Homework 2
Applied Machine Learning
Fall 2017
CSCI-P 556/INFO-I 526

Manoj Joshi
manjoshi@iu.edu

October 6, 2017

"All the work herein is solely mine." - Manoj Joshi

Problem 1 [20 points]

From textbook, Chapter 10 exercise 2 (Page 414).
a) We have the matrix

$$M = \begin{bmatrix} 0 & 0.3 & 0.4 & 0.7 \\ 0.3 & 0 & 0.5 & 0.8 \\ 0.4 & 0.5 & 0 & 0.45 \\ 0.7 & 0.8 & 0.45 & 0 \end{bmatrix}$$

The matrix is symmetric with respect to the diagonal elements. We can just look at the lower half(below the diagonal) of the matrix to form clusters. In the above matrix, the minimum value is 0.3 which corresponds to points $1, 2$. **So, we can combine points $1, 2$ to form a single cluster at height** 0.3
After combining, we are left with the below matrix:

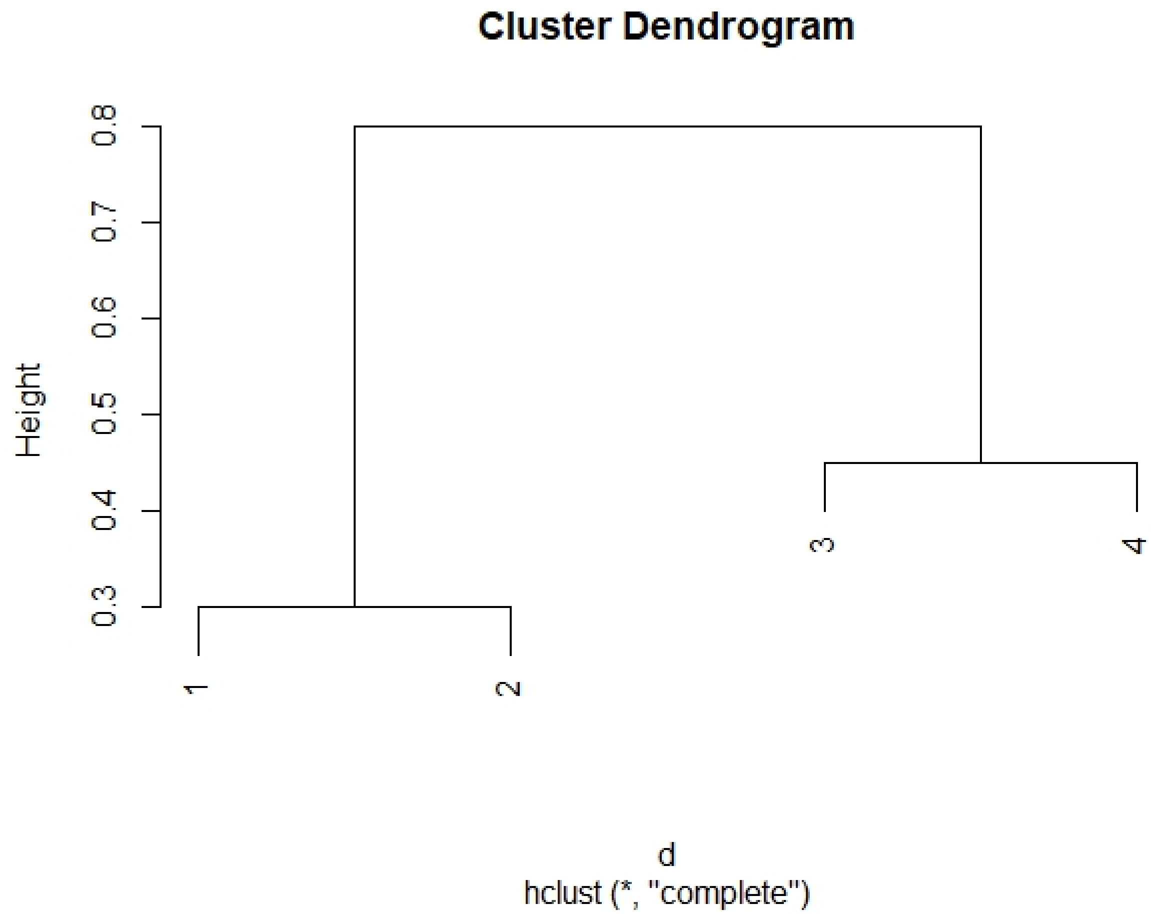
$$M = \begin{bmatrix} 0 & - & - & - \\ - & 0 & 0.5 & 0.8 \\ - & 0.5 & 0 & 0.45 \\ - & 0.8 & 0.45 & 0 \end{bmatrix}$$

In the above matrix, we have 0.45 as the minimum value which corresponds to the points $3, 4$. **So, we can combine points $3, 4$ to form a single cluster at height** 0.45
After combining, we are left with the below matrix:

$$M = \begin{bmatrix} 0 & - & - & - \\ - & 0 & - & 0.8 \\ - & - & 0 & - \\ - & 0.8 & - & 0 \end{bmatrix}$$

Now we are left with only 0.8 at the bottom. We have to combine points $2, 4$. But $1, 2$ are clustered together and $3, 4$ are clustered together. **So, we combine these two clusters to form one cluster at height 0.8.**

Below is the dendogram using R.



b) We have the matrix

$$M = \begin{bmatrix} 0 & 0.3 & 0.4 & 0.7 \\ 0.3 & 0 & 0.5 & 0.8 \\ 0.4 & 0.5 & 0 & 0.45 \\ 0.7 & 0.8 & 0.45 & 0 \end{bmatrix}$$

The matrix is symmetric with respect to the diagonal elements. We can just look at the lower half(below the diagonal) of the matrix to form clusters. In the above matrix, the minimum dissimilarity is 0.3 which corresponds to points $1, 2$. **So, we can combine points** $1, 2$ **to form a single cluster at height** 0.3
After combining, we are left with the below matrix:

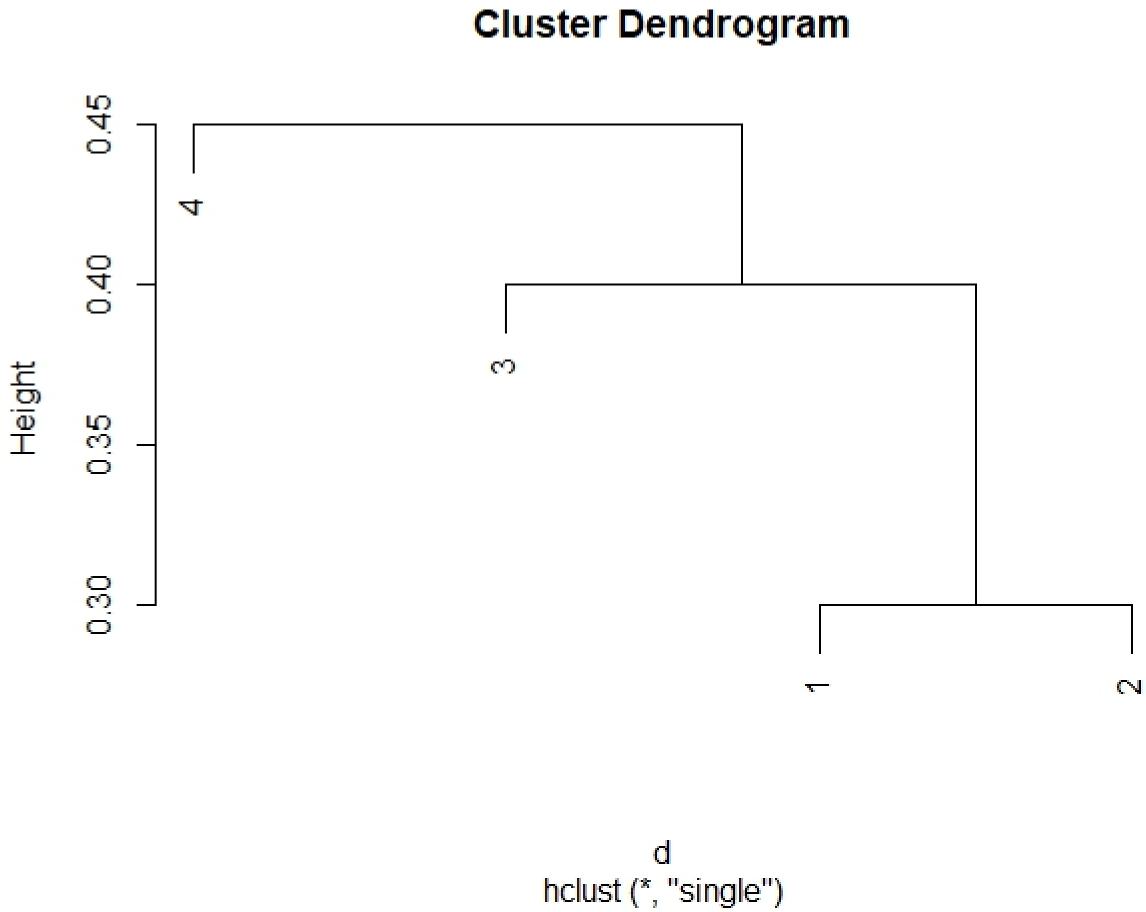
$$M = \begin{bmatrix} 0 & 0.4 & 0.7 \\ 0.4 & 0 & 0.45 \\ 0.7 & 0.45 & 0 \end{bmatrix}$$

