8. Worksheet: Among Site (Beta) Diversity – Part 1

Thomas Zambiasi; Z620: Quantitative Biodiversity, Indiana University

01 February, 2023

OVERVIEW

In this worksheet, we move beyond the investigation of within-site α-diversity. We will explore β-diversity, which is defined as the diversity that occurs among sites. This requires that we examine the compositional similarity of assemblages that vary in space or time.

After completing this exercise you will know how to:

1. formally quantify β-diversity
2. visualize β-diversity with heatmaps, cluster analysis, and ordination
3. test hypotheses about β-diversity using multivariate statistics

Directions:

1. In the Markdown version of this document in your cloned repo, change "Student Name" on line 3 (above) with 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 examples of proper scripting needed to carry out the exercises.
4. Answer questions in the worksheet. Space for your answers is provided in this document and is 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 today, it is *imperative* that you **push** this file to your GitHub repo, at whatever stage you are. This will enable you to pull your work onto your own computer.
6. When you have completed the worksheet, **Knit** the text and code into a single PDF file by pressing the `Knit` button in the RStudio scripting panel. This will save the PDF output in your '6.BetaDiversity' folder.
7. After Knitting, please submit the worksheet by making a **push** to your GitHub repo and then create a **pull request** via GitHub. Your pull request should include this file (**6.BetaDiversity_1_Worksheet.Rmd**) with all code blocks filled out and questions answered) and the PDF output of `Knitr` (**6.BetaDiversity_1_Worksheet.pdf**).

The completed exercise is due on **Wednesday, February 1st, 2023 before 12:00 PM (noon)**.

1) R SETUP

Typically, the first thing you will do in either an R script or an RMarkdown file is setup your environment. This includes things such as setting the working directory and loading any packages that you will need.

In the R code chunk below, provide the code to:

1. clear your R environment,
2. print your current working directory,
3. set your working directory to your "*/6.BetaDiversity*" folder, and
4. load the `vegan` R package (be sure to install if needed).

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

```
## [1] "C:/Users/tmzam/GitHub/QB2023_Zambiasi/2.Worksheets/6.BetaDiversity"
```

```
setwd("C:/Users/tmzam/GitHub/QB2023_Zambiasi/2.Worksheets/6.BetaDiversity")
library("vegan")
```

```
## Warning: package 'vegan' was built under R version 4.2.2
```

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

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

```
## This is vegan 2.6-4
```

```
library("ade4")
library("viridis")
```

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

```
library("gplots")
```

```
## Warning: package 'gplots' was built under R version 4.2.2
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
#library("BiodiversityR"); might need a different package for PCoA
library("indicspecies")
```

```
## Warning: package 'indicspecies' was built under R version 4.2.2
```

2) LOADING DATA

Load dataset

In the R code chunk below, do the following:

1. load the `doubs` dataset from the `ade4` package, and
2. explore the structure of the dataset.

```
# note, pleae do not print the dataset when submitting
data(doubs)
str(doubs,max.level=1)
```

```
## List of 4
##  $ env    :'data.frame': 30 obs. of  11 variables:
##  $ fish   :'data.frame': 30 obs. of  27 variables:
##  $ xy     :'data.frame': 30 obs. of  2 variables:
##  $ species:'data.frame': 27 obs. of  4 variables:
```

```
head(doubs$env)
```

```
##   dfs alt   slo flo pH har pho nit amm oxy bdo
## 1   3 934 6.176  84 79  45   1  20   0 122  27
## 2  22 932 3.434 100 80  40   2  20  10 103  19
## 3 102 914 3.638 180 83  52   5  22   5 105  35
## 4 185 854 3.497 253 80  72  10  21   0 110  13
## 5 215 849 3.178 264 81  84  38  52  20  80  62
## 6 324 846 3.497 286 79  60  20  15   0 102  53
```

Question 1: Describe some of the attributes of the `doubs` dataset.

a. How many objects are in `doubs`?
b. How many fish species are there in the `doubs` dataset?
c. How many sites are in the `doubs` dataset?

Answer 1a: The doubs dataset includes 4 objects, all of which are dataframes. ***Answer 1b***: There are 27 fish species present in this dataset.
Answer 1c: This dataset contains data for 30 different sites.

Visualizing the Doubs River Dataset

Question 2: Answer the following questions based on the spatial patterns of richness (i.e., α-diversity) and Brown Trout (*Salmo trutta*) abundance in the Doubs River.

a. How does fish richness vary along the sampled reach of the Doubs River?
b. How does Brown Trout (*Salmo trutta*) abundance vary along the sampled reach of the Doubs River?
c. What do these patterns say about the limitations of using richness when examining patterns of biodiversity?

Answer 2a: Based on the graphs present in the worksheet, fish species richness appears to be much higher downstream (low- to mid- 20s) than upstream (singles to low-teens). ***Answer 2b***: Brown trout are much more abundant upstream and were very sparsely sampled downstream. ***Answer 2c***: These contrasting patterns show that spp. richness is not a great metric to look at biodiversity in some cases because not all species may follow the overall community trends. As seen with the brown trout in this example, some species may be present in larger populations in habitats that appear to be less diverse if richness is being considered as a metric of diversity.

3) QUANTIFYING BETA-DIVERSITY

In the R code chunk below, do the following:

1. write a function (`beta.w()`) to calculate Whittaker's β-diversity (i.e., β_w) that accepts a site-by-species matrix with optional arguments to specify pairwise turnover between two sites, and
2. use this function to analyze various aspects of β-diversity in the Doubs River.

```
beta.w<-function(site.by.species="",sitenum1="",sitenum2="",pairwise=FALSE){
  #IF we specify pairwise as true, function should run this part:
  if(pairwise==TRUE){
    #code to print an error if we don't give necessary arguments
    if(sitenum1==""|sitenum2==""){
      print("Error: you need to specify some sites to compare")
      return(NA)
    }
    #now code for calculating pairwise beta diversity
    site1=site.by.species[sitenum1,] #selecting site 1
    site2=site.by.species[sitenum2,] #selecting site 2
    site1=subset(site1,select=site1>0) #removes absences in site 1
    site2=subset(site2,select=site2>0) #removes absences in site 2
    gamma=union(colnames(site1),colnames(site2)) #creates gamma species pool
    s=length(gamma) #gamma richness
    a.bar=mean(c(specnumber(site1),specnumber(site2))) #mean sample richness
    b.w=round(s/a.bar-1,3)
    return(b.w)
  }
  #otherwise, function defaults to false and runs this:
  SbyS.pa<-decostand(site.by.species,method="pa") #convert to presence/absence
  S<-ncol(SbyS.pa[,which(colSums(SbyS.pa)>0)]) #number of species in the region
  a.bar<-mean(specnumber(SbyS.pa)) #average richness at each site
  b.w<-round(S/a.bar,3) #rounding to 3 decimals
  return(b.w)
}

#First, finding general beta diversity for Doubs river data
beta.w(doubs$fish)
```

```
## [1] 2.16
```

```
#Now finding pairwise beta diversity for sites 1 and 2 (0 = minimum Bw, 1 = max.)
beta.w(doubs$fish,"1","2",TRUE)
```

```
## [1] 0.5
```

```
#Finding pairwise beta diversity for sites 1 and 10
beta.w(doubs$fish,"1","10",TRUE)
```

```
## [1] 0.714
```

Question 3: Using your `beta.w()` function above, answer the following questions:

a. Describe how local richness (α) and turnover (β) contribute to regional (γ) fish diversity in the Doubs.
b. Is the fish assemblage at site 1 more similar to the one at site 2 or site 10?
c. Using your understanding of the equation $\beta_w = \gamma/\alpha$, how would your interpretation of β change if we instead defined beta additively (i.e., $\beta = \gamma - \alpha$)?

