*Create a title page here*

**Leaf Clustering and Classification using the K-means Algorithm**

**1.0 Introduction**

Leaf classification is a fundamental task with numerous real-world applications, ranging from botany and environmental science to agriculture and bioinformatics. The objective of our analysis is to employ K-Means clustering to group leaves based on various morphological and textural attributes, including Eccentricity, Aspect Ratio, Elongation, Solidity, Convexity, Isoperimetric Factor, Depth, Lobedness, Intensity, Contrast, Smoothness, Third moment, Uniformity, and Entropy. Through this analysis, we aim to uncover inherent patterns within the dataset, facilitating the automatic identification and categorization of plant species by their leaves. Specifically, we seek to answer questions related to the natural grouping of leaves based on these attributes and understand whether these attributes can effectively distinguish different species of plants. This analysis will aid in botanical research, species classification, and potentially contribute to our understanding of plant diversity.

The problem domain of this analysis centers on botany and plant classification. Accurate classification and identification of plant species are crucial for various fields, including agriculture, ecology, and environmental science. Understanding the unique features that distinguish different species of leaves is fundamental for plant taxonomy and can also have practical applications in species conservation and horticulture. It is estimated that there are over 390,000 plant species known to science, and new discoveries continue to expand our knowledge. Accurate and efficient species identification can help researchers and botanists manage and protect plant diversity, as well as contribute to ecosystem preservation and sustainable agriculture.

The choice of k-Means clustering as the analytical method is justified by its suitability for unsupervised pattern recognition and clustering. K-means clustering is particularly relevant for this problem because it can identify natural groupings within the leaf dataset without any prior information about the species or categories. By applying k-means, we aim to uncover inherent structures and similarities among the leaves based on their morphological features, which can lead to the classification of leaves into distinct clusters. This method is well-suited for exploring the dataset's inherent variability and will allow us to observe whether certain features are strong indicators of species categorization. Ultimately, the results of this analysis will contribute to the field of botany by providing insights into leaf diversity and aiding in the identification and classification of plant species based on morphological attributes.

**2.1 Understanding the Data**

The dataset consists of a collection of shape and texture features extracted from digital images of leaf specimens originating from a total of 40 different plant species, harvesting an average number of 16 leaf specimens from each plant. There are 340 observations and 16 study variables in total.

**Table 1.0: Description of Data Variables**

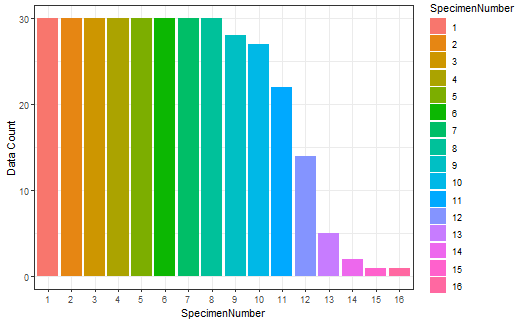
|  |  |  |
| --- | --- | --- |
|  | **Variable Name** | **Type** |
| 1 | Class (Species) | Categorical |
| 2 | Specimen Number | Categorical |
| 3 | Eccentricity | Continuous |
| 4 | Aspect Ratio | Continuous |
| 5 | Elongation | Continuous |
| 6 | Solidity | Continuous |
| 7 | Stochastic Convexity | Continuous |
| 8 | Isoperimetric Factor | Continuous |
| 9 | Maximal Indentation Depth | Continuous |
| 10 | Lobedness | Continuous |
| 11 | Average Intensity | Continuous |
| 12 | Average Contrast | Continuous |
| 13 | Smoothness | Continuous |
| 14 | Third Moment | Continuous |
| 15 | Uniformity | Continuous |
| 16 | Entropy | Continuous |

**2.2. Exploratory Analysis**

Using the ‘str’ function, it was observed that the variables in the dataset consist of numeric majorly, except for the Species and Specimen which are in integer form. Converting the integer data to factor, the ‘summary’ of the data shows that there are 40 different plant species, harvesting an average number of 16 leaf specimens from each plant. Further, the minimum and maximum, as well as measures of central tendency (mean, median) and spread (1st and 3rd quartiles) for each of these variables were obtained

