Assignment 5: K-Means Clustering Using R

Written by Daanish Ahmed

DATA 630 9041

Semester Summer 2017

Professor Edward Herranz

UMUC

July 31, 2017

Introduction

K-means clustering is a popular form of unsupervised classification that is used to group together similar instances in a dataset. It works by finding shared characteristics between instances and grouping those instances into a cluster. Each cluster will contain instances with strong similarities to those in the same cluster and few similarities (if any) to those in different clusters (Han, Kamber, & Pei, 2011). K-means clustering is distinct from other clustering methods in that it requires the user to choose the number of clusters (k) before implementing the method. There are many ways to determine the value of k: one such way is to use the formula $k \approx \sqrt(n/2)$, where n is equal to the number of instances in the dataset (Bati, 2015). Another method to determine the number of clusters is to use the "elbow method," which involves creating a plot of the within clusters sum of squares as a function of k and choosing the k-value where the slope appears to change into the shape of an "elbow" (Bati, 2015). My goal is to use R software to implement the k-means clustering method on a dataset containing leaf species information (Silva, Marcal, & Almeida da Silva, 2014). I will use the two methods described to select the number of clusters, in addition to experimenting with other k-values that I find interesting. I will then evaluate the results to determine the ideal number of clusters to use in my dataset. I expect that the results of my analysis will yield an effective clustering model that allows for the easy identification of leaf specimens based on their characteristics.

Data Exploration

I will begin my analysis with a description of the initial dataset. This dataset contains 16 variables with a total of 340 observations (see Figure 1 in Appendix B). These variables include the leaf species, specimen number, aspect ratio, elongation, solidity, smoothness, and uniformity (Silva et al., 2014). The leaf species would ideally be the dependent variable, but I will not be using this variable because k-means clustering is not used for prediction. All variables are in numeric or integer form, but the values for species and specimen number represent categorical values. For further exploration, I generated a set of descriptive statistics from the dataset (see Figure 2 in Appendix B). Note that this figure is missing the species and specimen number variables because they were removed during preprocessing. These numbers reveal that the average leaf has a high solidarity and convexity while having a relatively low smoothness and entropy. The figure also reveals that there are no missing values in any of the variables. This dataset will now require preprocessing before I implement the clustering method.

Preprocessing

There are a few important preprocessing steps to perform before model implementation. First, I installed and activated the "cluster" package in R, which is required for using the k-means clustering method. Next, I created a copy of the original dataset to preserve the initial data for later use. All further activities will be performed on this dataset copy. I then removed the species and specimen number variables from the dataset. One reason why I removed these variables is that their values represent categorical outputs. The k-means clustering algorithm cannot function if there are any categorical variables, and thus it requires all variables to be numeric ("K-Means Clustering," 2017). Although the variables are integers, their categorical representations might alter the clustering results. Another reason for their removal is that these variables are not necessary for my analysis. The dependent variable (species) is not needed because I am grouping the data according to similar characteristics and I do not need to know the species beforehand.

Likewise, the specimen number contains little useful information. From here, the remaining variables are all numeric and there are no missing values to address. The last step is to set all input variables to the same scale such that they have the same mean and standard deviation. This is necessary because variables with high variance can impact the results of the clustering (Bati, 2015). After scaling the variables and verifying that they have means equal to 0 (see Figure 3 in Appendix B), I have finished preparing the data for algorithm implementation.

Initial Clustering Model

I will now discuss the first clustering model that I will create for this analysis. This model will use the formula $k \approx \sqrt{}(n/2)$ to determine the number of clusters. Since there are 340 instances in this dataset, the number of clusters is equal to 13. The first step is to generate a random seed in R that will allow us to reproduce the results whenever we run the code. The next step is to build the model by calling the "kmeans" method on our dataset. This method will use the k-means clustering algorithm to split the instances of a given dataset into the specified number of clusters. For the input parameters, I used the leaf dataset copy as the data source and set the number of clusters to 13. Once the model has been built, I generated the model's output which contains several pieces of information (see Figure 4 in Appendix B). The first piece of information is the size of each cluster. We can see that the 13 clusters contain between 12 and 42 instances. The next section contains the cluster means, which shows the average value for each variable within each cluster. For instance, cluster 1 has a mean eccentricity of -0.997 and a mean solidity of 0.514. Next, we see the clustering vector—indicating the cluster in which each of the dataset's instances belongs to. For example, instances 1 and 2 belong to cluster 11 while instances 3 through 7 belong to cluster 7. After this section is the sum of squares within clusters, which indicates the sum of squares distance between a cluster’s center and an instance within that cluster (“K-Means Clustering,” 2017). We see that the within clusters sum of squares ranges from 26.45 to 194.95 for the current set of clusters. The last section of this output shows the available components, which can be called to obtain additional information about the model.

I will call two additional components from this list (see Figure 5 in Appendix B). First, I examine the between clusters sum of squares, which shows the sum of squares distance between an instance and the center of a cluster outside of the current cluster (“K-Means Clustering,” 2017). I found that the sum of squares between clusters is 3855.63. I then examined the number of iterations required to cluster the dataset, which is equal to 4. Next, I created a cross-table to compare the model’s clusters to the expected leaf species from the original dataset (see Figure 6 in Appendix B). This table can be used to find the dominant species in each cluster. For example, species 11 is the dominant species in cluster 4—with 16 out of 26 instances. But more importantly, the table can be used to calculate the percentage of instances in this model that match their actual leaf species. This is done by adding the highest numbers in each row and dividing by the total number of instances (“K-Means Clustering,” 2017). We find that 235 out of 340 instances were correctly predicted, meaning that the percentage of correct instances is 69.1%. Finally, I generated a plot of the clustering model, showing all 13 clusters and the instances that are assigned to them (see Figure 7 in Appendix B). I will now examine these results in the following section.

Initial Results

These findings offer some useful insights as to whether the current clustering model is the most effective form of classification for this dataset. Firstly, the size of each cluster ranges from

12 to 48 instances, which means that some clusters may be too small to adequately represent the data (see Figure 4 in Appendix B). However, the ideal clustering model should have a low sum of squares within clusters and a high sum of squares between clusters ("K-Means Clustering," 2017). In this case, the average sum of squares within clusters is quite small, with most of these values being less than 100 (see Figure 4 in Appendix B). Furthermore, the sum of squares between clusters has a relatively high value of 3855.63 (see Figure 5 in Appendix B). Because of this, it seems that using the current number of clusters results in a very effective model. But the consequence of using a higher number of clusters is that it causes the clusters to have less meaning, since many of them will only have a few observations. Another metric for evaluation is obtained from the cross-table that compares the clusters to the expected class (see Figure 6 in Appendix B). This table reveals that 69.1% of leaf classifications match their original species, meaning that the error rate is over 30%. This finding makes it difficult to recommend the current model for classification, even though it has good values for the sum of squares distances. Finally, the plot of the clustering model highlights the fact that some clusters have very few instances while others are packed with instances (see Figure 7 in Appendix B). According to Bati (2015), one way to address this issue is to decrease the number of clusters used in the model. Thus, I will continue this analysis by using other methods that involve smaller k-values.

Elbow Method

Another effective method for choosing the number of clusters is to use the "elbow method." As stated before, this method involves plotting the within clusters sum of squares over k and choosing the k-value where the graph appears to form an "elbow." According to Ng (2017), the disadvantage of using this method is that the graph might have a consistent change of slope, meaning that there may not actually be an “elbow” within the graph. If this happens, it is better to use a different method to determine the number of clusters (Ng, 2017). This method was implemented by computing the within clusters sum of squares for k between 2 and 15, and then plotting the within clusters sum of squares as a function of k (Anand, 2017). The resulting graph indicates that the possible “elbows” occur at k = 3 and k = 4 (see Figure 8 in Appendix B). Though it is somewhat ambiguous whether these points are indeed “elbows,” it still seems that the slope changes the most at these points when compared to any other point in the graph. Because of this, I will use the results from the elbow method and build two clustering models with these k-values to determine which model is ideal for my dataset.

Elbow Method Implementation

First, I will create a clustering model using k = 3 clusters. I once again called the “kmeans” method to build the model, but the difference is that I set the number of clusters to 3. From here, I examined some of the properties of the model (see Figure 9 in Appendix B). This figure reveals that the average cluster size is much larger than that of the previous model, with values ranging between 36 and 200. Also, this new model only requires 3 iterations, which makes it computationally easier to build. However, the sum of squares between clusters is 2294.04, which is much lower than that of the first model. Likewise, the average sum of squares within clusters is significantly higher than that of the original model. These findings seem to indicate that the previous model—which involves a much higher number of clusters—may possibly be a stronger model than the current one. However, the cross-table reveals that the current clustering model has correctly grouped 278 out of 340 instances (see Figure 10 in Appendix B), giving it an accuracy rate of 81.8% and an error rate that is lower than 20%. This suggests that the current model is much more effective at classification for this dataset—since the previous model had an accuracy of only 69.1%. Furthermore, the plot of the current clustering model indicates that each cluster has a higher number of instances when compared to those in the previous model (see Figure 11 in Appendix B). Since each cluster contains more data, it is much more likely that they will accurately represent the data when compared to the smaller clusters of the previous model. Thus, this model contains advantages as well as disadvantages compared to the last model.

