Assignment: Among Site (Beta) Diversity

Venus Kuo; 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 wit 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).

```r
# Set working directory #
rm(list = ls())
getwd()
```

```
## [1] "C:/Users/Venus/Github/QB2017_Kuo/Week3-Beta"
```

```r
setwd("C:/Users/Venus/Github/QB2017_Kuo/Week3-Beta/")

# Require or install packages #
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)
} }
```

This is vegan 2.4-1

##
Attaching package: 'ade4'

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

##
Attaching package: 'gplots'

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

BiodiversityR 2.7-2: use function 'BiodiversityRGUI()' to launch the BiodiversityR Graphical User In

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.

```
# Loading doubs dataset #
data(doubs)

# Observing structure of doubs dataset #
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***: 4 objects (env, fish, xy, and species) ***Answer 1b***: 27 species ***Answer 1c***: 30
> sites(xy)

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***: There is a lot of heterogenity in the rish richness along the sampled Doubs River. Generally, fish richness is higher (S=30) downstream and lower (S=10,0) upstream, while the intermediate zone was a mixure of medium (S=20) richness. ***Answer 2b***: Brown troubte abundnace is low or absent downstream but is very abundant upstream and in between down and upstream. There is still a lot of heterogenity in the sampled sites. ***Answer 2c***: Just looking at richness local sites along a river is not very informative when trying to compare the richness among sites (beta-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.

```r
# Modified function for pariwise beta-diversity for turnover #
beta.w <- function(site.by.species = "", sitenum1 = "", sitenum2 = "", pairwise = FALSE) {
  # ONLY if we specify pairwise as TRUE, do this:
  if (pairwise == TRUE) {

    # As a check, let's print an error if we do not provide needed arguements
    if (sitenum1 == "" | sitenum2 == "") {
      print("Error: please specify sites to compare")
      return(NA)}

    # If our function made it this far, let's calculate pairwise beta diversity
    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)
    a.bar = mean(c(specnumber(site1), specnumber(site2)))
    b.w   = round(s/a.bar - 1, 3)
    return(b.w)
  }
# OTHERWISE pairwise defaults to FALSE, so do this, like before:
  else{
    SbyS.pa <- decostand(site.by.species, method = "pa")
```

3

```
    S <- ncol(SbyS.pa[ , which(colSums(SbyS.pa) > 0)])
    a.bar <- mean(specnumber(SbyS.pa))
    b.w <- round(S/a.bar, 3)
    return(b.w)
  }
}

# Doubs dataset #
beta.w(site.by.species = doubs$fish, sitenum1 = "1", sitenum2 = "10" , pairwise=TRUE)
```

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

```
beta.w(site.by.species = doubs$fish, sitenum1 = "1", sitenum2 = "2" , pairwise=TRUE)
```

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

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***: The equation for Whittaker's β diversity is defined as γ/α-1. γ is dependent on the richness across all sites, so all α. If α increases in most sites then it would be reasonable to conclude that regional diversity would increase as well. Moreover, regional diversity is the product of α and β, so the rate of turnover or differences between local site diversity is what constitutes regional diversity. ***Answer 3b***: The beta diversity of fish between 1 and 2 is 0.5, while the beta diversity between 1 and 10 is 0.714. So site 1 is more similar to site 10 than to site 2. ***Answer 3c***: If we defined beta additively then beta diversity would be defined by the regional richness (gamma) subtracted with the local diversity of one site $\beta = \gamma - \alpha$. In other words, beta diversity would be the number of species that is not present within a local site.

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 (the presence/absence) metrics counts rare species equally to common species. But abundance-based metrics will weight common species more heaviliy than singlton or doubleton (rare) species.

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 SÃ¸rensen's Similarity ("fish.ds"), and
4. construct a resemblance matrix based on Bray-Curtis Distance ("fish.db").

```
# Make object fish #
fish <- doubs$fish
# Remove sites where no fish were observed #
fish <- fish[-8, ]
# Construct resemblance matrix based on Sorensen's similarity #
fish.ds <- vegdist(fish, method = "bray", binary = TRUE, diag = TRUE)
# Construct resemblance matrix based on Bray-Curtis #
```

```r
fish.db <- vegdist(fish, method = "bray", upper = TRUE , diag = TRUE)
fish.db
```

```
##             1          2          3          4          5          6
## 1  0.00000000 0.60000000 0.68421053 0.75000000 0.89189189 0.75000000
## 2  0.60000000 0.00000000 0.14285714 0.33333333 0.69565217 0.39393939
## 3  0.68421053 0.14285714 0.00000000 0.18918919 0.68000000 0.29729730
## 4  0.75000000 0.33333333 0.18918919 0.00000000 0.49090909 0.19047619
## 5  0.89189189 0.69565217 0.68000000 0.49090909 0.00000000 0.41818182
## 6  0.75000000 0.39393939 0.29729730 0.19047619 0.41818182 0.00000000
## 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
## 10 0.88235294 0.38461538 0.40000000 0.37142857 0.54166667 0.25714286
## 11 0.57142857 0.30434783 0.40740741 0.43750000 0.68888889 0.43750000
## 12 0.71428571 0.20000000 0.23529412 0.33333333 0.69230769 0.38461538
## 13 0.72727273 0.29032258 0.31428571 0.45000000 0.73584906 0.55000000
## 14 0.80645161 0.40000000 0.31818182 0.34693878 0.67741935 0.42857143
## 15 0.83333333 0.51111111 0.46938776 0.40740741 0.55223881 0.37037037
## 16 0.86046512 0.65384615 0.57142857 0.47540984 0.45945946 0.37704918
## 17 0.91489362 0.67857143 0.63333333 0.50769231 0.51282051 0.44615385
## 18 0.95555556 0.74074074 0.72413793 0.58730159 0.50000000 0.52380952
## 19 1.00000000 0.79310345 0.70967742 0.61194030 0.50000000 0.52238806
## 20 1.00000000 0.91176471 0.88888889 0.74025974 0.48888889 0.68831169
## 21 1.00000000 0.94594595 0.92307692 0.78313253 0.50000000 0.73493976
## 22 1.00000000 0.97619048 0.95454545 0.82795699 0.52830189 0.78494624
## 23 1.00000000 1.00000000 1.00000000 0.92000000 0.89473684 0.84000000
## 24 1.00000000 1.00000000 1.00000000 0.88888889 0.79591837 0.77777778
## 25 1.00000000 1.00000000 0.92592593 0.81250000 0.68888889 0.68750000
## 26 1.00000000 0.96363636 0.93220339 0.78125000 0.55844156 0.68750000
## 27 1.00000000 0.97333333 0.94936709 0.83333333 0.56701031 0.76190476
## 28 1.00000000 0.97560976 0.95348837 0.82417582 0.57692308 0.78021978
## 29 0.97777778 0.93939394 0.92233010 0.81481481 0.53719008 0.77777778
## 30 1.00000000 1.00000000 0.98095238 0.87272727 0.59349593 0.83636364
##             7          9         10         11         12         13
## 1  0.68421053 1.00000000 0.88235294 0.57142857 0.71428571 0.72727273
## 2  0.14285714 0.69230769 0.38461538 0.30434783 0.20000000 0.29032258
## 3  0.12500000 0.73333333 0.40000000 0.40740741 0.23529412 0.31428571
## 4  0.24324324 0.65714286 0.37142857 0.43750000 0.33333333 0.45000000
## 5  0.64000000 0.58333333 0.54166667 0.68888889 0.69230769 0.73584906
## 6  0.24324324 0.54285714 0.25714286 0.43750000 0.38461538 0.55000000
## 7  0.00000000 0.66666667 0.26666667 0.33333333 0.17647059 0.37142857
## 9  0.66666667 0.00000000 0.57142857 0.76000000 0.68750000 0.81818182
## 10 0.26666667 0.57142857 0.00000000 0.44000000 0.37500000 0.57575758
## 11 0.33333333 0.76000000 0.44000000 0.00000000 0.24137931 0.33333333
## 12 0.17647059 0.68750000 0.37500000 0.24137931 0.00000000 0.18918919
## 13 0.37142857 0.81818182 0.57575758 0.33333333 0.18918919 0.00000000
## 14 0.36363636 0.76190476 0.47619048 0.43589744 0.21739130 0.19148936
## 15 0.38775510 0.65957447 0.40425532 0.50000000 0.33333333 0.38461538
## 16 0.53571429 0.70370370 0.51851852 0.64705882 0.55172414 0.59322034
## 17 0.60000000 0.68965517 0.51724138 0.63636364 0.58064516 0.61904762
## 18 0.68965517 0.64285714 0.57142857 0.69811321 0.66666667 0.70491803
## 19 0.67741935 0.66666667 0.63333333 0.82456140 0.75000000 0.81538462
## 20 0.86111111 0.68571429 0.77142857 0.91044776 0.89189189 0.92000000
## 21 0.89743590 0.76315789 0.81578947 0.91780822 0.92500000 0.95061728
```

