Stats503 Hw5

David Li

April 16, 2018

Data Report for Stats503 Homework 5

– Introduction: Clustering –

Alternatively to classification which is a supervised learning method, sometimes we may not have true class labels to guide building the statistical learning model. This other approach is known as unsupervised learning, where clustering is one class type of such methods. Clustering has various approaches, but regardless are all based around identifying and finding clusters where a cluster is a designed way of grouping similar objects than into other clusters. For instance, clusters might be identified by between-distances or density measurements of hypothesized clusters. Unlike supervised learning where we may measure misclassification error to assess performance, building clustering models is an iterative and experimental process to continuously observe the possible models until a satisfactory sensible solution is achieved. This will be practiced in the following analysis, where we will execute clustering and continue to tune needed parameters for a final clustering model.

– Data & Procedure for Demonstration –

We perform our analysis on the crabs data set, found at the Canvas website. It details physical features of Australian 'Leptograpsus variegatus' crabs such as sex, frontal lobe size, carapace descriptions, and more. A detailed description of each feature is listed below in Figure 1.

Name of Crab Feature	Feature Description
Species	1 = Blue crabs, 2 = Orange crabs
Sex	1 = Male, 2 = Female
FL	Frontal Lobe Size (mm)
RW	Rear Width (mm)
CL	Carapace Length (cm)
CW	Carapace Width (cm)
BD	Body Depth (cm)

Before any clustering methods can be considered, the differing magnitudes of the data features need to be addressed. For instance, frontal lobe size is measured in millimeters while carapace width is measured in centimeters. Clustering methods are utilizing an aspect of "distance" between observations, so extremely imbalanced scales of different features can cause issues in distance measuring for the clustering procedure. Additionally, scaling can make the features more comparable to each other so as to give features equal weight for sensible clustering results. If the features have well-defined meanings that cannot afford to be distorted (such as latitude or longitude), then the scales should be preserved in these cases. Currently, scaling all length measurements in the crabs dataset is sensibly valid and will increase reliability and accuracy in the conclusions from the results. Snippet of the scaling process are shown below:

Before:

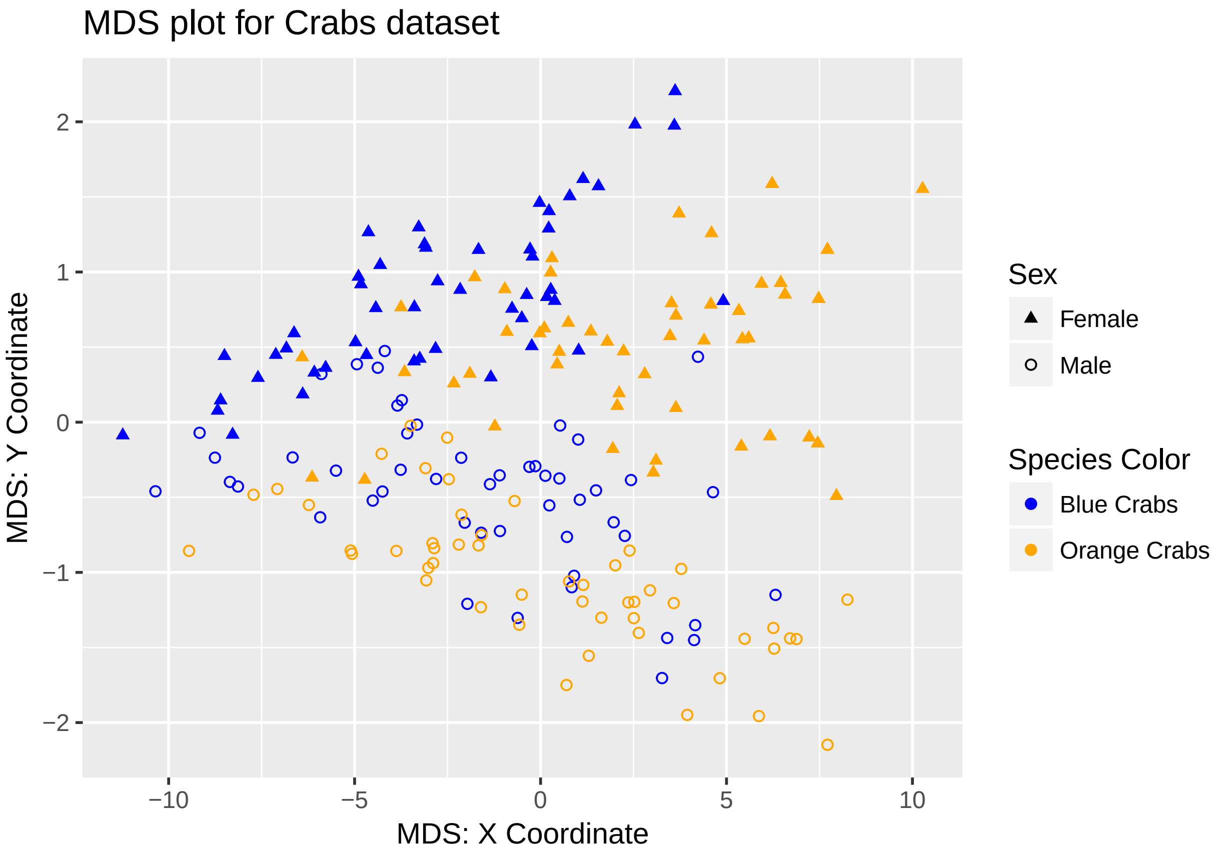
```
##   Species Sex  FL  RW   CL   CW   BD
## 1       1   1 8.1 6.7 1.61 1.90 0.70
## 2       1   1 8.8 7.7 1.81 2.08 0.74
## 3       1   1 9.2 7.8 1.90 2.24 0.77
```

After:

```
##   Species Sex        FL        RW        CL        CW        BD
## 1       1   1 -2.168888 -2.326227 -2.299249 -2.267843 -2.093423
## 2       1   1 -1.974395 -1.958431 -2.021520 -2.041076 -1.979088
## 3       1   1 -1.863257 -1.921651 -1.896542 -1.839506 -1.893337
```

– Approach 1: –

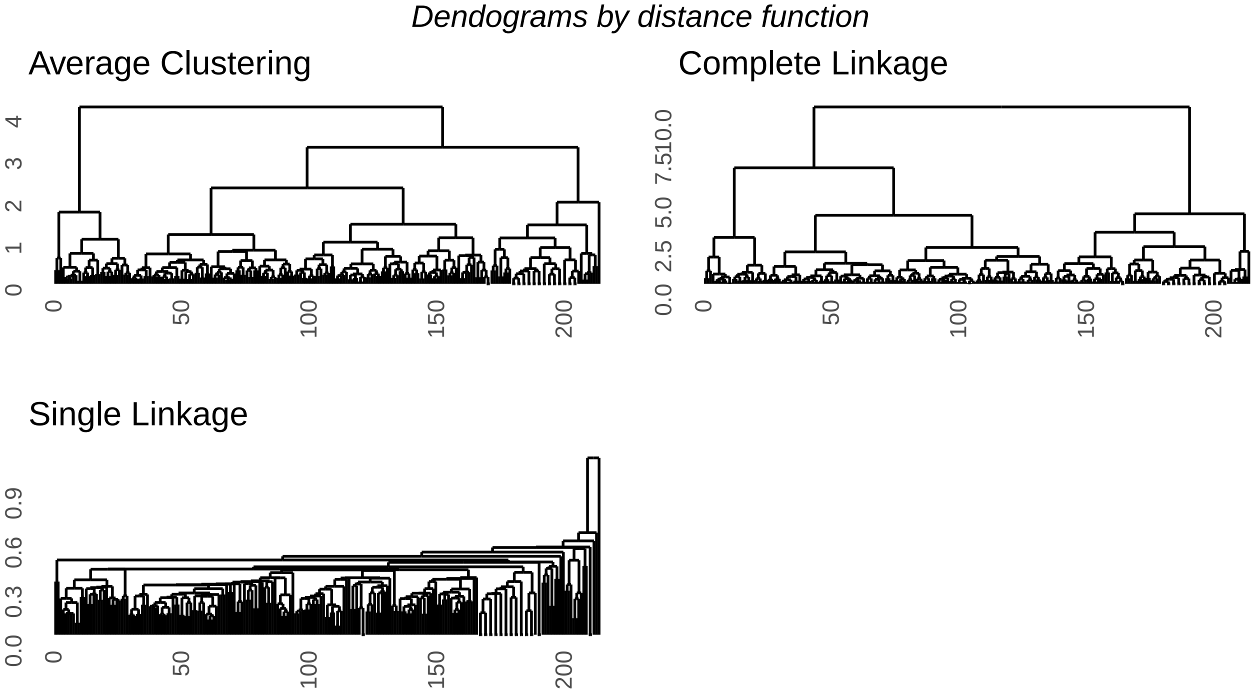
Consider clustering only the 5 scaled numerical variables for the crabs dataset: FL, RW, CL, CW, BD. Comparisons between Hierarchical clustering, K-means, and mixture modeling will be pursued in analysis below. Visualization of the clustering can be difficult in 5 dimensions due to the 5 numerical variables, so Multidimensional Scaling can be first applied to reduce the dimension of the data to two dimensions. An initial MDS plot is provided for viewing:



The MDS plot already indicates that there is a fair amount of overlapping of the observations by species color, foreshadowing that certain given patterns of crab features will not reliably indicate what kind of color the crab might be (since either color might be possible). However certain given patterns of crab features might be good indication of what likely sex the crab may be; this is supported that the MDS shows a fairly nice divide between the female crab observations versus the male crab observations.

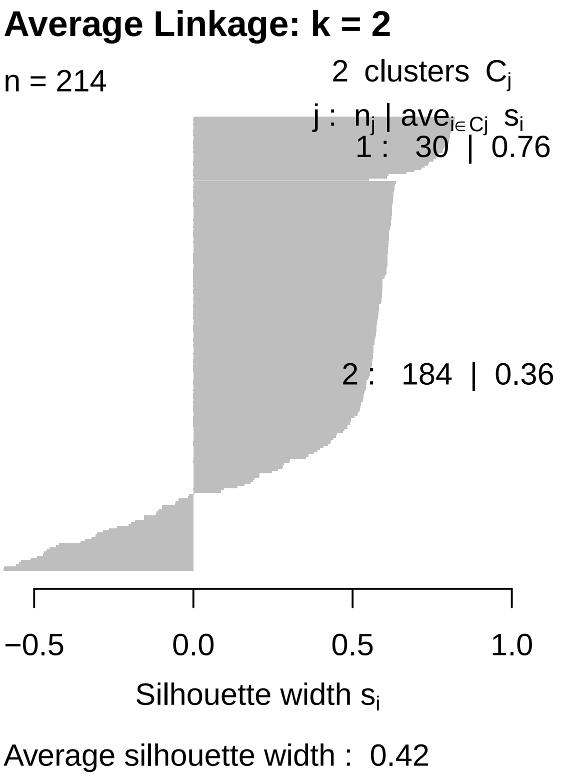
Hierarchical Clustering

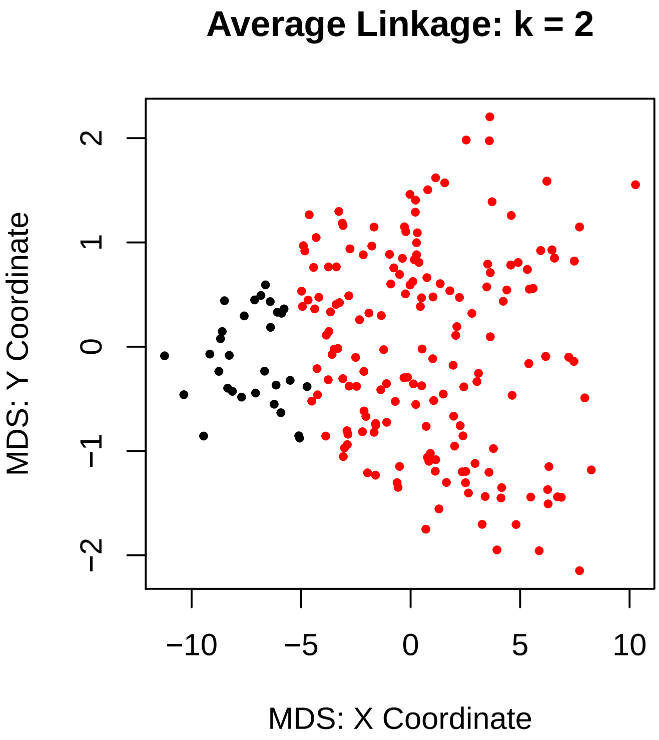
Average Linkage is a distance definition for the hierarchical clustering, based on the average distance between each point in one cluster to all other points in another cluster. Complete Linkage is another distance definition, based on the farthest distance obtainable between any two points; one in one cluster and one in another cluster. Finally, Single Linkage is our 3rd and last considered distance definition which is the same as complete linkage except the minimum distance is found in place of the maximum distance between two points from different clusters. To evaluate the "performance" of the clustering methods, we may analyze the silhouette plots and comment upon them. This is done by comparing how close the object is to other objects in its own cluster with how close it is to objects in other clusters, on a range of 0 (for not a good placement in the cluster) and 1 (for good placement in the cluster). Dendrograms are also created, showing the cluster processing in action.

