Aquatic Food Web Analysis

Hannah M. Carroll and Derek D. Houston

April 17th, 2019

* This is an R Notebook, which is written in RMarkdown format. Code is run inline in a code chunk, and the output appears directly below it.
* Today’s exercise will walk you through analysis of your aquatic isotope data and show you how to build a food web.
* When typing into an R Notebook, you don’t need to use # to mark text as a comment, unless you’re in an code chunk.

NOTE: WHEN YOU ARE DONE WITH THIS EXERCISE, KNIT THIS TO FILE TO HTML AND SUBMIT VIA BLACKBOARD

Your name:

A code chunk is standalone code that runs within the window. Code chunks begin with {r} and end with

Try executing this chunk by clicking the *Run* button (green right arrow) within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

# Set your working directory by choosing Session from the menu at the top, then Set Working Directory -> To Source File Location  
  
# (1pt) Write the command getwd() on the line below and run it to find out what your working directory is.  
getwd()

## [1] "C:/Users/Hannah/Documents/Research/For Derek/Derek teaching/Aquatic isotopes"

We will start by loading required packages

packages <- c("readxl", "ggplot2", "cluster", "factoextra", "tRophicPosition", "viridis") # Make a list of required packages  
  
# (1pt) How many packages are in our list?  
# 6  
  
new.packages <- packages[!(packages %in% installed.packages()[,"Package"])] # Which ones haven't been installed yet?  
  
# (1pt) What new packages are we installing?  
# This will vary. Probably 3-5.  
  
if(length(new.packages)) install.packages(new.packages) # Install any that are missing  
  
lapply(packages, library, character.only = TRUE) # Load packages into memory for use

## Warning: package 'ggplot2' was built under R version 3.5.3

## Warning: package 'factoextra' was built under R version 3.5.3

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

## Warning: package 'tRophicPosition' was built under R version 3.5.3

## This is tRophicPosition 0.7.7

## Loading required package: viridisLite

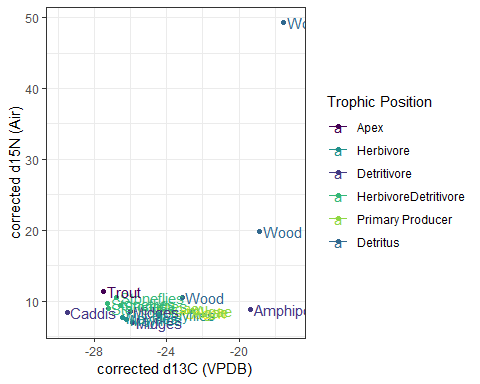
## [[1]]  
## [1] "readxl" "stats" "graphics" "grDevices" "utils" "datasets"   
## [7] "methods" "base"   
##   
## [[2]]  
## [1] "ggplot2" "readxl" "stats" "graphics" "grDevices" "utils"   
## [7] "datasets" "methods" "base"   
##   
## [[3]]  
## [1] "cluster" "ggplot2" "readxl" "stats" "graphics"   
## [6] "grDevices" "utils" "datasets" "methods" "base"   
##   
## [[4]]  
## [1] "factoextra" "cluster" "ggplot2" "readxl" "stats"   
## [6] "graphics" "grDevices" "utils" "datasets" "methods"   
## [11] "base"   
##   
## [[5]]  
## [1] "tRophicPosition" "factoextra" "cluster"   
## [4] "ggplot2" "readxl" "stats"   
## [7] "graphics" "grDevices" "utils"   
## [10] "datasets" "methods" "base"   
##   
## [[6]]  
## [1] "viridis" "viridisLite" "tRophicPosition"  
## [4] "factoextra" "cluster" "ggplot2"   
## [7] "readxl" "stats" "graphics"   
## [10] "grDevices" "utils" "datasets"   
## [13] "methods" "base"

Now, read in your dataset

isotope.data <- read\_excel("2019\_EA\_Houston\_class project.xlsx")  
  
# The dataset should now appear in your global environment (top right)  
  
# (1pt) How many observations are there?  
# 19  
  
# (1pt) How many variables are there?  
# 8

Next, make a simple dual isotope plot of your data. Error bars are plotted for both the x and y axes.

ggplot(data=isotope.data, aes(x=`corrected d13C (VPDB)`, y=`corrected d15N (Air)`, color=Trophic)) +   
 geom\_errorbar(aes(ymin=(`corrected d15N (Air)`-0.09), ymax=(`corrected d15N (Air)`+0.09))) +   
 geom\_errorbarh(aes(xmin=(`corrected d13C (VPDB)`-0.11), xmax=(`corrected d13C (VPDB)`+0.11))) +   
 geom\_point() +  
 geom\_text(aes(label=ID, hjust=0.5), check\_overlap = FALSE, hjust = 0, nudge\_x = 0.2) + xlim(-30,-17) +  
 scale\_color\_viridis\_d(name="Trophic Position", breaks=c("Apex", "Sec Consumer", "Herbivore", "Detritivore",  
 "HerbivoreDetritivore", "Primary Producer", "Detritus")) + theme\_bw()