Answer 3a: Since beta diversity (turnover) is defined as gamma (regional) diversity divided by mean local (alpha) diversity; we can say that alpha and beta diversity both correlate positively to gamma diversity. Higher average values of local species richness will result in increased values of regional diversity. The higher turnover is as well (i.e. the more differentiated community compositions are between local sites), the greater regional diversity there will be.
Answer 3b: The pairwise beta diversity comparisons will return results ranging from 0 to 1, with 0 being the minimum diversity value (sites are identical in community composition) and 1 being the maximum (sites are completely different in community composition). The pairwise beta diversity between sites 1 and 2 is 0.5, while it is 0.714 between sites 1 and 10. I would conclude from this that the fish community at site 2 is more similar to site 1 than site 10 is. ***Answer 3c***: If beta diversity were defined additively, it would then be equal to regional diversity - average local diversity. If this were the case, minimum beta diversity (identical richness across all sites) would be seen by a value of zero for both pairwise and overall comparisons (regional species richness would be the same as average local richness). Maximum beta diversity (or any high values) would be more difficult to work with, since those values would change depending on the number of species present at local and regional levels (regions with fewer species, like the arctic, would have lower B-diversity compared to regions like the tropics). An additive measure of beta diversity wouldn't be able to compare the ratios of regional diversity to average local diversity, making it more difficult to assess differences in biodiversity across all scales.

The Resemblance Matrix

In order to quantify β-diversity for more than two samples, we need to introduce a new primary ecological data structure: the **Resemblance Matrix**.

Question 4: How do incidence- and abundance-based metrics differ in their treatment of rare species?

Answer 4: Incidence-based metrics do not account for how rare or common species are in the communities being compared and instead focus on the number of species at each site. Abundance-based metrics do account for how common a species may be in a community. Metrics such as the Bray-Curtis Dissimilarity metric and Morisita-Horn calculate ratios based on differences in abundance of each species at multiple sites being compared. More abundant species in each of these would have a greater effect on the values determined by these metrics than would rarer species, unlike the incidence metrics where the addition of a rare or common species would have the same effect.

In the R code chunk below, do the following:

