Lab 5 Cluster Analysis Part I

The goal of this lab is to become familiar with the application the non-hierarchical clustering procedure K-means clustering.

# Set up R session

**Again, you are pros at this by now…**

The goal of this lab is to become familiar with the application the non-hierarchical clustering procedure K-means clustering.

## Data

Call in the data set “snail\_data.csv” from your working directory and name it **snails**. This data set consists of three morphological measurements and four environmental measurements for the snail *Littoraria angulifera* at 9 geographic locations ranging from western Africa to the Caribbean and South America. **For this analysis we are only going to use the first three variables** which are the shell morphology measurements. Remember to denote that the first row and column are row names and column names respectively.

snails<-read.csv("Data/lab\_5/snail\_data.csv", row=1, header=TRUE)[,1:3]

## Download packages

You will be using new packages cluster and raster

library(raster)  
library(cluster)  
library(mvnormtest)  
library(MVN)

# K-means clustering

**K-means** clustering is a **non-hierarchical** clustering method that seeks to find groups that maximize within-group homogeneity:

?kmeans

K-means clustering does have the assumption of multivariate normality. Check the histograms of each variable (as discussed in lab 2) and determine if you need to transform them. Let me know what you think.

Let’s also test for multivariate and univariate normality (even though these tests are conservative)

mshapiro.test(t(snails))  
  
mvn(snails, mvnTest = "mardia")

Compare test results to the histograms you plotted.

If variables (columns) are measured on different scales or have large differences in variance, you must scale the variables (columns). Let’s check this using the coefficient of variation (cv) on column totals:

snail.tot<- apply(snails,2, sum)  
cv(snail.tot)

## [1] 39.58346

What do you think? Should you scale the snail data set?

If so, *z standardize* (scale function) the variables.

snails<-scale(snails)

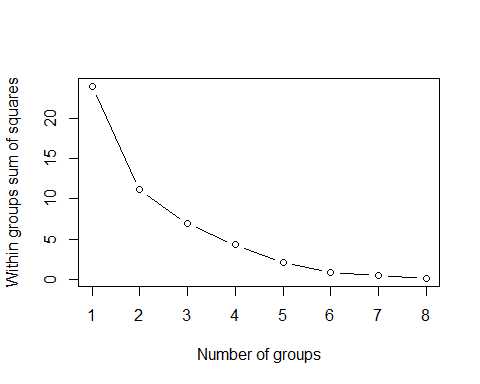
If you remember from lecture, one way to determine the number of clusters, *k*, for the k-means cluster analysis is to look at a scree plot for the within-group sum of squares. Let’s do a loop that will find the within group sum of squares for the full range of possible cluster solutions (nrows-1). The snails data set has nine samples (rows), so you will explore solutions from 1-8 clusters.

#Set a vector for the loop to fill.   
wss <- rep(0, 8)  
  
#Run a loop for 1 to 8 clusters:  
for (i in 1:8) # sets the number of times the loop will be run i.e., the number of clusters in this case)  
   
wss[i] <- sum(kmeans(snails, centers = i,nstart=25)$withinss) # run the kmeans function for each number of clusters (i) and extract the within sum of squares for each.  
  
#Check out you vector of within group sum of squares for one to eight groups:  
wss

## [1] 24.0000000 11.1259137 6.9344101 4.2727792 2.0731364 0.8488141 0.4801028  
## [8] 0.1127559

Make the scree plot:

plot(1:8, wss, type = "b", xlab = "Number of groups", ylab = "Within groups sum of squares")



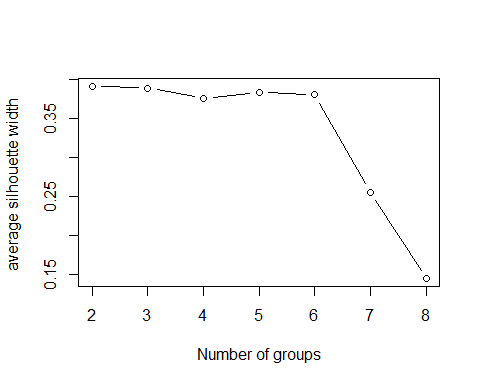
How many clusters are present in the best K-means solution?

Another indicator of how many cluster to use is looking at the average silhouette width.

?silhouette

We can run a similar loop to look at the average silhouette width:

sil <- rep(0,8)  
for (i in 2:8)  
 sil[i] <- summary(silhouette(kmeans(snails, centers=i, iter.max=100, nstart=25)$cluster, dist(snails)))$avg.width  
plot(2:8, sil[2:8], type = "b", xlab = "Number of groups", ylab = "average silhouette width ")



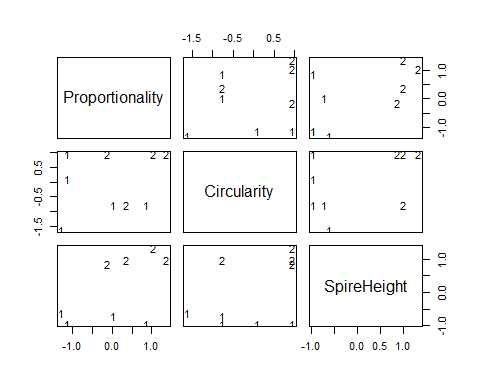
Does the plot for the average silhouette width concur with the scree plot for the within-group sum of squares?

