

Assignment: Among Site (Beta) Diversity

Matt Gibson; Z620: Quantitative Biodiversity, Indiana University

OVERVIEW

In this Assignment, 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. Change “Student Name” on line 3 (above) with your name.
2. Complete as much of the exercise as possible during class; what you do not complete in class will need to be done on your own outside of class.
3. Use the Handout as a guide; it contains a more complete description of data sets along with the proper scripting needed to carry out the exercise.
4. Be sure to **answer the questions** in this exercise document; they also correspond to the Handout. 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 “>”.
5. Before you leave the classroom, **push** this file to your GitHub repo.
6. When you are done with the Assignment, **Knit** the text and code into a html file.
7. After Knitting, please submit the completed Assignment by creating a **pull request** via GitHub. Your pull request should include this file *beta_assignment.Rmd* and the html output of Knitr (*beta_assignment.html*).

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 “/Week3-Beta” folder, and
4. load the **vegan** R package (be sure to install if needed).

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

```
## [1] "C:/Users/matth/Documents/bin/QB2017_Gibson/Week3-Beta"
```

```
setwd("c:/Users/matth/Documents/bin/QB2017_Gibson/Week3-Beta")
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 3.2.5
```

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

```
## Warning: package 'permute' was built under R version 3.2.5
```

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

```
## This is vegan 2.4-2
```

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.

```
package.list <- c('vegan', 'ade4', 'viridis', 'gplots', 'BiodiversityR', 'indicspecies')
for (package in package.list) {
  if (!require(package, character.only=T, quietly=T)){
    install.packages(package)
    library(package, character.only=T)
  }
}
```

```
## Warning: package 'ade4' was built under R version 3.2.5
```

```
##
```

```
## Attaching package: 'ade4'
```

```
## The following object is masked from 'package:vegan':
```

```
##
```

```
##      cca
```

```
## Warning: package 'viridis' was built under R version 3.2.5
```

```
## Warning: package 'gplots' was built under R version 3.2.5
```

```
##
```

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

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

```
##
```

```
##      lowess
```

```
## Warning: package 'BiodiversityR' was built under R version 3.2.5
```

```
## BiodiversityR 2.8-0: Use command BiodiversityRGUI() to launch the Graphical User Interface and to le
```

```
## Warning: package 'indicspecies' was built under R version 3.2.5
```

```
data(doubs)
```

```
str(doubs)
```

```
## List of 4
```

```
## $ env : 'data.frame': 30 obs. of 11 variables:
```

```
## ..$ dfs: num [1:30] 3 22 102 185 215 324 268 491 705 990 ...
```

```
## ..$ alt: num [1:30] 934 932 914 854 849 846 841 792 752 617 ...
```

```
## ..$ slo: num [1:30] 6.18 3.43 3.64 3.5 3.18 ...
```

```
## ..$ flo: num [1:30] 84 100 180 253 264 286 400 130 480 1000 ...
```

```
## ..$ pH : num [1:30] 79 80 83 80 81 79 81 81 80 77 ...
```

```
## ..$ har: num [1:30] 45 40 52 72 84 60 88 94 90 82 ...
```

```
## ..$ pho: num [1:30] 1 2 5 10 38 20 7 20 30 6 ...
```

```
## ..$ nit: num [1:30] 20 20 22 21 52 15 15 41 82 75 ...
```

```
## ..$ amm: num [1:30] 0 10 5 0 20 0 0 12 12 1 ...
```

```
## ..$ oxy: num [1:30] 122 103 105 110 80 102 111 70 72 100 ...
```

```
## ..$ bdo: num [1:30] 27 19 35 13 62 53 22 81 52 43 ...
```

```
## $ fish : 'data.frame': 30 obs. of 27 variables:
```

```
## ..$ Cogo: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Satr: num [1:30] 3 5 5 4 2 3 5 0 0 1 ...
```

```
## ..$ Phph: num [1:30] 0 4 5 5 3 4 4 0 1 4 ...
```

```
## ..$ Neba: num [1:30] 0 3 5 5 2 5 5 0 3 4 ...
```

```
## ..$ Thth: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Teso: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Chna: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Chto: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Lele: num [1:30] 0 0 0 0 5 1 1 0 0 2 ...
```

```
## ..$ Lece: num [1:30] 0 0 0 1 2 2 1 0 5 2 ...
```

```
## ..$ Baba: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Spbi: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Gogo: num [1:30] 0 0 0 1 2 1 0 0 0 1 ...
```

```
## ..$ Eslu: num [1:30] 0 0 1 2 4 1 0 0 0 0 ...
```

```
## ..$ Pefl: num [1:30] 0 0 0 2 4 1 0 0 0 0 ...
```

```
## ..$ Rham: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Legi: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Scer: num [1:30] 0 0 0 0 2 0 0 0 0 0 ...
```

```
## ..$ Cyca: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Titi: num [1:30] 0 0 0 1 3 2 0 0 1 0 ...
```

```
## ..$ Abbr: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Icme: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Acce: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Ruru: num [1:30] 0 0 0 0 5 1 0 0 4 0 ...
```

```
## ..$ Blbj: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Alal: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## ..$ Anan: num [1:30] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ xy : 'data.frame': 30 obs. of 2 variables:
```

```
## ..$ x: num [1:30] 88 94 102 100 106 112 114 110 136 168 ...
```

```
## ..$ y: num [1:30] 7 14 18 28 39 51 61 76 100 112 ...
## $ species:'data.frame': 27 obs. of 4 variables:
## ..$ Scientific: chr [1:27] "Cottus gobio" "Salmo trutta fario" "Phoxinus phoxinus" "Nemacheilus ba
## ..$ French : chr [1:27] "chabot" "truite fario" "vairon" "loche franche" ...
## ..$ English : chr [1:27] "european bullhead" "brown trout" "minnow" "stone loach" ...
## ..$ code : Factor w/ 27 levels "Abbr","Acce",...: 9 22 19 17 26 25 7 8 16 14 ...
```

```
#help(doubs)
```

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

- How many objects are in `doubs`?
- How many fish species are there in the `doubs` dataset?
- How many sites are in the `doubs` dataset?

Answer 1a: 4 objects. An `environment` data frame, a `fish` dataframe which is the site x species matrix, an `xy` dataframe containing the spatial coordinates, and a `species` dataframe containing the scientific and other names of each species. **Answer 1b:** 27 **Answer 1c:** 30

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.

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

Answer 2a: Fish richness is overall higher downstream and midstream than upstream. There is a large increase in richness in the curve of the Doubs river in the middle. **Answer 2b:** Brown trout abundance is higher towards the upstream than the downstream. This difference between the two sides of the river is very extreme... with abundance values of mostly 0 on the downstream side. **Answer 2c:** That it may not capture the entire picture. Richness is depicted as being relatively high in places where brown trout abundance is 0! In other words, richness is a very condensed metric and we may be losing important information.

3) QUANTIFYING BETA-DIVERSITY

In the R code chunk below, do the following:

- 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
- use this function to analyze various aspects of β -diversity in the Doubs River.

```
beta.w <- function(site.by.species = "", sitenum1 = "", sitenum2 = "", pairwise = F){
  if (pairwise == T){

    if (sitenum1 == "" | sitenum2 == ""){
      print("Error: specify sites to compare")
    }
  }
}
```

```

    return(NA)
  }

  site1 = site.by.species[sitenum1,]
  site2 = site.by.species[sitenum2,]
  site1 = subset(site1, select = site1 > 0)
  site2 = subset(site2, select = site2 > 0)
  gamma = union(colnames(site1), colnames(site2))

  s = length(gamma)
  print(s)
  a.bar = mean(c(specnumber(site1), specnumber(site2)))
  print(a.bar)
  #print(a.bar)
  b.w = round(s/a.bar - 1,3)
  return(b.w)
}
else{
  SbyS.pa <- decostand(site.by.species, method = "pa")
  S <- ncol(SbyS.pa[,which(colSums(SbyS.pa) > 0)])
  a.bar <- mean(specnumber(SbyS.pa))
  print(S)
  print(a.bar)
  b.w <- round(S/a.bar, 3)
  return(b.w)
}}

beta.w(doubs$fish)

```

```
## [1] 27
## [1] 12.5
```

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

```
beta.w(doubs$fish, 1, 2, T)
```

```
## [1] 3
## [1] 2
```

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

```
beta.w(doubs$fish, 1, 10, T)
```

```
## [1] 6
## [1] 3.5
```

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

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

- Describe how local richness (α) and turnover (β) contribute to regional (γ) fish diversity in the Doubs.
- Is the fish assemblage at site 1 more similar to the one at site 2 or site 10?
- 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: They are multiplicative. $\gamma = \beta\alpha$. $\alpha = 12.5$ and $\beta = 2.16$. Together, local richness and species turnover leads to higher levels of regional diversity, even when average local richness is relatively low. **Answer 3b:** Site 1 is more similar to site 10 based on the pairwise β_w for the two sites. **Answer 3c:** If it were instead $\beta = \gamma - \alpha$, our estimate for β diversity would not be between 0 and 1. Numbers close to 0, though, would indicate a high degree of similarity between two sites.

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 explicitly take into account the number of shared species between the two sites (a), the number of unique species in site 1 (b) and the number of unique species in site 2 (c). The abundance-based metrics rather take sums of, for example, the difference in abundance between sites for each species. In the abundance-based metrics, the rarity of a species is captured quantitatively.

In the R code chunk below, do the following:

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

```
fish <- doubs$fish
fish <- fish[-8, ]

#ds
fish.ds <- vegdist(fish, method = "bray", binary= T, diag = T)
#db
fish.db <- vegdist(fish, method = "bray", diag = T)

x <- ncol(as.matrix(fish.ds))

fish.ds <- as.matrix(fish.ds)
fish.db <- as.matrix(fish.db)
i <- 0
x <- seq(1, x)
for (i in x){
  print(i)
  print(rbind(fish.ds[,i], fish.db[,i]))
}
```

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

```

##      1      2      3      4      5      6      7 9      10
## [1,] 0 0.5 0.6000000 0.7777778 0.8333333 0.8181818 0.6666667 1 0.7142857
## [2,] 0 0.6 0.6842105 0.7500000 0.8918919 0.7500000 0.6842105 1 0.8823529
##      11      12      13      14      15      16      17
## [1,] 0.7142857 0.7142857 0.7142857 0.8181818 0.8333333 0.8888889 0.9130435
## [2,] 0.5714286 0.7142857 0.7272727 0.8064516 0.8333333 0.8604651 0.9148936
##      18 19 20 21 22 23 24 25 26 27 28      29 30
## [1,] 0.9166667 1 1 1 1 1 1 1 1 1 1 0.9259259 1
## [2,] 0.9555556 1 1 1 1 1 1 1 1 1 1 0.9777778 1
## [1] 2
##      1 2      3      4      5      6      7      9
## [1,] 0.5 0 0.1428571 0.4545455 0.5714286 0.5384615 0.2500000 0.5000000
## [2,] 0.6 0 0.1428571 0.3333333 0.6956522 0.3939394 0.1428571 0.6923077
##      10      11      12      13      14      15      16
## [1,] 0.3333333 0.3333333 0.3333333 0.3333333 0.5384615 0.5714286 0.7000000
## [2,] 0.3846154 0.3043478 0.2000000 0.2903226 0.4000000 0.5111111 0.6538462
##      17      18      19      20      21      22 23 24 25
## [1,] 0.7600000 0.7692308 0.8461538 0.8400000 0.8461538 0.9200000 1 1 1
## [2,] 0.6785714 0.7407407 0.7931034 0.9117647 0.9459459 0.9761905 1 1 1
##      26      27      28      29 30
## [1,] 0.9166667 0.9200000 0.9200000 0.7931034 1
## [2,] 0.9636364 0.9733333 0.9756098 0.9393939 1
## [1] 3
##      1      2 3      4      5      6      7
## [1,] 0.6000000 0.1428571 0 0.3333333 0.4666667 0.4285714 0.3333333
## [2,] 0.6842105 0.1428571 0 0.1891892 0.6800000 0.2972973 0.1250000
##      9 10      11      12      13      14      15
## [1,] 0.5555556 0.4 0.4000000 0.4000000 0.4000000 0.4285714 0.6000000
## [2,] 0.7333333 0.4 0.4074074 0.2352941 0.3142857 0.3181818 0.4693878
##      16      17      18      19      20      21      22
## [1,] 0.6190476 0.6923077 0.7037037 0.7777778 0.7692308 0.7777778 0.8461538
## [2,] 0.5714286 0.6333333 0.7241379 0.7096774 0.8888889 0.9230769 0.9545455
##      23 24      25      26      27      28      29      30
## [1,] 1 1 0.8333333 0.8400000 0.8461538 0.8461538 0.7333333 0.9200000
## [2,] 1 1 0.9259259 0.9322034 0.9493671 0.9534884 0.9223301 0.9809524
## [1] 4
##      1      2      3 4      5      6      7
## [1,] 0.7777778 0.4545455 0.3333333 0 0.1578947 0.1111111 0.3846154
## [2,] 0.7500000 0.3333333 0.1891892 0 0.4909091 0.1904762 0.2432432
##      9      10      11      12      13      14      15
## [1,] 0.3846154 0.2857143 0.4285714 0.4285714 0.5714286 0.3333333 0.3684211
## [2,] 0.6571429 0.3714286 0.4375000 0.3333333 0.4500000 0.3469388 0.4074074
##      16      17      18      19      20      21      22
## [1,] 0.3600000 0.4666667 0.4838710 0.5483871 0.5333333 0.5483871 0.6000000
## [2,] 0.4754098 0.5076923 0.5873016 0.6119403 0.7402597 0.7831325 0.827957
##      23      24      25      26      27      28      29
## [1,] 0.8181818 0.7500000 0.6250 0.5862069 0.6000000 0.6000000 0.5294118
## [2,] 0.9200000 0.8888889 0.8125 0.7812500 0.8333333 0.8241758 0.8148148
##      30
## [1,] 0.6551724
## [2,] 0.8727273
## [1] 5
##      1      2      3      4 5      6      7      9
## [1,] 0.8333333 0.5714286 0.4666667 0.1578947 0 0.04761905 0.375 0.3750000

