Math 466 Advanced Big Data Analysis

Hero Ozagho
Claremont Graduate University
1237 North Dartmouth Avenue, Claremont, CA 91711, USA

March 19, 2021

ABSTRACT

This paper gives complete application of machine learning analysis to the Iris flowers data provided by the University of California Irvine Machine Learning Repository. The goal is to design a model that gives classification of iris flowers from sepal and petal dimensions measurements of three species (setosa, versicolor or virginica). The iris flower data contains 3 classes of 50 instances each of setosa, versicolor and virginica .

Part 1: Fundamentals

1.1 Data Type

The iris data set is a multivariate data used to quantify the variation of Iris flowers. The dataset consists of 50 samples from each of the three species of the Iris flowers such as Iris setosa, Iris virginica, and Iris versicolor. Note must be taken that four features were measured from each sample in terms of the length and the width of sepal and petal. Based on the combination of the four features, a linear model used to distinguish the species from each other. Therefore, the iris dataset contains four measurements for 150 flowers representing three species mentioned earlier in this paragraph. The dataset interpreted as a data that contains three (3) classes of fifty instances each, where each class refers to a type of setosa, virginica and versicolor iris plant. One class is linearly separable from the other. The data contains the following attributes:

1. Sepal length in cm 2. Sepal width in cm 3. Petal length in cm 4. Petal width in cm 5. Class: - Iris setosa - Iris versicolor - Iris virginica

The iris dataset is loaded onto R from .csv extension file. There is need to know the dimensions of the iris dataset, followed by the attributes, then levels of the class attributes and followed by the analysis of the instances in each class and the statistical summary of all attributes.

1.2 Data Transformation

Yes, data transformation is necessary. The first four column describing the length and width of sepals and petals of the iris data is not a strongly skewed data. Therefore, log-transformation of the data does not really change the results much. It happened that the code used for the log transformation to the continuous variables is set to center and scale to prcomp to standardize the variables prior to performed PCA to get insight of the general structure of the iris data set. Therefore, the iris data were centered, scaled and log-transform to filter off some effects, which could dominate the PCA. The algorithm of this PCA in turn find the rotation of each PC to minimize the squared residuals, namely the sum of squared perpendicular distances from any sample to the PCs.

1.3 Raw Data Visualization

The inputs are double while the class value is a factor. This gives an idea of the type of attributes one is dealing with such that one can summarize the data and know the type of transformation to use in order to prepare the iris data for the necessary model.

In figure 5, one can clearly see one well separated group, two overlapping groups. Moreso,covariates for lengths and widths of sepals and petals in the plot.

Boxplot

In figure 6, one can see in the boxplot the difference in distribution of each attribute by class value. The Gaussian-like distribution curve of each attribute are displayed for sepal length, sepal width, petal length and petal width. Sepal length and sepal width seem to spread evenly about their own centers. Petal lengths and petal widths' values are more spread below values lower than the average value. It seems like each species seem to have its own set of petal length and width values apart from each other, which is good

1.4 Data Pre-processing

The first step is the sampling and pre-processing. The iris data need to be scaled, limited or amplified to match the requirements of the objectives for the iris flower data. The iris flower data was standardized for performing principal component analysis. Similarly, for the k-means clustering centered and scaled iris data was shown to be quite suitable as well for the iris flower data.

```
 > dim(iris)
[1] 150   5

 > sapply(iris, class)
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
   "numeric"    "numeric"    "numeric"    "numeric"     "factor"

> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa

> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
```

```
 virginica :50
```

```
> iris$Class
  [1] setosa     setosa     setosa     setosa     setosa     setosa
  [7] setosa     setosa     setosa     setosa     setosa     setosa
 [13] setosa     setosa     setosa     setosa     setosa     setosa
 [19] setosa     setosa     setosa     setosa     setosa     setosa
 [25] setosa     setosa     setosa     setosa     setosa     setosa
 [31] setosa     setosa     setosa     setosa     setosa     setosa
 [37] setosa     setosa     setosa     setosa     setosa     setosa
 [43] setosa     setosa     setosa     setosa     setosa     setosa
 [49] setosa     setosa     versicolor versicolor versicolor versicolor
 [55] versicolor versicolor versicolor versicolor versicolor versicolor
 [61] versicolor versicolor versicolor versicolor versicolor versicolor
 [67] versicolor versicolor versicolor versicolor versicolor versicolor
 [73] versicolor versicolor versicolor versicolor versicolor versicolor
 [79] versicolor versicolor versicolor versicolor versicolor versicolor
 [85] versicolor versicolor versicolor versicolor versicolor versicolor
 [91] versicolor versicolor versicolor versicolor versicolor versicolor
 [97] versicolor versicolor versicolor versicolor virginica  virginica
[103] virginica  virginica  virginica  virginica  virginica  virginica
[109] virginica  virginica  virginica  virginica  virginica  virginica
[115] virginica  virginica  virginica  virginica  virginica  virginica
[121] virginica  virginica  virginica  virginica  virginica  virginica
[127] virginica  virginica  virginica  virginica  virginica  virginica
[133] virginica  virginica  virginica  virginica  virginica  virginica
[139] virginica  virginica  virginica  virginica  virginica  virginica
[145] virginica  virginica  virginica  virginica  virginica  virginica
Levels: setosa versicolor virginica
```

One can deduce from the below exploration that the petal length and petal width were somewhat similar among the same species but widely varied between different species.

```
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1
```

In figure 5, one can clearly see one well separated group, two overlapping groups. More so,covariates for lengths and widths of sepals and petals in the plot.

Part 2: Learning from Data

2.1 Unsupervised Learning :Hierarchical Clustering

The basic principle of hierarchical clustering creates a hierarchy of clusters, which represents a tree structure called a dendrogram. The tree consists of a single cluster containing all observations and the leaves correspond to individual observations.

Refer to figures 11 and 12, these visualizations easily demonstrates how the separation of the hierarchical clustering is very good with the Setosa species, but misses in labeling many Versicolor species as Virginica. The hanging of the tree also helps to locate extreme observations. Similarly, one can see that observation virginica is not very similar to the Versicolor species, but still, it is among them. Also, Versicolor is located too much "within" the group of Virginica flowers. On the other hand, as one flower is remove from each group, and repeat clustering, the same observation takes place with Setosa clearly separates out compare to versicolor and virginica that tend to overlaps.

In figure 10, on the heatmap one see how sepal.length, petal.length and pedal.width showed very low value to show clear distinction from sepal.width in that all of the species , that is setosa , versicolor and virginica.

Figure 22, gives clear picture of the features that are consistent as noted earlier in other figures related PCA: Petal Length, Petal Width and Sepal Length are somewhat in the same domain, within the circle.

```
> summary(X)
  Sepal.Length       Sepal.Width       Petal.Length      Petal.Width
 Min.   :-1.86378   Min.   :-2.4258   Min.   :-1.5623   Min.   :-1.4422
 1st Qu.:-0.89767   1st Qu.:-0.5904   1st Qu.:-1.2225   1st Qu.:-1.1799
 Median :-0.05233   Median :-0.1315   Median : 0.3354   Median : 0.1321
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.67225   3rd Qu.: 0.5567   3rd Qu.: 0.7602   3rd Qu.: 0.7880
 Max.   : 2.48370   Max.   : 3.0805   Max.   : 1.7799   Max.   : 1.7064

> head(X)
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]   -0.8976739  1.01560199    -1.335752   -1.311052
[2,]   -1.1392005 -0.13153881    -1.335752   -1.311052
[3,]   -1.3807271  0.32731751    -1.392399   -1.311052
[4,]   -1.5014904  0.09788935    -1.279104   -1.311052
[5,]   -1.0184372  1.24503015    -1.335752   -1.311052
[6,]   -0.5353840  1.93331463    -1.165809   -1.048667
```

On can deduce from the above that both Petal.Length and Sepal.Length are highly correlated not as Petal.Length and Petal Width. However, Sepal.Width is completely off and not correlated with any of the others.

```
> plot(PC$x[,1],PC$x[,2],col=color,pch=point)
> legend("topright",legend=levels(species), pch=unique(point))
> H = hclust(dist(X))
> cl = cutree(H,k=3)
> plot(H,labels=species,hang=-1,cex=0.75)
```

```
> points(x=1:nrow(X),y=rep(0,nrow(X)),pch=point[H$order],col=cl[H$order])


> table(cl, species)
   species
cl  setosa versicolor virginica
  1     49          0         0
  2      1         21         2
  3      0         29        48


> idx = c(-41,-98,-144)
> X1 = X[idx,]
> color1 = color[idx]
> point1 = point[idx]
> species1 = species[idx]
> H1 = hclust(dist(X1))
> cl1 = cutree(H1,k=3)
> plot(H1,labels=species1,hang=-1,cex=0.75, main="Cluster Dendrogram after Rem
> points(x=1:nrow(X1),y=rep(0,nrow(X1)),pch=point1[H1$order],col=cl1[H1$order]


> table(cl[idx],cl1)
   cl1
     1  2  3
  1 41  7  0
  2 24  0  0
  3 27  0 48
> table(species1,cl1)
            cl1
species1      1  2  3
  setosa     42  7  0
  versicolor 36  0 13
  virginica  14  0 35


> H2 = hclust(dist(X,method="maximum"))
> cl2 = cutree(H2,k=3)
> table(cl2, species)
   species
cl2 setosa versicolor virginica
  1     37         40        20
  2     13          0         0
  3      0         10        30

Correlation measure
> D = (1-cor(t(X)))/2
```

```
> D = as.dist(D)
> H3 = hclust(D)
> cl3 = cutree(H3,k=3)
> table(cl3, species)
   species
cl3 setosa versicolor virginica
  1     49          2         0
  2      1         10        14
  3      0         38        36

Bootstrap (r = 0.5)... Done.
Bootstrap (r = 0.6)... Done.
Bootstrap (r = 0.7)... Done.
Bootstrap (r = 0.8)... Done.
Bootstrap (r = 0.9)... Done.
Bootstrap (r = 1.0)... Done.
Bootstrap (r = 1.1)... Done.
Bootstrap (r = 1.2)... Done.
Bootstrap (r = 1.3)... Done.
Bootstrap (r = 1.4)... Done.

Cluster method: average
Distance      : correlation
Estimates on edges:
  au bp se.au se.bp v c pchi
1  1  1     0     0 0 0    0
2  1  1     0     0 0 0    0
3  1  1     0     0 0 0    0
```

A consensus matrix (n x n) can be produced from a bootstrap replicate matrix. To create the consensus matrix, a function to order the objects is required. The consensusmatrix function provides reorder in which a reorder method can be supplied. The reorder method must have two input arguments, namely a distance matrix and a number of clusters. Meanwhile, the output is only a membership. For example, hierarchical cluster algorithm with ward linkage is applied to order the objects in the consensus matrix.

