colnames(jellydata) == "Chili.Mango")]

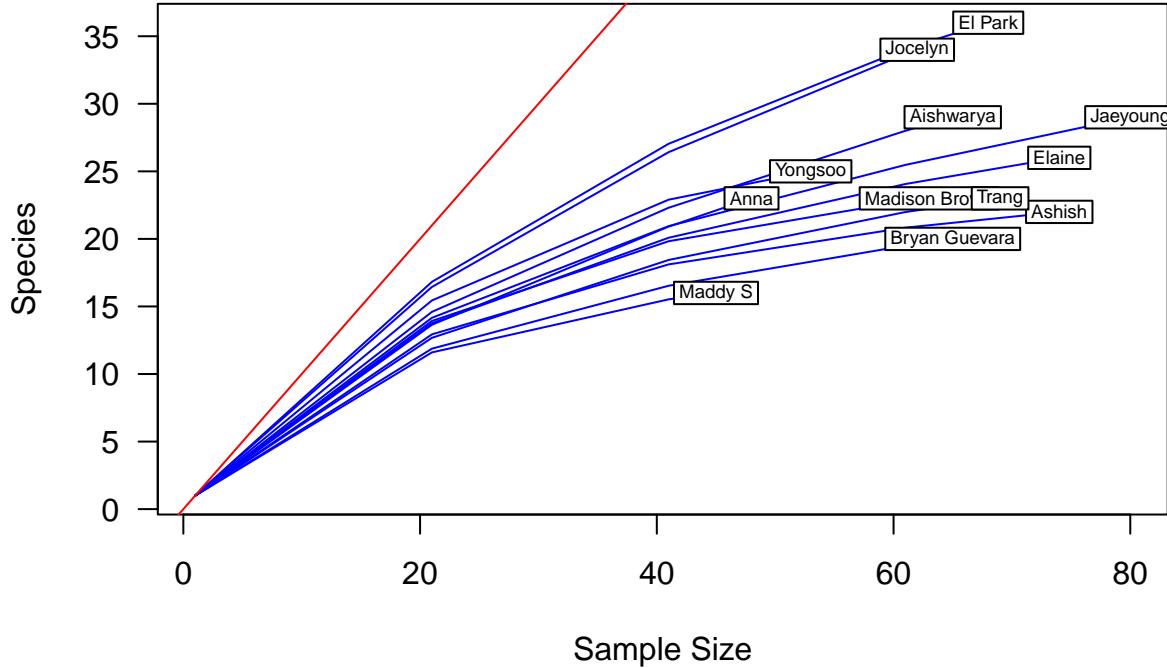
S.obs <- function(x = ""){
  rowSums( x > 0 ) * 1
}

jellydata.S <- S.obs(jellydata)
```

```

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

```



**Answer 1:** Each line above, associated with each person or site, falls below the 1:1 line. This means that the true species richness is being underestimated due to poor sampling efforts. There is also a general trend of larger sample size equal to more observed richness, though there are several “sites” that break this rule. This rarefaction plot demonstrates the power of collective sampling, as each individual by themselves does not properly encapsulate the true diversity of the communities. Aggregation of this data into a singular site would bring us much closer to an accurate estimate of richness.

**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?

```

#Sorenson's similarity (presence-absence)
jelly.ds <- vegdist(jellydata, method = "bray", diag = TRUE, binary = TRUE)

jellypcoa.ds <- cmdscale(jelly.ds, eig = TRUE, k = 3)

explainvar1.ds <- round(jellypcoa.ds$eig[1] / sum(jellypcoa.ds$eig), 3) * 100
explainvar2.ds <- round(jellypcoa.ds$eig[2] / sum(jellypcoa.ds$eig), 3) * 100
explainvar3.ds <- round(jellypcoa.ds$eig[3] / sum(jellypcoa.ds$eig), 3) * 100
sumeig.ds <- sum(explainvar1.ds, explainvar2.ds, explainvar3.ds)
print(sumeig.ds)