```
## 22 0.93181818 0.76744186 0.86046512 0.95180723 0.95555556 0.97802198
## 23 0.90000000 0.77777778 0.88888889 0.86666667 0.90909091 1.00000000
## 24 0.93548387 0.72413793 0.79310345 0.92307692 0.93939394 1.00000000
## 25 0.85185185 0.84000000 0.76000000 0.90909091 0.93103448 1.00000000
## 26 0.89830508 0.71929825 0.82456140 0.92592593 0.93442623 0.96774194
## 27 0.92405063 0.76623377 0.84415584 0.94594595 0.95061728 0.97560976
## 28 0.93023256 0.76190476 0.85714286 0.95061728 0.95454545 0.97752809
## 29 0.90291262 0.78217822 0.84158416 0.89795918 0.90476190 0.90566038
## 30 0.96190476 0.84466019 0.90291262 0.98000000 0.98130841 1.00000000
##             14         15         16         17         18         19
## 1  0.80645161 0.83333333 0.86046512 0.91489362 0.95555556 1.00000000
## 2  0.40000000 0.51111111 0.65384615 0.67857143 0.74074074 0.79310345
## 3  0.31818182 0.46938776 0.57142857 0.63333333 0.72413793 0.70967742
## 4  0.34693878 0.40740741 0.47540984 0.50769231 0.58730159 0.61194030
## 5  0.67741935 0.55223881 0.45945946 0.51282051 0.50000000 0.50000000
## 6  0.42857143 0.37037037 0.37704918 0.44615385 0.52380952 0.52238806
## 7  0.36363636 0.38775510 0.53571429 0.60000000 0.68965517 0.67741935
## 9  0.76190476 0.65957447 0.70370370 0.68965517 0.64285714 0.66666667
## 10 0.47619048 0.40425532 0.51851852 0.51724138 0.57142857 0.63333333
## 11 0.43589744 0.50000000 0.64705882 0.63636364 0.69811321 0.82456140
## 12 0.21739130 0.33333333 0.55172414 0.58064516 0.66666667 0.75000000
## 13 0.19148936 0.38461538 0.59322034 0.61904762 0.70491803 0.81538462
## 14 0.00000000 0.24590164 0.44117647 0.50000000 0.60000000 0.67567568
## 15 0.24590164 0.00000000 0.26027397 0.40259740 0.46666667 0.56962025
## 16 0.44117647 0.26027397 0.00000000 0.26190476 0.34146341 0.39534884
## 17 0.50000000 0.40259740 0.26190476 0.00000000 0.13953488 0.31111111
## 18 0.60000000 0.46666667 0.34146341 0.13953488 0.00000000 0.25000000
## 19 0.67567568 0.56962025 0.39534884 0.31111111 0.25000000 0.00000000
## 20 0.83333333 0.70786517 0.58333333 0.42000000 0.32653061 0.23529412
## 21 0.86666667 0.76842105 0.62745098 0.49056604 0.40384615 0.29629630
## 22 0.90000000 0.77142857 0.66071429 0.55172414 0.47368421 0.38983051
## 23 0.93750000 0.94594595 0.90909091 0.83333333 0.82608696 0.84000000
## 24 0.90697674 0.87500000 0.81818182 0.69491525 0.64912281 0.63934426
## 25 0.84615385 0.81818182 0.76470588 0.74545455 0.66037736 0.61403509
## 26 0.85915493 0.76315789 0.63855422 0.54022989 0.45882353 0.32584270
## 27 0.89010989 0.77083333 0.66990291 0.57009346 0.48571429 0.37614679
## 28 0.89795918 0.78640777 0.69090909 0.57894737 0.50000000 0.41379310
## 29 0.84347826 0.73333333 0.65354331 0.51145038 0.44186047 0.41353383
## 30 0.93162393 0.81967213 0.72093023 0.57894737 0.52671756 0.48148148
##             20         21         22         23         24         25
## 1  1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
## 2  0.91176471 0.94594595 0.97619048 1.00000000 1.00000000 1.00000000
## 3  0.88888889 0.92307692 0.95454545 1.00000000 1.00000000 0.92592593
## 4  0.74025974 0.78313253 0.82795699 0.92000000 0.88888889 0.81250000
## 5  0.48888889 0.50000000 0.52830189 0.89473684 0.79591837 0.68888889
## 6  0.68831169 0.73493976 0.78494624 0.84000000 0.77777778 0.68750000
## 7  0.86111111 0.89743590 0.93181818 0.90000000 0.93548387 0.85185185
## 9  0.68571429 0.76315789 0.76744186 0.77777778 0.72413793 0.84000000
## 10 0.77142857 0.81578947 0.86046512 0.88888889 0.79310345 0.76000000
## 11 0.91044776 0.91780822 0.95180723 0.86666667 0.92307692 0.90909091
## 12 0.89189189 0.92500000 0.95555556 0.90909091 0.93939394 0.93103448
## 13 0.92000000 0.95061728 0.97802198 1.00000000 1.00000000 1.00000000
## 14 0.83333333 0.86666667 0.90000000 0.93750000 0.90697674 0.84615385
## 15 0.70786517 0.76842105 0.77142857 0.94594595 0.87500000 0.81818182
```

6

```
## 16 0.58333333 0.62745098 0.66071429 0.90909091 0.81818182 0.76470588
## 17 0.42000000 0.49056604 0.55172414 0.83333333 0.69491525 0.74545455
## 18 0.32653061 0.40384615 0.47368421 0.82608696 0.64912281 0.66037736
## 19 0.23529412 0.29629630 0.38983051 0.84000000 0.63934426 0.61403509
## 20 0.00000000 0.10169492 0.18750000 0.86666667 0.57746479 0.67164179
## 21 0.10169492 0.00000000 0.10447761 0.87878788 0.61038961 0.69863014
## 22 0.18750000 0.10447761 0.00000000 0.89473684 0.65517241 0.73493976
## 23 0.86666667 0.87878788 0.89473684 0.00000000 0.57894737 0.46666667
## 24 0.57746479 0.61038961 0.65517241 0.57894737 0.00000000 0.46153846
## 25 0.67164179 0.69863014 0.73493976 0.46666667 0.46153846 0.00000000
## 26 0.21212121 0.20000000 0.25217391 0.82978723 0.48275862 0.59259259
## 27 0.19327731 0.13600000 0.12592593 0.88059701 0.61538462 0.70270270
## 28 0.22222222 0.16666667 0.12676056 0.89189189 0.64705882 0.72839506
## 29 0.24475524 0.18120805 0.11949686 0.91208791 0.70588235 0.77551020
## 30 0.29655172 0.23178808 0.18012422 0.91397849 0.71153846 0.78000000
##              26         27         28         29         30
## 1    1.00000000 1.00000000 1.00000000 0.97777778 1.00000000
## 2    0.96363636 0.97333333 0.97560976 0.93939394 1.00000000
## 3    0.93220339 0.94936709 0.95348837 0.92233010 0.98095238
## 4    0.78125000 0.83333333 0.82417582 0.81481481 0.87272727
## 5    0.55844156 0.56701031 0.57692308 0.53719008 0.59349593
## 6    0.68750000 0.76190476 0.78021978 0.77777778 0.83636364
## 7    0.89830508 0.92405063 0.93023256 0.90291262 0.96190476
## 9    0.71929825 0.76623377 0.76190476 0.78217822 0.84466019
## 10   0.82456140 0.84415584 0.85714286 0.84158416 0.90291262
## 11   0.92592593 0.94594595 0.95061728 0.89795918 0.98000000
## 12   0.93442623 0.95061728 0.95454545 0.90476190 0.98130841
## 13   0.96774194 0.97560976 0.97752809 0.90566038 1.00000000
## 14   0.85915493 0.89010989 0.89795918 0.84347826 0.93162393
## 15   0.76315789 0.77083333 0.78640777 0.73333333 0.81967213
## 16   0.63855422 0.66990291 0.69090909 0.65354331 0.72093023
## 17   0.54022989 0.57009346 0.57894737 0.51145038 0.57894737
## 18   0.45882353 0.48571429 0.50000000 0.44186047 0.52671756
## 19   0.32584270 0.37614679 0.41379310 0.41353383 0.48148148
## 20   0.21212121 0.19327731 0.22222222 0.24475524 0.29655172
## 21   0.20000000 0.13600000 0.16666667 0.18120805 0.23178808
## 22   0.25217391 0.12592593 0.12676056 0.11949686 0.18012422
## 23   0.82978723 0.88059701 0.89189189 0.91208791 0.91397849
## 24   0.48275862 0.61538462 0.64705882 0.70588235 0.71153846
## 25   0.59259259 0.70270270 0.72839506 0.77551020 0.78000000
## 26   0.00000000 0.18867925 0.23893805 0.33846154 0.36363636
## 27   0.18867925 0.00000000 0.09774436 0.18666667 0.19736842
## 28   0.23893805 0.09774436 0.00000000 0.14649682 0.15723270
## 29   0.33846154 0.18666667 0.14649682 0.00000000 0.14772727
## 30   0.36363636 0.19736842 0.15723270 0.14772727 0.00000000
```

`fish.ds`