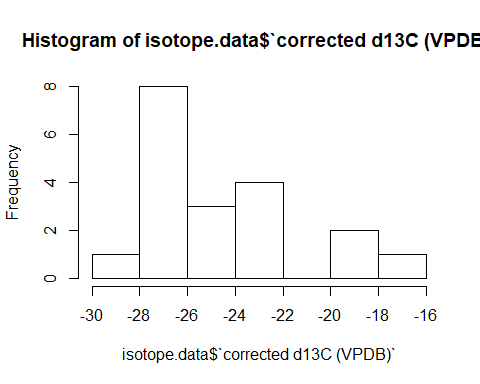


(2pts) Describe what you see in the plot you just created. Which members of the community group together and which do not? Do you notice anything that doesn’t seem to fit with the rest of the community?

(Answers will vary, but should note grouping of like creatures, wood off by itself, etc.)

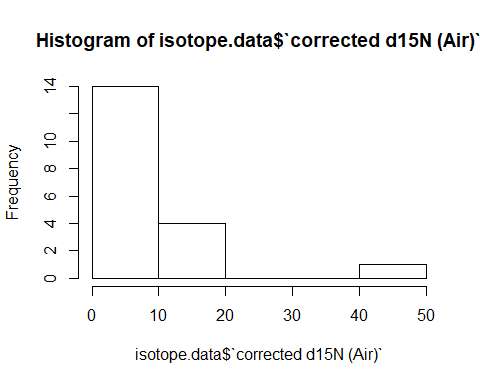
Now we’ll determine whether there are outliers in either d13C or d15N.

hist(isotope.data$`corrected d13C (VPDB)`)



# (1pt) Describe the distribution of our d13C data. Are they right skewed, left skewed, bimodal, or normally distributed?  
# Right skewed

hist(isotope.data$`corrected d15N (Air)`)

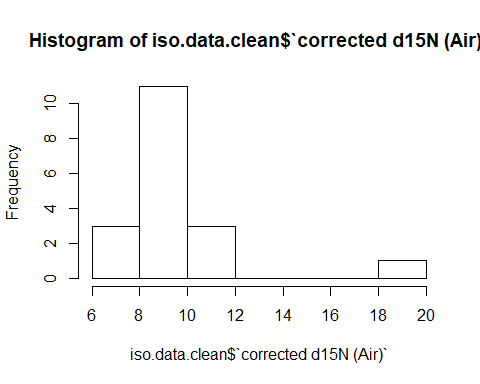


# (1pt) Describe the distribution of our d15N data. Are they right skewed, left skewed, bimodal, or normally distributed?  
# Right skewed

We can tell from our histograms that some of the observations in our d15N dataset are true outliers. We need to remove those to complete our analyses.

# The command below creates a new dataset by taking our original data and keeping only those rows where the value of d15N is less than 40.   
  
iso.data.clean <- isotope.data[isotope.data$`corrected d15N (Air)` < 40,]  
  
# (1pt) Why was this chosen as the cutoff point?  
# There is a big gap in the data and this looks an outlier.  
  
# (1pt) How many observations does our new dataset contain?  
# 18  
  
# (1pt) How many variables?  
# 8

# Now we plot the cleaned data:  
  
hist(iso.data.clean$`corrected d15N (Air)`)

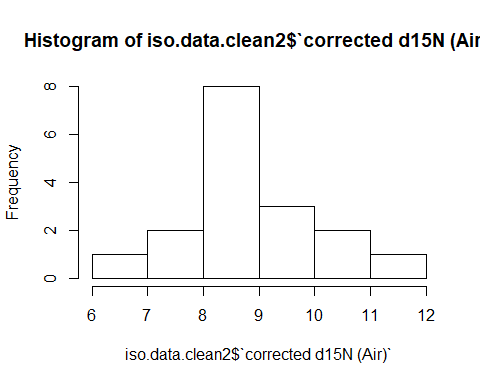


It looks a lot better. However, we still have one outlier.

(2pts) Complete the line of code below to remove it, using what we did before as a guide:

iso.data.clean2 <- isotope.data[isotope.data$`corrected d15N (Air)` < 18,]  
  
# iso.data.clean2 <- isotope.data[isotope.data$`corrected d15N (Air)` < 18,]

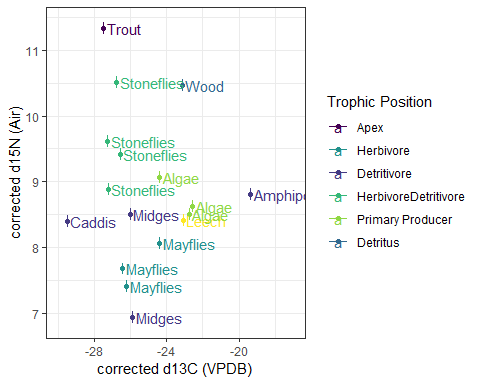
# Now plot the histogram:  
  
hist(iso.data.clean2$`corrected d15N (Air)`)



