STA 546 : STATISTICAL DATA MINING 2

HOMEWORK 2

BY

ABINESH SENTHIL KUMAR

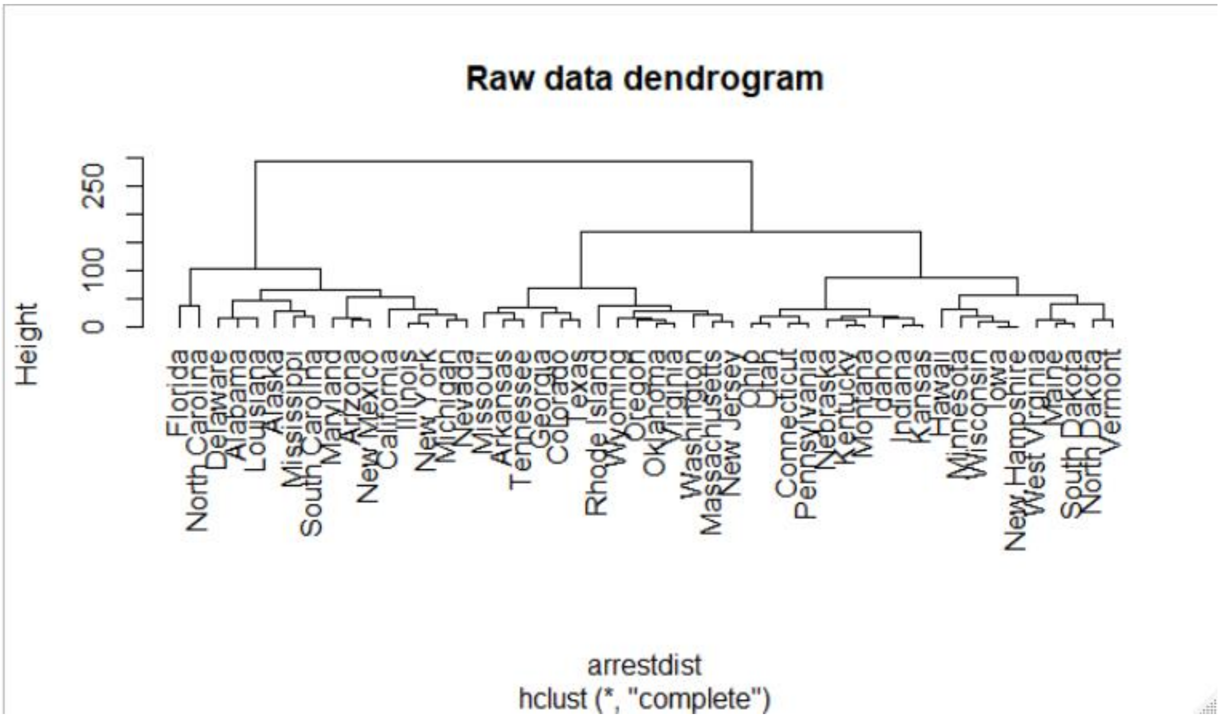
#50320934

Question 1

1. Consider the USArrests data. We will now perform hierarchical clustering on the states.

(a) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

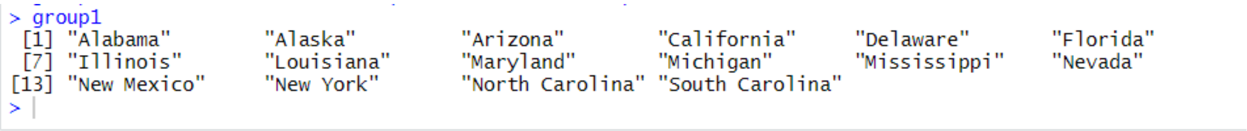
Hierarchical clustering using Euclidean distance and complete linkage,



(b) Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?

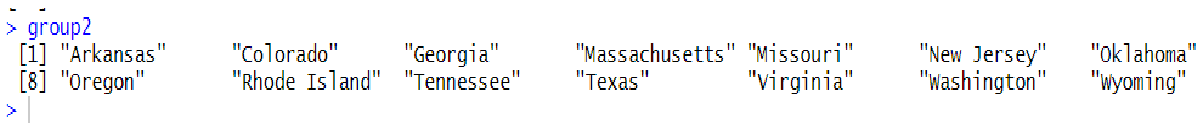
To have three different clusters, we can cut the dendogram at height of 120.

We can see that the cluster 1 contains,



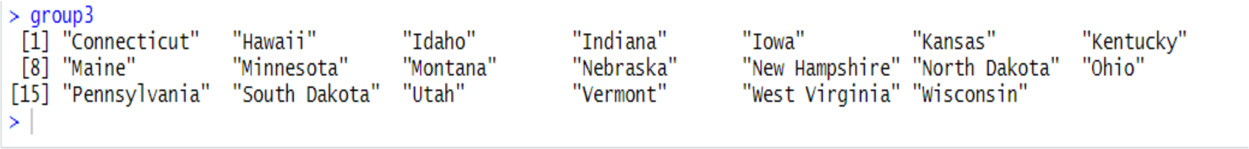
```
> group1
 [1] "Alabama"        "Alaska"         "Arizona"        "California"     "Delaware"       "Florida"
 [7] "Illinois"       "Louisiana"      "Maryland"       "Michigan"       "Mississippi"    "Nevada"
[13] "New Mexico"     "New York"       "North Carolina" "South Carolina"
> 
```

Cluster 2 contains,



```
> group2
 [1] "Arkansas"      "Colorado"     "Georgia"      "Massachusetts" "Missouri"      "New Jersey"    "Oklahoma"
 [8] "Oregon"        "Rhode Island" "Tennessee"    "Texas"         "Virginia"      "Washington"    "Wyoming"
> 
```

