7. Worksheet: Diversity Synthesis

Yongsoo Choi; Z620: Quantitative Biodiversity, Indiana University

19 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.

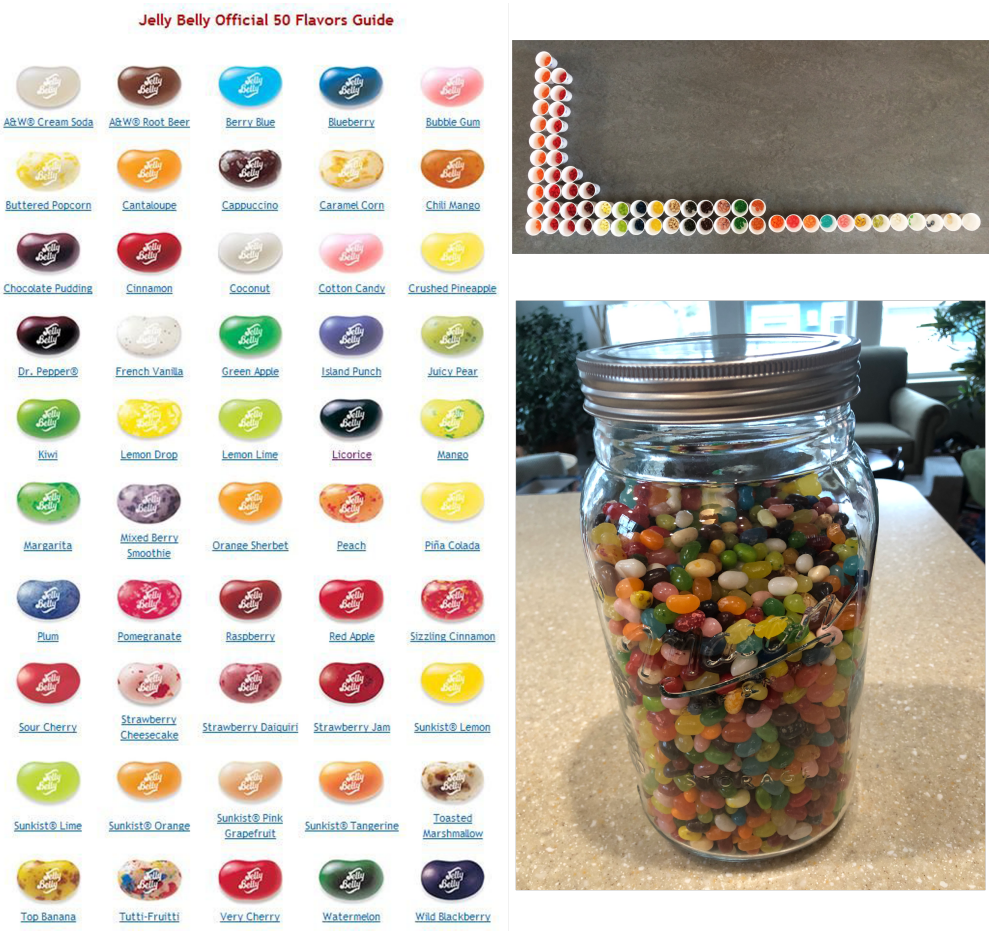


Figure 1: **Left**: taxonomic key, **Top right**: rank abundance distribution, **Bottom right**: source community