**Variable Analysis**

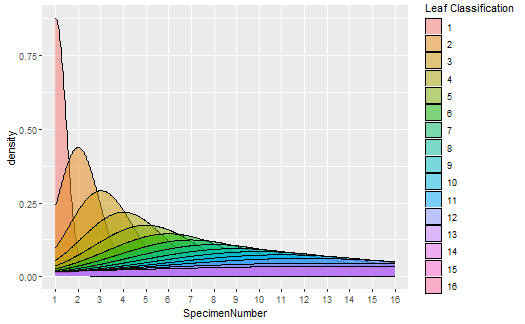
The bar chart illustrates the distribution of specimens across various categories, with number 1 specimens having the highest count, followed by number 2 specimens, and so on. This suggests that the category labeled as "number 1" contains the most abundant specimens, while the category labeled "number 2" ranks second in terms of quantity. The chart's vertical bars visually convey the disparity in specimen counts, making it evident that number 1 specimens dominate the dataset, with a gradual decrease in frequency as we move to higher-numbered categories. This hierarchical arrangement of bars helps viewers quickly grasp the relative distribution of specimens within each ategory, making it a valuable tool for visual data interpretation.



**Figure 2.1: Specimen Number of the Leaf Plant**

**Density Plot**

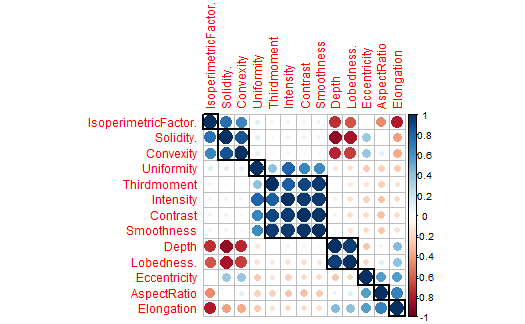
We can gather that the majority of the dataset lies with the first specimen and the least with the sixteenth specimen.



**Figure 2.2: Density graph for Specimen Number of the Leaf Plant**

**Correlation**

Correlations is used to determine whether and to what extent two or more variables are related linearly. Pearson's Correlation Coefficient, one of the most commonly used correlation measures, provides a numerical value that represents the strength and direction of this linear relationship. The correlation matrix in the figure below shows that there is a great correlation between most of the variables like IsoperimetricFactor, Solidity, Convexity, Depth, Lobedness, Aspectratio, and Elongation. While Uniformity, Thirdmoment, Intensity, Contrast, Smoothness, and Eccentricity are least correlated.

****

**Figure 2.3: Correlation Matrix of study variables**

**2.3. Preprocessing**

**Check for missing data**

Next, we check how many NA records we have, per column. Fortunately, there are no records of missing data in the dataset.

**Data Splitting** *- Splitting the data set into Train and Test*

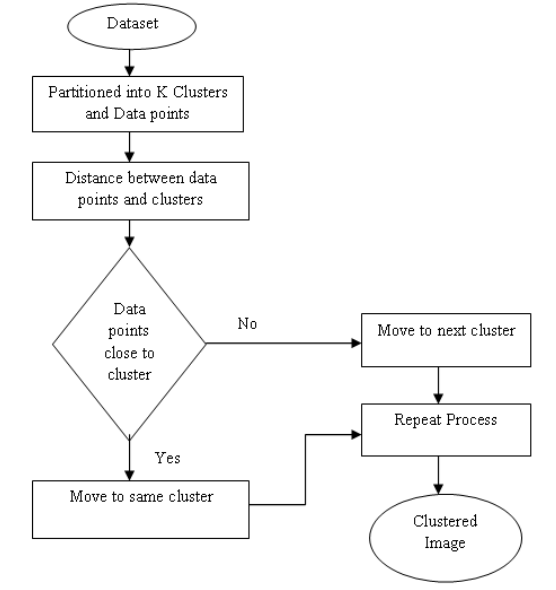
Data Splitting Functions A series of test/training partitions are created using createDataPartition while createResample creates one or more bootstrap samples.

**2.4. K-means Clustering Algorithm**

K-Means is a well-established unsupervised learning algorithm used for solving the clustering problem. This method offers a straightforward approach to classify a given dataset into a predefined number of clusters (k). The core concept involves determining k centroids, one for each cluster, and strategically placing them to maximize their separation. The goal is to assign each data point to its nearest centroid. When all points have been assigned, the initial grouping is complete. Subsequently, new centroids are computed as the barycenter of the clusters formed in the previous step. This process is iterated until the centroids no longer change, indicating convergence.