```

```

## [2,] 0.8918919 0.6956522 0.6800000 0.4909091 0 0.41818182 0.640 0.5833333
##      10      11      12      13      14      15      16
## [1,] 0.2941176 0.5294118 0.5294118 0.6470588 0.4285714 0.3636364 0.2857143
## [2,] 0.5416667 0.6888889 0.6923077 0.7358491 0.6774194 0.5522388 0.4594595
##      17      18      19      20      21      22      23
## [1,] 0.3939394 0.4117647 0.4117647 0.3939394 0.4117647 0.4545455 0.7142857
## [2,] 0.5128205 0.5000000 0.5000000 0.4888889 0.5000000 0.5283019 0.8947368
##      24      25      26      27      28      29      30
## [1,] 0.6842105 0.3684211 0.4375000 0.4545455 0.4545455 0.4054054 0.5000000
## [2,] 0.7959184 0.6888889 0.5584416 0.5670103 0.5769231 0.5371901 0.5934959
## [1] 6
##      1      2      3      4      5 6      7
## [1,] 0.8181818 0.5384615 0.4285714 0.1111111 0.04761905 0 0.3333333
## [2,] 0.7500000 0.3939394 0.2972973 0.1904762 0.41818182 0 0.2432432
##      9      10      11      12      13      14      15
## [1,] 0.3333333 0.2500000 0.5000 0.5000000 0.625 0.4000000 0.3333333
## [2,] 0.5428571 0.2571429 0.4375 0.3846154 0.550 0.4285714 0.3703704
##      16      17      18      19      20      21      22
## [1,] 0.2592593 0.3750000 0.3939394 0.4545455 0.4375000 0.4545455 0.5000000
## [2,] 0.3770492 0.4461538 0.5238095 0.5223881 0.6883117 0.7349398 0.7849462
##      23      24      25      26      27      28      29
## [1,] 0.6923077 0.6666667 0.4444444 0.483871 0.5000000 0.5000000 0.4444444
## [2,] 0.8400000 0.7777778 0.6875000 0.687500 0.7619048 0.7802198 0.7777778
##      30
## [1,] 0.5483871
## [2,] 0.8363636
## [1] 7
##      1      2      3      4      5      6 7      9
## [1,] 0.6666667 0.2500000 0.3333333 0.3846154 0.375 0.3333333 0 0.4000000
## [2,] 0.6842105 0.1428571 0.1250000 0.2432432 0.640 0.2432432 0 0.6666667
##      10      11      12      13      14      15
## [1,] 0.09090909 0.2727273 0.2727273 0.4545455 0.4666667 0.3750000
## [2,] 0.2666667 0.3333333 0.1764706 0.3714286 0.3636364 0.3877551
##      16      17      18      19      20      21      22
## [1,] 0.5454545 0.6296296 0.6428571 0.7142857 0.7037037 0.7142857 0.7777778
## [2,] 0.5357143 0.6000000 0.6896552 0.6774194 0.8611111 0.8974359 0.9318182
##      23      24      25      26      27      28      29
## [1,] 0.75 0.8461538 0.6923077 0.7692308 0.7777778 0.7777778 0.6774194
## [2,] 0.90 0.9354839 0.8518519 0.8983051 0.9240506 0.9302326 0.9029126
##      30
## [1,] 0.8461538
## [2,] 0.9619048
## [1] 8
##      1      2      3      4      5      6      7 9
## [1,] 1 0.5000000 0.5555556 0.3846154 0.3750000 0.3333333 0.4000000 0
## [2,] 1 0.6923077 0.7333333 0.6571429 0.5833333 0.5428571 0.6666667 0
##      10      11      12      13      14      15      16
## [1,] 0.4545455 0.4545455 0.4545455 0.6363636 0.6000000 0.5000000 0.5454545
## [2,] 0.5714286 0.7600000 0.6875000 0.8181818 0.7619048 0.6595745 0.7037037
##      17      18      19      20      21      22      23
## [1,] 0.6296296 0.6428571 0.6428571 0.6296296 0.6428571 0.7037037 0.5000000
## [2,] 0.6896552 0.6428571 0.6666667 0.6857143 0.7631579 0.7674419 0.7777778
##      24      25      26      27      28      29      30
## [1,] 0.6923077 0.6923077 0.6923077 0.7037037 0.7037037 0.6774194 0.7692308

```



```

## [2,] 0.7241379 0.8400000 0.7192982 0.7662338 0.7619048 0.7821782 0.8446602
## [1] 9
##          1          2          3          4          5          6          7
## [1,] 0.7142857 0.3333333 0.4 0.2857143 0.2941176 0.2500000 0.09090909
## [2,] 0.8823529 0.3846154 0.4 0.3714286 0.5416667 0.2571429 0.26666667
##          9 10          11          12          13          14          15
## [1,] 0.4545455 0 0.3333333 0.3333333 0.5000000 0.3750000 0.2941176
## [2,] 0.5714286 0 0.4400000 0.3750000 0.5757576 0.4761905 0.4042553
##          16          17          18          19          20          21          22
## [1,] 0.4782609 0.5714286 0.5862069 0.6551724 0.6428571 0.6551724 0.7142857
## [2,] 0.5185185 0.5172414 0.5714286 0.6333333 0.7714286 0.8157895 0.8604651
##          23          24          25          26          27          28          29
## [1,] 0.7777778 0.7142857 0.5714286 0.7037037 0.7142857 0.7142857 0.6250000
## [2,] 0.8888889 0.7931034 0.7600000 0.8245614 0.8441558 0.8571429 0.8415842
##          30
## [1,] 0.7777778
## [2,] 0.9029126
## [1] 10
##          1          2          3          4          5          6          7
## [1,] 0.7142857 0.3333333 0.4000000 0.4285714 0.5294118 0.5000 0.2727273
## [2,] 0.5714286 0.3043478 0.4074074 0.4375000 0.6888889 0.4375 0.3333333
##          9          10 11          12          13          14          15
## [1,] 0.4545455 0.3333333 0 0.0000000 0.1666667 0.2500000 0.2941176
## [2,] 0.7600000 0.4400000 0 0.2413793 0.3333333 0.4358974 0.5000000
##          16          17          18          19          20          21          22
## [1,] 0.5652174 0.5714286 0.5862069 0.7931034 0.7857143 0.7931034 0.8571429
## [2,] 0.6470588 0.6363636 0.6981132 0.8245614 0.9104478 0.9178082 0.9518072
##          23          24          25          26          27          28          29
## [1,] 0.7777778 0.8571429 0.8571429 0.8518519 0.8571429 0.8571429 0.6875000
## [2,] 0.8666667 0.9230769 0.9090909 0.9259259 0.9459459 0.9506173 0.8979592
##          30
## [1,] 0.9259259
## [2,] 0.9800000
## [1] 11
##          1          2          3          4          5          6          7
## [1,] 0.7142857 0.3333333 0.4000000 0.4285714 0.5294118 0.5000000 0.2727273
## [2,] 0.7142857 0.2000000 0.2352941 0.3333333 0.6923077 0.3846154 0.1764706
##          9          10          11 12          13          14          15
## [1,] 0.4545455 0.3333333 0.0000000 0 0.1666667 0.2500000 0.2941176
## [2,] 0.6875000 0.3750000 0.2413793 0 0.1891892 0.2173913 0.3333333
##          16          17          18          19          20          21          22
## [1,] 0.5652174 0.5714286 0.5862069 0.7931034 0.7857143 0.7931034 0.8571429
## [2,] 0.5517241 0.5806452 0.6666667 0.7500000 0.8918919 0.9250000 0.9555556
##          23          24          25          26          27          28          29
## [1,] 0.7777778 0.8571429 0.8571429 0.8518519 0.8571429 0.8571429 0.6875000
## [2,] 0.9090909 0.9393939 0.9310345 0.9344262 0.9506173 0.9545455 0.9047619
##          30
## [1,] 0.9259259
## [2,] 0.9813084
## [1] 12
##          1          2          3          4          5          6          7
## [1,] 0.7142857 0.3333333 0.4000000 0.5714286 0.6470588 0.625 0.4545455
## [2,] 0.7272727 0.2903226 0.3142857 0.4500000 0.7358491 0.550 0.3714286
##          9          10          11          12 13          14          15

```

```

## [1,] 0.6363636 0.5000000 0.1666667 0.1666667 0 0.2500000 0.2941176
## [2,] 0.8181818 0.5757576 0.3333333 0.1891892 0 0.1914894 0.3846154
##      16      17      18      19      20      21      22
## [1,] 0.5652174 0.5714286 0.5862069 0.7931034 0.8571429 0.8620690 0.9285714
## [2,] 0.5932203 0.6190476 0.7049180 0.8153846 0.9200000 0.9506173 0.9780220
##      23 24 25      26      27      28      29 30
## [1,] 1 1 1 0.9259259 0.9285714 0.9285714 0.6875000 1
## [2,] 1 1 1 0.9677419 0.9756098 0.9775281 0.9056604 1
## [1] 13
##      1      2      3      4      5      6      7
## [1,] 0.8181818 0.5384615 0.4285714 0.3333333 0.4285714 0.4000000 0.4666667
## [2,] 0.8064516 0.4000000 0.3181818 0.3469388 0.6774194 0.4285714 0.3636364
##      9      10      11      12      13 14      15
## [1,] 0.6000000 0.3750000 0.2500000 0.2500000 0.2500000 0 0.1428571
## [2,] 0.7619048 0.4761905 0.4358974 0.2173913 0.1914894 0 0.2459016
##      16      17      18      19      20      21      22
## [1,] 0.3333333 0.375 0.3939394 0.5757576 0.6250000 0.6363636 0.6875
## [2,] 0.4411765 0.500 0.6000000 0.6756757 0.8333333 0.8666667 0.9000
##      23      24      25      26      27      28      29
## [1,] 0.8461538 0.7777778 0.6666667 0.6774194 0.6875000 0.6875000 0.5000000
## [2,] 0.9375000 0.9069767 0.8461538 0.8591549 0.8901099 0.8979592 0.8434783
##      30
## [1,] 0.7419355
## [2,] 0.9316239
## [1] 14
##      1      2      3      4      5      6      7
## [1,] 0.8333333 0.5714286 0.6000000 0.3684211 0.3636364 0.3333333 0.3750000
## [2,] 0.8333333 0.5111111 0.4693878 0.4074074 0.5522388 0.3703704 0.3877551
##      9      10      11      12      13      14 15
## [1,] 0.5000000 0.2941176 0.2941176 0.2941176 0.2941176 0.1428571 0
## [2,] 0.6595745 0.4042553 0.5000000 0.3333333 0.3846154 0.2459016 0
##      16      17      18      19      20      21      22
## [1,] 0.2857143 0.3333333 0.3529412 0.5294118 0.5757576 0.5882353 0.6363636
## [2,] 0.2602740 0.4025974 0.4666667 0.5696203 0.7078652 0.7684211 0.7714286
##      23      24      25      26      27      28      29
## [1,] 0.8571429 0.7894737 0.6842105 0.6250000 0.6363636 0.6363636 0.4594595
## [2,] 0.9459459 0.8750000 0.8181818 0.7631579 0.7708333 0.7864078 0.7333333
##      30
## [1,] 0.6875000
## [2,] 0.8196721
## [1] 15
##      1      2      3      4      5      6      7
## [1,] 0.8888889 0.7000000 0.6190476 0.3600000 0.2857143 0.2592593 0.5454545
## [2,] 0.8604651 0.6538462 0.5714286 0.4754098 0.4594595 0.3770492 0.5357143
##      9      10      11      12      13      14      15
## [1,] 0.5454545 0.4782609 0.5652174 0.5652174 0.5652174 0.3333333 0.2857143
## [2,] 0.7037037 0.5185185 0.6470588 0.5517241 0.5932203 0.4411765 0.2602740
##      16      17      18      19      20      21      22
## [1,] 0 0.1282051 0.1500000 0.2500000 0.2820513 0.3000000 0.3333333
## [2,] 0 0.2619048 0.3414634 0.3953488 0.5833333 0.627451 0.6607143
##      23      24      25      26      27      28      29
## [1,] 0.8000000 0.6800000 0.6000000 0.3684211 0.3333333 0.3333333 0.2558140
## [2,] 0.9090909 0.8181818 0.7647059 0.6385542 0.6699029 0.6909091 0.6535433
##      30