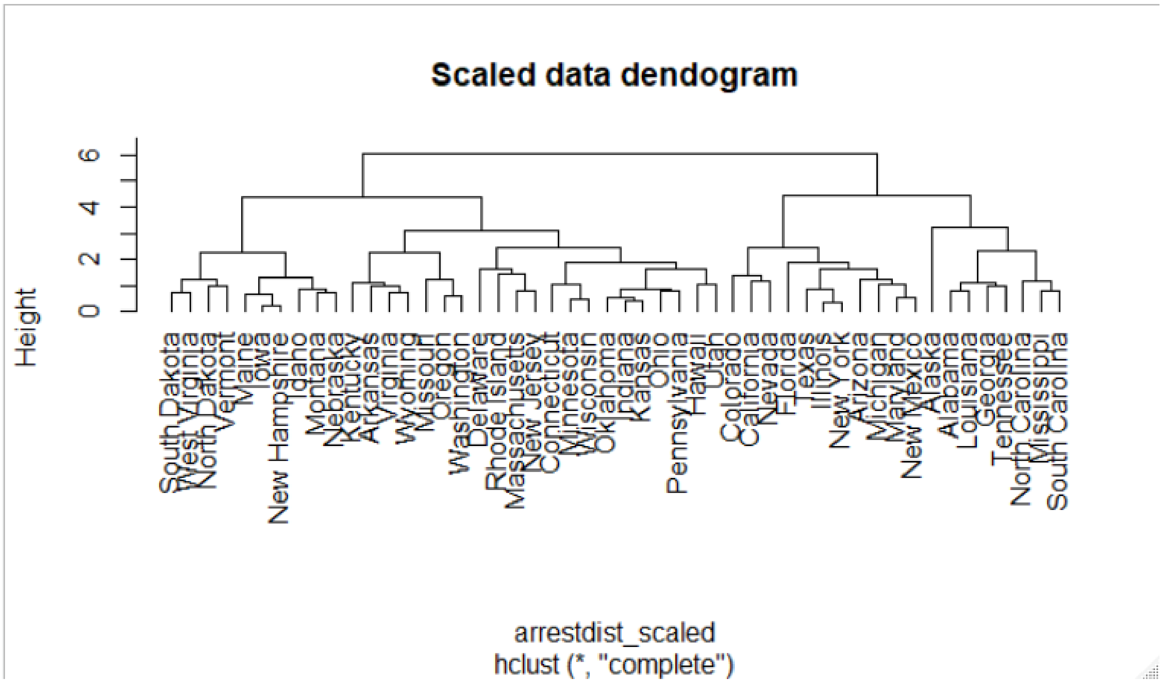
Cluster 3 contains,



```
> group3
 [1] "Connecticut"  "Hawaii"       "Idaho"        "Indiana"       "Iowa"          "Kansas"        "Kentucky"
 [8] "Maine"        "Minnesota"    "Montana"      "Nebraska"      "New Hampshire" "North Dakota"  "Ohio"
[15] "Pennsylvania" "South Dakota" "Utah"         "Vermont"       "West Virginia" "Wisconsin"
> 
```

(c) Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one.

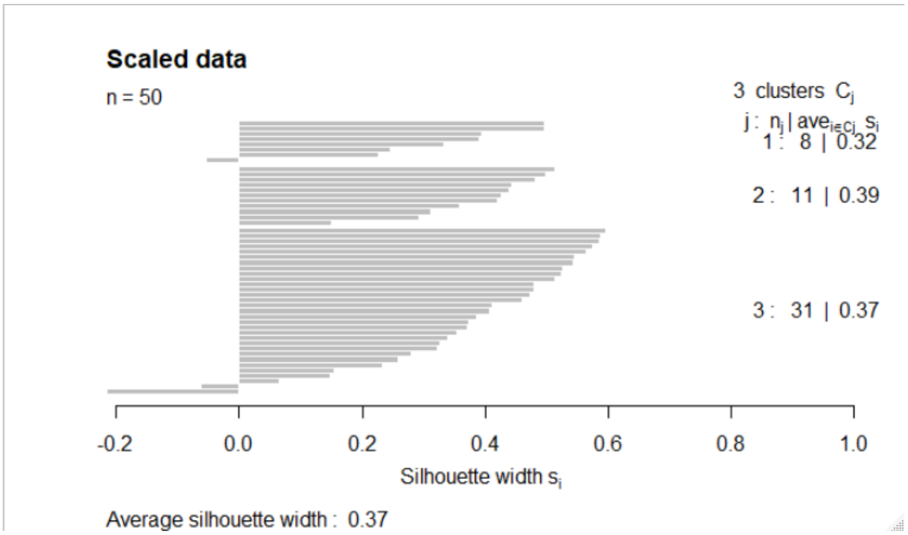
Applying hierarchial clustering for scaled data (to have standard deviation one) with eucledian distamce and complete linkage.



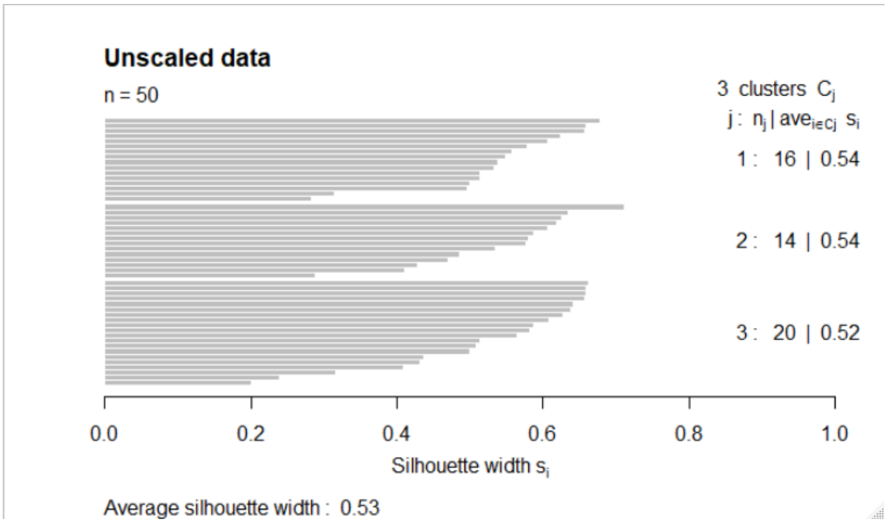
(d) What effect does scaling the variables have on the hierarchical clustering obtained? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed? Provide a justification for your answer.

For this particular dataset, when comparing the silhoutte plot before and after scaling we can see that hierarcjial clustering performs better in the raw data.

The average silhouette width from scaled data is 0.37



The average silhouette width from unscaled data is 0.53



We can conclude that for this particular dataset scaling does not help. Also, from the dendrogram of scaled data we can see the distinct cluster formed is two rather than our original clustering of three.

However, usually we must scale the data before calculating Euclidian distance and clustering since all of the variables might have different range for example one variable might have range of 0-100 while other might have a range of 0-1, in this case the variable with largest range would dominate the clustering algorithm unless we scale the variables first.