## [1] 62.5

```

```

par(mar = c(5, 5, 1, 2) + 0.1)
plot(jellypcoa.ds$points[,1], jellypcoa.ds$points[,2], ylim = c(-0.4, 0.7),
  main = "Sorenson's Similarity",
  xlab = paste("PCoA 1 (", explainvar1.ds, "%)", sep = ""),
  ylab = paste("PCoA 2 (", explainvar2.ds, "%)", sep = ""),
  pch = 16, cex = 2.0, type = "n", cex.lab = 1.5,
  cex.axis = 1.2, axes = FALSE)

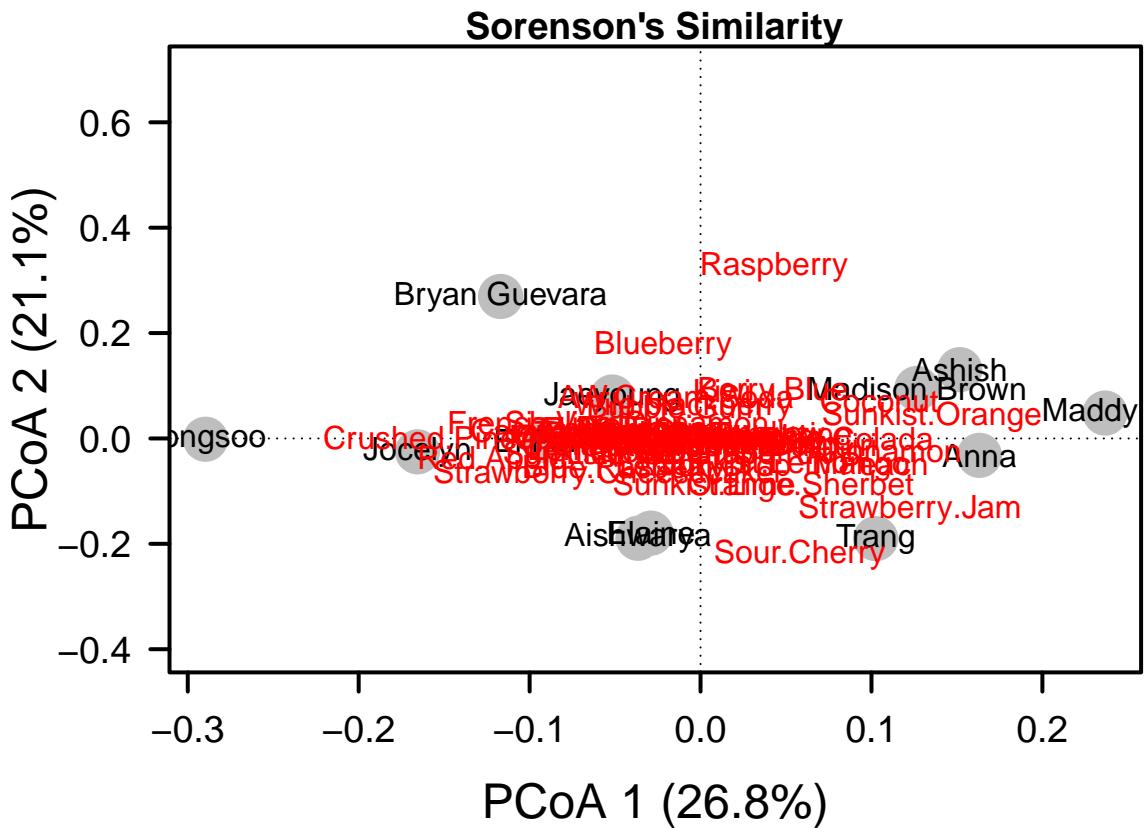
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

points(jellypcoa.ds$points[,1], jellypcoa.ds$points[,2],
  pch = 19, cex = 3, bg = "gray", col = "gray")
text(jellypcoa.ds$points[,1], jellypcoa.ds$points[,2],
  labels = row.names(jellypcoa.ds$points))

jellyREL <- jellydata
for(i in 1:nrow(jellydata)){
  jellyREL[i, ] = jellydata[i, ] / sum(jellydata[i, ])
}

jellypcoa.ds <- add.spec.scores.class(jellypcoa.ds, jellyREL, method = "pcoa.scores")
text(jellypcoa.ds$cproj[,1], jellypcoa.ds$cproj[,2],
  labels = row.names(jellypcoa.ds$cproj), col = "red")

```