1. make a new object, `fish`, containing the fish abundance data for the Doubs River,
2. remove any sites where no fish were observed (i.e., rows with sum of zero),
3. construct a resemblance matrix based on Sørensen's Similarity ("fish.ds"), and
4. construct a resemblance matrix based on Bray-Curtis Distance ("fish.db").

```
#making fish abundance data its own object
fish<-doubs$fish
#removing sites with no fish observed
fish<-fish[rowSums(fish)!=0,]
#code for resemblance matrix (Sørensen)
```

```
fish.ds<-vegdist(fish,method="bray",binary=TRUE)
fish.ds
```

```
##            1          2          3          4          5          6          7
## 2  0.50000000
## 3  0.60000000 0.14285714
## 4  0.77777778 0.45454545 0.33333333
## 5  0.83333333 0.57142857 0.46666667 0.15789474
## 6  0.81818182 0.53846154 0.42857143 0.11111111 0.04761905
## 7  0.66666667 0.25000000 0.33333333 0.38461538 0.37500000 0.33333333
## 9  1.00000000 0.50000000 0.55555556 0.38461538 0.37500000 0.33333333 0.40000000
## 10 0.71428571 0.33333333 0.40000000 0.28571429 0.29411765 0.25000000 0.09090909
## 11 0.71428571 0.33333333 0.40000000 0.42857143 0.52941176 0.50000000 0.27272727
## 12 0.71428571 0.33333333 0.40000000 0.42857143 0.52941176 0.50000000 0.27272727
## 13 0.71428571 0.33333333 0.40000000 0.57142857 0.64705882 0.62500000 0.45454545
## 14 0.81818182 0.53846154 0.42857143 0.33333333 0.42857143 0.40000000 0.46666667
## 15 0.83333333 0.57142857 0.60000000 0.36842105 0.36363636 0.33333333 0.37500000
## 16 0.88888889 0.70000000 0.61904762 0.36000000 0.28571429 0.25925926 0.54545455
## 17 0.91304348 0.76000000 0.69230769 0.46666667 0.39393939 0.37500000 0.62962963
## 18 0.91666667 0.76923077 0.70370370 0.48387097 0.41176471 0.39393939 0.64285714
## 19 1.00000000 0.84615385 0.77777778 0.54838710 0.41176471 0.45454545 0.71428571
## 20 1.00000000 0.84000000 0.76923077 0.53333333 0.39393939 0.43750000 0.70370370
## 21 1.00000000 0.84615385 0.77777778 0.54838710 0.41176471 0.45454545 0.71428571
## 22 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000 0.77777778
## 23 1.00000000 1.00000000 1.00000000 0.81818182 0.71428571 0.69230769 0.75000000
## 24 1.00000000 1.00000000 1.00000000 0.75000000 0.68421053 0.66666667 0.84615385
## 25 1.00000000 1.00000000 0.83333333 0.62500000 0.36842105 0.44444444 0.69230769
## 26 1.00000000 0.91666667 0.84000000 0.58620690 0.43750000 0.48387097 0.76923077
## 27 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000 0.77777778
## 28 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000 0.77777778
## 29 0.92592593 0.79310345 0.73333333 0.52941176 0.40540541 0.44444444 0.67741935
## 30 1.00000000 1.00000000 0.92000000 0.65517241 0.50000000 0.54838710 0.84615385
##            9         10         11         12         13         14         15
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10 0.45454545
## 11 0.45454545 0.33333333
## 12 0.45454545 0.33333333 0.00000000
## 13 0.63636364 0.50000000 0.16666667 0.16666667
## 14 0.60000000 0.37500000 0.25000000 0.25000000 0.25000000
## 15 0.50000000 0.29411765 0.29411765 0.29411765 0.29411765 0.14285714
## 16 0.54545455 0.47826087 0.56521739 0.56521739 0.56521739 0.33333333 0.28571429
## 17 0.62962963 0.57142857 0.57142857 0.57142857 0.57142857 0.37500000 0.33333333
## 18 0.64285714 0.58620690 0.58620690 0.58620690 0.58620690 0.39393939 0.35294118
## 19 0.64285714 0.65517241 0.79310345 0.79310345 0.79310345 0.57575758 0.52941176
## 20 0.62962963 0.64285714 0.78571429 0.78571429 0.85714286 0.62500000 0.57575758
## 21 0.64285714 0.65517241 0.79310345 0.79310345 0.86206897 0.63636364 0.58823529
## 22 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143 0.68750000 0.63636364
```

```
## 23 0.50000000 0.77777778 0.77777778 0.77777778 1.00000000 0.84615385 0.85714286
## 24 0.69230769 0.71428571 0.85714286 0.85714286 1.00000000 0.77777778 0.78947368
## 25 0.69230769 0.57142857 0.85714286 0.85714286 1.00000000 0.66666667 0.68421053
## 26 0.69230769 0.70370370 0.85185185 0.85185185 0.92592593 0.67741935 0.62500000
## 27 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143 0.68750000 0.63636364
## 28 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143 0.68750000 0.63636364
## 29 0.67741935 0.62500000 0.68750000 0.68750000 0.68750000 0.50000000 0.45945946
## 30 0.76923077 0.77777778 0.92592593 0.92592593 1.00000000 0.74193548 0.68750000
##            16         17         18         19         20         21         22
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17 0.12820513
## 18 0.15000000 0.02222222
## 19 0.25000000 0.15555556 0.13043478
## 20 0.28205128 0.18181818 0.15555556 0.02222222
## 21 0.30000000 0.20000000 0.17391304 0.04347826 0.02222222
## 22 0.33333333 0.22727273 0.20000000 0.06666667 0.04545455 0.02222222
## 23 0.80000000 0.76000000 0.76923077 0.76923077 0.76000000 0.76923077 0.76000000
## 24 0.68000000 0.60000000 0.54838710 0.48387097 0.46666667 0.48387097 0.46666667
## 25 0.60000000 0.60000000 0.54838710 0.48387097 0.46666667 0.48387097 0.46666667
## 26 0.36842105 0.25581395 0.22727273 0.09090909 0.06976744 0.04545455 0.02325581
## 27 0.33333333 0.22727273 0.20000000 0.06666667 0.04545455 0.02222222 0.00000000
## 28 0.33333333 0.22727273 0.20000000 0.06666667 0.04545455 0.02222222 0.00000000
## 29 0.25581395 0.12500000 0.10204082 0.06122449 0.08333333 0.06122449 0.08333333
## 30 0.36842105 0.25581395 0.22727273 0.09090909 0.06976744 0.04545455 0.02325581
##            23         24         25         26         27         28         29
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
```

```
## 19
## 20
## 21
## 22
## 23
## 24 0.45454545
## 25 0.45454545 0.37500000
## 26 0.75000000 0.44827586 0.44827586
## 27 0.76000000 0.46666667 0.46666667 0.02325581
## 28 0.76000000 0.46666667 0.46666667 0.02325581 0.00000000
## 29 0.79310345 0.52941176 0.52941176 0.10638298 0.08333333 0.08333333
## 30 0.75000000 0.44827586 0.44827586 0.04761905 0.02325581 0.02325581 0.10638298
```

```
#code for resemblance matrix (Bray-Curtis)
fish.db<-vegdist(fish,method="bray")
fish.db
```

```
##            1          2          3          4          5          6          7
## 2  0.60000000
## 3  0.68421053 0.14285714
## 4  0.75000000 0.33333333 0.18918919
## 5  0.89189189 0.69565217 0.68000000 0.49090909
## 6  0.75000000 0.39393939 0.29729730 0.19047619 0.41818182
## 7  0.68421053 0.14285714 0.12500000 0.24324324 0.64000000 0.24324324
## 9  1.00000000 0.69230769 0.73333333 0.65714286 0.58333333 0.54285714 0.66666667
## 10 0.88235294 0.38461538 0.40000000 0.37142857 0.54166667 0.25714286 0.26666667
## 11 0.57142857 0.30434783 0.40740741 0.43750000 0.68888889 0.43750000 0.33333333
## 12 0.71428571 0.20000000 0.23529412 0.33333333 0.69230769 0.38461538 0.17647059
## 13 0.72727273 0.29032258 0.31428571 0.45000000 0.73584906 0.55000000 0.37142857
## 14 0.80645161 0.40000000 0.31818182 0.34693878 0.67741935 0.42857143 0.36363636
## 15 0.83333333 0.51111111 0.46938776 0.40740741 0.55223881 0.37037037 0.38775510
## 16 0.86046512 0.65384615 0.57142857 0.47540984 0.45945946 0.37704918 0.53571429
## 17 0.91489362 0.67857143 0.63333333 0.50769231 0.51282051 0.44615385 0.60000000
## 18 0.95555556 0.74074074 0.72413793 0.58730159 0.50000000 0.52380952 0.68965517
## 19 1.00000000 0.79310345 0.70967742 0.61194030 0.50000000 0.52238806 0.67741935
## 20 1.00000000 0.91176471 0.88888889 0.74025974 0.48888889 0.68831169 0.86111111
## 21 1.00000000 0.94594595 0.92307692 0.78313253 0.50000000 0.73493976 0.89743590
## 22 1.00000000 0.97619048 0.95454545 0.82795699 0.52830189 0.78494624 0.93181818
## 23 1.00000000 1.00000000 1.00000000 0.92000000 0.89473684 0.84000000 0.90000000
## 24 1.00000000 1.00000000 1.00000000 0.88888889 0.79591837 0.77777778 0.93548387
## 25 1.00000000 1.00000000 0.92592593 0.81250000 0.68888889 0.68750000 0.85185185
## 26 1.00000000 0.96363636 0.93220339 0.78125000 0.55844156 0.68750000 0.89830508
## 27 1.00000000 0.97333333 0.94936709 0.83333333 0.56701031 0.76190476 0.92405063
## 28 1.00000000 0.97560976 0.95348837 0.82417582 0.57692308 0.78021978 0.93023256
## 29 0.97777778 0.93939394 0.92233010 0.81481481 0.53719008 0.77777778 0.90291262
## 30 1.00000000 1.00000000 0.98095238 0.87272727 0.59349593 0.83636364 0.96190476
##             9         10         11         12         13         14         15
## 2
## 3
## 4
## 5
## 6
## 7
## 9
```

```
## 10 0.57142857
## 11 0.76000000 0.44000000
## 12 0.68750000 0.37500000 0.24137931
## 13 0.81818182 0.57575758 0.33333333 0.18918919
## 14 0.76190476 0.47619048 0.43589744 0.21739130 0.19148936
## 15 0.65957447 0.40425532 0.50000000 0.33333333 0.38461538 0.24590164
## 16 0.70370370 0.51851852 0.64705882 0.55172414 0.59322034 0.44117647 0.26027397
## 17 0.68965517 0.51724138 0.63636364 0.58064516 0.61904762 0.50000000 0.40259740
## 18 0.64285714 0.57142857 0.69811321 0.66666667 0.70491803 0.60000000 0.46666667
## 19 0.66666667 0.63333333 0.82456140 0.75000000 0.81538462 0.67567568 0.56962025
## 20 0.68571429 0.77142857 0.91044776 0.89189189 0.92000000 0.83333333 0.70786517
## 21 0.76315789 0.81578947 0.91780822 0.92500000 0.95061728 0.86666667 0.76842105
## 22 0.76744186 0.86046512 0.95180723 0.95555556 0.97802198 0.90000000 0.77142857
## 23 0.77777778 0.88888889 0.86666667 0.90909091 1.00000000 0.93750000 0.94594595
## 24 0.72413793 0.79310345 0.92307692 0.93939394 1.00000000 0.90697674 0.87500000
## 25 0.84000000 0.76000000 0.90909091 0.93103448 1.00000000 0.84615385 0.81818182
## 26 0.71929825 0.82456140 0.92592593 0.93442623 0.96774194 0.85915493 0.76315789
## 27 0.76623377 0.84415584 0.94594595 0.95061728 0.97560976 0.89010989 0.77083333
## 28 0.76190476 0.85714286 0.95061728 0.95454545 0.97752809 0.89795918 0.78640777
## 29 0.78217822 0.84158416 0.89795918 0.90476190 0.90566038 0.84347826 0.73333333
## 30 0.84466019 0.90291262 0.98000000 0.98130841 1.00000000 0.93162393 0.81967213
##            16         17         18         19         20         21         22
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17 0.26190476
## 18 0.34146341 0.13953488
## 19 0.39534884 0.31111111 0.25000000
## 20 0.58333333 0.42000000 0.32653061 0.23529412
## 21 0.62745098 0.49056604 0.40384615 0.29629630 0.10169492
## 22 0.66071429 0.55172414 0.47368421 0.38983051 0.18750000 0.10447761
## 23 0.90909091 0.83333333 0.82608696 0.84000000 0.86666667 0.87878788 0.89473684
## 24 0.81818182 0.69491525 0.64912281 0.63934426 0.57746479 0.61038961 0.65517241
## 25 0.76470588 0.74545455 0.66037736 0.61403509 0.67164179 0.69863014 0.73493976
## 26 0.63855422 0.54022989 0.45882353 0.32584270 0.21212121 0.20000000 0.25217391
## 27 0.66990291 0.57009346 0.48571429 0.37614679 0.19327731 0.13600000 0.12592593
## 28 0.69090909 0.57894737 0.50000000 0.41379310 0.22222222 0.16666667 0.12676056
## 29 0.65354331 0.51145038 0.44186047 0.41353383 0.24475524 0.18120805 0.11949686
## 30 0.72093023 0.57894737 0.52671756 0.48148148 0.29655172 0.23178808 0.18012422
##            23         24         25         26         27         28         29
## 2
## 3
## 4
```

```
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24 0.57894737
## 25 0.46666667 0.46153846
## 26 0.82978723 0.48275862 0.59259259
## 27 0.88059701 0.61538462 0.70270270 0.18867925
## 28 0.89189189 0.64705882 0.72839506 0.23893805 0.09774436
## 29 0.91208791 0.70588235 0.77551020 0.33846154 0.18666667 0.14649682
## 30 0.91397849 0.71153846 0.78000000 0.36363636 0.19736842 0.15723270 0.14772727
```

