Q1. Agglomerative Clustering

Salman

2024-03-18

```r
if (!requireNamespace("datasetsICR", quietly = TRUE)) {
  install.packages("datasetsICR")
}
library(datasetsICR)


data("seeds")




str(seeds)
```

```
## 'data.frame':    210 obs. of  8 variables:
##  $ area                   : num  15.3 14.9 14.3 13.8 16.1 ...
##  $ perimeter              : num  14.8 14.6 14.1 13.9 15 ...
##  $ compactness            : num  0.871 0.881 0.905 0.895 0.903 ...
##  $ length of kernel       : num  5.76 5.55 5.29 5.32 5.66 ...
##  $ width of kernel        : num  3.31 3.33 3.34 3.38 3.56 ...
##  $ asymmetry coefficient  : num  2.22 1.02 2.7 2.26 1.35 ...
##  $ length of kernel groove: num  5.22 4.96 4.83 4.8 5.17 ...
##  $ variety                : Factor w/ 3 levels "Kama","Rosa",..: 1 1 1 1 1
1 1 1 1 1 ...
```

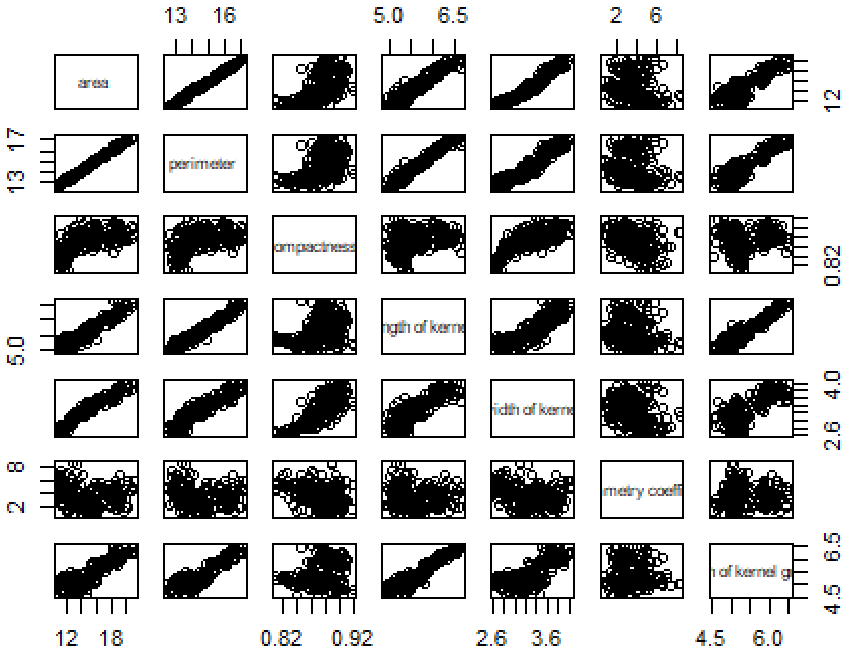
```r
summary(seeds)
```

```
##       area         perimeter      compactness     length of kernel
##  Min.   :10.59   Min.   :12.41   Min.   :0.8081   Min.   :4.899   
##  1st Qu.:12.27   1st Qu.:13.45   1st Qu.:0.8569   1st Qu.:5.262   
##  Median :14.36   Median :14.32   Median :0.8734   Median :5.524   
##  Mean   :14.85   Mean   :14.56   Mean   :0.8710   Mean   :5.629   
##  3rd Qu.:17.30   3rd Qu.:15.71   3rd Qu.:0.8878   3rd Qu.:5.980   
##  Max.   :21.18   Max.   :17.25   Max.   :0.9183   Max.   :6.675   
##  width of kernel asymmetry coefficient length of kernel groove     variety  
##  Min.   :2.630   Min.   :0.7651        Min.   :4.519           Kama    :70  
##  1st Qu.:2.944   1st Qu.:2.5615        1st Qu.:5.045           Rosa    :70  
##  Median :3.237   Median :3.5990        Median :5.223           Canadian:70  
##  Mean   :3.259   Mean   :3.7002        Mean   :5.408                        
##  3rd Qu.:3.562   3rd Qu.:4.7687        3rd Qu.:5.877                        
##  Max.   :4.033   Max.   :8.4560        Max.   :6.550                        
```