```

```

## [1,] 0.3684211
## [2,] 0.7209302
## [1] 16
##           1           2           3           4           5           6           7
## [1,] 0.9130435 0.7600000 0.6923077 0.4666667 0.3939394 0.3750000 0.6296296
## [2,] 0.9148936 0.6785714 0.6333333 0.5076923 0.5128205 0.4461538 0.6000000
##           9           10          11          12          13          14          15
## [1,] 0.6296296 0.5714286 0.5714286 0.5714286 0.5714286 0.375 0.3333333
## [2,] 0.6896552 0.5172414 0.6363636 0.5806452 0.6190476 0.500 0.4025974
##          16 17          18          19          20          21          22
## [1,] 0.1282051 0 0.02222222 0.1555556 0.1818182 0.200000 0.2272727
## [2,] 0.2619048 0 0.13953488 0.3111111 0.4200000 0.490566 0.5517241
##          23          24          25          26          27          28          29
## [1,] 0.7600000 0.6000000 0.6000000 0.2558140 0.2272727 0.2272727 0.1250000
## [2,] 0.8333333 0.6949153 0.7454545 0.5402299 0.5700935 0.5789474 0.5114504
##          30
## [1,] 0.2558140
## [2,] 0.5789474
## [1] 17
##           1           2           3           4           5           6           7
## [1,] 0.9166667 0.7692308 0.7037037 0.4838710 0.4117647 0.3939394 0.6428571
## [2,] 0.9555556 0.7407407 0.7241379 0.5873016 0.5000000 0.5238095 0.6896552
##           9           10          11          12          13          14          15
## [1,] 0.6428571 0.5862069 0.5862069 0.5862069 0.5862069 0.3939394 0.3529412
## [2,] 0.6428571 0.5714286 0.6981132 0.6666667 0.7049180 0.6000000 0.4666667
##          16          17 18          19          20          21          22
## [1,] 0.1500000 0.02222222 0 0.1304348 0.1555556 0.1739130 0.2000000
## [2,] 0.3414634 0.13953488 0 0.2500000 0.3265306 0.4038462 0.4736842
##          23          24          25          26          27 28          29
## [1,] 0.7692308 0.5483871 0.5483871 0.2272727 0.2000000 0.2 0.1020408
## [2,] 0.8260870 0.6491228 0.6603774 0.4588235 0.4857143 0.5 0.4418605
##          30
## [1,] 0.2272727
## [2,] 0.5267176
## [1] 18
##           1           2           3           4           5           6           7
## [1,] 1 0.8461538 0.7777778 0.5483871 0.4117647 0.4545455 0.7142857
## [2,] 1 0.7931034 0.7096774 0.6119403 0.5000000 0.5223881 0.6774194
##           9           10          11          12          13          14          15
## [1,] 0.6428571 0.6551724 0.7931034 0.7931034 0.7931034 0.5757576 0.5294118
## [2,] 0.6666667 0.6333333 0.8245614 0.7500000 0.8153846 0.6756757 0.5696203
##          16          17          18 19          20          21          22
## [1,] 0.2500000 0.1555556 0.1304348 0 0.02222222 0.04347826 0.06666667
## [2,] 0.3953488 0.3111111 0.2500000 0 0.23529412 0.29629630 0.38983051
##          23          24          25          26          27          28
## [1,] 0.7692308 0.4838710 0.4838710 0.09090909 0.06666667 0.06666667
## [2,] 0.8400000 0.6393443 0.6140351 0.32584270 0.37614679 0.41379310
##          29          30
## [1,] 0.06122449 0.09090909
## [2,] 0.41353383 0.48148148
## [1] 19
##           1           2           3           4           5           6           7
## [1,] 1 0.8400000 0.7692308 0.5333333 0.3939394 0.4375000 0.7037037
## [2,] 1 0.9117647 0.8888889 0.7402597 0.4888889 0.6883117 0.8611111

```

```

##          9          10          11          12          13          14          15
## [1,] 0.6296296 0.6428571 0.7857143 0.7857143 0.8571429 0.6250000 0.5757576
## [2,] 0.6857143 0.7714286 0.9104478 0.8918919 0.9200000 0.8333333 0.7078652
##          16          17          18          19 20          21          22
## [1,] 0.2820513 0.1818182 0.1555556 0.02222222 0 0.02222222 0.04545455
## [2,] 0.5833333 0.4200000 0.3265306 0.23529412 0 0.10169492 0.18750000
##          23          24          25          26          27          28
## [1,] 0.7600000 0.4666667 0.4666667 0.06976744 0.04545455 0.04545455
## [2,] 0.8666667 0.5774648 0.6716418 0.21212121 0.19327731 0.22222222
##          29          30
## [1,] 0.08333333 0.06976744
## [2,] 0.24475524 0.29655172
## [1] 20
##          1          2          3          4          5          6          7
## [1,] 1 0.8461538 0.7777778 0.5483871 0.4117647 0.4545455 0.7142857
## [2,] 1 0.9459459 0.9230769 0.7831325 0.5000000 0.7349398 0.8974359
##          9          10          11          12          13          14          15
## [1,] 0.6428571 0.6551724 0.7931034 0.7931034 0.8620690 0.6363636 0.5882353
## [2,] 0.7631579 0.8157895 0.9178082 0.9250000 0.9506173 0.8666667 0.7684211
##          16          17          18          19          20 21          22
## [1,] 0.300000 0.200000 0.1739130 0.04347826 0.02222222 0 0.02222222
## [2,] 0.627451 0.490566 0.4038462 0.29629630 0.10169492 0 0.10447761
##          23          24          25          26          27          28
## [1,] 0.7692308 0.4838710 0.4838710 0.04545455 0.02222222 0.02222222
## [2,] 0.8787879 0.6103896 0.6986301 0.20000000 0.13600000 0.16666667
##          29          30
## [1,] 0.06122449 0.04545455
## [2,] 0.18120805 0.23178808
## [1] 21
##          1          2          3          4          5          6          7
## [1,] 1 0.9200000 0.8461538 0.600000 0.4545455 0.5000000 0.7777778
## [2,] 1 0.9761905 0.9545455 0.827957 0.5283019 0.7849462 0.9318182
##          9          10          11          12          13          14          15
## [1,] 0.7037037 0.7142857 0.8571429 0.8571429 0.9285714 0.6875 0.6363636
## [2,] 0.7674419 0.8604651 0.9518072 0.9555556 0.9780220 0.9000 0.7714286
##          16          17          18          19          20          21 22
## [1,] 0.3333333 0.2272727 0.2000000 0.06666667 0.04545455 0.02222222 0
## [2,] 0.6607143 0.5517241 0.4736842 0.38983051 0.18750000 0.10447761 0
##          23          24          25          26          27          28
## [1,] 0.7600000 0.4666667 0.4666667 0.02325581 0.0000000 0.0000000
## [2,] 0.8947368 0.6551724 0.7349398 0.25217391 0.1259259 0.1267606
##          29          30
## [1,] 0.08333333 0.02325581
## [2,] 0.11949686 0.18012422
## [1] 22
##          1 2 3          4          5          6          7          9          10
## [1,] 1 1 1 0.8181818 0.7142857 0.6923077 0.75 0.5000000 0.7777778
## [2,] 1 1 1 0.9200000 0.8947368 0.8400000 0.90 0.7777778 0.8888889
##          11          12 13          14          15          16          17
## [1,] 0.7777778 0.7777778 1 0.8461538 0.8571429 0.8000000 0.7600000
## [2,] 0.8666667 0.9090909 1 0.9375000 0.9459459 0.9090909 0.8333333
##          18          19          20          21          22 23          24
## [1,] 0.7692308 0.7692308 0.7600000 0.7692308 0.7600000 0 0.4545455
## [2,] 0.8260870 0.8400000 0.8666667 0.8787879 0.8947368 0 0.5789474

```

```

##          25          26          27          28          29          30
## [1,] 0.4545455 0.7500000 0.7600000 0.7600000 0.7931034 0.7500000
## [2,] 0.4666667 0.8297872 0.880597 0.8918919 0.9120879 0.9139785
## [1] 23
##      1 2 3          4          5          6          7          9          10
## [1,] 1 1 1 0.7500000 0.6842105 0.6666667 0.8461538 0.6923077 0.7142857
## [2,] 1 1 1 0.8888889 0.7959184 0.7777778 0.9354839 0.7241379 0.7931034
##          11          12 13          14          15          16          17
## [1,] 0.8571429 0.8571429 1 0.7777778 0.7894737 0.6800000 0.6000000
## [2,] 0.9230769 0.9393939 1 0.9069767 0.8750000 0.8181818 0.6949153
##          18          19          20          21          22          23 24
## [1,] 0.5483871 0.4838710 0.4666667 0.4838710 0.4666667 0.4545455 0
## [2,] 0.6491228 0.6393443 0.5774648 0.6103896 0.6551724 0.5789474 0
##          25          26          27          28          29          30
## [1,] 0.3750000 0.4482759 0.4666667 0.4666667 0.5294118 0.4482759
## [2,] 0.4615385 0.4827586 0.6153846 0.6470588 0.7058824 0.7115385
## [1] 24
##      1 2          3          4          5          6          7          9
## [1,] 1 1 0.8333333 0.6250 0.3684211 0.4444444 0.6923077 0.6923077
## [2,] 1 1 0.9259259 0.8125 0.6888889 0.6875000 0.8518519 0.8400000
##          10          11          12 13          14          15          16
## [1,] 0.5714286 0.8571429 0.8571429 1 0.6666667 0.6842105 0.6000000
## [2,] 0.7600000 0.9090909 0.9310345 1 0.8461538 0.8181818 0.7647059
##          17          18          19          20          21          22          23
## [1,] 0.6000000 0.5483871 0.4838710 0.4666667 0.4838710 0.4666667 0.4545455
## [2,] 0.7454545 0.6603774 0.6140351 0.6716418 0.6986301 0.7349398 0.4666667
##          24 25          26          27          28          29          30
## [1,] 0.3750000 0 0.4482759 0.4666667 0.4666667 0.5294118 0.4482759
## [2,] 0.4615385 0 0.5925926 0.7027027 0.7283951 0.7755102 0.7800000
## [1] 25
##      1          2          3          4          5          6          7
## [1,] 1 0.9166667 0.8400000 0.5862069 0.4375000 0.483871 0.7692308
## [2,] 1 0.9636364 0.9322034 0.7812500 0.5584416 0.687500 0.8983051
##          9          10          11          12          13          14          15
## [1,] 0.6923077 0.7037037 0.8518519 0.8518519 0.9259259 0.6774194 0.6250000
## [2,] 0.7192982 0.8245614 0.9259259 0.9344262 0.9677419 0.8591549 0.7631579
##          16          17          18          19          20          21
## [1,] 0.3684211 0.2558140 0.2272727 0.09090909 0.06976744 0.04545455
## [2,] 0.6385542 0.5402299 0.4588235 0.32584270 0.21212121 0.20000000
##          22          23          24          25 26          27          28
## [1,] 0.02325581 0.7500000 0.4482759 0.4482759 0 0.02325581 0.02325581
## [2,] 0.25217391 0.8297872 0.4827586 0.5925926 0 0.18867925 0.23893805
##          29          30
## [1,] 0.1063830 0.04761905
## [2,] 0.3384615 0.36363636
## [1] 26
##      1          2          3          4          5          6          7
## [1,] 1 0.9200000 0.8461538 0.6000000 0.4545455 0.5000000 0.7777778
## [2,] 1 0.9733333 0.9493671 0.8333333 0.5670103 0.7619048 0.9240506
##          9          10          11          12          13          14          15
## [1,] 0.7037037 0.7142857 0.8571429 0.8571429 0.9285714 0.6875000 0.6363636
## [2,] 0.7662338 0.8441558 0.9459459 0.9506173 0.9756098 0.8901099 0.7708333
##          16          17          18          19          20          21
## [1,] 0.3333333 0.2272727 0.2000000 0.06666667 0.04545455 0.02222222