Now, I will build a clustering model using the second "elbow" of k = 4 clusters. I first used the "kmeans" method to implement the model using 4 clusters as an input parameter. After building the model, I once again examined its major properties (see Figure 12 in Appendix B). On average, this model has fewer instances per cluster than the model using 3 clusters. However, its clusters are still larger than those from the model with 13 clusters. This makes sense, as it implies that clusters will get smaller as the number of clusters increases. This model requires 3 iterations, giving it the same computational complexity as the previous model. Additionally, its sum of squares between clusters is 2794.88, which is higher than that of the last model. Likewise, the sum of squares within clusters has a lower average value. Although the results are not as good as those from the first model, it implies that using 4 clusters may lead to a stronger model than using only 3 clusters. The plot of the current model is similar to that of the last model, except that it contains 4 clusters instead of 3 (see Figure 13 in Appendix B). Though some clusters clearly have more instances than others, even the smallest clusters are still larger than most of the clusters in the first model. Most interestingly, the cross-table reveals that the current model has correctly classified 289 out of 340 instances—giving it an accuracy rate of 85.0% and an error of only 15% (see Figure 14 in Appendix B). This is the most accurate model my analysis so far, and thus it might be the best option for classification among the models I have tested.

Additional Model

So far in my analysis, I have noticed that models with higher numbers of clusters have higher sums of squares between clusters and lower sums of squares within clusters. However, the downside to using large k-values is that the models tend to be less accurate with regards to classification. For my final model, I seek to determine if this will still be the case when using a very large k-value. A good way to choose the number of clusters is to think about the purpose that the clusters serve and divide them in a way that makes sense logically (Ng, 2017). One logical way to split the clusters in this dataset is to create one cluster for each species of leaf. Since there are 30 different leaf species in this dataset, I will use a k-value of 30.

I once again called the "kmeans" method on the dataset, this time using k = 30 as the input parameter. After building the model, I examined its important properties (see Figure 15 in Appendix B). One issue is that most of the clusters contain very few instances, meaning that most clusters will not represent the data adequately. This issue is highlighted in the clustering model's plot, which shows that some clusters contain many instances while others contain only 4 or 5 instances (see Figure 17 in Appendix B). The image in Figure 15 also reveals that the model required 5 iterations, making it the most computationally difficult model to build. The figure indicates that the sum of squares between clusters is 4285.48, which is the highest of any model in this analysis. Likewise, the average sum of squares within clusters is lower than that of any other model. These results might suggest that the current model is the best option for my dataset, but only if its classification accuracy is higher than those of my previous models. When examining the model's cross-table, we find that it correctly categorized only 206 out of 340 instances—resulting in an accuracy rate of 60.6% and an error rate that is almost 40% (see Figure 16 in Appendix B). These results indicate that the current model is the least effective clustering model to use for classification purposes. Additionally, these findings strongly suggest that the classification accuracy decreases as the number of clusters increases. It is possible that using a large number of clusters may produce more accurate results if using a much larger dataset. But as it stands, I would not recommend using a large k-value for the current dataset.

Conclusion

By the end of this analysis, I have created a total of four models using the k-means clustering method in R. By examining the results, my goal is to determine the number of clusters that would be needed to build the most effective clustering model for my dataset. To make the analysis easier, I have compiled all of my major findings into a single table (see Figure 18 in Appendix B). I found that there are two ways to evaluate the effectiveness of a clustering model. One method is to evaluate the sum of squares between clusters and the sum of squares within clusters, in which the goal is to maximize the former and minimize the latter. Through this method, it is apparent that using higher k-values will result in higher sums of squares between clusters and lower sums of squares within clusters. But the consequence of selecting large numbers of clusters is that each cluster will contain fewer entries—making it less likely that the clusters will adequately represent the data. In this dataset, using large k-values such as 13 or 30 produces many clusters that contain fewer than 20 instances.

The second evaluation method is to use a cross-table to determine the percent of matching classes. Using this method, it is evident that larger k-values coincide with less accurate classification and higher error rates. However, I have found that using 4 clusters results in a higher percentage of matching classes (85.0%) than using 3 clusters (81.8%). Furthermore, using a k-value of 4 also involves a higher sum of squares between clusters and a lower sum of squares within clusters when compared to using a k-value of 3. Therefore, I claim that the most effective clustering model used on my dataset is the one containing 4 clusters. However, this does not mean that a more effective model cannot be obtained. It is possible that using a slightly higher k-value (such as 5 or 6) may yield a higher percentage of matching classes. In fact, the main limitation with this algorithm was the process of determining the number of clusters to use. Based on my results, it seems that the "elbow method" yielded the most effective k-value of 4. Still, there is a chance that stronger k-values can be obtained from additional experimentation or by using other methods to select the number of clusters. For future research, I would recommend the development of additional models using different cluster sizes to test my dataset. If we can derive an improved model that has an accuracy of at least 90%, then such a model will serve as a vital unsupervised learning method for the classification of leaf specimens.

References

Anand, S. (2017, February 9). Finding Optimal Number of Clusters. Retrieved July 30, 2017, from https://www.r-bloggers.com/finding-optimal-number-of-clusters/

Bati, F. (2015, Fall). Clustering. Lecture presented at UMUC. Retrieved July 9, 2017.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques* (3rd ed.). Retrieved June 27, 2017.

K-Means Clustering in R – Exercise 7 (2017). Retrieved July 24, 2017.

Ng, A. (2017). *Lecture 81 - Choosing the Number of Clusters*. Lecture presented in Stanford University. Retrieved July 31, 2017, from https://www.coursera.org/learn/machine-learning/lecture/Ks0E9/choosing-the-number-of-clusters

Silva, P. F., Marcal, A. R., & Almeida da Silva, R. M. (2014, February 24). Leaf Data Set. Retrieved July 30, 2017, from https://archive.ics.uci.edu/ml/datasets/Leaf

Appendix A

R Script for Assignment 5

```
# DATA 630 Assignment 5

# Written by Daanish Ahmed

# Semester Summer 2017

# July 31, 2017

# Professor Edward Herranz


# This R script implements the k-means clustering method on a dataset containing

# leaf species information.  The purpose of this assignment is to build a model

# that splits the instances into clusters according to similar characteristics.

# Four models will be created, each using a different number of clusters.  The

# first model uses the formula k = sqrt(n/2) to find k.  The second and third

# models will use the "elbow method" to determine k.  The final model will set

# k equal to the number of classes (i.e. number of species, which is 30).



# This section of code covers opening the dataset and initializing the packages

# that are used in this script.


# Sets the working directory for this assignment.  Please change this directory

# to whichever directory you are using, and make sure that all files are placed
# in that location.
setwd("~/Class Documents/2016-17 Summer/DATA 630/R/Assignment 5")

# In order to run the clustering commands, we need to install the "cluster"
# package:

# If you have not installed this package yet, remove the # symbol below.
# install.packages("cluster")

# Loads the cluster package into the system.
library("cluster")

# Opens the CSV file "leaf.csv".
leaf <- read.csv("leaf.csv", head=TRUE, sep=",")

# Creates a copy of the dataset.
newleaf <- leaf

# End of opening the dataset.


# This section of code covers data preprocessing.  It includes exploration of
# the original dataset, removing variables, and dealing with missing values.
# Previews the dataset.
View(newleaf)

# Shows the initial structure of the dataset.
str(newleaf)

# Removes the species and specimen number variables from the dataset, since
# they represent categorical variables.
newleaf$Species <- NULL
newleaf$SpecimenNumber <- NULL

# Verifies that the variables have been removed.
str(newleaf)

# Shows the descriptive statistics of the dataset.
summary(newleaf)

# Sets the variables to the same scale, such that they have the same
# mean and standard deviation.
newleaf[1:14] <- scale(newleaf[1:14])

# Verifies that the variables are set to the same scale (see the mean).
```

```
summary(newleaf)

# End of data preprocessing.



# This section of code covers the implementation of the k-means clustering
# algorithm on the dataset using k = 13 clusters.  This k-value was
# obtained using the formula k = sqrt(n/2), where n equals the number of
# instances (340 in this dataset).

# Generates a random seed to allow us to reproduce the results.
set.seed(1234)

# Implements k-means clustering on the dataset using k = 13 clusters.
kc <- kmeans(newleaf, 13)

# Shows the output of the clustered dataset, including the number of
# instances in each cluster, the average values for each variable in all
# four clusters, and the sum of squares for each cluster.
kc

# Shows the between clusters sum of squares.
kc$betweenss
```

```
# Shows the number of iterations required to cluster the dataset.
kc$iter

# Shows the clustering of instances according to the actual leaf species.
table(leaf$Species, kc$cluster)

# Creates the cluster plot for the dataset.
clusplot(newleaf, kc$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)

# End of creating the first clustering model.


# This section of code covers the implementation of the "elbow method" to
# find the number of clusters to use.  I will create a clustering model
# for each possible "elbow" indicated by the method.

# The following code is from "Finding Optimal Number of Clusters"
# Based on Anand (2017)
# https://www.r-bloggers.com/finding-optimal-number-of-clusters/
# Modified for UMUC DATA 630 by Daanish Ahmed

# Generates a random seed to allow us to reproduce the results.
```

```
set.seed(1234)


# Computes the within clusters sum of squares with a minimum k of 2 and

# a maximum k of 15.

k.max <- 15

wss <- sapply(1:k.max,

          function(k){kmeans(newleaf, k, nstart=50,

                        iter.max = 15)$tot.withinss})


# Plots the graph of the within clusters sum of squares vs. the number

# of clusters.

plot(1:k.max, wss,

    type="b", pch = 19, frame = FALSE,

    xlab = "Number of clusters K",

    ylab = "Total within-clusters sum of squares")


# We see that possible "elbows" are k = 3 and k = 4.


# End of Anand code.




# First elbow: k = 3
# Generates a random seed to allow us to reproduce the results.
set.seed(1234)

# Implements k-means clustering on the dataset using k = 3 clusters.
kc <- kmeans(newleaf, 3)

# Shows the number of instances in each cluster.
kc$size

# Shows the between clusters sum of squares.
kc$betweenss

# Shows the within clusters sum of squares.
kc$withinss

# Shows the number of iterations required to cluster the dataset.
kc$iter

# Shows the clustering of instances according to the actual leaf species.
table(leaf$Species, kc$cluster)

# Creates the cluster plot for the dataset.
clusplot(newleaf, kc$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```

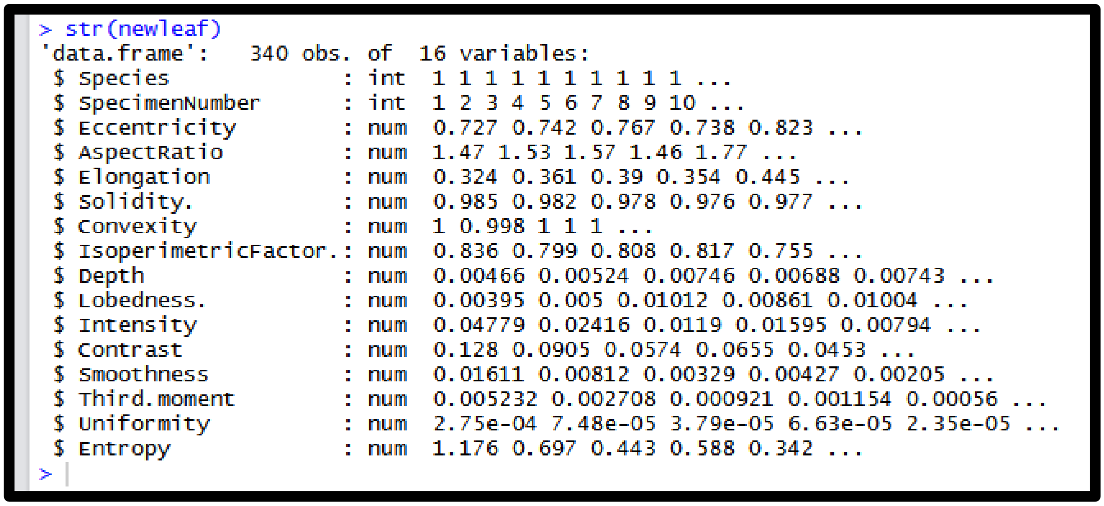
```
# End of creating clustering model for the first elbow.


