Assignment #5

Janelle Morano

6 May 2022

1. Hierarchical Clustering to Explore Partitioning

Use stream reach properties to determine whether brook trout are likely to reside within a given stream reach

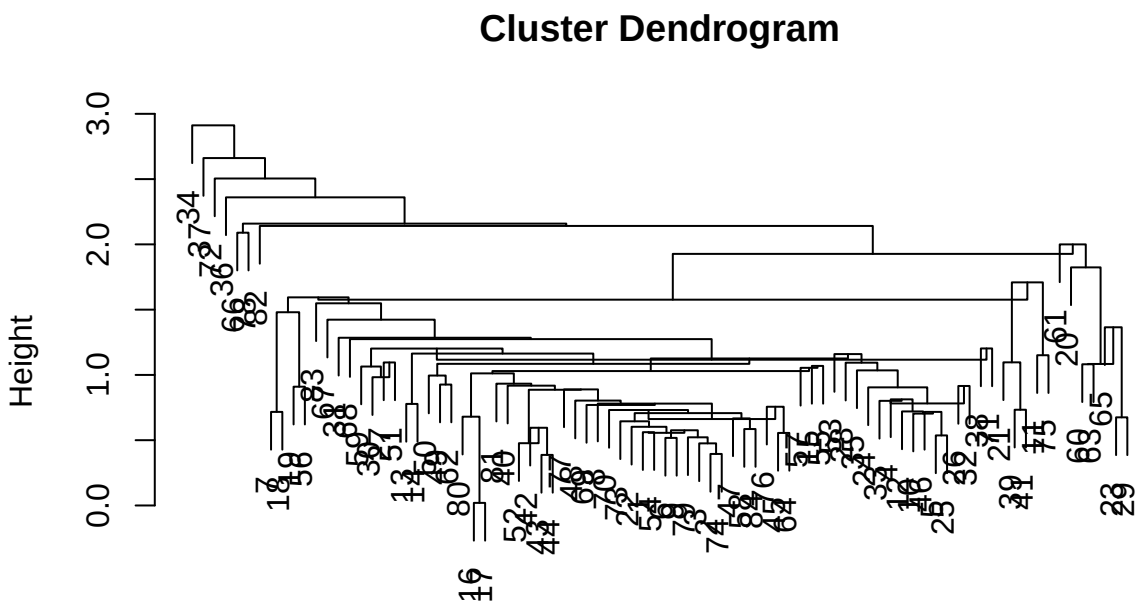
```
data <- read.csv("/Users/janellemorano/Git/Reference-R-scripts/Envtl-Multivariate-Stats/data/Maryland.T
    header = TRUE)
```

a) Calculate a distance matrix between scale reach properties (X.Ag, LOG_RD,TEMP_FLD,RIFQUAL,DO_FLD)