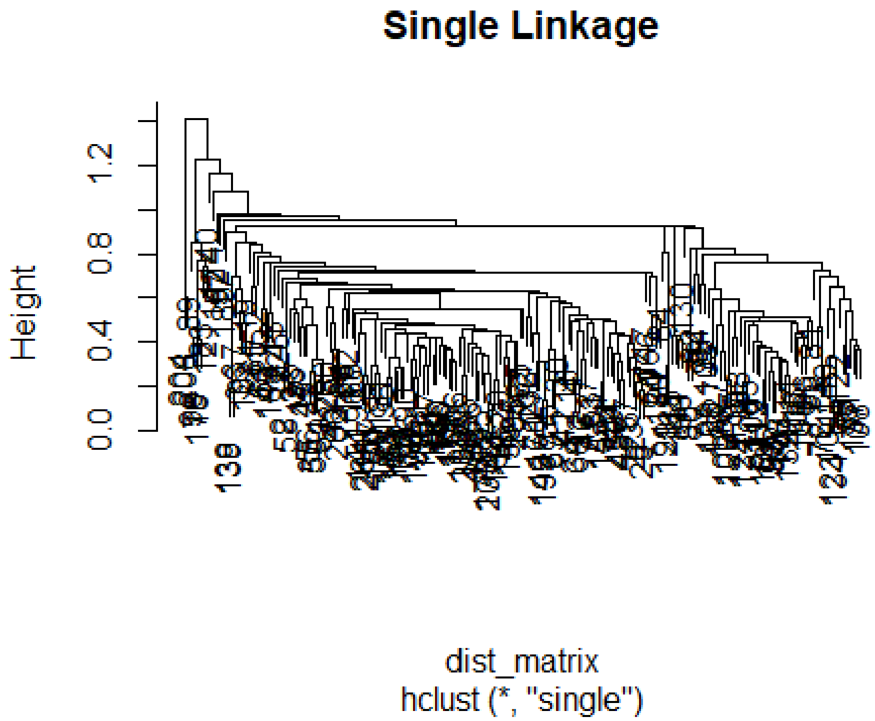
```r
cor_matrix <- cor(seeds[,1:7])
print(cor_matrix)
```

```
##                               area  perimeter compactness length of kernel
## area                     1.0000000  0.9943409   0.6082884        0.9499854
## perimeter                0.9943409  1.0000000   0.5292436        0.9724223
## compactness              0.6082884  0.5292436   1.0000000        0.3679151
## length of kernel         0.9499854  0.9724223   0.3679151        1.0000000
## width of kernel          0.9707706  0.9448294   0.7616345        0.8604149
## asymmetry coefficient   -0.2295723 -0.2173404  -0.3314709       -0.1715624
## length of kernel groove  0.8636927  0.8907839   0.2268248        0.9328061
##                         width of kernel asymmetry coefficient
## area                          0.9707706           -0.22957233
## perimeter                     0.9448294           -0.21734037
## compactness                   0.7616345           -0.33147087
## length of kernel              0.8604149           -0.17156243
## width of kernel               1.0000000           -0.25803655
## asymmetry coefficient        -0.2580365            1.00000000
## length of kernel groove       0.7491315           -0.01107902
##                         length of kernel groove
## area                                 0.86369275
## perimeter                            0.89078390
## compactness                          0.22682482
## length of kernel                     0.93280609
## width of kernel                      0.74913147
## asymmetry coefficient               -0.01107902
## length of kernel groove               1.00000000
```

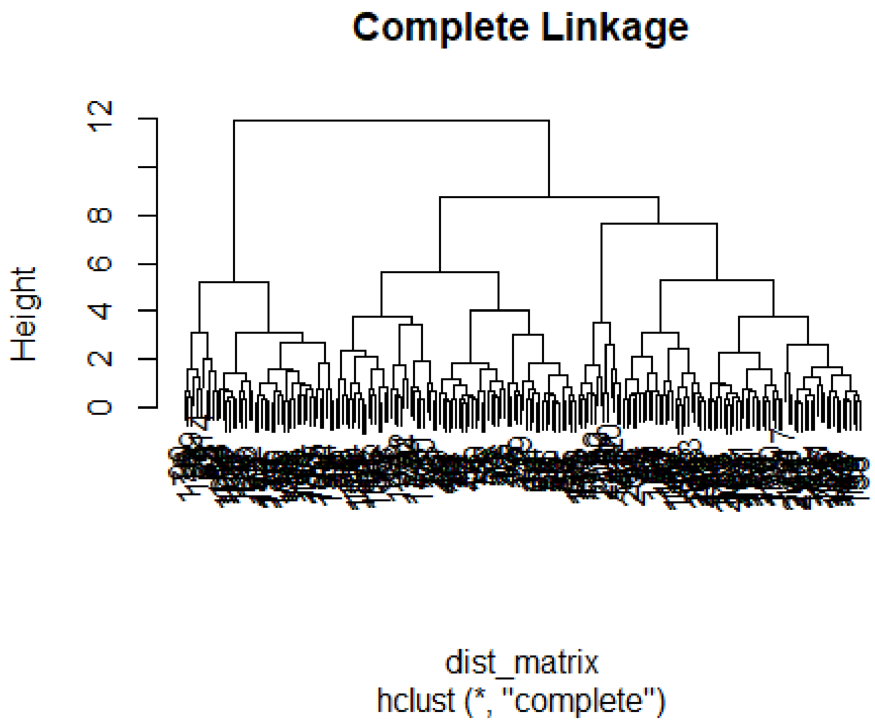
```
pairs(seeds[,1:7])
```



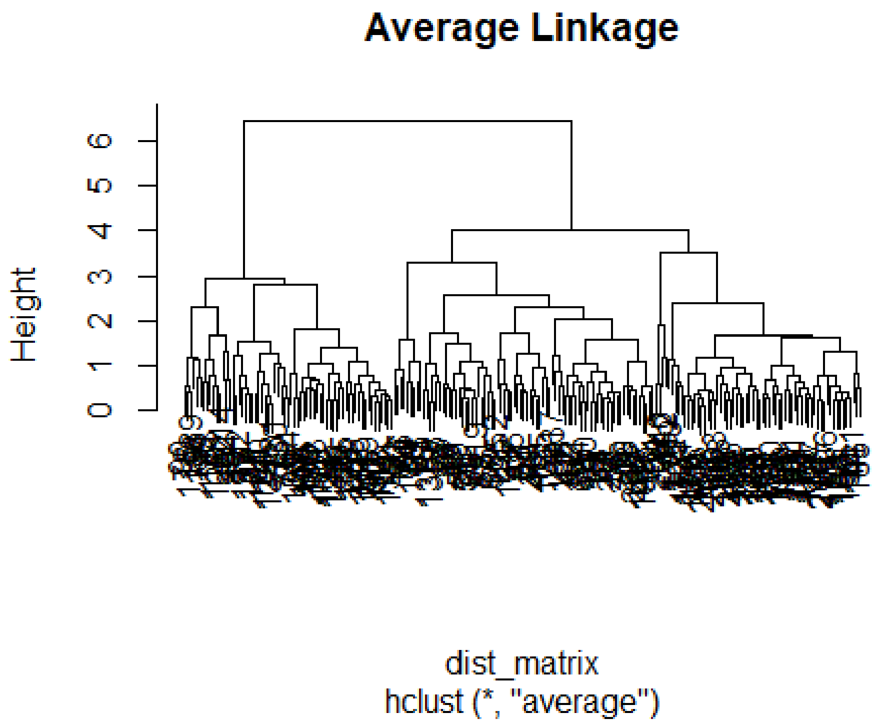
```r
dist_matrix <- dist(seeds[,1:7])


# Agglomerative clustering using single linkage
hc_single <- hclust(dist_matrix, method = "single")

# Agglomerative clustering using complete linkage
hc_complete <- hclust(dist_matrix, method = "complete")

# Agglomerative clustering using average linkage
hc_average <- hclust(dist_matrix, method = "average")



# Plot dendrogram for single linkage
plot(hc_single, main = "Single Linkage")
```



```r
# Plot dendrogram for complete linkage
plot(hc_complete, main = "Complete Linkage")
```



```r
# Plot dendrogram for average linkage
plot(hc_average, main = "Average Linkage")
```



