Lab 4 Ordination Part II

## Objective

Today we will work through examples of Principal Coordinates Analysis (PCoA) and Non-Metric Multidimensional Scaling (NMDS).

## Data

Call in the data set “Current\_Hawaiian\_Birds.csv” from your working directory and name it *birds*. This data set consists of presence/absence data for bird species on the 6 main Hawaiian Islands in the current time period. You will be using this data set for the **Principal Coordinates Analysis (PCoA)**.

birds <- read.csv("G:/Shared drives/MultivariateStatistics/Data/LabData/Lab4/Current\_Hawaiian\_Birds.csv",  
 row = 1, header = TRUE)

Next, call in the data set “combined\_birds.csv” and call it *birds2*. This data set consists of presence/absence data for bird species on the 6 main Hawaiian Islands in the current time period and the historical time period (i.e. before colonization by Europeans). You will be using this data set for the **Non-Metric Multidimensional Analysis (NMDS)**.

birds2 <- read.csv("G:/Shared drives/MultivariateStatistics/Data/LabData/Lab4/combined\_birds.csv",  
 row = 1, header = TRUE)

## Download packages

You will be using the packages vegan and ca. If you do not already have them installed, remember that you will first need to install them using the ‘install.packages’ function.

library(vegan)  
library(ca)

# Principal Coordinates Analysis (PCoA)

PCoA is a flexible analysis that is performed on a variety of distance matrices (e.g. Euclidean, Jaccard index, Sorensen index). birds is a binary data set so lets use the *Sørensens’s index*. *Note that there are many possible indices to use for binary data; see the optional Koleff et al. 2003 reading listed in our readings on Brightspace.*

Please note that the Bray–Curtis dissimilarity is directly related to the quantitative Sørensen similarity index when using binary data. That is why the argument method in the function ‘vegdist’ below is set to “bray”:

jbirds <- vegdist(birds, "bray")

## Question 1: Which island pair is the most similar? (15 pts)

You are going to use the cmdscale function in the *stats* package to run the PCoA:

`?`(wcmdscale)

## starting httpd help server ... done

wcmd <- wcmdscale(jbirds, k = 5, eig = TRUE)

The “points” are the coordinates of each island. They are the *eigenvectors* scaled by the square root of their *eigenvalues* (i.e. the standard deviation):

wcmd$points

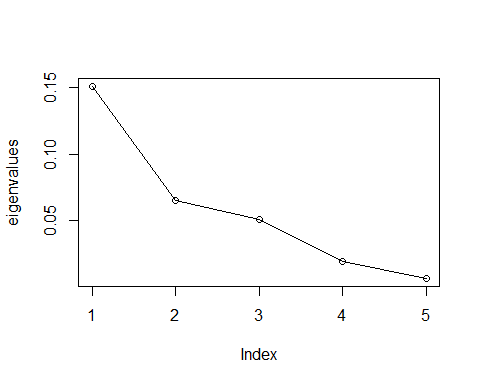
Let’s make a PCoA table to look at the eigenvalues, and the proportional and cumulative variance:

eigenvalues <- wcmd$eig[1:5]  
propVar <- eigenvalues/sum(eigenvalues)  
cumVar <- cumsum(propVar)  
PCoA\_Table <- cbind(eigenvalues, propVar, cumVar)  
PCoA\_Table

## eigenvalues propVar cumVar  
## [1,] 0.150923062 0.51612551 0.5161255  
## [2,] 0.065338058 0.22344258 0.7395681  
## [3,] 0.050880324 0.17400013 0.9135682  
## [4,] 0.018859552 0.06449575 0.9780640  
## [5,] 0.006414435 0.02193603 1.0000000