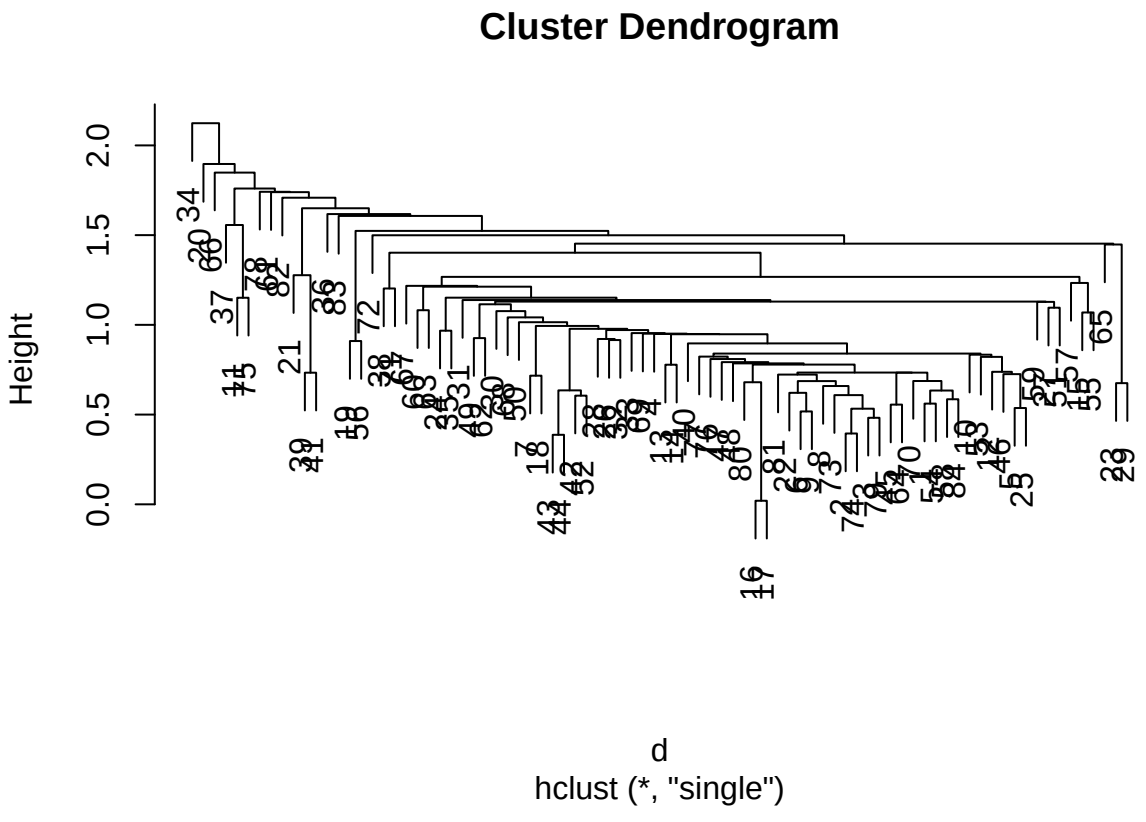
```
# function to standardize
scale2 <- function(x) {
    x <- (x - mean(x))/sd(x)
    return(x)
}

# Standardize each reach property in the dataset MARGIN = 2 applies the
# function scale2 to each column
scale.covariates <- apply(data[, 3:7], MARGIN = 2, FUN = scale2)
# head(scale.covariates)

d <- dist(scale.covariates)  # will be a lower triangular matrix of distances between each of 7 variabl
```

b) Using hclust() function, generate and plot 4 different dendrograms based on the distance matrix, using four different methods for clustering groups with more than one member ("centroid", "single linkage", "complete linkage", and "average").

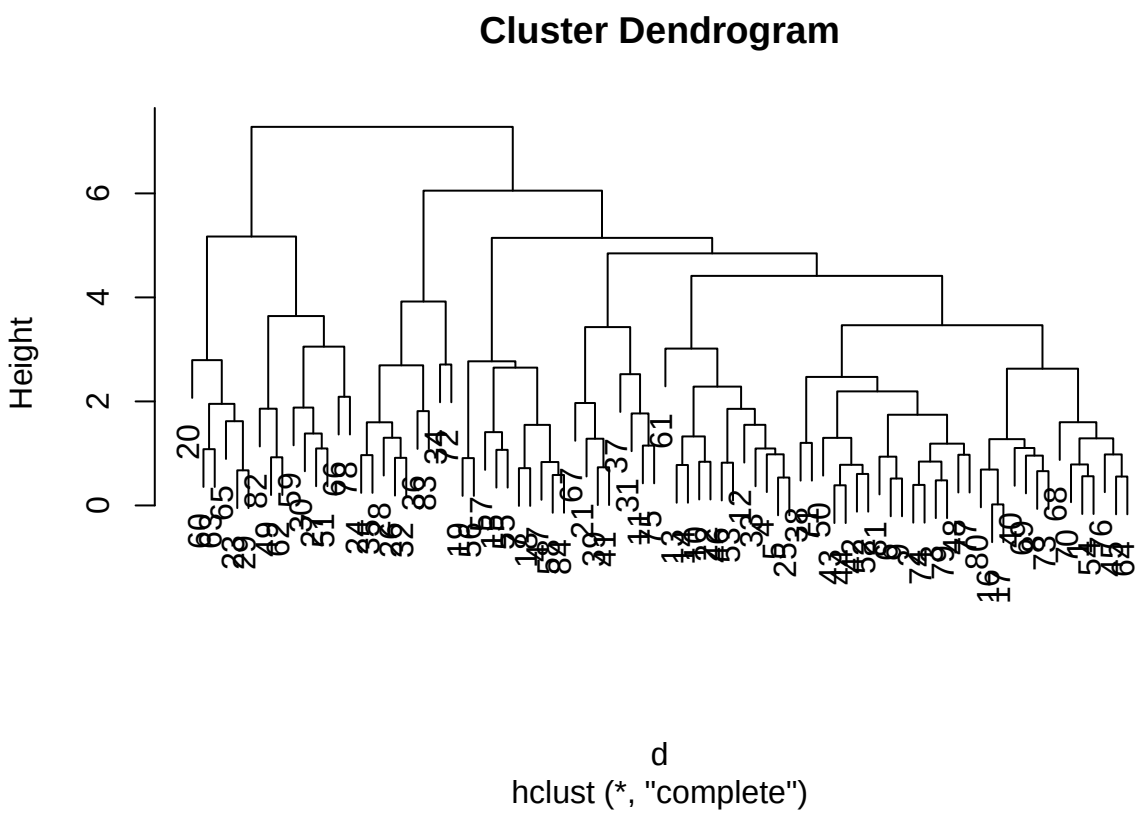
```
h.centroid <- hclust(d, "centroid")
h.single <- hclust(d, "single")
h.complete <- hclust(d, "complete")
h.average <- hclust(d, "average")

# par(mfrow = c(2,2))
plot(h.centroid)
```