R-Code:

```
if (!requireNamespace("datasetsICR", quietly = TRUE)) {
  install.packages("datasetsICR")
}
library(datasetsICR)

data("seeds")

str(seeds)

summary(seeds)

cor_matrix <- cor(seeds[,1:7])
print(cor_matrix)

pairs(seeds[,1:7])

dist_matrix <- dist(seeds[,1:7])

# Agglomerative clustering using single linkage
hc_single <- hclust(dist_matrix, method = "single")

# Agglomerative clustering using complete linkage
hc_complete <- hclust(dist_matrix, method = "complete")

# Agglomerative clustering using average linkage
hc_average <- hclust(dist_matrix, method = "average")
```

```
# Plot dendrogram for single linkage
plot(hc_single, main = "Single Linkage")

# Plot dendrogram for complete linkage
plot(hc_complete, main = "Complete Linkage")

# Plot dendrogram for average linkage
plot(hc_average, main = "Average Linkage")
```

Interpretation:

Pairwise Scatter Plots and Histograms:

These two show a very tight linear relationship, which was also indicated by the high correlation coefficient (0.9943). The distribution of both variables appears right skewed, suggesting that there are more smaller seeds than larger ones in the dataset.

The relationship between compactness and other variables such as area and perimeter is weaker, which aligns with the lower correlation coefficients (area-compactness: 0.6083, perimeter-compactness: 0.5292).

There's a strong positive linear relationship, although not as tight as area and perimeter, which is consistent with their correlation coefficient (0.8604).

It does not show a clear linear relationship with other variables and often has a negative correlation. This might indicate that it's an independent trait or influenced by different factors than the other measurements.

This shows a strong positive relationship with variables such as area, perimeter, and length of the kernel, which was also suggested by the correlation matrix.

Single Linkage Dendrogram

The single linkage dendrogram indicates that there's a chaining effect, with many seeds only slightly different from their nearest neighbor. The structure does not suggest clear, distinct clusters because merges occur at a very low height for most observations. This finding is consistent with the known tendency of single linkage to be sensitive to outliers, resulting in a less useful clustering for this dataset.

Complete Linkage Dendrogram

The complete linkage dendrogram shows a different picture. Clusters are more distinct, with branches merging at higher heights, indicating that within-cluster similarity is much higher before they merge with other clusters. This is aligned with the expectation that complete linkage creates more compact and well-separated clusters. However, there's a significant merge late in the process, indicating a substantial difference between the two major groups of seeds.

Average Linkage Dendrogram

The average linkage dendrogram seems to balance between the high sensitivity of single linkage and the strictness of complete linkage. Clusters are not as tightly bound as in complete linkage, nor as diffused as in single linkage. The height of merges suggests that seeds within clusters are on average, more like each other than to seeds in other clusters. This method may provide the most useful clustering for this dataset as it avoids the extremes of the other two methods.

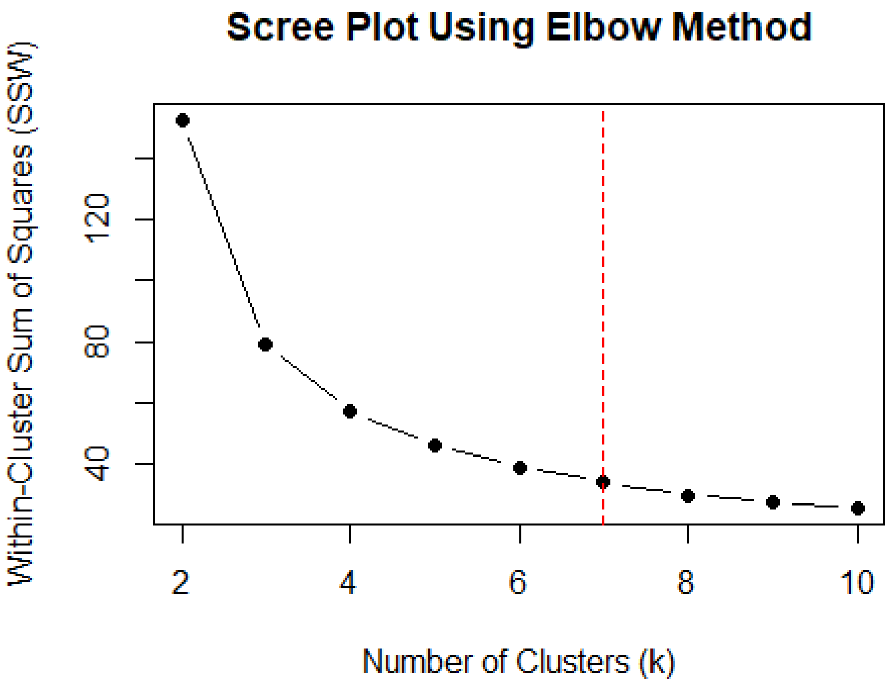
If one were to select several clusters based on the dendrograms, the average linkage may indicate a more distinct and physiologically significant split than a single connection. While the full linkage may over-segment the data into an excessive number of clusters, it also suggests certain appropriate categories. When determining the ideal number of clusters, one would probably consider external information about the seeds, such as known kinds, and locations of notable merging height increases.