Question 2

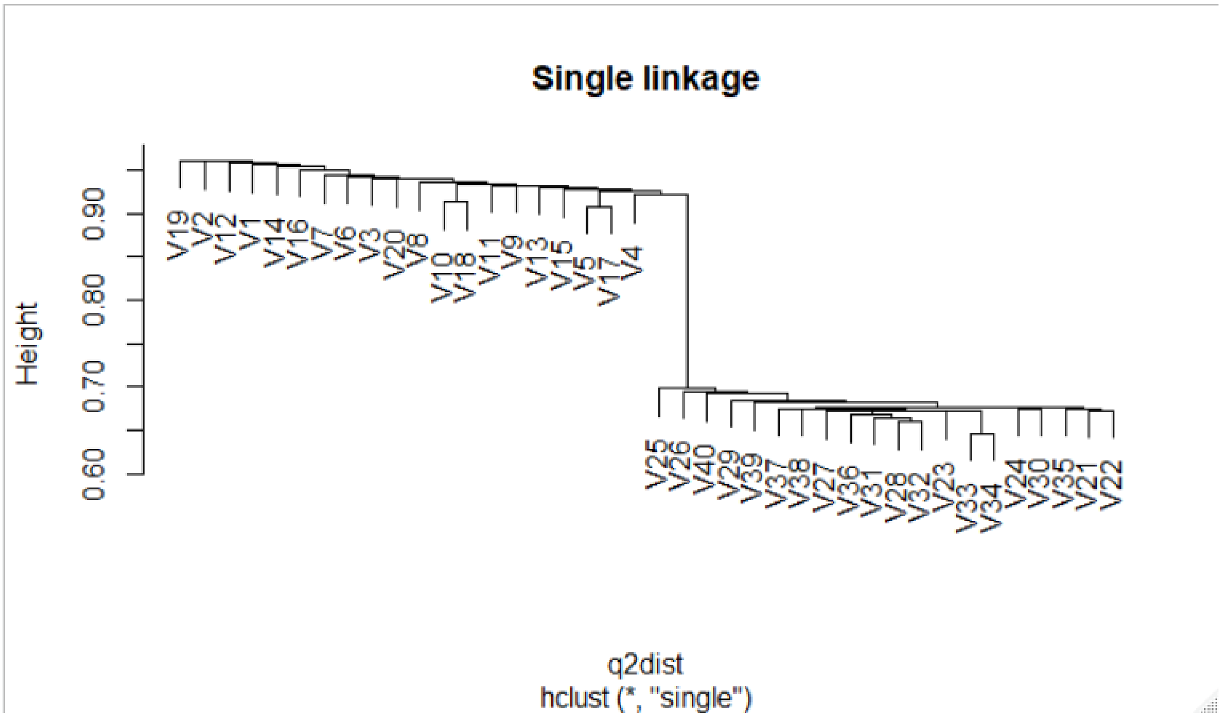
2. On the book website, www.StatLearning.com, there is a gene expression data set (Ch10Ex11.csv) that consists of 40 tissue samples with measurements on 1,000 genes. The first 20 samples are from healthy patients, while the second 20 are from a diseased group. 418 10. Unsupervised Learning

(a) Load in the data using read.csv(). You will need to select header=F.

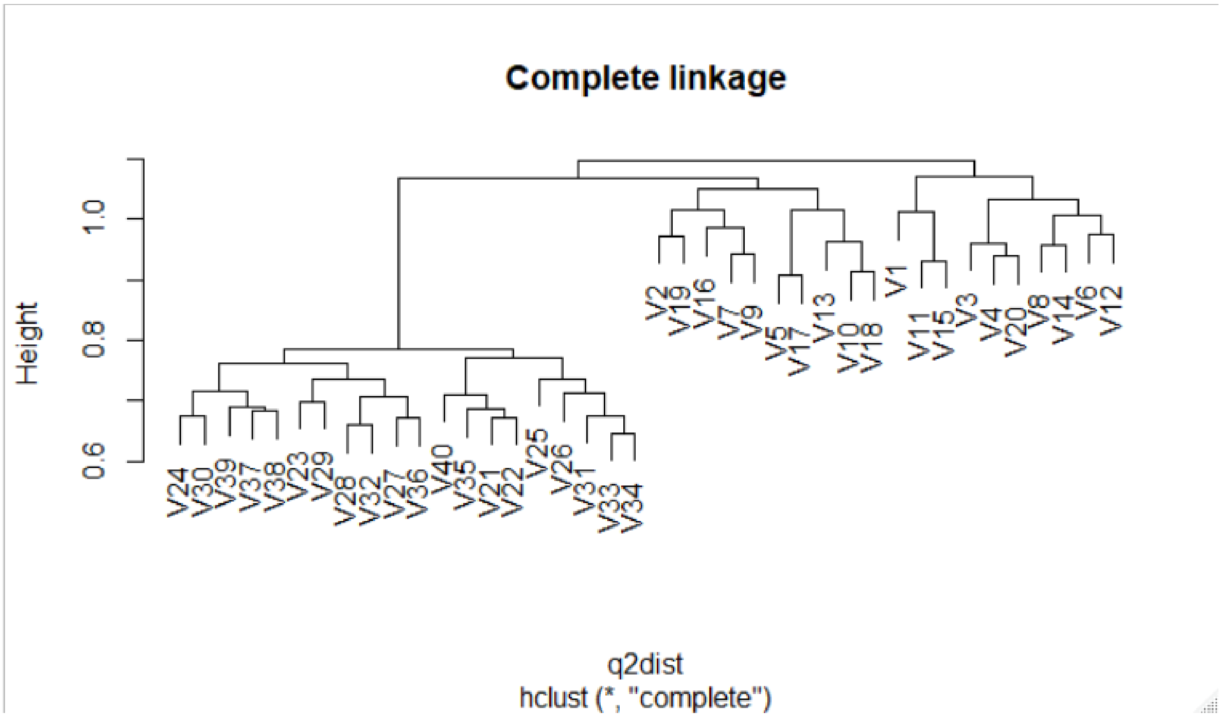
(b) Apply hierarchical clustering to the samples using correlation based distance and plot the dendrogram. Do the genes separate the samples into the two groups? Do your results depend on the type of linkage used?