# Second elbow: k = 4

# Generates a random seed to allow us to reproduce the results.
set.seed(1234)

# Implements k-means clustering on the dataset using k = 4 clusters.
kc <- kmeans(newleaf, 4)

# Shows the number of instances in each cluster.
kc$size

# Shows the between clusters sum of squares.
kc$betweenss

# Shows the within clusters sum of squares.
kc$withinss

# Shows the number of iterations required to cluster the dataset.
kc$iter
```

```
# Shows the clustering of instances according to the actual leaf species.
table(leaf$Species, kc$cluster)

# Creates the cluster plot for the dataset.
clusplot(newleaf, kc$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)

# End of creating clustering model for the second elbow.


# This section of code covers the creation of an additional clustering
# model that uses k = 30, meaning that there is one cluster for every
# leaf species.

# Generates a random seed to allow us to reproduce the results.
set.seed(1234)

# Implements k-means clustering on the dataset using k = 30 clusters.
kc <- kmeans(newleaf, 30)

# Shows the number of instances in each cluster.
kc$size
```

```
# Shows the between clusters sum of squares.

kc$betweenss


# Shows the within clusters sum of squares.

kc$withinss


# Shows the number of iterations required to cluster the dataset.

kc$iter


# Shows the clustering of instances according to the actual leaf species.

table(leaf$Species, kc$cluster)


# Creates the cluster plot for the dataset.

clusplot(newleaf, kc$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)


# End of creating clustering model for k = 30.


# End of script.
```

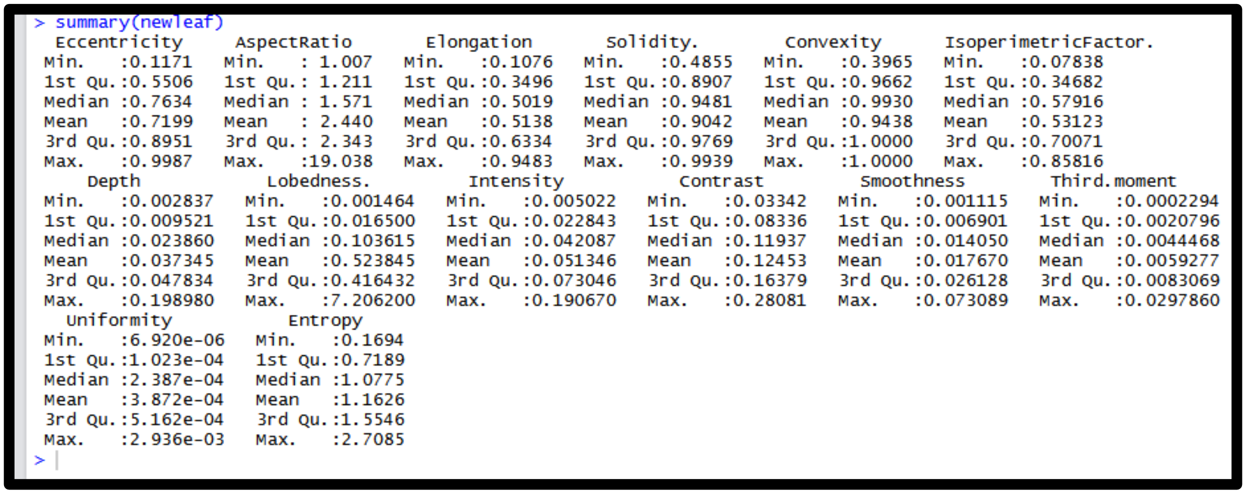
Appendix B

Relevant R Output Images



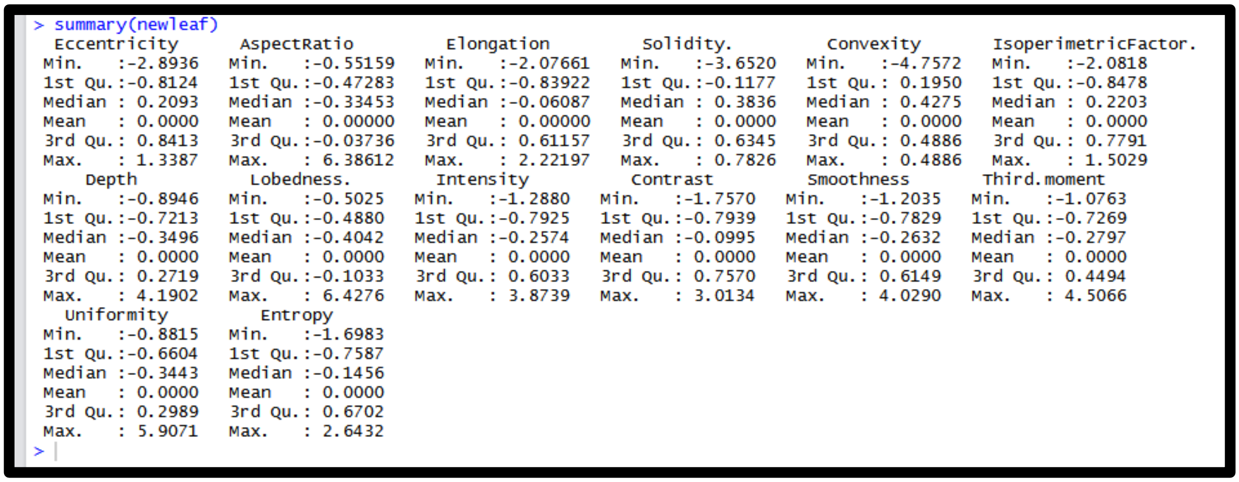
```
> str(newleaf)
'data.frame':	340 obs. of  16 variables:
 $ Species             : int  1 1 1 1 1 1 1 1 1 1 ...
 $ SpecimenNumber      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Eccentricity        : num  0.727 0.742 0.767 0.738 0.823 ...
 $ AspectRatio         : num  1.47 1.53 1.57 1.46 1.77 ...
 $ Elongation          : 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 ...
 $ Third.moment        : 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 ...
> 
```

Figure 1. Initial Data Structure of Leaf Dataset.



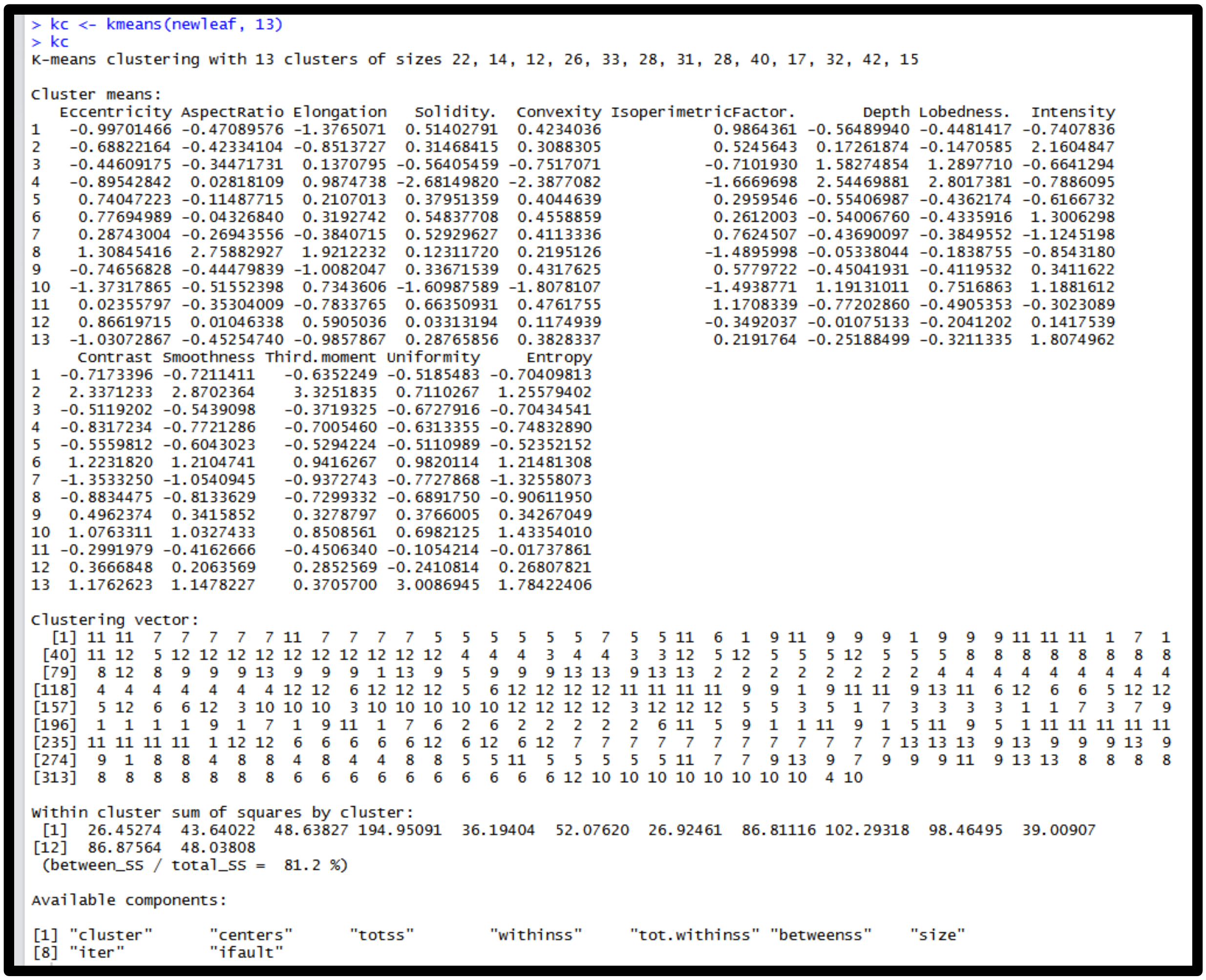
```
> summary(newleaf)
  Eccentricity     AspectRatio       Elongation        Solidity.        Convexity      IsoperimetricFactor.
 Min.   :0.1171   Min.   : 1.007   Min.   :0.1076   Min.   :0.4855   Min.   :0.3965   Min.   :0.07838     
 1st Qu.:0.5506   1st Qu.: 1.211   1st Qu.:0.3496   1st Qu.:0.8907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