Q2 K Means Clustering

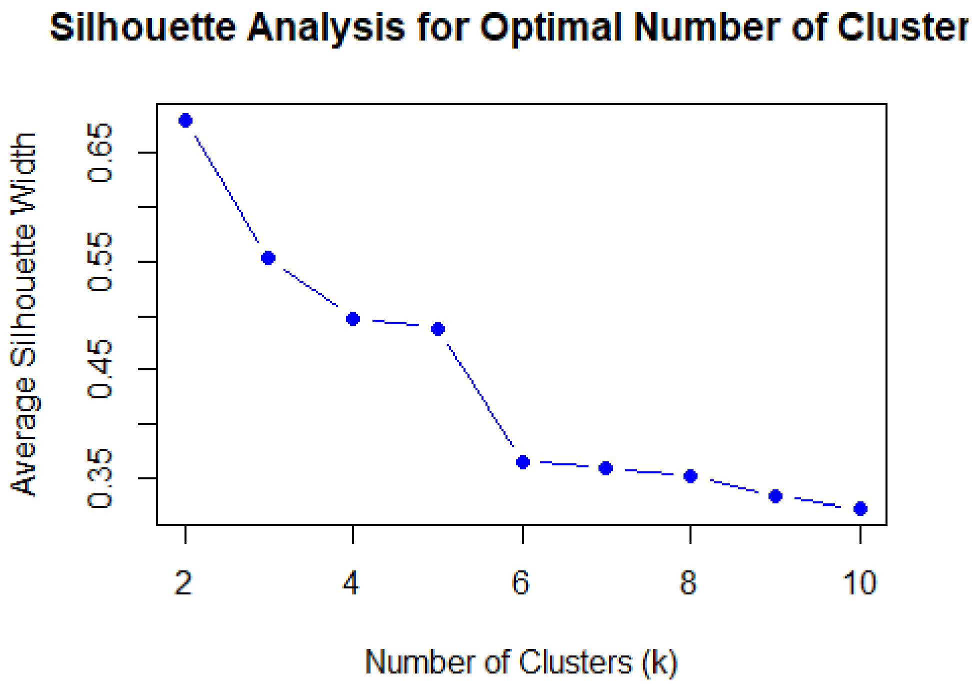
Salman

2024-03-18

```r
if (!requireNamespace("cluster", quietly = TRUE)) {
  install.packages("cluster")
}
library(cluster)  # For silhouette analysis

data(iris)
seeds_data <- iris[, 1:4]

ssw <- numeric(9)
avg_sil_width <- numeric(9)

for (k in 2:10) {
  set.seed(123)
  kmeans_result <- kmeans(seeds_data, centers = k, nstart = 25)
  ssw[k - 1] <- kmeans_result$tot.withinss


  silhouette_result <- silhouette(kmeans_result$cluster, dist(seeds_data))
  avg_sil_width[k - 1] <- mean(silhouette_result[, "sil_width"])
}

plot(2:10, ssw, type = "b", pch = 19, xlab = "Number of Clusters (k)", ylab =
"Within-Cluster Sum of Squares (SSW)",
     main = "Scree Plot Using Elbow Method")

abline(v = which.min(diff(diff(ssw))), col = "red", lty = 2)
```



```r
plot(2:10, avg_sil_width, type = "b", pch = 19, col = "blue", xlab = "Number
of Clusters (k)", ylab = "Average Silhouette Width",
     main = "Silhouette Analysis for Optimal Number of Clusters")

abline(v = which.max(avg_sil_width), col = "red", lty = 2)
```



```
optimal_k <- 3
set.seed(123)
final_kmeans_result <- kmeans(seeds_data, centers = optimal_k, nstart = 25)

# Print the cluster centers
print(final_kmeans_result$centers)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.006000    3.428000     1.462000    0.246000
## 2     5.901613    2.748387     4.393548    1.433871
## 3     6.850000    3.073684     5.742105    2.071053

# Print the size of each cluster
print(final_kmeans_result$size)

## [1] 50 62 38
```

