Lab 4 Ordination Part II

# Set up R session

**You are pros at this by now…**

## 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("Data/lab\_4/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("Data/lab\_4/combined\_birds.csv", row = 1, header = TRUE)

Finally, call in the data set “tree.csv” and call it *tree*. This is the tree reproduction data set from lecture that you will use for the 4. Correspondence Analysis (CA).

tree <- read.csv("Data/lab\_4/tree.csv", row = 1, header = TRUE)

## Download packages

you will be using the packages vegan and ca

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 Koleff et al. 2003 reading from module 2.*

Sørensens’s index \*\* Bray-Curtis distance is = to Sørensens’s when using binary data. That is why method = “bray” below\*\*:

jbirds <- vegdist(birds, "bray")

Which island pair is the most similar?

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

`?`(cmdscale)

## starting httpd help server ... done

cmd <- cmdscale(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):

cmd$points

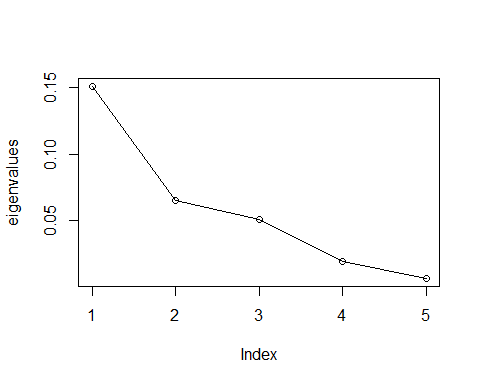
Let’s make a PCoA table to look at the eigenvalues, and the proportional and cumulative variance:

eigenvalues <- cmd$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**:

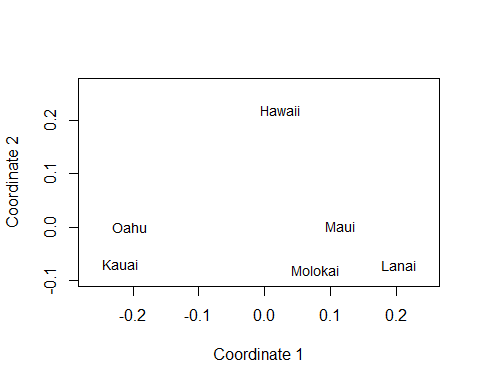
plot(eigenvalues)  
lines(lowess(eigenvalues))



How many axes should you keep?

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

x <- cmd$points[, 1]  
y <- cmd$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(cmd$points), cex = 0.9)

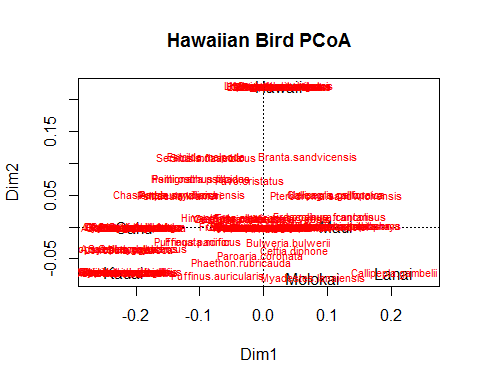


Another way to plot:

`?`(ordiplot)  
  
ordiplot(scores(cmd)[, 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  
`?`(wascores)  
  
species <- wascores(cmd$points[, 1:2], birds)  
text(species, rownames(species), cex = 0.7, col = "red")



## By “hand”

Ok, let’s now run a PCoA following the directions that I gave during lecture. The steps are up on the screen in front of the class:

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

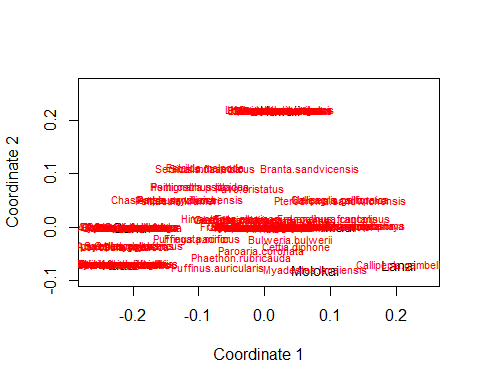
# You scale the eigenvectors by the square root of their eigenvalues to get  
# the coordinates (points):  
points2 <- sweep(EG$vectors[, 1:5], MARGIN = 2, sqrt(eigenvalues2), FUN = "\*")  
points2

## [,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:  
  
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.