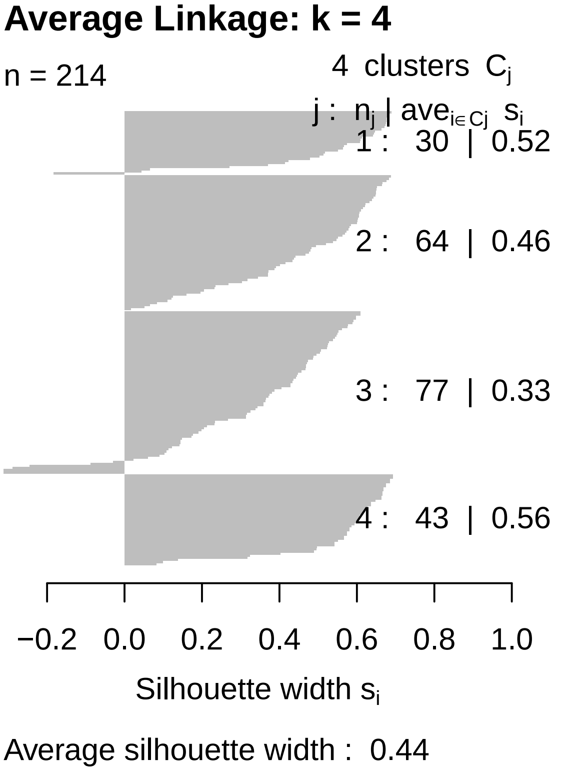


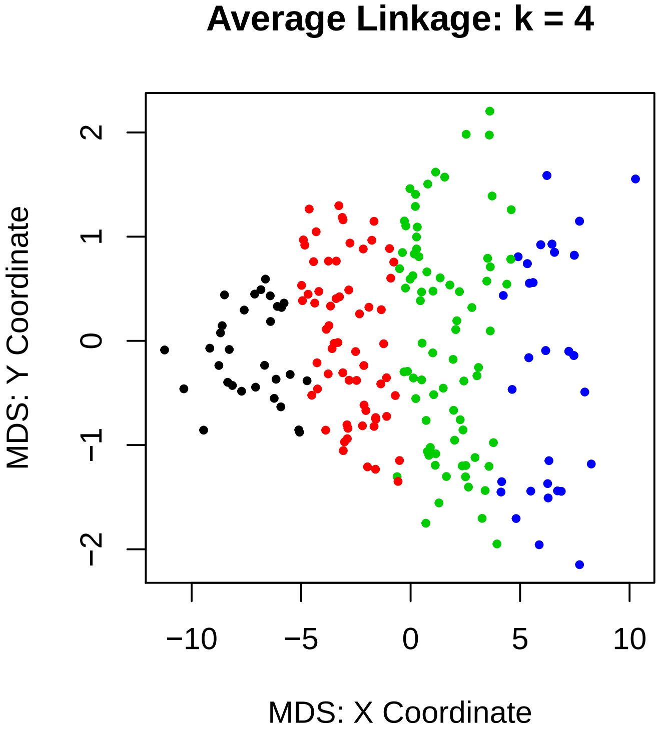
These are highly simplified dendrograms for comparison purposes only, but more magnified versions with labels could provide detailed microscopic insights. At a grand level, this also foreshadows that single linkage will perform poorly due to the extensive splitting the model does. Complete and average linkage look neatly splitted, giving an intuitive thought that their cluster processing will be much cleaner and sensical. We can gauge performance better by next analyzing the silhouette plots side-by-side with the proposed clustering applied to the scaled crabs data shown through MDS. An easy way to evaluate the clustering is to simply look at the average silhouette length outputted by the silhouette plots. 0.71-1.0 means a strong structure has been found, 0.51-0.70 means a reasonable structure has been found, 0.26-0.50 means the structure is weak and could be artificial, and < 0.25 means no substantial structure has been found.