In the above matrix, we have 0.4 as the minimum dissimilarity which corresponds to the point 3. **So, we can combine point** 3 **with** $(1, 2)$ **to form a single cluster** $((1, 2), 3)$ **at height** 0.4
After combining, we are left with the below matrix:

$$M = \begin{bmatrix} 0 & 0.45 \\ 0.45 & 0 \end{bmatrix}$$

Now we are left with only 0.45 at the bottom. We have to combine points $((1, 2), 3))$ and 4 to get a **final cluster at height** 0.45**.**

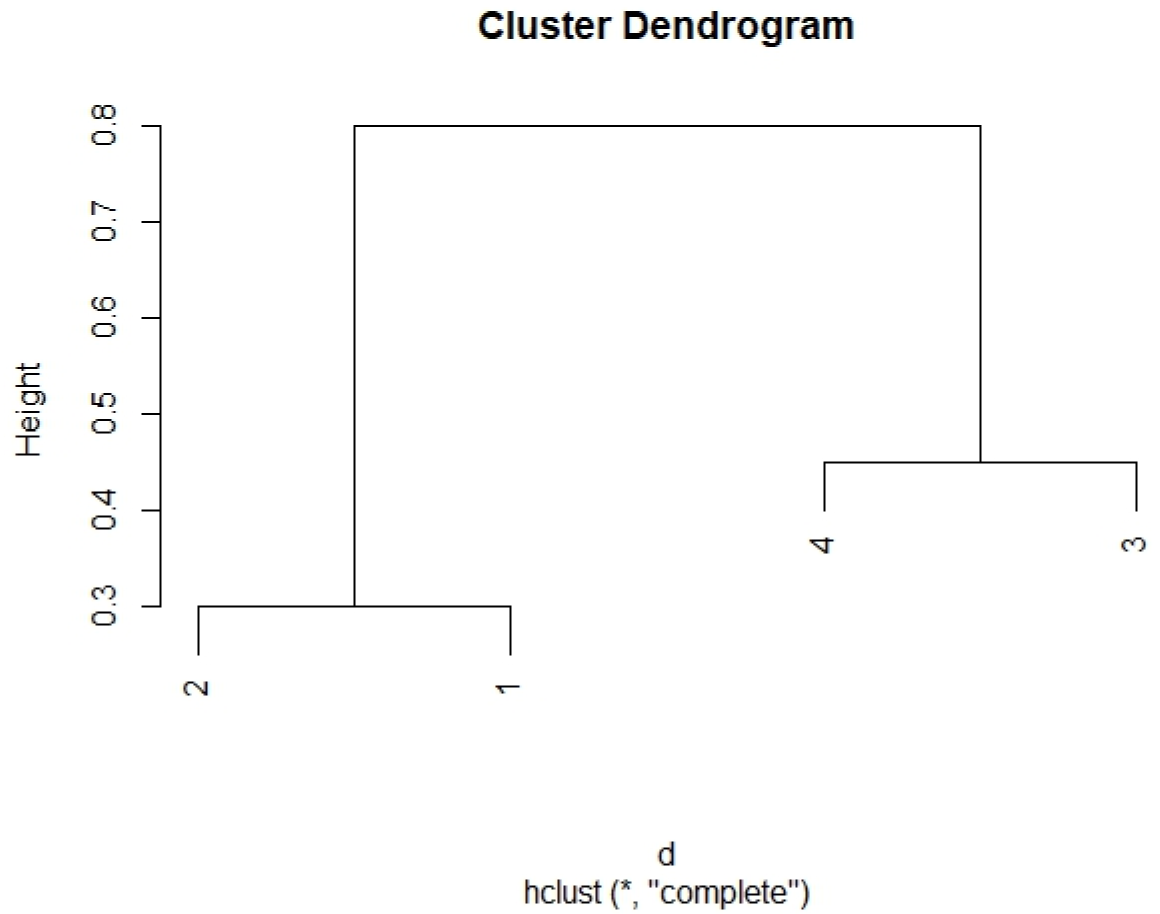
Below is the dendogram using R.



c) If we want two clusters from dendogram in a) , we see that observations 1 and 2 are in ”cluster 1” and observations 2 and 3 are in ”cluster 2”.

d) If we want two clusters from dendogram in b) , we see that observations 1, 2 and 3 are in ”cluster 1” and observation 4 is in ”cluster 2”.

e)From the dendogram below, 1 and 2 are swapped, 3 and 4 are swapped but dendogram remains same.



Problem 2 [50 points]

Implement expectation-maximization algorithm for Gaussian mixture models (see the EM algorithm below) in *R* and call this program G_k. As you present your code explain your protocol for

3.1 initializing each Gaussian
To initialize k Gaussians, I use the sample function to get k values which lie between 1 and (number of rows) of input. The k numbers generated are used as indices of the input data. These k data points act as the initial values for the Gaussian distribution. Basically, I am selecting k data point from the input.

3.2 maintaining k Gaussian
Since EM does not do hard assignment, every point will belong to every cluster with some probability. Therefore, $w_{i,j}$ will not be zero.
For each i, $i = 1, 2, 3..k$ - I multiply the "weight matrix" $w_(i, j)$ with $x_(j)$ and sum it for all "j"'s. I divide this by the sum of "weight matrix" $w_(i, j)$ for all "j"'s. This process returns me a 1 x d matrix for each i where "d" is the dimension of each data point. After repeating the process for all $i = 1, 2, 3..k$, I maintain the k Gaussians.

3.3 deciding ties
EM does not do hard assignment which means every point belongs to each cluster with some probability.

If we decide to do a hard assignment based on the probability values, we can assign a point to the cluster which has the maximum probability. If there is a tie in the probabilities between two clusters, we can assign the point to any of the two clusters randomly.