R-Code:

```r
if (!requireNamespace("cluster", quietly = TRUE)) {
  install.packages("cluster")
}
library(cluster)  # For silhouette analysis


data(iris)
seeds_data <- iris[, 1:4]


ssw <- numeric(9)
avg_sil_width <- numeric(9)


for (k in 2:10) {
  set.seed(123)
  kmeans_result <- kmeans(seeds_data, centers = k, nstart = 25)
  ssw[k - 1] <- kmeans_result$tot.withinss



  silhouette_result <- silhouette(kmeans_result$cluster, dist(seeds_data))
  avg_sil_width[k - 1] <- mean(silhouette_result[, "sil_width"])
}


plot(2:10, ssw, type = "b", pch = 19, xlab = "Number of Clusters (k)", ylab = "Within-Cluster Sum of Squares (SSW)",
     main = "Scree Plot Using Elbow Method")


abline(v = which.min(diff(diff(ssw))), col = "red", lty = 2)
```

```
plot(2:10, avg_sil_width, type = "b", pch = 19, col = "blue", xlab = "Number of Clusters (k)",
ylab = "Average Silhouette Width",
    main = "Silhouette Analysis for Optimal Number of Clusters")


abline(v = which.max(avg_sil_width), col = "red", lty = 2)


optimal_k <- 3
set.seed(123)
final_kmeans_result <- kmeans(seeds_data, centers = optimal_k, nstart = 25)


# Print the cluster centers
print(final_kmeans_result$centers)


# Print the size of each cluster
print(final_kmeans_result$size)
```

Interpretation:

K-Means Clustering for k = 2 to 10:

I conducted k-means clustering with multiple values of k ranging from 2 to 10. This iterative process helps in determining the number of clusters that best capture the inherent groupings in the dataset.

Scree Plot Analysis (Elbow Method):

The scree plot reveals the within-cluster sum of squares (SSW) for different values of k. The plot shows a sharp decline from k=2 to k=3, and then the rate of decline lessens. This indicates that adding clusters beyond k=3 does not significantly improve the within-cluster compactness relative to k=2. The presence of the "elbow" around k=3, especially highlighted with a red dashed line in the second image, suggests that k=3 is a good choice for the number of clusters.

Silhouette Analysis:

The other image displays the silhouette analysis. The silhouette width measures how similar an object is to its cluster compared to other clusters. The values range from -1 to 1, where a high value indicates that the object is well-matched to its cluster and poorly matched to neighboring clusters. In this case, the silhouette width is highest for k=2, and it significantly drops as k increases to 3, then levels off. However, since the width is not dramatically higher for k=2 compared to k=3, and considering the elbow method's indication, k=3 is still a reasonable choice.

Optimal Number of Clusters:

Combining the insights from both the scree plot and the silhouette analysis, k=3 is chosen as the optimal number of clusters. It balances between minimizing SSW (as suggested by the elbow method) and maximizing the silhouette width.

Interpretation of Clusters for k=3:

The final k-means clustering with k=3 provides the following insights:

Cluster 1: Contains 50 elements. The cluster centers for the Sepal Length, Sepal Width, Petal Length, and Petal Width are relatively small, which may suggest this cluster groups the smaller flowers or a particular species of Iris (likely the Iris Setosa given its distinct characteristics in the dataset). Cluster 1 with centers (5.006, 3.428, 1.462, 0.246) likely corresponds to Iris Setosa, characterized by smaller petals and sepals.

Cluster 2: With 62 elements, this is the largest cluster. The center values for this cluster are intermediate in size. This cluster might represent another species that has medium-sized features (possibly Iris Versicolor). Cluster 2 with centers (5.901613, 2.748387, 4.393548, 1.433871) likely corresponds to Iris Versicolor, which has intermediate measurements among the three species.

Cluster 3: The smallest cluster, with 38 elements. It has the largest center values, indicating that it might contain the largest flowers or represent a different species (possibly Iris Virginica). Cluster 3 with centers (6.850000, 3.073684, 5.742105, 2.071053) likely corresponds to Iris Virginica, which tends to have the largest petals and sepals of the three species.

Given the known structure of the Iris dataset, where there are indeed three species (Setosa, Versicolor, and Virginica), the analysis aligns well with the biological classification. These interpretations align with the known dimensions of the Iris flower parts for each species. The clustering has successfully captured the natural groupings in the data, and the choice of k=3 for the number of clusters is well-justified by both the Elbow method and the Silhouette analysis, even though the latter suggested a slightly higher silhouette width for k=2. The clustering results suggest that while two species (Versicolor and Virginica) may be similar to each other, they are distinctly different from Setosa, which is well separated from the other two. This might explain why the silhouette width for k=2 was relatively high but still was not chosen over k=3, which provides a more nuanced separation in line with the known biological species classification.

Q3. Model Based Clustering

Salman

2024-03-18

```
if (!requireNamespace("datasetsICR", quietly = TRUE)) {
  install.packages("datasetsICR")
}
library(datasetsICR)

if (!requireNamespace("mclust", quietly = TRUE)) {
  install.packages("mclust")
}
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.3.3
```

```
## Package 'mclust' version 6.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
data("seeds", package = "datasetsICR")


str(seeds)
```

```
## 'data.frame':    210 obs. of  8 variables:
##  $ area                   : num  15.3 14.9 14.3 13.8 16.1 ...
##  $ perimeter              : num  14.8 14.6 14.1 13.9 15 ...
##  $ compactness            : num  0.871 0.881 0.905 0.895 0.903 ...
##  $ length of kernel       : num  5.76 5.55 5.29 5.32 5.66 ...
##  $ width of kernel        : num  3.31 3.33 3.34 3.38 3.56 ...
##  $ asymmetry coefficient  : num  2.22 1.02 2.7 2.26 1.35 ...
##  $ length of kernel groove: num  5.22 4.96 4.83 4.8 5.17 ...
##  $ variety                : Factor w/ 3 levels "Kama","Rosa",..: 1 1 1 1 1 
1 1 1 1 1 ...
```

```
summary(seeds)
```

```
##       area         perimeter      compactness     length of kernel
##  Min.   :10.59   Min.   :12.41   Min.   :0.8081   Min.   :4.899   
##  1st Qu.:12.27   1st Qu.:13.45   1st Qu.:0.8569   1st Qu.:5.262   
##  Median :14.36   Median :14.32   Median :0.8734   Median :5.524   
##  Mean   :14.85   Mean   :14.56   Mean   :0.8710   Mean   :5.629   
##  3rd Qu.:17.30   3rd Qu.:15.71   3rd Qu.:0.8878   3rd Qu.:5.980   
##  Max.   :21.18   Max.   :17.25   Max.   :0.9183   Max.   :6.675   
##  width of kernel asymmetry coefficient length of kernel groove     variety
##  Min.   :2.630   Min.   :0.7651        Min.   :4.519            Kama    :70  
##  1st Qu.:2.944   1st Qu.:2.5615        1st Qu.:5.045            Rosa    :70  
##  Median :3.237   Median :3.5990        Median :5.223          Canadian:70
##  Mean   :3.259   Mean   :3.7002        Mean   :5.408
##  3rd Qu.:3.562   3rd Qu.:4.7687        3rd Qu.:5.877
##  Max.   :4.033   Max.   :8.4560        Max.   :6.550
```