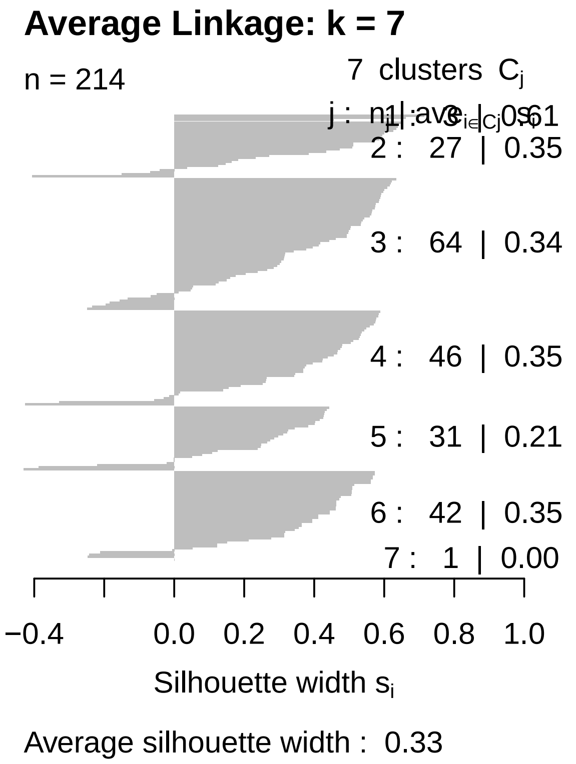
Average Linkage

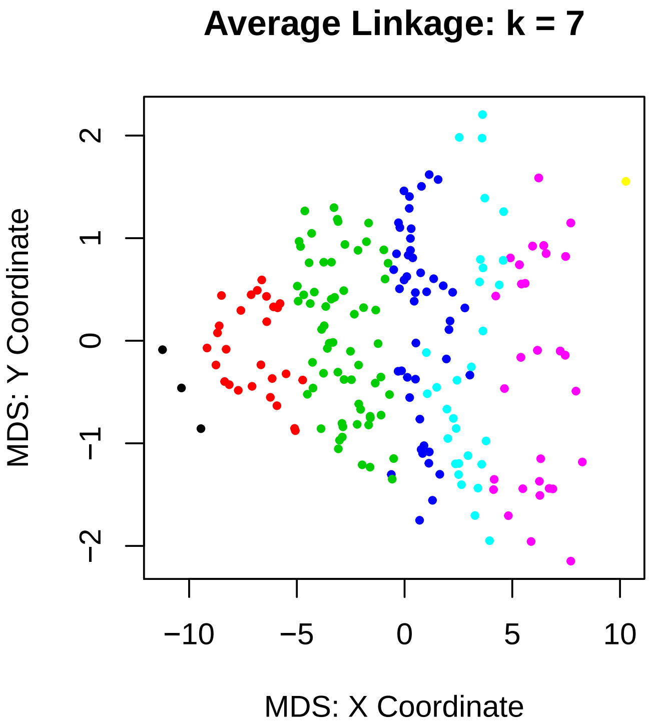






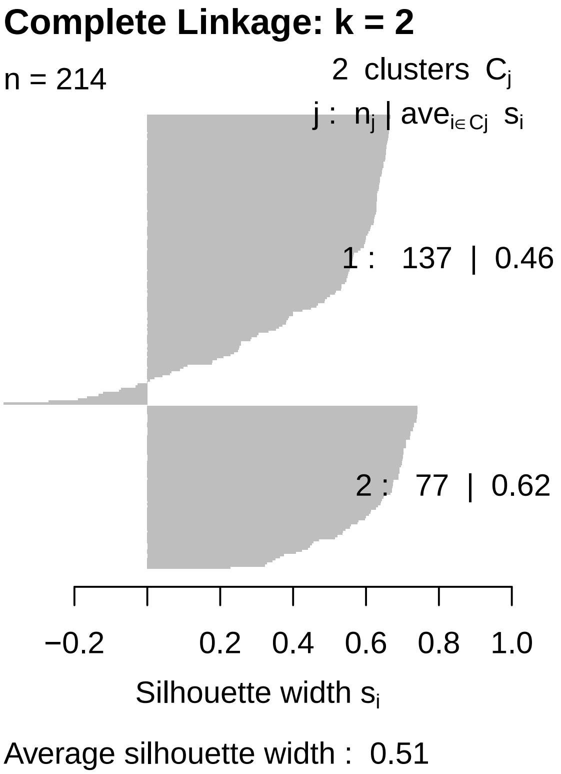


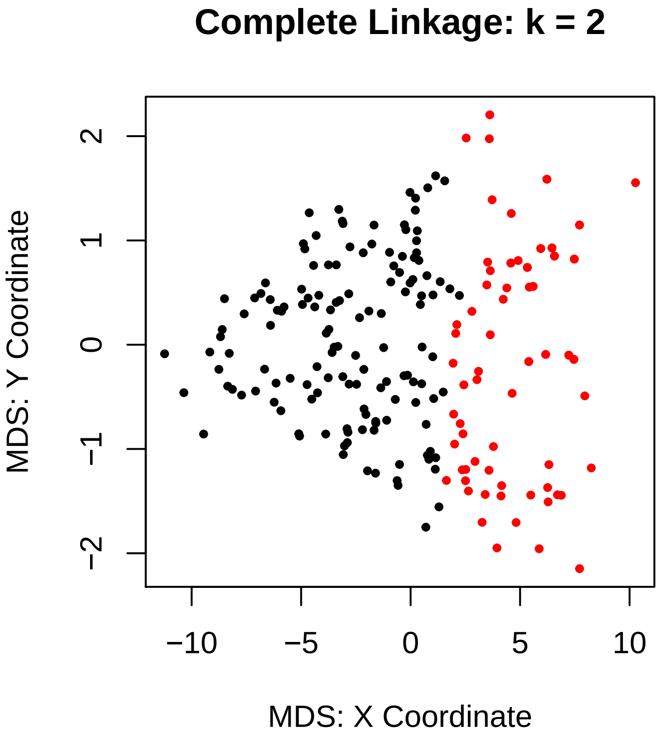


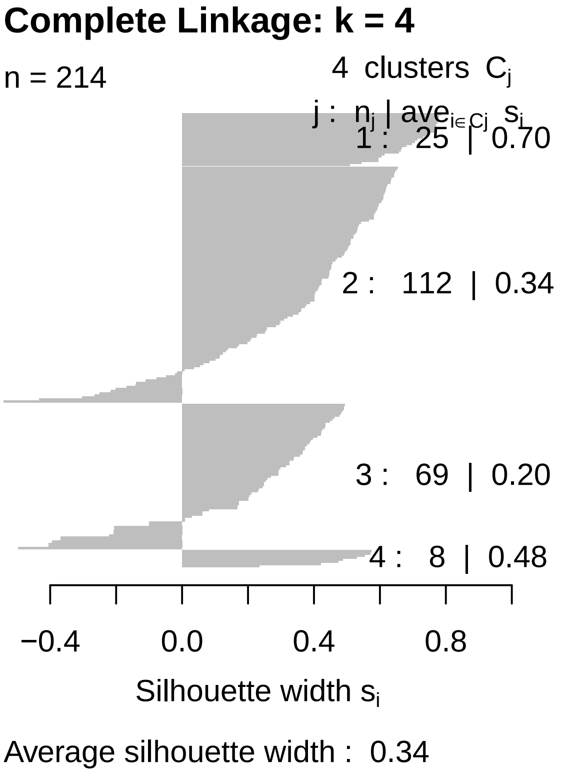


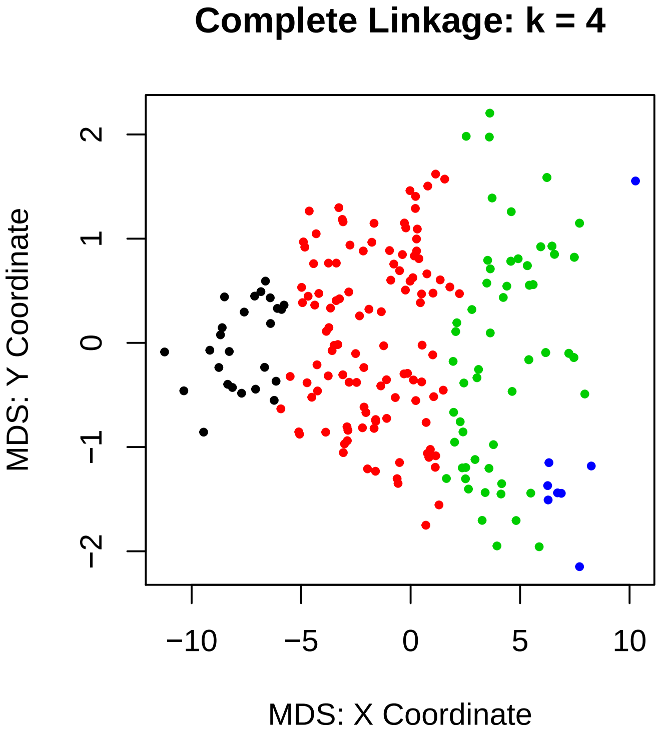
Overall, average linkage hierarchical clustering performed somewhat weakly. The graphs for k = 2 are not as evenly split as k = 2 for complete linkage, but the splitting is still reasonable unlike compared to single linkage. As k increased, the splitting of the clusters looks gradually similar to the splittings in complete linkage. Respectively for 2, 4, and 7 clusters the average silhouette lengths were 0.42, 0.44, and 0.33 so numerical evidence also agrees average linkage could be a weak option.

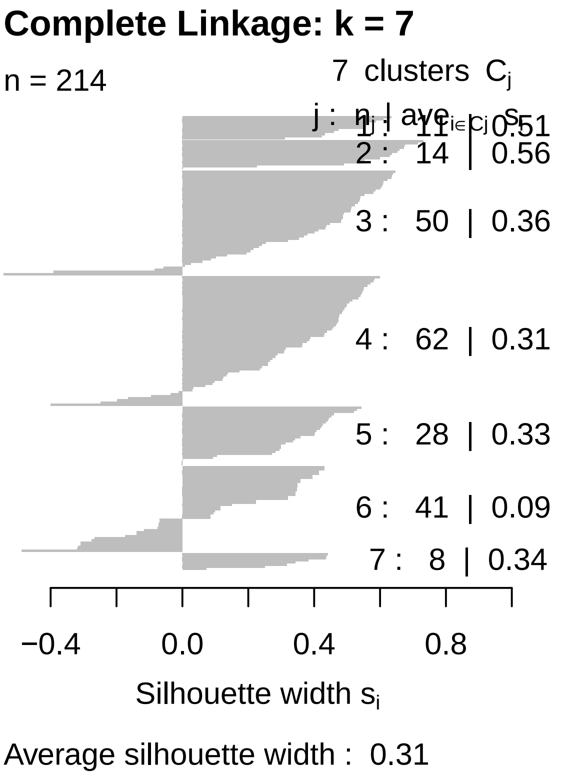
Complete Linkage

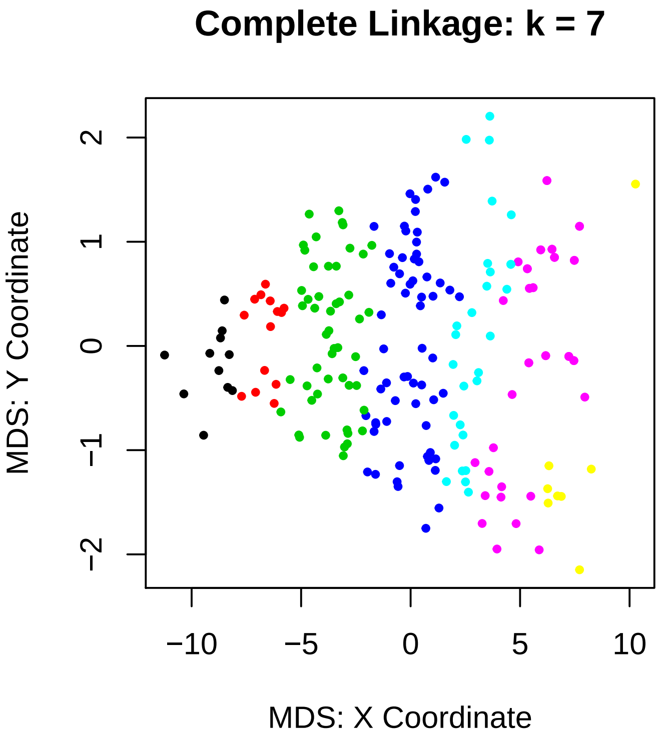






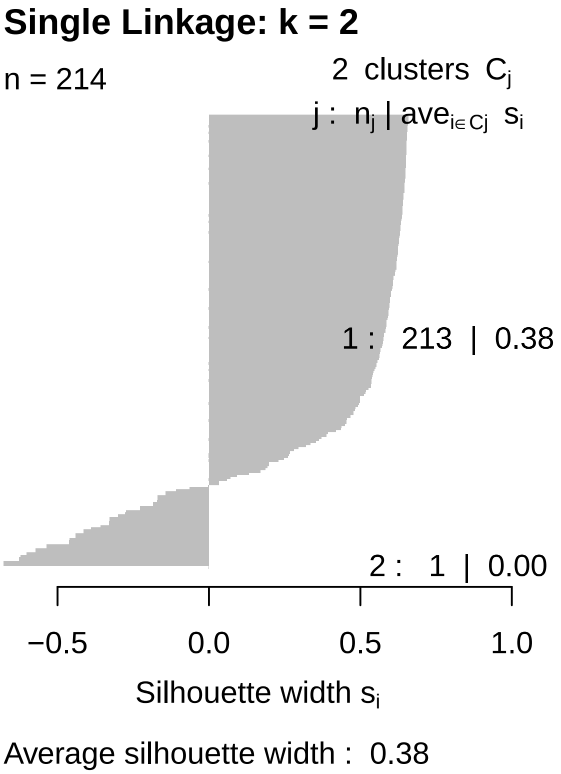


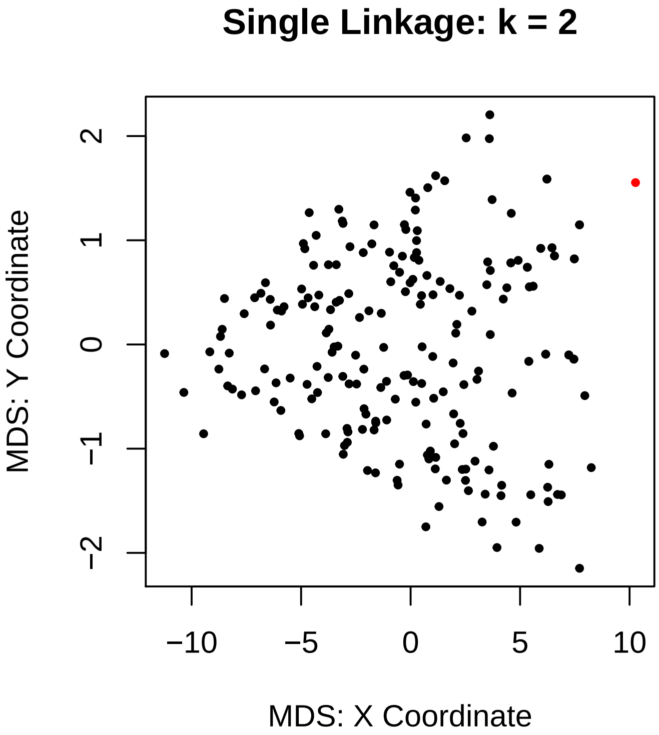


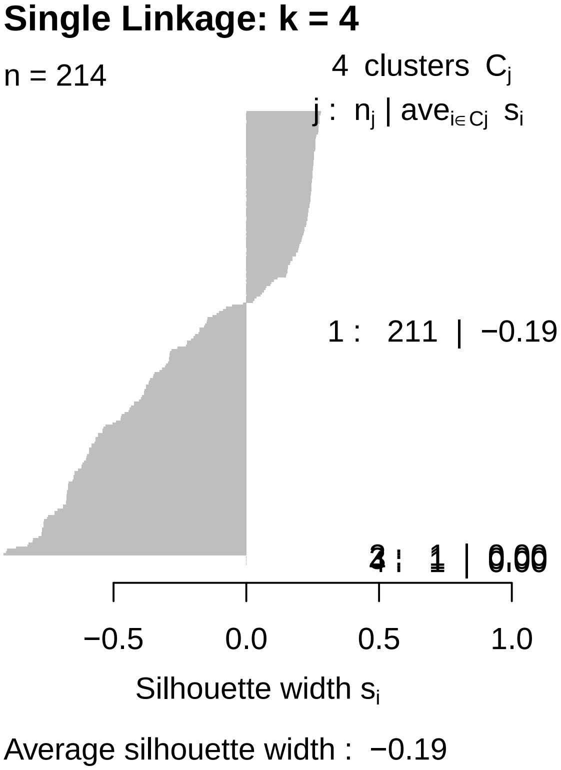


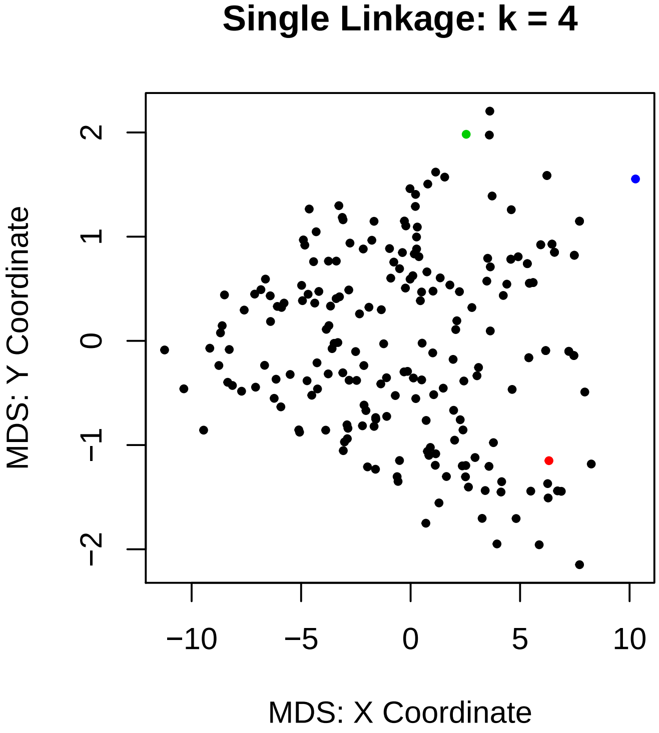
Overall, complete linkage hierarchical clustering performed moderately well. The graphs for k = 2 are fairly evenly split in cluster assignment. As k increased, the splitting of the clusters might bring an argument that the clustering is starting to become artifically excessive. Respectively for 2, 4, and 7 clusters the average silhouette lengths were 0.51, 0.34, and 0.31 so a small number of clusters could make complete linkage a convincingly reasonable procedure to use.