Correlation based distance is calculated using **as.dist(1-cor(dataset),** and hierarchical clustering is done using all three methods,

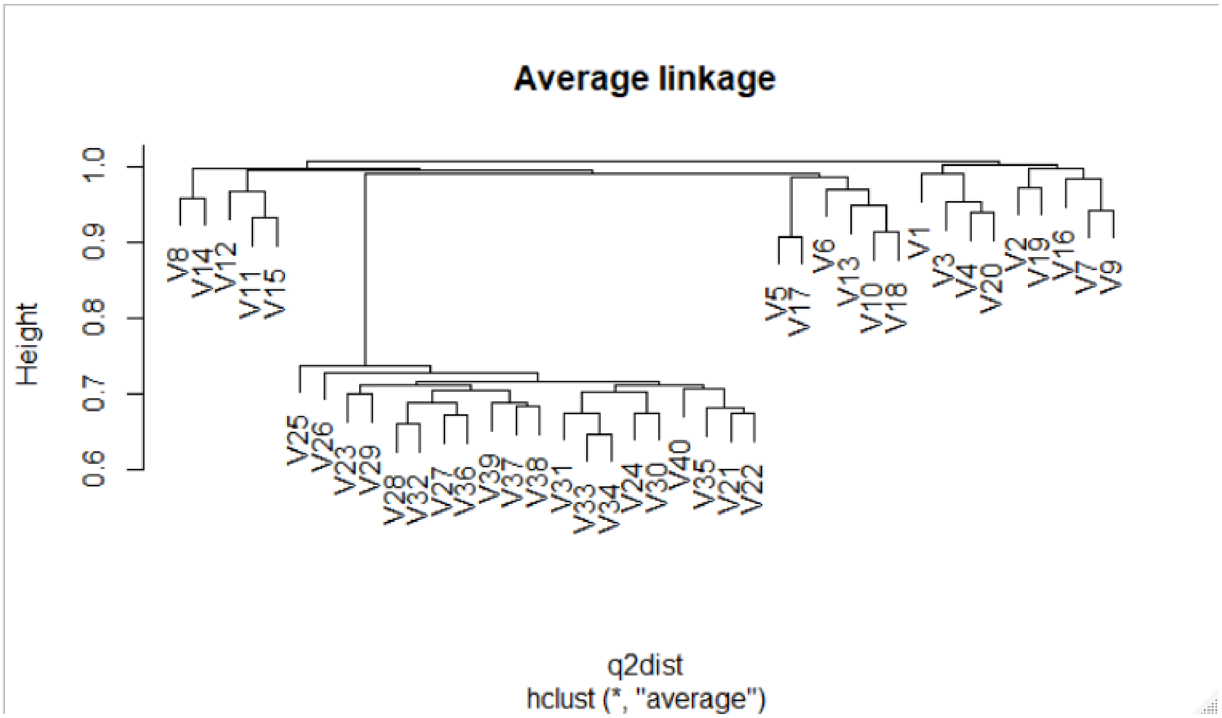
Single linkage,



Complete linkage,



Average linkage,

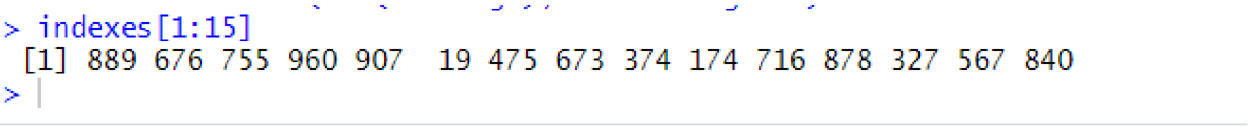


Inference,

- Yes, the results are depended on the type of linkage used.
- Single linkage method produces poor result which gives unstable dendrogram.
- Average linkage produces dendrogram with three clusters.
- Complete linkage has the dendrogram with two clusters.
- From this we can say that the genes are able to separate the sample into two and three groups respectively.

(c) Your collaborator wants to know which genes differ the most across the two groups. Suggest a way to answer this question and apply it here.

- To find which genes differ from each other we can use principal component analysis and look at the loading vector from the PCA.
- Applying PCA to the dataset and after finding the total weight given to each gene for different principle components,



```
> indexes[1:15]
 [1] 889 676 755 960 907  19 475 673 374 174 716 878 327 567 840
> 
```

- We can say that these are the top 15 genes that differ the most across two groups.

Question 3

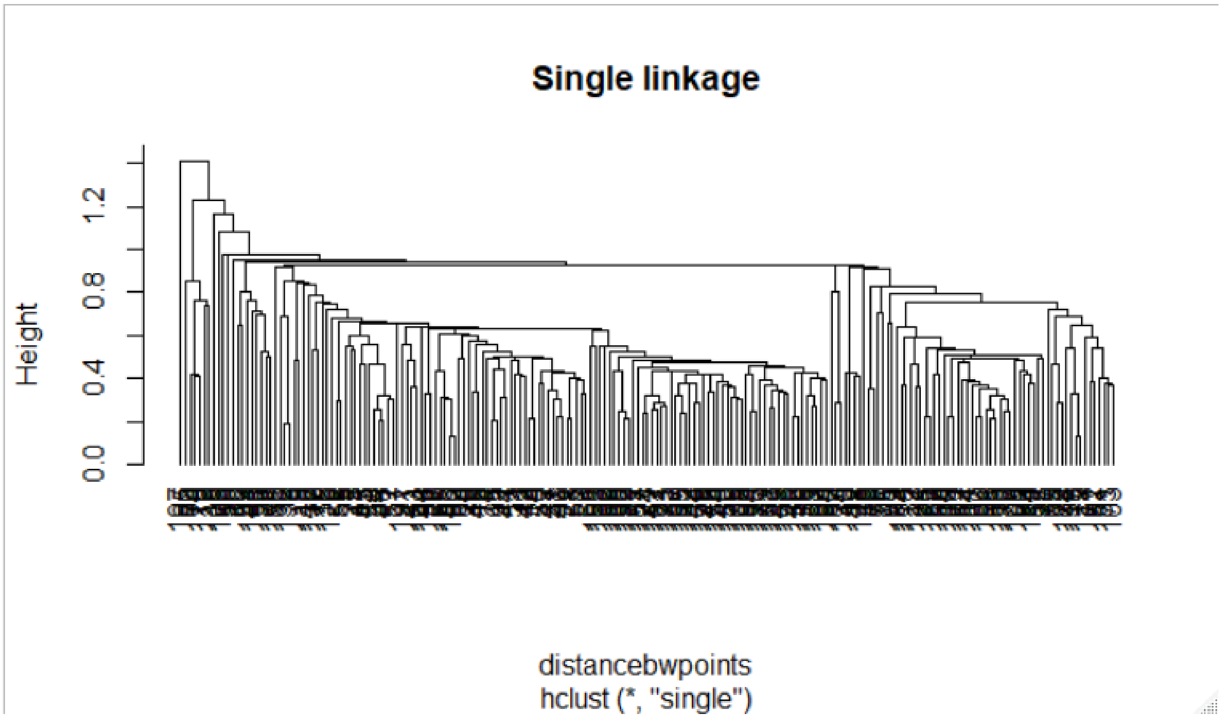
3) (10 points) Access the data "seeds data" (on UB learns). This data contains the geometrical properties of kernels belonging to three different varieties of wheat (seed group). The original data can be found: https://archive.ics.uci.edu/ml/datasets/seeds, although I have modified the data slightly.