```

#Bray-Curtis dissimilarity (abundance-based)
jelly.db <- vegdist(jellydata, method = "bray", diag = TRUE)

jellypcoa.db <- cmdscale(jelly.db, eig = TRUE, k = 3)

explainvar1.db <- round(jellypcoa.db$eig[1] / sum(jellypcoa.db$eig), 3) * 100
explainvar2.db <- round(jellypcoa.db$eig[2] / sum(jellypcoa.db$eig), 3) * 100
explainvar3.db <- round(jellypcoa.db$eig[3] / sum(jellypcoa.db$eig), 3) * 100
sumeig.db <- sum(explainvar1.db, explainvar2.db, explainvar3.db)
print(sumeig.db)

## [1] 51.1

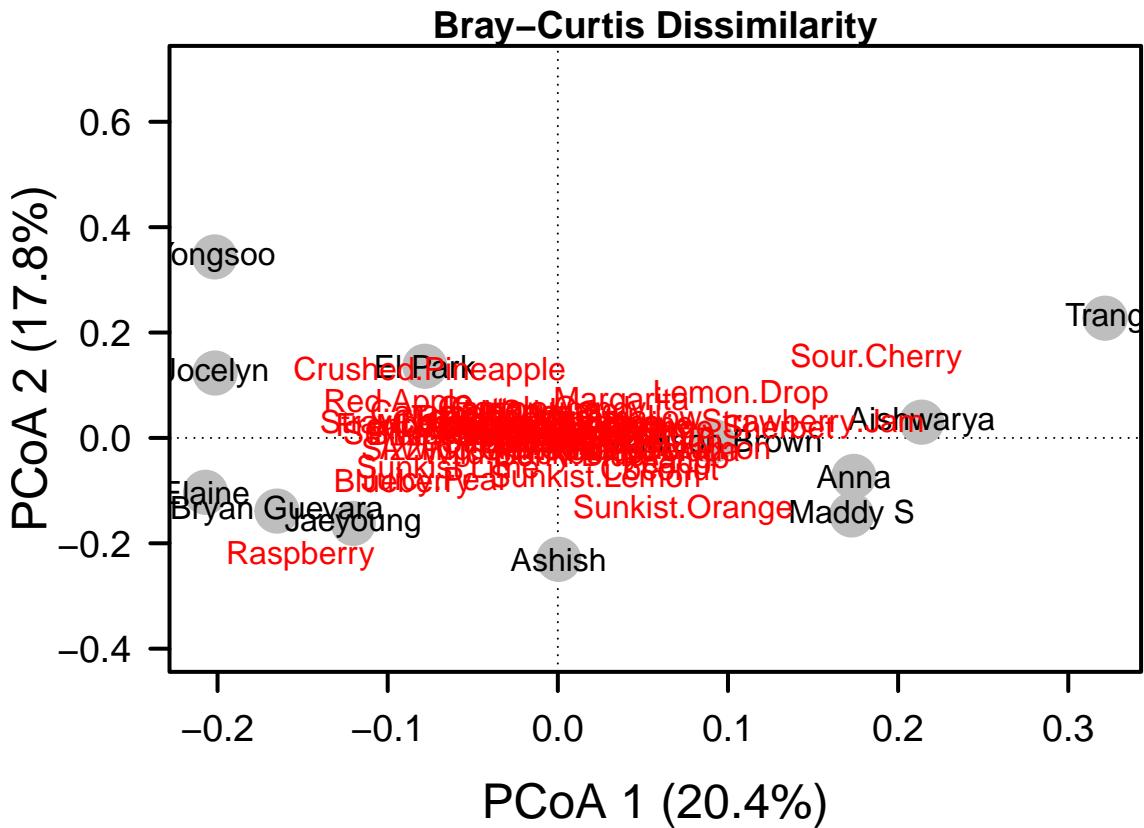
par(mar = c(5, 5, 1, 2) + 0.1)
plot(jellypcoa.db$points[,1], jellypcoa.db$points[,2], ylim = c(-0.4, 0.7),
     main = "Bray-Curtis Dissimilarity",
     xlab = paste("PCoA 1 (", explainvar1.db, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2.db, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5,
     cex.axis = 1.2, axes = FALSE)