To evaluate a clustering algorithm, a bootstrap evaluation function (clustboot) can be applied. Before applying clustboot, a clustering algorihtm that will be evaluated must be created first. The evaluated clustering algorithm must use the distance and / or matrix and the number of cluster. Then, the output must be a vector of membership. One can create two functions of clustering algorihtm that will be evaluated by the bootstrap.

For approximately unbiased and Bootstrap Probability Plots Discussion

For figure 13, one can see that the four attributes of Iris flower are examined and hierarchical clustering has been completed. Values on the edges are p-values percentage. The red values are AU p-values and green values are BP values. Clusters with AU greater than 95 percent are revealed which are highlighted by red (supposed to be rectangle shape) but there is a cut off, and thus are strongly supported by the Iris data for sepal length, petal length and petal width.

```
> qplot(iris$Sepal.Length,iris$Petal.Length)
> qplot(iris$Sepal.Length,iris$Petal.Width)
> summary(lm(Sepal.Length~Petal.Length+Petal.Width,iris))

Call:
lm(formula = Sepal.Length ~ Petal.Length + Petal.Width, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-1.18534 -0.29838 -0.02763  0.28925  1.02320

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.19058    0.09705  43.181  < 2e-16 ***
Petal.Length  0.54178    0.06928   7.820 9.41e-13 ***
Petal.Width  -0.31955    0.16045  -1.992   0.0483 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4031 on 147 degrees of freedom
Multiple R-squared:  0.7663,    Adjusted R-squared:  0.7631
F-statistic:   241 on 2 and 147 DF,  p-value: < 2.2e-16

> coef(lm(Sepal.Length~Petal.Length+Petal.Width,iris))
 (Intercept) Petal.Length  Petal.Width
   4.1905824    0.5417772   -0.3195506

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
 boot(data = iris, statistic = boot.lmCoef, R = 500, formula = "Sepal.Length~P
 Bootstrap Statistics :
     original     bias    std. error
 t1*   4.1906  0.0019586     0.10299
 t2*   0.5418 -0.0007954     0.07333
 t3*  -0.3196  0.0005106     0.16333

> boot.ci(lmCoef.booted,index=1)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
 Based on 500 bootstrap replicates
 CALL :
 boot.ci(boot.out = lmCoef.booted, index = 1)
 Intervals :
 Level      Normal              Basic
```

```
 95%   ( 3.987,  4.390 )   ( 3.988,  4.380 )
 Level     Percentile            BCa
 95%   ( 4.001,  4.393 )   ( 4.001,  4.395 )
 Calculations and Intervals on Original Scale

>boot.ci(lmCoef.booted,index=2)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 500 bootstrap replicates
 CALL :
 boot.ci(boot.out = lmCoef.booted, index = 2)
 Intervals :
 Level      Normal              Basic
 95%   ( 0.3989,  0.6863 )   ( 0.4006,  0.6863 )
 Level     Percentile            BCa
 95%   ( 0.3972,  0.6830 )   ( 0.3973,  0.6831 )
 Calculations and Intervals on Original Scale

>boot.ci(lmCoef.booted,index=3)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
 Based on 500 bootstrap replicates
CALL :
boot.ci(boot.out = lmCoef.booted, index = 3)
Intervals :
Level      Normal              Basic
95%   (-0.6402,  0.0001 )   (-0.6421, -0.0103 )
Level     Percentile            BCa
95%   (-0.6288,  0.0030 )   (-0.6229,  0.0089 )
Calculations and Intervals on Original Scale
```

2.2 Supervised Learning: K-means clustering

There are various means to cluster the iris data but K-Means algorithm is the one of the most robust algorithms Which tries to improve the inter group similarity while keeping the groups as far as possible from each other. However, the technique, K-Means runs on distance calculations, which again uses "Euclidean Distance" to ensure the objective or goal is done correctly. Euclidean distance, therefore, calculates the distance between two given points using the Pythagoras theorem like formula for Euclidean Distance. This formula captures the distance in 2-Dimensional space but the same is applicable in multi-dimensional space as well with increase in number of terms getting added. "K" in K-Means represents the number of clusters in which we want our data to divide into. The basic restriction for K-Means algorithm is that the Iris data in this case study must be continuous in nature.

K-Means is an iterative process of clustering; which keeps iterating until it reaches the best solution or clusters in the problem space.

In the algorithm, one Start with number of clusters chosen as in K = 3 Then, K-Means algorithm start the process with random centers in the iris data, and then tries to attach the nearest points to these centers

Secondly, the algorithm then moves the randomly allocated centers to the means of created groups
Thirdly, the data points are again reassigned to these newly created centers.
Finally, Steps 2 and 3 are repeated until no member changes their groups.

```
> cl = kmeans(X,centers=3, nstart=10)$cluster
> plot(PC$x[,1],PC$x[,2],col = cl,pch=point)
```

An elbow plot is developed shown in figure 15, as one can see there is a huge drop with K =3, but after that the variation starts to go down as quickly as possible. That is, there is a sudden drop in the value of WSS (within sum of squares) as the number of clusters increases from 1 to 3. Therefore, the bend at K = 3 gives stability in the value of WSS. In other words, one can see that there is not much decrease in WSS even if we increase the number of clusters beyond 7. This graph is also known as "Elbow Curve" where the bending point K = 7 is known as "Elbow Point". From the plot in figure 15 one can conclude that if one keep number of clusters, K = 3, one should be able to get good clusters with good homogeneity within themselves.

In figure 16, shows a silhouette method algorithm in action whereby a different coefficient is computed. This method allows the iris data be partitioned into K clusters and set the proximities between iris flower type. The minimum similarities between the iris data attributes to the nearest clusters are computed such that individual attributes silhouette coefficient are computed for each attributes types. Then the average silhouette coefficient to determine the clusters that are perfectly separated. In the figure, there is a huge reduction in variation with K=2, but after that the variation it goes down gradually as the number of cluster moves to K= 3 and then goes down again at K=4, then sharply drops again to K=5 and almost remain constant drop through K =6 to K = 9 and then goes up a little at K = 10.

2.3 Data Reduction Technique: Principal Components Analysis

The goal of the PCA is to visualize 4 (four) dimensional Iris data in 2 (two) dimension. This gives the best possible representation based on variability.

In figure 9, one can see that the first class representing the red dots (setosa) are far away from the other two with blue and green colors (virginica and versicolor). The blue and green dots seem to be separated but there are overlaps among them as well. This is because of the projection that provides separation that they appear to be. In three dimension, it is possible that one of the colors may appear closer while in four dimensions, there maybe more room for separation. Also, one can notice that each color tend to set up or move in a particular line, that is , each of the color follow different line but the red and green color lines are very close. Therefore, in this PCA, a linear boundary decision can be made because there is a separation between the blue and green colors. This is because the colors (red, blue and green) dots or points scattered around the real position or region with high variance.

```
> log.transform <- log(iris[, 1:4])
> transform.species <- iris[, 5]
> transform.pca <- prcomp(log.transform,
+                   center = TRUE,
+                   scale. = TRUE)
> print(transform.pca)
Standard deviations (1, .., p=4):
[1] 1.7124583 0.9523797 0.3647029 0.1656840
```

```
Rotation (n x k) = (4 x 4):
                    PC1         PC2        PC3         PC4
Sepal.Length  0.5038236 -0.45499872  0.7088547  0.19147575
Sepal.Width  -0.3023682 -0.88914419 -0.3311628 -0.09125405
Petal.Length  0.5767881 -0.03378802 -0.2192793 -0.78618732
Petal.Width   0.5674952 -0.03545628 -0.5829003  0.58044745

> summary(transform.pca)
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7125 0.9524 0.36470 0.16568
Proportion of Variance 0.7331 0.2268 0.03325 0.00686
Cumulative Proportion  0.7331 0.9599 0.99314 1.00000

> predict(transform.pca,
+         newdata=tail(log.transform, 5))
          PC1         PC2         PC3         PC4
146 1.5854470 -0.48131261 -0.01503973  0.16720005
147 1.6025558  0.85952833  0.22587694  0.13946811
148 1.3966185 -0.37860884 -0.08431475  0.04357352
149 1.0809930 -1.01155751 -0.70822894 -0.06811063
150 0.9712116 -0.06158655 -0.50086737 -0.12411524

> predict(transform.pca,
+         newdata=tail(log.transform, 10))
          PC1         PC2         PC3         PC4
141 1.6131365 -0.69091516 -0.14372854  0.07268257
142 1.6021154 -0.77881493  0.06393658  0.21204093
143 1.1641446  0.64648210 -0.37482319 -0.04815585
144 1.6253264 -0.93746400 -0.13700625 -0.02213506
145 1.5218423 -1.08203286 -0.31925846  0.03332150
146 1.5854470 -0.48131261 -0.01503973  0.16720005
147 1.6025558  0.85952833  0.22587694  0.13946811
148 1.3966185 -0.37860884 -0.08431475  0.04357352
149 1.0809930 -1.01155751 -0.70822894 -0.06811063
150 0.9712116 -0.06158655 -0.50086737 -0.12411524

> g <- ggbiplot(transform.pca, obs.scale = 1, var.scale = 1,
+               groups = transform.species, ellipse = TRUE,
+               circle = TRUE)
> g <- g + scale_color_discrete(name = '')
> g <- g + theme(legend.direction = 'horizontal',
+                legend.position = 'top')
> print(g)

 biplot(transform.pca)
```

```
theta <- seq(0,2*pi,length.out = 100)
circle <- data.frame(x = cos(theta), y = sin(theta))
p <- ggplot(circle,aes(x,y)) + geom_path()
loadings <- data.frame(transform.pca$rotation,
                       .names = row.names(transform.pca$rotation))

 p + geom_text(data=loadings,
               mapping=aes(x = PC1, y = PC2, label = .names, colour = .names))

  coord_fixed(ratio=1) +

  labs(x = "PC1", y = "PC2")
```

This method describe the importance of the four principal components for iris flower data. The first row represents the standard deviation associated with each PCs. The second row gives the proportion of variance in the iris data as explained by each principal component while the third row represents the cumulative proportion of the variance. One can see that the first two components accounts for more than 95 percent variability in the data. One can see that the first principal component gives 73.31 percent of the variance in the iris data while the second principal component accounts for 22.68 percent of variance in the iris data. This primarily implies that the features are correlated and the variance driving this correlation is captured by principal component 1 and principal component 2. Perhaps, PC1 expresses the size of the blossom which is seen in all four features. More so, the eigenvalue of PC1 is greater than 1.0 while other PCs are less than the value of 1.