During this iterative process, the centroids gradually settle into their final positions. In other words, they stop moving, signifying the end of the K-Means clustering procedure. This method yields a set of distinct, non-overlapping clusters, providing a valuable tool for data analysis and pattern recognition. K-Means is numerical, unsupervised, and works through iterative refinement. It is particularly useful for segmenting data into groups with minimal intragroup variability and maximum intergroup separation. Moreover, it is commonly applied in various domains, including image segmentation, where it is a popular choice for dividing images into meaningful regions.

There are always K clusters. There is always at least one item in each cluster. The clusters are non-hierarchical, and they do not overlap. Every member of a cluster is closer to its cluster than any other cluster because closeness does not always involve the ‘centre’ of clusters. The k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.



**Figure 2.4: Flow Chart for K-means Algorithm**

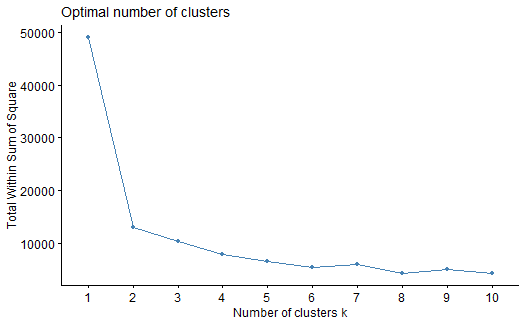
**2.5. Model Fitting**

The dataset was divided into training and test set using R code, after which we perform the neural network model and evaluate the confusion matrix with model methods such as sensitivity, specificity, positive prediction value, negative prediction value, and prevalence of the data.

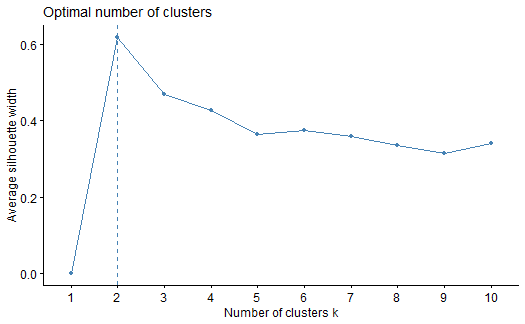
**3. Results**

**3.1 Output**

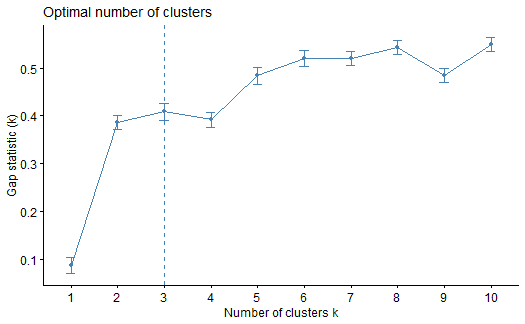
In Kmeans Algorithm we have to define the number of Cluster which represent with k fortunatly in R, We have some intersting functions for selecting k. Figures 3.1, 3.2 and 3.3 show the visualization of various methods to estimate the optimal number of clusters for k-means clustering, usingvdifferent metrics as a function of the number of clusters. The two methods ('wss' and 'silhouette') show us it's better choose k = 2. While the gap statistics shows the the iptimal k is 3.