3.4 stopping criteria
There are two ways in which the algorithm can terminate:

a : I store all the $(i-1)$TH iteration mu(Gaussian) in a matrix called "previous-mu" and all the (i)TH iteration mu in a matrix called "mu". I calculate the distance between each vector in "previous-mu" and "mu" and sum over the entire matrix. If the value is less than a threshold value, I stop the algorithm at that iteration. What this means is that we have found k Gaussians that fit the data correctly.

b : The algorithm did not meet the criteria mentioned in a) but has completed the maximum number of iterations specified, this is when the algorithm terminates.

Problem 3 [70 points]

In this questions, you are asked to run your program, G_k, against the Ringnorm and Ionosphere data sets and compare G_k with C_k (k-means algorithm from previous homework). Click on the below links to download the data sets.

- Ringnorm Data Set
- Ionosphere Data Set

Answer the following questions:

3.1 Initialize G_k and C_k with the same set of initial points (initial centroids for C_k and μ_i-s for G_k are identical) and run them for $k = 2, \ldots, 5$ for 20 runs each. Report error rates and iteration counts for each k using whisker plots that reveal comparison of C_k and G_k. An example of whisker plot is given below. A simple error rate can be calculated as follows:

- If $k = 2$: C_k and G_k will predict two clusters. Error calculation is trivial for two clusters.
- If $k > 2$: after C_k and G_k converge, combine the clusters as follows to ended up with two clusters: since the true clusters are known for a given arbitrary blocks number, final clusters are determined by measuring the Euclidean (this is the easiest choice) distances between true cluster centers and predicted cluster centers.

In other words, you will always calculate the error for $k = 2$ since there are only 2 clusters in the given data sets. Below is an example of error calculation for Ionosphere data set. You can similarly calculate an error rate for Ringnorm data set.

For each centroid C_i, and each Gaussian G_k form two counts (over Ionosphere Data Set) :

$$\begin{aligned} g_i &\leftarrow \sum_{\delta \in c_i.B} [\delta.C = \text{“g”}], \quad \text{good} \\ b_i &\leftarrow \sum_{\delta \in c_i.B} [\delta.C == \text{“b”}], \quad \text{bad} \end{aligned}$$

where $[x = y]$ returns 1 if True, 0 otherwise. For example, $[2 = 3] + [0 = 0] + [34 = 34] = 2$

The centroid C_i and Gaussian G_k is classified as good if $g_i > b_i$ and bad otherwise. We can now calculate a simple error rate. Assume C_i is good. Then the error is:

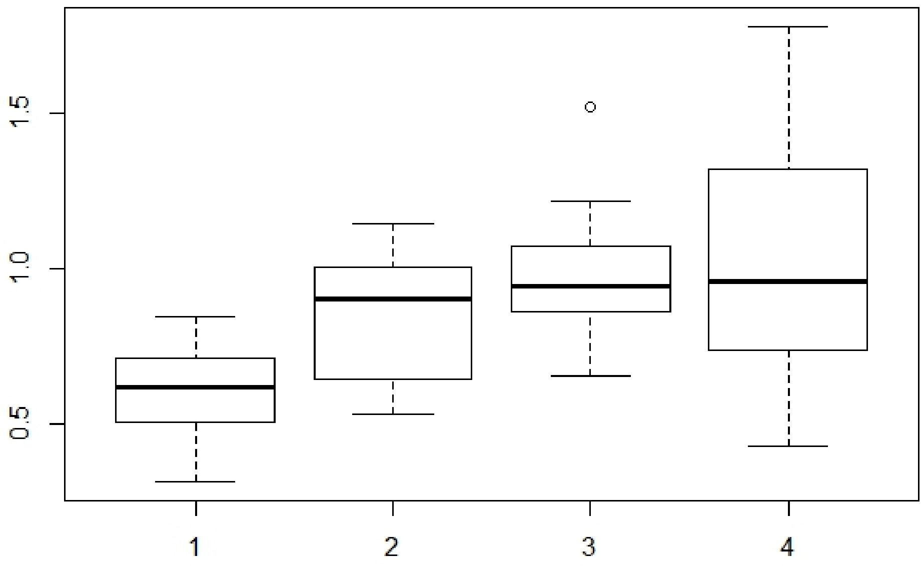
$$error(C_i) \quad = \quad \frac{b_i}{b_i + g_i} \quad \text{[same for error}(G_i)\text{]}$$

We can find the total error rate easily:

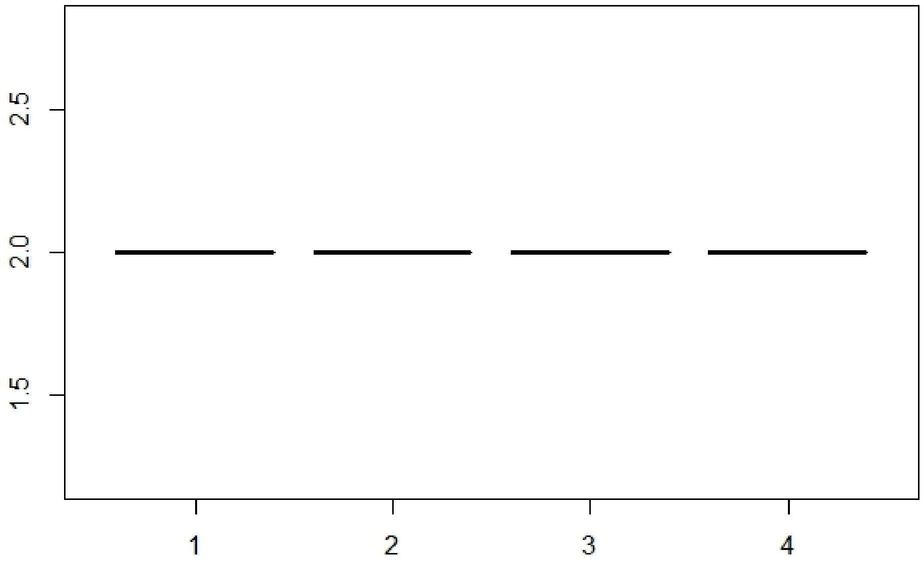
$$Error(\{C_1, C_2\}) \quad = \quad \sum_{i=1}^{2} error(C_i)$$

Discuss your results, i.e., which one performs better.

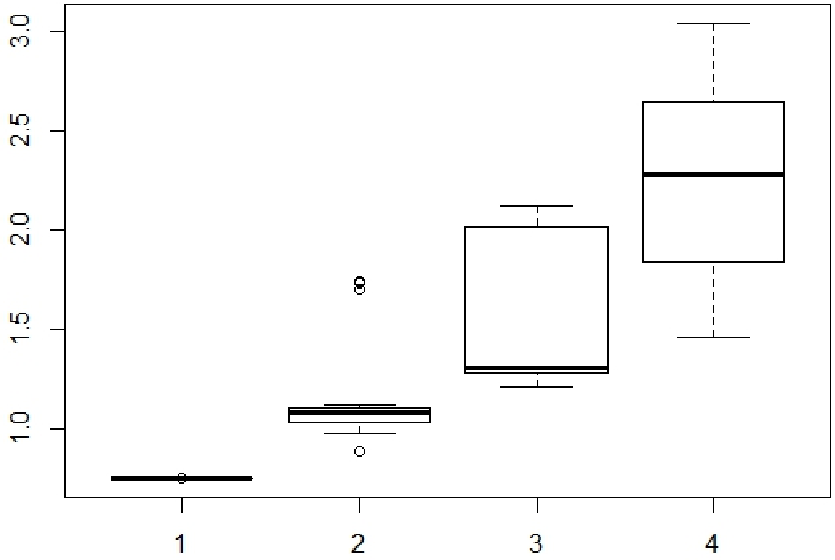
Below are the plots for Ionosphere dataset using G_k
Plot of k versus Error