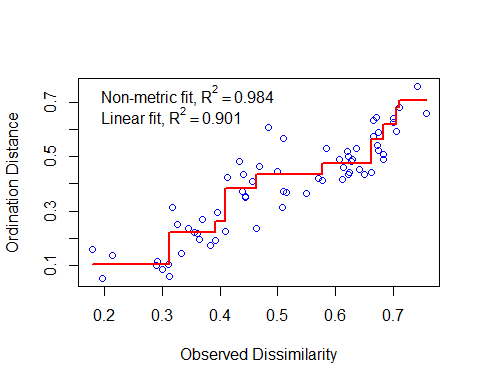
Create Sørensens’s disimilarity martix for the birds data:

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 dimension (which is common for NMDS).

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

stressplot(nmdsBird)



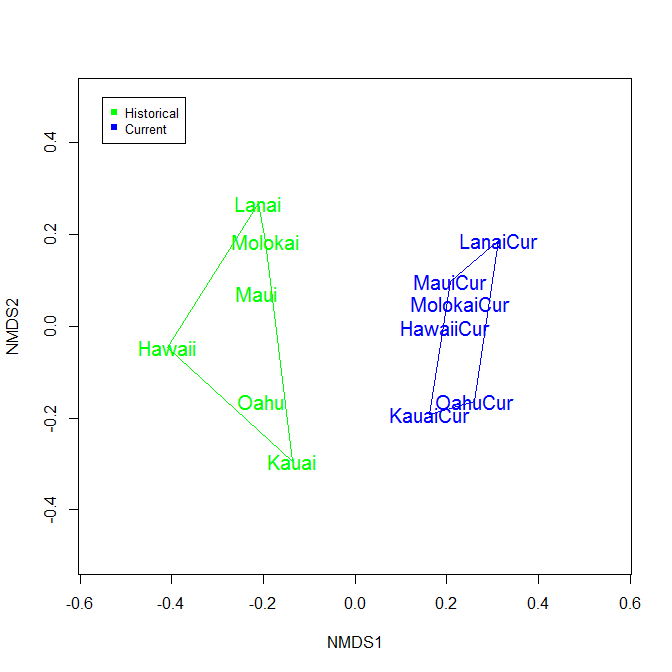
What do the stress value and the fit (R2) of the monotonic regression tell you about the NMDS plot?

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))

## species scores not available

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 a convex hull around each group:  
  
ordihull(nmdsBird, treat, display = "si", lty = 1, col = "green", show.groups = "Historical")  
ordihull(nmdsBird, treat, display = "si", lty = 1, col = "blue", show.groups = "Current")



# Correspondence Analysis (CA)

Correspondence analysis allows for the simultaneous ordination or rows and columns. It assumes a unimodal relationship between variables and the axes. We are going to use the *ca* package to start on the tree data set from lecture that looks at tree reproductive status and tree age.

`?`(ca)  
  
caTree <- ca(tree)

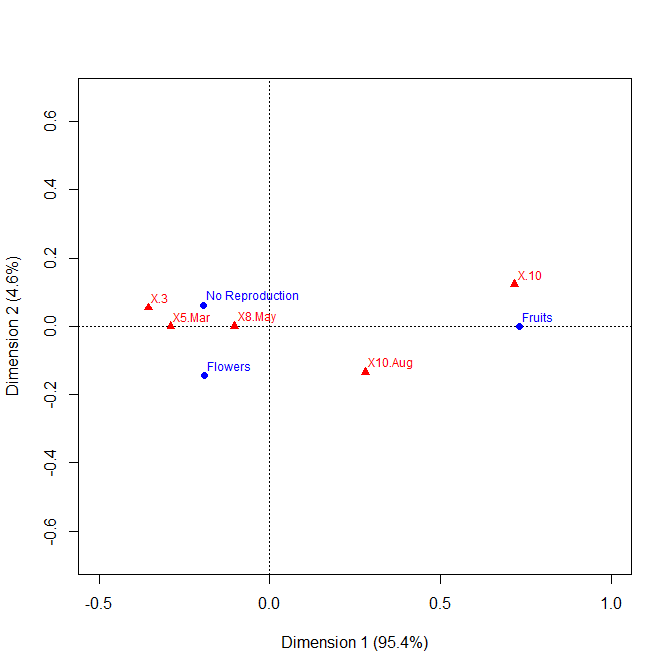
Let’s look at the results:

print(caTree)