```

```

## [2,] 0.6699029 0.5700935 0.4857143 0.37614679 0.19327731 0.13600000
##          22          23          24          25          26 27          28
## [1,] 0.0000000 0.7600000 0.4666667 0.4666667 0.02325581 0 0.00000000
## [2,] 0.1259259 0.880597 0.6153846 0.7027027 0.18867925 0 0.09774436
##          29          30
## [1,] 0.08333333 0.02325581
## [2,] 0.18666667 0.19736842
## [1] 27
##          1          2          3          4          5          6          7
## [1,] 1 0.9200000 0.8461538 0.6000000 0.4545455 0.5000000 0.7777778
## [2,] 1 0.9756098 0.9534884 0.8241758 0.5769231 0.7802198 0.9302326
##          9          10          11          12          13          14          15
## [1,] 0.7037037 0.7142857 0.8571429 0.8571429 0.9285714 0.6875000 0.6363636
## [2,] 0.7619048 0.8571429 0.9506173 0.9545455 0.9775281 0.8979592 0.7864078
##          16          17 18          19          20          21          22
## [1,] 0.3333333 0.2272727 0.2 0.06666667 0.04545455 0.02222222 0.0000000
## [2,] 0.6909091 0.5789474 0.5 0.41379310 0.22222222 0.16666667 0.1267606
##          23          24          25          26          27 28          29
## [1,] 0.7600000 0.4666667 0.4666667 0.02325581 0.00000000 0 0.08333333
## [2,] 0.8918919 0.6470588 0.7283951 0.23893805 0.09774436 0 0.14649682
##          30
## [1,] 0.02325581
## [2,] 0.15723270
## [1] 28
##          1          2          3          4          5          6          7
## [1,] 0.9259259 0.7931034 0.7333333 0.5294118 0.4054054 0.4444444 0.6774194
## [2,] 0.9777778 0.9393939 0.9223301 0.8148148 0.5371901 0.7777778 0.9029126
##          9          10          11          12          13          14          15
## [1,] 0.6774194 0.6250000 0.6875000 0.6875000 0.6875000 0.5000000 0.4594595
## [2,] 0.7821782 0.8415842 0.8979592 0.9047619 0.9056604 0.8434783 0.7333333
##          16          17          18          19          20          21
## [1,] 0.2558140 0.1250000 0.1020408 0.06122449 0.08333333 0.06122449
## [2,] 0.6535433 0.5114504 0.4418605 0.41353383 0.24475524 0.18120805
##          22          23          24          25          26          27
## [1,] 0.08333333 0.7931034 0.5294118 0.5294118 0.1063830 0.08333333
## [2,] 0.11949686 0.9120879 0.7058824 0.7755102 0.3384615 0.18666667
##          28 29          30
## [1,] 0.08333333 0 0.1063830
## [2,] 0.14649682 0 0.1477273
## [1] 29
##          1 2          3          4          5          6          7          9
## [1,] 1 1 0.9200000 0.6551724 0.5000000 0.5483871 0.8461538 0.7692308
## [2,] 1 1 0.9809524 0.8727273 0.5934959 0.8363636 0.9619048 0.8446602
##          10          11          12 13          14          15          16
## [1,] 0.7777778 0.9259259 0.9259259 1 0.7419355 0.6875000 0.3684211
## [2,] 0.9029126 0.9800000 0.9813084 1 0.9316239 0.8196721 0.7209302
##          17          18          19          20          21          22
## [1,] 0.2558140 0.2272727 0.09090909 0.06976744 0.04545455 0.02325581
## [2,] 0.5789474 0.5267176 0.48148148 0.29655172 0.23178808 0.18012422
##          23          24          25          26          27          28
## [1,] 0.7500000 0.4482759 0.4482759 0.04761905 0.02325581 0.02325581
## [2,] 0.9139785 0.7115385 0.7800000 0.36363636 0.19736842 0.15723270
##          29 30
## [1,] 0.1063830 0

```

```
## [2,] 0.1477273 0
```

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

- Does the resemblance matrix (`fish.db`) represent similarity or dissimilarity? What information in the resemblance matrix led you to arrive at your answer?
- Compare the resemblance matrices (`fish.db` or `fish.ds`) you just created. How does the choice of the $\tilde{S}_{\text{A,rensen}}$ or Bray-Curtis distance influence your interpretation of site (dis)similarity?

Answer 5a: It is a DISSIMILARITY resemblance matrix. When I also include the diagonal values, they are all 0. If the diagonals were all 1, it would be a similarity matrix. **Answer 5b:** I looped over all the values and printed them next to each other to assess their similarity (the similarity of their dissimilarity...). The two resemblance matrices agree relatively well, though there are some exceptions where the estimates for one cell in the matrix are very different. In several comparisons, the $\tilde{S}_{\text{A,rensen}}$ value is half that of the Bray-Curtis value. Given this finding, if abundance data is available, the abundance-based metrics should always be used. The $\tilde{S}_{\text{A,rensen}}$ metric appears to be consistently lower than the Bray-Curtis metric. When using incidence data to assess resemblance, it may be wise to expect dissimilarity to be underestimated (or similarity to be overestimated).

4) VISUALIZING BETA-DIVERSITY

A. Heatmaps

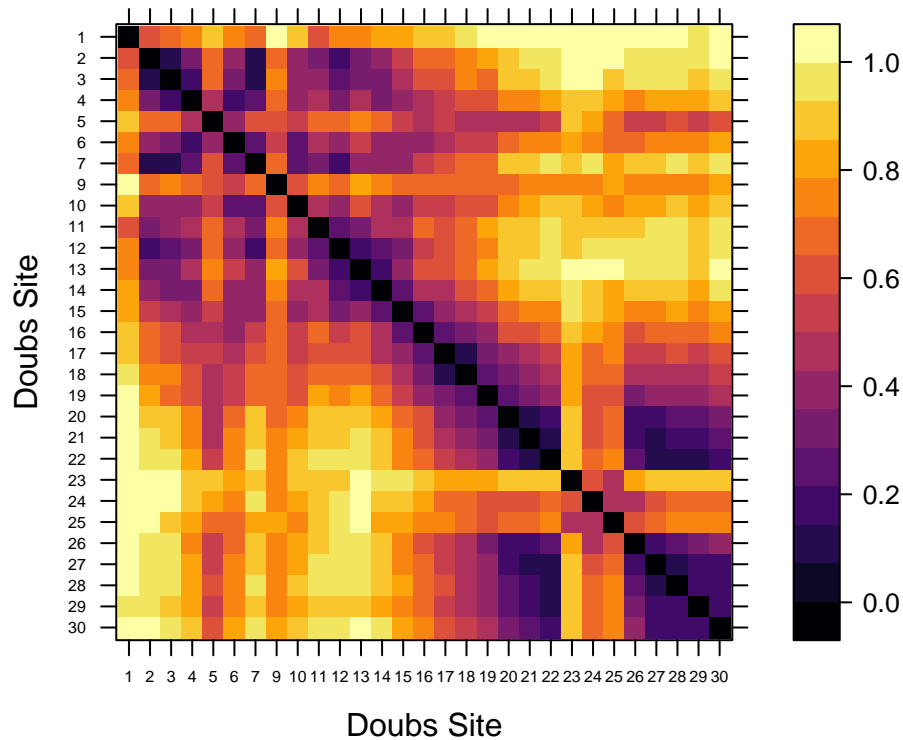
In the R code chunk below, do the following:

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

```
fish.db <- vegdist(fish, method = "bray", diag = T)
order <- rev(attr(fish.db, "Labels"))

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

Bray-Curtis Distance



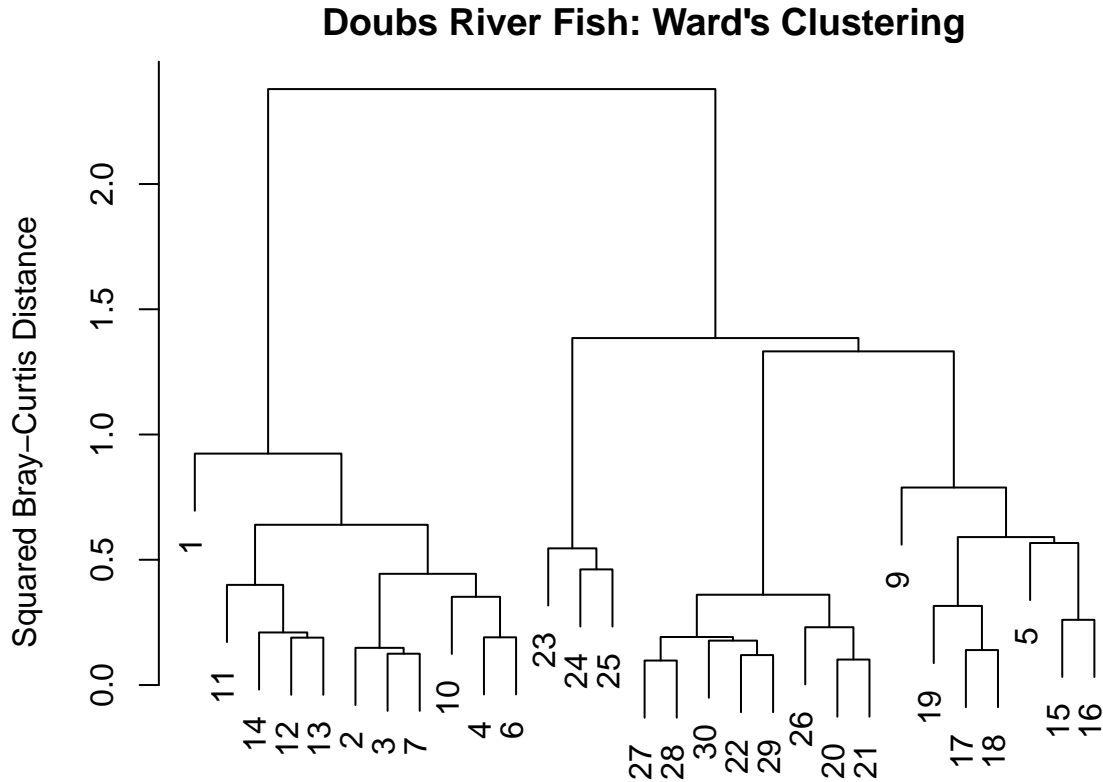
B. Cluster Analysis

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`).

```
fish.ward <- hclust(fish.db, method="ward.D2")

par(mar = c(1, 5, 2, 2) + 0.1)
plot(fish.ward, 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: If the site numbers provided represent the order of sites (e.g. site 22 is next to site 23), and I think they do, then a possible ecological hypothesis is that specific parts of the river harbor specific fish and that migration up and down the river may be relatively rare. I make this hypothesis based on the fact that clustering tended to group adjacent sites. The first big cluster on the left contained sites 1-14 (give or take a few sites), the next cluster contained sites 23-25, and so on. Looking at the Fish richness introductory plot, it is clear that sites in the same region share relatively the same richness values. We could break this analysis down even further to look at what species are overrepresented in what parts of the river.

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.

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

```
#Variance explained by first axis
explainvar1
```

```
## [1] 54.6
```

```
#variance explained by second axis
explainvar2
```

```
## [1] 16.2
```

```
#variance explained by third axis
explainvar3
```

```
## [1] 10.5
```

```
sum.eig
```

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

```
par(mar = c(5, 5, 1, 2) + 0.1)

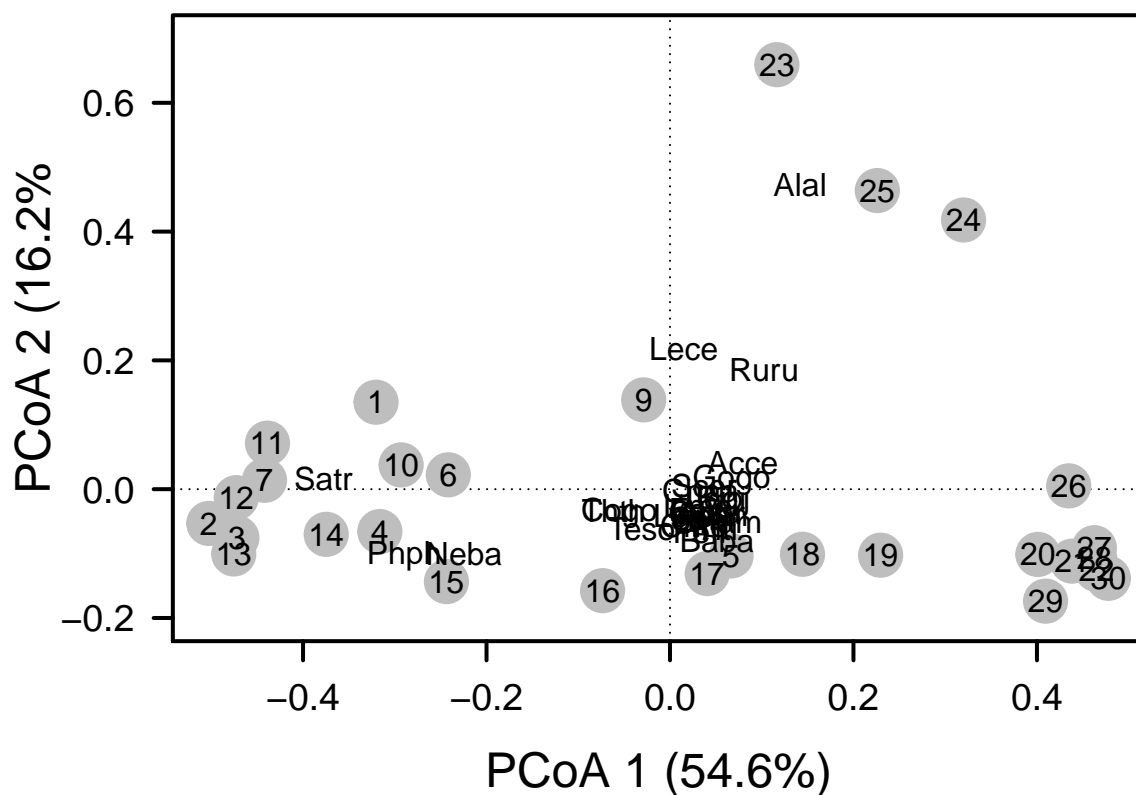
plot(fish.pcoa$points[,1], fish.pcoa$points[,2], ylim = c(-0.2, 0.7),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = F)

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(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))

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

fish.pcoa <- add.spec.scores(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.