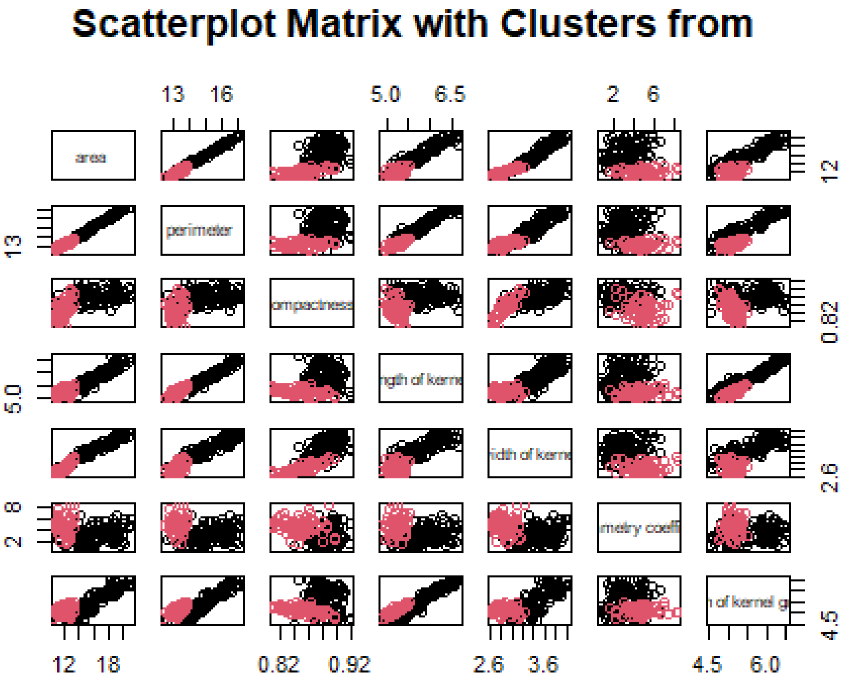
```
set.seed(123)
fitGPCM <- Mclust(seeds, G = 2:6)


summary(fitGPCM)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 2 components:
##
##  log-likelihood   n df      BIC      ICL
##        1231.203 210 82 2023.943 2023.943
##
## Clustering table:
##   1   2
## 140  70
```

```
bestModel <- fitGPCM$bestModel
cat("The best model according to BIC is:", bestModel, "\n")
```

```
## The best model according to BIC is:
```

```
clusters <- fitGPCM$classification


seeds_with_clusters <- cbind(seeds, Cluster = as.factor(clusters))

pairs(seeds_with_clusters[, 1:7], col =
as.numeric(seeds_with_clusters$Cluster),
      main = paste("Scatterplot Matrix with Clusters from", bestModel))
```


Scatterplot Matrix with Clusters from
area
perimeter
ompactness
ngth of kerne
idth of kerne
metry coeffi
n of kernel g
13
16
5.0
6.5
2
6
12
13
0.82
5.0
2.6
8
2
4.5
12
18
0.82
0.92
2.6
3.6
4.5
6.0

R-Code:

```
if (!requireNamespace("datasetsICR", quietly = TRUE)) {
  install.packages("datasetsICR")
}
library(datasetsICR)

if (!requireNamespace("mclust", quietly = TRUE)) {
  install.packages("mclust")
}
library(mclust)



data("seeds", package = "datasetsICR")


str(seeds)
summary(seeds)


set.seed(123)
fitGPCM <- Mclust(seeds, G = 2:6)
summary(fitGPCM)



bestModel <- fitGPCM$bestModel
cat("The best model according to BIC is:", bestModel, "\n")

clusters <- fitGPCM$classification
```

```r
seeds_with_clusters <- cbind(seeds, Cluster = as.factor(clusters))


pairs(seeds_with_clusters[, 1:7], col = as.numeric(seeds_with_clusters$Cluster),
      main = paste("Scatterplot Matrix with Clusters from", bestModel))
```

Interpretation:

The best Gaussian finite mixture model chosen by BIC is the VEV model with 2 components. This suggests that the data is best described by two clusters when considering ellipsoidal shapes with equal volume and variable orientation. The exact BIC value is not provided in the text output included, but the VEV model was selected as the best among the models for G=2 to 6.

According to the clustering table in the output, the data is split into two clusters, with 140 observations in one cluster and 70 in another. This represents a significant difference in cluster sizes, which could reflect different group densities or variability within the data.

The scatterplots indicate varying degrees of linear relationships between different pairs of attributes. For instance, there seems to be a strong positive linear relationship between the "area" and "perimeter," which is expected as larger seeds will generally have a longer perimeter.

There are a few data points that stand apart from the main clusters, especially visible in plots involving the "asymmetry coefficient" variable. These may be outliers or unusual observations that could warrant further investigation.