Similarly, one assume to observe new data and thus analyze for their principal components value. One can use their last 5 rows of the same Iris data and computed for their principal components as analyzed above. Same also goes for the last 10 rows of the iris data. The iris data is well trained in this manner for validation purpose

Part 3 Research Oriented on Consensus clustering

One can say that Consensus clustering alleviates common issues that arise in most clustering methods, such as random initialization, choosing K, intuitive visualization and assessing stability of clusters. This method gathers a consensus of cluster assignments based on sub-sampling the dataset and running a chosen clustering algorithm multiple times. There are various metrics one can use to validate the chosen K. One can also assess clusters using cluster consensus and see which items best represent a given cluster using item consensus.

A consensus matrix (n x n) can be produced from a bootstrap replicate matrix. To create the consensus matrix, a function to order the objects is required. The consensusmatrix function provides reorder in which a reorder method can be supplied. The reorder method must have two input arguments, namely a distance

matrix and a number of clusters. Meanwhile, the output is only a membership. For example, hierarchical cluster algorithm with ward linkage is applied to order the objects in the consensus matrix.

consensus matrix are ordered by the consensus clustering which is represented as a dendrogram atop the heatmap. for analysis, the cluster membership are marked by colored rectangles between the dendrogram and heatmap. This allow one to compare clusters member in the consensus.

3.1 Data

The iris input data is in the form of matrix where columns are the class, rows are the features and the cells are numerical values.The columns and rows are maintained in the output based on the assigned names. The data was transformed using the Bioconductor method, the Pearson correlation distance and Manhattan methods were utilized separate in this analysis.

Consensus Hierarchical Clustering was utilized to improve stability of the Iris flower data clustering by performing on the original iris dataset, then clustering and finally,summarizing the results. Refer to figure 7 for ConsensusClusterPlus result and code below:

```
> library(ConsensusClusterPlus)
Warning message:
package 'ConsensusClusterPlus' was built under R version 3.5.1
> results = ConsensusClusterPlus(t(X),maxK=6,reps=50,pItem=0.8,pFeature=1,titl

end fraction
clustered
clustered
clustered
clustered
clustered
> plot(PC$x[,1],PC$x[,2],col=results[[3]]$consensusClass, pch=point)
> legend("topright",legend=levels(species), pch=unique(point))
```

3.2 Method

Consensus Clustering is generally a method that provides quantitative evidence for determining the number and memberships of possible clusters within a dataset. The pairwise consensus values, the proportion that two items occupied the same cluster out of the numbers of times they occurred in the same subsample, are calculated and stored in a symmetrical consensus matrix for each K. The consensus matrix is summarized in several graphical displays that enable a user to decide upon a reasonable cluster number and membership.

```
> dt = as.dist(1-cor(X,method="pearson"))
> ConsensusClusterPlus(dt,maxK=3,reps=100,pItem=0.8,pFeature=1,title="Iris",di
end fraction
clustered
clustered
[[1]]
     [,1]      [,2]      [,3]      [,4]
[1,] "#A6CEE3" "#1F78B4" "#A6CEE3" "#A6CEE3"
[2,] "#1F78B4" "#1F78B4" "#B2DF8A" "#33A02C"

[[2]]
```

```
[[2]]$`consensusMatrix`
          [,1] [,2]      [,3] [,4]
[1,] 1.0000000    0 0.4489796  0.5
[2,] 0.0000000    1 0.0000000  0.0
[3,] 0.4489796    0 1.0000000  1.0
[4,] 0.5000000    0 1.0000000  1.0

[[2]]$consensusTree

Call:
hclust(d = as.dist(1 - fm), method = finalLinkage)

Cluster method   : average
Number of objects: 4


[[2]]$consensusClass
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
           1            2            1            1

[[2]]$ml
          [,1] [,2]      [,3] [,4]
[1,] 1.0000000    0 0.4489796  0.5
[2,] 0.0000000    1 0.0000000  0.0
[3,] 0.4489796    0 1.0000000  1.0
[4,] 0.5000000    0 1.0000000  1.0

[[2]]$clrs
[[2]]$clrs[[1]]
[1] "#A6CEE3" "#1F78B4" "#A6CEE3" "#A6CEE3"

[[2]]$clrs[[2]]
[1] 2

[[2]]$clrs[[3]]
[1] "#A6CEE3" "#1F78B4"



[[3]]
[[3]]$`consensusMatrix`
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
```

```
[[3]]$consensusTree

Call:
hclust(d = as.dist(1 - fm), method = finalLinkage)

Cluster method   : average
Number of objects: 4


[[3]]$consensusClass
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
           1            1            2            3

[[3]]$ml
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1

[[3]]$clrs
[[3]]$clrs[[1]]
[1] "#1F78B4" "#1F78B4" "#B2DF8A" "#33A02C"

[[3]]$clrs[[2]]
[1] 4

[[3]]$clrs[[3]]
[1] "#1F78B4" "#B2DF8A" "#33A02C"
```

```
> myDistFunc = function(x){ dist(x,method="manhattan")}
> ConsensusClusterPlus(X,maxK=2,reps=100,pItem=0.8,pFeature=1,title="iris",dis
end fraction
clustered
```

3.3 Results: Consensus Cumulative Distribution Function and Delta Area

Figures 20 and 21 show the consensusCDF, delta area and tracking plot for when K = 3 and K = 2 at maximum set points. Figure 20 revealed the value at which cumulative distribution function reaches maximum of the consensus matrix for K = 3 is at maximum which gives the consensus and cluster confidence at that maximum value of K = 3. For figure 21, the cumulative distribution function reaches maximum, therefore, consensus and cluster confidence is at maximum at K = 2.

For delta area in figure 20, there is a relative change in area under the CDF curve comparing K = 3 and K = 2. However, for K = 2 in figure 21, there is nothing to compare since there is no K = 1. So the total area under the CDF curve is constant, nothing to compare with.

The tracking plot simply gives the cluster assignment in the column for each K in the row as shown in figure 20. The plot gives gives membership across different K to be able to track the history of clusters

relative to the prior clusters. Any items that change will show or reveal instability membership by the color.

3.4 Discussion, Conclusions, and Personal Thought

Conclusively, one can see that the major problem with the hierarchical clustering is that it does give low stability whereas for the consensus clustering there is no guarantee that one will get what is expected among the three (3) group consensus hierarchical clustering. However, the K-means clustering provides much more stability but somewhat sensitive to outliers in the iris data. It is well known that the K-medoids (partitioning around medoids) a version of the k-means is more robust to outliers. The best method so far is the K-means.

The Iris data set is only 4-dimensional, making it possible to explore using pairs or parallel coordinates plot. It is clear from these that two main clusters are visible, while the separation of the third cluster is difficult.

In this analysis, having played with the codes, I discovered that the complete method fails to do the proper separation of the two main clusters when k = 2 but very good, if K = 3 clusters. This is quite different from all the other methods available in hclust, which do succeed in separating the 2 main clusters from the beginning. I also noticed that all clustering algorithms utilized share a relatively high proportion of common nodes between 80 percent to 90 percent. Finally, when it comes to trying to separate the flowers into 3 species, the average clustering method performed excellently well.

Also, I discovered that data transformation is indeed very important but not limited to log transformation. I believe other transformation such as Box-transformation or cox-transformation may also be applicable on other cases too or combination of box-cox transformation.

References

1. Monti, S., Tamayo, P., Mesirov, J. and Golub T. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Dat. Machine Learning (2003) 52: 91. doi:10.1023/A:1023949509487
2. Matthew D. Wilkerson and D. Neil Hayes. ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking. Bioinformatics Vol. 26 no. 12 (2010), pages 1572?1573 doi:10.1093/bioinformatics/btq170
3. Yasin Senbabaoglu, George Michailidis & Jun Z. Li. Critical limitations of consensus clustering in class discovery. Scientific Reports 4, Article number: 6207 (2014) doi:10.1038/srep06207