Figure 2. Descriptive Statistics After Removing Variables.



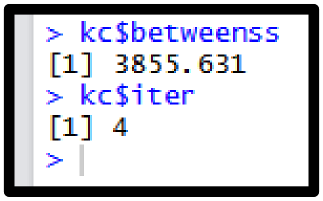
```
> summary(newleaf)
  Eccentricity        AspectRatio          Elongation           Solidity.           Convexity         IsoperimetricFactor.
 Min.   :-2.8936   Min.   :-0.55159   Min.   :-2.07661   Min.   :-3.6520   Min.   :-4.7572   Min.   :-2.0818
 1st Qu.:-0.8124   1st Qu.:-0.47283   1st Qu.:-0.83922   1st Qu.:-0.1177   1st Qu.: 0.1950   1st Qu.:-0.8478
 Median : 0.2093   Median :-0.33453   Median :-0.06087   Median : 0.3836   Median : 0.4275   Median : 0.2203
 Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.8413   3rd Qu.:-0.03736   3rd Qu.: 0.61157   3rd Qu.: 0.6345   3rd Qu.: 0.4886   3rd Qu.: 0.7791
 Max.   : 1.3387   Max.   : 6.38612   Max.   : 2.22197   Max.   : 0.7826   Max.   : 0.4886   Max.   : 1.5029
     Depth            Lobedness.         Intensity           Contrast          Smoothness        Third.moment
 Min.   :-0.8946   Min.   :-0.5025   Min.   :-1.2880   Min.   :-1.7570   Min.   :-1.2035   Min.   :-1.0763
 1st Qu.:-0.7213   1st Qu.:-0.4880   1st Qu.:-0.7925   1st Qu.:-0.7939   1st Qu.:-0.7829   1st Qu.:-0.7269
 Median :-0.3496   Median :-0.4042   Median :-0.2574   Median :-0.0995   Median :-0.2632   Median :-0.2797
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.2719   3rd Qu.:-0.1033   3rd Qu.: 0.6033   3rd Qu.: 0.7570   3rd Qu.: 0.6149   3rd Qu.: 0.4494
 Max.   : 4.1902   Max.   : 6.4276   Max.   : 3.8739   Max.   : 3.0134   Max.   : 4.0290   Max.   : 4.5066
   Uniformity         Entropy
 Min.   :-0.8815   Min.   :-1.6983
 1st Qu.:-0.6604   1st Qu.:-0.7587
 Median :-0.3443   Median :-0.1456
 Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.2989   3rd Qu.: 0.6702
 Max.   : 5.9071   Max.   : 2.6432
> 
```

Figure 3. Descriptive Statistics After Scaling Variables.