**Scree plot**:

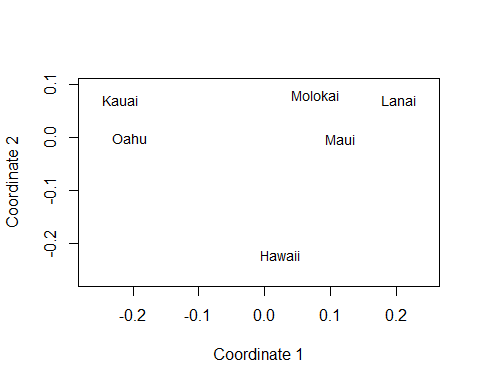
plot(eigenvalues)  
lines(lowess(eigenvalues))



## Question 2: How many axes should you keep? (15 pts)

Now, let’s plot the first two PCoA axes:

x <- wcmd$points[, 1]  
y <- wcmd$points[, 2]  
plot(x, y, xlab = "Coordinate 1", ylab = "Coordinate 2", xlim = range(x) \* 1.2, ylim = range(y) \*  
 1.2, type = "n")  
text(x, y, labels = rownames(wcmd$points), cex = 0.9)

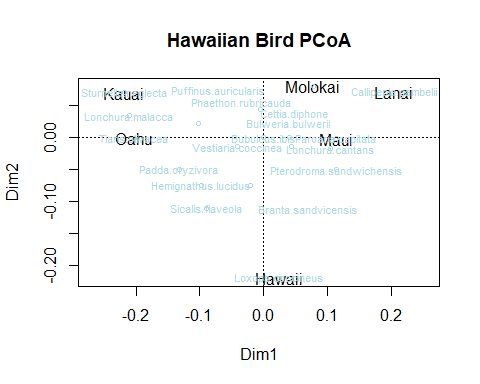


Another way to plot:

`?`(orditorp)  
  
ordiplot(scores(wcmd)[, c(1, 2)], type = "t", cex = 1, main = "Hawaiian Bird PCoA")

## species scores not available

abline(h = 0, lty = 3)  
abline(v = 0, lty = 3)  
  
# Add species by computing their weighted average ordination scores  
`?`(wascores)  
  
species <- wascores(wcmd$points[, 1:2], birds)  
# text(species,rownames(species),cex=.7, col='red')  
orditorp(species, display = "species", labels = rownames(species), col = "lightblue")



## Question 3: What species is most associated with the island of Hawaii? (15 pts)

## By “hand” for those who are interested in it. Or skip ahead to the NMDS section.

Ok, let’s now run a PCoA following the directions that I gave during lecture.

jbirds <- vegdist(birds, "bray")  
CORD <- -1/2 \* jbirds^2  
C <- as.matrix(CORD)  
cs <- colMeans(C)  
rs <- rowMeans(C)  
C1 <- sweep(C, MARGIN = 2, cs, FUN = "-")  
C2 <- sweep(C1, MARGIN = 1, rs, FUN = "-")  
delta <- mean(C) + C2  
  
# Next, run an eigen analysis:  
EG <- eigen(delta)  
eigenvalues2 <- EG$values[1:5]  
  
# And make our PCoA table:  
propVar2 <- eigenvalues2/sum(eigenvalues2)  
cumVar2 <- cumsum(propVar2)  
PCoA\_Table2 <- cbind(eigenvalues2, propVar2, cumVar2)  
PCoA\_Table2

## eigenvalues2 propVar2 cumVar2  
## [1,] 0.150923062 0.51612551 0.5161255  
## [2,] 0.065338058 0.22344258 0.7395681  
## [3,] 0.050880324 0.17400013 0.9135682  
## [4,] 0.018859552 0.06449575 0.9780640  
## [5,] 0.006414435 0.02193603 1.0000000

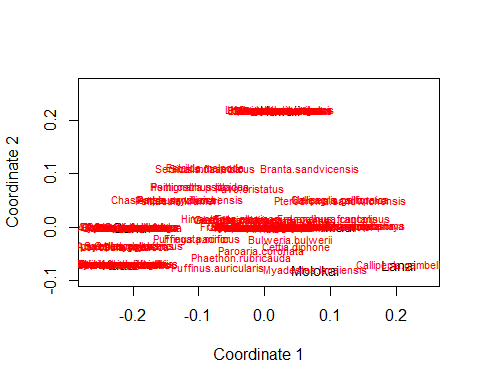
# You scale the eigenvectors by the square root of their eigenvalues to get the  
# coordinates (points):  
points2 <- sweep(EG$vectors[, 1:5], MARGIN = 2, sqrt(eigenvalues2), FUN = "\*")  
points2