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_Choi/Week5-Confection"
```

```
library(vegan)
```

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

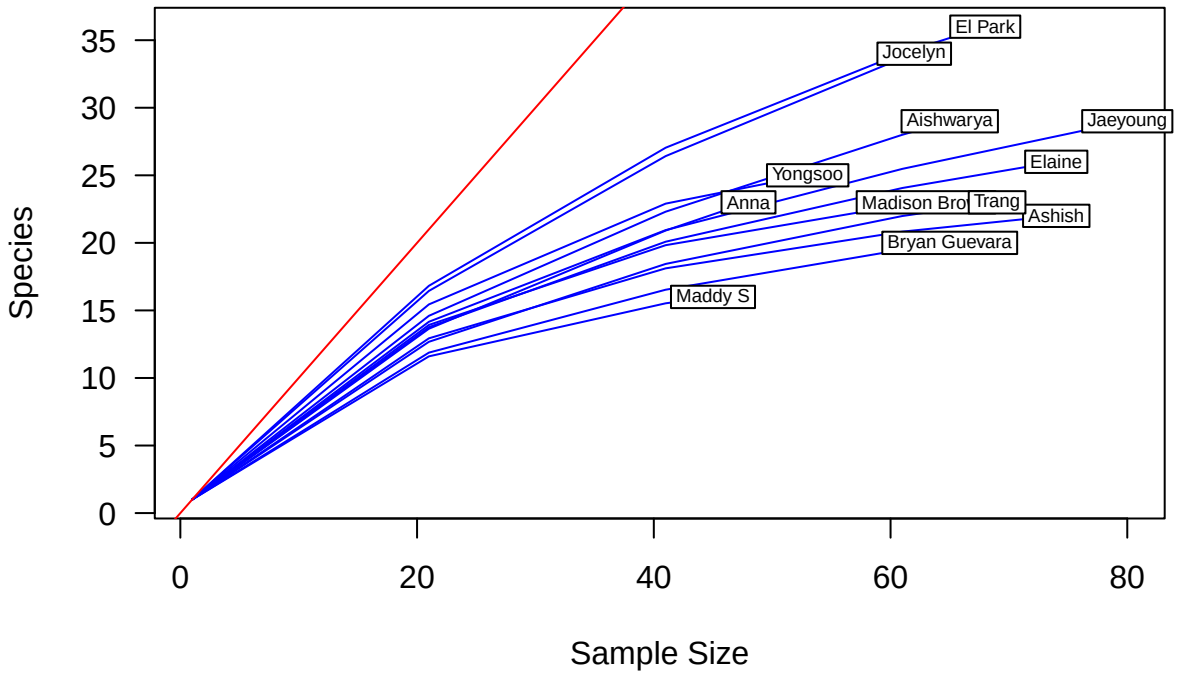
```
## Loading required package: lattice
```

```
## This is vegan 2.6-8
```

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.

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



Answer 1: As we learned at the previous class, we can see the number of species increases as the sample size increases. Additionally, even though all the samples are from the same population, we can see each sample show highly diverse rarefaction curve and speciess richness. Thus, I think collective sampling enable to capture more rare species campared to individual samples.

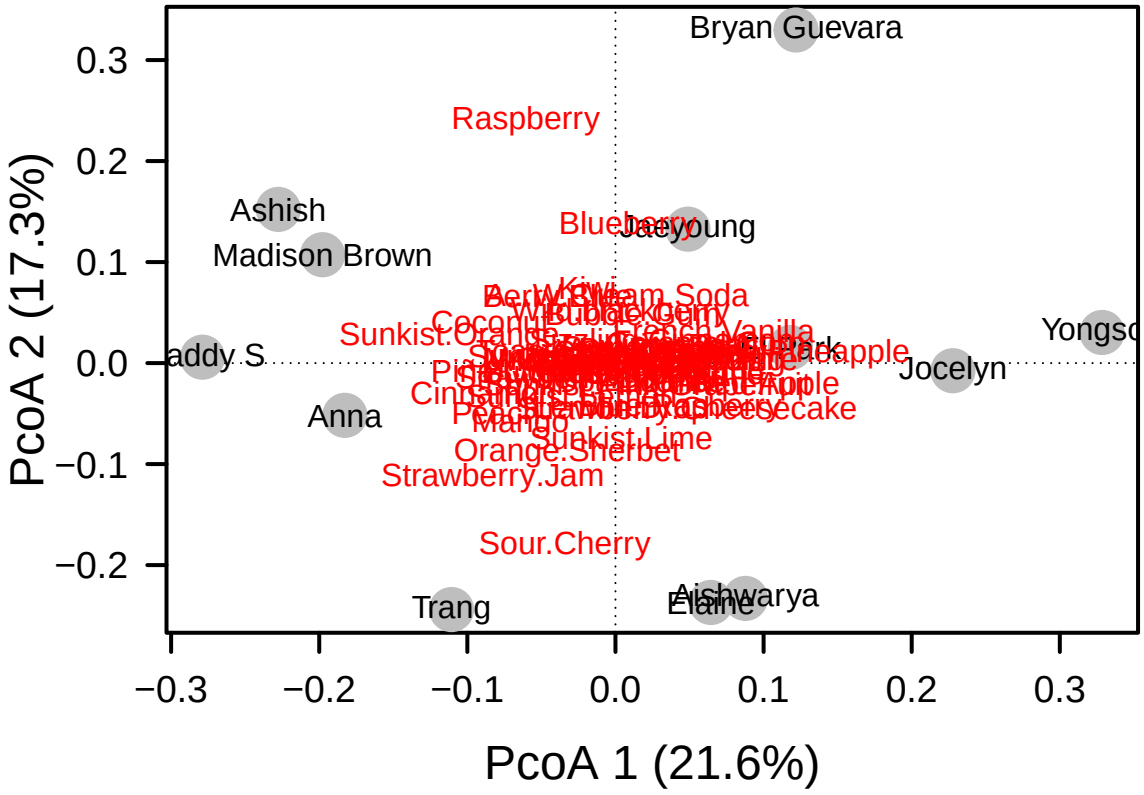
Question 2: Starting with the site-by-species matrix, visualize beta diversity. In the code chunk below, conduct principal coordinates analyses (PCoA) using both an abundance- and incidence-based resemblance matrix. Plot the sample scores in species space using different colors, symbols, or labels. Which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