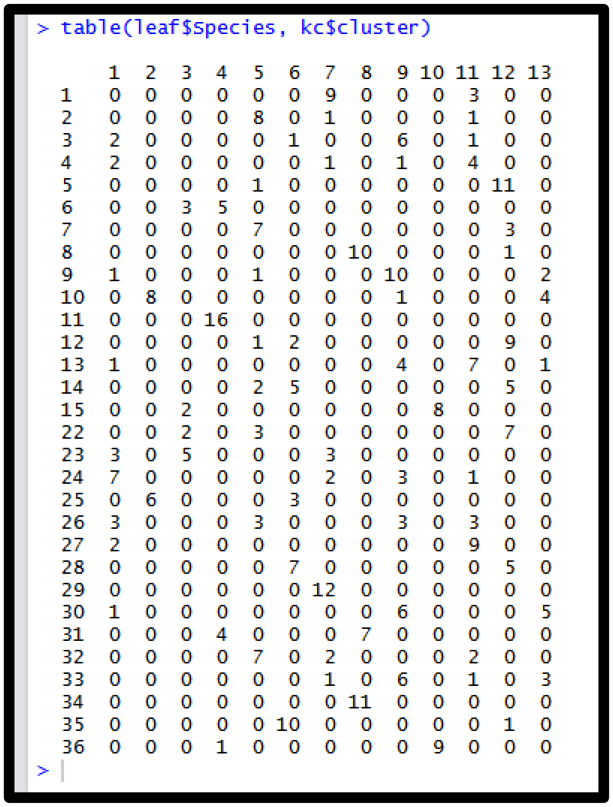
```
> kc <- kmeans(newleaf, 13)
> kc
K-means clustering with 13 clusters of sizes 22, 14, 12, 26, 33, 28, 31, 28, 40, 17, 32, 42, 15

Cluster means:
   Eccentricity AspectRatio Elongation   Solidity.  Convexity IsoperimetricFactor.       Depth Lobedness.  Intensity
1   -0.99701466 -0.47089576 -1.3765071  0.51402791  0.4234036            0.9864361 -0.56489940 -0.4481417 -0.7407836
2   -0.68822164 -0.42334104 -0.8513727  0.31468415  0.3088305            0.5245643  0.17261874 -0.1470585  2.1604847
3   -0.44609175 -0.34471731  0.1370795 -0.56405459 -0.7517071           -0.7101930  1.58274854  1.2897710 -0.6641294
4   -0.89542842  0.02818109  0.9874738 -2.68149820 -2.3877082           -1.6669698  2.54469881  2.8017381 -0.7886095
5    0.74047223 -0.11487715  0.2107013  0.37951359  0.4044639            0.2959546 -0.55406987 -0.4362174 -0.6166732
6    0.77694989 -0.04326840  0.3192742  0.54837708  0.4558859            0.2612003 -0.54006760 -0.4335916  1.3006298
7    0.28743004 -0.26943556 -0.3840715  0.52929627  0.4113336            0.7624507 -0.43690097 -0.3849552 -1.1245198
8    1.30845416  2.75882927  1.9212232  0.12311720  0.2195126           -1.4895998 -0.05338044 -0.1838755 -0.8543180
9   -0.74656828 -0.44479839 -1.0082047  0.33671539  0.4317625            0.5779722 -0.45041931 -0.4119532  0.3411622
10  -1.37317865 -0.51552398  0.7343606 -1.60987589 -1.8078107           -1.4938771  1.19131011  0.7516863  1.1881612
11   0.02355797 -0.35304009 -0.7833765  0.66350931  0.4761755            1.1708339 -0.77202860 -0.4905353 -0.3023089
12   0.86619715  0.01046338  0.5905036  0.03313194  0.1174939           -0.3492037 -0.01075133 -0.2041202  0.1417539
13  -1.03072867 -0.45254740 -0.9857867  0.28765856  0.3828337            0.2191764 -0.25188499 -0.3211335  1.8074962
     Contrast Smoothness Third.moment Uniformity     Entropy
1  -0.7173396 -0.7211411   -0.6352249 -0.5185483 -0.70409813
2   2.3371233  2.8702364    3.3251835  0.7110267  1.25579402
3  -0.5119202 -0.5439098   -0.3719325 -0.6727916 -0.70434541
4  -0.8317234 -0.7721286   -0.7005460 -0.6313355 -0.74832890
5  -0.5559812 -0.6043023   -0.5294224 -0.5110989 -0.52352152
6   1.2231820  1.2104741    0.9416267  0.9820114  1.21481308
7  -1.3533250 -1.0540945   -0.9372743 -0.7727868 -1.32558073
8  -0.8834475 -0.8133629   -0.7299332 -0.6891750 -0.90611950
9   0.4962374  0.3415852    0.3278797  0.3766005  0.34267049
10  1.0763311  1.0327433    0.8508561  0.6982125  1.43354010
11 -0.2991979 -0.4162666   -0.4506340 -0.1054214 -0.01737861
12  0.3666848  0.2063569    0.2852569 -0.2410814  0.26807821
13  1.1762623  1.1478227    0.3705700  3.0086945  1.78422406

Clustering vector:
  [1] 11 11  7  7  7  7  7 11  7  7  7  7  5  5  5  5  5  5  7  5  5 11  6  1  9 11  9  9  9  1  9  9  9 11 11 11  1  7  1
 [40] 11 12  5 12 12 12 12 12 12 12 12 12 12  4  4  4  3  4  4  3  3 12  5 12  5  5  5 12  5  5  5  8  8  8  8  8  8  8  8
 [79]  8 12  8  9  9  9 13  9  9  9  1 13  9  5  9  9  9 13 13  9 13 13  2  2  2  2  2  2  2  2  4  4  4  4  4  4  4  4  4
[118]  4  4  4  4  4  4  4 12 12  6 12 12 12  5  6 12 12 12 12 11 11 11 11  9  9  1  9 11 11  9 13 11  6 12  6  6  5 12 12
[157]  5 12  6  6 12  3 10 10 10  3 10 10 10 10 10 12 12 12 12  3 12 12 12  5  5  3  5  1  7  3  3  3  3  1  1  7  3  7  9
[196]  1  1  1  1  9  1  7  1  9 11  1  7  6  2  6  2  2  2  2  2  6 11  5  9  1  1 11  9  1  5 11  9  5  1 11 11 11 11 11
[235] 11 11 11 11  1 12 12  6  6  6  6  6 12  6 12  6 12  7  7  7  7  7  7  7  7  7  7  7  7 13 13 13  9 13  9  9  9 13  9
[274]  9  1  8  8  4  8  8  4  8  4  4  8  8  5  5 11  5  5  5  5  5 11  7  7  9 13  9  7  9  9  9 11  9 13 13  8  8  8  8
[313]  8  8  8  8  8  8  8  6  6  6  6  6  6  6  6  6  6 12 10 10 10 10 10 10 10 10  4 10

Within cluster sum of squares by cluster:
 [1]  26.45274  43.64022  48.63827 194.95091  36.19404  52.07620  26.92461  86.81116 102.29318  98.46495  39.00907
[12]  86.87564  48.03808
 (between_SS / total_SS =  81.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"        
[8] "iter"         "ifault"      
```

Figure 4. Properties of the K-Means Clustering Model (K = 13).



```
> kc$betweenss
[1] 3855.631
> kc$iter
[1] 4
> 
```

Figure 5. Additional Properties of the Clustering Model (K = 13).



```
> table(leaf$Species, kc$cluster)
    
      1  2  3  4  5  6  7  8  9 10 11 12 13
  1   0  0  0  0  0  0  9  0  0  0  3  0  0
  2   0  0  0  0  8  0  1  0  0  0  1  0  0
  3   2  0  0  0  0  1  0  0  6  0  1  0  0
  4   2  0  0  0  0  0  1  0  1  0  4  0  0
  5   0  0  0  0  1  0  0  0  0  0  0 11  0
  6   0  0  3  5  0  0  0  0  0  0  0  0  0
  7   0  0  0  0  7  0  0  0  0  0  0  3  0
  8   0  0  0  0  0  0  0 10  0  0  0  1  0
  9   1  0  0  0  1  0  0  0 10  0  0  0  2
  10  0  8  0  0  0  0  0  0  1  0  0  0  4
  11  0  0  0 16  0  0  0  0  0  0  0  0  0
  12  0  0  0  0  1  2  0  0  0  0  0  9  0
  13  1  0  0  0  0  0  0  0  4  0  7  0  1
  14  0  0  0  0  2  5  0  0  0  0  0  5  0
  15  0  0  2  0  0  0  0  0  0  8  0  0  0
  22  0  0  2  0  3  0  0  0  0  0  0  7  0
  23  3  0  5  0  0  0  3  0  0  0  0  0  0
  24  7  0  0  0  0  0  2  0  3  0  1  0  0
  25  0  6  0  0  0  3  0  0  0  0  0  0  0
  26  3  0  0  0  3  0  0  0  3  0  3  0  0
  27  2  0  0  0  0  0  0  0  0  0  9  0  0
  28  0  0  0  0  0  7  0  0  0  0  0  5  0
  29  0  0  0  0  0  0 12  0  0  0  0  0  0
  30  1  0  0  0  0  0  0  0  6  0  0  0  5
  31  0  0  0  4  0  0  0  7  0  0  0  0  0
  32  0  0  0  0  7  0  2  0  0  0  2  0  0
  33  0  0  0  0  0  0  1  0  6  0  1  0  3
  34  0  0  0  0  0  0  0 11  0  0  0  0  0
  35  0  0  0  0  0 10  0  0  0  0  0  1  0
  36  0  0  0  1  0  0  0  0  0  9  0  0  0
> 
```

Figure 6. Cross-Table Comparing Clusters to Actual Classes (K = 13).

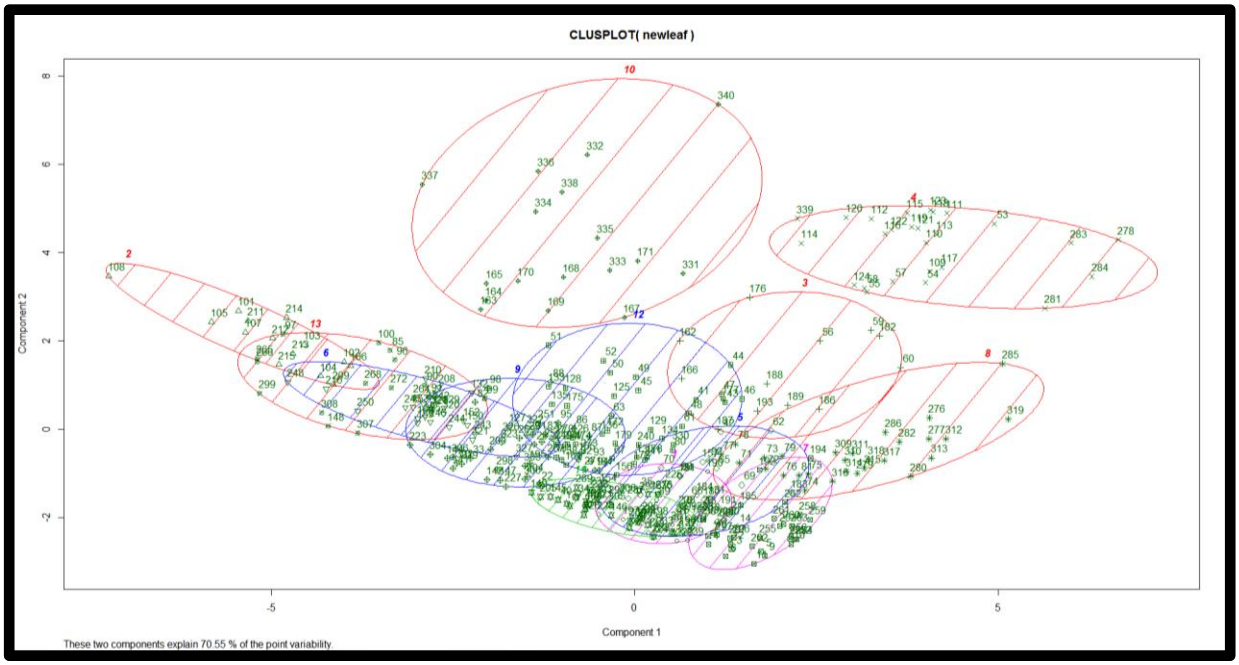


Figure 7. Visualization of the K-Means Clustering Model (K = 13).