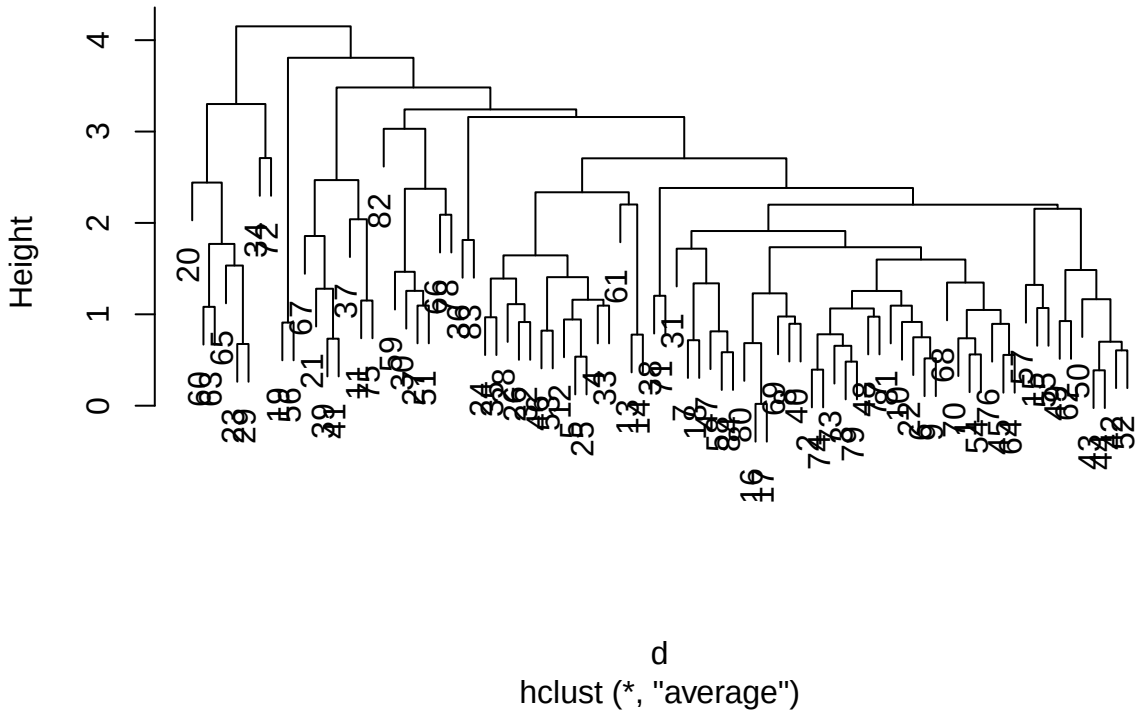

```
plot(h.single)
```



```
plot(h.complete)
```




```
plot(h.average)
```




```
# par(mfrow = c(1,1))
```

c) Interpret dendograms

The centroid and single dendograms have sequential chaining, or each step separates into a new cluster, so there aren't obvious major groupings. Additionally, the centroid has multiple inversions, making it difficult to interpret. Both of these don't appear to have major differences between clusters, and all points are more similiar to each other than not. The average dendogram shows better structure of major clusters at larger distances, but there still are many points that are more similar (closer) to each other than apart or different. The complete dendogram has a clearer cluster of 2, where the distance between the 2 is larger and more distinguishable than others (and especially compared to the other methods.)