```
suppressWarnings({
dat.dj <- vegdist(dat, method = "jaccard", binary = TRUE)
dat.pcoa <- cmdscale(dat.dj, eig = TRUE, k = 3)
explainvar1 <- round(dat.pcoa$eig[1] / sum(dat.pcoa$eig), 3) * 100
explainvar2 <- round(dat.pcoa$eig[2] / sum(dat.pcoa$eig), 3) * 100
explainvar3 <- round(dat.pcoa$eig[3] / sum(dat.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

par(mar = c(5, 5, 1, 2) + 0.1)
plot(dat.pcoa$points[ ,1], dat.pcoa$points[ ,2],
     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)
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(dat.pcoa$points[ ,1], dat.pcoa$points[ ,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(dat.pcoa$points[ ,1], dat.pcoa$points[ ,2],
     labels = row.names(dat.pcoa$points))
```

```r
datREL <- dat
   for(i in 1:nrow(dat)){
     datREL[i, ] = dat[i, ] / sum(dat[i, ])
   }
dat.pcoa <- add.spec.scores.class(dat.pcoa, datREL, method = "pcoa.scores")
text(dat.pcoa$cproj[ ,1], dat.pcoa$cproj[ ,2],
     labels = row.names(dat.pcoa$cproj), col = "red")

spe.corr <- add.spec.scores.class(dat.pcoa, datREL, method = "cor.scores")$cproj
corrcut <- 0.7
imp.spp <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]
print(imp.spp)
    fit <- envfit(dat.pcoa, datREL, perm = 999)
    print(fit)
})
```