The histograms on the diagonal show the distribution of each variable. For most variables, the distribution appears to be roughly normal or slightly skewed, which is generally suitable for Gaussian mixture modeling.

Scatterplot Matrix Observations:

The scatterplots indicate a clear distinction between the two clusters for most variable pairings. This distinction is particularly noticeable in the plots involving "perimeter" and "area" against other variables, which suggests that these two variables are good indicators for clustering in this context.

The colors in the scatterplot matrix show that Cluster 1 has more observations than Cluster 2, consistent with the summary output.

Some variables, like "compactness," "length of kernel," and "length of kernel groove," show a strong linear relationship, indicating that these physical characteristics of seeds increase together.

The scatterplots for the "asymmetry coefficient" show less clear clustering, indicating that this variable may be less discriminative for the two identified groups or has higher within-group variability.

The histograms on the diagonal of the scatterplot matrix suggest varying distributions of variables. For instance, the "area" and "perimeter" show a slightly skewed distribution, whereas "compactness" appears more symmetric.

Findings:

The strong linear relationships in certain plots suggest that some physical dimensions of the seeds are directly correlated. For example, as seeds get larger in "area," they also have a longer "perimeter" and "length of the kernel."

Cluster separation in the scatterplots suggests that the chosen model is effective in grouping seeds with similar measurements together. However, it would be essential to look at the overlap areas and consider whether a different number of clusters might capture the data's structure more accurately.

Outliers and points that do not group well in the scatterplots might warrant additional investigation to understand if they represent noise, data entry errors, or inherent variability in seed characteristics.

Q4. Overall Comparison

Salman

2024-03-18

```
library(datasetsICR)
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.3.3
```

```
## Package 'mclust' version 6.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(cluster)
library(flexclust)
```

```
## Warning: package 'flexclust' was built under R version 4.3.3
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
data("seeds")

str(seeds)
```

```
## 'data.frame':    210 obs. of  8 variables:
##  $ area                   : num  15.3 14.9 14.3 13.8 16.1 ...
##  $ perimeter              : num  14.8 14.6 14.1 13.9 15 ...
##  $ compactness            : num  0.871 0.881 0.905 0.895 0.903 ...
##  $ length of kernel       : num  5.76 5.55 5.29 5.32 5.66 ...
##  $ width of kernel        : num  3.31 3.33 3.34 3.38 3.56 ...
##  $ asymmetry coefficient  : num  2.22 1.02 2.7 2.26 1.35 ...
##  $ length of kernel groove: num  5.22 4.96 4.83 4.8 5.17 ...
##  $ variety                : Factor w/ 3 levels "Kama","Rosa",..: 1 1 1 1 1
1 1 1 1 1 ...
```

```
summary(seeds)
```

```
##       area         perimeter      compactness      length of kernel
##  Min.   :10.59   Min.   :12.41   Min.   :0.8081    Min.   :4.899
##  1st Qu.:12.27   1st Qu.:13.45   1st Qu.:0.8569    1st Qu.:5.262
##  Median :14.36   Median :14.32   Median :0.8734    Median :5.524
##  Mean   :14.85   Mean   :14.56   Mean   :0.8710    Mean   :5.629
##  3rd Qu.:17.30   3rd Qu.:15.71   3rd Qu.:0.8878    3rd Qu.:5.980
##  Max.   :21.18   Max.   :17.25   Max.   :0.9183   Max.   :6.675
##  width of kernel asymmetry coefficient length of kernel groove     variety
##  Min.   :2.630   Min.   :0.7651         Min.   :4.519           Kama    :70
##  1st Qu.:2.944   1st Qu.:2.5615         1st Qu.:5.045           Rosa    :70
##  Median :3.237   Median :3.5990         Median :5.223           Canadian:70
##  Mean   :3.259   Mean   :3.7002         Mean   :5.408
##  3rd Qu.:3.562   3rd Qu.:4.7687         3rd Qu.:5.877
##  Max.   :4.033   Max.   :8.4560         Max.   :6.550

true_labels <- seeds$variety


dist_seeds <- dist(seeds[,1:7])
hc_complete <- hclust(dist_seeds, method = "complete")
predicted_labels_hc <- cutree(hc_complete, k = 3)

# K-means clustering
set.seed(123) # Set seed for reproducibility
predicted_labels_km <- kmeans(seeds[,1:7], centers = 3)$cluster

# Fit GPCM
best_model <- Mclust(seeds[,1:7], G = 3)
predicted_labels_gpcm <- best_model$classification

# Classification tables for each method
table_hc <- table(True = true_labels, PredictedHC = predicted_labels_hc)
table_km <- table(True = true_labels, PredictedKM = predicted_labels_km)
table_gpcm <- table(True = true_labels, PredictedGPCM =
predicted_labels_gpcm)

# Print classification tables
cat("Hierarchical Clustering Classification Table:\n")

## Hierarchical Clustering Classification Table:

print(table_hc)

##           PredictedHC
## True         1  2  3
##   Kama      52 18  0
##   Rosa      23  0 47
##   Canadian  0 70  0

cat("\nK-Means Clustering Classification Table:\n")

##
## K-Means Clustering Classification Table:

print(table_km)
```

```
##           PredictedKM
## True        1  2  3
##   Kama      1 60  9
##   Rosa     60 10  0
##   Canadian  0  2 68
```

```
cat("\nGPCM Classification Table:\n")
```

```
##
## GPCM Classification Table:
```

```
print(table_gpcm)
```

```
##           PredictedGPCM
## True        1  2  3
##   Kama     60  4  6
##   Rosa      5 65  0
##   Canadian  5  0 65
```

```
# Calculate and print ARIs
ari_hc <- adjustedRandIndex(true_labels, predicted_labels_hc)
ari_km <- adjustedRandIndex(true_labels, predicted_labels_km)
ari_gpcm <- adjustedRandIndex(true_labels, predicted_labels_gpcm)

cat("\nARI for Hierarchical Clustering:", ari_hc, "\n")
```

```
##
## ARI for Hierarchical Clustering: 0.546135
```

```
cat("ARI for K-Means Clustering:", ari_km, "\n")
```

```
## ARI for K-Means Clustering: 0.7166199
```

```
cat("ARI for GPCM Clustering:", ari_gpcm, "\n")
```

```
## ARI for GPCM Clustering: 0.7373941
```