Question 5: Using the distance matrices from above, answer the following questions:

a. Does the resemblance matrix (`fish.db`) represent similarity or dissimilarity? What information in the resemblance matrix led you to arrive at your answer?
b. Compare the resemblance matrices (`fish.db` or `fish.ds`) you just created. How does the choice of the Sørensen or Bray-Curtis distance influence your interpretation of site (dis)similarity?

Answer 5a: The fish.db resemblance matrix was created via Bray-Curtis Dissimilarity, so it represents dissimilarity. The resemblance matrix gives values for % dissimilarity between sites. Assuming the sites are sequential along the river, those nearest to a site of interest are the least dissimilar (closer to 0) and those furthest are the most (closer to 1). We can see this when looking at the column for site 1; sites 2 and 3 have the lowest value, while nearly all of those much further along the river (19 - 30) had dissimilarity values of 1.
Answer 5b: Both the Sørensen and Bray-Curtis metrics measure dissimilarity (0 being identical, 1 being completely different), so the outputs of both of these resemblance matrices are very similar. For any pairwise comparisons where the site communities are completely distinct, both metrics give an output of 1. Otherwise, the Sørensen matrix tends to output lower percent differences. This is likely because the Sørensen metric gives more weight to species that are shared between sites than Bray-Curtis does, so it will make it seem like communities that have any common composition are more alike.

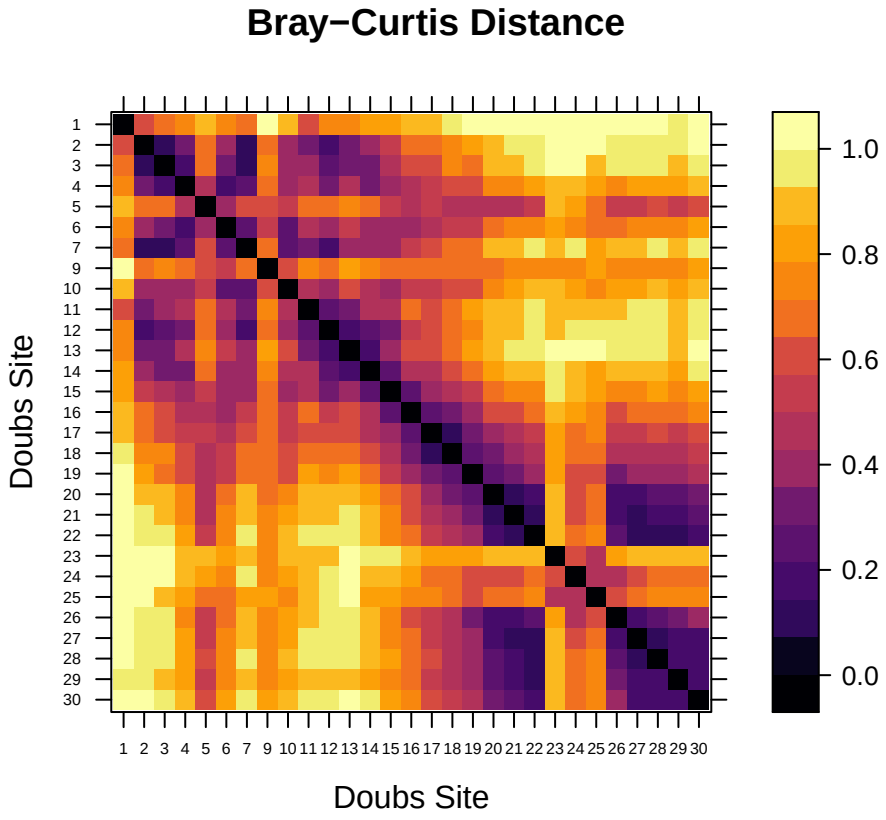
4) VISUALIZING BETA-DIVERSITY

A. Heatmaps

In the R code chunk below, do the following:

1. define a color palette,
2. define the order of sites in the Doubs River, and
3. use the `levelplot()` function to create a heatmap of fish abundances in the Doubs River.

```
#making color palette
palette<-inferno(20)
#Defining order of sites in Doubs
order<-rev(attr(fish.db,"Labels"))
#creating heatmap plot
levelplot(as.matrix(fish.db)[,order],aspect="iso",col.regions=palette,xlab="Doubs Site",ylab="Doubs Site
```

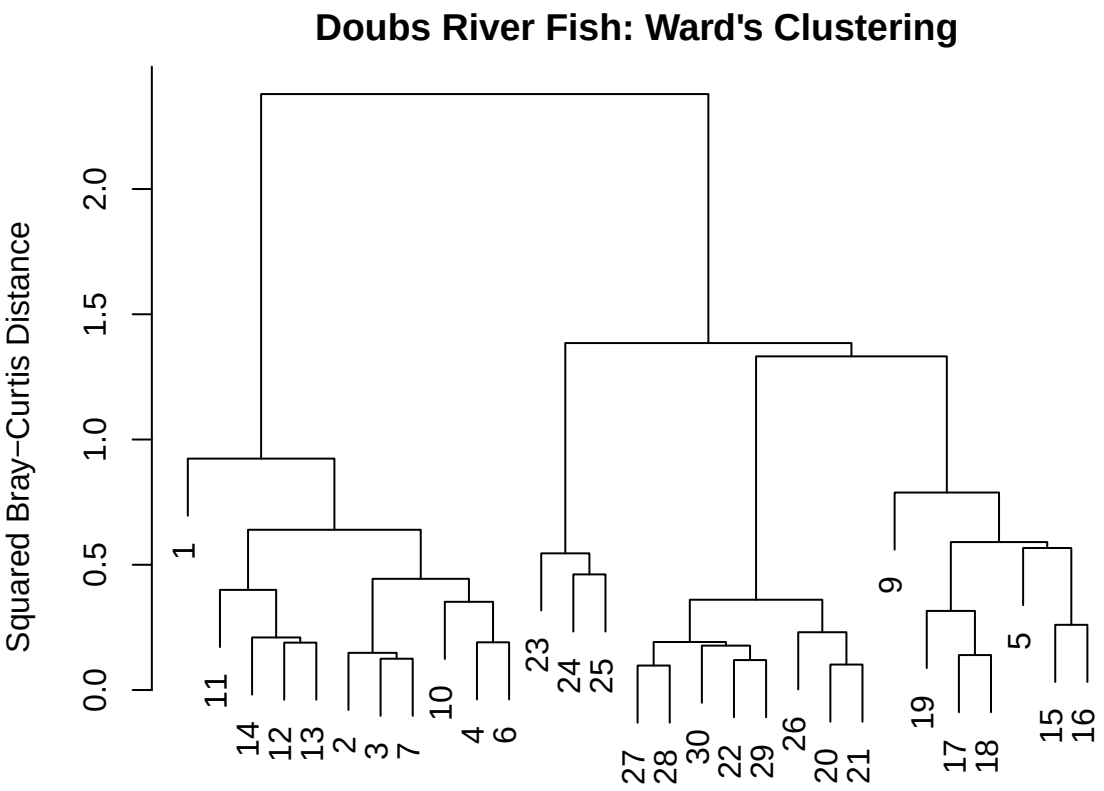


B. Cluster Analysis

In the R code chunk below, do the following:

1. perform a cluster analysis using Ward's Clustering, and
2. plot your cluster analysis (use either `hclust` or `heatmap.2`).

```
#cluster analysis code
fish.cluster<-hclust(fish.db,method="ward.D2")
#plotting cluster analysis
par(mar=c(1,5,2,2)+0.1)
plot(fish.cluster,main="Doubs River Fish: Ward's Clustering",ylab="Squared Bray-Curtis Distance")
```



Question 6: Based on cluster analyses and the introductory plots that we generated after loading the data, develop an ecological hypothesis for fish diversity the `doubs` data set?