##   
## Principal inertias (eigenvalues):  
## 1 2   
## Value 0.141348 0.006884  
## Percentage 95.36% 4.64%   
##   
##   
## Rows:  
## No Reproduction Flowers Fruits  
## Mass 0.553957 0.237410 0.208633  
## ChiDist 0.202728 0.239408 0.732220  
## Inertia 0.022767 0.013607 0.111858  
## Dim. 1 -0.514222 -0.511666 1.947588  
## Dim. 2 0.735371 -1.717649 0.002029  
##   
##   
## Columns:  
## X.3 X5.Mar X8.May X10.Aug X.10  
## Mass 0.223022 0.237410 0.172662 0.223022 0.143885  
## ChiDist 0.358977 0.289724 0.103281 0.311025 0.727481  
## Inertia 0.028740 0.019928 0.001842 0.021574 0.076148  
## Dim. 1 -0.943541 -0.770620 -0.274712 0.746272 1.906944  
## Dim. 2 0.663129 -0.003638 -0.001297 -1.617752 1.487225

In this analysis the *eigenvalues* are called *inertias*. The *Mass* is simply the column or row total. *ChiDist* is the distance from the origin in ordination space. Let’s now plot the ordination:

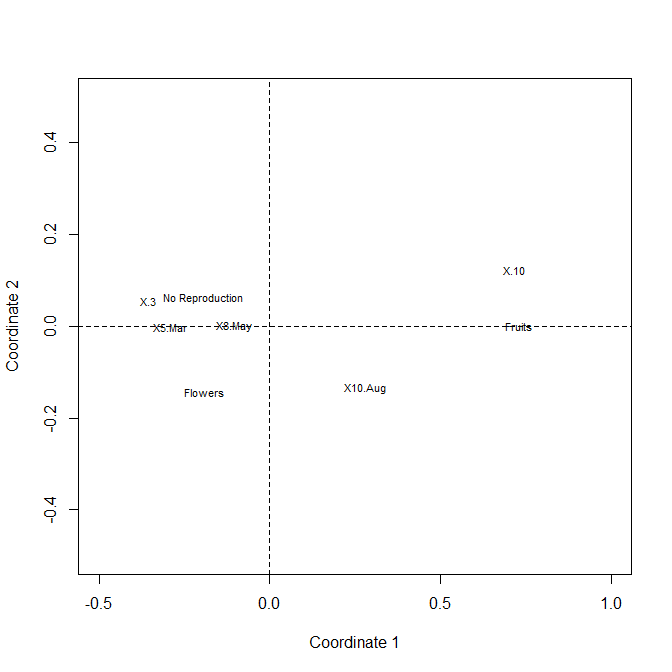
plot(caTree, xlim = c(-0.5, 1), ylim = c(-0.5, 0.5))



## By “hand”

Than was a painless function to run. Now let’s do it by hand (all but the singular value decomposition that is):

# Divide the data matrix caTree by the grand total of the matrix:  
  
p <- as.matrix(tree/sum(tree))  
  
# Cross tabulate row and column sums to be used in calculating expected  
# values for the Chi Square values:  
  
rs <- as.vector(apply(p, 1, sum))  
cs <- as.vector(apply(p, 2, sum))  
  
# Calculate expected values for the Chi Square calculation:  
cp <- rs %\*% t(cs)  
  
# Calculate Chi Square values and check them out:  
Qbar <- as.matrix((p - cp)/sqrt(cp))  
  
# Conduct singular value decomposition (svd):  
  
Q.svd <- svd(Qbar)  
  
  
# Scale eigenvectors for rows and columns by the square root of row and  
# column sums respectively:  
V <- diag(1/sqrt(cs)) %\*% Q.svd$v  
Vhat <- diag(1/sqrt(rs)) %\*% Q.svd$u  
  
# Calculate ordination coordinates for both rows and columns:  
  
F <- diag(1/rs) %\*% p %\*% V  
Fhat <- diag(1/cs) %\*% t(p) %\*% Vhat  
  
  
# Plot row and column coordinates in ordination space.  
  
plot(Fhat[, 1:2], xlim = c(-0.5, 1), ylim = c(-0.5, 0.5), type = "n", xlab = "Coordinate 1",   
 ylab = "Coordinate 2", lwd = 2)  
text(Fhat[, 1:2], labels = colnames(tree), cex = 0.7)  
text(F[, 1:2], labels = rownames(tree), cex = 0.7)  
abline(h = 0, lty = 2)  
abline(v = 0, lty = 2)



**If you are done early run these analyses on your own data. If you don’t have your own data or your data set is not suitable for PCoA, NMDS, or CA, use the *New\_England\_Ants.csv*. This data set contains presence/absence data for ant genera at four New England Sites.**