```
##                  Dim1       Dim2       Dim3
## Cantaloupe  0.7534984 -0.1026134 -0.3690934
## <NA>               NA         NA         NA
## Cinnamon   -0.8140563 -0.2414250 -0.2748045
## Red.Apple   0.8437566 -0.1706459  0.2263436
##
## ***VECTORS
##
##                            Dim1     Dim2     r2 Pr(>r)
## A...W.Cream.Soda         0.01527  0.99988 0.4466  0.071 .
## A...W.Root.Beer         -0.97245 -0.23312 0.0288  0.885
## Berry.Blue              -0.52357  0.85198 0.4112  0.092 .
```

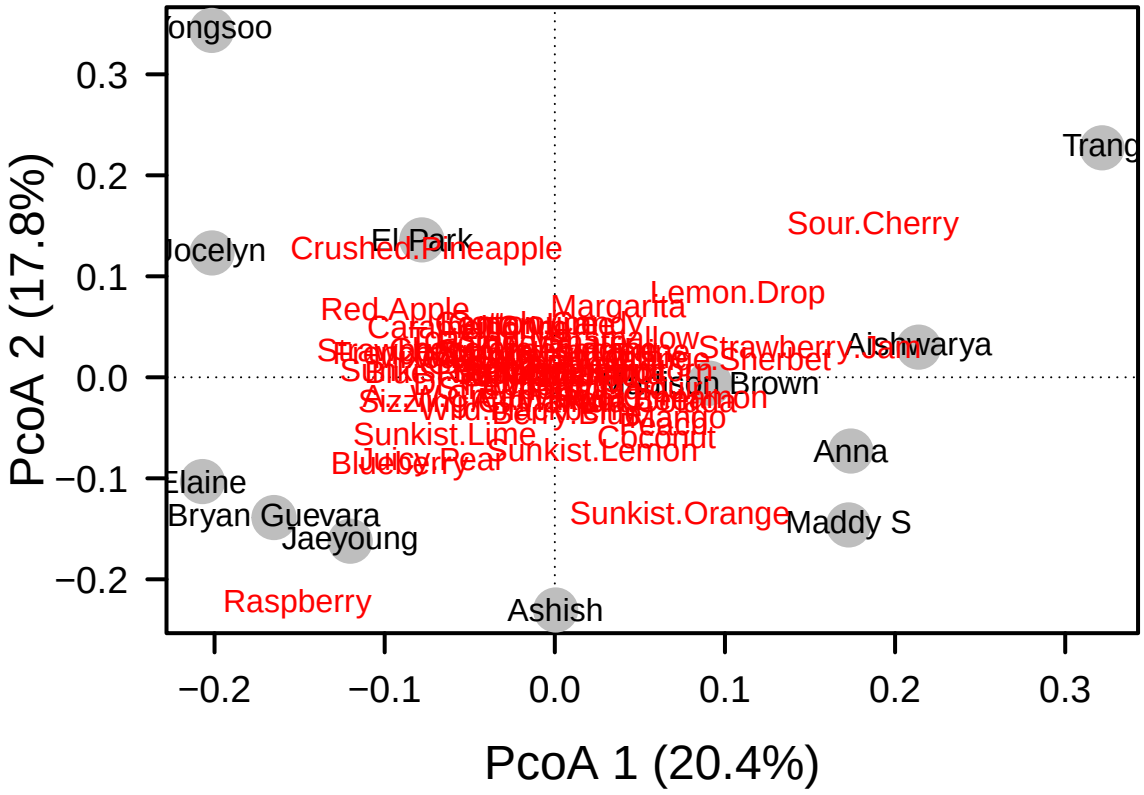
```
## Blueberry                0.06088  0.99815 0.4040  0.067 .
## Bubble.Gum               0.23279  0.97253 0.1808  0.403
## Buttered.Popcorn        -0.51051 -0.85987 0.1062  0.599
## Cantaloupe               0.98865 -0.15026 0.5783  0.008 **
## Cappuccino               0.28944 -0.95720 0.1969  0.497
## Caramel.Corn             0.94817  0.31776 0.5327  0.027 *
## Chili.Mango              0.00000  0.00000 0.0000  1.000
## Chocolate.Pudding        0.99840  0.05660 0.4557  0.013 *
## Cinnamon                -0.94935 -0.31422 0.7210  0.004 **
## Coconut                 -0.89981  0.43627 0.4568  0.068 .
## Cotton.Candy             0.94357  0.33116 0.1103  0.596
## Crushed.Pineapple        0.99606  0.08870 0.3957  0.047 *
## Dr..Pepper               0.91558  0.40213 0.2196  0.340
## French.Vanilla           0.90161  0.43255 0.5186  0.037 *
## Green.Apple              0.83470 -0.55070 0.0002  0.997
## Island.Punch             0.98714  0.15984 0.0372  1.000
## Juicy.Pear               0.94466  0.32806 0.0695  0.723
## Kiwi                    -0.24664  0.96911 0.3365  0.151
## Lemon.Drop               0.07653 -0.99707 0.0307  0.884
## Lemon.Lime               0.97403 -0.22643 0.3598  0.125
## Licorice                 0.90697 -0.42118 0.4358  0.055 .