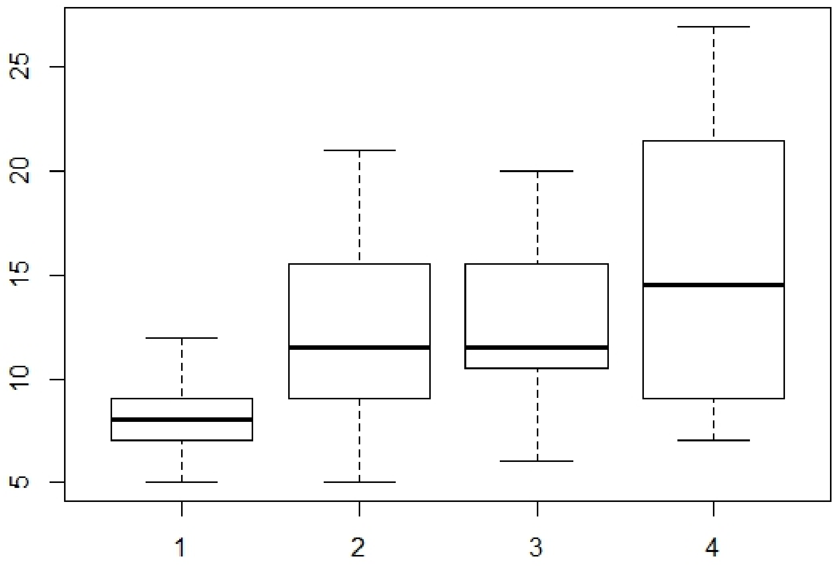
Plot of k versus Iteration count



Below are the plots for Ionosphere dataset using C_k
Plot of k versus Error

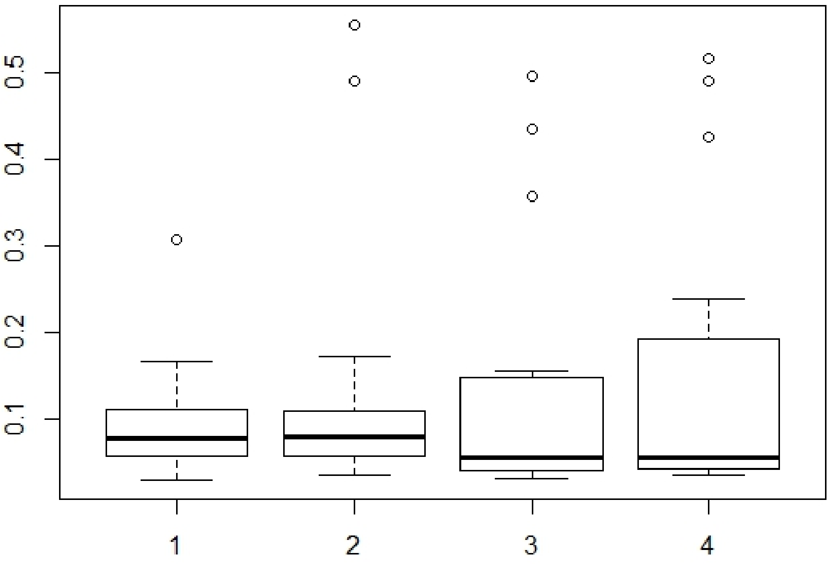


Plot of k versus Iteration count

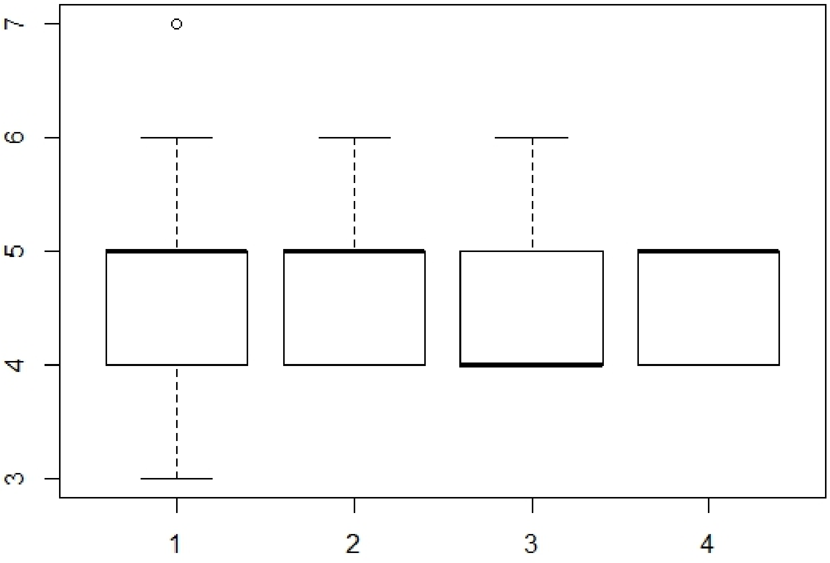


Looking at the plots above, it is clear that error rate for G_k(EM) is less compared to C_k(k-means). For $k = 5$ (marked as 4th point in the error box plot), the error for G_k does not cross 2 but in C_k, the error goes above 3 for the same k ($k = 5$). In terms of the error rate, G_k outperforms C_k for Ionosphere dataset.

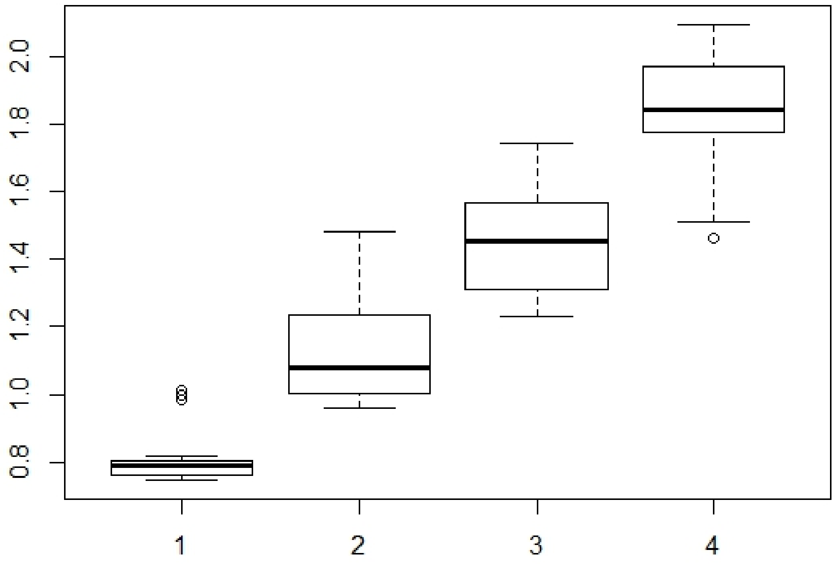
Below are the plots for Ringnorm dataset using G_k
Plot of k versus Error