> ***Answer 6***: The cluster analysis and heat map seem to indicate that there are two major groupings of collection sites in the Doubs river. These figures seem to indicate that community composition of any particular site is more similar to sites within its same group than to sites in the other. However, I would hypothesize that the cluster containing the most sites would have a higher beta diversity than the other. The right-hand cluster in the figure above has more sites, so I would predict that it would have the higher beta diversity than the left cluster. This may be the result of more resources in the river around these sites that support a greater network of trophic interactions.

C. Ordination

Principal Coordinates Analysis (PCoA)

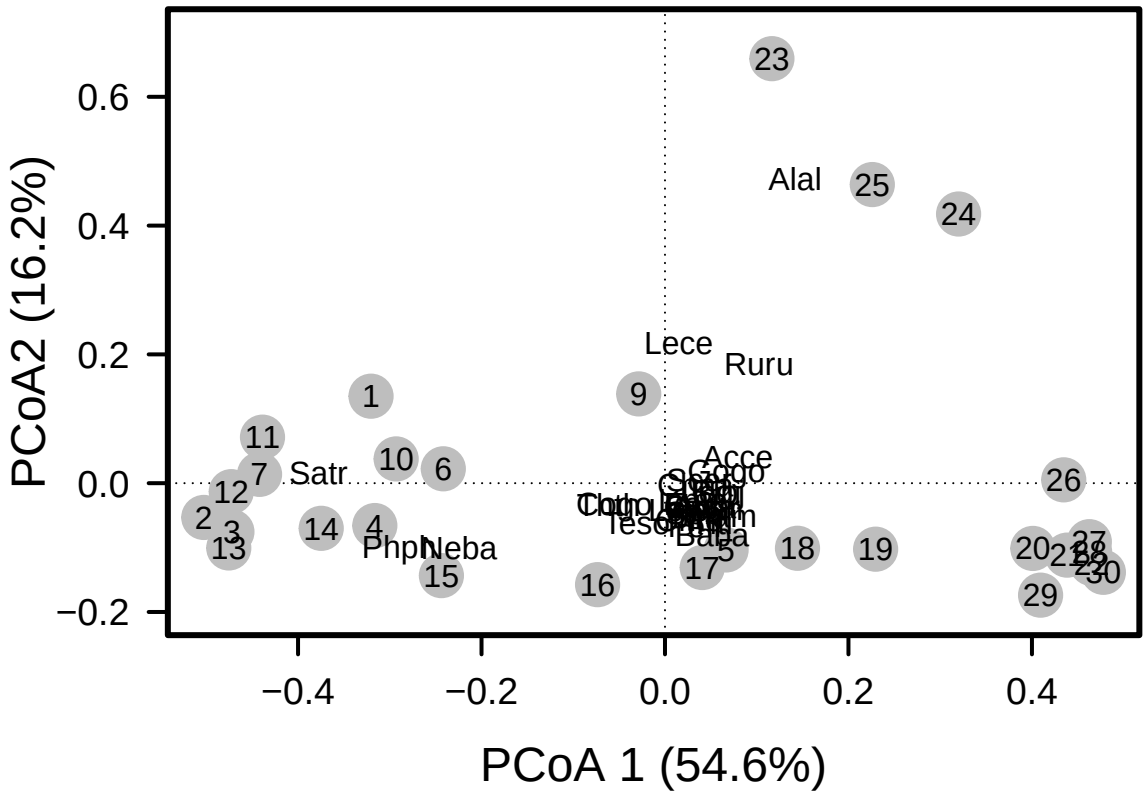
In the R code chunk below, do the following:

1. perform a Principal Coordinates Analysis to visualize beta-diversity
2. calculate the variation explained by the first three axes in your ordination
3. plot the PCoA ordination,
4. label the sites as points using the Doubs River site number, and
5. identify influential species and add species coordinates to PCoA plot.

```
#Code for PCoA
fish.pcoa<-cmdscale(fish.db,eig=TRUE,k=3)
#summing variation explained by first three axes
exvar1<-round(fish.pcoa$eig[1]/sum(fish.pcoa$eig),3)*100
exvar2<-round(fish.pcoa$eig[2]/sum(fish.pcoa$eig),3)*100
exvar3<-round(fish.pcoa$eig[3]/sum(fish.pcoa$eig),3)*100
sum.var<-sum(exvar1,exvar2,exvar3)
sum.var
```

```
## [1] 81.3
```

```
#plotting PCoA
#defining plot parameters
par(mar=c(5,5,1,2)+0.1)
#starting plot
plot(fish.pcoa$points[,1],fish.pcoa$points[,2],ylim=c(-0.2,0.7),
     xlab=paste("PCoA 1 (",exvar1, "%)",sep=""),
     ylab=paste("PCoA2 (",exvar2, "%)",sep=""),
     pch=16,cex=2.0,type="n",cex.lab=1.5,cex.axis=1.2,axes=FALSE)
#adding 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=3)
#adding points and labels
points(fish.pcoa$points[,1],fish.pcoa$points[,2],pch=19,cex=3,bg="gray",
       col="gray")
text(fish.pcoa$points[,1],fish.pcoa$points[,2],labels=row.names(fish.pcoa$points))
#finding relative abundance of each spp. at each site
fishREL<-fish
  for(i in 1:nrow(fish)){
    fishREL[i,]=fish[i,]/sum(fish[i,])
  }
#reading in new spec.score function
source("C:/Users/tmzam/OneDrive/Documents/R/spec.scores.function.R")
#calculating and adding spp. scores to graph
fish.pcoa<-add.spec.scores.class(fish.pcoa,fishREL,method="pcoa.scores")
text(fish.pcoa$cproj[,1],fish.pcoa$cproj[,2],
     labels=row.names(fish.pcoa$cproj),col="black")
```



In the R code chunk below, do the following:

1. identify influential species based on correlations along each PCoA axis (use a cutoff of 0.70), and
2. use a permutation test (999 permutations) to test the correlations of each species along each axis.

```
#identifying influential species
spp.corr<-add.spec.scores.class(fish.pcoa,fishREL,method="cor.scores")$cproj
corrcut<-0.7 #specified cutoff value
impspp<-spp.corr[abs(spp.corr[,1])>=corrcut|abs(spp.corr[,2])>=corrcut,]
impspp
```

```
##            Dim1       Dim2        Dim3
## Phph -0.8674640 -0.1699316 -0.12463098
## Neba -0.7674114 -0.1855678 -0.36963830
## Rham  0.8088751 -0.4192567  0.14136301
## Legi  0.8201759 -0.1701803  0.12423941
## Cyca  0.7595122 -0.4442926  0.17313658
## Abbr  0.7704744 -0.3452714  0.29277803
## Acce  0.7635195  0.2155765  0.10288179
## Blbj  0.8118483 -0.1324698  0.25581178
## Alal  0.4471283  0.8119843 -0.05167131
## Anan  0.7974122 -0.3918972  0.20944968
```

```
#permutation test
fit<-envfit(fish.pcoa,fishREL,perm=999)
fit
```

```
##
## ***VECTORS
##
##          Dim1     Dim2     r2 Pr(>r)
## Cogo -0.83884 -0.54438 0.2982  0.008 **
## Satr -0.99904  0.04371 0.4326  0.005 **
## Phph -0.94110 -0.33813 0.7814  0.001 ***
## Neba -0.91413 -0.40543 0.6234  0.001 ***
## Thth -0.87692 -0.48063 0.2634  0.023 *
## Teso -0.44704 -0.89452 0.1700  0.093 .
## Chna  0.99707 -0.07644 0.4612  0.002 **
## Chto  0.42032 -0.90738 0.2579  0.027 *
## Lele  0.33041 -0.94384 0.0495  0.563
## Lece  0.06856  0.99765 0.3399  0.012 *
## Baba  0.54118 -0.84091 0.6752  0.001 ***
## Spbi  0.57341 -0.81927 0.4138  0.001 ***
## Gogo  0.97507  0.22188 0.3753  0.003 **
## Eslu  0.72044 -0.69352 0.1673  0.093 .
## Pefl  0.43762 -0.89916 0.3048  0.013 *
## Rham  0.72476 -0.68901 0.8301  0.001 ***
## Legi  0.93461 -0.35568 0.7016  0.001 ***
## Scer  0.98569  0.16858 0.3533  0.009 **
## Cyca  0.68181 -0.73153 0.7743  0.001 ***
## Titi  0.64378 -0.76521 0.4586  0.001 ***
## Abbr  0.77254 -0.63497 0.7128  0.001 ***
## Icme  0.75626 -0.65427 0.5270  0.001 ***
## Acce  0.88799  0.45986 0.6294  0.001 ***
## Ruru  0.48379  0.87518 0.5177  0.002 **
## Blbj  0.95802 -0.28671 0.6766  0.001 ***
## Alal  0.28755  0.95777 0.8592  0.001 ***
## Anan  0.74277 -0.66954 0.7894  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

Question 7: Address the following questions about the ordination results of the `doubs` data set:

a. Describe the grouping of sites in the Doubs River based on fish community composition.
b. Generate a hypothesis about which fish species are potential indicators of river quality.

Answer 7a: Based on the PCoA visualization, it looks like there's three distinct site groupings in the river. The smallest, sites 23, 24, and 25, looks to be characterized by populations of bleak (*Alburnus alburnus*; Alal). The next smallest site grouping (nearly identical to the left side of the cluster analysis) includes sites 1, 2, 3, 4, 6, 7, 10, 11, 12, 13, 14, and 15. This grouping of sites is characterized by their inclusion of brown trout (*Salmo trutta fario*; Satr), minnow (*Phoxinus phoxinus*; Phph), and stone loach (*Nemacheilus barbatulus*; Neba). The remaining sites (5, 9, 16, 17, 18, 19, 20, 21, 22, 26, 27, 28, 29, and 30) make up the broad remaining group and appear to host the most diverse communities of fish. Two fish species, chub (*Leuciscus cephalus cephalus*; Lece) and roach (*Rutilus rutilus*; Ruru), are found on the ordination visual between this largest group and the small grouping (sites 23-25) and may be particularly present in both of these clusters.

Answer 7b: Rivers tend to be less polluted near their headwaters; in the Doubs river, upstream

fish communities seem to have notable populations of brown trout, minnow, and stone loach. Based on this information, I would hypothesize that presence of brown trout, minnow, and stone loach in a river may be a sign of high water quality.

SYNTHESIS

Load the dataset from that you and your partner are using for the team project. Use one of the tools introduced in the beta diversity module to visualize your data. Describe any interesting patterns and identify a hypothesis is relevant to the principles of biodiversity.

```
#loading dataset
kbsdata1<-read.csv("kbs_set_1.csv")
#removing first row (has extra unneeded characters)
kbsdata1<-kbsdata1[-1,]
#isolating relevant data columns
kbsdata2<-kbsdata1[,c("replicate","disturbed","fertilized",
                      "species.name","biomass_g_m2")]
#sites will be determined by rep+treatment combo
#need to combine rep+treatment combos into single label column
kbsdata2$site<-paste(kbsdata2$replicate,kbsdata2$disturbed,kbsdata2$fertilized)
#rearranging data again
kbsdata3<-kbsdata2[,c("site","species.name","biomass_g_m2")]
#creating list of sites
kbssitelist<-as.data.frame(unique(kbsdata3$site))
#creating list of species
kbsspplist<-unique(kbsdata3$species.name)
#reworking into dataframe
kbsspplist<-as.data.frame(kbsspplist)
#renaming column to be able to join matrix components easier
colnames(kbsspplist)<-c("species.name")
#need to load dpylr to do joining
library("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
```

```
#creating site-specific data and adding to sXspp. by spp. name, steps
#repeat for each site
#r1, dist. unfert.
R1du<-kbsdata3[kbsdata3$site=="R1 disturbed unfertilized",] #isolating site data
R1du<-R1du[,2:3] #removing site column
sXspp<-left_join(kbsspplist,R1du,by="species.name") #joining to spp. list
colnames(sXspp)<-c("species.name","R1du") #renaming sXspp columns
#r1, dist. fert.
```

```
R1df<-kbsdata3[kbsdata3$site=="R1 disturbed fertilized",]
R1df<-R1df[,2:3]
sXspp<-left_join(sXspp,R1df,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df")
#r1, undist. unfert.
R1uu<-kbsdata3[kbsdata3$site=="R1 undisturbed unfertilized",]
R1uu<-R1uu[,2:3]
sXspp<-left_join(sXspp,R1uu,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu")
#r1, undist. fert.
R1uf<-kbsdata3[kbsdata3$site=="R1 undisturbed fertilized",]
R1uf<-R1uf[,2:3]
sXspp<-left_join(sXspp,R1uf,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf")
#r2, dist. unfert.
R2du<-kbsdata3[kbsdata3$site=="R2 disturbed unfertilized",]
R2du<-R2du[,2:3]
sXspp<-left_join(sXspp,R2du,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du")
#r2, dist. fert.
R2df<-kbsdata3[kbsdata3$site=="R2 disturbed fertilized",]
R2df<-R2df[,2:3]
sXspp<-left_join(sXspp,R2df,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df")
#r2, undist. unfert.
R2uu<-kbsdata3[kbsdata3$site=="R2 undisturbed unfertilized",]
R2uu<-R2uu[,2:3]
sXspp<-left_join(sXspp,R2uu,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                   "R2uu")
#r2, undist. fert.
R2uf<-kbsdata3[kbsdata3$site=="R2 undisturbed fertilized",]
R2uf<-R2uf[,2:3]
sXspp<-left_join(sXspp,R2uf,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                   "R2uu","R2uf")
#r3, dist. unfert.
R3du<-kbsdata3[kbsdata3$site=="R3 disturbed unfertilized",]
R3du<-R3du[,2:3]
sXspp<-left_join(sXspp,R3du,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                   "R2uu","R2uf","R3du")
#r3, dist. fert.
R3df<-kbsdata3[kbsdata3$site=="R3 disturbed fertilized",]
R3df<-R3df[,2:3]
sXspp<-left_join(sXspp,R3df,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                   "R2uu","R2uf","R3du","R3df")
#r3, undist. unfert.
R3uu<-kbsdata3[kbsdata3$site=="R3 undisturbed unfertilized",]
R3uu<-R3uu[,2:3]
sXspp<-left_join(sXspp,R3uu,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
```