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)

points(jellypcoa.db$points[,1], jellypcoa.db$points[,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(jellypcoa.db$points[,1], jellypcoa.db$points[,2],
     labels = row.names(jellypcoa.db$points))

jellypcoa.db <- add.spec.scores.class(jellypcoa.db, jellyREL, method = "pcoa.scores")
text(jellypcoa.db$cproj[,1], jellypcoa.db$cproj[,2],
     labels = row.names(jellypcoa.db$cproj), col = "red")

```



```
#influential species using Sorenson's
jellycorr.ds <- add.spec.scores.class(jellypcoa.ds, jellyREL, method = "cor.scores")$cproj
corrcut <- 0.7
impspp.ds <- jellycorr.ds[abs(jellycorr.ds[,1]) >= corrcut | abs(jellycorr.ds[,2]) >= corrcut, ]
print(impps.ds)

##                                     Dim1      Dim2      Dim3
## Cantaloupe      -0.7574624 -0.1887735 -0.35385045
## Caramel.Corn   -0.7017202  0.1174595  0.08843585
## Cinnamon        0.8088156 -0.1490863 -0.31819281
## French.Vanilla -0.7147216  0.2090501  0.22350791
## Red.Apple       -0.8207350 -0.2511152  0.25296966

fit.ds <- envfit(jellypcoa.ds, jellyREL, perm = 999)
print(fit.ds)

##
## ***VECTORS
##
##                                     Dim1      Dim2      r2 Pr(>r)
## AW.Cream.Soda    -0.17858  0.98393 0.4083  0.089 .
## AW.Root.Beer     0.91190 -0.41042 0.0277  0.883
## Berry.Blue       0.42816  0.90370 0.4464  0.063 .
## Blueberry        -0.12386  0.99230 0.4305  0.068 .
## Bubble.Gum       -0.24255  0.97014 0.1778  0.383
## Buttered.Popcorn 0.49330 -0.86986 0.1167  0.561
## Cantaloupe       -0.96282 -0.27013 0.6094  0.007 **
## Cappuccino       -0.15062 -0.98859 0.1890  0.390
```

```

## Caramel.Corn          -0.98270  0.18518  0.5062  0.045 *
## Chocolate.Pudding    -0.99768 -0.06810  0.4780  0.028 *
## Cinnamon              0.97914 -0.20318  0.6764  0.008 **
## Coconut                0.82454  0.56581  0.5220  0.039 *
## Cotton.Candy          -0.99684  0.07939  0.1443  0.495
## Crushed.Pineapple     -0.99946 -0.03281  0.4451  0.045 *
## Dr.Pepper              -0.91232  0.40948  0.2016  0.348
## French.Vanilla         -0.94983  0.31276  0.5545  0.026 *
## Green.Apple            -0.94587 -0.32453  0.0073  0.964
## Island.Punch           -0.99882 -0.04849  0.0336  1.000
## Juicy.Pear              0.96953  0.24499  0.0454  0.799
## Kiwi                   0.13760  0.99049  0.3008  0.182
## Lemon.Drop              0.03160 -0.99950  0.0278  0.906
## Lemon.Lime              0.93124 -0.36441  0.3692  0.130
## Licorice                0.86776 -0.49699  0.4798  0.041 *
## Mango                  0.87581 -0.48266  0.2424  0.280
## Margarita               0.99852 -0.05436  0.0167  0.946
## Mixed.Berry.SMOOTHIE   -0.99270 -0.12059  0.1125  0.628
## Orange.Sherbet          0.53868 -0.84251  0.1372  0.539
## Peach                   0.91328 -0.40733  0.3732  0.139
## Piña.Colada             0.99997  0.00753  0.4456  0.072 .
## Plum                    -0.25202 -0.96772  0.0161  0.947
## Pomegranate             0.94983  0.31277  0.1919  0.401
## Raspberry               0.13130  0.99134  0.4671  0.072 .
## Red.Apple                0.94549 -0.32567  0.7367  0.001 ***
## Sizzling.Cinnamon        0.83464  0.55080  0.2124  0.356
## Sour.Cherry              0.25512 -0.96691  0.3871  0.097 .
## Strawberry.Cheesecake   -0.63628 -0.77146  0.3941  0.101
## Strawberry.Daiquiri     0.80202 -0.59730  0.1311  0.624
## Strawberry.Jam           0.67181 -0.74073  0.2004  0.373
## Sunkist.Lemon             0.62396 -0.78146  0.0415  0.805
## Sunkist.Lime              0.02441 -0.99970  0.1153  0.681
## Sunkist.Orange             0.95809  0.28648  0.3282  0.167
## Sunkist.Pink.Grapefruit   -0.53315 -0.84602  0.1177  0.595
## Sunkist.Tangerine          0.98448  0.17548  0.0104  0.949
## Toasted.Marsmallow        -0.76731  0.64128  0.0293  0.867
## Top.Banana                0.19977 -0.97984  0.0751  0.722
## Tutti.Fruitti             0.57191 -0.82031  0.0001  0.999
## Very.Cherry                0.90481  0.42581  0.1334  0.540
## Watermelon                 0.99264  0.12110  0.1194  0.550
## Wild.blackberry            -0.16545  0.98622  0.3592  0.132
## Blue.Rasberry              0.59494 -0.80377  0.3224  0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999