```
##            1          2          3          4          5          6
## 1  0.00000000
## 2  0.50000000 0.00000000
## 3  0.60000000 0.14285714 0.00000000
## 4  0.77777778 0.45454545 0.33333333 0.00000000
## 5  0.83333333 0.57142857 0.46666667 0.15789474 0.00000000
## 6  0.81818182 0.53846154 0.42857143 0.11111111 0.04761905 0.00000000
```

```
## 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
## 10 0.71428571 0.33333333 0.40000000 0.28571429 0.29411765 0.25000000
## 11 0.71428571 0.33333333 0.40000000 0.42857143 0.52941176 0.50000000
## 12 0.71428571 0.33333333 0.40000000 0.42857143 0.52941176 0.50000000
## 13 0.71428571 0.33333333 0.40000000 0.57142857 0.64705882 0.62500000
## 14 0.81818182 0.53846154 0.42857143 0.33333333 0.42857143 0.40000000
## 15 0.83333333 0.57142857 0.60000000 0.36842105 0.36363636 0.33333333
## 16 0.88888889 0.70000000 0.61904762 0.36000000 0.28571429 0.25925926
## 17 0.91304348 0.76000000 0.69230769 0.46666667 0.39393939 0.37500000
## 18 0.91666667 0.76923077 0.70370370 0.48387097 0.41176471 0.39393939
## 19 1.00000000 0.84615385 0.77777778 0.54838710 0.41176471 0.45454545
## 20 1.00000000 0.84000000 0.76923077 0.53333333 0.39393939 0.43750000
## 21 1.00000000 0.84615385 0.77777778 0.54838710 0.41176471 0.45454545
## 22 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000
## 23 1.00000000 1.00000000 1.00000000 0.81818182 0.71428571 0.69230769
## 24 1.00000000 1.00000000 1.00000000 0.75000000 0.68421053 0.66666667
## 25 1.00000000 1.00000000 0.83333333 0.62500000 0.36842105 0.44444444
## 26 1.00000000 0.91666667 0.84000000 0.58620690 0.43750000 0.48387097
## 27 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000
## 28 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000
## 29 0.92592593 0.79310345 0.73333333 0.52941176 0.40540541 0.44444444
## 30 1.00000000 1.00000000 0.92000000 0.65517241 0.50000000 0.54838710
##              7          9         10         11         12         13
## 1
## 2
## 3
## 4
## 5
## 6
## 7  0.00000000
## 9  0.40000000 0.00000000
## 10 0.09090909 0.45454545 0.00000000
## 11 0.27272727 0.45454545 0.33333333 0.00000000
## 12 0.27272727 0.45454545 0.33333333 0.00000000 0.00000000
## 13 0.45454545 0.63636364 0.50000000 0.16666667 0.16666667 0.00000000
## 14 0.46666667 0.60000000 0.37500000 0.25000000 0.25000000 0.25000000
## 15 0.37500000 0.50000000 0.29411765 0.29411765 0.29411765 0.29411765
## 16 0.54545455 0.54545455 0.47826087 0.56521739 0.56521739 0.56521739
## 17 0.62962963 0.62962963 0.57142857 0.57142857 0.57142857 0.57142857
## 18 0.64285714 0.64285714 0.58620690 0.58620690 0.58620690 0.58620690
## 19 0.71428571 0.64285714 0.65517241 0.79310345 0.79310345 0.79310345
## 20 0.70370370 0.62962963 0.64285714 0.78571429 0.78571429 0.85714286
## 21 0.71428571 0.64285714 0.65517241 0.79310345 0.79310345 0.86206897
## 22 0.77777778 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143
## 23 0.75000000 0.50000000 0.77777778 0.77777778 0.77777778 1.00000000
## 24 0.84615385 0.69230769 0.71428571 0.85714286 0.85714286 1.00000000
## 25 0.69230769 0.69230769 0.57142857 0.85714286 0.85714286 1.00000000
## 26 0.76923077 0.69230769 0.70370370 0.85185185 0.85185185 0.92592593
## 27 0.77777778 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143
## 28 0.77777778 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143
## 29 0.67741935 0.67741935 0.62500000 0.68750000 0.68750000 0.68750000
## 30 0.84615385 0.76923077 0.77777778 0.92592593 0.92592593 1.00000000
##             14         15         16         17         18         19
```

```
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14 0.00000000
## 15 0.14285714 0.00000000
## 16 0.33333333 0.28571429 0.00000000
## 17 0.37500000 0.33333333 0.12820513 0.00000000
## 18 0.39393939 0.35294118 0.15000000 0.02222222 0.00000000
## 19 0.57575758 0.52941176 0.25000000 0.15555556 0.13043478 0.00000000
## 20 0.62500000 0.57575758 0.28205128 0.18181818 0.15555556 0.02222222
## 21 0.63636364 0.58823529 0.30000000 0.20000000 0.17391304 0.04347826
## 22 0.68750000 0.63636364 0.33333333 0.22727273 0.20000000 0.06666667
## 23 0.84615385 0.85714286 0.80000000 0.76000000 0.76923077 0.76923077
## 24 0.77777778 0.78947368 0.68000000 0.60000000 0.54838710 0.48387097
## 25 0.66666667 0.68421053 0.60000000 0.60000000 0.54838710 0.48387097
## 26 0.67741935 0.62500000 0.36842105 0.25581395 0.22727273 0.09090909
## 27 0.68750000 0.63636364 0.33333333 0.22727273 0.20000000 0.06666667
## 28 0.68750000 0.63636364 0.33333333 0.22727273 0.20000000 0.06666667
## 29 0.50000000 0.45945946 0.25581395 0.12500000 0.10204082 0.06122449
## 30 0.74193548 0.68750000 0.36842105 0.25581395 0.22727273 0.09090909
##                20         21         22         23         24         25
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20 0.00000000
## 21 0.02222222 0.00000000
## 22 0.04545455 0.02222222 0.00000000
## 23 0.76000000 0.76923077 0.76000000 0.00000000
## 24 0.46666667 0.48387097 0.46666667 0.45454545 0.00000000
## 25 0.46666667 0.48387097 0.46666667 0.45454545 0.37500000 0.00000000
```

```
## 26 0.06976744 0.04545455 0.02325581 0.75000000 0.44827586 0.44827586
## 27 0.04545455 0.02222222 0.00000000 0.76000000 0.46666667 0.46666667
## 28 0.04545455 0.02222222 0.00000000 0.76000000 0.46666667 0.46666667
## 29 0.08333333 0.06122449 0.08333333 0.79310345 0.52941176 0.52941176
## 30 0.06976744 0.04545455 0.02325581 0.75000000 0.44827586 0.44827586
##                 26          27          28          29          30
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26 0.00000000
## 27 0.02325581 0.00000000
## 28 0.02325581 0.00000000 0.00000000
## 29 0.10638298 0.08333333 0.08333333 0.00000000
## 30 0.04761905 0.02325581 0.02325581 0.10638298 0.00000000
```

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 SÃ¸rensen or Bray-Curtis distance influence your interpretation of site (dis)similarity?

   Answer 5a: Dissimilarity because when I added the vegdist arguement to include the diagonal, the value was zero which means that the difference between a site by itself is zero.
   Answer 5b: Sorenson index simply compares the number of shared species to the mean of species in a local site - So this uses incidence-based data and doesn't consider abundance. While Bray-Curtis dissimilarity is a quantitative version of the Sorenson index and considers the abundance of species - The Bray curtis dissimilarity values are slightly different than that of Sorensons, so abundance is likely quite uneven in this dataset.

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.

```
# Color palette already predefined #
# Define order of sites #
order <- rev(attr(fish.db, "Labels"))

# Plot Heatmap #
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
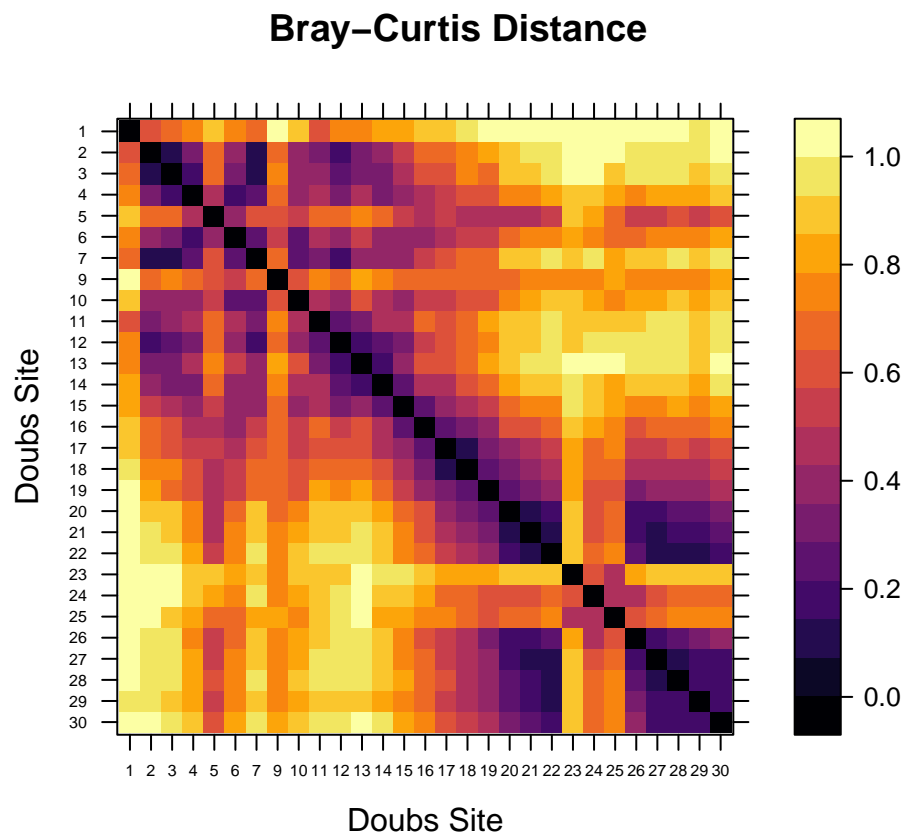
B. Cluster Analysis

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`).