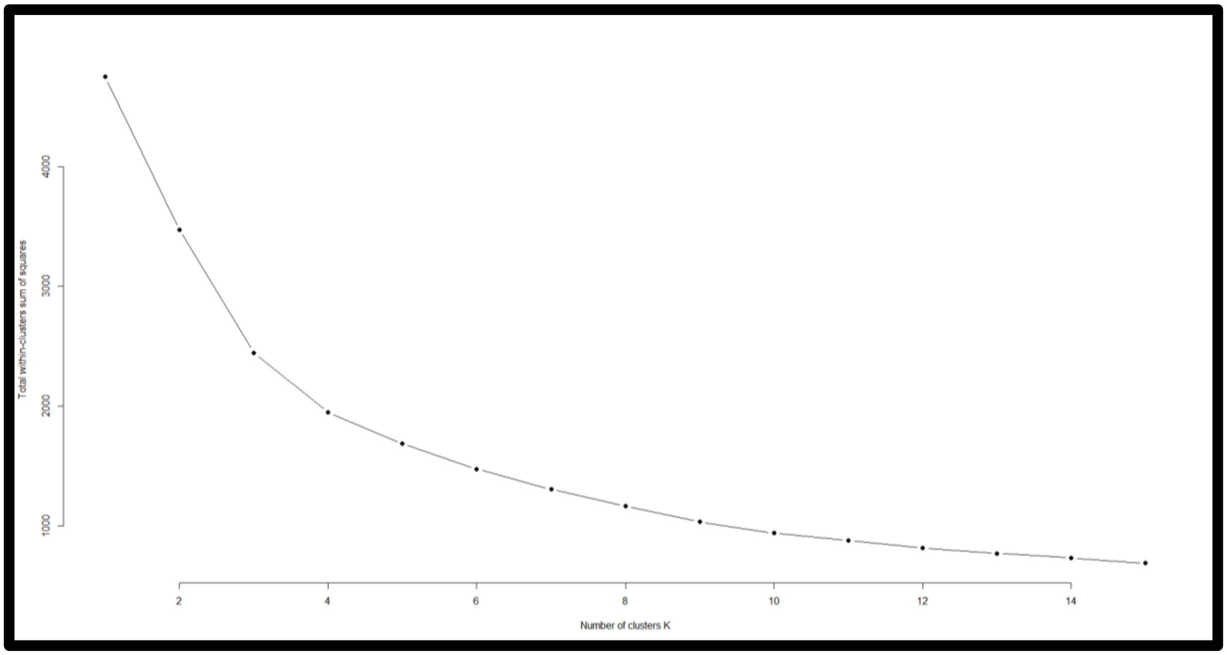
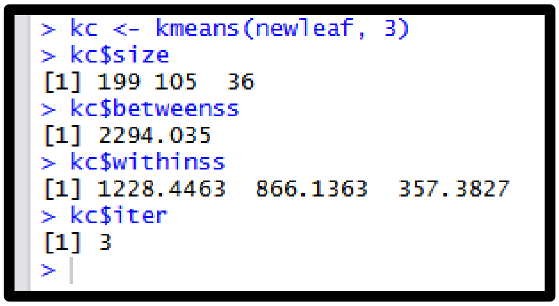
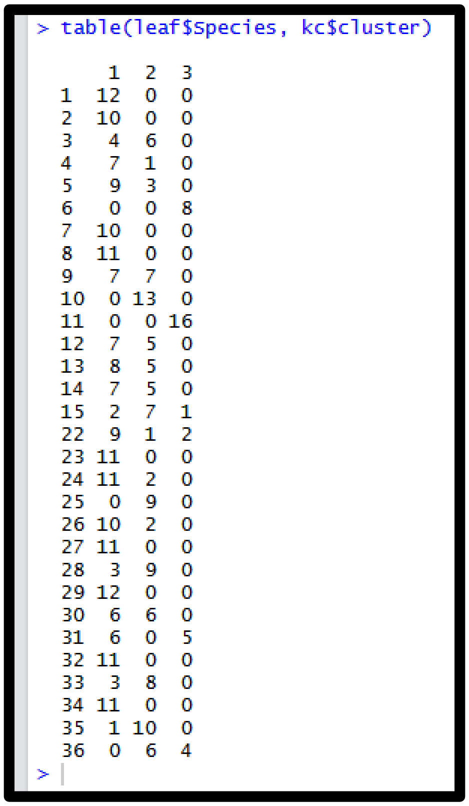


Figure 8. “Elbow Method” Visualization on Leaf Dataset.



```
> kc <- kmeans(newleaf, 3)
> kc$size
[1] 199 105  36
> kc$betweenss
[1] 2294.035
> kc$withinss
[1] 1228.4463  866.1363  357.3827
> kc$iter
[1] 3
> 
```

Figure 9. Major Properties of the K-Means Clustering Model (K = 3).



```
> table(leaf$Species, kc$cluster)
    
      1  2  3
  1  12  0  0
  2  10  0  0
  3   4  6  0
  4   7  1  0
  5   9  3  0
  6   0  0  8
  7  10  0  0
  8  11  0  0
  9   7  7  0
  10  0 13  0
  11  0  0 16
  12  7  5  0
  13  8  5  0
  14  7  5  0
  15  2  7  1
  22  9  1  2
  23 11  0  0
  24 11  2  0
  25  0  9  0
  26 10  2  0
  27 11  0  0
  28  3  9  0
  29 12  0  0
  30  6  6  0
  31  6  0  5
  32 11  0  0
  33  3  8  0
  34 11  0  0
  35  1 10  0
  36  0  6  4
> 
```

Figure 10. Cross-Table Comparing Clusters to Actual Classes (K = 3).

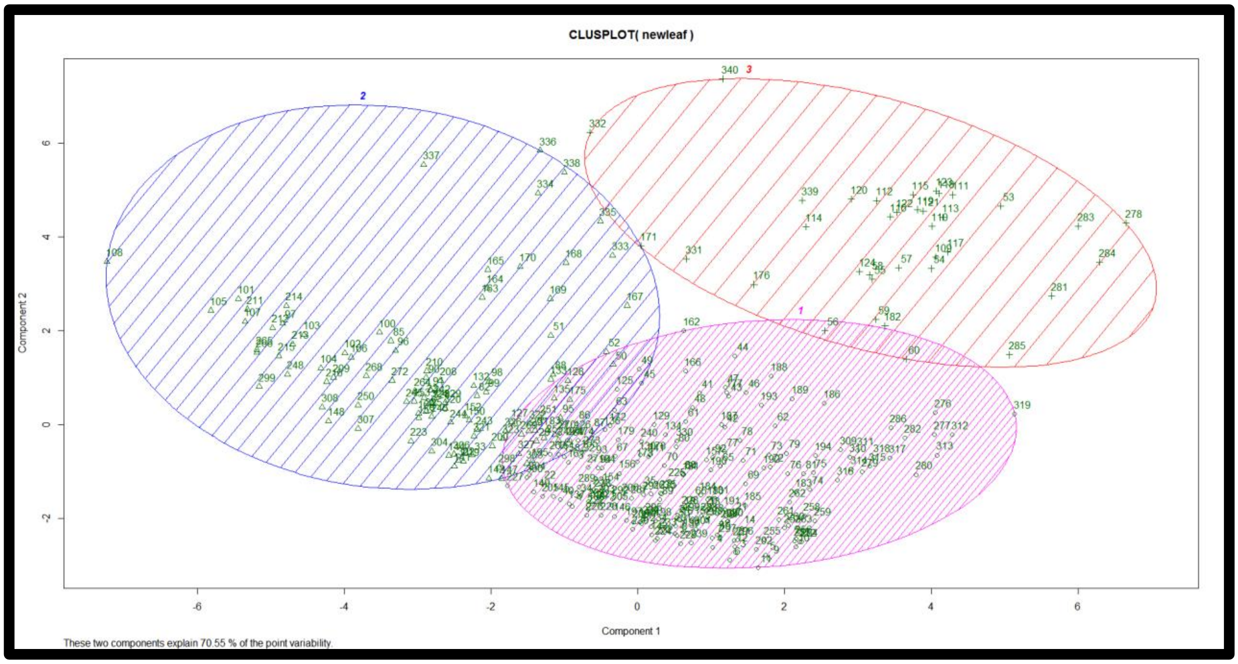


Figure 11. Visualization of the K-Means Clustering Model (K = 3).