a) Cluster the data based on single-linkage, average linkage, and complete-linkage agglomerative hierarchical clustering. Do not use the "seed group" column to perform the clustering, but use it to help evaluate your results.

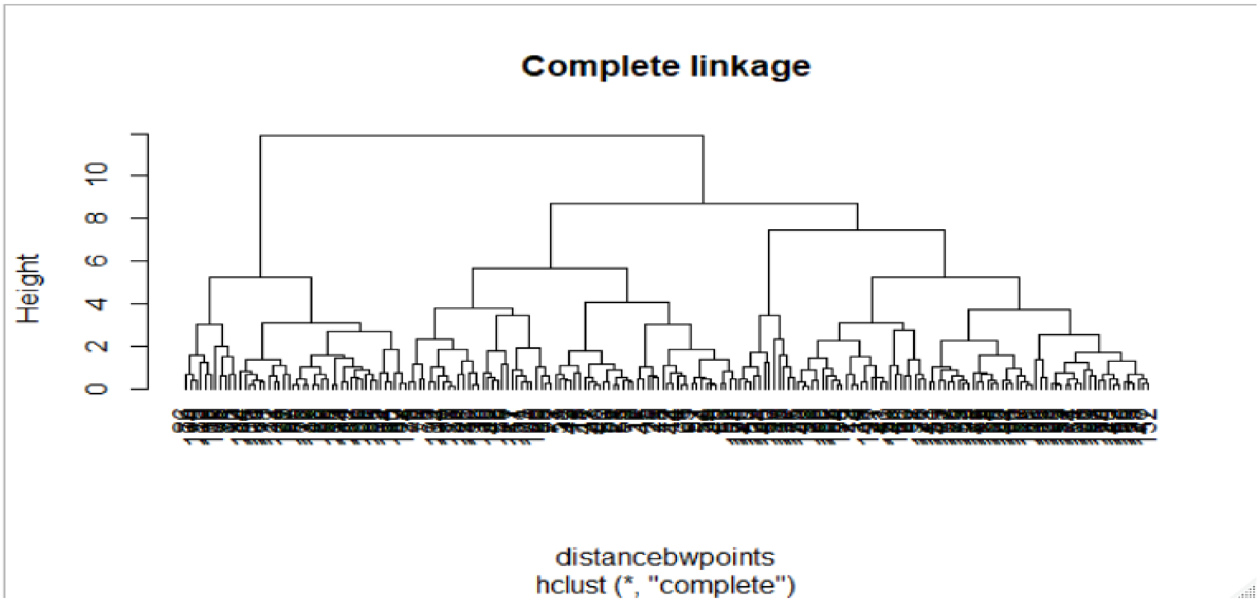
Decide on the groupings, and justify it, for all three methods. The justification should be based on a measure (you select which) that we learned in class.

Which method "performed" the best and which method performed the worst? Was the result in line with your expectations?

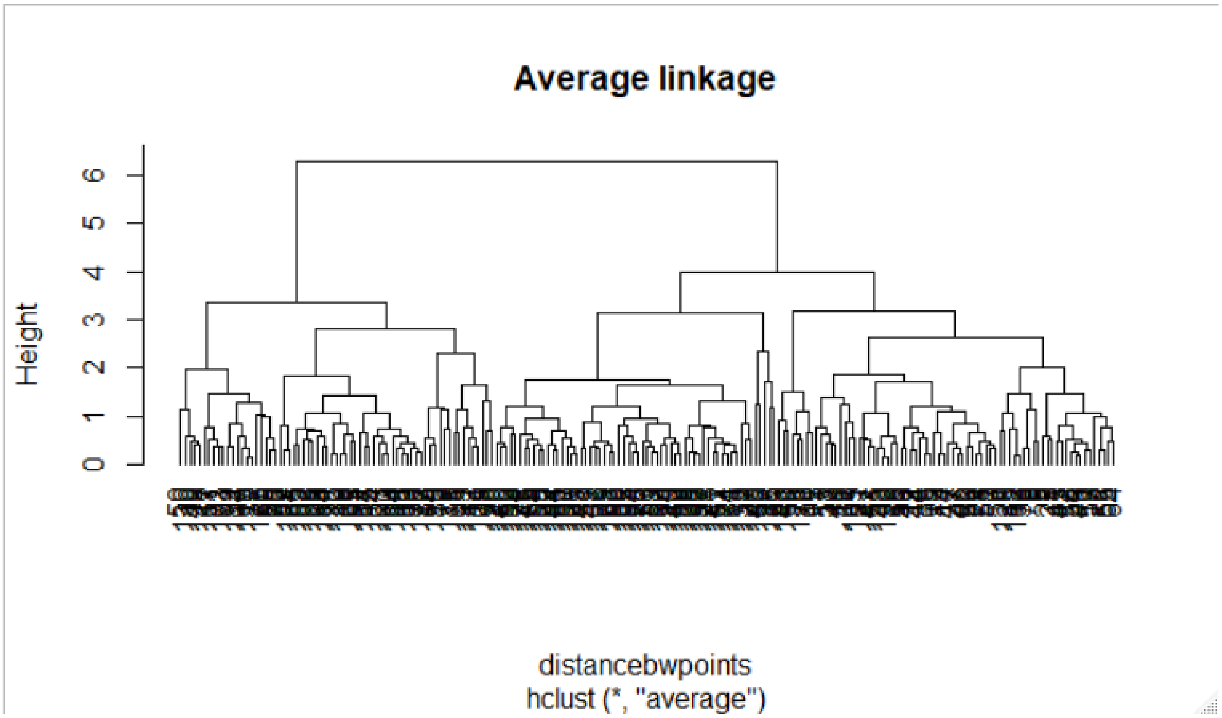
Hierarchical clustering using single linkage