```
                    "R2uu","R2uf","R3du","R3df","R3uu")
#r3, undist. fert.
R3uf<-kbsdata3[kbsdata3$site=="R3 undisturbed fertilized",]
R3uf<-R3uf[,2:3]
sXspp<-left_join(sXspp,R3uf,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf")
#r4, dist. unfert.
R4du<-kbsdata3[kbsdata3$site=="R4 disturbed unfertilized",]
R4du<-R4du[,2:3]
sXspp<-left_join(sXspp,R4du,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du")
#r4, dist. fert.
R4df<-kbsdata3[kbsdata3$site=="R4 disturbed fertilized",]
R4df<-R4df[,2:3]
sXspp<-left_join(sXspp,R4df,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df")
#r4, undist. unfert.
R4uu<-kbsdata3[kbsdata3$site=="R4 undisturbed unfertilized",]
R4uu<-R4uu[,2:3]
sXspp<-left_join(sXspp,R4uu,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu")
#r4, undist. fert.
R4uf<-kbsdata3[kbsdata3$site=="R4 undisturbed fertilized",]
R4uf<-R4uf[,2:3]
sXspp<-left_join(sXspp,R4uf,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu","R4uf")
#r5, dist. unfert.
R5du<-kbsdata3[kbsdata3$site=="R5 disturbed unfertilized",]
R5du<-R5du[,2:3]
sXspp<-left_join(sXspp,R5du,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu","R4uf","R5du")
#r5, dist. fert.
R5df<-kbsdata3[kbsdata3$site=="R5 disturbed fertilized",]
R5df<-R5df[,2:3]
sXspp<-left_join(sXspp,R5df,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu","R4uf","R5du","R5df")
#r5, undist. unfert.
R5uu<-kbsdata3[kbsdata3$site=="R5 undisturbed unfertilized",]
R5uu<-R5uu[,2:3]
sXspp<-left_join(sXspp,R5uu,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
```

```
                    "R4uu","R4uf","R5du","R5df","R5uu")
#r5, undist. fert.
R5uf<-kbsdata3[kbsdata3$site=="R5 undisturbed fertilized",]
R5uf<-R5uf[,2:3]
sXspp<-left_join(sXspp,R5uf,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu","R4uf","R5du","R5df","R5uu","R5uf")
#r6, dist. unfert.
R6du<-kbsdata3[kbsdata3$site=="R6 disturbed unfertilized",]
R6du<-R6du[,2:3]
sXspp<-left_join(sXspp,R6du,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu","R4uf","R5du","R5df","R5uu","R5uf","R6du")
#r6, dist. fert.
R6df<-kbsdata3[kbsdata3$site=="R6 disturbed fertilized",]
R6df<-R6df[,2:3]
sXspp<-left_join(sXspp,R6df,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu","R4uf","R5du","R5df","R5uu","R5uf","R6du","R6df")
#r6, undist. unfert.
R6uu<-kbsdata3[kbsdata3$site=="R6 undisturbed unfertilized",]
R6uu<-R6uu[,2:3]
sXspp<-left_join(sXspp,R6uu,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu","R4uf","R5du","R5df","R5uu","R5uf","R6du","R6df",
                    "R6uu")
#r6, undist. fert.
R6uf<-kbsdata3[kbsdata3$site=="R6 undisturbed fertilized",]
R6uf<-R6uf[,2:3]
sXspp<-left_join(sXspp,R6uf,by="species.name")
colnames(sXspp)<-c("species.name","R1du","R1df","R1uu","R1uf","R2du","R2df",
                    "R2uu","R2uf","R3du","R3df","R3uu","R3uf","R4du","R4df",
                    "R4uu","R4uf","R5du","R5df","R5uu","R5uf","R6du","R6df",
                    "R6uu","R6uf")
#converting column 1 to row names (species names)
row.names(sXspp)<-sXspp$species.name
sXspp<-sXspp[,-1]
#converting all NAs to zeros
sXspp[is.na(sXspp)]<-0
#converting to numeric (columns all chr)
sXspp<-mutate_all(sXspp,function(x)as.numeric(x))
#also need to convert to matrix and transform
sXspp.mtx<-t(as.matrix(sXspp))

#finding b-diversity value across whole sXspp matrix
beta.w(sXspp.mtx,pairwise=FALSE)
```

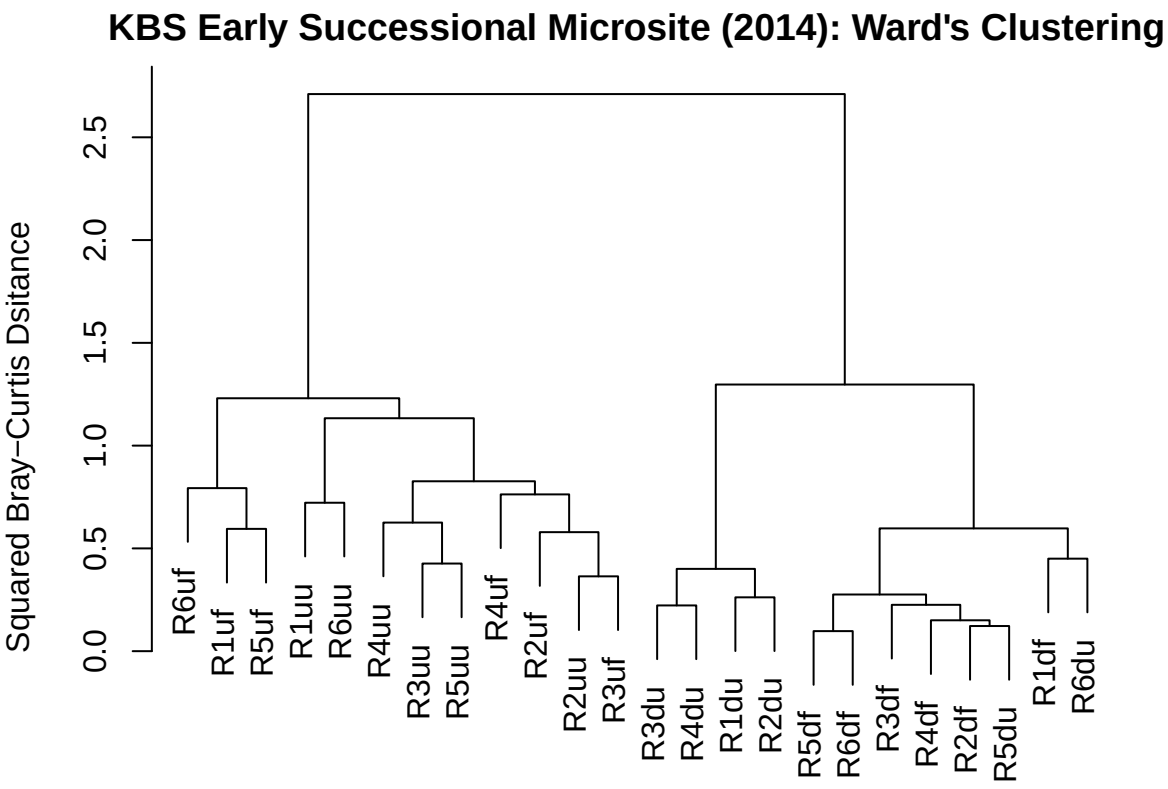
```
## [1] 5.255
```

```r
#building resemblance matrix using Bray-Curtis
kbs.db<-vegdist(sXspp.mtx,method="bray")
kbs.db
```