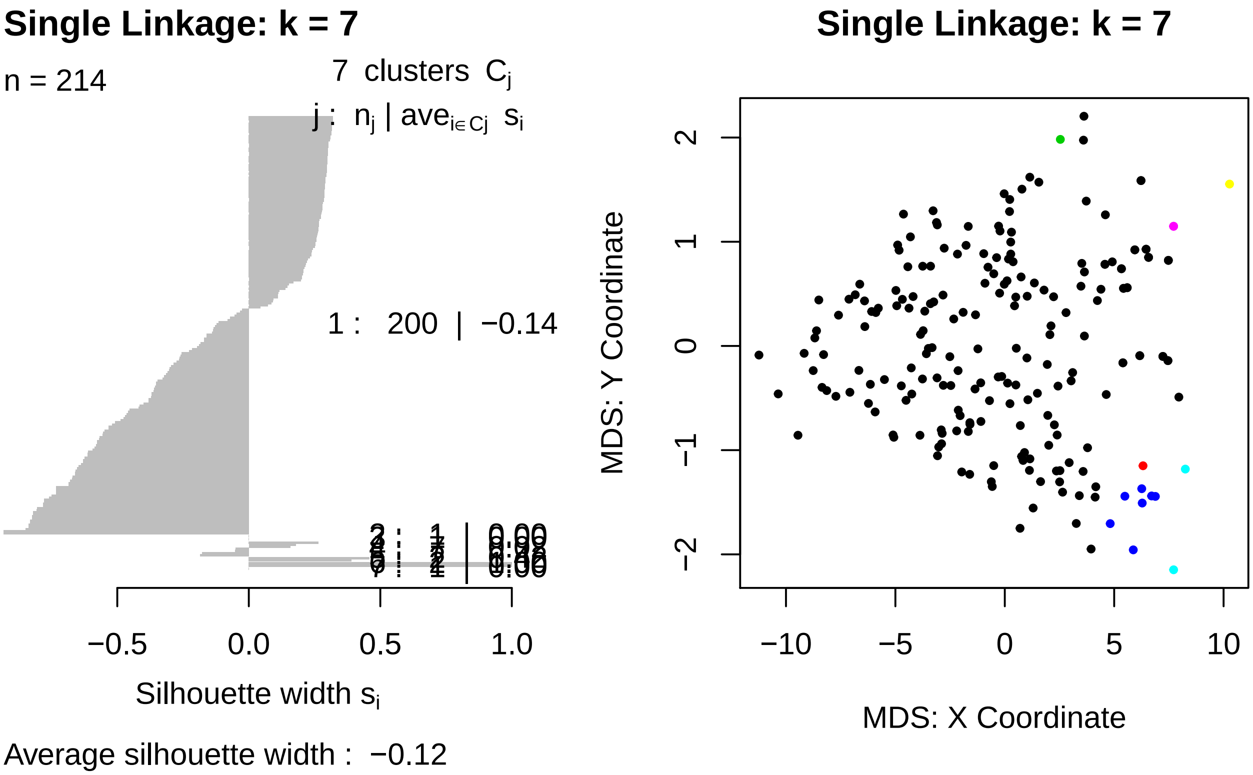
Single Linkage







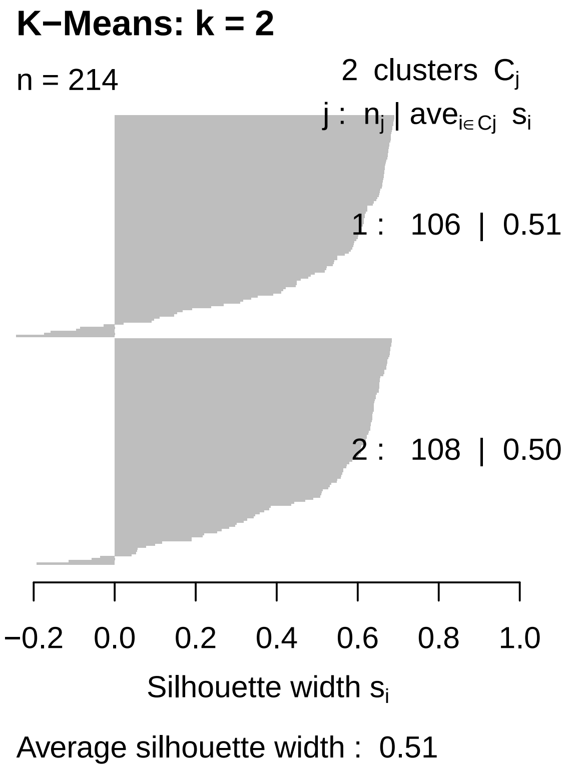


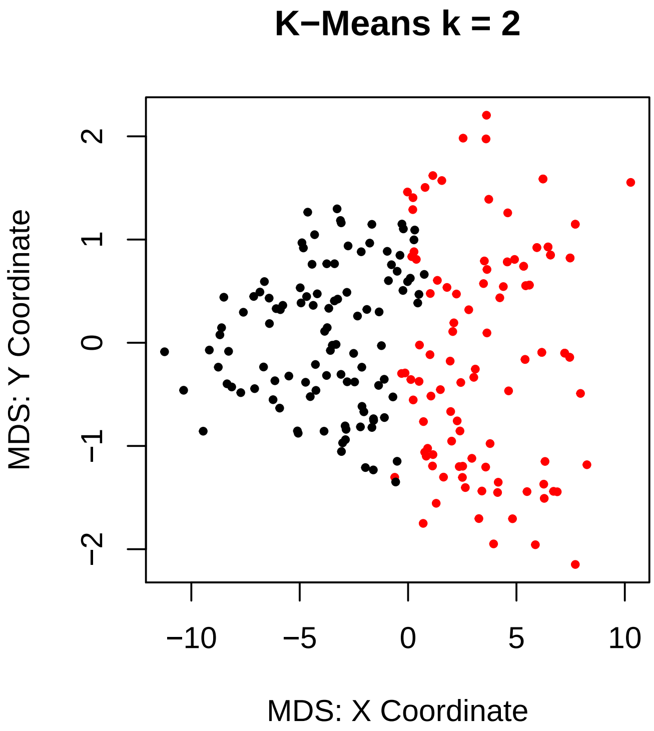


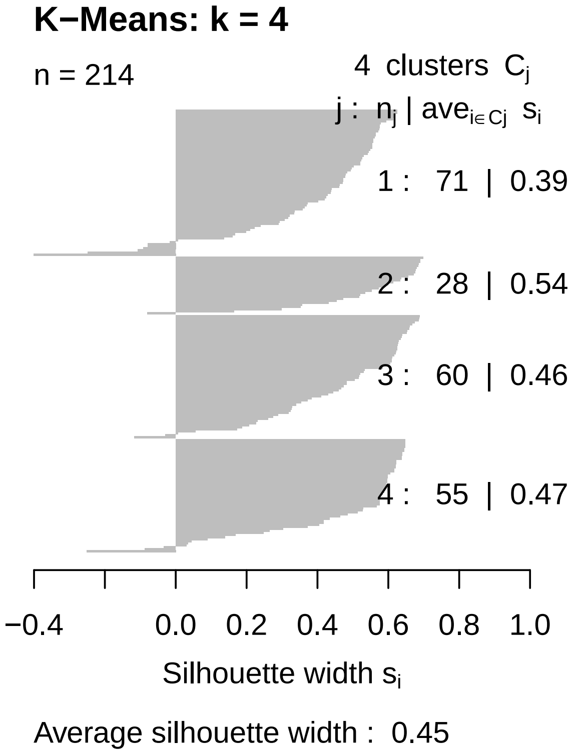
Single Linkage performed terribly, no question about that. Immediately the MDS plots show absurd and confusing clustering decisions so single linkage should not be a method to consider. Respectively for 2, 4, and 7 clusters the average silhouette lengths were 0.38, -0.19, and -0.12 so single linkage is horrible to a point that completely wrong clustering is very much possible.

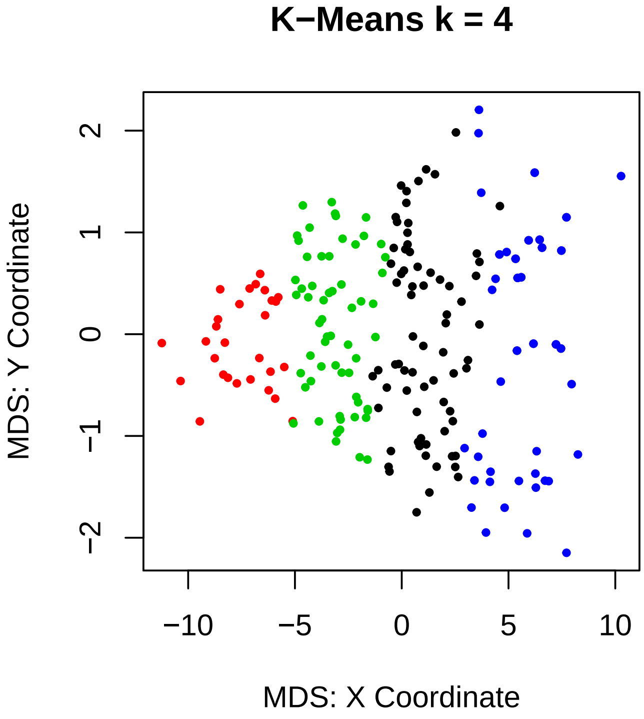
K-Means

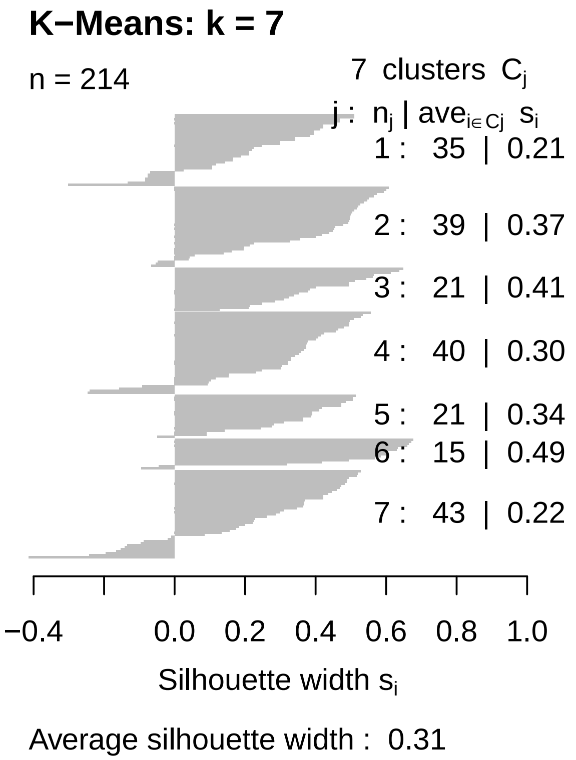
K-means (different from k-nearest neighbors which is a classification algorithm), is another clustering method based upon “means” within the clusters. More specifically, k-means clustering partitions the observations into k clusters where cluster membership is based on belonging to the cluster with the nearest mean. More computationally complex than hierarchical clustering, but can produce tighter clusters and can be computationally faster than hierarchical clusters with a large number of variables. As before, we may vary the number of desired k clusters and continuously evaluate the performance through more silhouette plots interpreted similarly as before. Dendrograms are not appropriate here since there is no “hierarchy” aspect.