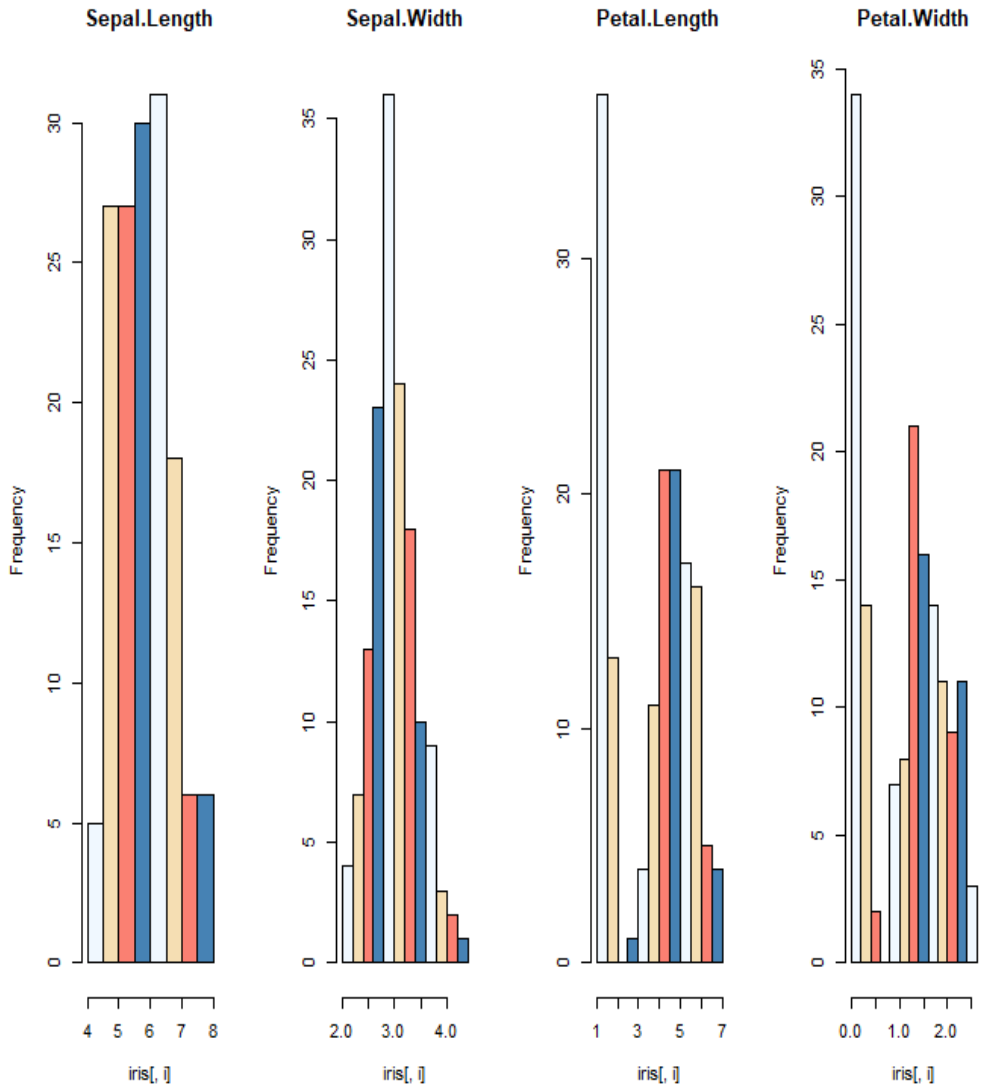

Figure 1. Histogram for Iris Flower

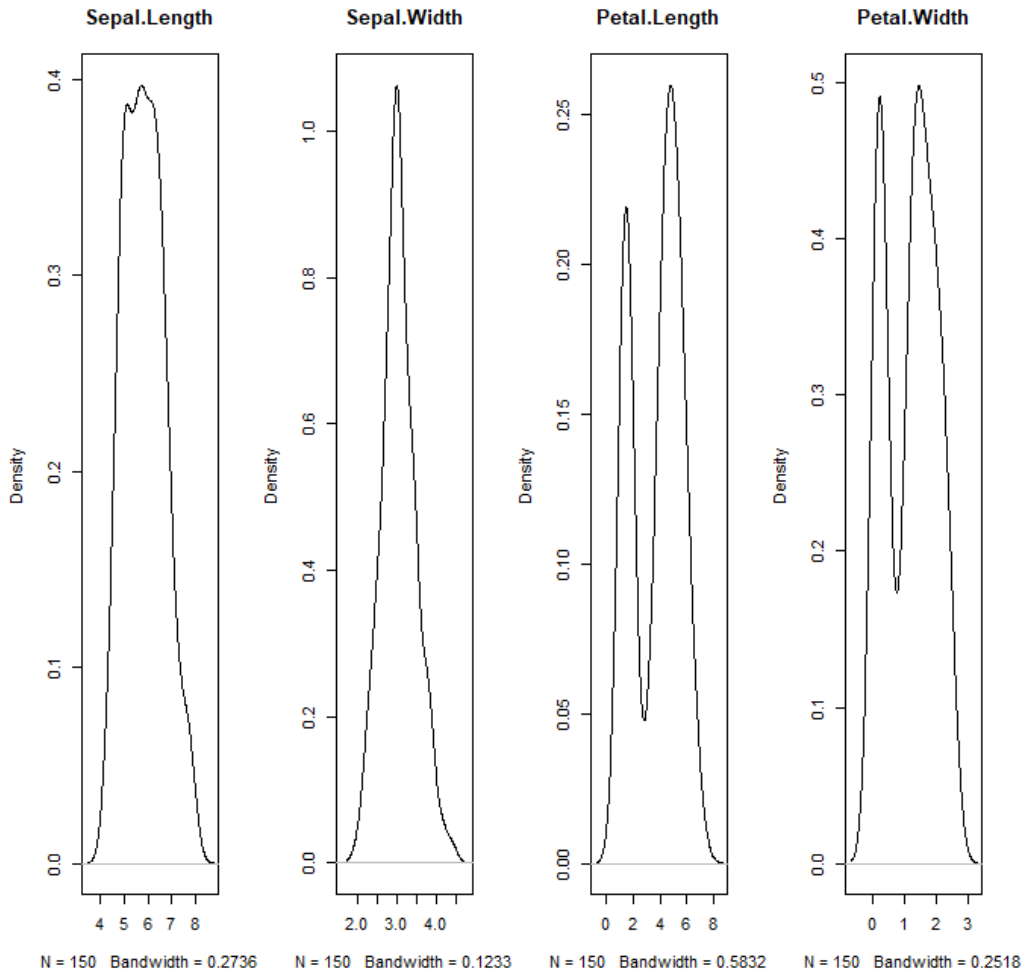

Figure 2. Density Plot for Iris Flower Data

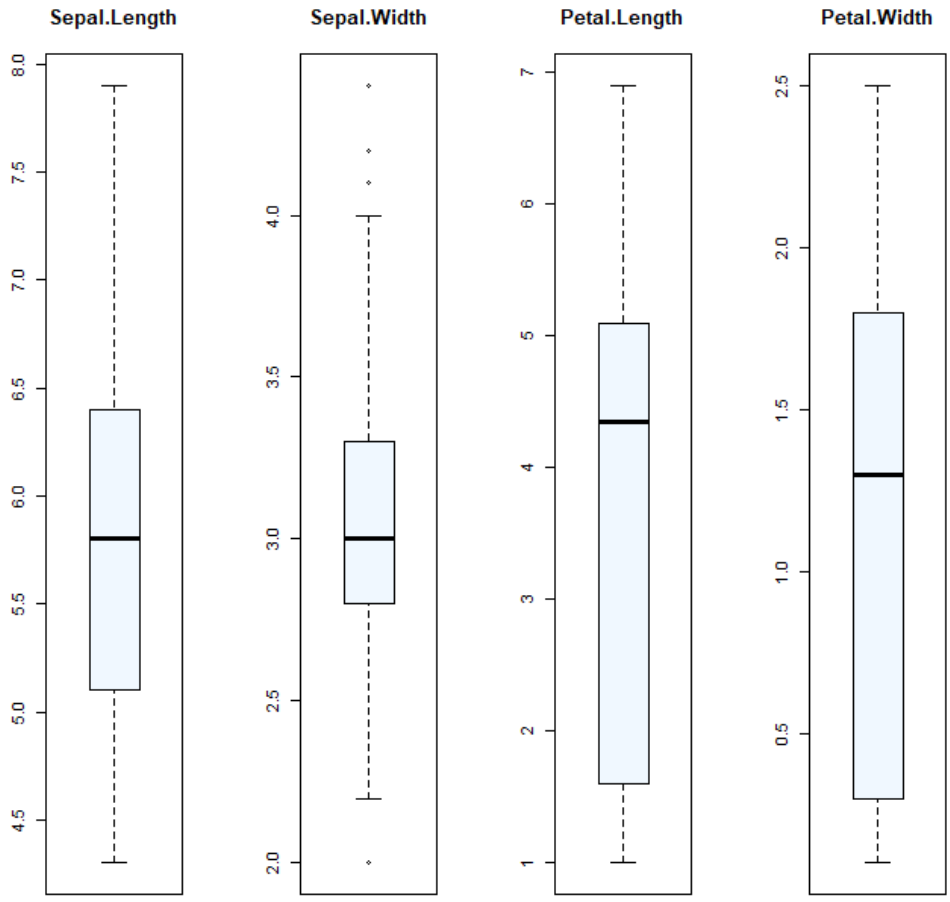

Figure 3. Boxplot for Iris Flower

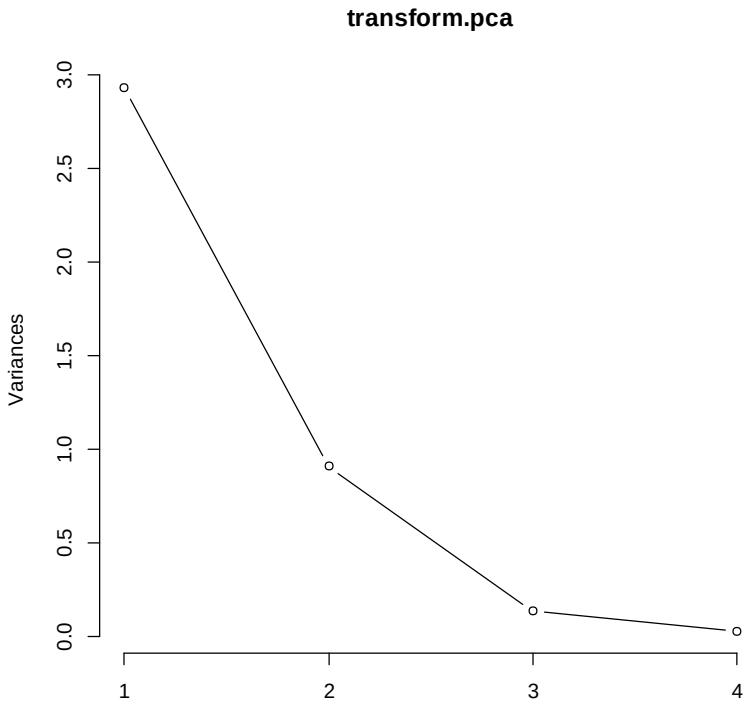

Figure 4. Variance and PC Plot for Iris Flower Data

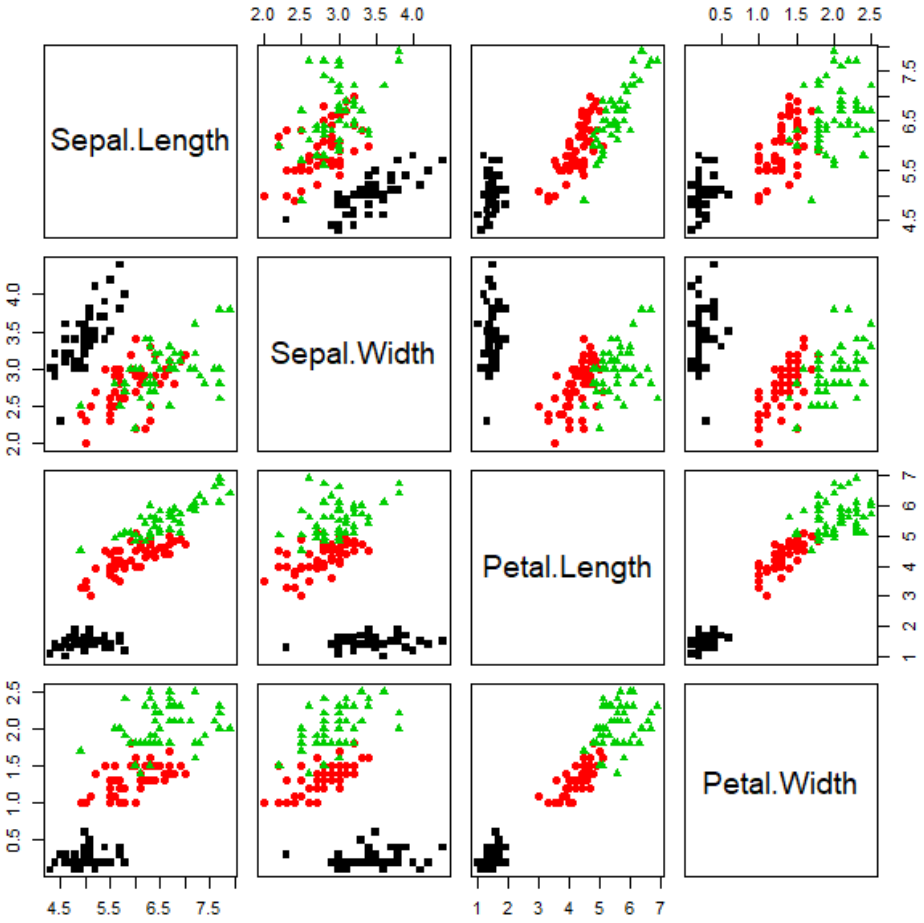

Figure 5. Scatter Plot Matrix by Class for Iris Flower