****

**Figure 3.1: Optimal Number of Cluster k using Variance (WSS) method**

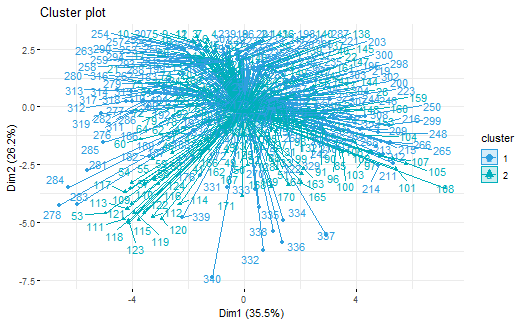
****

**Figure 3.2: Optimal Number of cluster k using the Silhouette Score**

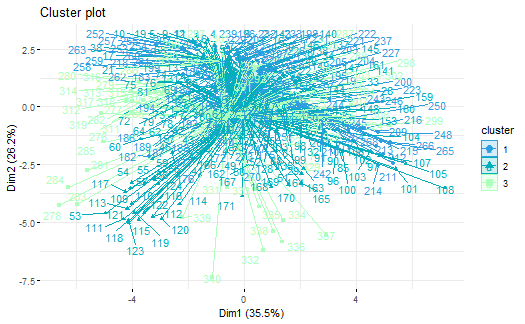
****

**Optimal Number of cluster k using Gap Statistics**

The function fviz\_cluster() [factoextra package] is used to easily visualize k-means clusters. It takes k-means results and the original data as arguments. In the resulting plot, observations are represented by points, using principal components provided the number of variables is greater than 2. The visualization reveals the outcomes of the k-means clustering analysis performed on the dataset. Each distinct cluster is depicted with a different color, representing different cluster groups. The ellipses surrounding each cluster are drawn based on the Euclidean distance, showcasing the concentration of data points within each group. The star plots, which connect the cluster centroids to individual data points within the clusters, offer insight into the distribution and relationship of data points in relation to their respective cluster centers. This helps us understand the dispersion and arrangement of data within each cluster. The use of label repelling ensures that the cluster labels are clearly displayed and prevents any overlap that might make them unreadable, resulting in a more informative visualization.

****

**Figure 3.3: K-means Cluster Plot of the Leaf Dataset with k = 2**

****

**Figure 3.4: K-means Cluster Plot of the Leaf Dataset with k = 3**

**Model Properties and Evaluation**

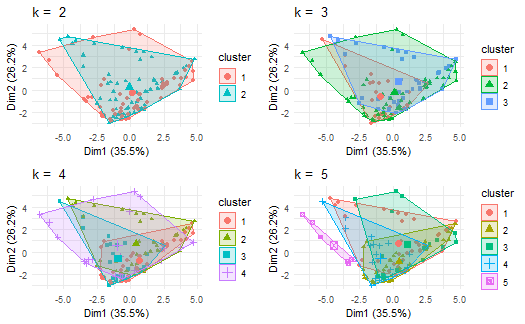
The grid of plots displays the results of k-means clustering for different values of 'k' (the number of clusters) applied to the 'testdata' dataset. Each plot represents a specific clustering configuration. As 'k' increases from 2 to 5, we observe the impact of having a different number of clusters on the data.

In k = 2: In this plot, the data is divided into two distinct clusters. It appears that the data points have been separated into two relatively well-defined groups. This may indicate a clear division in the dataset, and the two clusters likely represent meaningful distinctions in the data.

k = 3: With three clusters, the data is further divided. We observe the emergence of a third cluster, indicating that the algorithm has found more subtle patterns or subgroups within the data. This suggests that the data might have some finer-grained structure that is revealed when using three clusters.

k = 4: When the number of clusters is increased to four, the data is divided even further. This suggests that the algorithm is attempting to capture even more nuanced patterns in the dataset. However, we should be cautious about over-segmenting the data, as this may lead to clusters that don't have clear, practical interpretations.

k = 5: With five clusters, the data is partitioned into even smaller segments. This may be indicative of an attempt to uncover highly specific subgroups within the data. It's essential to consider whether such divisions are meaningful or merely artifacts of the clustering process.



**Figure 3.5: Clustering Plot of the Leaf Dataset for different values of 'k'**

**4. Conclusion**

**4.1 Summary**

This study explores the utilization of k-means algorithms to analyze herbal leaf species, focusing on assessing the similarity between different leaves based on key features, including color and shape. The k-means algorithm offers a straightforward approach, making it suitable for handling large datasets. It also presents an uncomplicated process, requiring users to specify the number of clusters (k). To determine the optimal number of clusters, we employed several methods, including Within-Cluster Sum of Squares (WSS), Gap Statistics, and the Silhouette method. Our analysis consistently identified k = 2 as the optimal number of clusters for most of the datasets. Additionally, we discussed distribution analysis using bar charts for each plant leaf. The outcomes of this study aim to assist researchers and botanists who face challenges in identifying plant species due to variations in size, shape, and color. For smaller values of k the algorithms give good results. For larger values of k, the segmentation is very coarse many clusters appear in the images at discrete places. Different initial partitions can result in different final clusters .The advantage of K

Means algorithm is simple and quite efficient. It works well when clusters are not well separated from each other.