#influential species using Bray-Curtis
jellycorr.db <- add.spec.scores.class(jellypcoa.db, jellyREL, method = "cor.scores")$cproj
corrcut <- 0.7
impspp.db <- jellycorr.db[abs(jellycorr.db[,1]) >= corrcut | abs(jellycorr.db[,2]) >= corrcut, ]
print(impspp.db)

##                      Dim1      Dim2      Dim3
## Cantaloupe -0.3548269  0.7580758 -0.2066158

```

```

## Cinnamon    0.7285807 -0.1553521  0.2845498
## Licorice   -0.3699198  0.7411796  0.1555568
## Red.Apple  -0.7572700  0.4902194 -0.2707572

fit.db <- envfit(jellypcoa.db, jellyREL, perm = 999)
print(fit.db)

## 
## ***VECTORS
##
##                               Dim1      Dim2      r2 Pr(>r)
## AW.Cream.Soda      -0.89319 -0.44967 0.1744  0.427
## AW.Root.Beer        0.65385 -0.75662 0.1208  0.565
## Berry.Blue          0.18531 -0.98268 0.1051  0.588
## Blueberry          -0.72143 -0.69249 0.3779  0.112
## Bubble.Gum          0.74613  0.66580 0.0085  0.966
## Buttered.Popcorn    0.97311  0.23032 0.2931  0.229
## Cantaloupe         -0.40079  0.91617 0.7006  0.006 **
## Cappuccino          0.98746  0.15790 0.1310  0.664
## Caramel.Corn        -0.71317  0.70099 0.5944  0.016 *
## Chocolate.Pudding   -0.58845  0.80853 0.6583  0.008 **
## Cinnamon            0.97495 -0.22243 0.5550  0.025 *
## Coconut              0.71298 -0.70119 0.3318  0.188
## Cotton.Candy        -0.10558  0.99441 0.3877  0.105
## Crushed.Pineapple   -0.51309  0.85833 0.6290  0.008 **
## Dr.Pepper            -0.97379 -0.22745 0.4077  0.095 .
## French.Vanilla       0.94613  0.32378 0.5200  0.035 *
## Green.Apple          -0.87564  0.48296 0.1057  0.637
## Island.Punch         -0.44836  0.89385 0.0764  0.911
## Juicy.Pear           -0.65475 -0.75585 0.3442  0.140
## Kiwi                 0.53679 -0.84372 0.0327  0.841
## Lemon.Drop           0.79666  0.60443 0.2860  0.225
## Lemon.Lime           -0.29156  0.95655 0.5125  0.049 *
## Licorice              -0.42274  0.90625 0.6862  0.010 **
## Mango                0.86334 -0.50462 0.2175  0.308
## Margarita            0.48440  0.87485 0.1717  0.433
## Mixed.Berry.Smoothie -0.62910  0.77733 0.1936  0.416
## Orange.Sherbet        0.98649  0.16380 0.1848  0.395
## Peach                0.81267 -0.58272 0.2392  0.256
## Piña.Colada          0.92603 -0.37745 0.3147  0.192
## Plum                 0.46063 -0.88759 0.0233  0.912
## Pomegranate          -0.38122  0.92448 0.1989  0.380
## Raspberry             -0.55868 -0.82938 0.5298  0.042 *
## Red.Apple             -0.82207  0.56939 0.8138  0.002 **
## Sizzling.Cinnamon    -0.82505 -0.56506 0.3291  0.144
## Sour.Cherry           0.78012  0.62563 0.7361  0.004 **
## Strawberry.Cheesecake -0.85184  0.52380 0.1708  0.443
## Strawberry.Daiquiri   0.65912 -0.75204 0.0653  0.838
## Strawberry.Jam        0.98391  0.17866 0.2272  0.281
## Sunkist.Lemon          0.29528 -0.95541 0.1041  0.613
## Sunkist.Lime            -0.76039 -0.64947 0.1947  0.332
## Sunkist.Orange          0.47286 -0.88114 0.5270  0.035 *
## Sunkist.Pink.Grapefruit -0.98730  0.15887 0.1319  0.551
## Sunkist.Tangerine       0.20534  0.97869 0.0362  0.842
## Toasted.Marsmallow     -0.01500  0.99989 0.2469  0.271

```