## [,1] [,2] [,3] [,4] [,5]  
## [1,] -0.21870425 -0.0703630384 -0.13447147 -0.01417970 0.021871617  
## [2,] -0.20331071 0.0004064133 0.16477058 0.02159888 -0.002679892  
## [3,] 0.07763682 -0.0806737197 -0.03608826 0.02499324 -0.063824813  
## [4,] 0.20464754 -0.0716258058 0.01841217 0.05912239 0.042779988  
## [5,] 0.11506406 0.0014705806 0.03800194 -0.11606159 0.004349585  
## [6,] 0.02466655 0.2207855700 -0.05062495 0.02452678 -0.002496485

x <- points2[, 1]  
y <- points2[, 2]

**Lets plot this**

# The coordinates:  
plot(x, y, xlab = "Coordinate 1", ylab = "Coordinate 2", xlim = range(x) \* 1.2, ylim = range(y) \*  
 1.2, type = "n")  
text(x, y, labels = rownames(birds), cex = 0.9)  
  
  
# Calculate weighted species scores:  
scores1 <- sweep(birds, MARGIN = 1, x, FUN = "\*")  
species1 <- colSums(scores1)/colSums(birds)  
scores2 <- sweep(birds, MARGIN = 1, y, FUN = "\*")  
species2 <- colSums(scores2)/colSums(birds)  
  
# Add to the plot. Note that using vegan enables a lot nicer plotting than  
# these hand calculations.  
  
text(cbind(species1, species2), colnames(birds), cex = 0.7, col = "red")



# Non-Metric Multidimensional Analysis (NMDS)

**NMDS** is the most flexible ordination technique. It operates on a distance matrix and projects samples that are similar, close together and ones that are different, far apart.

Create Sørensens’s disimilarity martix for the birds data:

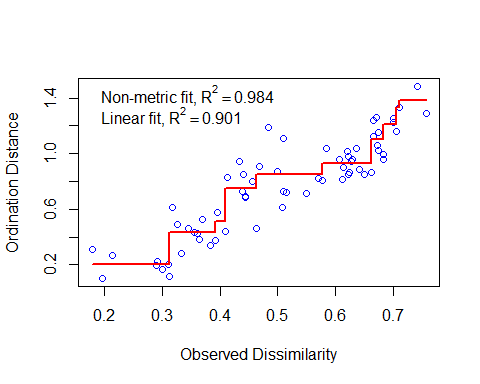
jbirds2 <- vegdist(birds2, "bray")

You are going to use the metaMDS function in the vegan package. K = 2 because we are interested in only two dimensions (which is common for NMDS).

`?`(metaMDS)  
  
nmdsBird <- metaMDS(jbirds2, k = 2, trace = T)