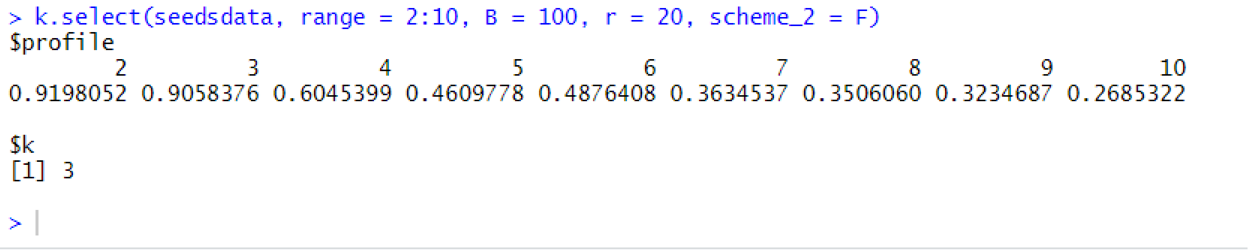
Hierarchal clustering using complete linkage



Hierarchical clustering using Average linkage



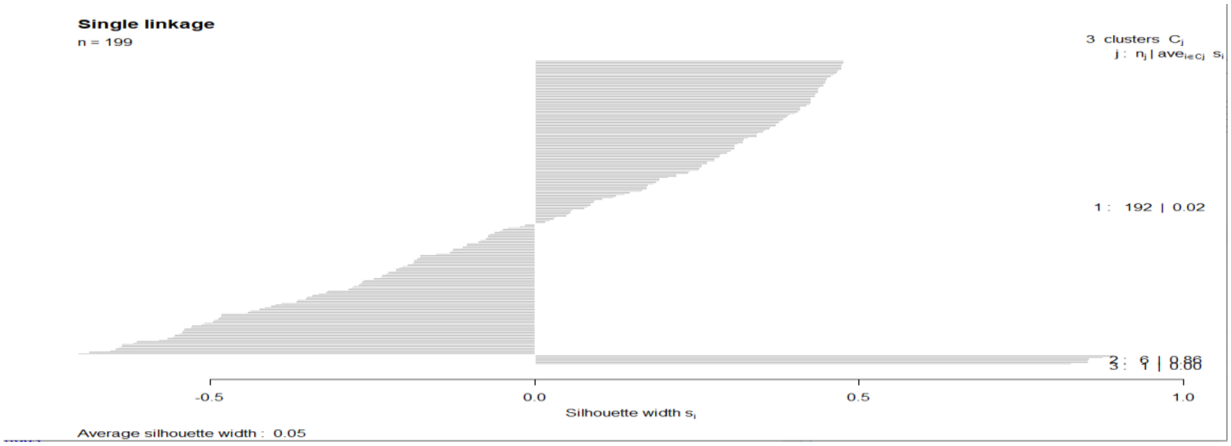
We find the optimal number of groups as three from bootstrapping method (k.select())



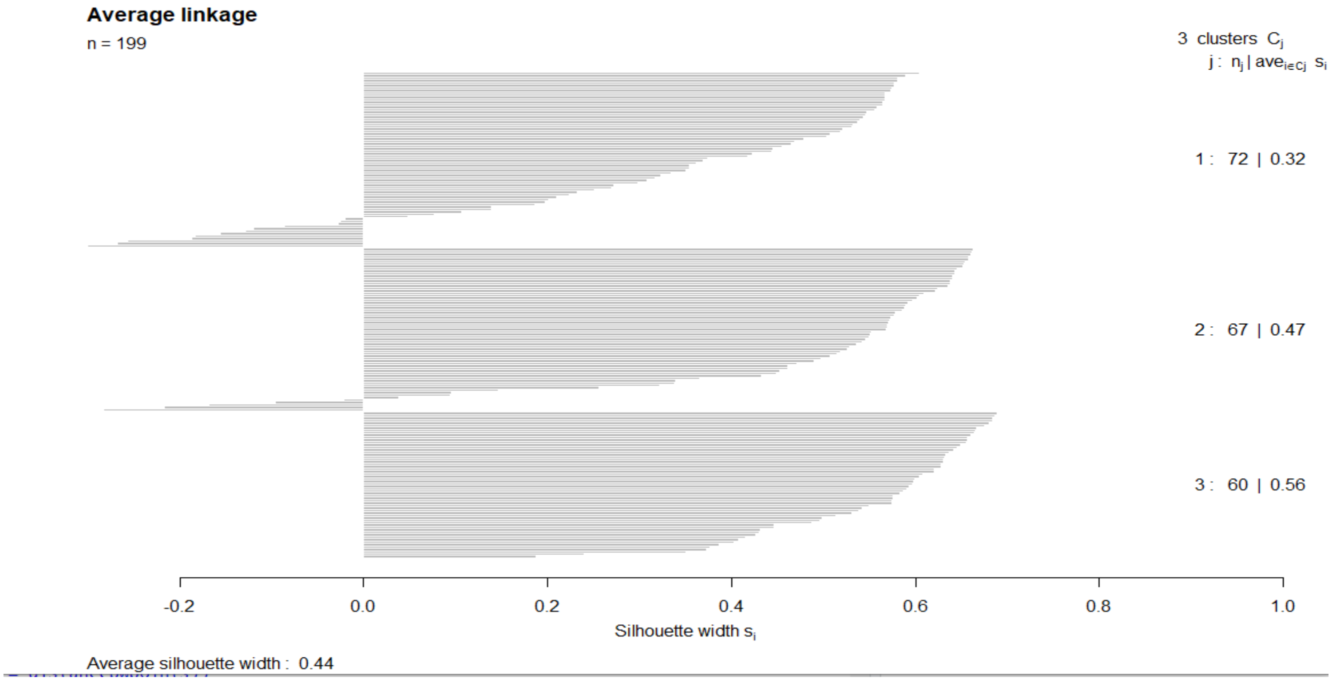
```
> k.select(seedsdata, range = 2:10, B = 100, r = 20, scheme_2 = F)
$profile
        2         3         4         5         6         7         8         9        10 
0.9198052 0.9058376 0.6045399 0.4609778 0.4876408 0.3634537 0.3506060 0.3234687 0.2685322 

$k
[1] 3

> 
```

We use the cuttree() function to cut the dendrogram for k = 3 to cluster the data into three groups respectively for all three methods.