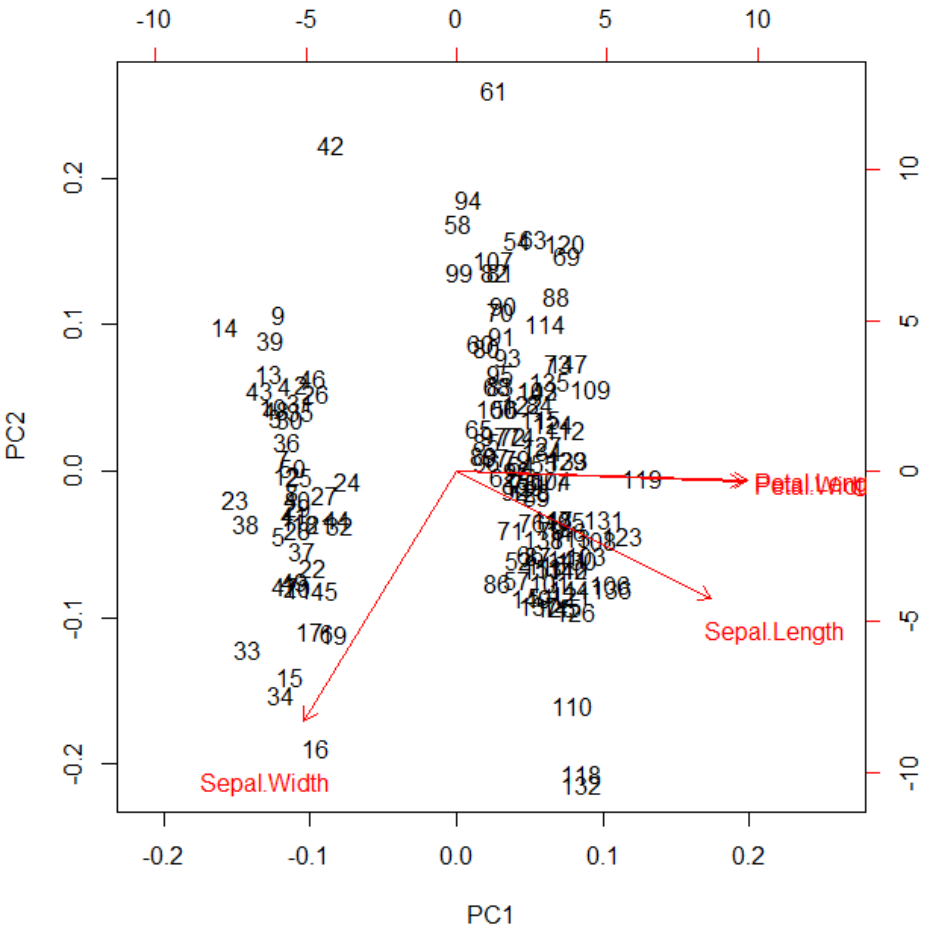

Figure 6. PCA Biplot for Iris Flower Data

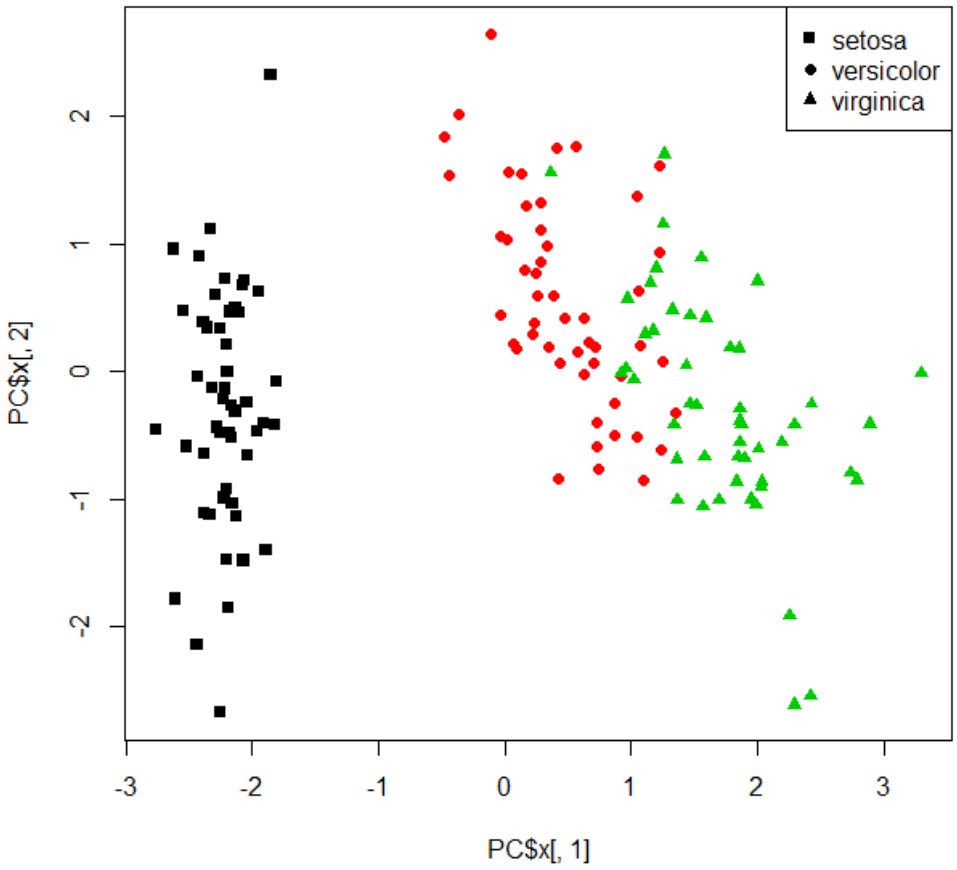

Figure 7. ConsensusClusterPlus for Iris Flower without Variability

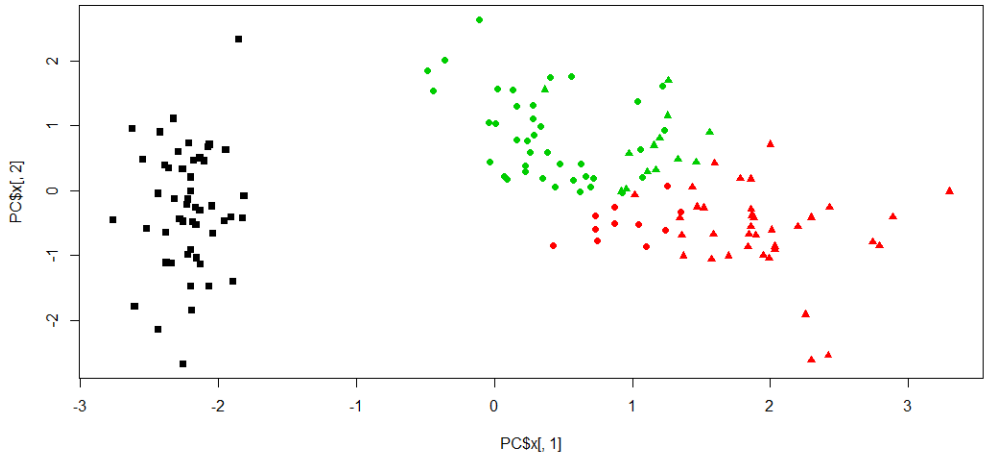

Figure 8. Kmeans clustering for Iris Flower Data without Variability

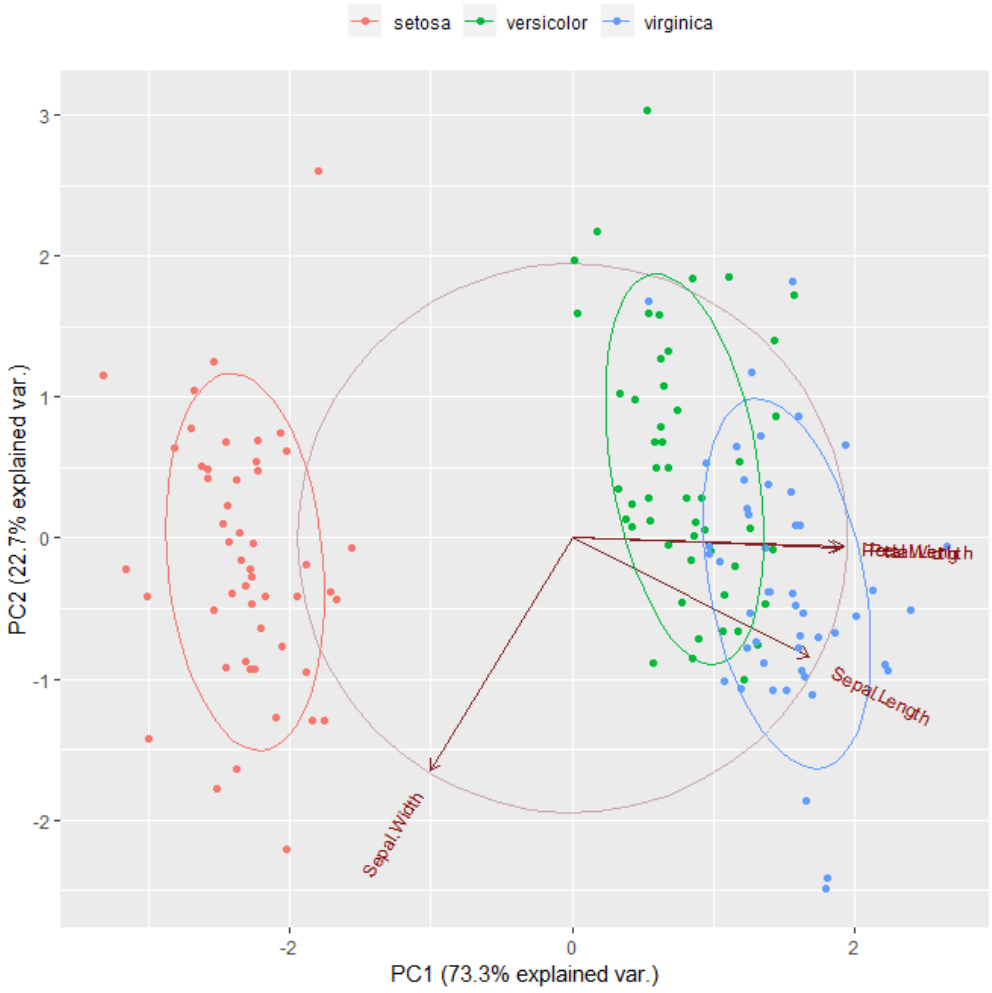

Figure 9. PCA for Iris Flower Data with Variability