```
spe.corr <- add.spec.scores(fish.pcoa, fishREL, method = "cor.scores")$cproj
corrcut <- 0.7
imp.spp <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]

fit <- envfit(fish.pcoa, fishREL, perm = 999)
print(fit)
```

```
##
## ***VECTORS
##
##          Dim1      Dim2      r2 Pr(>r)
## Cogo -0.83884 -0.54438 0.2982 0.010 **
## Satr -0.99904  0.04371 0.4326 0.003 **
## 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 0.022 *
## Teso -0.44704 -0.89452 0.1700 0.097 .
## Chna  0.99707 -0.07644 0.4612 0.001 ***
## Chto  0.42032 -0.90738 0.2579 0.022 *
## Lele  0.33041 -0.94384 0.0495 0.515
```

```

## Lece 0.06856 0.99765 0.3399 0.009 **
## 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 0.002 **
## Eslu 0.72044 -0.69352 0.1673 0.090 .
## 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 0.004 **
## 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 0.002 **
## 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:

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

Answer 7a: It appears that there are roughly three site groupings. Those towards the bottom left of the plot (driven by `Satr`, `Phph` and `Neba`), those toward the bottom right (apparently driven by the large assemblage of species seen to the right of the origin), and the few sites towards the top of the plot (driven by `Lece`, `Ruru`, and `Alal`). Relatively few species appear to strongly contribute to variation in fish community composition throughout the river (`Satr`, `Phph`, `Neba`, `Leve`, `Ruru`, & `Alal`) but the large grouping of species coordinates to the right of the origin may indicate that smaller differences in the composition of several species are driving the variation in diversity of the sites seen in the bottom right of the graph. **Answer 7b:** Strong indicators of river quality (if we consider river quality to be a function of fish diversity) could be the following species: `Satr`, `Phph`, `Neba`, `Alal`, `Lece`, and `Ruru`. Indeed, these species have small, significant p-values based on the permutations tests for correlations along the PcoA axes. But, many other species are also highly (and significantly) correlated... I would think that these species are those that are clustered to the right of the origin.

5) HYPOTHESIS TESTING

A. Multivariate Procedures for Categorical Designs

Earlier work done in the Doubs River suggested that the river has four distinct regions of habitat quality: the first region (sites 1-14) of “high quality”; the second (sites 15 - 19) and fourth (sites 26 - 30) of “moderate quality”; and the third (sites 20 - 25) of “low quality”.

In the code chunk below, test the hypothesis that fish community composition varies with river quality.

- create a factor vector that categorizes habitat quality in the Doubs River,

2. use the multivariate analyses for categorical predictors to describe how fish community structure relates to habitat quality.

```
quality <- c(rep("HQ", 13), rep("MQ", 5), rep("LQ", 6), rep("MQ", 5))
adonis(fish ~ quality, method = "bray", permutations = 999)
```

```
##
## Call:
## adonis(formula = fish ~ quality, permutations = 999, method = "bray")
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## quality    2     3.0947  1.54733   10.97 0.45765  0.001 ***
## Residuals 26     3.6674  0.14105     0.54235
## Total     28     6.7621           1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#Quality is significant in the PERMANOVA (p = 0.001)

```
indval <- multipatt(fish, cluster = quality, func = "IndVal.g", control = how(nperm=999))
summary(indval)
```

```
##
## Multilevel pattern analysis
## -----
##
## Association function: IndVal.g
## Significance level (alpha): 0.05
##
## Total number of species: 27
## Selected number of species: 23
## Number of species associated to 1 group: 1
## Number of species associated to 2 groups: 22
##
## List of species associated to each combination:
##
## Group MQ #sps. 1
##      stat p.value
## Teso 0.686  0.022 *
##
## Group HQ+MQ #sps. 2
##      stat p.value
## Satr 0.860  0.004 **
## Phph 0.859  0.008 **
##
## Group LQ+MQ #sps. 20
##      stat p.value
## Alal 0.935  0.001 ***
```

```
## Gogo 0.933 0.001 ***
## Ruru 0.916 0.001 ***
## Legi 0.901 0.001 ***
## Baba 0.895 0.001 ***
## Chna 0.866 0.001 ***
## Spbi 0.866 0.001 ***
## Cyca 0.866 0.002 **
## Acce 0.866 0.001 ***
## Lele 0.863 0.002 **
## Titi 0.853 0.006 **
## Chto 0.829 0.002 **
## Rham 0.829 0.001 ***
## Anan 0.829 0.001 ***
## Eslu 0.827 0.014 *
## Pefl 0.806 0.010 **
## Blbj 0.791 0.002 **
## Scer 0.766 0.004 **
## Abbr 0.750 0.007 **
## Icme 0.661 0.023 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fish.rel <- decostand(fish, method = "total")
phi <- multipatt(fish.rel, cluster=quality, func = "r.g", control = how(nperm=999))
summary(phi)
```

```
##
## Multilevel pattern analysis
## -----
##
## Association function: r.g
## Significance level (alpha): 0.05
##
## Total number of species: 27
## Selected number of species: 18
## Number of species associated to 1 group: 9
## Number of species associated to 2 groups: 9
##
## List of species associated to each combination:
##
## Group HQ #sps. 3
##      stat p.value
## Phph 0.802 0.001 ***
## Neba 0.734 0.001 ***
## Satr 0.650 0.001 ***
##
## Group LQ #sps. 2
##      stat p.value
## Alal 0.693 0.001 ***
## Ruru 0.473 0.030 *
##
## Group MQ #sps. 4
##      stat p.value
## Anan 0.571 0.004 **
```

```
## Spbi 0.557    0.009 **
## Chto 0.542    0.009 **
## Icme 0.475    0.029 *
##
## Group LQ+MQ #sps. 9
##      stat p.value
## Legi 0.658    0.001 ***
## Baba 0.645    0.002 **
## Rham 0.600    0.004 **
## Acce 0.594    0.004 **
## Cyca 0.586    0.003 **
## Chna 0.571    0.004 **
## Blbj 0.571    0.008 **
## Gogo 0.523    0.008 **
## Abbr 0.499    0.026 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 8: Based on the PERMANOVA, IndVal, and phi coefficient analyses, what did you learn about the relationship between habitat quality and the fish species composition?

Answer 8: Based on the PERMANOVA, the factor `quality` is significantly related to species composition ($P = 0.001$). To look at how individuals species are related to sites (or groups of sites) we used indicator values and phi coefficients. Based on IndVals, only one species is significantly associated with a single group (Teso with MQ site). Two species are significantly associated with the group combination HQ+MQ (Satr and Phph) and 20 species are significantly associated with the combination LQ+MQ. Using phi coefficients of association to assess the habitat preferences of each species, we find that three species are significantly associated with the HQ group (Phph, Neba, and Satr), two species are associated with the LQ group (Alal, and Ruru), four species are associated with the MQ group (Anan, Spbi, Chto, and Icme), and nine species are associated with the combination LQ+MQ. All together, it appears that specific species of fish are in fact associated with river quality (if we agree with the a priori classification of the sites by river quality).

B. Multivariate Procedures for Continuous Designs

i. Mantel Test

In the R code chunk below, do the following:

1. create distance matrices for both fish communities and environmental factors, and
2. use a Mantel test to determine if these matrices are correlated, and test the hypothesis that fish assemblages are correlated with stream environmental variables.

```
fish.dist <- vegdist(doubs$fish[-8, ], method = "bray")
env.dist <- vegdist(scale(doubs$env[-8,]), method = "euclid")

mantel(fish.dist, env.dist)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
```

```
## Call:
## mantel(xdis = fish.dist, ydis = env.dist)
##
## Mantel statistic r: 0.604
##      Significance: 0.001
##
## Upper quantiles of permutations (null model):
##      90%      95%    97.5%     99%
## 0.0917 0.1345 0.1695 0.2005
## Permutation: free
## Number of permutations: 999
```

Question 9: What do the results from our Mantel test suggest about fish diversity and stream environmental conditions? How does this relate to your hypothesis about stream quality influencing fish communities?

Answer 9: Fish diversity and stream environmental conditions are significantly correlated at $\alpha = 0.05$ ($r = 0.604$, $P = 0.001$). This result agrees with the hypothesis that stream quality influences fish communities and agrees with the results of the species-site group association tests.

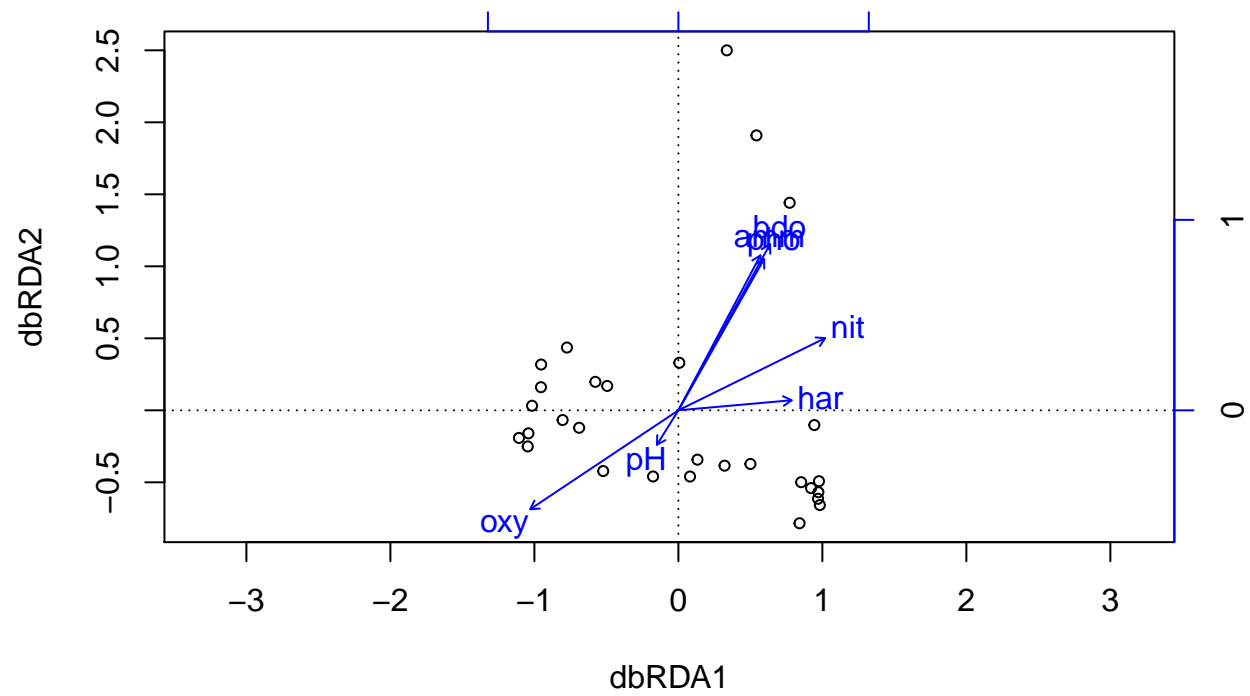
ii. Constrained Ordination

In the R code chunk below, do the following:

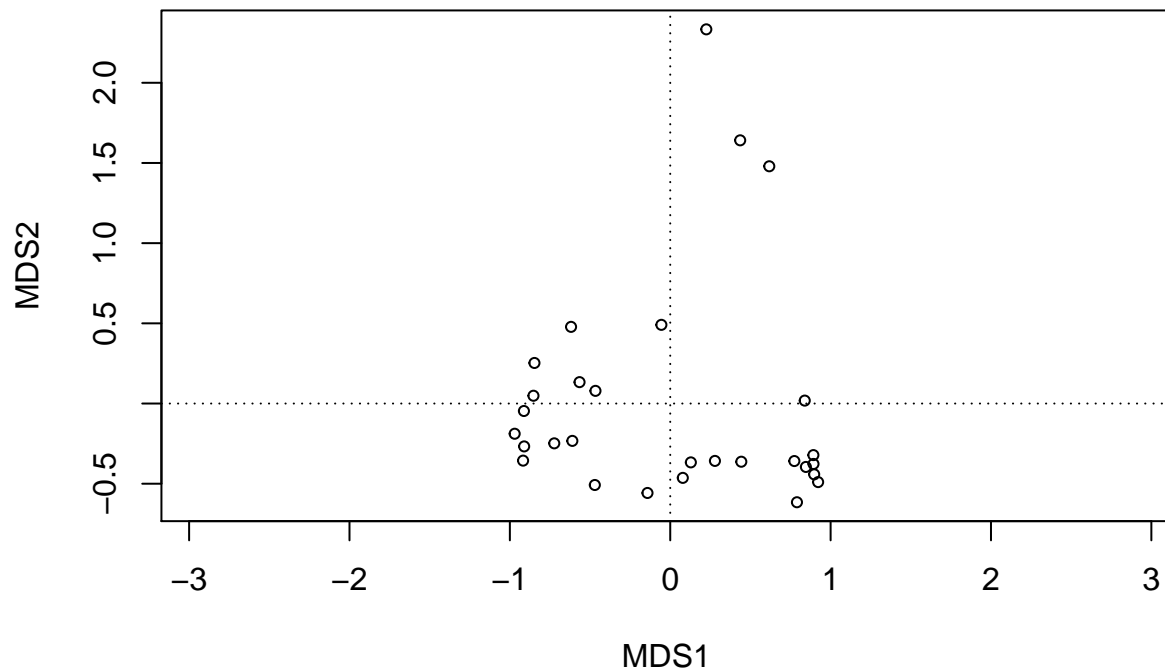
1. create an environmental matrix of the water chemistry data included in the `doubs` dataset using forward and reverse selection of variables,???
2. conduct a redundancy analysis on the fish assemblages of the Doubs River,
3. use a permutation test to determine the significance of the constrained analysis,
4. use a permutation test to determine the correlation of each environmental factor on the constrained axes,
5. calculate the explained variation on the first and second constrained axes,
6. plot the constrained ordination results including labeled points for each site, and
7. add vectors that demonstrate the influence of each environmental factor the constrained ordination.

```
env.chem <- as.matrix(doubs$env[-8, 5:11])

doubs.dbrda <- dbrda(fish.db ~ ., as.data.frame(env.chem))
ordiplot(doubs.dbrda)
```

```
doubs.dbrda.mod0 <- dbrda(fish.db ~ 1, as.data.frame(env.chem))
ordiplot(doubs.dbrda.mod0)
```



```
doubs.dbrda.mod1 <- dbrda(fish.db ~ ., as.data.frame(env.chem))
```

```
#Model selection
```

```
doubs.dbrda <- ordiR2step(doubs.dbrda.mod0, doubs.dbrda.mod1, perm.max= 200)
```

```
## Step: R2.adj= 0
```

```
## Call: fish.db ~ 1
```

```
##
```

```
## R2.adjusted
```

```
## <All variables> 0.53032584
```

```
## + oxy 0.27727176
```

```
## + nit 0.25755208
```

```
## + bdo 0.17477787
```

```
## + pho 0.14568614
```

```
## + har 0.14174915
```

```
## + amm 0.14142804
```

```
## <none> 0.00000000
```

```
## + pH -0.01827054
```

```
##
```

```
## Df AIC F Pr(>F)
```

```
## + oxy 1 47.939 11.742 0.002 **
```

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Step: R2.adj= 0.2772718
```

```

## Call: fish.db ~ oxy
##
##               R2.adjusted
## <All variables> 0.5303258
## + bdo          0.4009000
## + amm          0.3474192
## + pho          0.3452702
## + har          0.3331357
## + nit          0.3316120
## <none>         0.2772718
## + pH           0.2586983
## - oxy          0.0000000
##
##      Df      AIC      F Pr(>F)
## + bdo  1 43.404 6.5716 0.006 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.4009
## Call: fish.db ~ oxy + bdo
##
##               R2.adjusted
## <All variables> 0.5303258
## + nit          0.4980793
## + har          0.4695121
## <none>         0.4009000
## + pho          0.3938042
## + amm          0.3869134
## + pH           0.3865240
## - bdo          0.2772718
## - oxy          0.1747779
##
##      Df      AIC      F Pr(>F)
## + nit  1 39.134 6.034 0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.4980793
## Call: fish.db ~ oxy + bdo + nit
##
##               R2.adjusted
## + amm          0.5415705
## <All variables> 0.5303258
## + pho          0.5277128
## + har          0.5218852
## <none>         0.4980793
## + pH           0.4843267
## - nit          0.4009000
## - oxy          0.3420426
## - bdo          0.3316120

```

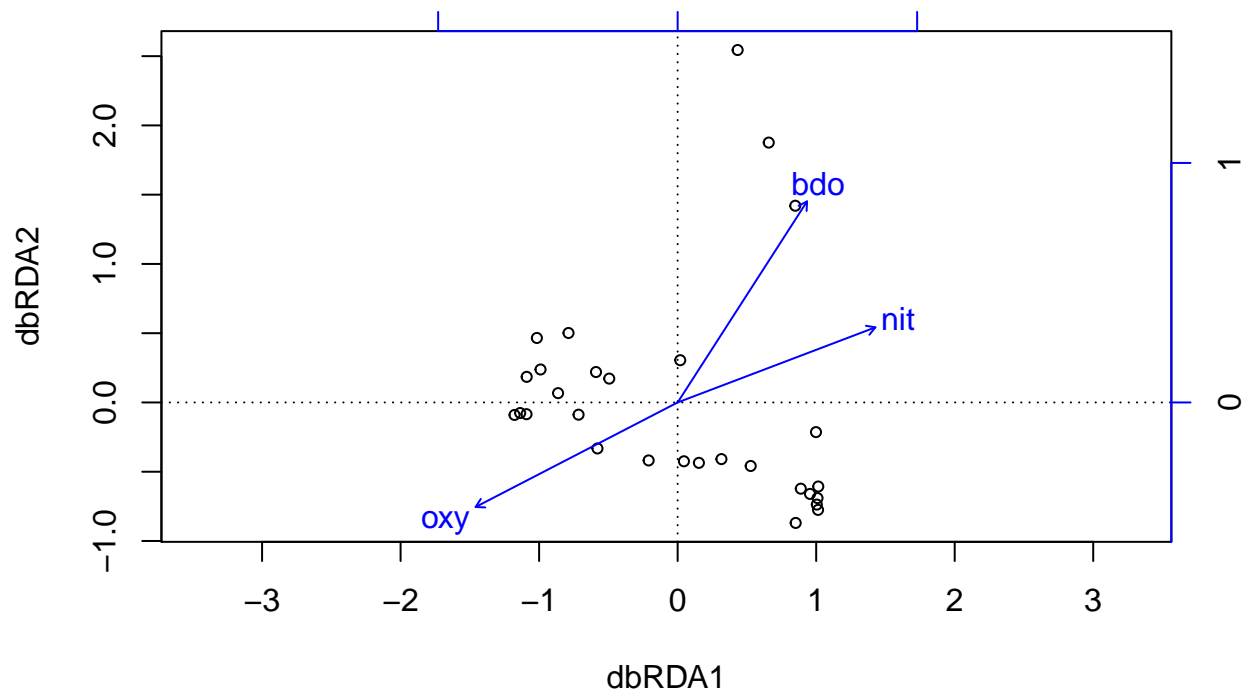
```
doubs.dbrda$call
```

```
## dbrda(formula = fish.db ~ oxy + bdo + nit, data = as.data.frame(env.chem))
```

```
doubs.dbrda$anova
```

```
##           R2.adj Df      AIC      F Pr(>F)
## + oxy       0.27727 1 47.939 11.7421 0.002 **
## + bdo       0.40090 1 43.404  6.5716 0.006 **
## + nit       0.49808 1 39.134  6.0340 0.004 **
## <All variables> 0.53033
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ordiplot(doubs.dbrda)
```



```
#3
permutest(doubs.dbrda, permutations = 999)
```

```
##
## Permutation test for dbrda
##
## Permutation: free
## Number of permutations: 999
##
## Call: dbrda(formula = fish.db ~ oxy + bdo + nit, data =
## as.data.frame(env.chem))
## Permutation test for all constrained eigenvalues
```

```
## Pseudo-F:      10.2619 (with 3, 25 Degrees of Freedom)
## Significance:   0.001
```

```
#4
envfit(doubs.dbrda, env.chem[,c(4, 6, 7)], perm = 999)
```

```
##
## ***VECTORS
##
##      dbRDA1   dbRDA2      r2 Pr(>r)
## nit  0.87724  0.48005 0.6431  0.001 ***
## oxy -0.82864 -0.55979 0.7656  0.001 ***
## bdo  0.55603  0.83116 0.8939  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```
#5
dbrda.explainvar1 <- round(doubs.dbrda$CCA$eig[1] /
                          sum(c(doubs.dbrda$CCA$eig, doubs.dbrda$CA$eig)), 3) * 100
dbrda.explainvar2 <- round(doubs.dbrda$CCA$eig[2] /
                          sum(c(doubs.dbrda$CCA$eig, doubs.dbrda$CA$eig)), 3) * 100
dbrda.explainvar1
```

```
## dbRDA1
##    38.6
```

```
dbrda.explainvar2
```

```
## dbRDA2
##     13
```

```
#6
par(mar = c(5, 5, 4, 4) + 0.1)

plot(scores(doubs.dbrda, display = "wa"), xlim = c(-3.5, 2), ylim = c(-3.2, 3.2),
xlab = paste("dbRDA 1 (", dbrda.explainvar1, "%)", sep = ""),
ylab = paste("dbRDA 2 (", dbrda.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(scores(doubs.dbrda, display = "wa"),
       pch = 19, cex = 3, bg = "gray", col = "gray")

text(scores(doubs.dbrda, display = "wa"),
     labels = row.names(scores(doubs.dbrda, display = "wa")))
```

```

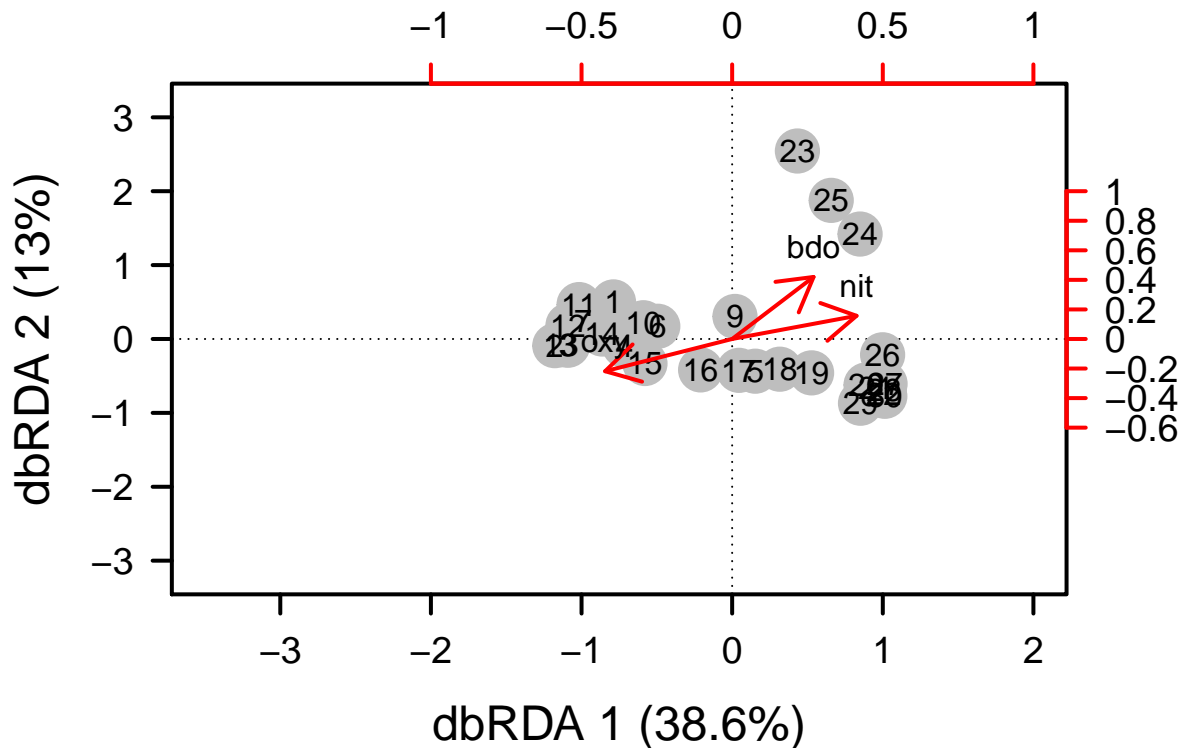
vectors <- scores(doubs.dbrda, display = "bp")