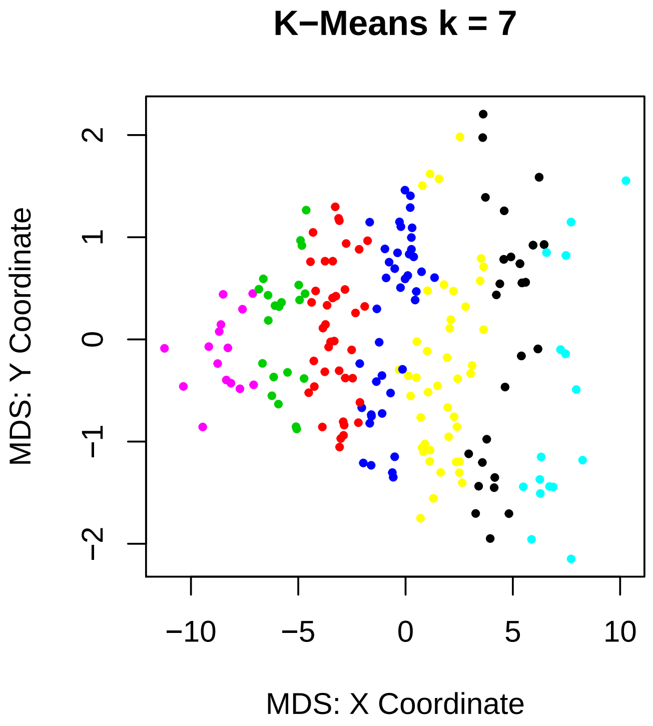

K−Means: k = 2
n = 214
2 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 106 | 0.51
2 : 108 | 0.50
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.51


K−Means k = 2
MDS: Y Coordinate
2
1
0
−1
−2
−10
−5
0
5
10
MDS: X Coordinate

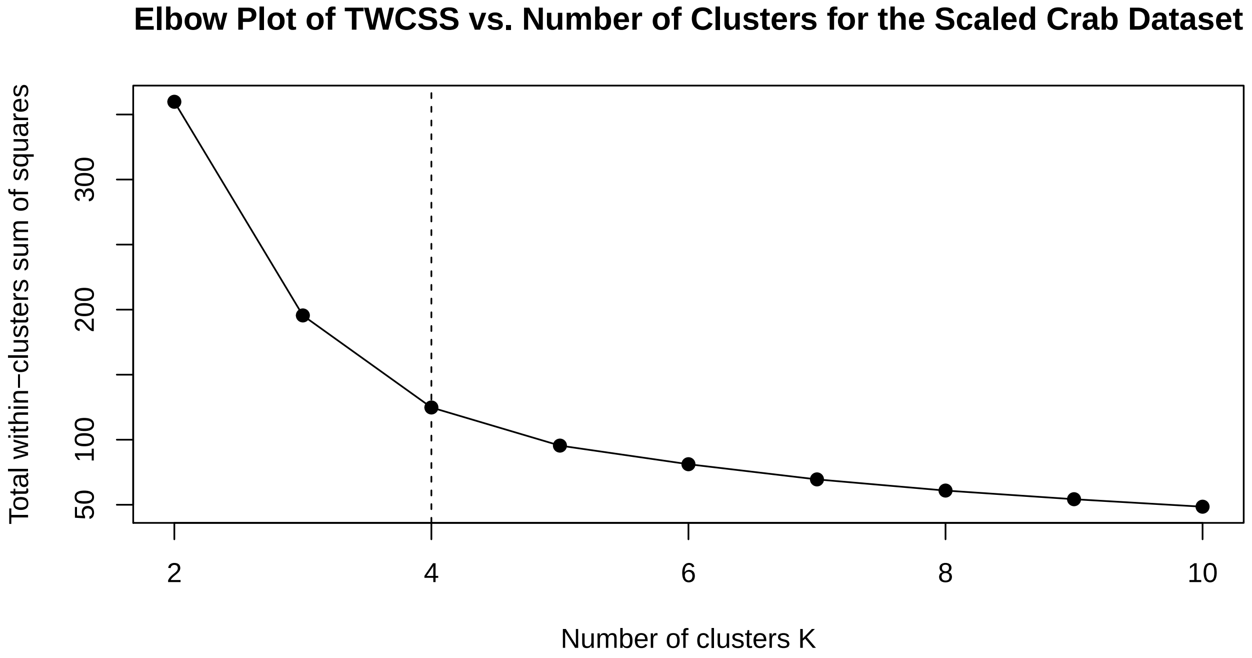

K−Means: k = 4
n = 214
4 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 71 | 0.39
2 : 28 | 0.54
3 : 60 | 0.46
4 : 55 | 0.47
−0.4
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.45


K−Means k = 4
MDS: Y Coordinate
2
1
0
−1
−2
−10
−5
0
5
10
MDS: X Coordinate



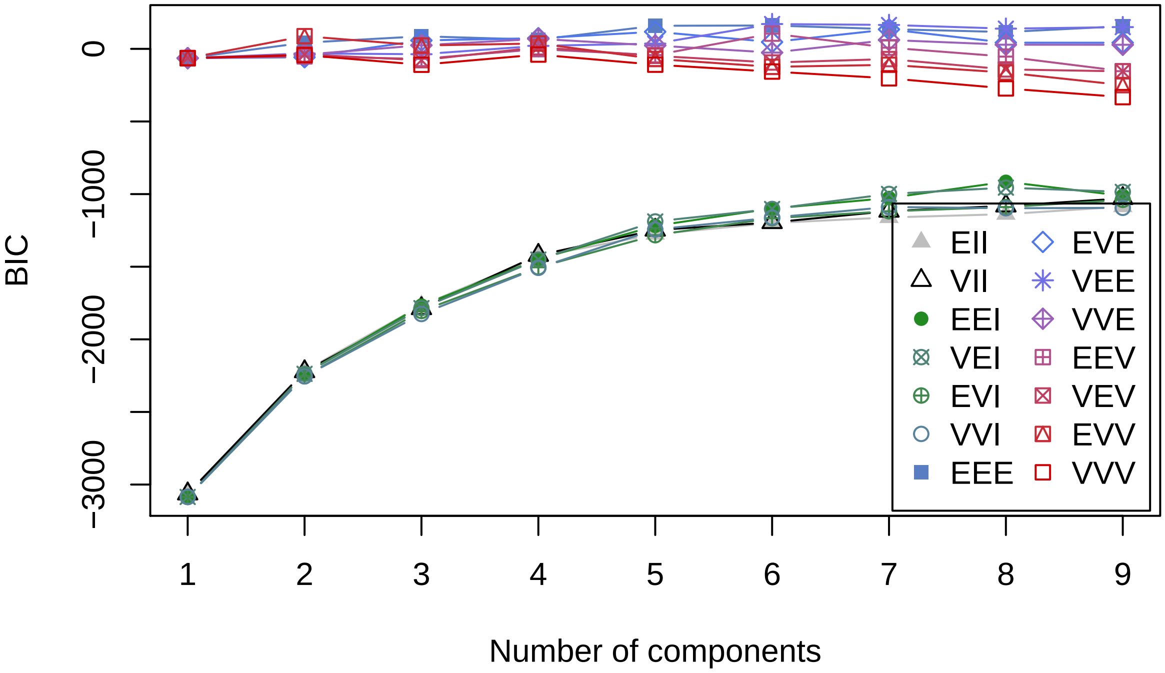


Compared to hierarchial clustering, K-Means had similarly positive performance as Complete Linkage with respective average silhouette lengths of 0.51, 0.45, and 0.31 at k = 2, 4, and 7. Viewing the clustering schematic in the MDS plot, they also look reasonably clustered and make the most sense at small k (less and less as k increases). Particularly a common issue is that the K number of clusters must be self-specified, so picking the optimal K can be a tricky process. We can consider an elbow plot of Total within-cluster sum of squares upon various K's, and seek an "elbow" that balances Total within-cluster sum of squares and number of clusters. Based on the elbow plot below, it looks like 4 (possibly 3) clusters is the best choice through the k-means process.



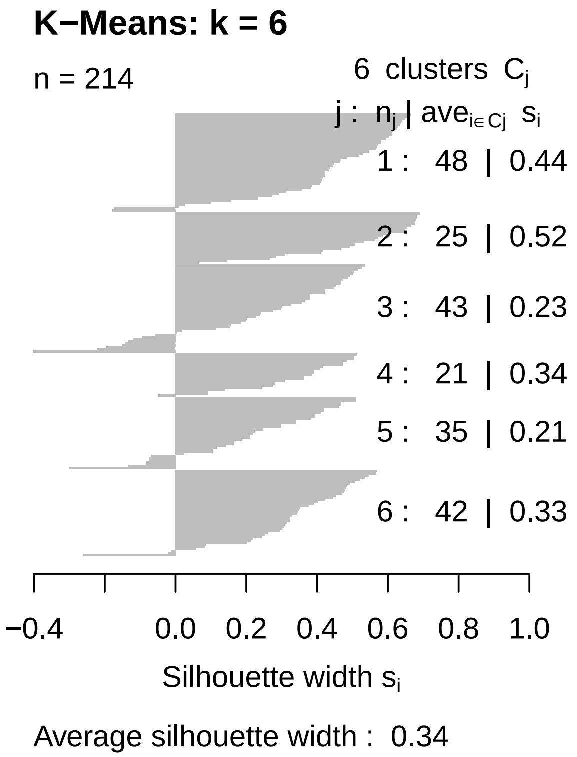
Mixture Modeling

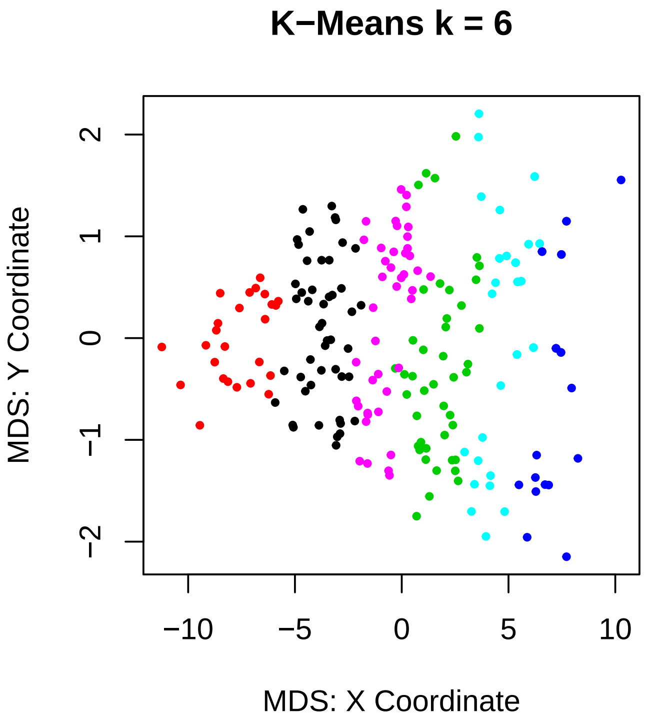
Finally, we might consider Mixture Modeling to tackle clustering problems. The idea of mixture modeling is to use a mixture of parametric distributions (such as Gaussian or Poisson distributions) to assist in modeling the clusters and optimize the model and data's fit. It may be fairly complex due to studying the nature of the mixture of the distributions, but does have some nice advantages; there is a lot more flexibility in specifying the model for the data and knowledge upon the distributions can provide additional information such as the cluster densities or relatable statistical inference for these parametric distributions. The MClust package will be extremely helpful in building a mixture model and assessing this cluster method's performance. The goal is to select the optimal model according to the number of clusters + largest BIC for EM initialized by hierarchical clustering for parameterized Gaussian mixture models.

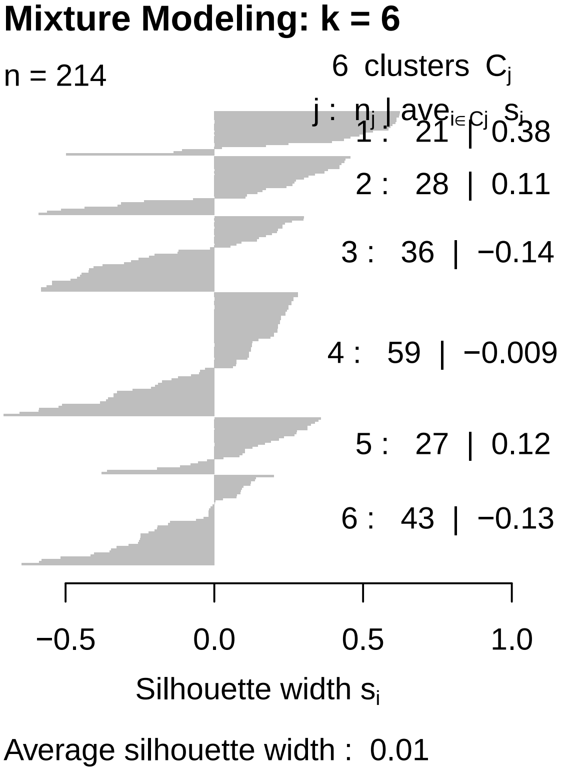


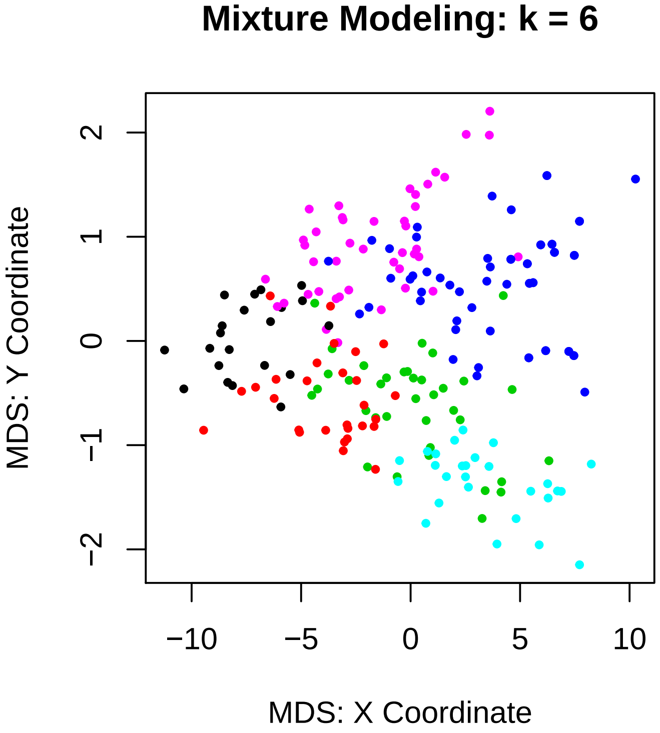
The optimal model chosen was outputted to be "VEE", meaning the optimal mixture model was one based on ellipsoidal, equal shape and orientation of the clusters. This is actually somewhat "verifiable" from the original MDS plot where the clustering by Sex was ellipsoidal, equal shape and orientation. As shown from the plot, the optimal mixture model is a close competition between VEE, EEE, and VVE (the latter two being "ellipsoidal, equal volume, shape, and orientation" and "ellipsoidal, equal orientation", respectively). VEE yielded BICs between -63.80 and 170.96657 for number of components being between 1 and 9. Perusing through the BICs for different given components (# of clusters) in the VEE model, the number of clusters seem optimal at 6 clusters with a BIC of 170.96657.