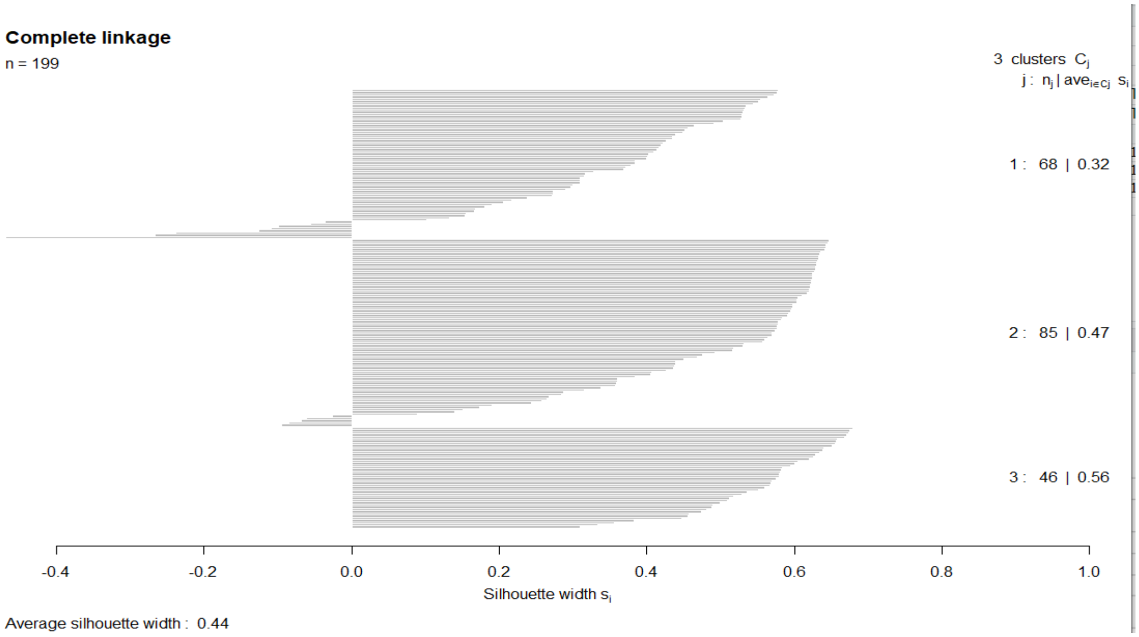
Silhouette plot for single linkage and three clusters,



Silhouette plot for Average linkage and three clusters,



Silhouette plot for Complete linkage and three clusters



We can see that the average silhouette width for average and complete linkage are almost the same, so we will evaluate using another metric.

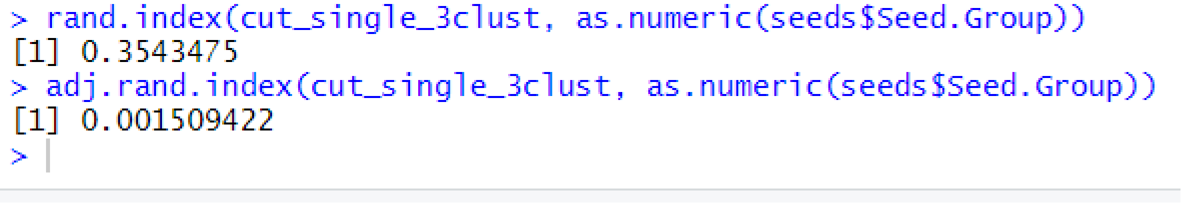
Evaluating using **Confusion-matrix** from **caret** package,

- The single linkage method has accuracy of 36.68%
- The average linkage method has accuracy of 90.95%
- The complete linkage method has accuracy of 23.12%

Out of all the three methods **average linkage method** seems to have **higher accuracy** and complete linkage has the least accuracy out of the three methods.

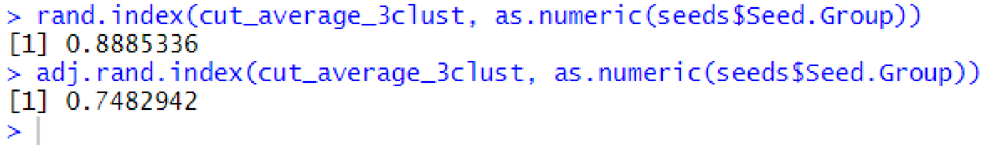
Evaluating using rand index and adjusted rand index,

- Rand index and Adjusted rand index for Single linkage method,



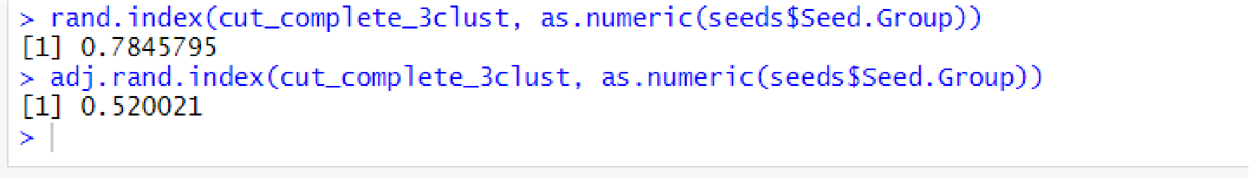
```
> rand.index(cut_single_3clust, as.numeric(seeds$Seed.Group))
[1] 0.3543475
> adj.rand.index(cut_single_3clust, as.numeric(seeds$Seed.Group))
[1] 0.001509422
> 
```

- Rand index and Adjusted rand index for Average linkage method,



```
> rand.index(cut_average_3clust, as.numeric(seeds$Seed.Group))
[1] 0.8885336
> adj.rand.index(cut_average_3clust, as.numeric(seeds$Seed.Group))
[1] 0.7482942
> 
```

- Rand index and Adjusted rand index for Complete linkage method,



```
> rand.index(cut_complete_3clust, as.numeric(seeds$Seed.Group))
[1] 0.7845795
> adj.rand.index(cut_complete_3clust, as.numeric(seeds$Seed.Group))
[1] 0.520021
> 
```