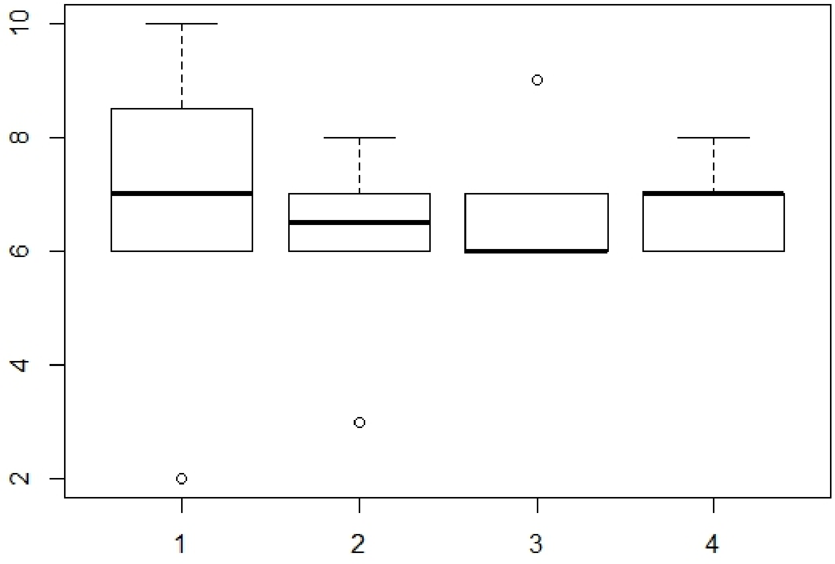
Plot of k versus Iteration count



Below are the plots for Ringnorm dataset using C_k
Plot of k versus Error



Plot of k versus Iteration count

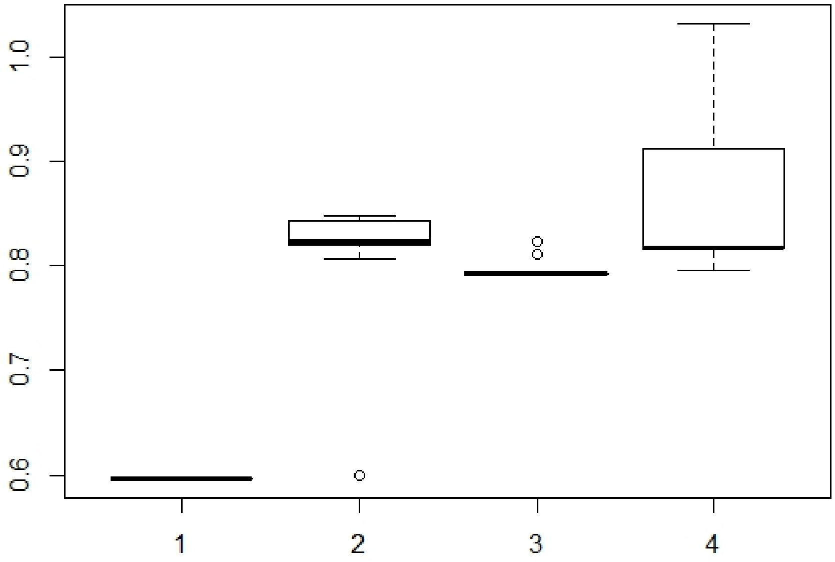


Looking at the plots above, it is clear that error rate for G_k(EM) is very less compared to C_k(k-means). For every k, the error in G_k is less compared to C_k. In terms of the error

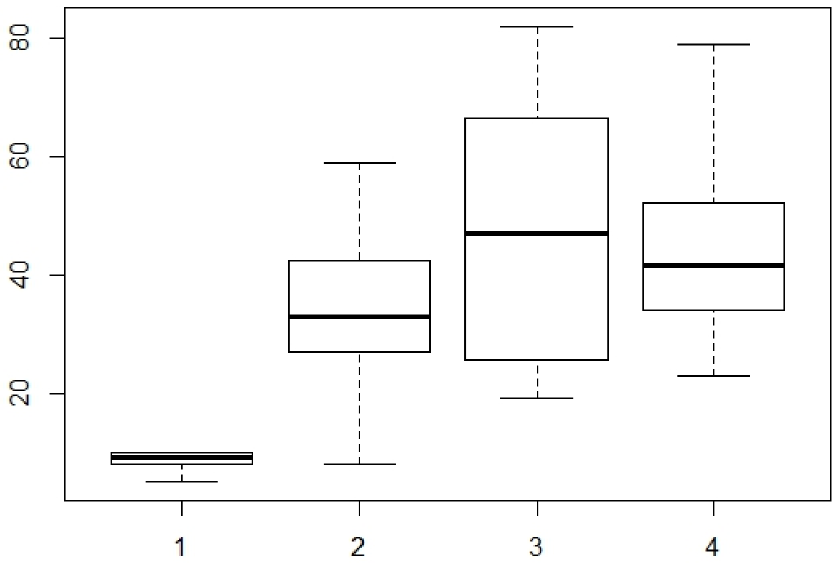
rate, G_k outperforms C_k for Ringnorm dataset.

3.2 In this question, we will run your G_k with fixing the variances to ones and the priors to be uniform. Do not update the variances and priors throughout iterations. As explained in question 3.1, compare your new G_k and C_k using whisker plots. Discuss your results, i.e., which one performed better.

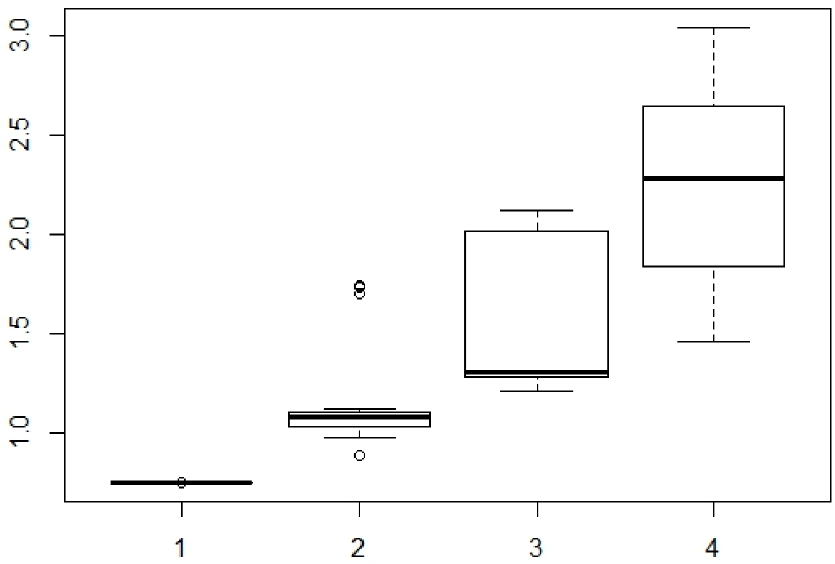
Below are the plots for Ionosphere dataset using G_k
Plot of k versus Error