In short summary, k-means indicated 4 clusters was the optimal number of clusters while mixture modeling indicated 6 clusters was the optimal number of clusters. We proceed assuming 6 clusters is the chosen number of clusters, and below is the clustering differences between K-means and mixture modeling in the style of the crabs data projected into MDS plots along with the silhouette plots. K-means still outperformed Mixture Modeling with an average silhouette length of 0.34 over the VEE mixture modeling average silhouette length of 0.01, with the MDS plots providing visual evidence of this discrepancy.





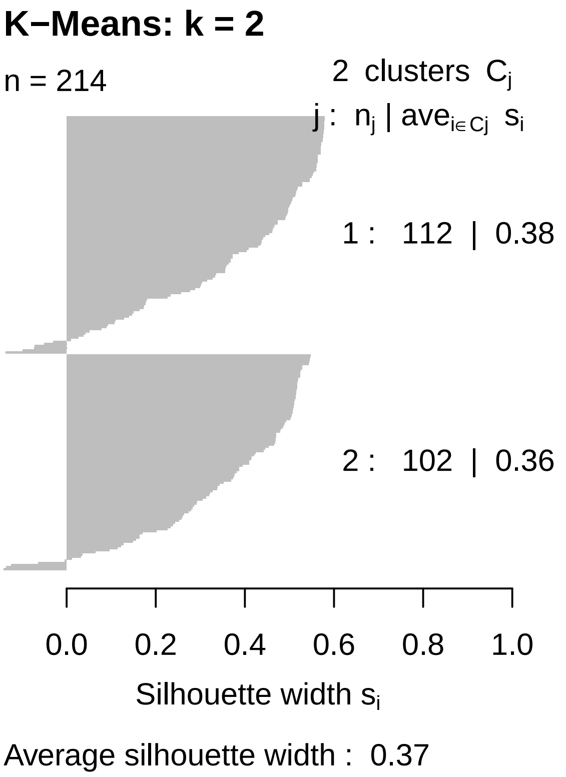


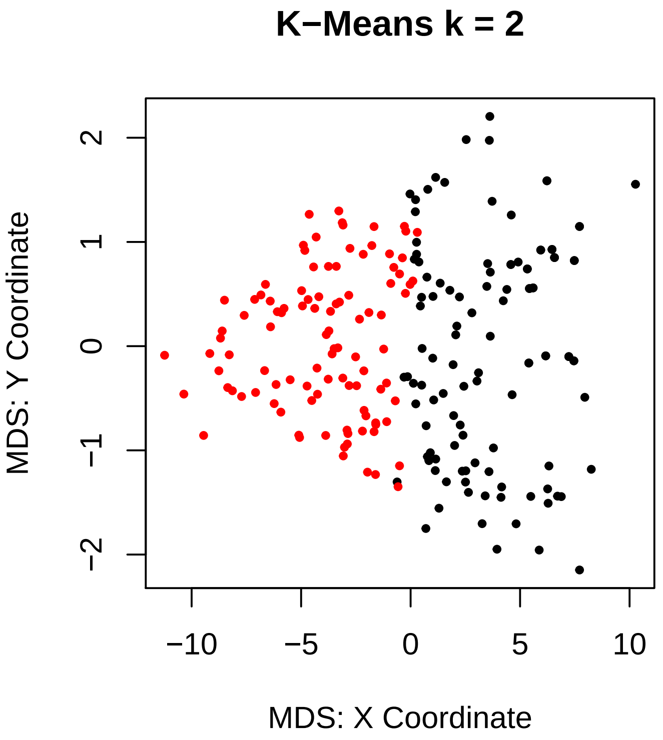


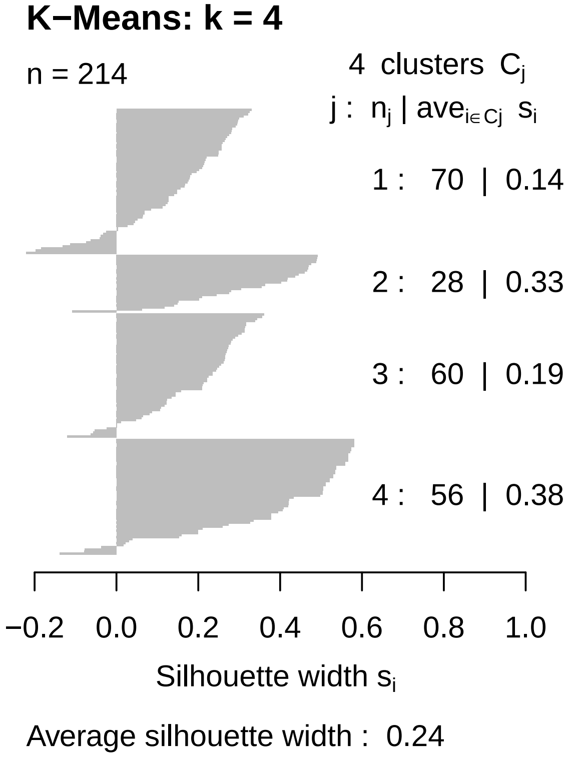
– Approach 2: –

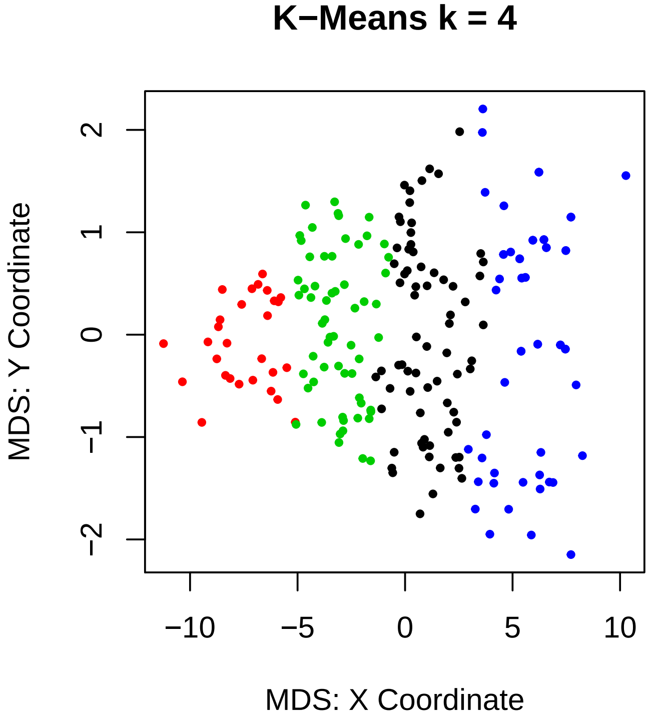
Now consider clustering along with the categorical variables: Species and Sex. Comparisons between K-means and mixture modeling will be emphasized in the analysis below. Most of the approach is the same, except we should be using a Gower distance matrix due to the fact that we have mixed types of variables (categorical and numerical)