11

```
# Perform cluster analysis using Ward's Clustering #
fish.ward <- hclust(fish.db, method = "ward.D2")

# Plot Cluster #
par(mar = c(1, 5, 2, 2) + 0.1)
plot(fish.ward, main = "Doubs River Fish: Ward's Clustering",
     ylab = "Squared Bray-Curtis Distance")
```
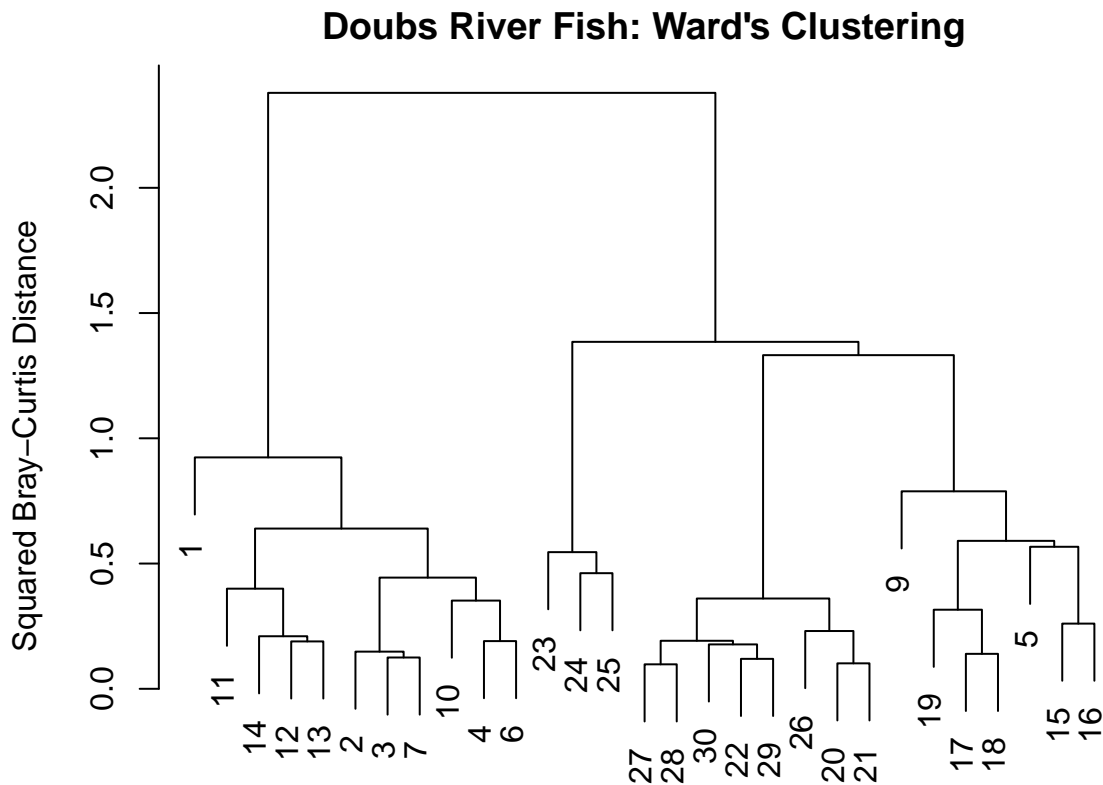


```
# Plot cluster analysis using headmap.2 #
gplots::heatmap.2(as.matrix(fish), distfun = function(x) vegdist(x, method = "bray"),
                  hclustfun = function(x) hclust(x, method = "ward.D2"),
                  col = viridis, trace = "none", density.info = "none")
```
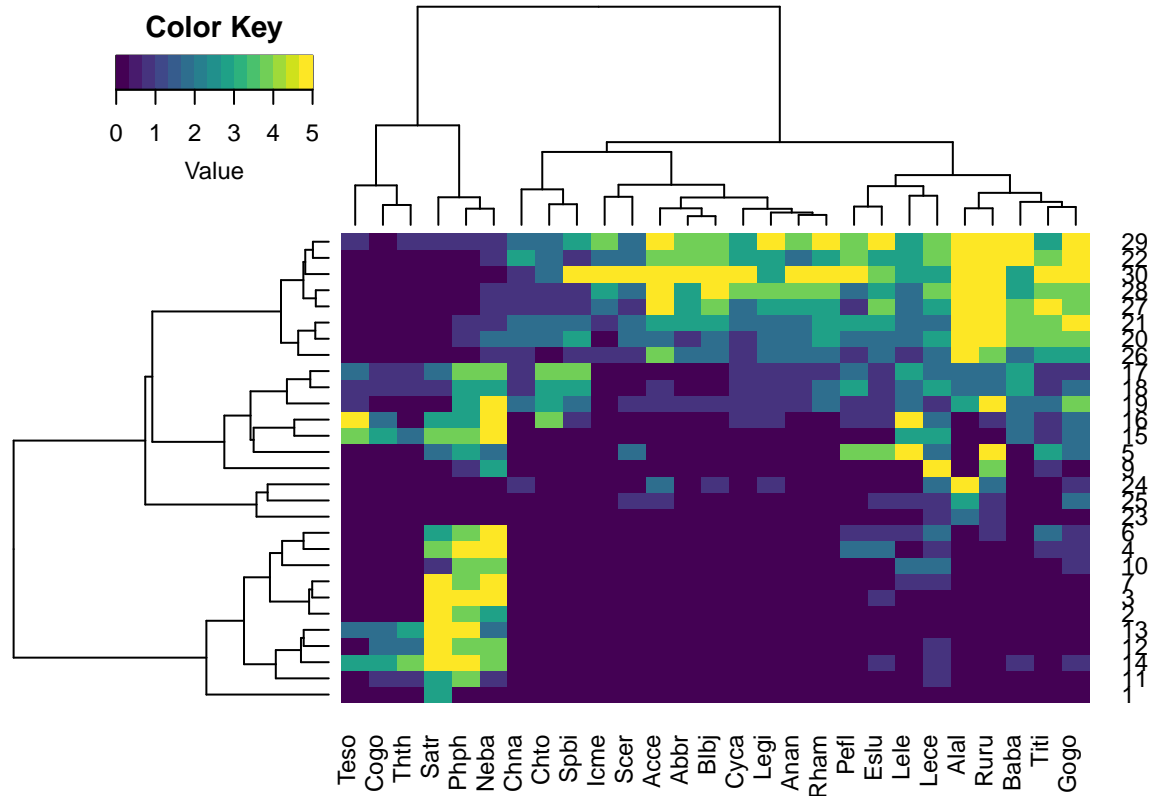
Question 6: Based on cluster analyses and the introductory plots that we generated after loading the data, develop an ecological hypothesis for fish diversity the `doubs` data set?

> **Answer 6**: I predict that since some sites are more dissimilar than others, the differences may be due to enviromental quality differences. If so, then the presences and abundances of certain fish species in some sites but not other sites is indicative of enviromental senstitivity of fish species.

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.

```
# Principal Coordinate Analysis #
fish.pcoa <- cmdscale(fish.db, eig = TRUE, k = 3)
# Interpreting PCoA output #
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)
```

```
explainvar1
```