## Mango                   -0.73153 -0.68181 0.2409  0.293
## Margarita               -0.99994 -0.01064 0.0392  0.882
## Mixed.Berry.Smoothie     0.99836  0.05732 0.1290  0.579
## Orange.Sherbet          -0.34453 -0.93878 0.1561  0.498
## Peach                   -0.85193 -0.52365 0.3670  0.127
## Piña.Colada             -0.99312 -0.11710 0.4119  0.071 .
## Plum                     0.69983 -0.71431 0.0075  0.978
## Pomegranate              0.90112  0.43357 0.1854  0.421
## Raspberry               -0.24450  0.96965 0.4223  0.099 .
## Red.Apple                0.97546 -0.22018 0.7410  0.003 **
## Sizzling.Cinnamon        0.83921  0.54380 0.2288  0.307
## Sour.Cherry             -0.18605 -0.98254 0.3951  0.081 .
## Strawberry.Cheesecake    0.71493 -0.69919 0.3603  0.133
## Strawberry.Daiquiri     -0.71064 -0.70356 0.1169  0.667
## Strawberry.Jam          -0.58948 -0.80778 0.1813  0.452
## Sunkist.Lemon           -0.67848 -0.73462 0.0387  0.842
## Sunkist.Lime             0.05469 -0.99850 0.1338  0.662
## Sunkist.Orange          -0.97436  0.22498 0.3382  0.178
## Sunkist.Pink.Grapefruit  0.67563 -0.73724 0.1377  0.544
## Sunkist.Tangerine       -0.95659  0.29145 0.0164  0.922
## Toasted.Marsmallow       0.31906  0.94773 0.0268  0.878
## Top.Banana               0.42184 -0.90667 0.0657  0.737
## Tutti.Fruitti            0.64454  0.76457 0.0018  0.988
## Very.Cherry              0.85257  0.52261 0.1582  0.477
## Watermelon               0.95380  0.30044 0.1214  0.556
## Wild.blackberry          0.06433  0.99793 0.3749  0.139
## Blue.Rasberry            0.70082 -0.71334 0.3533  0.148
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```r
suppressWarnings({
dat.db <- vegdist(dat, method = "bray", upper = TRUE, diag = TRUE)
dat.pcoa <- cmdscale(dat.db, eig = TRUE, k = 3)
explainvar1 <- round(dat.pcoa$eig[1] / sum(dat.pcoa$eig), 3) * 100
explainvar2 <- round(dat.pcoa$eig[2] / sum(dat.pcoa$eig), 3) * 100
explainvar3 <- round(dat.pcoa$eig[3] / sum(dat.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