#row.names(vectors) <- c("pH", "har", "pho", "nit", "amm", "oxy", "bdo")

arrows(0, 0, vectors[,1], vectors[, 2],
      lwd = 2, lty = 1, length = 0.2, col = "red")
text(vectors[,1], vectors[, 2], pos = 3,
     labels = row.names(vectors))

axis(side = 3, lwd.ticks=2, cex.axis=1.2, las = 1, col = "red", lwd = 2.2,
     at = pretty(range(vectors[, 1])) * 2, labels = pretty(range(vectors[, 1])))
axis(side = 4, lwd.ticks=2, cex.axis=1.2, las = 1, col = "red", lwd = 2.2,
     at = pretty(range(vectors[, 2])) * 2, labels = pretty(range(vectors[, 2])))

```



Question 10: Based on the constrained ordination , what are the environmental variables (or groups of correlated variables) that seem to be contributing to variation in fish community structure?

Answer 10: Based on the distance-based redundancy analysis (dbRDA), the environmental variables **bdo**(biological demand for oxygen; $P = 0.001$), **nit**(nitrates; $P = 0.001$), and **oxy**(dissolved oxygen; $P = 0.001$) appear to be contributing the most to variation in fish community structure.

iii. Variation Partitioning

In the code chunk below,

1. Create a matrix model of the selected environmental variables,
2. Create a matrix model of the selected PCNM axes,
3. Perform constrained and partial constrained ordinations using the spatial and environmental models you just created,
4. Test the significance of each of your constrained ordinations using permutation tests,
5. Partition the variation among sites into the relative importance of space, environment, spatially structured environment, and residuals,
6. Plot the variation partitioning output to visualize it.

```
doubs.dbrda$anova
```

```
##              R2.adj Df      AIC      F Pr(>F)
## + oxy          0.27727  1 47.939 11.7421  0.002 **
## + bdo          0.40090  1 43.404  6.5716  0.006 **
## + nit          0.49808  1 39.134  6.0340  0.004 **
## <All variables> 0.53033
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
env.mod <- model.matrix(~ oxy + bdo + nit, as.data.frame(env.chem))[, -1]
rs <- rowSums(fish)/sum(fish)
```

```
doubs.pcnmw <- pcnm(dist(doubs$xy[-8,]), w = rs, dist.ret = T)
```

```
doubs.pcnmw$values > 0
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [12] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE
## [23] FALSE FALSE FALSE FALSE
```

```
doubs.space <- as.data.frame(scores(doubs.pcnmw))
doubs.pcnm.mod0 <- dbrda(fish.db ~ 1, doubs.space)
doubs.pcnm.mod1 <- dbrda(fish.db ~ ., doubs.space)
step.pcnm <- ordiR2step(doubs.pcnm.mod0, doubs.pcnm.mod1, perm.max = 200)
```

```
## Step: R2.adj= 0
## Call: fish.db ~ 1
##
##              R2.adjusted
## <All variables> 0.626011301
## + PCNM2          0.235370423
## + PCNM3          0.078394885
## + PCNM13         0.065305668
## + PCNM5          0.046185074
## + PCNM6          0.032809156
## + PCNM16         0.030486700
## + PCNM14         0.029680999
## + PCNM9          0.020357410
## + PCNM15         0.013632610
## + PCNM8          0.009411968
## + PCNM1          0.003986221
## + PCNM17         0.002415012
```

```

## + PCNM10          0.001326442
## <none>            0.000000000
## + PCNM7           -0.001861430
## + PCNM11          -0.006841522
## + PCNM4           -0.007089863
## + PCNM12          -0.014396973
##
##           Df      AIC      F Pr(>F)
## + PCNM2  1 49.574 9.619 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.2353704
## Call: fish.db ~ PCNM2
##
##           R2.adjusted
## <All variables> 0.6260113
## + PCNM3         0.3429270
## + PCNM5         0.3057368
## + PCNM1         0.2885396
## + PCNM16        0.2786746
## + PCNM14        0.2744520
## + PCNM15        0.2692809
## + PCNM6         0.2659866
## + PCNM13        0.2636194
## + PCNM9         0.2517847
## + PCNM8         0.2496240
## + PCNM10        0.2434688
## + PCNM7         0.2431476
## + PCNM17        0.2404343
## + PCNM11        0.2366833
## <none>          0.2353704
## + PCNM12        0.2288789
## + PCNM4         0.2189522
## - PCNM2         0.0000000
##
##           Df      AIC      F Pr(>F)
## + PCNM3  1 46.083 5.4196 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.342927
## Call: fish.db ~ PCNM2 + PCNM3
##
##           R2.adjusted
## <All variables> 0.62601130
## + PCNM5         0.40760197
## + PCNM1         0.39703000
## + PCNM16        0.38532100
## + PCNM15        0.38287481
## + PCNM14        0.37818268
## + PCNM13        0.37703761
## + PCNM6         0.35956442
## + PCNM8         0.35568849

```



```

## + PCNM7          0.35416308
## + PCNM10         0.35267745
## + PCNM17         0.35136832
## + PCNM9          0.34336720
## <none>           0.34292704
## + PCNM11         0.34163988
## + PCNM12         0.33965471
## + PCNM4          0.33115086
## - PCNM3          0.23537042
## - PCNM2          0.07839489
##
##           Df      AIC      F Pr(>F)
## + PCNM5    1 43.941 3.8385 0.016 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.407602
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5
##
##           R2.adjusted
## <All variables> 0.6260113
## + PCNM1         0.4721469
## + PCNM16        0.4631976
## + PCNM15        0.4589111
## + PCNM14        0.4535248
## + PCNM13        0.4511582
## + PCNM6         0.4305640
## + PCNM7         0.4261965
## + PCNM8         0.4224505
## + PCNM17        0.4181666
## + PCNM10        0.4154485
## + PCNM11        0.4112178
## + PCNM9         0.4111995
## + PCNM12        0.4087602
## <none>          0.4076020
## + PCNM4         0.3976526
## - PCNM5         0.3429270
## - PCNM3         0.3057368
## - PCNM2         0.1195237
##
##           Df      AIC      F Pr(>F)
## + PCNM1    1 41.411 4.057 0.006 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.4721469
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5 + PCNM1
##
##           R2.adjusted
## <All variables> 0.6260113
## + PCNM13        0.5212427
## + PCNM16        0.5208668
## + PCNM15        0.5161770
## + PCNM14        0.5147355

```

```

## + PCNM6          0.4999020
## + PCNM7          0.4936559
## + PCNM8          0.4904113
## + PCNM17         0.4856884
## + PCNM10         0.4835952
## + PCNM11         0.4760087
## + PCNM9          0.4751424
## + PCNM12         0.4747221
## <none>           0.4721469
## + PCNM4          0.4651218
## - PCNM1          0.4076020
## - PCNM5          0.3970300
## - PCNM3          0.3691841
## - PCNM2          0.1269210
##
##           Df      AIC      F Pr(>F)
## + PCNM13  1 39.346 3.4612 0.014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.5212427
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5 + PCNM1 + PCNM13
##
##           R2.adjusted
## <All variables> 0.6260113
## + PCNM16         0.5767968
## + PCNM15         0.5715331
## + PCNM14         0.5698343
## + PCNM6          0.5475140
## + PCNM7          0.5392074
## + PCNM8          0.5379134
## + PCNM11         0.5281106
## + PCNM9          0.5267003
## + PCNM10         0.5265029
## + PCNM12         0.5255581
## <none>           0.5212427
## + PCNM17         0.5171800
## + PCNM4          0.5152311
## - PCNM13         0.4721469
## - PCNM1          0.4511582
## - PCNM5          0.4350790
## - PCNM3          0.4111185
## - PCNM2          0.2307026
##
##           Df      AIC      F Pr(>F)
## + PCNM16  1 36.48 4.0192 0.018 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.5767968
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5 + PCNM1 + PCNM13 + PCNM16
##
##           R2.adjusted
## <All variables> 0.6260113

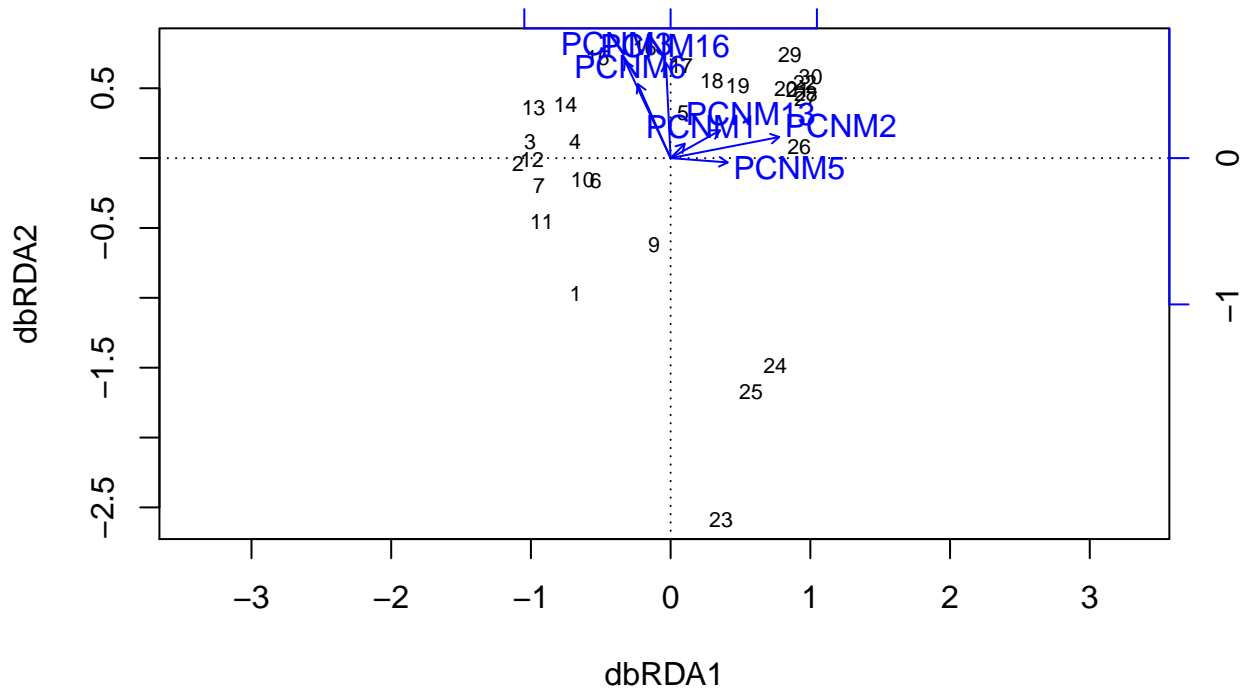
```

```

## + PCNM6          0.6043089
## + PCNM8          0.5970286
## + PCNM12         0.5946888
## + PCNM7          0.5946475
## + PCNM9          0.5883735
## + PCNM10         0.5851333
## + PCNM15         0.5846468
## <none>           0.5767968
## + PCNM17         0.5748533
## + PCNM4          0.5733749
## + PCNM11         0.5711176
## + PCNM14         0.5652509
## - PCNM16         0.5212427
## - PCNM13         0.5208668
## - PCNM1          0.5136241
## - PCNM5          0.4764463
## - PCNM3          0.4676690
## - PCNM2          0.2646853
##
##           Df      AIC      F Pr(>F)
## + PCNM6   1 35.182 2.5296 0.036 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.6043089
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5 + PCNM1 + PCNM13 + PCNM16 + PCNM6
##
##           R2.adjusted
## <All variables> 0.6260113
## + PCNM8          0.6248697
## + PCNM12         0.6208788
## + PCNM10         0.6170988
## + PCNM7          0.6142419
## + PCNM15         0.6140369
## + PCNM9          0.6107110
## <none>           0.6043089
## + PCNM17         0.6037430
## + PCNM11         0.5978305
## + PCNM4          0.5963667
## + PCNM14         0.5932113
## - PCNM6          0.5767968
## - PCNM13         0.5495723
## - PCNM16         0.5475140
## - PCNM1          0.5362957
## - PCNM3          0.5092539
## - PCNM5          0.4965968
## - PCNM2          0.2824470
##
##           Df      AIC      F Pr(>F)
## + PCNM8   1 34.219 2.151 0.092 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
plot(step.pcnm)
```