**4.2 Limitations of the study**

* The main limitation of this study is to use the secondary database, future study should be done based on primary data for more accuracy of the results related to leaf identification.
* It is incredibly costly and time-consuming to conduct a classification process and check for accuracy in the area of botany.
* Furthermore, the dataset used in the study is limited in terms of the number of samples and, this may result in a loss of valuable information that could have contributed to a more comprehensive understanding of the study.
* It is hard to explain the detail of each attribute because there is no reference about this work. Because of that, we maybe make some misunderstanding to attributes.
* The outcomes may change by the period and environment when the dataset is collected.

**4.3 Recommendation/Improvement Areas:**

* Future research should improve the feature extraction method and add more features.
* We suggest several avenues for further research. One potential direction is to explore the use of other meta-learning algorithms and compare their performance with the approach presented in this study.
* Expansion of the developed database to contemplate more plants with other types of
* leaves, and to increase the number of specimens considered for each class.
* Development of a classification scheme which could integrate both simple and complex
* leaves in the same computational application;
* Comparison between other techniques not considered in this analysis in order to provide
* a non-biased evaluation of the classification results.

library(factoextra)

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(corrplot)

## corrplot 0.92 loaded

data <- read.csv('leaf.csv')

# Summary of the data  
summary(data)

## Species SpecimenNumber Eccentricity AspectRatio   
## Min. : 1.00 Min. : 1.000 Min. :0.1171 Min. : 1.007   
## 1st Qu.: 9.00 1st Qu.: 3.000 1st Qu.:0.5506 1st Qu.: 1.211   
## Median :15.00 Median : 6.000 Median :0.7634 Median : 1.571   
## Mean :18.54 Mean : 6.282 Mean :0.7199 Mean : 2.440   
## 3rd Qu.:29.00 3rd Qu.: 9.000 3rd Qu.:0.8951 3rd Qu.: 2.343   
## Max. :36.00 Max. :16.000 Max. :0.9987 Max. :19.038   
## Elongation Solidity. Convexity IsoperimetricFactor.  
## Min. :0.1076 Min. :0.4855 Min. :0.3965 Min. :0.07838   
## 1st Qu.:0.3496 1st Qu.:0.8907 1st Qu.:0.9662 1st Qu.:0.34682   
## Median :0.5019 Median :0.9481 Median :0.9930 Median :0.57916   
## Mean :0.5138 Mean :0.9042 Mean :0.9438 Mean :0.53123   
## 3rd Qu.:0.6334 3rd Qu.:0.9769 3rd Qu.:1.0000 3rd Qu.:0.70071   
## Max. :0.9483 Max. :0.9939 Max. :1.0000 Max. :0.85816   
## Depth Lobedness. Intensity Contrast   
## Min. :0.002837 Min. :0.001464 Min. :0.005022 Min. :0.03342   
## 1st Qu.:0.009521 1st Qu.:0.016500 1st Qu.:0.022843 1st Qu.:0.08336   
## Median :0.023860 Median :0.103615 Median :0.042087 Median :0.11937   
## Mean :0.037345 Mean :0.523845 Mean :0.051346 Mean :0.12453   
## 3rd Qu.:0.047834 3rd Qu.:0.416432 3rd Qu.:0.073046 3rd Qu.:0.16379   
## Max. :0.198980 Max. :7.206200 Max. :0.190670 Max. :0.28081   
## Smoothness Thirdmoment Uniformity Entropy   
## Min. :0.001115 Min. :0.0002294 Min. :6.920e-06 Min. :0.1694   
## 1st Qu.:0.006901 1st Qu.:0.0020796 1st Qu.:1.023e-04 1st Qu.:0.7189   
## Median :0.014050 Median :0.0044468 Median :2.387e-04 Median :1.0775   
## Mean :0.017670 Mean :0.0059277 Mean :3.872e-04 Mean :1.1626   
## 3rd Qu.:0.026128 3rd Qu.:0.0083069 3rd Qu.:5.162e-04 3rd Qu.:1.5546   
## Max. :0.073089 Max. :0.0297860 Max. :2.936e-03 Max. :2.7085