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

```
explainvar2
```

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

```
explainvar3
```

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

```
sum.eig
```

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

```r
# Plot the PCoA #
# Define Plot Parameters
par(mar = c(5, 5, 1, 2) + 0.1)
# Initiate Plot
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 = FALSE)

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

# Add Points & Labels
points(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))

# Label sites #
# First we calculate the relative abundances of each species at each site
fishREL <- fish
  for(i in 1:nrow(fish)){
    fishREL[i, ] = fish[i, ] / sum(fish[i, ])
  }

# Now, we use this information to calculate and add species scores
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")
```
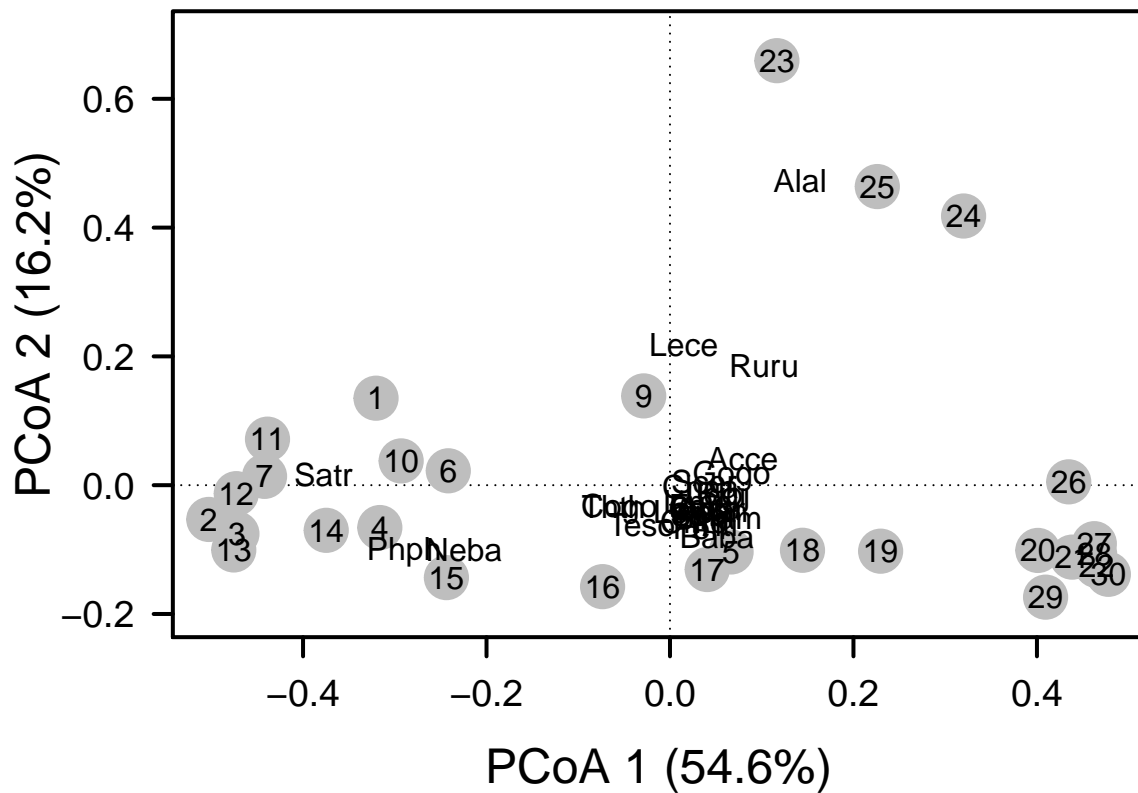
In the R code chunk below, do the following:

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.

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

```
##              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 for Species Abundances Across Axes
fit <- envfit(fish.pcoa, fishREL, perm = 999)
fit
```

```
##
```

```
## ***VECTORS
##
##          Dim1     Dim2     r2 Pr(>r)
## Cogo -0.83884 -0.54438 0.2982  0.010 **
## 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  0.029 *
## Teso -0.44704 -0.89452 0.1700  0.083 .
## Chna  0.99707 -0.07644 0.4612  0.001 ***
## Chto  0.42032 -0.90738 0.2579  0.028 *
## Lele  0.33041 -0.94384 0.0495  0.517
## Lece  0.06856  0.99765 0.3399  0.008 **
## Baba  0.54118 -0.84091 0.6752  0.001 ***
## Spbi  0.57341 -0.81927 0.4138  0.004 **
## Gogo  0.97507  0.22188 0.3753  0.003 **
## Eslu  0.72044 -0.69352 0.1673  0.082 .
## Pefl  0.43762 -0.89916 0.3048  0.012 *
## 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  0.005 **
## 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: It seems that certain communities are grouped by the relative abundance of one or a couple of fish species. For example, sites 23, 24, and 25 are grouped together primarily based on influential species "Alal". *Answer 7b*: Satr, Php, Neba, Alal are probably the indicators of river quality because these fish species are common in the sampling sites that are driving the variation seen in the PCoA.

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.

1. 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.

```r
# Create factor vector in Doubs River #
quality <- c(rep("HQ", 13), rep("MQ", 5), rep("LQ", 6), rep("MQ", 5))

# Run PERMANOVA with adonis function #
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
```

```r
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   0.016 *
##
##  Group HQ+MQ  #sps.  2
##       stat p.value
## Satr 0.860   0.003 **
## Phph 0.859   0.014 *
##
##  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    0.001 ***
## Acce 0.866    0.001 ***
## Lele 0.863    0.003 **
## Titi 0.853    0.004 **
## Chto 0.829    0.001 ***
## Rham 0.829    0.001 ***
## Anan 0.829    0.001 ***
## Eslu 0.827    0.019 *
## Pefl 0.806    0.015 *
## Blbj 0.791    0.003 **
## Scer 0.766    0.010 **
## Abbr 0.750    0.009 **
## Icme 0.661    0.025 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
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   0.002 **
## Satr 0.650   0.001 ***
##
##  Group LQ  #sps.  2
##       stat p.value
## Alal 0.693   0.001 ***
## Ruru 0.473   0.033 *
##
##  Group MQ  #sps.  4
##       stat p.value
## Anan 0.571   0.006 **
## Spbi 0.557   0.013 *
```

```
## Chto 0.542   0.015 *
## Icme 0.475   0.036 *
##
##  Group LQ+MQ  #sps.  9
##       stat p.value
## Legi 0.658   0.003 **
## Baba 0.645   0.002 **
## Rham 0.600   0.005 **
## Acce 0.594   0.007 **
## Cyca 0.586   0.006 **
## Chna 0.571   0.015 *
## Blbj 0.571   0.008 **
## Gogo 0.523   0.013 *
## Abbr 0.499   0.028 *
## ---
## 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*: From the PERMANOVA, it is clear that habitat quaility affects fish species composi-ton. The IndVal analyses shows that certain fish species presences may be indicative of stream quality, for example, Alal seems to have a significant IndVal score (close to 1) in low and medium quality stream sample sites. Alternatively, phi coeffecent anaylsis shows that groups of fish species display strong preference for certain stream quality types. For example, Phph, Meba, and Satr fish species perfer High quality river sites.

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.

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

#Mantel Test
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%
## 0.102 0.137 0.176 0.204
```

19

```
## 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 community assemblage (based from Bray-Curtis dissimilarity index) is highly correlated with enviromental conditions. This supports the hypothesis that stream quality influences fish communities.

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.

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

# Perform dbRDA
doubs.dbrda <- dbrda(fish.db ~ ., as.data.frame(env.chem))
ordiplot(doubs.dbrda)
```
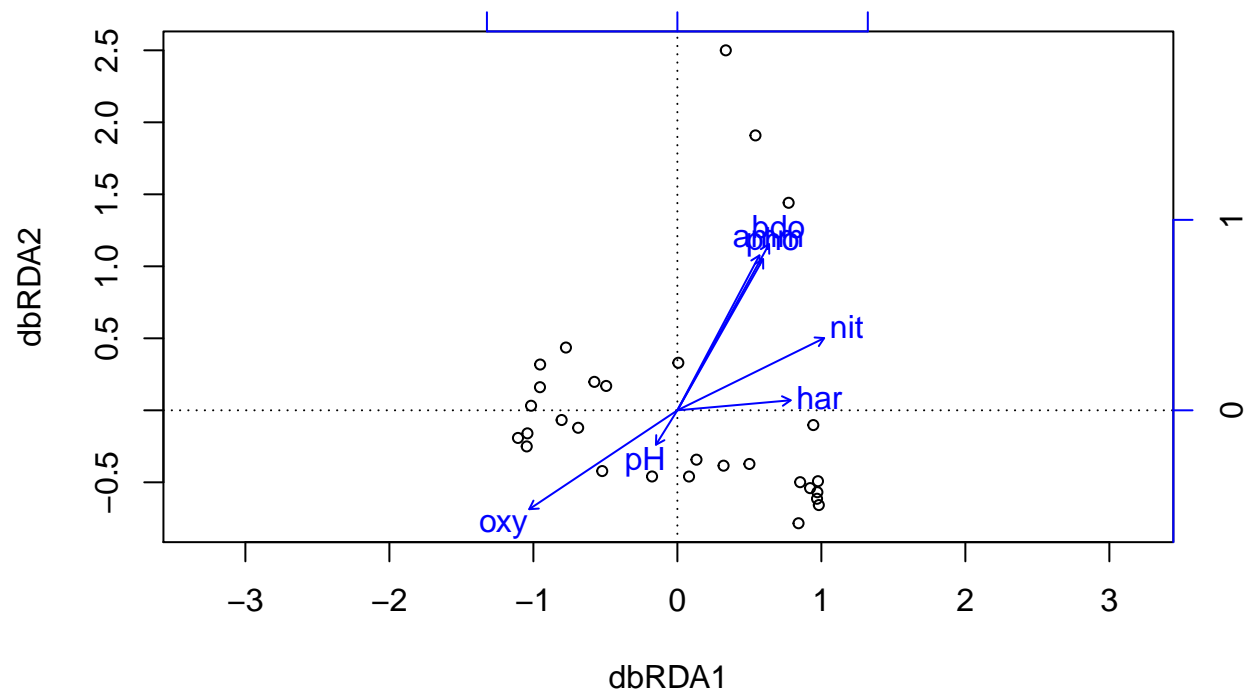
```
###############################################################
# First, we will model only the intercept
doubs.dbrda.mod0 <- dbrda(fish.db ~ 1, as.data.frame(env.chem))