par(mar = c(5, 5, 1, 2) + 0.1)
plot(dat.pcoa$points[ ,1], dat.pcoa$points[ ,2],
     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)
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(dat.pcoa$points[ ,1], dat.pcoa$points[ ,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(dat.pcoa$points[ ,1], dat.pcoa$points[ ,2],
     labels = row.names(dat.pcoa$points))

datREL <- dat
   for(i in 1:nrow(dat)){
     datREL[i, ] = dat[i, ] / sum(dat[i, ])
   }
dat.pcoa <- add.spec.scores.class(dat.pcoa, datREL,method = "pcoa.scores")
text(dat.pcoa$cproj[ ,1], dat.pcoa$cproj[ ,2],
     labels = row.names(dat.pcoa$cproj), col = "red")
spe.corr <- add.spec.scores.class(dat.pcoa, datREL, method = "cor.scores")$cproj
corrcut <- 0.7
imp.spp <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]
print(imp.spp)

    fit <- envfit(dat.pcoa, datREL, perm = 999)
    print(fit)
})
```




```
##                  Dim1       Dim2       Dim3
## Cantaloupe -0.3548269  0.7580758 -0.2066158
## <NA>               NA         NA         NA
## Cinnamon    0.7285807 -0.1553521  0.2845498
## Licorice   -0.3699198  0.7411796  0.1555568
## Red.Apple  -0.7572700  0.4902194 -0.2707572
##
## ***VECTORS
##
##                          Dim1     Dim2     r2 Pr(>r)
## A...W.Cream.Soda       -0.89319 -0.44967 0.1744  0.413
## A...W.Root.Beer         0.65385 -0.75662 0.1208  0.566
## Berry.Blue              0.18531 -0.98268 0.1051  0.600
## Blueberry              -0.72143 -0.69249 0.3779  0.112
## Bubble.Gum             -0.74613  0.66580 0.0085  0.958
## Buttered.Popcorn        0.97311  0.23032 0.2931  0.193
## Cantaloupe             -0.40079  0.91617 0.7006  0.006 **
## Cappuccino              0.98746  0.15790 0.1310  0.676
## Caramel.Corn           -0.71317  0.70099 0.5944  0.011 *
## Chili.Mango             0.00000  0.00000 0.0000  1.000
## Chocolate.Pudding      -0.58845  0.80853 0.6583  0.006 **
## Cinnamon                0.97495 -0.22243 0.5550  0.025 *
## Coconut                 0.71298 -0.70119 0.3318  0.162
## Cotton.Candy           -0.10558  0.99441 0.3877  0.108
## Crushed.Pineapple      -0.51309  0.85833 0.6290  0.006 **
## Dr..Pepper             -0.97379 -0.22745 0.4077  0.099 .
## French.Vanilla         -0.94613  0.32378 0.5200  0.032 *
## Green.Apple            -0.87564  0.48296 0.1057  0.604
## Island.Punch           -0.44836  0.89385 0.0764  0.905
```

```
## Juicy.Pear              -0.65475 -0.75585 0.3442  0.150
## Kiwi                     0.53679 -0.84372 0.0327  0.857
## Lemon.Drop               0.79666  0.60443 0.2860  0.206
## Lemon.Lime              -0.29156  0.95655 0.5125  0.034 *
## Licorice                -0.42274  0.90625 0.6862  0.007 **
## Mango                    0.86334 -0.50462 0.2175  0.331
## Margarita                0.48440  0.87485 0.1717  0.462
## Mixed.Berry.Smoothie    -0.62910  0.77733 0.1936  0.412
## Orange.Sherbet           0.98649  0.16380 0.1848  0.441
## Peach                    0.81267 -0.58272 0.2392  0.275
## Piña.Colada              0.92603 -0.37745 0.3147  0.195
## Plum                     0.46063 -0.88759 0.0233  0.913
## Pomegranate             -0.38122  0.92448 0.1989  0.376
## Raspberry               -0.55868 -0.82938 0.5298  0.046 *
## Red.Apple               -0.82207  0.56939 0.8138  0.002 **
## Sizzling.Cinnamon       -0.82505 -0.56506 0.3291  0.159
## Sour.Cherry              0.78012  0.62563 0.7361  0.004 **
## Strawberry.Cheesecake   -0.85184  0.52380 0.1708  0.428
## Strawberry.Daiquiri      0.65912 -0.75204 0.0653  0.825
## Strawberry.Jam           0.98391  0.17866 0.2272  0.262
## Sunkist.Lemon            0.29528 -0.95541 0.1041  0.583
## Sunkist.Lime            -0.76039 -0.64947 0.1947  0.308
## Sunkist.Orange           0.47286 -0.88114 0.5270  0.035 *
## Sunkist.Pink.Grapefruit -0.98730  0.15887 0.1319  0.549
## Sunkist.Tangerine        0.20534  0.97869 0.0362  0.868
## Toasted.Marsmallow      -0.01500  0.99989 0.2469  0.284
## Top.Banana               0.99314  0.11693 0.0365  0.818
## Tutti.Fruitti           -0.53303 -0.84610 0.0162  0.927
## Very.Cherry             -0.99388  0.11042 0.0229  0.909
## Watermelon              -0.82304  0.56799 0.1632  0.442
## Wild.blackberry         -0.40508 -0.91428 0.2208  0.342
## Blue.Rasberry           -0.99896  0.04563 0.2574  0.264
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