R-Code:

```
library(datasetsICR)

library(mclust)

library(cluster)

library(flexclust)


data("seeds")

str(seeds)

summary(seeds)


true_labels <- seeds$variety


dist_seeds <- dist(seeds[,1:7])

hc_complete <- hclust(dist_seeds, method = "complete")

predicted_labels_hc <- cutree(hc_complete, k = 3)


# K-means clustering

set.seed(123) # Set seed for reproducibility

predicted_labels_km <- kmeans(seeds[,1:7], centers = 3)$cluster


# Fit GPCM

best_model <- Mclust(seeds[,1:7], G = 3)

predicted_labels_gpcm <- best_model$classification


# Classification tables for each method

table_hc <- table(True = true_labels, PredictedHC = predicted_labels_hc)

table_km <- table(True = true_labels, PredictedKM = predicted_labels_km)
```

```r
table_gpcm <- table(True = true_labels, PredictedGPCM = predicted_labels_gpcm)

# Print classification tables
cat("Hierarchical Clustering Classification Table:\n")
print(table_hc)
cat("\nK-Means Clustering Classification Table:\n")
print(table_km)
cat("\nGPCM Classification Table:\n")
print(table_gpcm)

# Calculate and print ARIs
ari_hc <- adjustedRandIndex(true_labels, predicted_labels_hc)
ari_km <- adjustedRandIndex(true_labels, predicted_labels_km)
ari_gpcm <- adjustedRandIndex(true_labels, predicted_labels_gpcm)

cat("\nARI for Hierarchical Clustering:", ari_hc, "\n")
cat("ARI for K-Means Clustering:", ari_km, "\n")
cat("ARI for GPCM Clustering:", ari_gpcm, "\n")
```

Interpretation:

Classification Tables:

The hierarchical clustering table shows that this method has done reasonably well in distinguishing the Canadian variety from the others (perfectly classifying all 70 as cluster 2), but it seems to confuse Kama and Rosa somewhat, with 23 Kama classified as Rosa and 18 Rosa classified as Kama.

The k-means clustering table shows that it has performed better in distinguishing between Kama and Rosa (only 1 Kama is classified as Rosa and 10 Rosa as Kama) but still confuses some Canadian as Kama or Rosa (2 cases).

The GPCM table indicates the best performance among the three methods in separating the Kama and Rosa varieties, with only 5 cases of Kama classified as Rosa and 4 of Rosa as Kama. It also performs very well with Canadian variety, with only 5 cases being misclassified.

The ARI for hierarchical clustering is 0.546135, which suggests a moderate similarity between the true labels and the predicted labels from hierarchical clustering.

The ARI for k-means clustering is higher, at 0.7166199, indicating a substantial agreement and a better performance compared to hierarchical clustering.

The ARI for GPCM is the highest at 0.7373941, suggesting that the GPCM method results in a clustering solution most similar to the true labels, and is the best performer out of the three methods used.

Findings:

The results indicate that the GPCM has the highest ARI, suggesting that its clustering solution is closest to the true labels, which could imply a better fit for the underlying structure of the dataset. This model-based approach considers not just the distance between points but also the statistical properties of the data, leading to more nuanced clustering.

K-means, despite its simplicity, performs better than hierarchical clustering with complete linkage, which may be because k-means optimize cluster cohesion and separation in a way that happens to align better with the true labels in this dataset. It's worth noting that k-means clustering assumes clusters are of approximately equal sizes, which matches our data, as there are equal numbers of each variety of seeds.

Hierarchical clustering using complete linkage, while still useful, provides a less accurate match to the true labels compared to k-means and GPCM in this instance. This might be due to the nature of complete linkage that tends to find the most dissimilar pairs of observations in clusters, which can lead to less cohesive clusters if the data has more overlap.

The classification tables also reflect the confusion between varieties that each method experiences. In terms of classification performance, while all methods exhibit some misclassification, GPCM shows the lowest level of misclassification among the three, followed by k-means, and then hierarchical clustering.

Comment on the Results:

These findings suggest that for the seeds dataset, model-based clustering methods like GPCM might capture the structure of the data more effectively than k-means or hierarchical clustering. This could be due to GPCM's ability to model the covariance structure of the data, rather than just the distance between observations. The improved performance of GPCM as indicated by the ARI suggests it is better at accounting for the actual relationships between the different seed varieties in the dataset.

The misclassification rate, which is not directly provided in the output but can be inferred from the classification tables, would be lowest for GPCM, followed by k-means, and highest for hierarchical clustering. This reinforces the idea that model-based clustering can often provide superior results for complex datasets, although it might be more computationally intensive and requires more assumptions than k-means or hierarchical methods.

In summary, when comparing clustering algorithms, it's clear that while all methods provide some insight into the structure of the data, the choice of algorithm can have a significant impact on the clustering result, and it's important to consider the specific characteristics of the dataset when choosing a method.