```

## Top.Banana          0.99314  0.11693  0.0365  0.856
## Tutti.Fruitti     -0.53303 -0.84610  0.0162  0.930
## Very.Cherry        -0.99388  0.11042  0.0229  0.882
## Watermelon         -0.82304  0.56799  0.1632  0.447
## Wild.blackberry   -0.40508 -0.91428  0.2208  0.331
## Blue.Rasberry      -0.99896  0.04563  0.2574  0.271
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999

```

**Answer 2:** The choice of resemblance matrix does not make a large difference in interpretation. Sorenson's being a presence-absence metric, places equal weight on abundant and rare taxa. We can see that for both axes in our PCoA, using Sorenson's leads to more explanation of variance. The different metrics also influence the influential taxa that drive any clustering differences we see on the PCoA. Interestingly, none of the species that stand out on the PCoA maps like raspberry and sour cherry seem to be driving the differences. For both metrics, cantaloupe, cinnamon and red apple are influential species. For Sorenson's, caramel corn and french vanilla are also influential to the correlation coefficients, whereas with Bray-Curtis dissimilarity licorice also drives clustering.

**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).

```

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

jellypref.sums <- colSums(jellypref)
jellypref.sums <- jellypref.sums[order(-jellypref.sums)]
print(jellypref.sums)

```

##	Berry.Blue	Buttered.Popcorn	Raspberry
##	7	5	5
##	Blueberry	Caramel.Corn	Coconut
##	4	4	4
##	AW.Cream.Soda	AW.Root.Beer	Cinnamon
##	3	3	3
##	Dr.Pepper	Lemon.Drop	Orange.Sherbet
##	3	3	3
##	Piña.Colada	Sour.Cherry	Sunkist.Lemon
##	3	3	3
##	Toasted.Marsmallow	Wild.Blackberry	Bubble.Gum
##	3	3	2
##	Cantaloupe	French.Vanilla	Red.Apple
##	2	2	2
##	Sunkist.Tangerine	Cholcate.Pudding	Crushed.Pineapple
##	2	1	1
##	Lemon.Lime	Margarita	Mixed.Berry.Smoothie
##	1	1	1
##	Peach	Sizzling.Cinnamon	Strawberry.Daiquiri
##	1	1	1
##	Strawberry.Jam	Sunkist.Orange	Sunkist.Pink.Grapefruit
##	1	1	1
##	Top.Banana	Tutti.Fruitti	Cappuccino
##	1	1	0
##	Chili.Mango	Cotton.Candy	Green.Apple
##	0	0	0

```

##           Island.Punch        Juicy.Pear        Kiwi
##                 0                  0                  0
##           Licorice          Mango          Plum
##                 0                  0                  0
##           Pomegranate  Strawberry.Cheesecake Sunkist.Lime
##                         0                          0                  0

```

*Answer 3:* The most popular flavor is berry blue, with 7 votes total.

**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 dendrogram. 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.