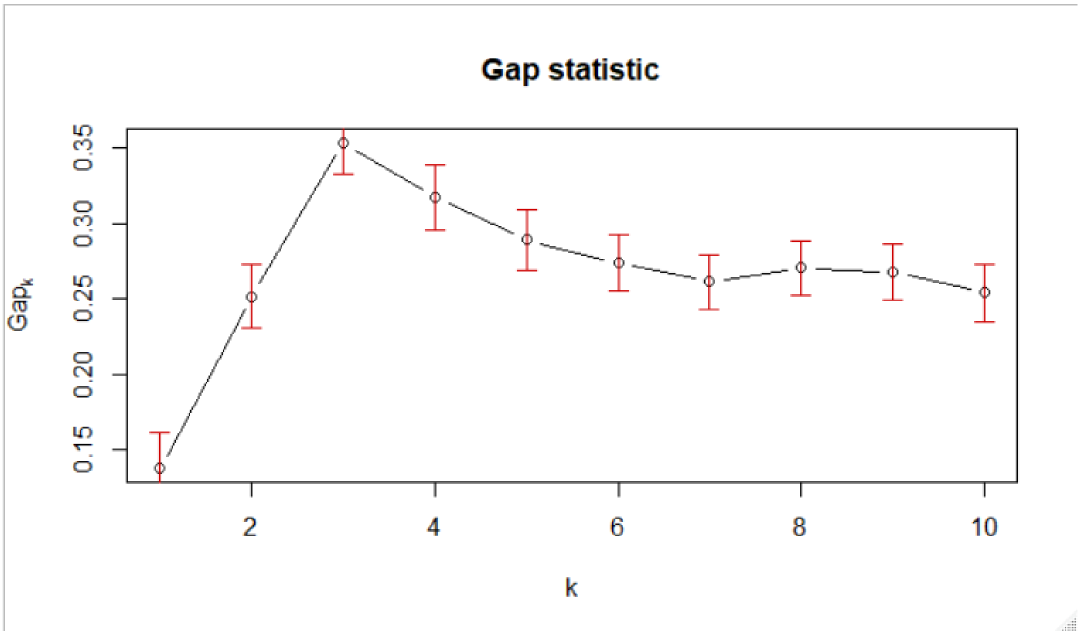
From the rand index and adjusted rand index values we can say that Average linkage method has the highest value and has the best performance out of all three and the single linkage method has performed worst in comparison of others.

I was expecting the complete linkage method to perform the best since the algorithm calculates the distance between elements that are farthest away from each other in the clusters, but it had turned out average linkage method gives the best performance for this data.

(b). Cluster the data based on K-means or K-medoids. Use an analytical technique to justify your choice in "k". How did the performance compare to the hierarchical clustering of part a? Which did you feel was a better method for this data?

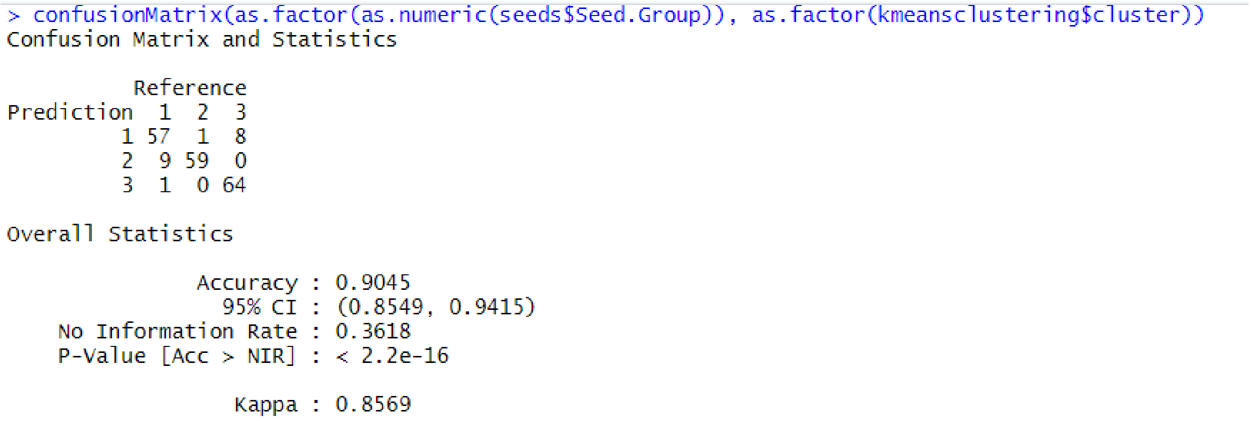
Using the bootstrapping method, we already found out the optimal number of clusters is **3**, to confirm it again we use the gap statistic method to find the number of clusters.

Finding optimal number of clusters using Gap statistic,



We find that the optimal number of clusters is 3 from the above plot. So, we run k-means clustering with k equals to 3.

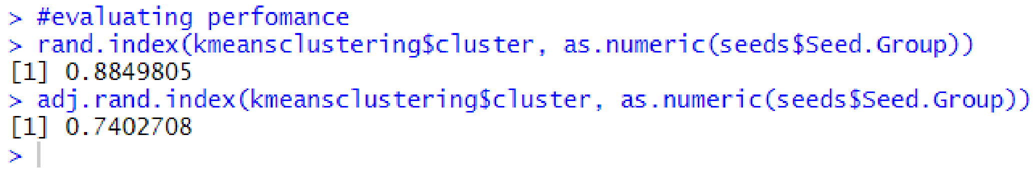
Assessing k-means performance,



```
> confusionMatrix(as.factor(as.numeric(seeds$Seed.Group)), as.factor(kmeansclustering$cluster))
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 57  1  8
         2  9 59  0
         3  1  0 64

Overall Statistics
                                          
               Accuracy : 0.9045          
                 95% CI : (0.8549, 0.9415)
    No Information Rate : 0.3618          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.8569          
```

From the confusion matrix we can see that k means has an accuracy of 90.45% which is 0.10% less compared to hierarchical clustering using average linkage.

Also calculating the rand and adjusted rand index,



```
> #evaluating perfomance
> rand.index(kmeansclustering$cluster, as.numeric(seeds$Seed.Group))
[1] 0.8849805
> adj.rand.index(kmeansclustering$cluster, as.numeric(seeds$Seed.Group))
[1] 0.7402708
> 
```

Inference and conclusion,

- We can see that k means has slightly less score than hierarchical clustering using average linkage.
- From this we can conclude that hierarchical clustering using average linkage is a better method for this data.
- However, in general, I would like to go with hierarchical clustering rather than k means since k means require us to have knowledge about the data and how many clusters to be formed before running the algorithm.
- In hierarchical clustering we end up with a dendrogram and we can later decide how many clusters by looking/cutting at different heights or can be grouped based on business requirement.