## Run 0 stress 0.1263741   
## Run 1 stress 0.1709141   
## Run 2 stress 0.1709141   
## Run 3 stress 0.1263741   
## ... New best solution  
## ... Procrustes: rmse 1.093409e-06 max resid 1.828413e-06   
## ... Similar to previous best  
## Run 4 stress 0.1263741   
## ... New best solution  
## ... Procrustes: rmse 1.224958e-06 max resid 2.263079e-06   
## ... Similar to previous best  
## Run 5 stress 0.1683714   
## Run 6 stress 0.1263741   
## ... Procrustes: rmse 1.626606e-06 max resid 3.621868e-06   
## ... Similar to previous best  
## Run 7 stress 0.1263741   
## ... New best solution  
## ... Procrustes: rmse 1.147623e-06 max resid 2.413406e-06   
## ... Similar to previous best  
## Run 8 stress 0.1263741   
## ... New best solution  
## ... Procrustes: rmse 6.870173e-07 max resid 1.758756e-06   
## ... Similar to previous best  
## Run 9 stress 0.153   
## Run 10 stress 0.153   
## Run 11 stress 0.3169867   
## Run 12 stress 0.1263741   
## ... Procrustes: rmse 1.174879e-06 max resid 2.739473e-06   
## ... Similar to previous best  
## Run 13 stress 0.1263741   
## ... Procrustes: rmse 1.038058e-06 max resid 2.461632e-06   
## ... Similar to previous best  
## Run 14 stress 0.1263741   
## ... Procrustes: rmse 7.531477e-07 max resid 1.674562e-06   
## ... Similar to previous best  
## Run 15 stress 0.1464151   
## Run 16 stress 0.1263741   
## ... Procrustes: rmse 1.141861e-06 max resid 1.882547e-06   
## ... Similar to previous best  
## Run 17 stress 0.166159   
## Run 18 stress 0.1291746   
## Run 19 stress 0.1263741   
## ... Procrustes: rmse 2.869786e-06 max resid 6.595308e-06   
## ... Similar to previous best  
## Run 20 stress 0.128085   
## \*\*\* Best solution repeated 6 times

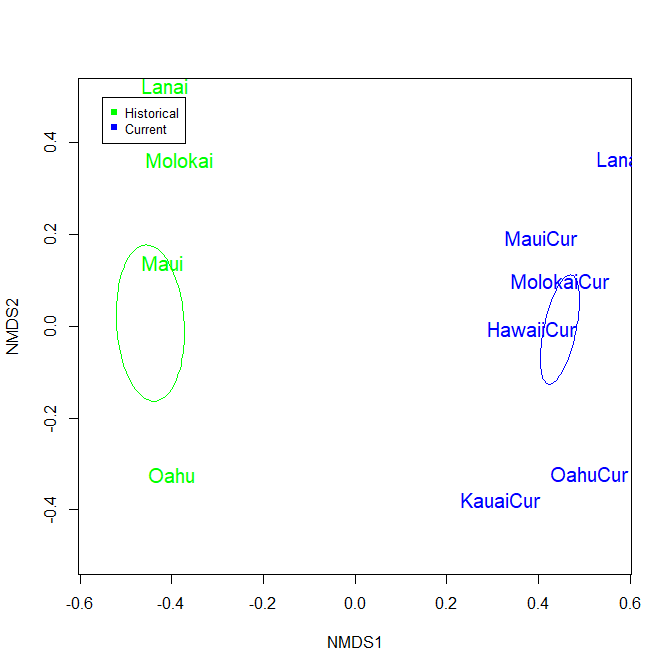
stressplot(nmdsBird)



## Question 4: What do the stress value and the fit (R2) of the monotonic regression tell you about the NMDS plot? (15 pts)

Let’s plot out our results and see if there is a difference between the historical and current Hawaiian bird assemblages?

# Identify the time period as groups:  
  
treat = as.matrix(c(rep("Historical", 6), rep("Current", 6)))  
  
# Plot out the points (islands):  
  
ordiplot(nmdsBird, type = "n", xlim = c(-0.5, 0.5), ylim = c(-0.5, 0.5))  
orditorp(nmdsBird, display = "sites", col = c(rep("green", 6), rep("blue", 6)), air = 0.01,  
 cex = 1.25)  
legend(-0.55, 0.5, c("Historical", "Current"), cex = 0.8, col = c("green", "blue"),  
 pch = 15:15)  
  
# Add an ellipse representing the standard error around each group:  
  
ordiellipse(nmdsBird, treat, display = "si", lty = 1, col = "green", show.groups = "Historical",  
 kind = "se")  
ordiellipse(nmdsBird, treat, display = "si", lty = 1, col = "blue", show.groups = "Current",  
 kind = "se")



## Question 5: Are the historic and current bird distributions on the islands statistically different? Why? (15 pts)

## Question 6: Does your personal data set for class meet the assumptions of either PCoA or NMDS? (25 pts)

**If you are done early, please go ahead and get started on Project 1.**