```
##            R1du       R1df       R1uu       R1uf       R2du       R2df
## R1df 0.69848198
## R1uu 0.99990540 0.99987327
## R1uf 1.00000000 0.99921629 0.78983122
## R2du 0.26199674 0.53054174 0.99991991 1.00000000
## R2df 0.62590889 0.32392392 0.99994877 0.99998097 0.45012252
## R2uu 0.99998463 0.99897721 0.75383526 0.72846799 0.99998625 0.99999008
## R2uf 1.00000000 1.00000000 0.87204800 0.81115993 1.00000000 0.99998518
## R3du 0.30520600 0.74050577 1.00000000 1.00000000 0.29979471 0.65938930
## R3df 0.73317660 0.29482139 0.99993205 1.00000000 0.49867343 0.22194218
## R3uu 1.00000000 0.99997908 0.74507337 0.73862487 1.00000000 1.00000000
## R3uf 1.00000000 1.00000000 0.79071886 0.74524882 1.00000000 0.99998014
## R4du 0.43885433 0.65704935 1.00000000 1.00000000 0.25139136 0.62608374
## R4df 0.72042469 0.39766327 1.00000000 1.00000000 0.53551817 0.14935562
## R4uu 1.00000000 1.00000000 0.74549487 0.69595210 1.00000000 1.00000000
## R4uf 1.00000000 1.00000000 0.85589399 0.78470166 1.00000000 0.99998375
## R5du 0.78246137 0.38454027 0.99978151 0.99994773 0.56993698 0.12238135
## R5df 0.74561735 0.44794423 1.00000000 1.00000000 0.58067508 0.19370436
## R5uu 1.00000000 0.99870335 0.58040077 0.61805779 1.00000000 1.00000000
## R5uf 1.00000000 0.99879499 0.94695735 0.59504232 1.00000000 0.99997637
## R6du 0.57603944 0.45015028 0.99967115 0.99993646 0.32431014 0.44621289
## R6df 0.75754055 0.39026143 0.99953080 0.99995370 0.58143407 0.14494648
## R6uu 0.99889246 0.99812973 0.72208695 0.99212950 0.99904529 0.99936638
## R6uf 1.00000000 0.99891129 0.80172152 0.86699688 1.00000000 1.00000000
##            R2uu       R2uf       R3du       R3df       R3uu       R3uf
## R1df
## R1uu
## R1uf
## R2du
## R2df
## R2uu
## R2uf 0.52257409
## R3du 1.00000000 1.00000000
## R3df 0.99998775 1.00000000 0.63545060
## R3uu 0.51930315 0.72865630 0.99998182 0.99997354
## R3uf 0.36414482 0.54410115 1.00000000 1.00000000 0.59658735
## R4du 1.00000000 1.00000000 0.22250941 0.53144857 0.99998283 1.00000000
## R4df 1.00000000 1.00000000 0.66059958 0.21587086 1.00000000 1.00000000
## R4uu 0.66997054 0.81208594 1.00000000 1.00000000 0.62962485 0.65521818
## R4uf 0.65939794 0.76395716 1.00000000 1.00000000 0.69824277 0.61567316
## R5du 1.00000000 1.00000000 0.69943309 0.15699628 0.99997657 1.00000000
## R5df 1.00000000 1.00000000 0.68518906 0.27871901 1.00000000 1.00000000
## R5uu 0.44128357 0.68697594 1.00000000 1.00000000 0.42641156 0.52624784
## R5uf 0.91339364 0.99655800 1.00000000 1.00000000 0.98643432 0.97908662
## R6du 0.99998657 1.00000000 0.55021057 0.38109958 0.99997076 1.00000000
## R6df 0.99999036 1.00000000 0.78406518 0.27441404 0.99998976 1.00000000
## R6uu 0.90372485 0.99545326 0.99963515 0.99916196 0.75925188 0.93823416
## R6uf 0.78767231 0.89814619 1.00000000 1.00000000 0.84233214 0.69366947
##            R4du       R4df       R4uu       R4uf       R5du       R5df
## R1df
```

```
## R1uu
## R1uf
## R2du
## R2df
## R2uu
## R2uf
## R3du
## R3df
## R3uu
## R3uf
## R4du
## R4df 0.53224433
## R4uu 1.00000000 1.00000000
## R4uf 1.00000000 1.00000000 0.74475712
## R5du 0.62286661 0.13797367 1.00000000 1.00000000
## R5df 0.60988638 0.14165538 1.00000000 1.00000000 0.19844496
## R5uu 1.00000000 1.00000000 0.53056665 0.65706661 0.99978557 1.00000000
## R5uf 1.00000000 1.00000000 0.95918829 0.98702703 1.00000000 1.00000000
## R6du 0.43947611 0.49169583 1.00000000 1.00000000 0.46196963 0.55225945
## R6df 0.70803109 0.18229807 1.00000000 1.00000000 0.17694932 0.09765351
## R6uu 0.99966036 0.99981709 0.93555047 0.97032911 0.99958235 0.99983445
## R6uf 1.00000000 1.00000000 0.80226659 0.77285026 0.99982409 1.00000000
##            R5uu       R5uf       R6du       R6df       R6uu
## R1df
## R1uu
## R1uf
## R2du
## R2df
## R2uu
## R2uf
## R3du
## R3df
## R3uu
## R3uf
## R4du
## R4df
## R4uu
## R4uf
## R5du
## R5df
## R5uu
## R5uf 0.87437195
## R6du 0.99975469 1.00000000
## R6df 0.99980556 1.00000000 0.50378761
## R6uu 0.83317446 0.97810879 0.99883005 0.99898337
## R6uf 0.78846211 0.60768912 0.99980970 0.99965166 0.94982049
```

```
#setting up cluster analysis
kbs.ward<-hclust(kbs.db,method="ward.D2")
#cluster analysis figure
par(mar=c(1,5,2,2)+0.1)
plot(kbs.ward,main="KBS Early Successional Microsite (2014): Ward's Clustering",
     ylab="Squared Bray-Curtis Dsitance")
```



Our datset requires quite a bit of wrangling to create a siteXspp. matrix, so for this synthesis problem we only used the 2014 data from the KBS early successional microplot. This dataset gives species biomass for the different treatments, so we will use this biomass as an approximation of abundance for diversity metrics (this likely won't be a perfect 1:1 to typical diversity metrics since different species may have vastly different biomass per individual, so if we find better approximations throughout the course of the course we'll implement those later on in the project). This dataset originally included some species that couldn't be identified upon collection and were labeled as "surface litter", "unidentified", or "unsorted"; we've removed these from the dataset and will be focusing on diversity of identifiable species. Each site will be defined by combinations of replicate and treatment (ex: Rep1, undisturbed and fertilized). Site IDs are a 4-digit alphanumeric code giving the replicate number (R1, R2, etc.), whether they were disturbed or undisturbed (d, u), and whether they were fertilized or unfertilized (f, u). After building the siteXspp. matix, I found the overall beta diversity metric for all sites. This turned out to be 5.255. I also intended to perform a cluster analysis (via Ward's Clustering), so first built a resemblance matrix using Bray-Curtis distance. The results of the cluster analysis can be seen visually in the plot above. One major pattern that strikes me immediately is that there are two major site groupings: one where plots were disturbed, and one where plots were undisturbed. Organization within each of these groups is much less consistent; many sites that were fertilized or unfertilized appear to have similar community compositions to those with the same nutrient treatment, but there are a few exceptions in both the disturbed and undisturbed categories. Because of the apparent differences in community structure between the two treatment groups, I would hypothesize that disturbance is a major determinant of plant community composition for early successional prairies. Additionally, I would predict that disturbed prairie communities would be less biodiverse than undisturbed prairies. Statistical methods such as a t-test would likely be appropriate for comparing biodiversity just between the two disturbance treatments, but an ANOVA might be more useful for comparing biodiversity across each of the sites. Such

testing may also give more clarity to the effects of the fertilization treatments or differences across replicates.