# (1pt) What kind of distribution do we have now? Is it right skewed, left skewed, bimodal, or normal?  
# Normal

Now we will plot the dataset with our two outliers removed. You can see the errorbars more clearly on this plot.

ggplot(data=iso.data.clean2, aes(x=`corrected d13C (VPDB)`, y=`corrected d15N (Air)`, color=Trophic)) +   
 geom\_errorbar(aes(ymin=(`corrected d15N (Air)`-0.09), ymax=(`corrected d15N (Air)`+0.09))) +   
 geom\_errorbarh(aes(xmin=(`corrected d13C (VPDB)`-0.11), xmax=(`corrected d13C (VPDB)`+0.11))) +   
 geom\_point() +  
 geom\_text(aes(label=ID, hjust=0.5), check\_overlap = FALSE, hjust = 0, nudge\_x = 0.2) + xlim(-30,-17) +  
 scale\_color\_viridis\_d(name="Trophic Position", breaks=c("Apex", "Sec Consumer", "Herbivore", "Detritivore",  
 "HerbivoreDetritivore", "Primary Producer", "Detritus")) + theme\_bw()



(1pt) Describe what you see in the new plot of the community. Which group together and which don’t?

* Answers will vary

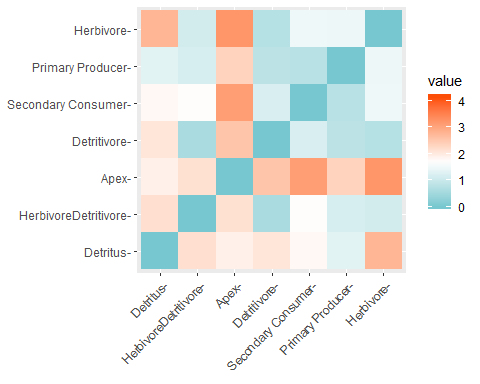
Now we can run analyses.

# This line scales the data so that they're comparable  
iso.scaled <- scale(iso.data.clean2[,c(5:6)])  
  
# We set the rownames to be the trophic level we think each sample belongs to.  
rownames(iso.scaled) <- iso.data.clean2$Trophic  
  
# You can find out the class of any object in your environment using the command class(name.of.object)  
# For example, the command class(iso.data.clean2) tells us that iso.data.clean2 is a data frame.  
  
# (1pt) On the line below, write the command to find the class of iso.scaled and run it.  
  
class(iso.scaled)

## [1] "matrix"

Now we can calculate the Euclidean distance of each sample in our community

distance <- get\_dist(iso.scaled)  
  
# Next, we plot the distances on a pairwise matrix  
  
fviz\_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))

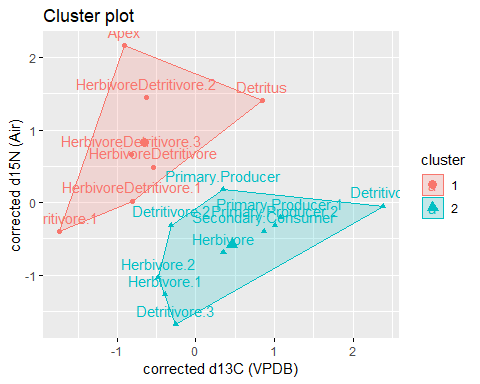


(5pts) Take a minute to study and describe this figure in your own words. What is it telling us about the community we sampled? Which trophic levels are most similar to each other? Which are the most different?

(2pts) The only secondary consumer in our dataset is a leech. Some leeches are parasitic, and some are free-living (i.e., are detritivores or herbivores). Look again at the graph above. Which kind of leech did we find? What evidence do you see?

Now we will perform a cluster analysis

# We have begun with two clusters (centers=2) and asked it to run 25 simulations of our data and find the best fit (nstart=25)  
# We've called it k2 to remind ourselves that we picked 2 clusters.  
  
k2 <- kmeans(iso.scaled, centers = 2, nstart = 25)  
fviz\_cluster(k2, data = iso.scaled)



(4pts) Try using different numbers of clusters to see how that changes the estimates of the community structure. Using the code above as a guide, try at least two more clusters of your own choosing in the code chunk below.

Remember to change BOTH the name of the object you create using the kmeans function, and the name of the object you call using the fviz\_cluster function.

# Students should use two different cluster numbers and include both code and output.

(2pts) How many clusters do you think best represent the data? Why? What are the risks of using too many clusters?

* Probably no more than 2 or 3 before you lose statistical power and create artificial grouping.

(3 extra credit points) Bonus question: Go back to the pairwise matrix we created. Which trophic level is most similar to the largest number of other members of the community? What does that tell you about its place in our food web?