# Note there are no vectors here (we didn't constrain anything)
# Therefore, the axes suggest this is a simple MDS (i.e., PCoA)
ordiplot(doubs.dbrda.mod0)
```
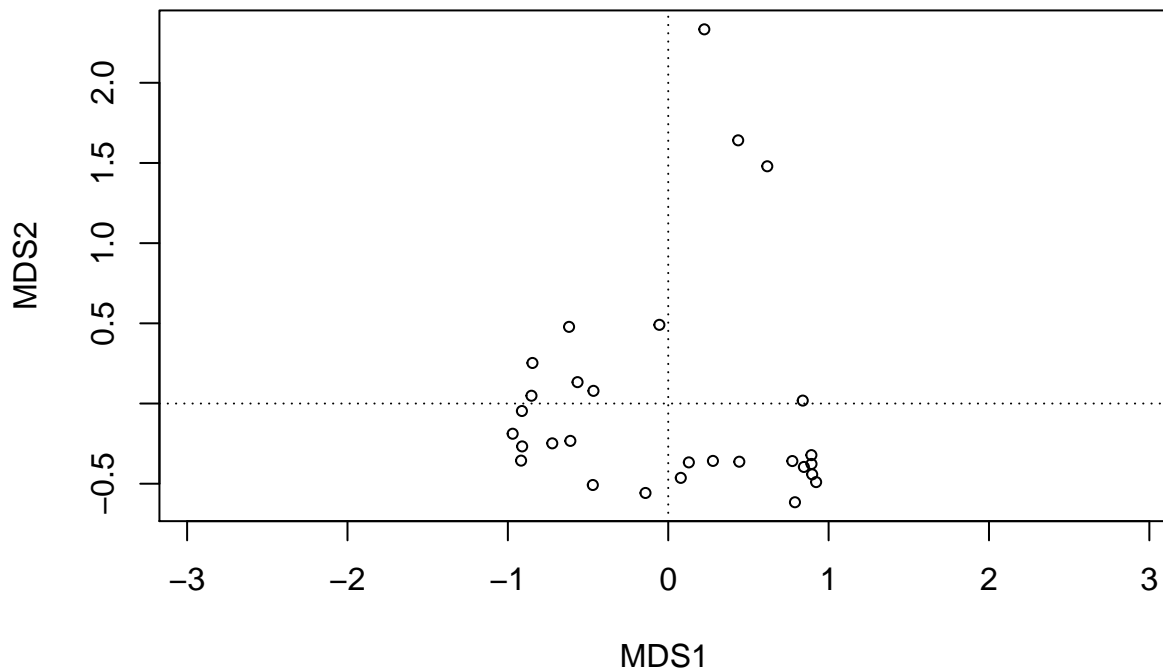
```
# Next, we will model the full model, with all explanatory variables
doubs.dbrda.mod1 <- dbrda(fish.db ~ ., as.data.frame(env.chem))

# Now we step through all combinations of explanatory variables in our model
# The function returns the model with the lowest AIC value
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   0.002 **
## ---
## 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
```

```r
# Lets look at the model that was selected
doubs.dbrda$call
```
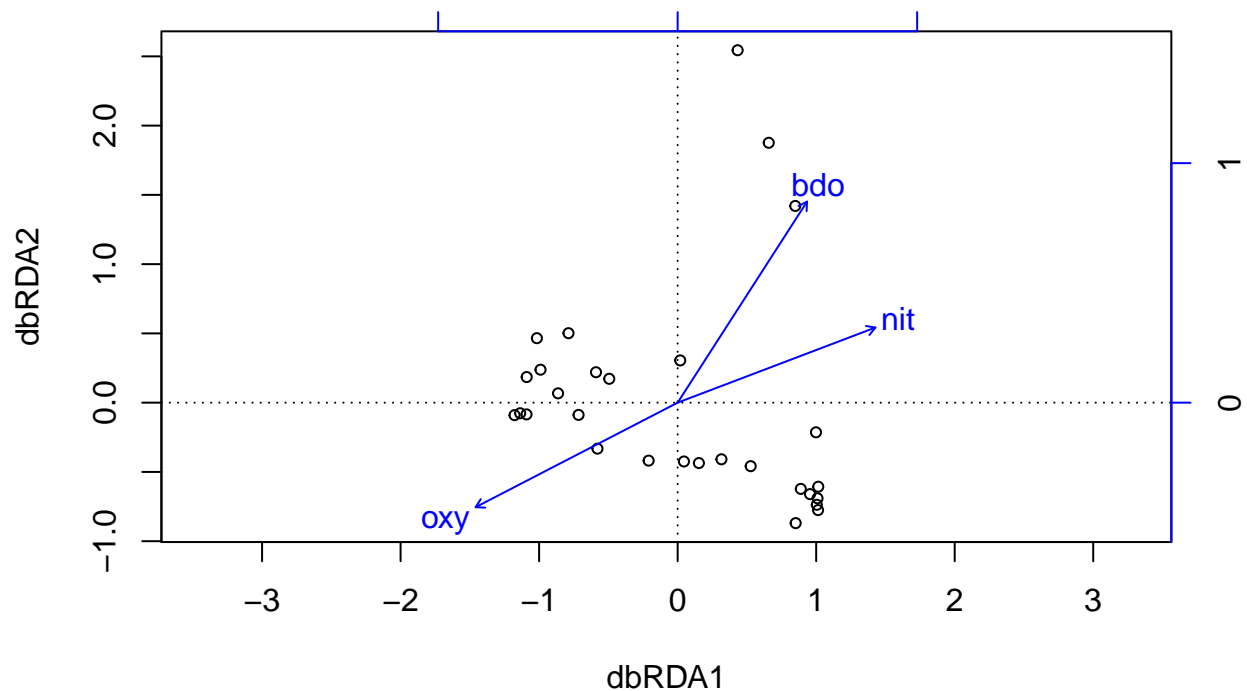
```
## dbrda(formula = fish.db ~ oxy + bdo + nit, data = as.data.frame(env.chem))
```

```
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  0.002 **
## + 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)
```



```
# Permutation tests to evaluate significance
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
```

```r
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
```

```r
# Calculate Explained Variation
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

# Define Plot Parameters
par(mar = c(5, 5, 4, 4) + 0.1)

# Initiate Plot
plot(scores(doubs.dbrda, display = "wa"), xlim = c(-1.3, 1.1), ylim = c(-1.1, 2.7),
     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)

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

# Add Points & Labels
points(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")))

# Add Environmental Vectors
vectors <- scores(doubs.dbrda, display = "bp")

#row.names(vectors) <- rownames(vectors)
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])))
```
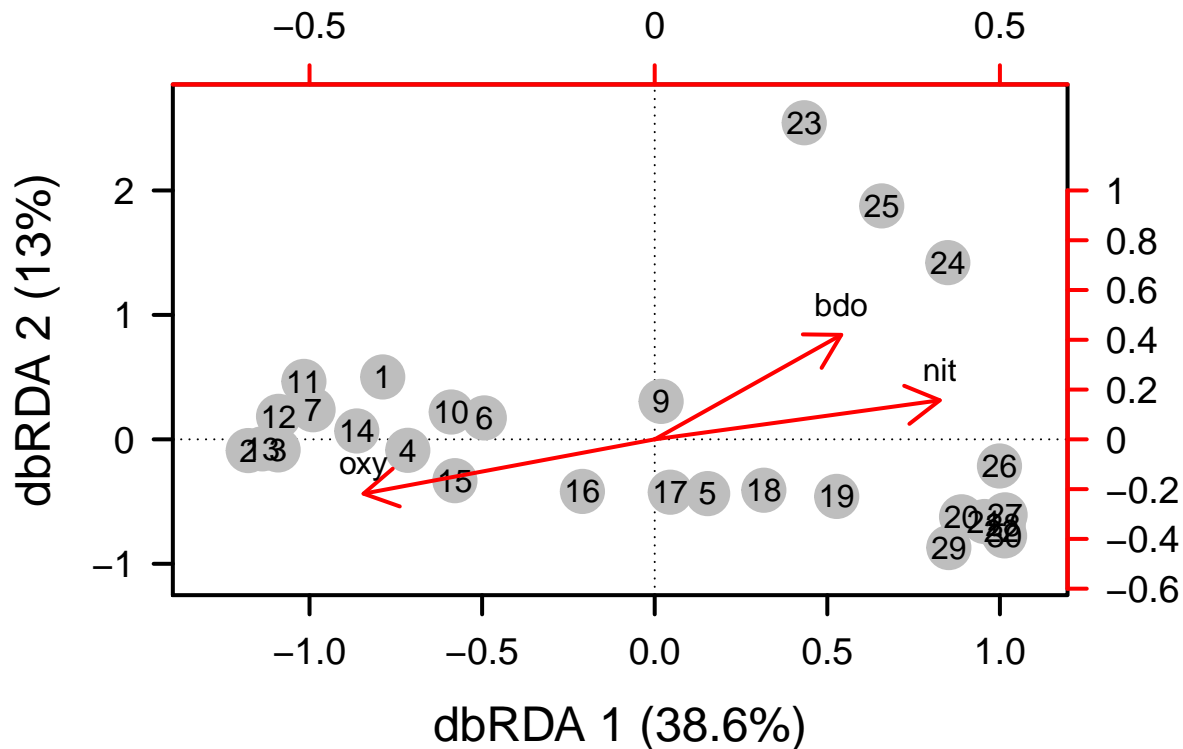
Question 10: Based on the constrained ordination, what are the environmental variables (or groups of correlated variables) that seem to be contributing to variation in fish community structure?

> **Answer 10**: Nitrogen, oxygen, and bdo envriomental variables were driving the variations 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.