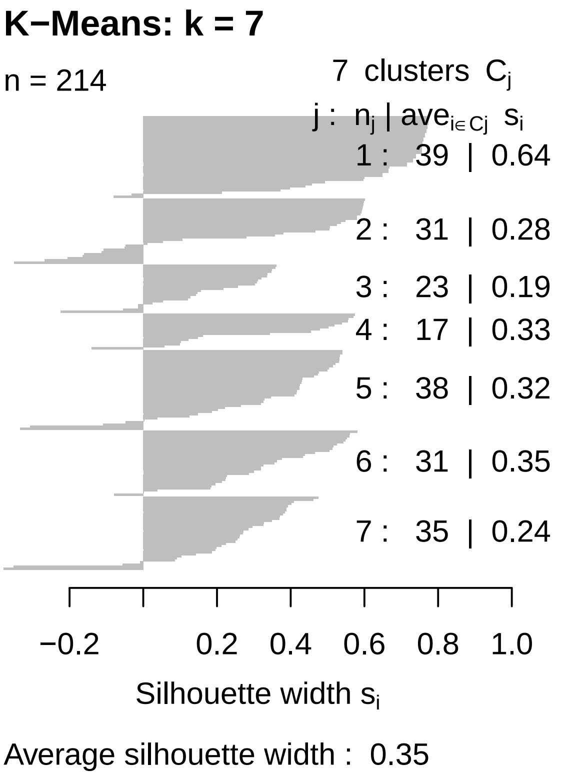
K-Means

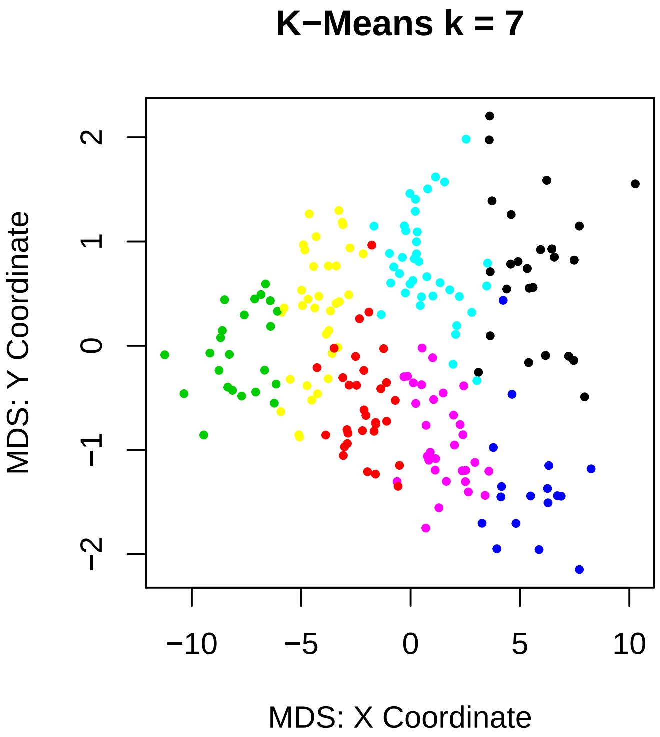




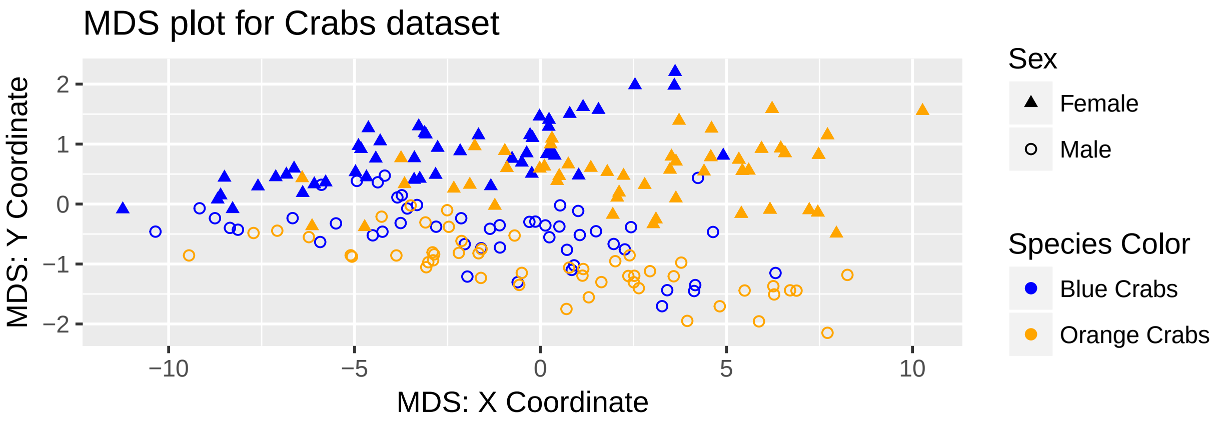


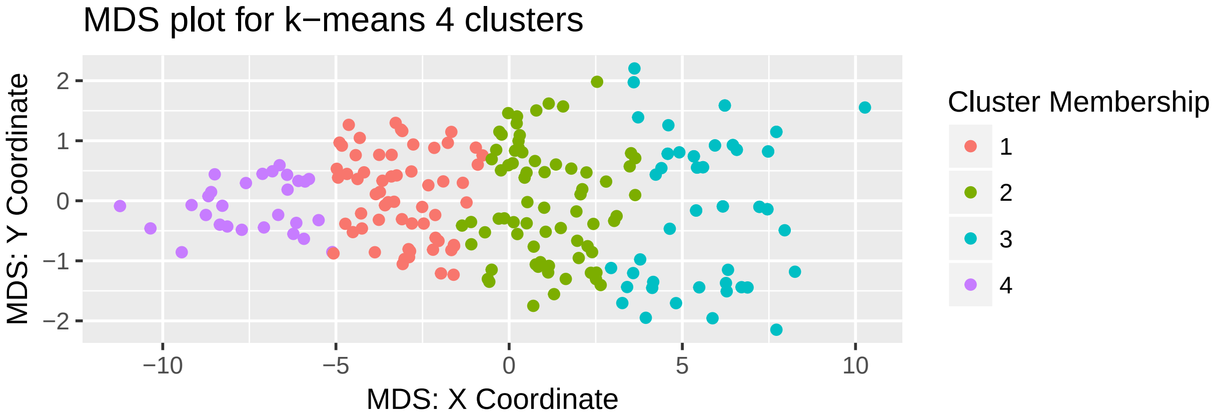






At k = 2, 4, 7, the average silhouette widths respectively were 0.37, 0.24, 0.31. Recall from earlier that only considering the scaled numerical features yielded average silhouette widths of 0.51, 0.45, and 0.31 for K-means; there was some drop in clustering performance when including the categorical variables. As a rough measurement of how well this clustering matched the pre-defined categories, we might consider that there are 4 possible combinations of crab species + sex. That is, we have combinations of orange males, orange females, blue males, blue females. A side-by-side plot of the original MDS projection versus K-means clustering estimated labels is provided below, and we can see that

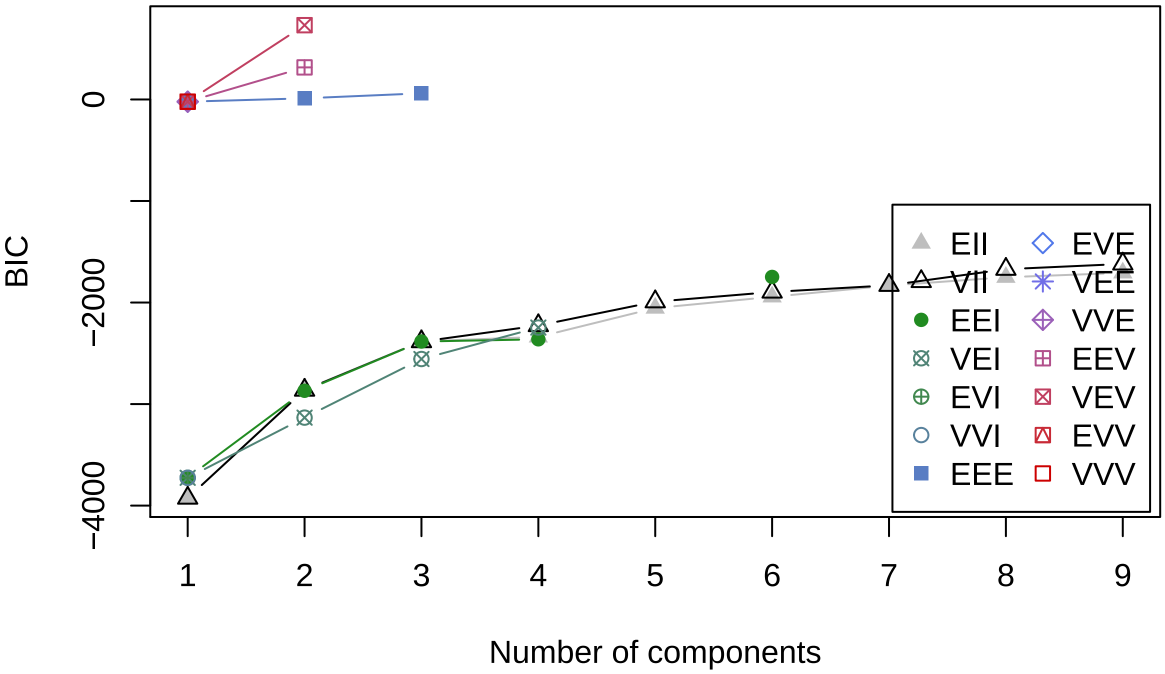

MDS plot for Crabs dataset

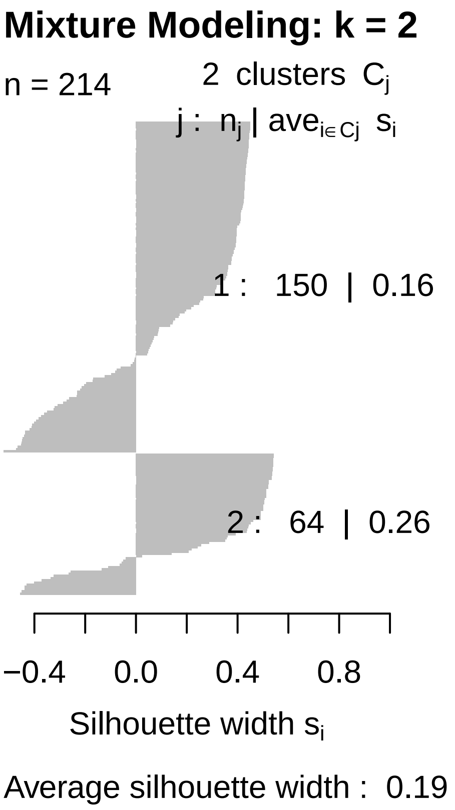

MDS plot for k−means 4 clusters

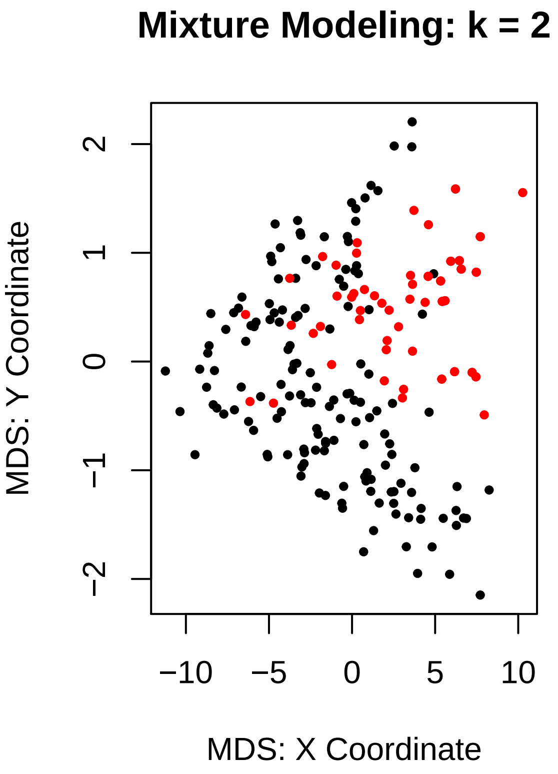
Unfortunately the clustering looks to be quite poor. We point out that females and males look to be horizontally divided in the MDS plot, but most divides are vertical in the attempted clustering through k-means. Also there is a lot of overlap between blue and orange crabs, but k-means doesn't recognize this distinguishing feature which can lead to wrong classifications.

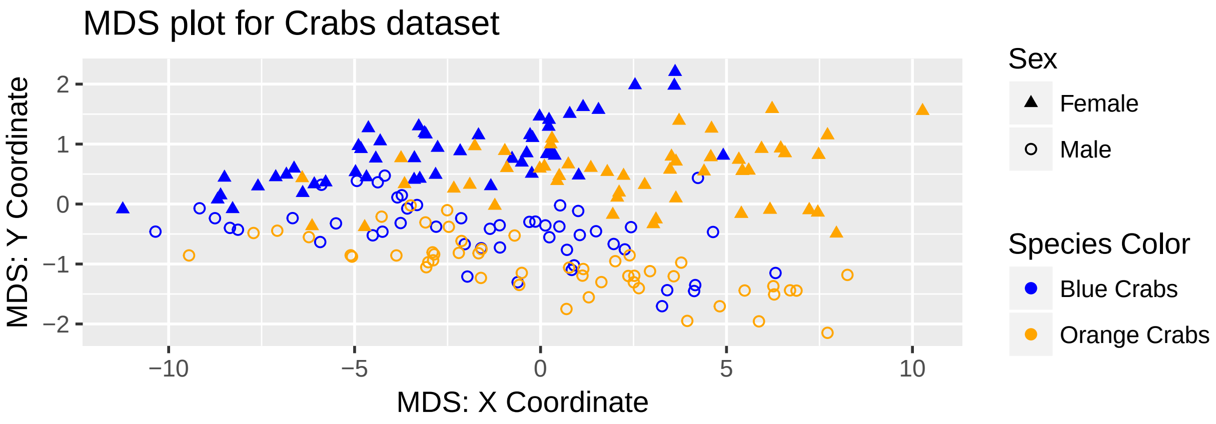
Mixture Modeling

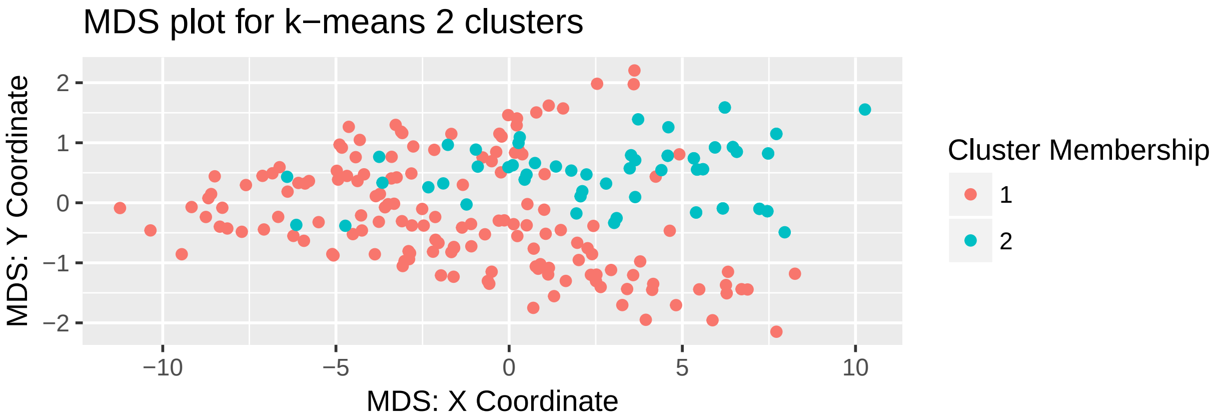
Running the same Mixture Modeling procedure on the scaled crabs dataset along with Species and Sex categorical features, it appears that the VEV mixture models with 2 clusters was the ideal model chalking in a BIC of 731.61327. VEV is a mixture model characterized as ellipsoidal, equal shape for the clusters. Once again, the silhouette plot presents an average silhouette length of 0.19 indicating supposedly a poor clustering fit. Translating to the MDS plot with the Mixture Modeling clusters, the 2 clusters do capture there being some overlap and surprisingly the cluster 2 group membership from the mixture modeling somewhat is close to the same pattern of orange female crabs in the original data MDS plot. This is visually clear in the final plot of the original data in MDS plot versus the Mixture modeling MDS plot for 2 clusters, so one could make an argument that mixture clustering for 2 clusters can point out a difference between orange female crabs against any other type of crabs.