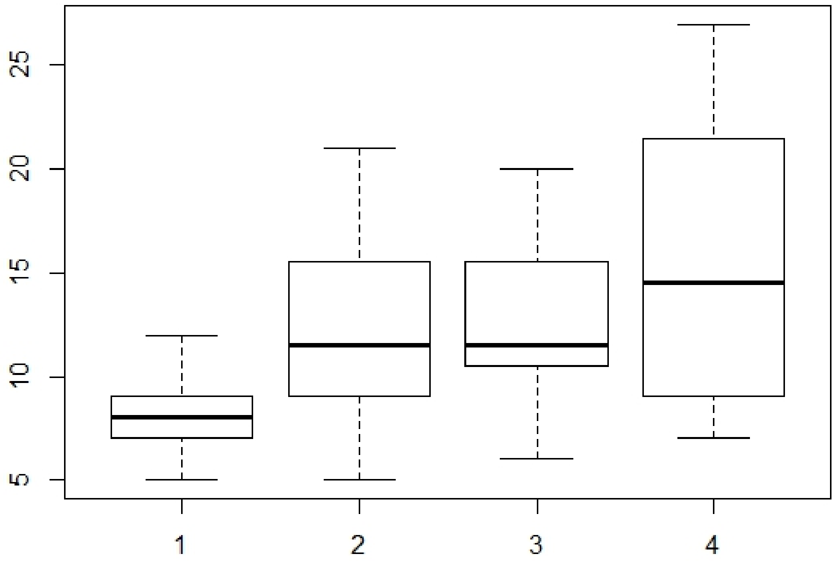
Plot of k versus Iteration count



Below are the plots for Ionosphere dataset using C_k
Plot of k versus Error

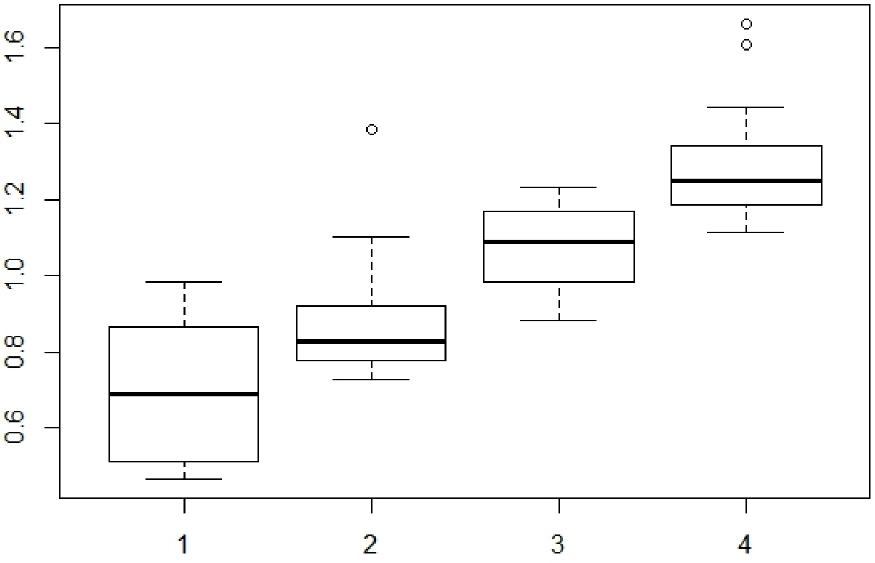


Plot of k versus Iteration count

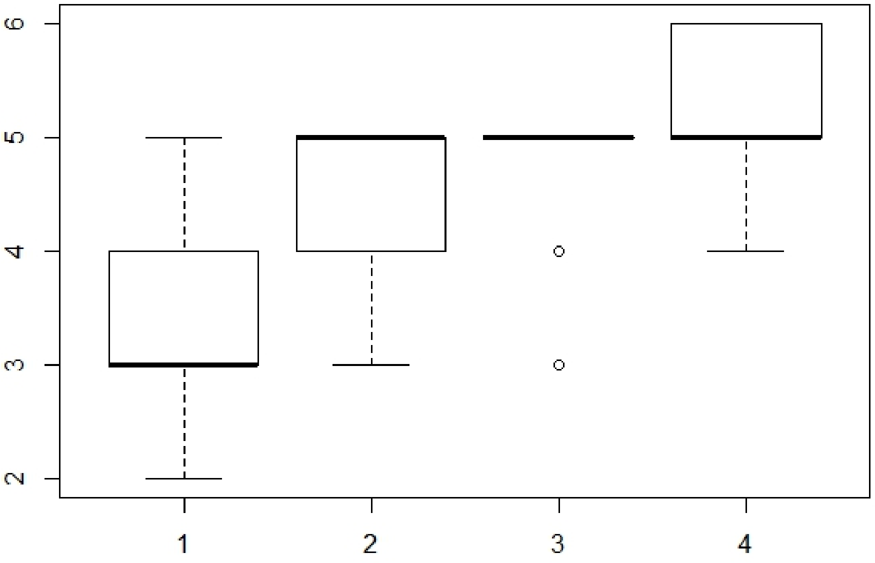


Looking at the plots above, it is clear that error rate for G_k(EM) is less compared to C_k(k-means). But, compared to G_k in 3.1, the EM here took more time to converge since it did involve updating the priors and variance. Even the C_k converged faster than the G_k in this case. We can conclude that updating the variance and priors help in faster convergence.

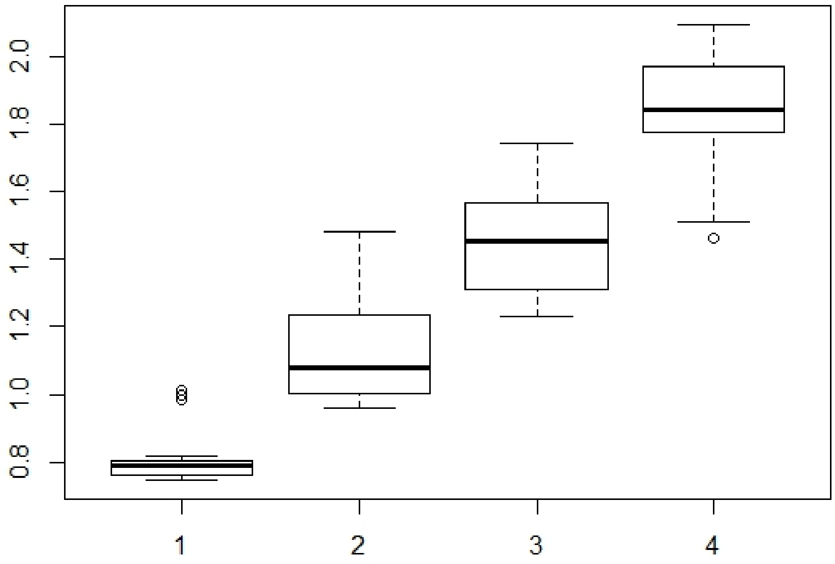
Below are the plots for Ringnorm dataset using G_k
Plot of k versus Error



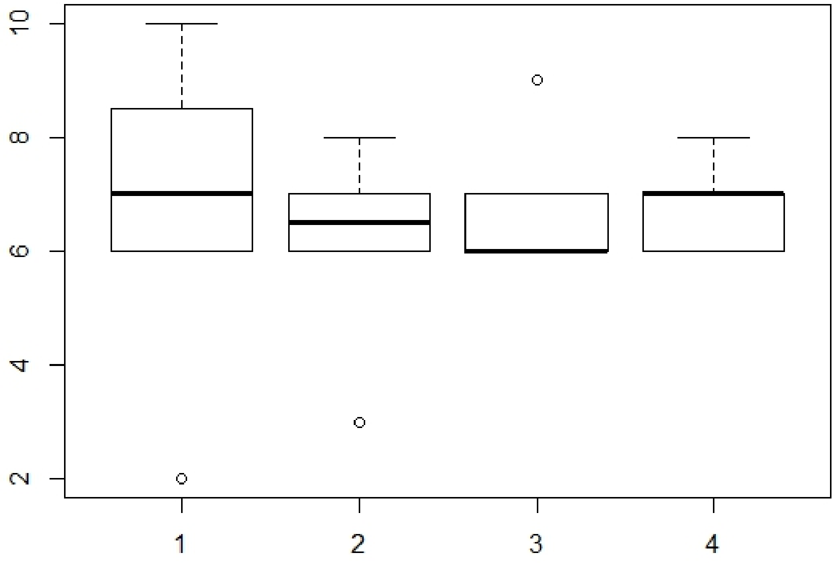
Plot of k versus Iteration count



Below are the plots for Ringnorm dataset using C_k
Plot of k versus Error



Plot of k versus Iteration count



Looking at the plots above, it is clear that error rate for G_k(EM) is less compared to C_k(k-means). But, compared to G_k in 3.1, the EM here has a little more.