```

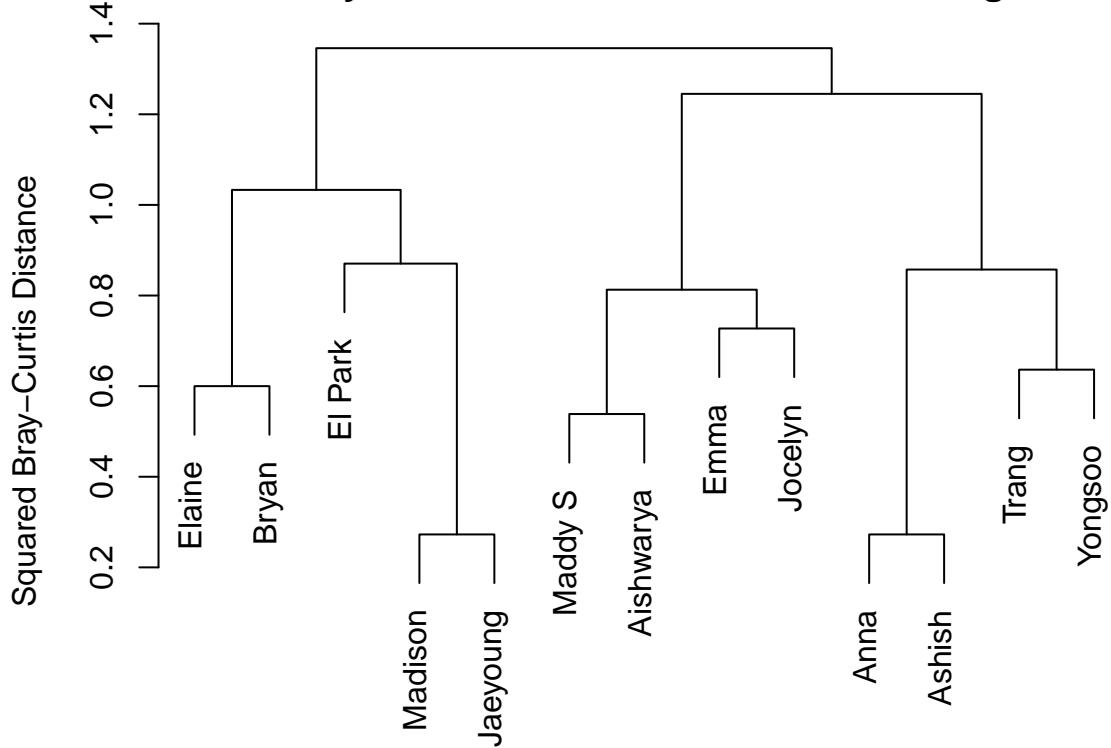
# Calculate Sorenson's distance
jellypref.ds <- vegdist(jellypref, method = "bray", diag = TRUE, binary = TRUE)

#Ward's Clustering
jellypref.ward <- hclust(jellypref.ds, method = "ward.D2")

par(mar = c(1, 5, 2, 2) + 0.1)
plot(jellypref.ward, main = "Jelly Bean Preference: Ward's Clustering",
     ylab = "Squared Bray-Curtis Distance")

```

**Jelly Bean Preference: Ward's Clustering**



*Answer 4:* My jelly buddy is Aishwarya, as we share 3 different favorite flavors in common. Overall, there are two general clusters being formed. On the right side, Elaine and Bryan, and Madison and Jaeyoung form two distinct clusters. El branches off before Madison and Jaeyoung and is not directly clustered with anyone. Madison and Jaeyoung branch off further from the other three, implying more distinct tastes. Aishwarya, Emma, Joselyn, and myself all cluster close together, with Aishwarya and myself clustering more closely. Trang, Yongsoo, Anna, and

Ashish cluster together on the right side. There is a larger distance between the two clusters here than was present in the last grouping (that included myself).

## 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 19<sup>th</sup>, 2025 at 12:00 PM (noon)**.