```
# Remember, our environmental model uses oxy, bdo, and nit and has R2 of 0.53
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  0.002 **
## + 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
# Let's create a matrix model for our environmental data
env.mod <- model.matrix(~ oxy + bdo + nit, as.data.frame(env.chem))[,-1]

# First, we will weight each site by its relative abundance
rs <- rowSums(fish)/sum(fish)

# Next, we will perform PCNM
doubs.pcnmw <- pcnm(dist(doubs$xy[-8,]), w = rs, dist.ret = T)

# PCNM can return negative eigenvalues, but only the
# eigenvectors associated with the positive eigenvalues are meaningful
doubs.pcnmw$values > 0
```

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

```
# Removing repetitive
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   0.008 **
## ---
## 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   0.008 **
## ---
## 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  0.012 *
## ---
## 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  0.012 *
## ---
## 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   0.04 *
## ---
## 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   0.07 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Because this is another dbRDA, we could visualize the biplot
# showing how each vector explains variation across sites
plot(step.pcnm)
```
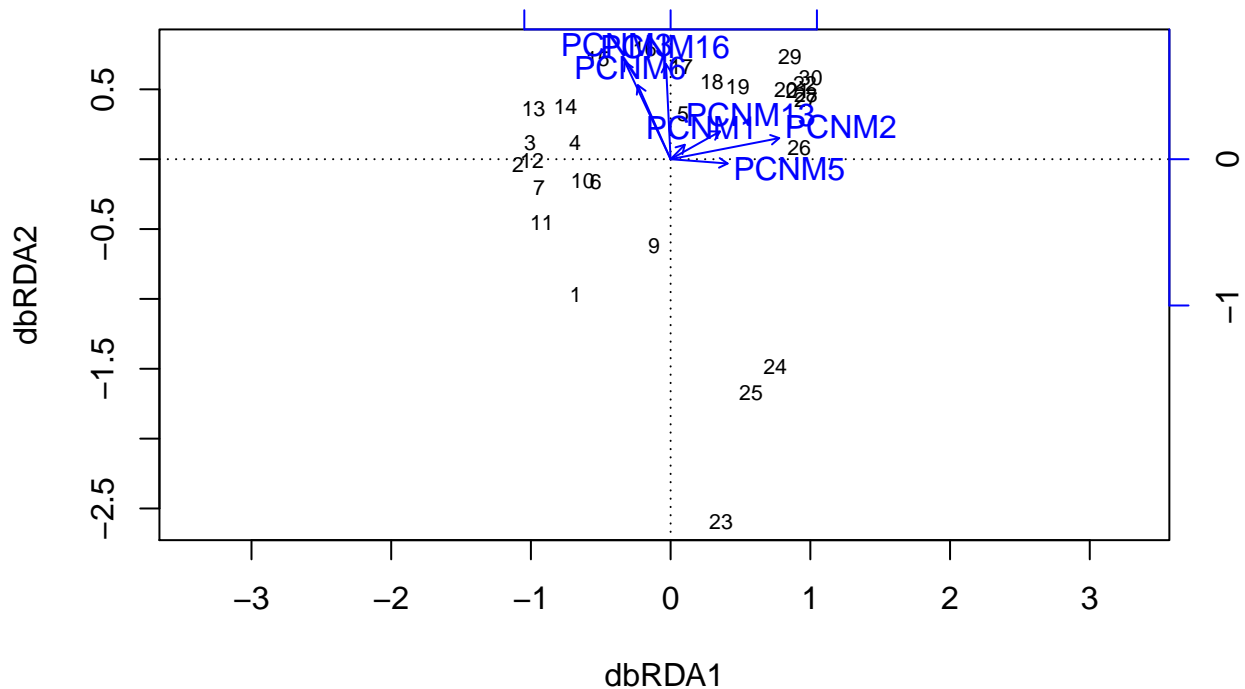
```
# The object `step.pcnm` now contains the selected model.
step.pcnm$anova
```

```
##                     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  0.008 **
## + PCNM1           0.47215  1 41.411 4.0570  0.008 **
## + PCNM13          0.52124  1 39.346 3.4612  0.012 *
## + PCNM16          0.57680  1 36.480 4.0192  0.012 *
## + PCNM6           0.60431  1 35.182 2.5296  0.040 *
## <All variables> 0.62601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# We can now construct a spatial model using only the selected PCNM axes.
space.mod <- model.matrix(~ PCNM2 + PCNM3 + PCNM5 + PCNM1 +
PCNM13 + PCNM16 + PCNM6, doubs.space)[,-1]

# First conduct constrained ordinations
doubs.total.env <- dbrda(fish.db ~ env.mod)
doubs.total.space <- dbrda(fish.db ~ space.mod)
# Next construct partial constrained ordinations
doubs.env.cond.space <- dbrda(fish.db ~ env.mod + Condition(space.mod))
doubs.space.cond.env <- dbrda(fish.db ~ space.mod + Condition(env.mod))
# Next test for significance of the dbRDA fractions.
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
```

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

```
## 
## Permutation test for dbrda
## 
## Permutation: free
## Number of permutations: 999
## 
## Call: dbrda(formula = fish.db ~ space.mod + Condition(env.mod))
## Permutation test for all constrained eigenvalues
## Pseudo-F:     4.174109 (with 7, 18 Degrees of Freedom)
## Significance:     0.001
```

```r
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
```

```r
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
```

```r
# Using the built-in varpart() function
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
```

```r
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)
```

Variation Partitioning of Doubs Fish Diversity

X1   X2

Env   Space

0.13   0.37   0.24

Residuals = 0.27
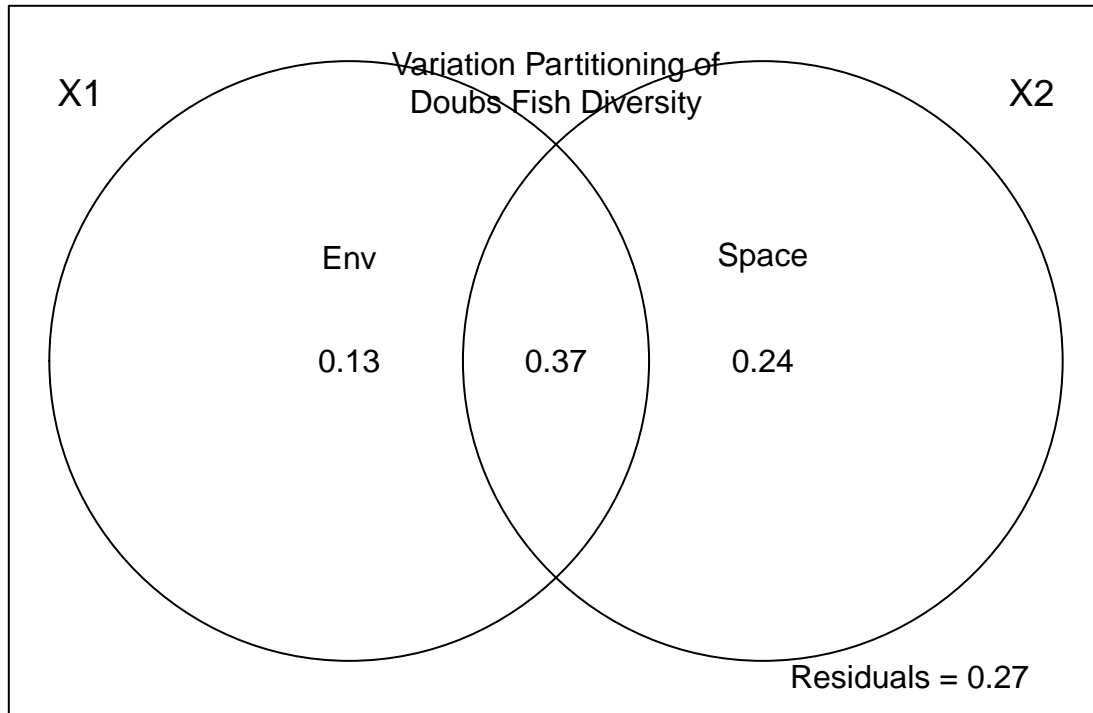
Question 11: Interpret the variation partitioning results.

> *Answer 11*: From the variance paritioning of the Doubs fish diversty dataset, it seems that fish communities are more driven by space than by enviromental factors, but together explain 0.37 of the vartiation in the dataset which is relatively high.

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.