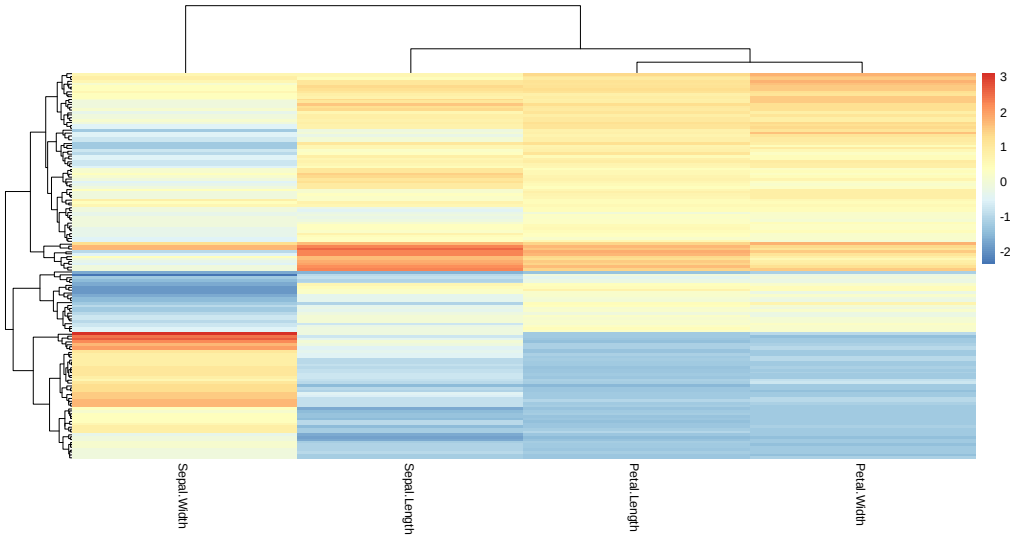

Figure 10. Heatmap for Iris Flower with Variability

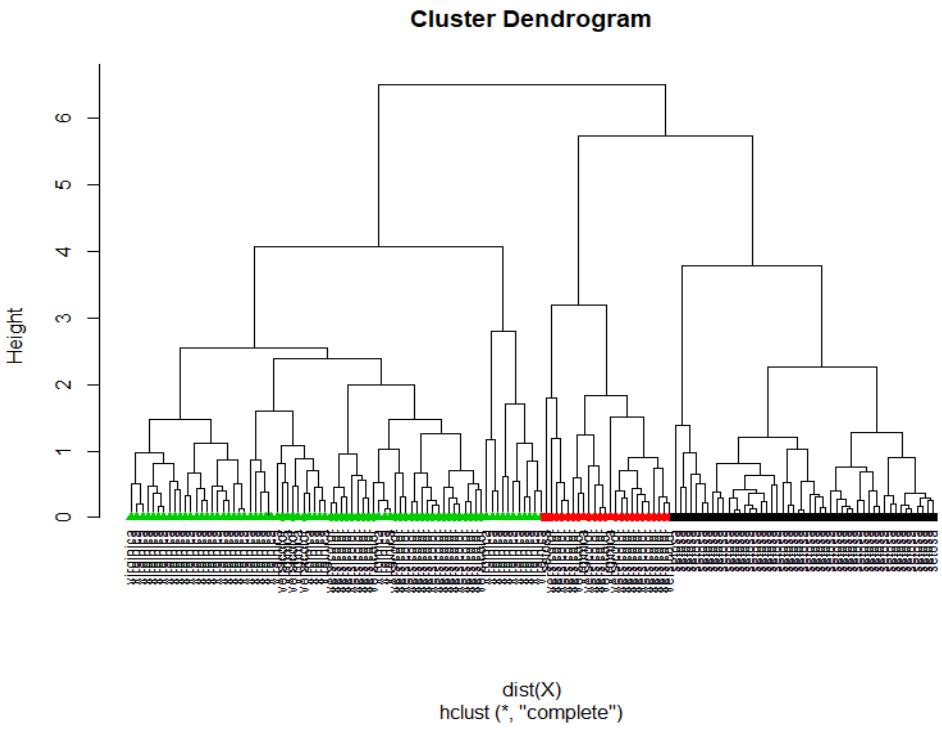

Figure 11. Hierarchical Clustering

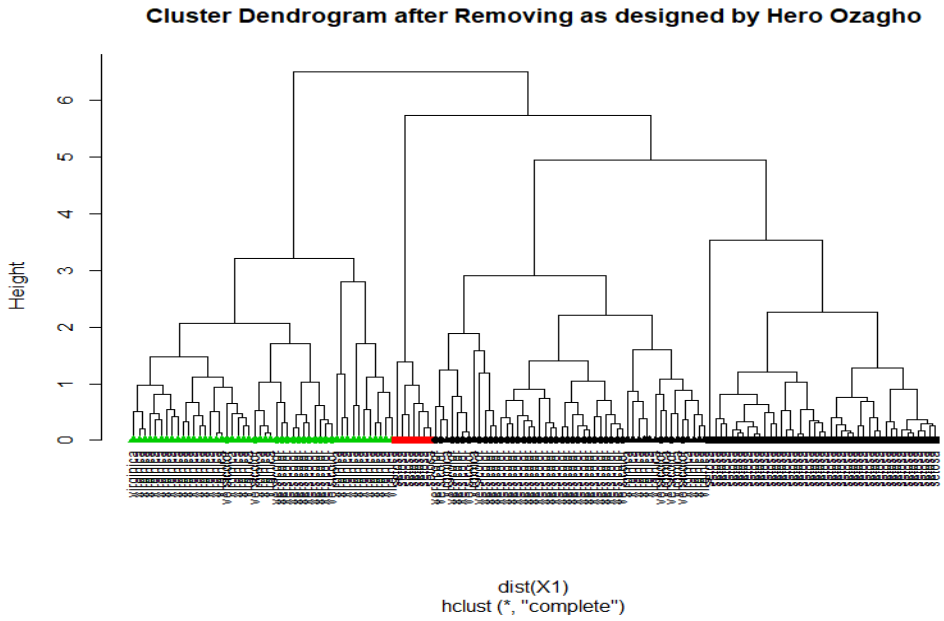

Figure 12. Hierarchical Clustering after Removing on Iris Flower

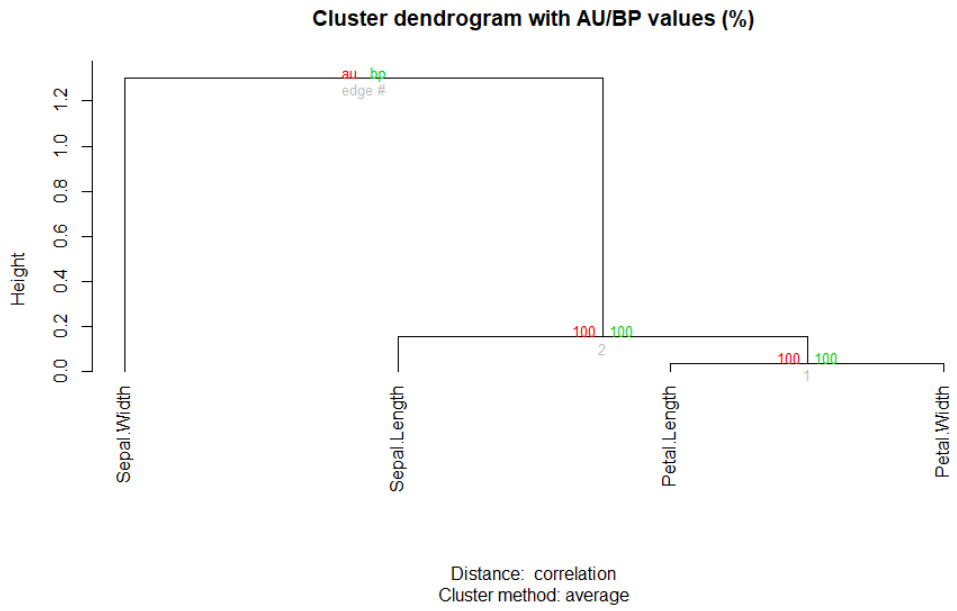

Figure 13. Hierarchical Clustering Boostrap for Iris Flower Data

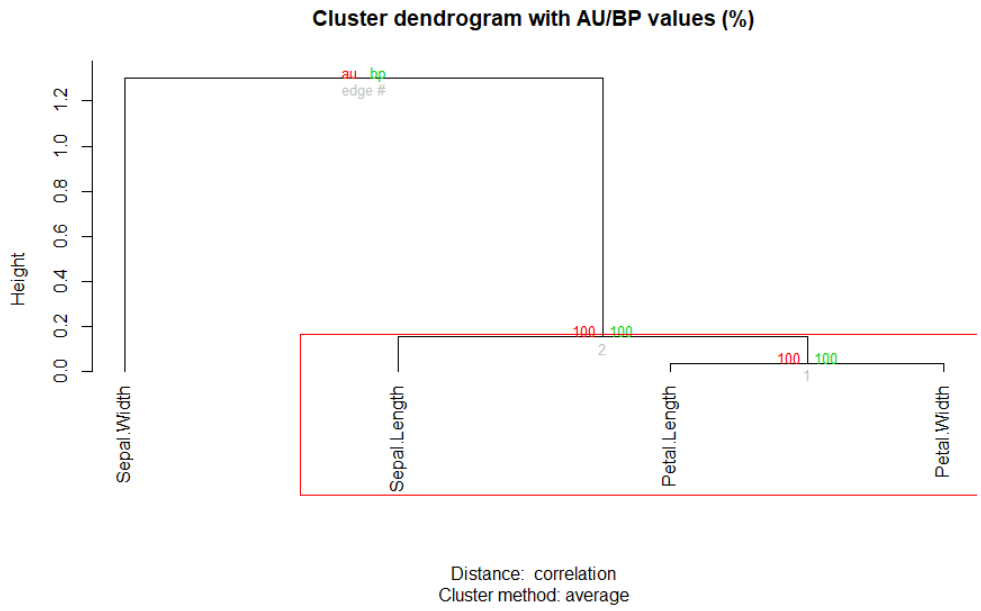