d) Using the complete-linkage dendrogram and the cutree function(), cut the tree to allocate different observations to two different clusters, #row id

```
cluster <- cutree(h.complete, k = 2)  #row id
obs <- data.frame(cbind(cluster, scale.covariates, data$bkt))

# rename
library(dplyr)
obs <- obs %>%
    rename(bkt = V7)
```

e) Report the mean across 2 clusters

```
cluster1 <- obs %>%
    filter(cluster == 1)

cluster2 <- obs %>%
    filter(cluster == 2)
```

```r
# mean of variables for both clusters
cluster1.mean <- apply(cluster1[, 2:6], MARGIN = 2, FUN = mean)
cluster2.mean <- apply(cluster2[, 2:6], MARGIN = 2, FUN = mean)
mean.cluster <- rbind(cluster1.mean, cluster2.mean)

# print mean of clusters
print(mean.cluster)
```

```
##                      X.Ag  RIFFQUAL      LOG_RD   TEMP_FLD     DO_FLD
## cluster1.mean -0.07662015  0.315446 -0.01391697 -0.1787079  0.2271562
## cluster2.mean  0.35245271 -1.451052  0.06401808  0.8220563 -1.0449184
```

e, cont) Discust these results.

The means of each variable have opposite signs for each cluster, meaning, the variables are responding in opposition to each other. This signifies that the clusters are distinctly different from each other.

e, cont.) 2x2 contigency table of how trout presence/absence is distributed (for each cluster, how many have trout, how many don't)

```r
table(cluster = obs$cluster, trout = obs$bkt)
```

```
##        trout
## cluster  0  1
##       1 28 41
##       2 14  1
```

e, cont) What can you conclude from these results?

The first cluster contains almost all of the presence of brook trout, meaning, the variables within that cluster are more indicative of habitat that is associated with brook trout. The second cluster, has variables associated with less brook trout presence, in that, the second cluster is not indicative of brook trout presence, in other words, is not associated with brook trout.

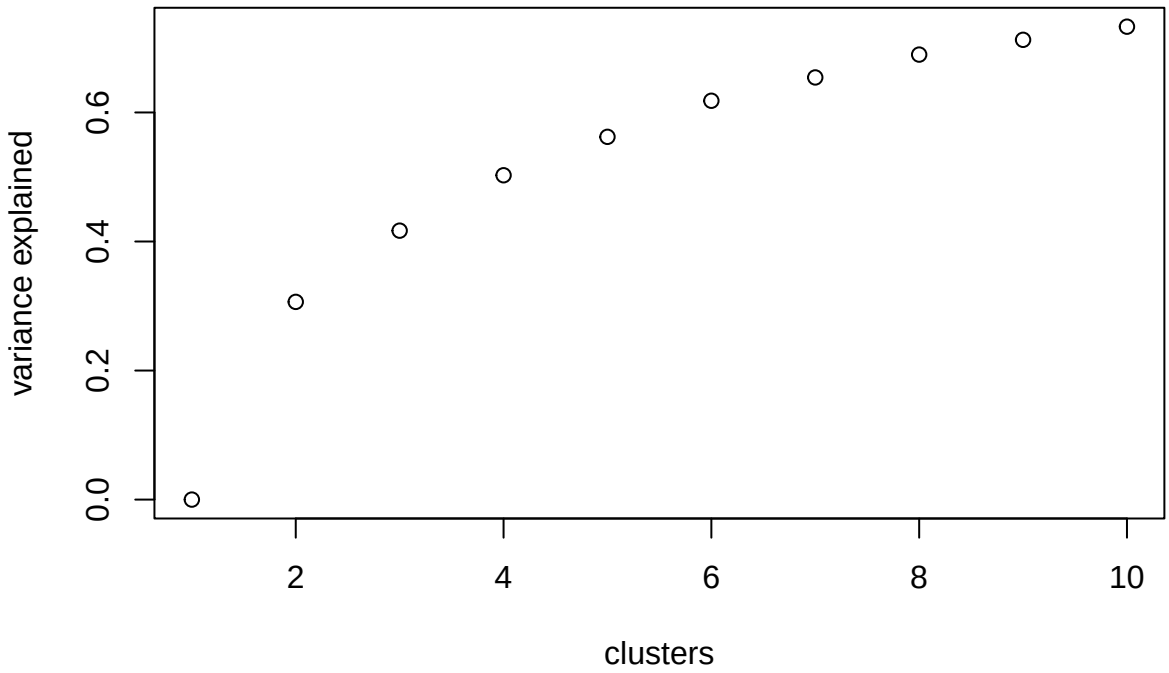
2. K-Means Clustering to Explore Partitioning