Let’s go ahead and plot out the result for the optimal cluster solution that you just found. Fill in the code for the number of centers (clusters) you found above:

snails.kop <- kmeans(snails, centers= 2, iter.max=10, nstart=25)

Plot a scatter plot showing cluster designations:

pairs(snails, panel=function(x,y,z) text(x,y,snails.kop$cluster))



Which morphological metric really separates the two groups?

Now, let’s plot them against principal components 1 and 2:

Run, plot and interpret the PCA:

snail.pc <- princomp(snails, cor=F)  
summary(snail.pc)

## Importance of components:  
## Comp.1 Comp.2 Comp.3  
## Standard deviation 1.3007722 0.8202981 0.5493355  
## Proportion of Variance 0.6345031 0.2523333 0.1131636  
## Cumulative Proportion 0.6345031 0.8868364 1.0000000

snail.pc$loadings

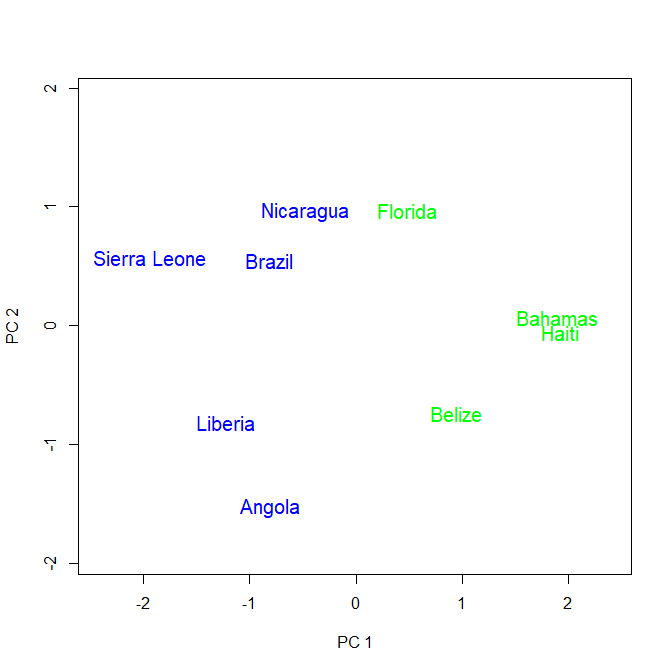
##   
## Loadings:  
## Comp.1 Comp.2 Comp.3  
## Proportionality 0.589 0.533 0.608  
## Circularity 0.486 -0.834 0.260  
## SpireHeight 0.646 0.143 -0.750  
##   
## Comp.1 Comp.2 Comp.3  
## SS loadings 1.000 1.000 1.000  
## Proportion Var 0.333 0.333 0.333  
## Cumulative Var 0.333 0.667 1.000

Set up colors for each cluster:

my.color.vector <- rep("green", times=nrow(snails))  
my.color.vector[snails.kop$cluster==1] <- "blue"  
my.color.vector[snails.kop$cluster==2] <- "green"

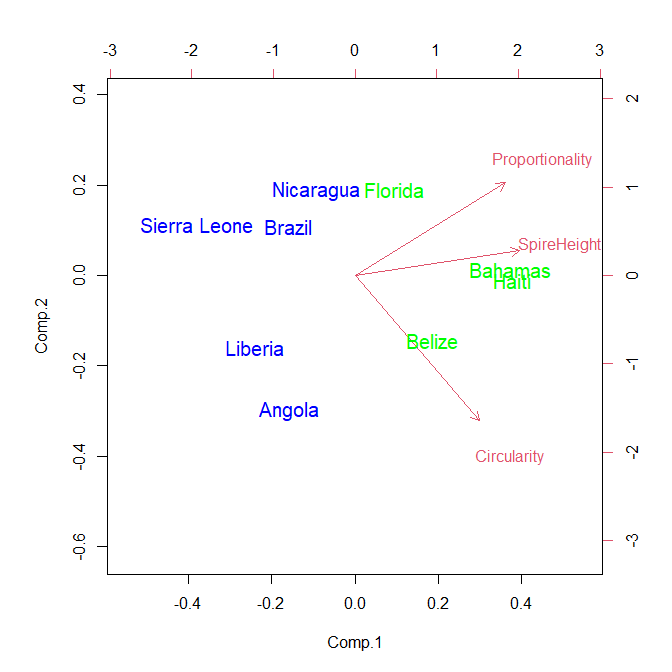
Plot clusters:

plot(snail.pc$scores[,1], snail.pc$scores[,2], ylim=range(snail.pc$scores[,1]),xlim=range(snail.pc$scores[,1]\*1.25), xlab="PC 1", ylab="PC 2", type ='n', lwd=2)  
text(snail.pc$scores[,1], snail.pc$scores[,2], labels=rownames(snails), cex=1.25, lwd=2,  
 col=my.color.vector)



Plot clusters onto biplot:

biplot(snail.pc, xlabs= rep("",9),xlim=range(-.55,.55))  
text(snail.pc$scores[,1], snail.pc$scores[,2], labels=rownames(snails), cex=1.25, lwd=2,  
 col=my.color.vector)



How does plotting your clusters on a bi-plot add to your interpretation of the cluster analysis and the PCA?

**This was potentially a short lab. Run through your own data with K-means clustering or use a data set from an earlier lab (e.g.usAirpollution). Take advantage of the extra time to ask questions about previous analyses or you own data.**