```
rm(list = ls())
setwd("C:/Users/Venus/Github/QB2017BenavidezKuo/data")

# Load packages #
require("vegan")  #Already contains BCI data
require("tidyr")
```

```
## Loading required package: tidyr
```

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Load dataset #
BCI2010 <- read.delim("BCI2010.txt", header=T)
BCI1982 <- read.delim("BCI1982.txt", header=T)

# Transform 2010 census data into site by species matrix #
BCI.2010.SbyS <- group_by(BCI2010, Quadrat) %>% count(Latin) %>% spread(key=Latin, value=n , fill=0)
dim(BCI.2010.SbyS)
```

```
## [1] 1250  298
```

```r
# Transform 1982 census data #
BCI.1982.SbyS <- group_by(BCI1982, Quadrat) %>% count(Latin) %>% spread(key=Latin, value=n , fill=0)

# Ordination: PCoA #
BCI.2010.db=vegdist(BCI.2010.SbyS,method="bray")

BCI.2010.pcoa=cmdscale(BCI.2010.db,eig=T,k=3)

explainvar1 <- round(BCI.2010.pcoa$eig[1] / sum(BCI.2010.pcoa$eig), 3) * 100
explainvar2 <- round(BCI.2010.pcoa$eig[2] / sum(BCI.2010.pcoa$eig), 3) * 100
explainvar3 <- round(BCI.2010.pcoa$eig[3] / sum(BCI.2010.pcoa$eig), 3) * 100

sum.eig <- sum(explainvar1, explainvar2, explainvar3)

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

plot(BCI.2010.pcoa$points[ ,1], BCI.2010.pcoa$points[ ,2], ylim = c(-0.2, 0.3),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)

axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)

abline(h = 0, v = 0, lty = 3)

box(lwd = 2)

points(BCI.2010.pcoa$points[ ,1], BCI.2010.pcoa$points[ ,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")

text(BCI.2010.pcoa$points[ ,1], BCI.2010.pcoa$points[ ,2],
     labels = row.names(BCI.2010.pcoa$points))
```
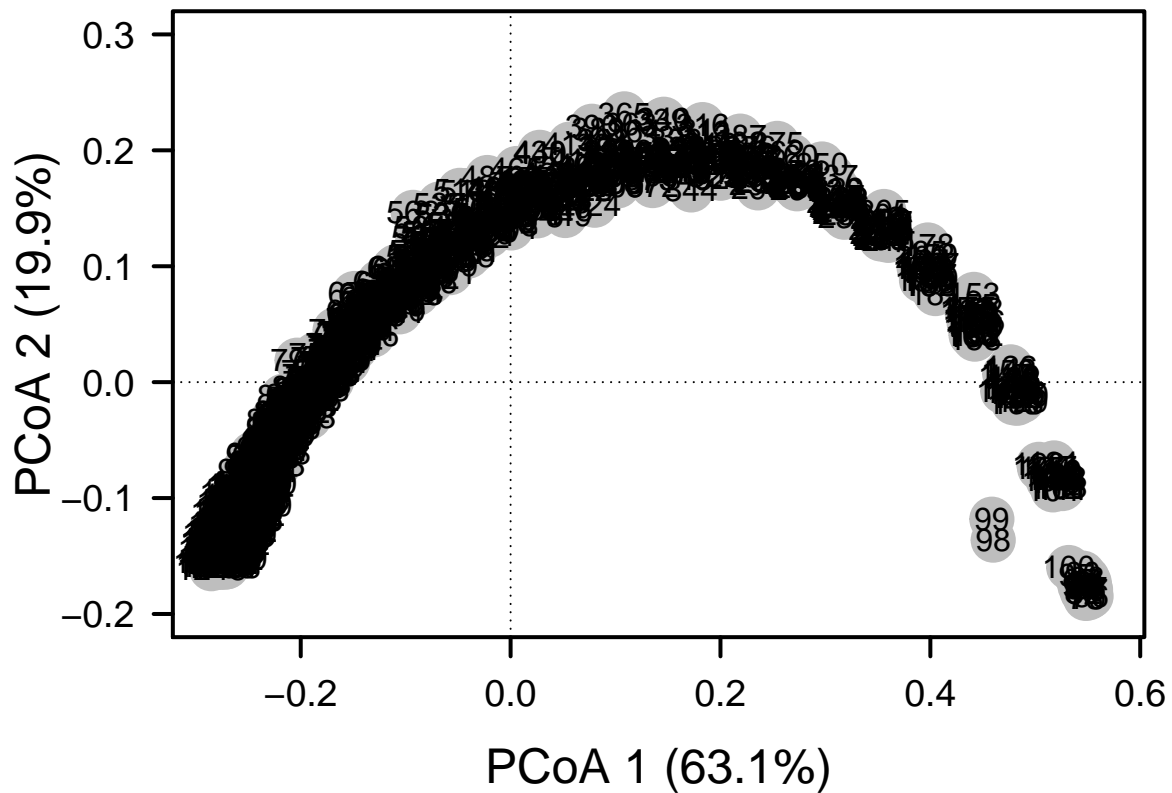
```
BCI.1982.db=vegdist(BCI.1982.SbyS,method="bray")

BCI.1982.pcoa=cmdscale(BCI.1982.db,eig=T,k=3)

explainvar1 <- round(BCI.1982.pcoa$eig[1] / sum(BCI.1982.pcoa$eig), 3) * 100
explainvar2 <- round(BCI.1982.pcoa$eig[2] / sum(BCI.1982.pcoa$eig), 3) * 100
explainvar3 <- round(BCI.1982.pcoa$eig[3] / sum(BCI.1982.pcoa$eig), 3) * 100

sum.eig <- sum(explainvar1, explainvar2, explainvar3)

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

plot(BCI.1982.pcoa$points[ ,1], BCI.1982.pcoa$points[ ,2], ylim = c(-0.2, 0.3),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)

axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)

abline(h = 0, v = 0, lty = 3)

box(lwd = 2)

points(BCI.1982.pcoa$points[ ,1], BCI.1982.pcoa$points[ ,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
```
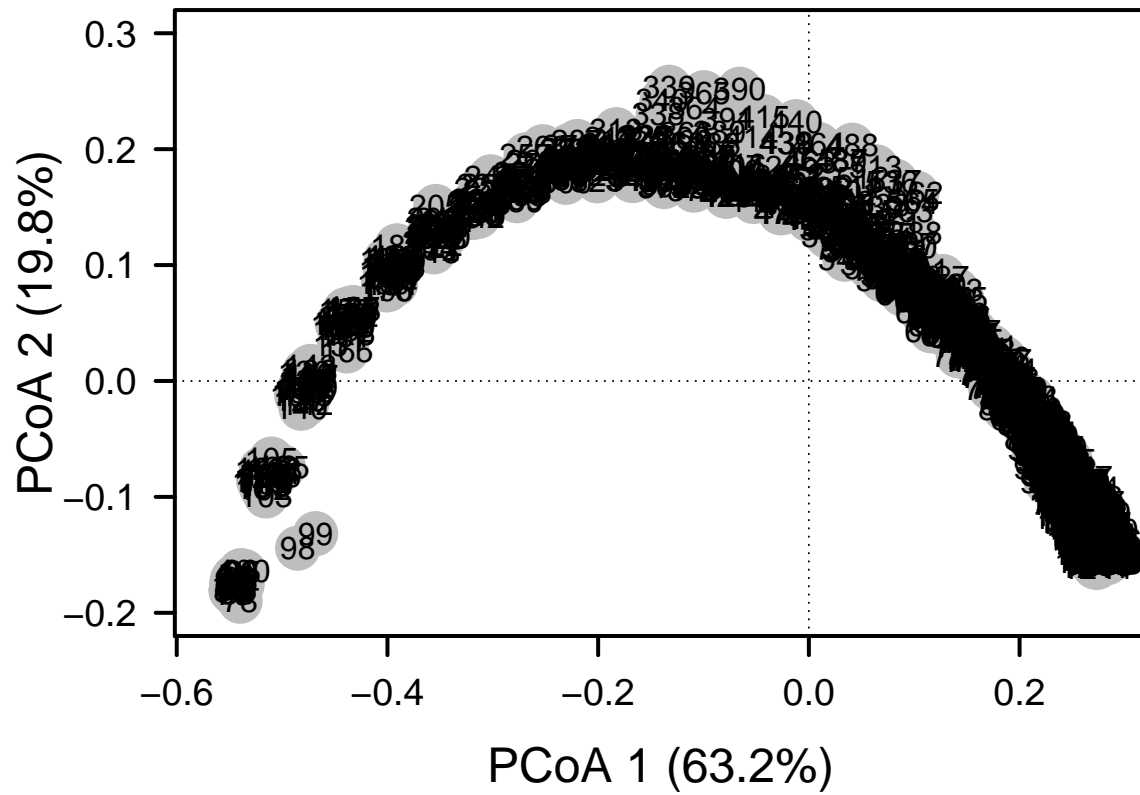
```
text(BCI.1982.pcoa$points[ ,1], BCI.1982.pcoa$points[ ,2],
     labels = row.names(BCI.1982.pcoa$points))
```



```
# The two PCoAs look nearly identical. We are rethinking our project dataset to incorporate 50 ha plots
```