f) run kmeans for groups of k=1:10

```r
kmeans.list <- vector(mode = "list", length = 10)
for (i in 1:10) {
    kmeans.list[[i]] <- kmeans(scale.covariates, centers = i, nstart = 10)
}
```

g) Plot the variance explained by the clusters against the number of clusters for each of the different values of K tested.

```r
evar <- c()
for (i in 1:10) {
    evar[i] <- kmeans.list[[i]]$betweenss/kmeans.list[[i]]$totss
}
clustevar <- cbind(rep(1:10), evar)
plot(clustevar[, 1], clustevar[, 2], xlab = "clusters", ylab = "variance explained")
```



K=2 is a reasonable number of clusters for further analysis because there is a larger difference in variance between the 2 clusters, making them more likely to be distinguishable from each other. The >2 clusters have variance that are more similar to each other, meaning the clusters may not be distinguishable or very different from each other.

h) show how the mean characteristics of reaches differ across the two clusters. Also similar to (e) above, present a table that shows how trout presence/absence is distributed across the clusters. What can you conclude from these results, and how do they compare to the results using the hierarchical clustering approach?

```
kmeans2 <- data.frame(cbind(kmeans.list[[2]]$cluster, scale.covariates, data$bkt))
kmeans2 <- kmeans2 %>%
    rename(cluster = V1, bkt = V7)
kmeans2.clst1 <- kmeans2 %>%
    filter(cluster == 1)
kmeans2.clst2 <- kmeans2 %>%
    filter(cluster == 2)

# mean of variables for both clusters
kmeans2.clst1.mean <- apply(kmeans2.clst1[, 2:6], MARGIN = 2, FUN = mean)
kmeans2.clst2.mean <- apply(kmeans2.clst2[, 2:6], MARGIN = 2, FUN = mean)
kmeans2.means <- rbind(kmeans2.clst1.mean, kmeans2.clst2.mean)

# print mean of clusters
print(kmeans2.means)
```

```
##                          X.Ag   RIFFQUAL      LOG_RD   TEMP_FLD     DO_FLD
## kmeans2.clst1.mean  0.8456312 -0.8326158 -0.17298792  0.9271163 -0.7570825
## kmeans2.clst2.mean -0.4458783  0.4390156  0.09121181 -0.4888431  0.3991890
```

Similar to the hierarchical clustering, the means of each variable have opposite signs for each cluster, meaning, the variables are responding in opposition to each other. This signifies that the 2 clusters are distinctly different from each other.

```r
# contingency table
table(cluster = kmeans2$cluster, trout = kmeans2$bkt)
```

```
##         trout
## cluster  0  1
##       1 21  8
##       2 21 34
```

Again, one of the K-means clusters has variables associated with more presence of brook trout than the other. The averages of the variables within that cluster (cluster 2 here) are similar to the hierarchical cluster that was indicative of brook trout presence: negative X.Ag (less percent of agricultural land cover in the watershed), positive RIFFQUAL (more suitable shallow areas of habitat), negative TEMP_FLD (lower spring water temperatures), and positive DO_FLD (higher concentrations of dissolved oxygen). Both approaches have opposing LOG_RD signs, where, here, the distance from the nearest road is larger, but in the hierarchical cluster, the distance was lower.