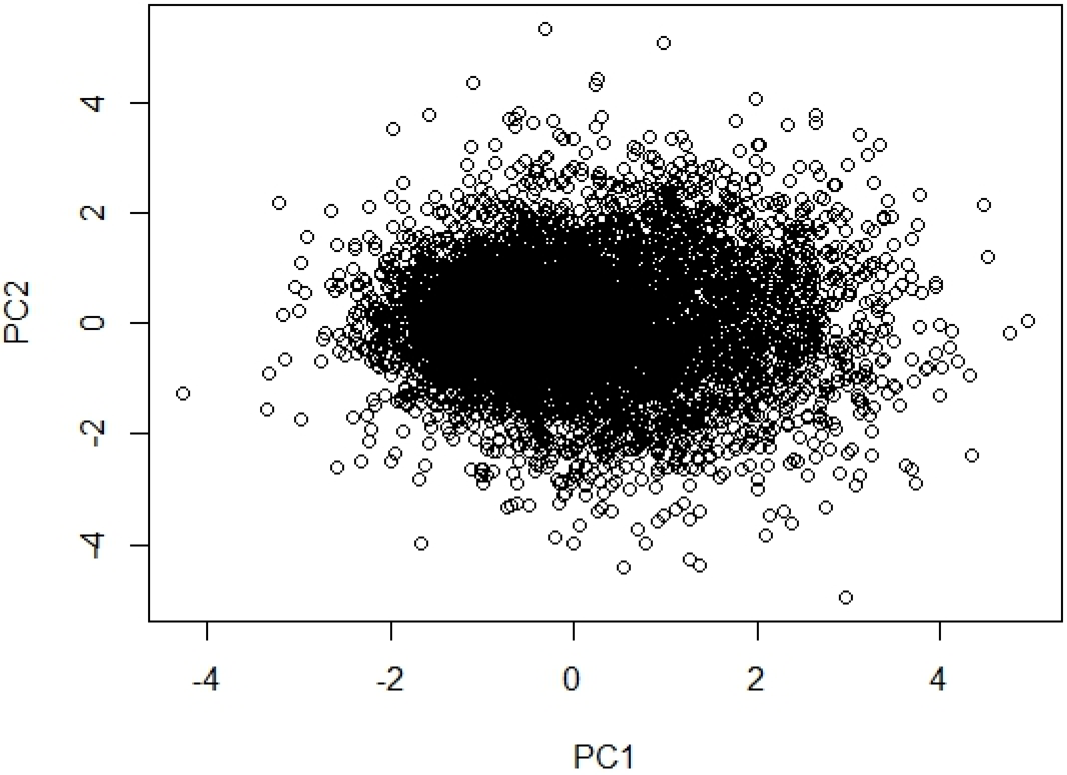
To conclude from both the experiments, updating priors and variance is necessary to have a faster convergence and also to have a better error rate.

Problem 4 [50 points]

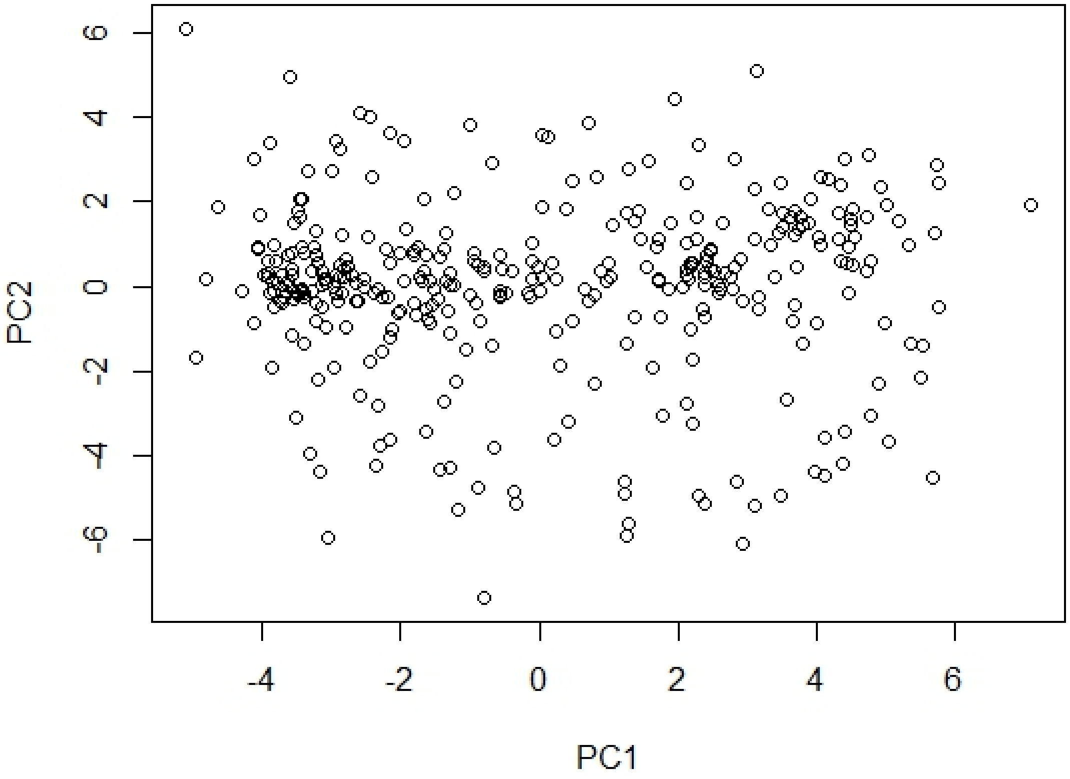
In this question, you will first perform principal component analysis (PCA) over Ionosphere and Rignorm data sets and then cluster the reduced data sets using G_k (from question 3.1) and C_k. You are allowed to use R packages for PCA. Ignore the class variables (35th and 1st variables for Ionosphere and Ringnorm data sets, respectively) while performing PCA. Answer the questions below:

4.1 Make a scatter plot of PC1 and PC2 for both data sets. Discuss principal components (The first and second principal components). What are PC1 and PC2?

Find the scatter plot of PC1 and PC2 of Ringnorm data set below.



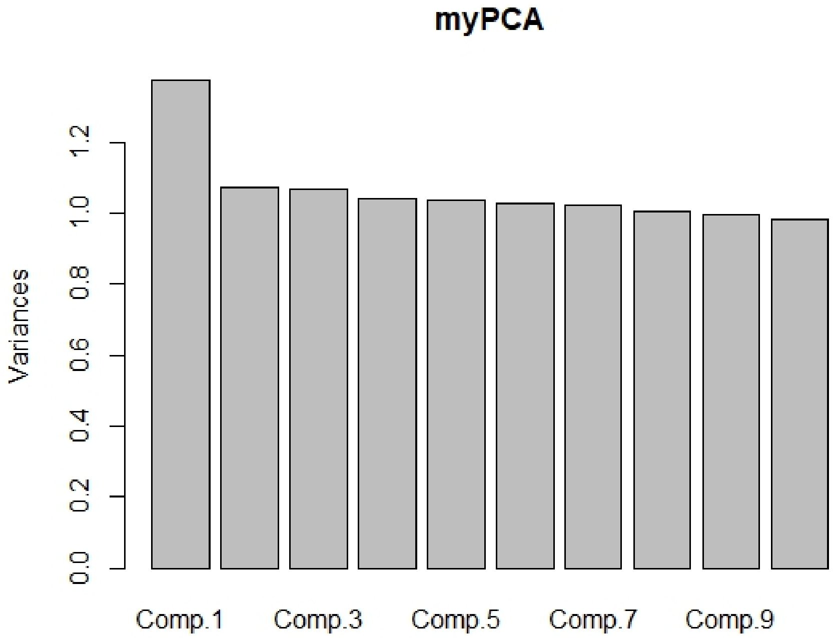
Find the scatter plot of PC1 and PC2 of Ionosphere data set below.



PC1 and PC2 are the two principal components of the data. These two features capture most of the variance in the data. The principal components are ordered according to the degree of variance that they capture from the data.