Answer 2: PCoA analysis based on bundance-based matrix and incidence-based matrix show very different results. However, when I identified influential species by cutoff of 0.7, these especies are highly overlapped. In incidence-based matrix, the influential species are Cantaloque, Cinnamon, and Red.Apple and in abundance-baed matrix, these are Cantaloque, Cinnamon, Red.Apple and Licorice. Based on these reaults, although I think both matrices provide very informative information, I'll choose the abundance-based matrix which can capture more significant species when I canducted permutatin test.

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).

```
pref <- read.csv("./data/pref.csv",header = TRUE, row.names = 1)
pref1 <- colSums(pref, na.rm = TRUE)
max_votes <- 0
popular <- c()

for (i in names(pref1)) {
  if (pref1[i] > max_votes) {
```

```
    max_votes <- pref1[i]
    popular <- i
  } else if (pref1[i] == max_votes) {
    popular <- c(popular, i)
  }
}

print(popular)
```

```
## [1] "Berry.Blue"
```

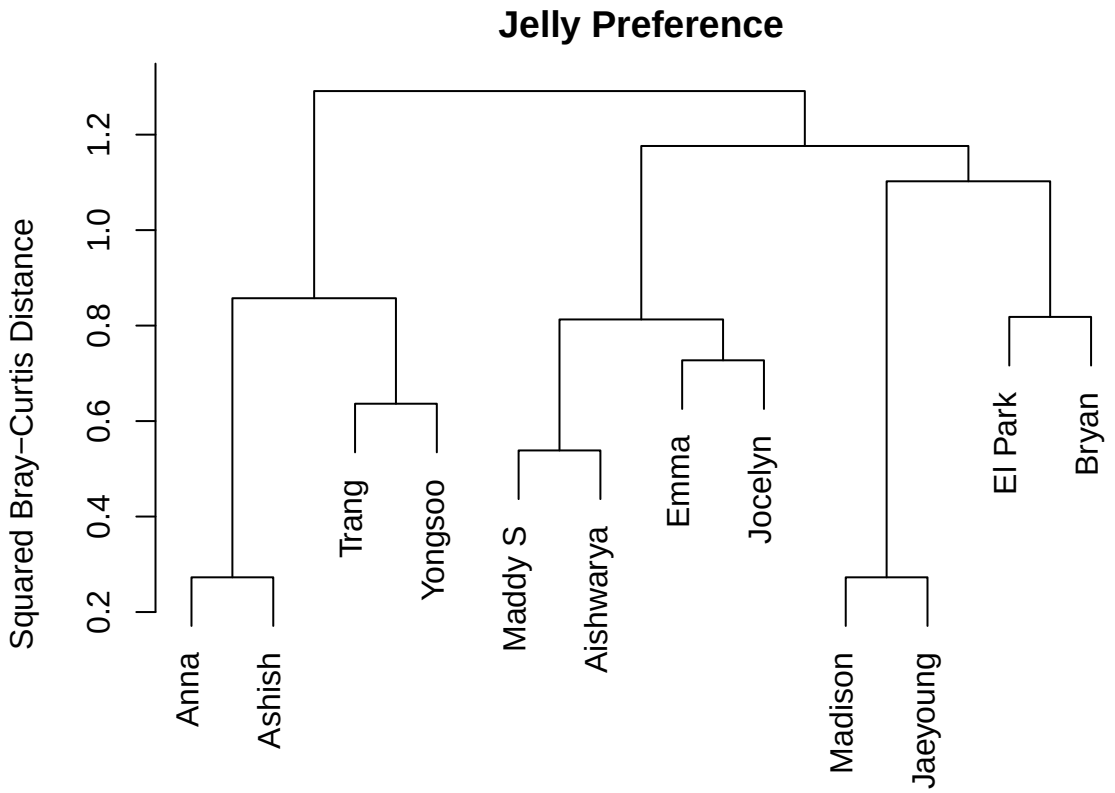
Answer 3: Berry.Blue is the most popular jelly bean in the class room.