3. Discrimination and Classification

i) First, split the original dataset into a training and testing dataset. Let the training dataset be composed of the first 60 observations, and the testing data be composed of the remaining 24 observations.

```r
# Take scaled covariates, but need also presence/absence of brook trout
trout.scaled <- apply(data[, 3:7], MARGIN = 2, FUN = scale2)
trout.scaled <- cbind(trout.scaled, data$bkt)

# Training data, first 60 obs
training <- trout.scaled[1:60, ]
# Testing data, last 24 obs
testing <- trout.scaled[61:84, ]
```

j) Now, split the training data into two groups, one with and one without the presence of trout.

```r
training <- data.frame(training)
training0 <- training[training$V6 == 0, ]
training0 <- as.matrix(training0[, 1:5])
training1 <- training[training$V6 == 1, ]
training1 <- as.matrix(training1[, 1:5])

# First, find the number of observations associated with each group.
n0 <- nrow(training0)
n1 <- nrow(training1)

# Second, find the pooled covariance matrix across the two groups. Spool =
# within group var-cov ...by first getting var-cov matrix for each group
cov0 <- cov(training0)  #could add use='pairwise.complete.obs' but isn't neccessary here
cov1 <- cov(training1)

# ...then calculate the Spool (pooled variance)
Spool <- ((n0 - 1)/(n0 + n1 - 2)) * cov0 + ((n1 - 1)/(n0 + n1 - 2)) * cov1

# Third, calculate the discrimination function (alpha).  ...get the mean of
# each group
mean0 <- apply(training0, MARGIN = 2, FUN = mean)
mean1 <- apply(training1, MARGIN = 2, FUN = mean)
```

```r
# ...then, alpha = (Spool)^-1 * (mean group 1 - mean group 2)
alpha <- solve(Spool) %*% (mean0 - mean1)
```

k) Finally, classify observations in testing based on alpha

```r
# ...Find sigma for each group (presence or absence) = t(alpha) %*% mean of
# training data
sigma.0 <- t(alpha) %*% mean0
sigma.1 <- t(alpha) %*% mean1
# This is what will be compared to sigma.star for each obs

# ...Then find sigma* for each obs in testing data...
testing.matrix <- as.matrix(testing[, 1:5])

predict <- c()
# i <-2
for (i in 1:nrow(testing.matrix)) {
    sigma.star <- t(alpha) %*% testing.matrix[i, ]
    # ...then compare sigma* to sigma.0 and sigma.1
    dis1 <- abs(sigma.star - sigma.0)
    dis2 <- abs(sigma.star - sigma.1)

    # Compare dis1 and dis2, if dis1 < dis2, that means dis1 is more similar to
    # sigma (or the 'score' of the variables) and therefore sigma.star is
    # absence of trout
    if (dis1 < dis2) {
        predict[i] <- 0
    } else {
        predict[i] <- 1
    }

}

testing.predict <- cbind(testing, predict)

# Table of 1: No trout present, and you estimate no trout present 2: No trout
# present, and but you estimate trout present 3: Trout present, but you
# estimate no trout present 4: Trout present, and you estimate trout present

table(true.trout = testing.predict[, 6], predict.trout = testing.predict[, 7])
```

```
##           predict.trout
## true.trout  0  1
##          0 12  5
##          1  1  6
```

```r
TPR <- (6/(6 + 1))  #True positive rate, TPR, sensitivity, TP/TP+FN
print(TPR)
```

```
## [1] 0.8571429
```

```r
TNR <- (12/(12 + 5))  #True negative rate, TNR, specificity, TN/TN+FP
print(TNR)
```

```
## [1] 0.7058824
```

```r
FP <- 1 - TNR  #False positive rate, overestimation, Type I error
print(FP)
```

```
## [1] 0.2941176
```

```r
FN <- 1 - TPR  #False negative rate, underestimation, Type II error
print(FN)
```

```
## [1] 0.1428571
```

k, cont) Discuss skill of classification model

The true positive rate (TPR = 86%) is high, such that the model correctly identifies the presence of trout 86% of the time, and the specificity, or true negative rate (TNR = 71%) is also fairly high, with the model correctly identifying the absence of trout 71% of the time. The false positive (29%) and false negative rates (14%) are low, meaning that the classification model does not have a high rate of overestimating or underestimating, respectively, brook trout.