Figure 14. Hierarchical Clustering Boostrap for Iris Flower Data

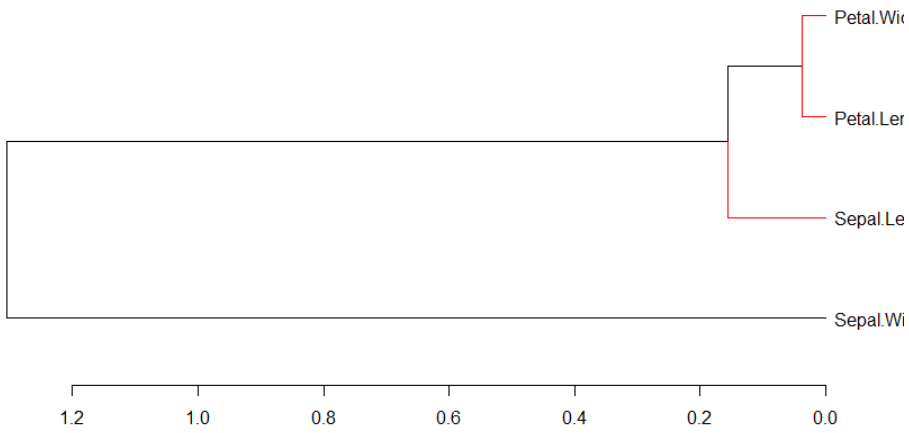

Figure 15. Hierarchical Clustering Boostrap for Iris Flower Data

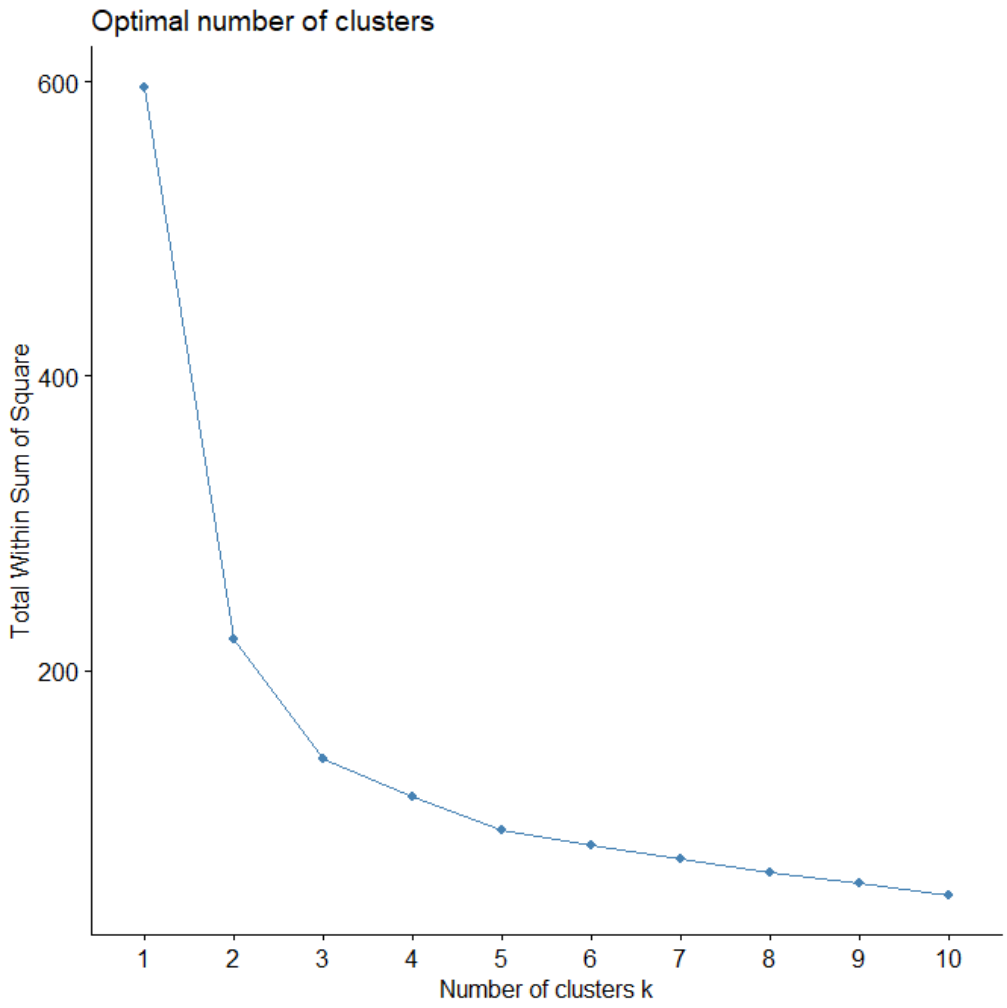

Figure 16. WSS for Iris flower Data

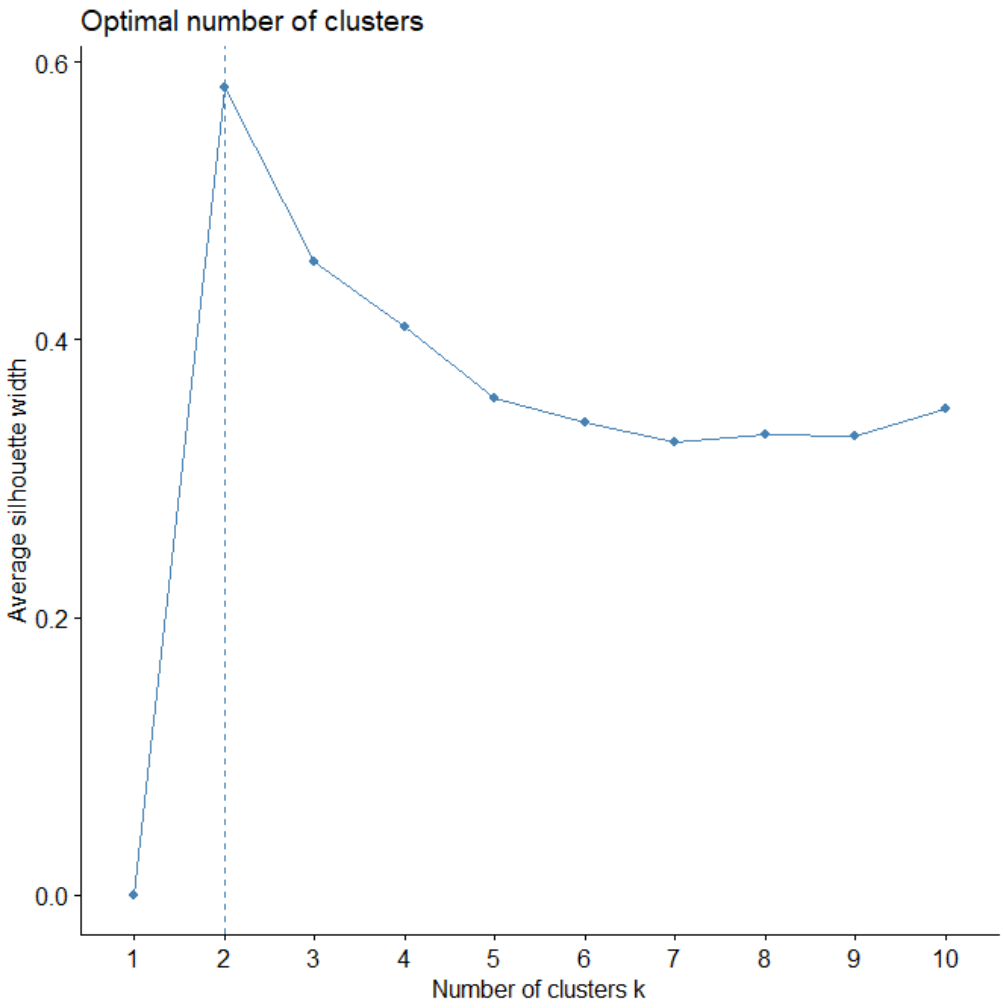

Figure 17. Silhouette Width for Iris Flower Data

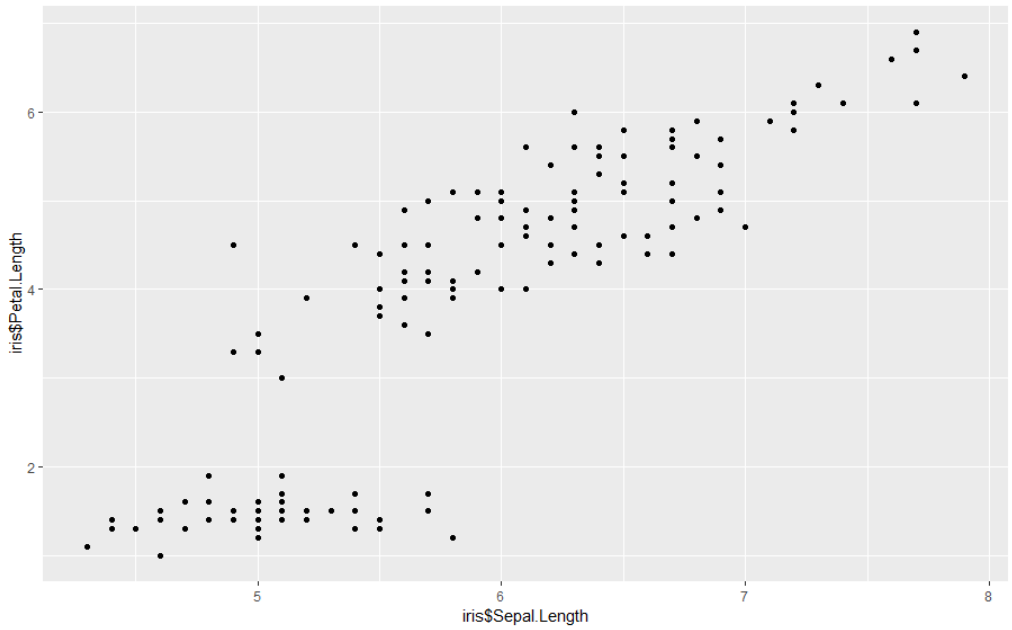

Figure 18. Bootstrap for Iris Flower qplot