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.

```
pref <- na.omit(pref)
pref.db <- vegdist(pref, method = "bray", diag = TRUE, upper = TRUE, binary = TRUE)
print(pref.db)
```

```
##             El Park     Trang   Madison      Emma   Maddy S      Anna  Jaeyoung
## El Park   0.0000000 0.8333333 0.8333333 0.8947368 1.0000000 1.0000000 0.6923077
## Trang     0.8333333 0.0000000 0.8000000 0.6470588 1.0000000 0.7777778 0.8181818
## Madison   0.8333333 0.8000000 0.0000000 0.7647059 1.0000000 1.0000000 0.2727273
## Emma      0.8947368 0.6470588 0.7647059 0.0000000 0.7647059 0.7500000 0.6666667
## Maddy S   1.0000000 1.0000000 1.0000000 0.7647059 0.0000000 0.7777778 1.0000000
## Anna      1.0000000 0.7777778 1.0000000 0.7500000 0.7777778 0.0000000 0.8000000
## Jaeyoung  0.6923077 0.8181818 0.2727273 0.6666667 1.0000000 0.8000000 0.0000000
## Jocelyn   0.8823529 0.8666667 0.7333333 0.7272727 0.8666667 1.0000000 0.7500000
## Bryan     0.8181818 1.0000000 1.0000000 0.8750000 1.0000000 1.0000000 1.0000000
## Aishwarya 0.8666667 0.8461538 0.8461538 0.6000000 0.5384615 0.8333333 0.8571429
## Yongsoo   1.0000000 0.6363636 0.8181818 0.8888889 0.8181818 0.6000000 0.8333333
## Ashish    0.8571429 0.8333333 0.8333333 0.7894737 0.6666667 0.2727273 0.6923077
##             Jocelyn     Bryan Aishwarya   Yongsoo    Ashish
## El Park   0.8823529 0.8181818 0.8666667 1.0000000 0.8571429
## Trang     0.8666667 1.0000000 0.8461538 0.6363636 0.8333333
## Madison   0.7333333 1.0000000 0.8461538 0.8181818 0.8333333
## Emma      0.7272727 0.8750000 0.6000000 0.8888889 0.7894737
## Maddy S   0.8666667 1.0000000 0.5384615 0.8181818 0.6666667
## Anna      1.0000000 1.0000000 0.8333333 0.6000000 0.2727273
## Jaeyoung  0.7500000 1.0000000 0.8571429 0.8333333 0.6923077
## Jocelyn   0.0000000 0.8571429 0.6666667 0.7500000 1.0000000
## Bryan     0.8571429 0.0000000 1.0000000 1.0000000 1.0000000
## Aishwarya 0.6666667 1.0000000 0.0000000 1.0000000 0.8666667
## Yongsoo   0.7500000 1.0000000 1.0000000 0.0000000 0.5384615
## Ashish    1.0000000 1.0000000 0.8666667 0.5384615 0.0000000
```

```
dat.ward <- hclust(pref.db, method = "ward.D2")
par(mar = c(1, 5, 2, 2) + 0.1)
plot(dat.ward, main = "Jelly Preference",
     ylab = "Squared Bray-Curtis Distance")
```



Answer 4: Based on Bray-Curtis distance, the one who has the lowest dissimilarity with me is Ashish (0.5384615). However, when I plotted the Ward's cluster, I was clustered with trang which has 0.6363636 dissimilarity because Ashish has much lower dissimilarity with Anna. According to the dendrogram, our class room's preference is divided into two major clusters which have dissimilarity of about 1 to each other.

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)**.