BIC
0
−2000
−4000
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
EVE
VEE
VVE
EEV
VEV
EVV
VVV


Mixture Modeling: k = 2
n = 214
2 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 150 | 0.16
2 : 64 | 0.26
−0.4
0.0
0.4
0.8
Silhouette width s_i
Average silhouette width : 0.19


Mixture Modeling: k = 2
MDS: Y Coordinate
2
1
0
−1
−2
−10
−5
0
5
10
MDS: X Coordinate


MDS plot for Crabs dataset
MDS: Y Coordinate
2
1
0
−1
−2
−10
−5
0
5
10
MDS: X Coordinate
Sex
▲ Female
○ Male
Species Color
● Blue Crabs
● Orange Crabs


MDS plot for k−means 2 clusters
MDS: Y Coordinate
2
1
0
−1
−2
−10
−5
0
5
10
MDS: X Coordinate
Cluster Membership
● 1
● 2

Appendix: R Code

```r
# Preliminary Packages
library(dplyr)
library(ggplot2)
library(reshape2)
library(grid)
library(gridExtra)
library(cluster)
library(mclust)
library(ggdendro)
library(mixtools)
library(knitr)

# Import Data, and treat Species & Sex as factors
crabs = read.table("~/Desktop/School/Stats503/Datasets/crabs.txt", header = T)

crabs$Species = as.factor(crabs$Species)
crabs$Sex = as.factor(crabs$Sex)

# Scale Data
scalecrabs = crabs
scalecrabs[,3:7] = scale(crabs[,3:7])

# Features table
columns = c("Species", "Sex", "FL", "RW", "CL", "CW", "BD")
desc = c("1 = Blue crabs, 2 = Orange crabs", "1 = Male, 2 = Female", "Frontal Lobe Size (mm)", "Rear Wi
table1 = cbind(columns, desc)
colnames(table1) = c("Name of Crab Feature", "Feature Description")
# Display table
kable(table1)

# Beginning of the original data
head(crabs, n = 3)

# Beginning of the scaled data
head(scalecrabs, n = 3)

# Chunk of original and scaled last 5 numerical features, for easy reference
crabnums = crabs[,3:7]
scaledcrabnums = scalecrabs[,3:7]

# Multidimensional scaling, for 2-D visualization of the clustering
distmat = dist(crabs[,3:7]) # euclidean distances between the rows
dim2points = cmdscale(distmat, eig=TRUE, k=2) # k is the number of dim

# Extract 2-D coordinates
mdsx = dim2points$points[,1]
mdsy = dim2points$points[,2]

# Append to end of crabs dataset, and scaled dataset
newcrabs = cbind(crabs, mdsx, mdsy)
newscalecrabs = cbind(scalecrabs, mdsx, mdsy)
```

```r
mdsplot = ggplot(newcrabs, aes(x = mdsx, y = mdsy, color = ifelse(Species == 1, "Blue Crabs", "Orange Cr
  geom_point() +
  scale_color_manual(name = "Species Color", values = c("blue","orange")) +
  scale_shape_manual(name = "Sex", values = c(17, 1)) +
  labs(x = "MDS: X Coordinate", y = "MDS: Y Coordinate", title = "MDS plot for Crabs dataset")

mdsplot

# agnes function computes agglomerative hierarchical clustering of the dataset.
avglink =  agnes(scaledcrabnums, diss=FALSE, method='average') # Average Linkage
complink = agnes(scaledcrabnums, diss=FALSE, method='complete') # Complete linkage
singlink = agnes(scaledcrabnums, diss=FALSE, method='single') # Single linkage

# Dendrograms for each 3 distance functions
den_avg = ggdendrogram(as.dendrogram(avglink),
                       leaf_labels = FALSE,
                       labels = FALSE) +
  ggtitle("Average Clustering")
den_comp = ggdendrogram(as.dendrogram(complink),
                        leaf_labels=FALSE,
                        labels=FALSE) +
  ggtitle("Complete Linkage")
den_sing = ggdendrogram(as.dendrogram(singlink),
                        leaf_labels=FALSE,
                        labels=FALSE) +
  ggtitle("Single Linkage")

grid.arrange(den_avg,den_comp,den_sing,
             ncol = 2, nrow = 2,
             top=textGrob("Dendograms by distance function",
                          gp = gpar(fontsize=12,font=3)))

# Side-by-side silhouette plot and the clusters shown in the MDS plot for average linkage
avglinkageplots = function(a){
  # Silhouette plot functions for a specific K parameter
  silhouette_avg = function(i){
    silhouette(cutree(avglink, k=i), distmat)
  }
  par(mfrow=c(1,2))
  # Silhouette plot
  plot(silhouette_avg(a), main = paste("Average Linkage:","k", "=", a, sep = " "))
  # MDS cluster plot
  tempdf = cbind(newscalecrabs, silhouette_avg(a)[,1])
  plot(x = tempdf[,8], y = tempdf[,9], col = tempdf[,10], main = paste("Average Linkage:","k", "=", a, s
       pch = 19, cex = 0.5, xlab = "MDS: X Coordinate", ylab = "MDS: Y Coordinate")
  par(mfrow=c(1,1))
}

# Side-by-side silhouette plot and the clusters shown in the MDS plot for complete linkage
complinkageplots = function(a){
  # Silhouette plot function for a specific K parameter
  silhouette_complete = function(i){
    silhouette(cutree(complink, k=i), distmat)
  }
  par(mfrow=c(1,2))
  # Silhouette plot
  plot(silhouette_complete(a), main = paste("Complete Linkage:","k", "=", a, sep = " "))
  # MDS cluster plot
  tempdf = cbind(newscalecrabs, silhouette_complete(a)[,1])
  plot(x = tempdf[,8], y = tempdf[,9], col = tempdf[,10], main = paste("Complete Linkage:","k", "=", a,
       pch = 19, cex = 0.5, xlab = "MDS: X Coordinate", ylab = "MDS: Y Coordinate")
  par(mfrow=c(1,1))
}

# Side-by-side silhouette plot and the clusters shown in the MDS plot for single linkage
singlinkageplots = function(a){
  # Silhouette plot function for a specific K parameter
  silhouette_single = function(i) {
    silhouette(cutree(singlink, k=i), distmat)
  }
  par(mfrow=c(1,2))
  # Silhouette plot
  plot(silhouette_single(a), main = paste("Single Linkage:","k", "=", a, sep = " "))
  # MDS cluster plot
  tempdf = cbind(newscalecrabs, silhouette_single(a)[,1])
  plot(x = tempdf[,8], y = tempdf[,9], col = tempdf[,10], main = paste("Single Linkage:","k", "=", a, se
       pch = 19, cex = 0.5, xlab = "MDS: X Coordinate", ylab = "MDS: Y Coordinate")
  par(mfrow=c(1,1))
}

# Average Linkage
avglinkageplots(2)
avglinkageplots(4)
avglinkageplots(7)

# Complete Linkage
complinkageplots(2)
complinkageplots(4)
complinkageplots(7)

# Single Linkage
singlinkageplots(2)
singlinkageplots(4)
singlinkageplots(7)

# Side-by-side silhouette plot and the clusters shown in the MDS plot for k-means
kmeanplots = function(a){
  kmeanoutput = kmeans(scaledcrabnums, centers = a, nstart = 25)

  par(mfrow=c(1,2))
  # Silhouette plot
  plot(silhouette(kmeanoutput$cluster, distmat), main = paste("K-Means:","k", "=", a, sep = " "))
  # MDS cluster plot
  tempdf = cbind(newscalecrabs, kmeanoutput$cluster)
  plot(x = tempdf[,8], y = tempdf[,9], col = tempdf[,10], main = paste("K-Means","k", "=", a, sep = " "
       pch = 19, cex = 0.5, xlab = "MDS: X Coordinate", ylab = "MDS: Y Coordinate")
  par(mfrow=c(1,1))
}

# K-Means
kmeanplots(2)
kmeanplots(4)
kmeanplots(7)

# Elbow Plot
wss <- sapply(2:10,
        function(k){kmeans(scaledcrabnums, k, nstart=25)$tot.withinss})

plot(2:10, wss,
       type="o", pch = 19,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares", main = "Elbow Plot of TWCSS vs. Number of Clusters
abline(v = 4, lty =2)

# Create Mixture Model
mixclust = Mclust(scaledcrabnums)

# Which mixture model was used, and the BICs
#mixclust$modelName
#mixclust$BIC
# BIC Plot
plot(mixclust, what="BIC", main = "BIC plot for various Mixture Models")

mixtureplots = function(a){
  par(mfrow=c(1,2))
  # Silhouette plot
  plot(silhouette(mixclust$classification, distmat), main = paste("Mixture Modeling:","k", "=", a, sep
  # MDS cluster plot
  tempdf = cbind(newscalecrabs, mixclust$classification)
  plot(x = tempdf[,8], y = tempdf[,9], col = tempdf[,10], main = paste("Mixture Modeling:","k", "=", a,
       pch = 19, cex = 0.5, xlab = "MDS: X Coordinate", ylab = "MDS: Y Coordinate")
  par(mfrow=c(1,1))
}

# At 6 clusters
kmeanplots(6)
mixtureplots(6)

# Gower distance matrix
gowerdistmat = daisy(scalecrabs, metric = "gower")

kmeanplots2 = function(a){
  kmeanoutput = kmeans(scalecrabs, centers = a, nstart = 25)

  par(mfrow=c(1,2))
  # Silhouette plot
  plot(silhouette(kmeanoutput$cluster, gowerdistmat), main = paste("K-Means:","k", "=", a, sep = " "))
  # MDS cluster plot
  tempdf = cbind(newscalecrabs, kmeanoutput$cluster)
  plot(x = tempdf[,8], y = tempdf[,9], col = tempdf[,10], main = paste("K-Means","k", "=", a, sep = " "
       pch = 19, cex = 0.5, xlab = "MDS: X Coordinate", ylab = "MDS: Y Coordinate")
  par(mfrow=c(1,1))
}

kmeanplots2(2)
kmeanplots2(4)
kmeanplots2(7)

kmeancombineplots = function(a){
  kmeanoutput = kmeans(scalecrabs, centers = a, nstart = 25)
  tempdf = cbind(newscalecrabs, kmeanoutput$cluster)
  kgg = ggplot(tempdf, aes(x = tempdf[,8], y = tempdf[,9], col = factor(tempdf[,10]))) + geom_point() +
    labs(x = "MDS: X Coordinate", y = "MDS: Y Coordinate", title = "MDS plot for k-means 4 clusters", c

  grid.arrange(mdsplot, kgg, ncol = 1)
}

kmeancombineplots(4)

# Create Mixture Model
mixclust2 = Mclust(scalecrabs)

# Which mixture model was used, and the BICs
#mixclust2$modelName
#mixclust2$BIC
# BIC Plot
plot(mixclust2, what="BIC", main = "BIC plot for various Mixture Models")

mixturecombineplots = function(a){

  tempdf = cbind(newscalecrabs, mixclust2$classification)
  kgg = ggplot(tempdf, aes(x = tempdf[,8], y = tempdf[,9], col = factor(tempdf[,10]))) + geom_point() +
    labs(x = "MDS: X Coordinate", y = "MDS: Y Coordinate", title = "MDS plot for k-means 2 clusters", c

  grid.arrange(mdsplot, kgg, ncol = 1)
}

mixturecombineplots(4)
```