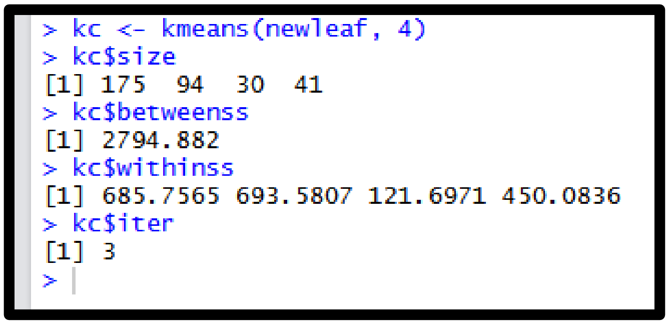
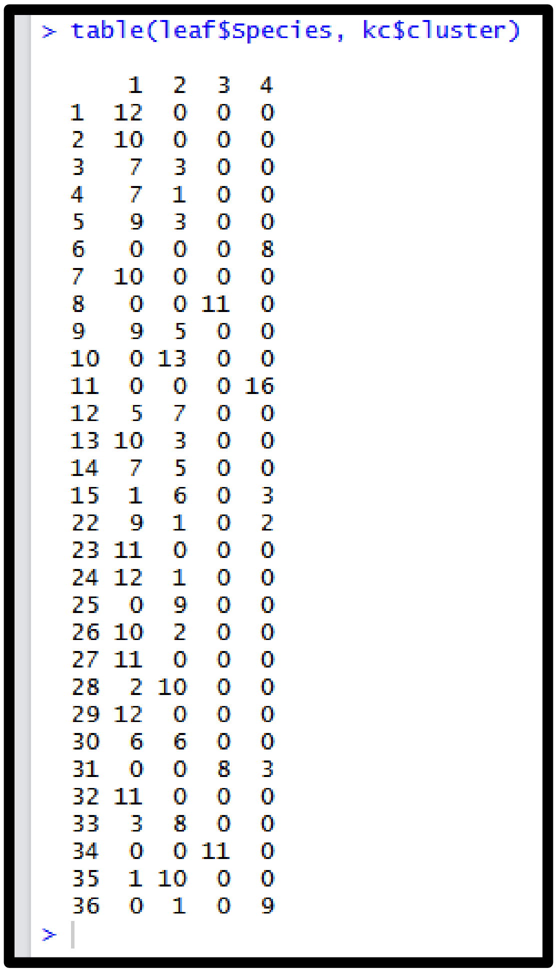

```
> kc <- kmeans(newleaf, 4)
> kc$size
[1] 175  94  30  41
> kc$betweenss
[1] 2794.882
> kc$withinss
[1] 685.7565 693.5807 121.6971 450.0836
> kc$iter
[1] 3
> 
```

Figure 12. Major Properties of the K-Means Clustering Model (K = 4).



```
> table(leaf$Species, kc$cluster)

      1  2  3  4
  1  12  0  0  0
  2  10  0  0  0
  3   7  3  0  0
  4   7  1  0  0
  5   9  3  0  0
  6   0  0  0  8
  7  10  0  0  0
  8   0  0 11  0
  9   9  5  0  0
  10  0 13  0  0
  11  0  0  0 16
  12  5  7  0  0
  13 10  3  0  0
  14  7  5  0  0
  15  1  6  0  3
  22  9  1  0  2
  23 11  0  0  0
  24 12  1  0  0
  25  0  9  0  0
  26 10  2  0  0
  27 11  0  0  0
  28  2 10  0  0
  29 12  0  0  0
  30  6  6  0  0
  31  0  0  8  3
  32 11  0  0  0
  33  3  8  0  0
  34  0  0 11  0
  35  1 10  0  0
  36  0  1  0  9
> 
```

Figure 13. Cross-Table Comparing Clusters to Actual Classes (K = 4).

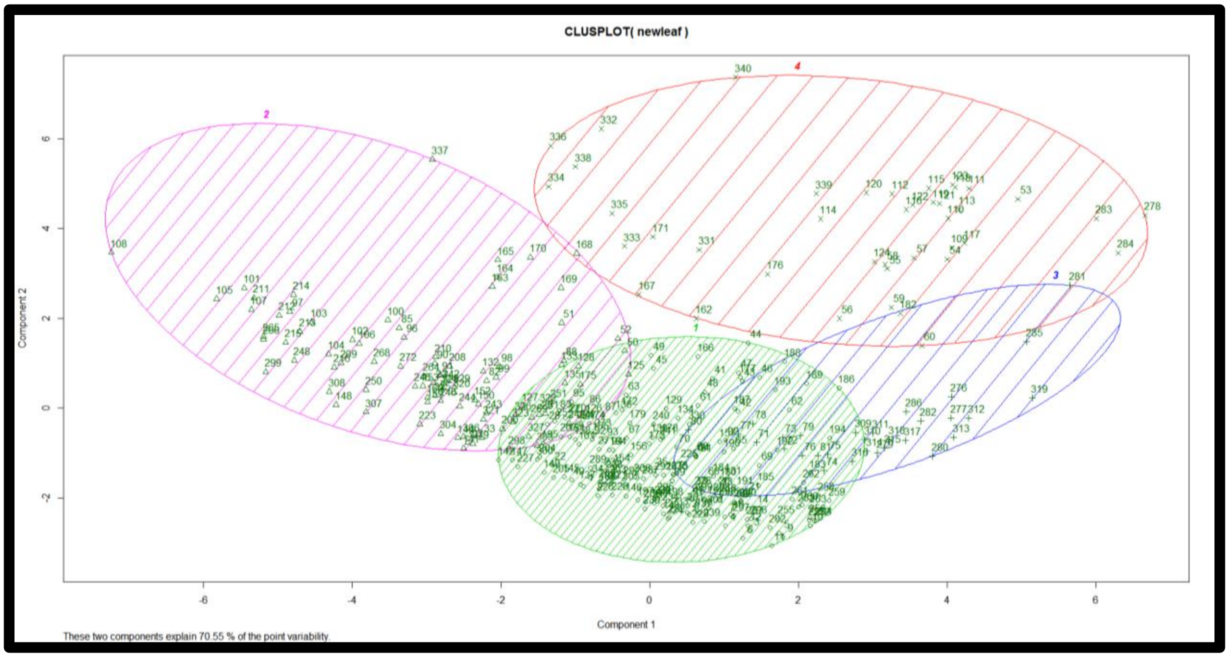
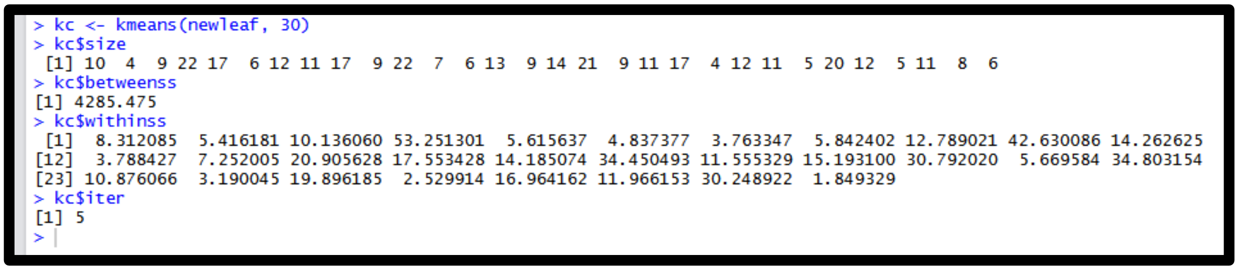
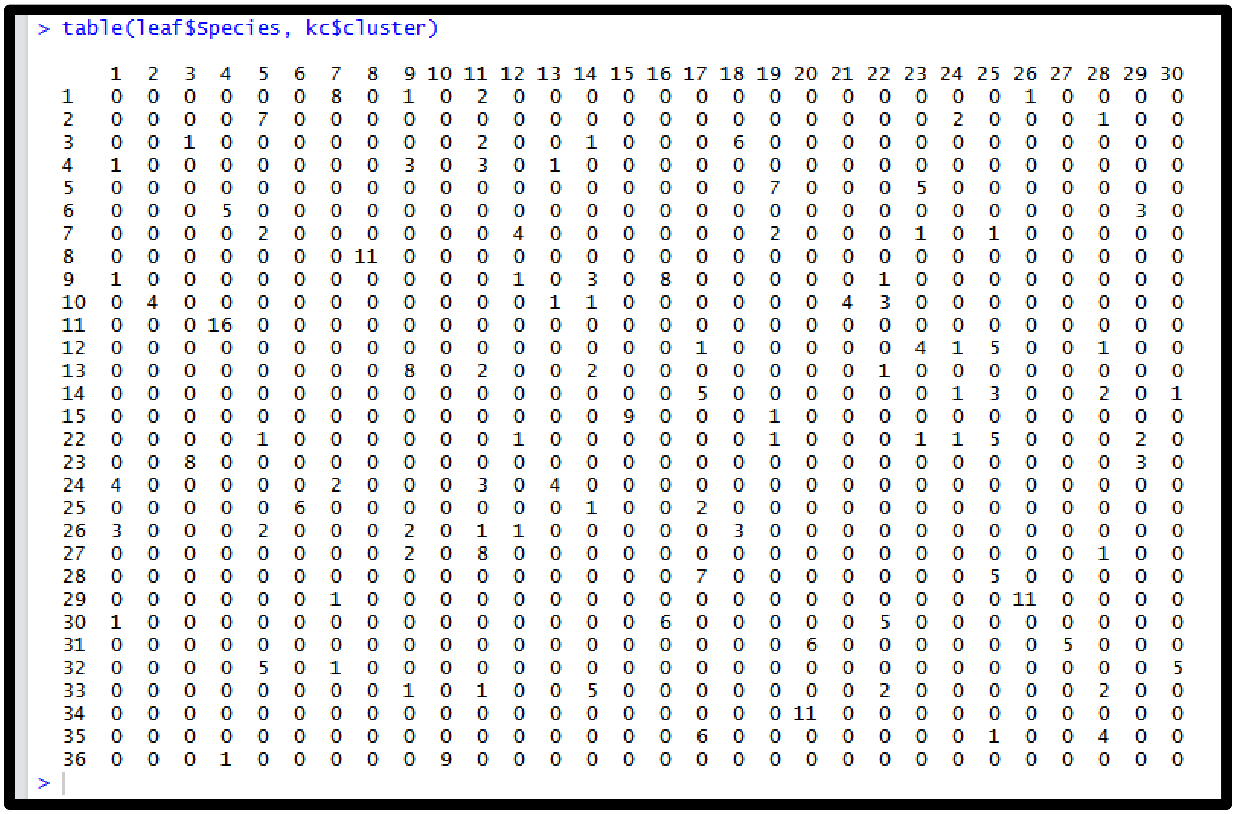