```
step.pcnm$anova
```

```
##           R2.adj Df      AIC      F Pr(>F)
## + PCNM2      0.23537  1 49.574 9.6190 0.002 **
## + PCNM3      0.34293  1 46.083 5.4196 0.002 **
## + PCNM5      0.40760  1 43.941 3.8385 0.016 *
## + PCNM1      0.47215  1 41.411 4.0570 0.006 **
## + PCNM13     0.52124  1 39.346 3.4612 0.014 *
## + PCNM16     0.57680  1 36.480 4.0192 0.018 *
## + PCNM6      0.60431  1 35.182 2.5296 0.036 *
## <All variables> 0.62601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
space.mod <- model.matrix(~ PCNM2 + PCNM3 + PCNM5 + PCNM1 + PCNM13 + PCNM16 + PCNM6, doub.s.space)[-1]
```

```
doub.s.total.env <- dbrda(fish.db ~ env.mod)
doub.s.total.space <- dbrda(fish.db ~ space.mod)
```

```
doub.s.env.cond.space <- dbrda(fish.db ~ env.mod + Condition(space.mod))
doub.s.space.cond.env <- dbrda(fish.db ~ space.mod + Condition(space.mod))
```

```
permutest(doubs.env.cond.space, permutations = 999)
```

```
##  
## Permutation test for dbrda  
##  
## Permutation: free  
## Number of permutations: 999  
##  
## Call: dbrda(formula = fish.db ~ env.mod + Condition(space.mod))  
## Permutation test for all constrained eigenvalues  
## Pseudo-F:      4.423025 (with 3, 18 Degrees of Freedom)  
## Significance:    0.001
```

```
permutest(doubs.space.cond.env, permutations = 999)
```

```
##  
## Permutation test for dbrda  
##  
##  
## Call: dbrda(formula = fish.db ~ space.mod + Condition(space.mod))  
## Permutation test for all constrained eigenvalues  
## Pseudo-F:      NA (with 0, 21 Degrees of Freedom)  
## Significance:    NA
```

```
permutest(doubs.total.env, permutations = 999)
```

```
##  
## Permutation test for dbrda  
##  
## Permutation: free  
## Number of permutations: 999  
##  
## Call: dbrda(formula = fish.db ~ env.mod)  
## Permutation test for all constrained eigenvalues  
## Pseudo-F:      10.2619 (with 3, 25 Degrees of Freedom)  
## Significance:    0.001
```

```
permutest(doubs.total.space, permutations = 999)
```

```
##  
## Permutation test for dbrda  
##  
## Permutation: free  
## Number of permutations: 999  
##  
## Call: dbrda(formula = fish.db ~ space.mod)  
## Permutation test for all constrained eigenvalues  
## Pseudo-F:      7.108896 (with 7, 21 Degrees of Freedom)  
## Significance:    0.001
```

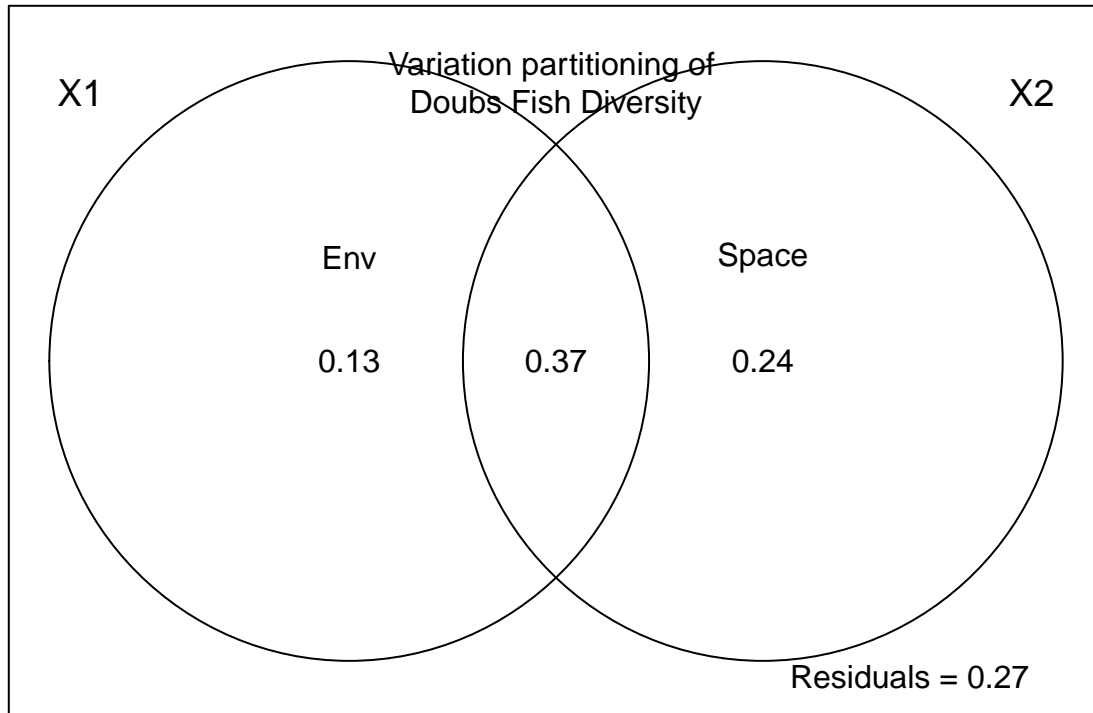
```
doubs.varpart <- varpart(fish.db, env.mod, space.mod)
doubs.varpart
```

```
##
## Partition of squared Bray distance in dbRDA
##
## Call: varpart(Y = fish.db, X = env.mod, space.mod)
##
## Explanatory tables:
## X1:  env.mod
## X2:  space.mod
##
## No. of explanatory tables: 2
## Total variation (SS): 6.7621
## No. of observations: 29
##
## Partition table:
##
```

	Df	R.squared	Adj.R.squared	Testable
[a+b] = X1	3	0.55186	0.49808	TRUE
[b+c] = X2	7	0.70323	0.60431	TRUE
[a+b+c] = X1+X2	10	0.82917	0.73426	TRUE
## Individual fractions				
[a] = X1 X2	3		0.12995	TRUE
[b]	0		0.36813	FALSE
[c] = X2 X1	7		0.23618	TRUE
[d] = Residuals			0.26574	FALSE

```
## ---
## Use function 'capscale' to test significance of fractions of interest
```

```
par(mar = c(2,2,2,2))
plot(doubs.varpart)
text(1, 0.25, "Space")
text(0, 0.25, "Env")
mtext("Variation partitioning of \nDoubs Fish Diversity", side = 3, line = -3)
```



Question 11: Interpret the variation partitioning results.

Answer 11: As one might expect, the highest amount of variation (37%) in fish diversity is explained by both environment and space. Space exclusively explains more variation in diversity than environment (24% vs 13%). This is a common finding...at least by my experience with the literature. Even when populations are experiencing very similar environments, isolation-by-distance is typically very strong. Mating and social behavior may limit fish in one region from moving to another. As well, physical boundaries or strong currents may prohibit homogenization of the river's diversity.

SYNTHESIS

Load the dataset you are using for your project. Perform an ordination to visualize your dataset. Using this ordination, develop some hypotheses relevant to β -diversity and identify the appropriate tools you would use to test them.

```
setwd("c:/Users/matth/Documents/bin/QB2017_Gibson/project")
myData <- read.table("data/speciesdata_clean.csv", sep=",", header=T, row.names = 1)
#Remove unneeded data
myData <- myData[, 4:ncol(myData)]

spec.bray <- vegdist(myData, method = "bray")

####Waaaay to many sites to get anything out of the heatmap...
```

```

#order <- rev(attr(spec.bray, "Labels"))

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

spec.pcoa <- cmdscale(spec.bray, eig=T, k=3)
explainvar1 <- round(spec.pcoa$eig[1] / sum(spec.pcoa$eig), 3) * 100
explainvar2 <- round(spec.pcoa$eig[2] / sum(spec.pcoa$eig), 3) * 100
explainvar3 <- round(spec.pcoa$eig[3] / sum(spec.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

#Variance explained by first axis
explainvar1

```

```
## [1] 17
```

```

#variance explained by second axis
explainvar2

```

```
## [1] 12.5
```

```

#variance explained by third axis
explainvar3

```

```
## [1] 7.7
```

```
sum.eig
```

```
## [1] 37.2
```

```

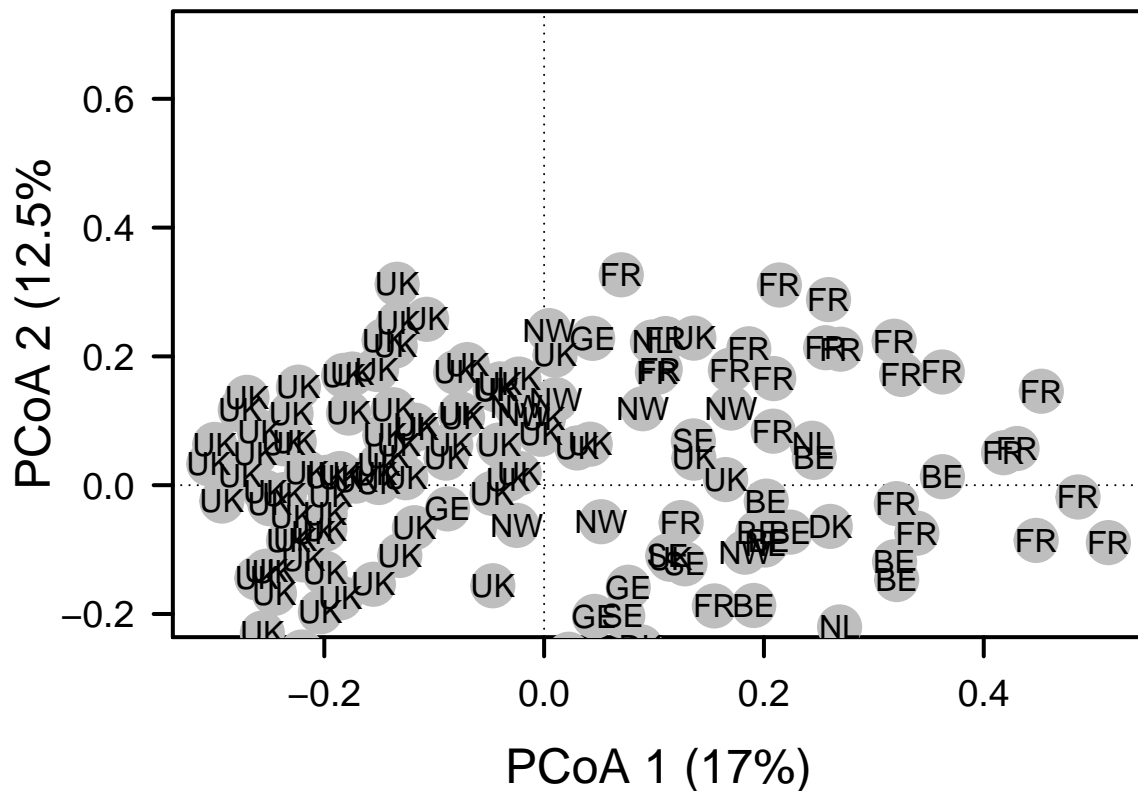
par(mar = c(5, 5, 1, 2) + 0.1)

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

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)

#I plotted just the country code...full site names were too long
points(spec.pcoa$points[,1], spec.pcoa$points[,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(spec.pcoa$points[,1], spec.pcoa$points[,2],
     labels = substr(row.names(spec.pcoa$points),1,2))

```

```
specREL <- myData
for(i in 1:nrow(myData)){
  specREL[i, ] = myData[i, ]/ sum(myData[i, ])
}

spec.pcoa <- add.spec.scores(spec.pcoa, specREL, method = "pcoa.scores")

#Is just a mess.....
#text(spec.pcoa$cproj[,1], spec.pcoa$cproj[,2],
#      labels = row.names(spec.pcoa$cproj), col = "black")

spe.corr <- add.spec.scores(spec.pcoa, specREL, method = "cor.scores")$cproj
corrcut <- 0.8
imp.spp <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]

#As expected, this takes quite a long time...
fit <- envfit(spec.pcoa, specREL, perm = 999)
```

Based on the ordination, sites appear to group by country. This is expected. We could do a PERMANOVA using `country` as a factor to formally test this. Though I am perhaps most interested in what environmental factors underlie this grouping. Further I would be interested in testing what the relative contributions of isolation-by-distance and isolation-by-environment are on the observed variation in species diversity.

We could begin our analysis by doing a Mantel test with the site X env (removing lat lon data) and site X species matrices. Next, we could do a constrained coordination (either an RDA or CCA) implementing

variable selection procedures to identify environmental variables contributing to species community structure (the variables that define the environments of the several European regions).

Next, using the environmental variables identified as being important and the latitude and longitude values for each site, we could do partial constrained ordination and variance partitioning to calculate the fractions of variance in species diversity explained by distance, environment, and by both distance and environment.