# Understanding the datatype of dataset  
str(data)

## 'data.frame': 340 obs. of 16 variables:  
## $ Species : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ SpecimenNumber : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Eccentricity : num 0.727 0.742 0.767 0.738 0.823 ...  
## $ AspectRatio : num 1.47 1.53 1.57 1.46 1.77 ...  
## $ Elongation : num 0.324 0.361 0.39 0.354 0.445 ...  
## $ Solidity. : num 0.985 0.982 0.978 0.976 0.977 ...  
## $ Convexity : num 1 0.998 1 1 1 ...  
## $ IsoperimetricFactor.: num 0.836 0.799 0.808 0.817 0.755 ...  
## $ Depth : num 0.00466 0.00524 0.00746 0.00688 0.00743 ...  
## $ Lobedness. : num 0.00395 0.005 0.01012 0.00861 0.01004 ...  
## $ Intensity : num 0.04779 0.02416 0.0119 0.01595 0.00794 ...  
## $ Contrast : num 0.128 0.0905 0.0574 0.0655 0.0453 ...  
## $ Smoothness : num 0.01611 0.00812 0.00329 0.00427 0.00205 ...  
## $ Thirdmoment : num 0.005232 0.002708 0.000921 0.001154 0.00056 ...  
## $ Uniformity : num 2.75e-04 7.48e-05 3.79e-05 6.63e-05 2.35e-05 ...  
## $ Entropy : num 1.176 0.697 0.443 0.588 0.342 ...

In Kmeans Algorithm we have to define the number of Cluster which represent with k fortunatly in R, We have some intersting functions for selecting k.

fviz\_nbclust(data, kmeans, method = "wss")

# method can be = 'gap\_stat', 'wss', 'silhouette'  
fviz\_nbclust(data, kmeans, method = "gap\_stat")

fviz\_nbclust(data, kmeans, method = "silhouette")

km1 <- kmeans(data, 2)  
km1

## K-means clustering with 2 clusters of sizes 171, 169  
##   
## Cluster means:  
## Species SpecimenNumber Eccentricity AspectRatio Elongation Solidity.  
## 1 8.321637 6.391813 0.6958871 1.892832 0.5089389 0.8785164  
## 2 28.887574 6.171598 0.7441049 2.994066 0.5186389 0.9301025  
## Convexity IsoperimetricFactor. Depth Lobedness. Intensity Contrast  
## 1 0.9278964 0.5105804 0.04143485 0.6018020 0.05205962 0.1258638  
## 2 0.9598777 0.5521311 0.03320595 0.4449653 0.05062465 0.1231901  
## Smoothness Thirdmoment Uniformity Entropy  
## 1 0.01761541 0.005884356 0.0003474819 1.241627  
## 2 0.01772520 0.005971592 0.0004274350 1.082697  
##   
## Clustering vector:  
## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [186] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [260] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [297] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [334] 2 2 2 2 2 2 2  
##   
## Within cluster sum of squares by cluster:  
## [1] 5839.998 7153.868  
## (between\_SS / total\_SS = 73.5 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

All method(‘wss’, ‘silhouette’, and gap\_stat) show us it’s better choose k = 2 The function fviz\_cluster() [factoextra package] can be used to easily visualize k-means clusters. It takes k-means results and the original data as arguments. In the resulting plot, observations are represented by points, using principal components if the number of variables is greater than 2

fviz\_cluster(km1, data = data,  
 palette = c("#2E9FDF", "#00AFBB"),  
 ellipse.type = "euclid", # Concentration ellipse  
 star.plot = TRUE, # Add segments from centroids to items  
 repel = TRUE, # Avoid label overplotting (slow)  
 ggtheme = theme\_minimal())