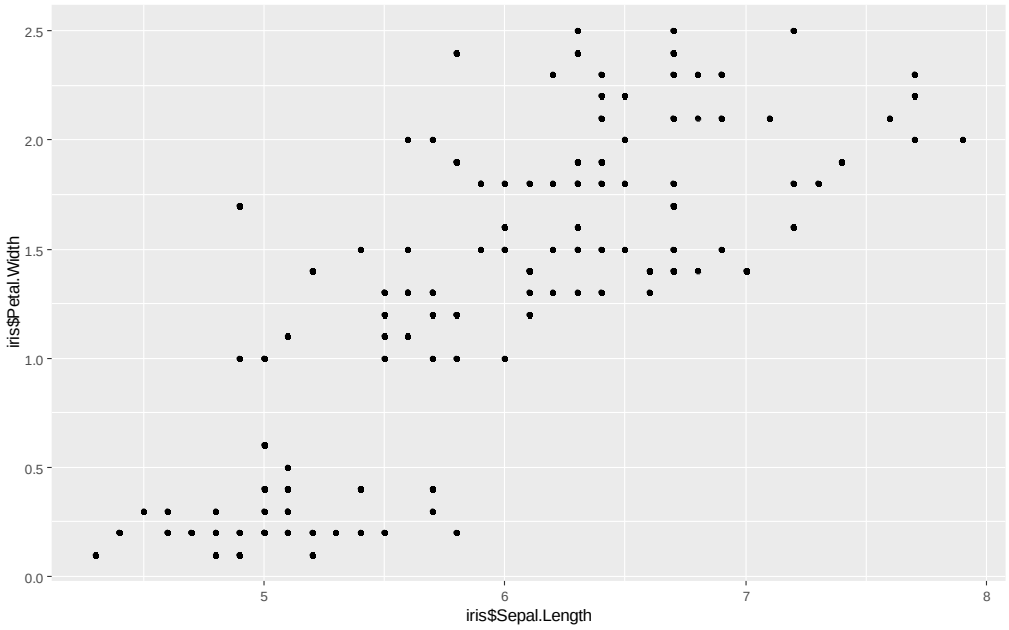

Figure 19. Bootstrap2 for Iris Flower qplot

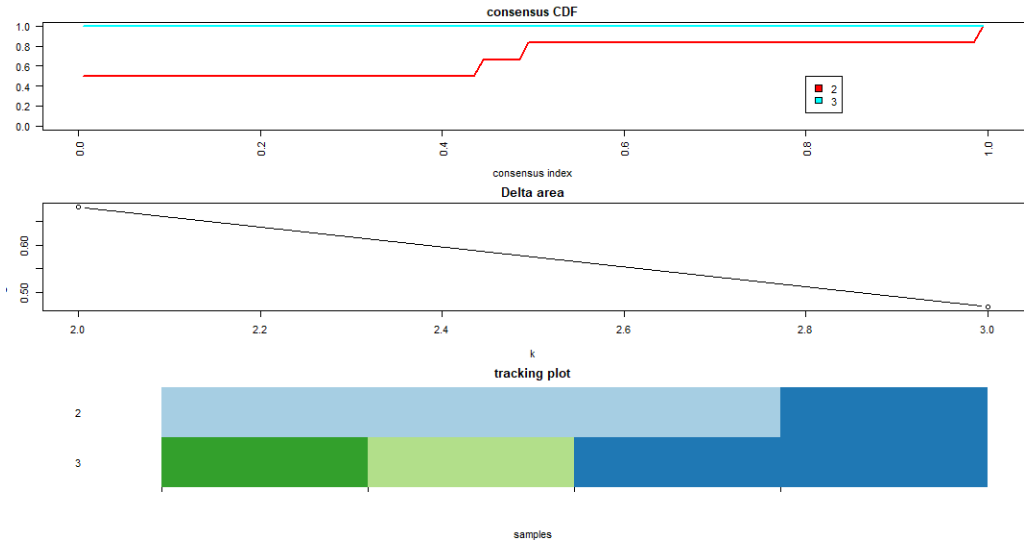

Figure 20. ConsensusCDF and Delta Area for Iris Flower Plot Pearson Method

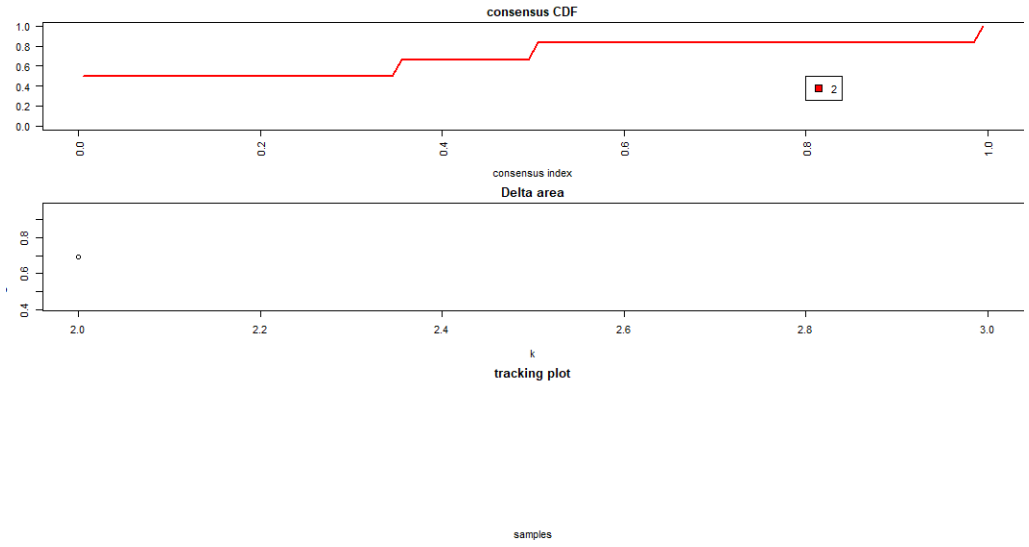

Figure 21. ConsensusCDF and Delta Area for Iris Flower Plot Manhattan Method

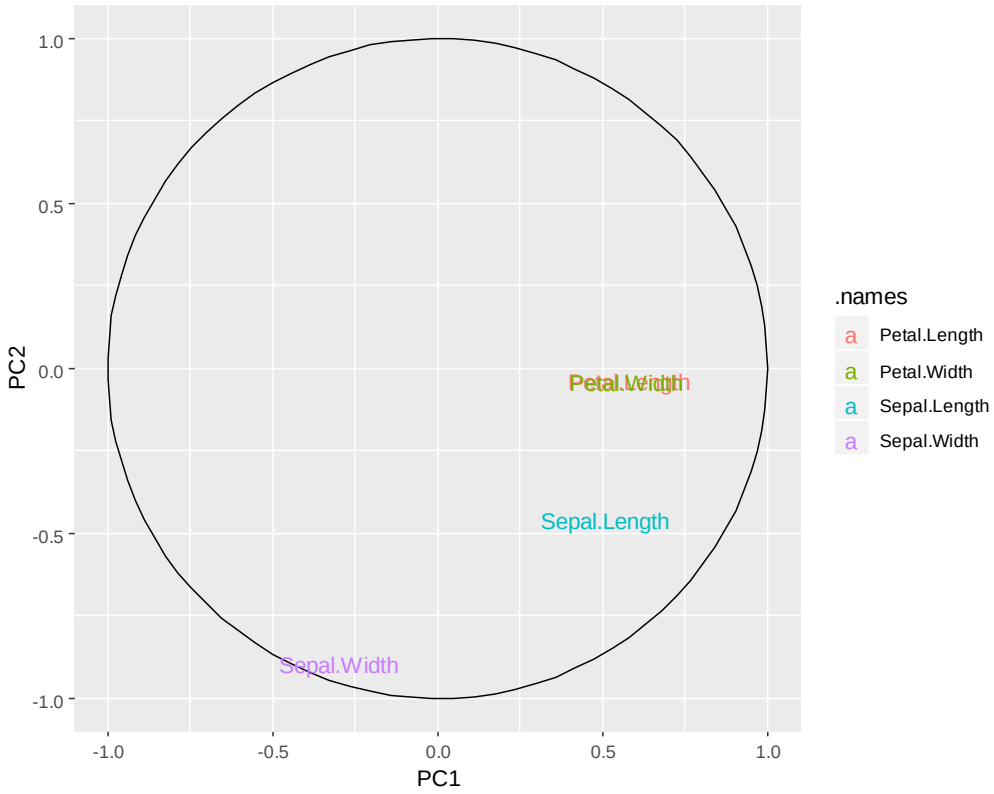

Figure 22. PCA Variable Coefficient in a Circle for Iris Flower Data