Figure 14. Visualization of the K-Means Clustering Model (K = 4).



```
> kc <- kmeans(newleaf, 30)
> kc$size
 [1] 10  4  9 22 17  6 12 11 17  9 22  7  6 13  9 14 21  9 11 17  4 12 11  5 20 12  5 11  8  6
> kc$betweenss
[1] 4285.475
> kc$withinss
 [1]  8.312085  5.416181 10.136060 53.251301  5.615637  4.837377  3.763347  5.842402 12.789021 42.630086 14.262625
[12]  3.788427  7.252005 20.905628 17.553428 14.185074 34.450493 11.555329 15.193100 30.792020  5.669584 34.803154
[23] 10.876066  3.190045 19.896185  2.529914 16.964162 11.966153 30.248922  1.849329
> kc$iter
[1] 5
> 
```

Figure 15. Major Properties of the K-Means Clustering Model (K = 30).



```
> table(leaf$Species, kc$cluster)

        1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30
  1     0   0   0   0   0   0   8   0   1   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
  2     0   0   0   0   7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   1   0   0
  3     0   0   1   0   0   0   0   0   0   0   2   0   0   1   0   0   0   6   0   0   0   0   0   0   0   0   0   0   0   0
  4     1   0   0   0   0   0   0   0   3   0   3   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  5     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   7   0   0   0   5   0   0   0   0   0   0   0
  6     0   0   0   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0
  7     0   0   0   0   2   0   0   0   0   0   0   4   0   0   0   0   0   0   2   0   0   0   1   0   1   0   0   0   0   0
  8     0   0   0   0   0   0   0  11   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  9     1   0   0   0   0   0   0   0   0   0   0   1   0   3   0   8   0   0   0   0   0   1   0   0   0   0   0   0   0   0
  10    0   4   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   0   0   0   4   3   0   0   0   0   0   0   0   0
  11    0   0   0  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  12    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   4   1   5   0   0   1   0   0
  13    0   0   0   0   0   0   0   0   8   0   2   0   0   2   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
  14    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5   0   0   0   0   0   0   1   3   0   0   2   0   1
  15    0   0   0   0   0   0   0   0   0   0   0   0   0   0   9   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
  22    0   0   0   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0   1   0   0   0   1   1   5   0   0   0   2   0
  23    0   0   8   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0
  24    4   0   0   0   0   0   2   0   0   0   3   0   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  25    0   0   0   0   0   6   0   0   0   0   0   0   0   1   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0
  26    3   0   0   0   2   0   0   0   2   0   1   1   0   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0
  27    0   0   0   0   0   0   0   0   2   0   8   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0
  28    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   7   0   0   0   0   0   0   0   5   0   0   0   0   0
  29    0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  11   0   0   0   0
  30    1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   6   0   0   0   0   0   5   0   0   0   0   0   0   0   0
  31    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   6   0   0   0   0   0   0   5   0   0   0
  32    0   0   0   0   5   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5
  33    0   0   0   0   0   0   0   0   1   0   1   0   0   5   0   0   0   0   0   0   0   2   0   0   0   0   0   2   0   0
  34    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  11   0   0   0   0   0   0   0   0   0   0
  35    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   6   0   0   0   0   0   0   0   1   0   0   4   0   0
  36    0   0   0   1   0   0   0   0   0   9   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
> 
```

Figure 16. Cross-Table Comparing Clusters to Actual Classes (K = 30).

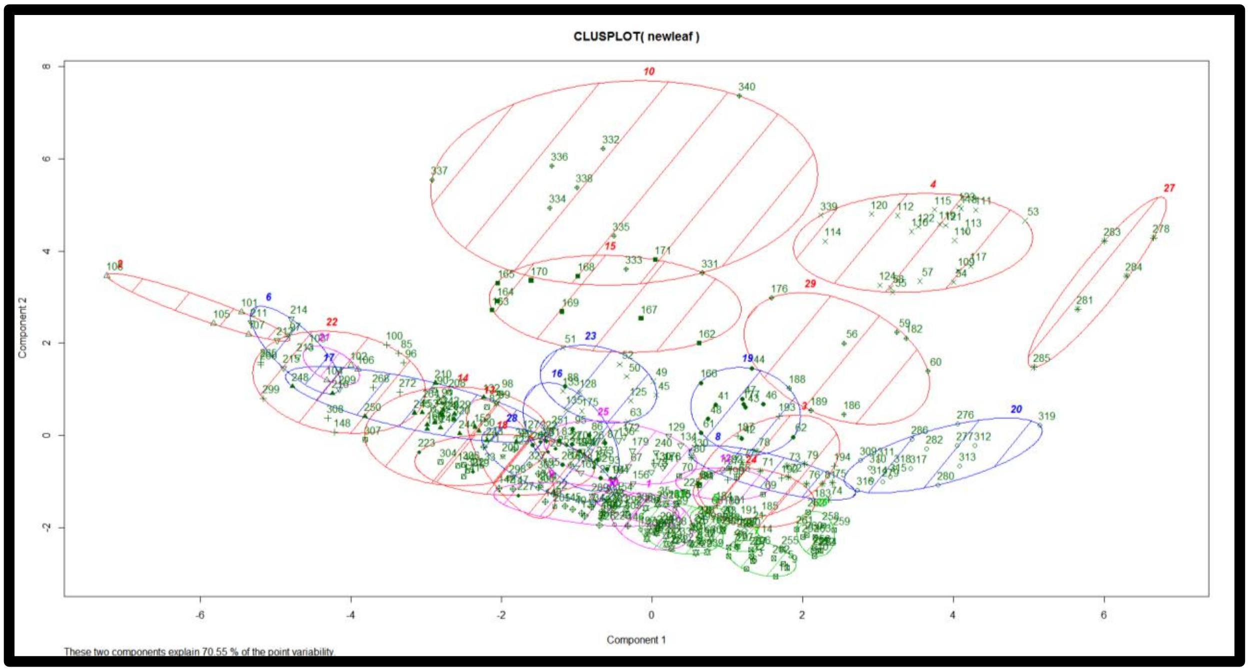


Figure 17. Visualization of the K-Means Clustering Model (K = 30).

k	Number of instances in each cluster	Between clusters sum of squares	Within clusters sum of squares	Number of iterations	Percent of matching classes
13	22, 14, 12, 26, 33, 28, 31, 28, 40, 17, 32, 42, 15	3855.63	26.45, 43.64, 48.64, 194.95, 36.19, 52.08, 26.92, 86.81, 102.29, 98.46, 39.01, 86.88, 48.04	4	69.1%
3	199, 105, 36	2294.04	1228.45, 866.14, 357.38	3	81.8%
4	175, 94, 30, 41	2794.88	685.76, 693.58, 121.70, 450.08	3	85.0%
30	10, 4, 9, 22, 17, 6, 12, 11, 17, 9, 22, 7, 6, 13, 9, 14, 21, 9, 11, 17, 4, 12, 11, 5, 20, 12, 5, 11, 8, 6	4285.48	8.31, 5.42, 10.14, 53.25, 5.62, 4.84, 3.76, 5.84, 12.79, 42.63, 14.26, 3.79, 7.25, 20.91, 17.55, 14.19, 34.45, 11.56, 15.19, 30.79, 5.67, 34.80, 10.88, 3.19, 19.90, 2.53, 16.96, 11.97, 30.25, 1.85	5	60.6%

Figure 18. Table for Evaluating Results of Clustering Models.