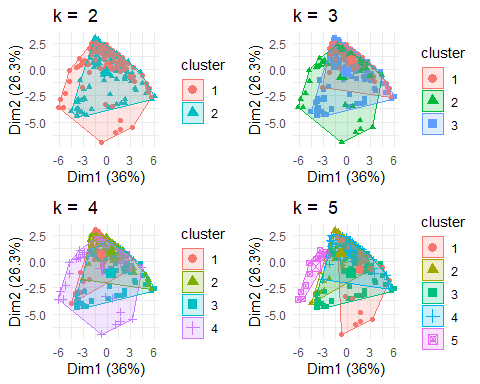
km2 <- kmeans(data, 3)  
km2

## K-means clustering with 3 clusters of sizes 169, 92, 79  
##   
## Cluster means:  
## Species SpecimenNumber Eccentricity AspectRatio Elongation Solidity.  
## 1 28.887574 6.171598 0.7441049 2.994066 0.5186389 0.9301025  
## 2 11.706522 7.271739 0.6340270 1.764072 0.5016842 0.8517743  
## 3 4.379747 5.367089 0.7679267 2.042780 0.5173875 0.9096590  
## Convexity IsoperimetricFactor. Depth Lobedness. Intensity Contrast  
## 1 0.9598777 0.5521311 0.03320595 0.4449653 0.05062465 0.1231901  
## 2 0.9178877 0.4804467 0.04953235 0.7833299 0.06671777 0.1457582  
## 3 0.9395520 0.5456728 0.03200484 0.3904024 0.03498937 0.1026957  
## Smoothness Thirdmoment Uniformity Entropy  
## 1 0.01772520 0.005971592 0.0004274350 1.0826974  
## 2 0.02267928 0.007457012 0.0004769076 1.4842542  
## 3 0.01171824 0.004052910 0.0001967584 0.9590748  
##   
## Clustering vector:  
## [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [75] 3 3 3 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [149] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [297] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [334] 1 1 1 1 1 1 1  
##   
## Within cluster sum of squares by cluster:  
## [1] 7153.868 1988.445 1392.896  
## (between\_SS / total\_SS = 78.5 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

fviz\_cluster(km2, data = data,  
 palette = c("#2E9FDF", "#00AFBB", "#ADFFBB"),  
 ellipse.type = "euclid", # Concentration ellipse  
 star.plot = TRUE, # Add segments from centroids to items  
 repel = TRUE, # Avoid label overplotting (slow)  
 ggtheme = theme\_minimal())

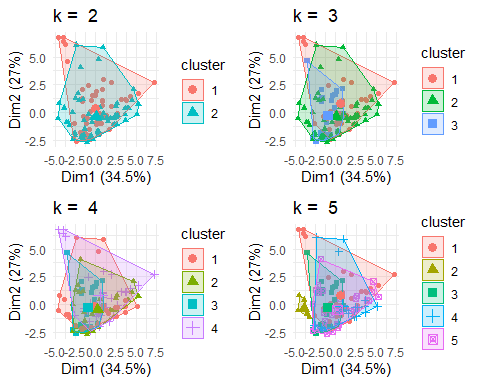
# Splitting data into training and testing with 60 and 40 percent respectively  
index <- sample(2, nrow(data), replace=TRUE, prob = c(0.60, 0.40))  
traindata <- data[index==1, ]  
testdata <- data[index==2, ]

plot\_ <- function(i){  
 cluster <- kmeans(traindata, centers = i, nstart = 25)  
 plot\_list <- fviz\_cluster(cluster, geom = "point", data = traindata) + ggtitle(paste("k = ", i)) + theme\_minimal()  
 return(plot\_list)  
 }  
   
  
library(gridExtra)  
grid.arrange(plot\_(2),plot\_(3),plot\_(4),plot\_(5), nrow = 2) + theme\_minimal()



## NULL

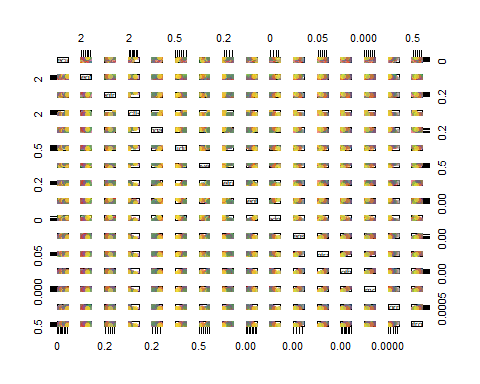
plot\_ <- function(i){  
 cluster <- kmeans(testdata, centers = i, nstart = 25)  
 plot\_list <- fviz\_cluster(cluster, geom = "point", data = testdata) + ggtitle(paste("k = ", i)) + theme\_minimal()  
 return(plot\_list)  
 }  
   
  
library(gridExtra)  
grid.arrange(plot\_(2),plot\_(3),plot\_(4),plot\_(5), nrow = 2) + theme\_minimal()



## NULL

leaf <- data[,-16]

library(RColorBrewer)  
palette(alpha(brewer.pal(9,'Set1'), 0.5))  
plot(testdata, col=leaf$Specimen, pch=16, bty="n") + theme\_minimal()



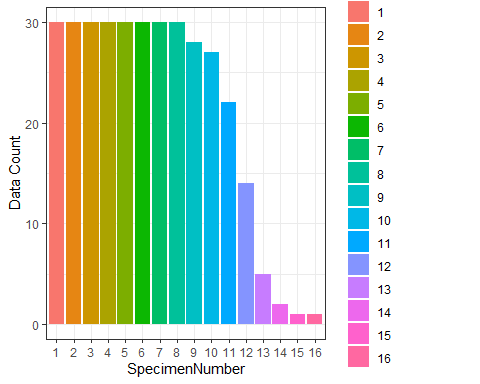
## NULL

# Converting the integer data to factor  
leaf$Species <- as.factor(leaf$Species)  
leaf$SpecimenNumber <- as.factor(leaf$SpecimenNumber)

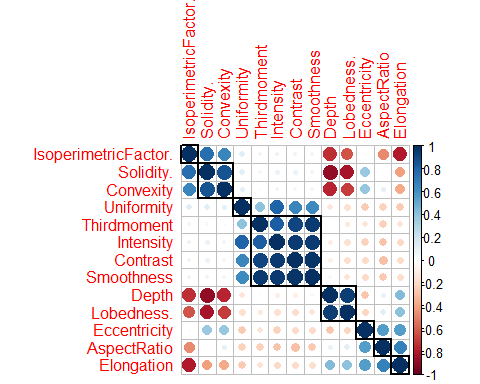
summary(leaf)

## Species SpecimenNumber Eccentricity AspectRatio   
## 11 : 16 1 : 30 Min. :0.1171 Min. : 1.007   
## 9 : 14 2 : 30 1st Qu.:0.5506 1st Qu.: 1.211   
## 10 : 13 3 : 30 Median :0.7634 Median : 1.571   
## 13 : 13 4 : 30 Mean :0.7199 Mean : 2.440   
## 24 : 13 5 : 30 3rd Qu.:0.8951 3rd Qu.: 2.343   
## 1 : 12 6 : 30 Max. :0.9987 Max. :19.038   
## (Other):259 (Other):160   
## Elongation Solidity. Convexity IsoperimetricFactor.  
## Min. :0.1076 Min. :0.4855 Min. :0.3965 Min. :0.07838   
## 1st Qu.:0.3496 1st Qu.:0.8907 1st Qu.:0.9662 1st Qu.:0.34682   
## Median :0.5019 Median :0.9481 Median :0.9930 Median :0.57916   
## Mean :0.5138 Mean :0.9042 Mean :0.9438 Mean :0.53123   
## 3rd Qu.:0.6334 3rd Qu.:0.9769 3rd Qu.:1.0000 3rd Qu.:0.70071   
## Max. :0.9483 Max. :0.9939 Max. :1.0000 Max. :0.85816   
##   
## Depth Lobedness. Intensity Contrast   
## Min. :0.002837 Min. :0.001464 Min. :0.005022 Min. :0.03342   
## 1st Qu.:0.009521 1st Qu.:0.016500 1st Qu.:0.022843 1st Qu.:0.08336   
## Median :0.023860 Median :0.103615 Median :0.042087 Median :0.11937   
## Mean :0.037345 Mean :0.523845 Mean :0.051346 Mean :0.12453   
## 3rd Qu.:0.047834 3rd Qu.:0.416432 3rd Qu.:0.073046 3rd Qu.:0.16379   
## Max. :0.198980 Max. :7.206200 Max. :0.190670 Max. :0.28081   
##   
## Smoothness Thirdmoment Uniformity   
## Min. :0.001115 Min. :0.0002294 Min. :6.920e-06   
## 1st Qu.:0.006901 1st Qu.:0.0020796 1st Qu.:1.023e-04   
## Median :0.014050 Median :0.0044468 Median :2.387e-04   
## Mean :0.017670 Mean :0.0059277 Mean :3.872e-04   
## 3rd Qu.:0.026128 3rd Qu.:0.0083069 3rd Qu.:5.162e-04   
## Max. :0.073089 Max. :0.0297860 Max. :2.936e-03   
##

ggplot(leaf, aes(x=SpecimenNumber, fill = SpecimenNumber)) +   
 theme\_bw()+  
 geom\_bar()+  
 labs(x = "SpecimenNumber", y = "Data Count")



corr\_mat <- cor(leaf[,3:ncol(leaf)])  
corrplot(corr\_mat, order = "hclust", tl.cex = 1, addrect = 8)



ggplot(leaf, aes(x = SpecimenNumber, fill = SpecimenNumber)) +  
 geom\_density(alpha=0.5) +  
 scale\_fill\_discrete(name = "Leaf Classification")

## Warning: Groups with fewer than two data points have been dropped.  
## Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning  
## -Inf  
  
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning  
## -Inf