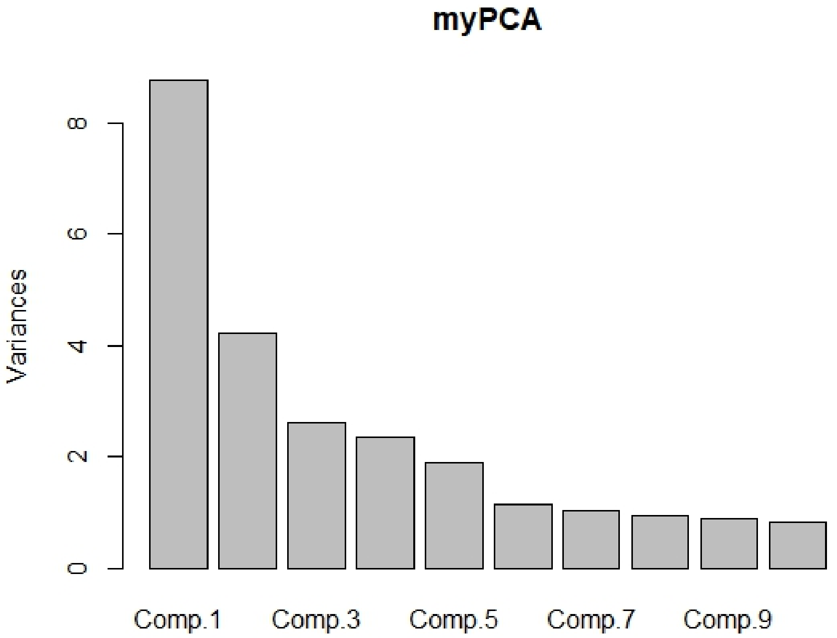
4.2 Create scree plots after PCA and explain the plots.

Find the scree plot of PC1 and PC2 of Ringnorm data set below.



From the plot, we can see that the first 10 components capture most of the variance of the data. The rest of the components capture less variance and hence they are not shown in the plot.

Find the scree plot of PC1 and PC2 of Ionosphere data set below.



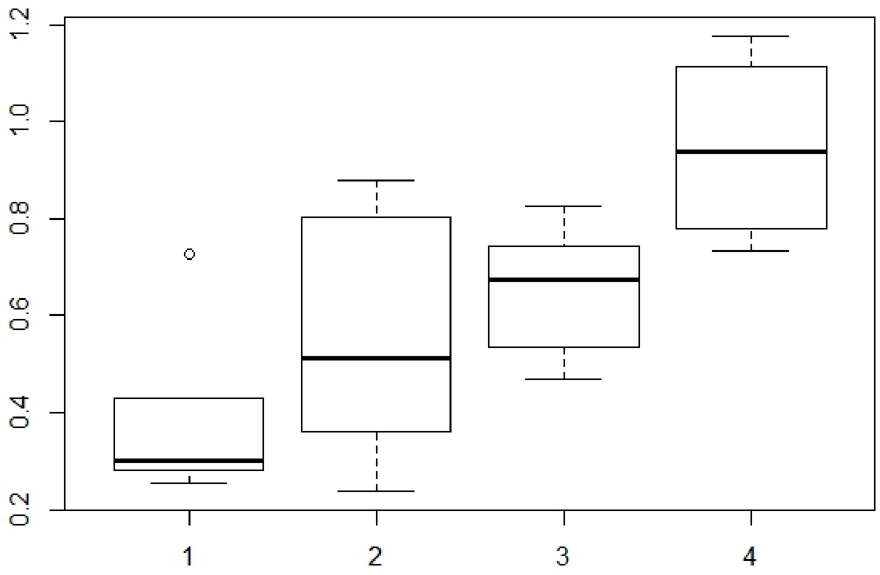
From the plot, we can see that the first 2 components capture most of the variance of the data. As we proceed from the second principal component, the variance decreases. It seems that PCA1 and PCA2 themselves explain most of the variance in the data.

4.3 Observe the loadings using prcomp() or princomp() functions in R and discuss loadings in PCA?i.e., how are principal components and original variables related?

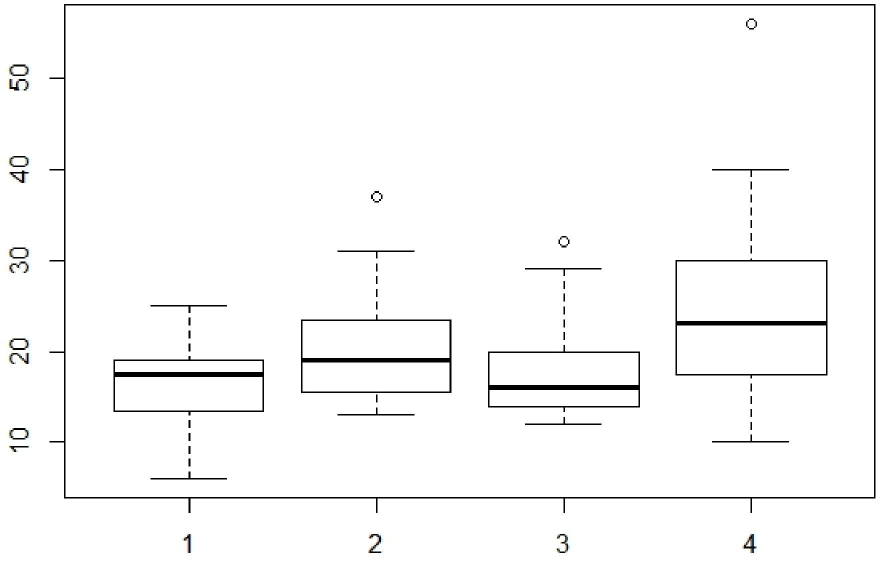
The loadings of PCA represent the components in a different co-ordinate system where the eigen vectors are orthogonal to each other. These components will be arranged in decreasing order of the variance that they capture from the data.
Each original variable can contribute to every principal component to some degree i.e., since PC1 captures most of the variance, probably each original variable has some contribution to it.

4.4 Keep 90% of variance after PCA and reduce Ionosphere and Rignorm data sets. Run C_k and G_k with the reduced data sets and compare them using whisker plots as shown in question 3.1

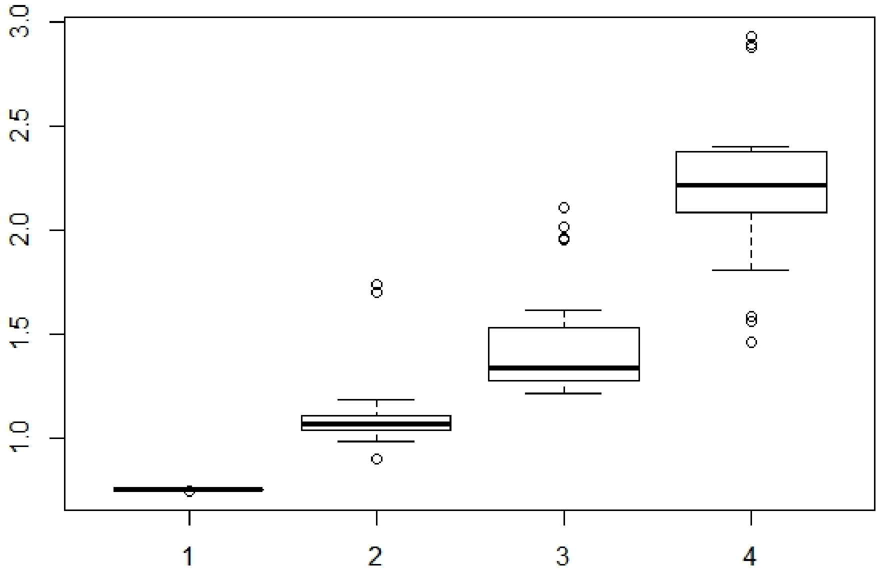
Below are the plots for Ionosphere dataset using G_k
Plot of k versus Error



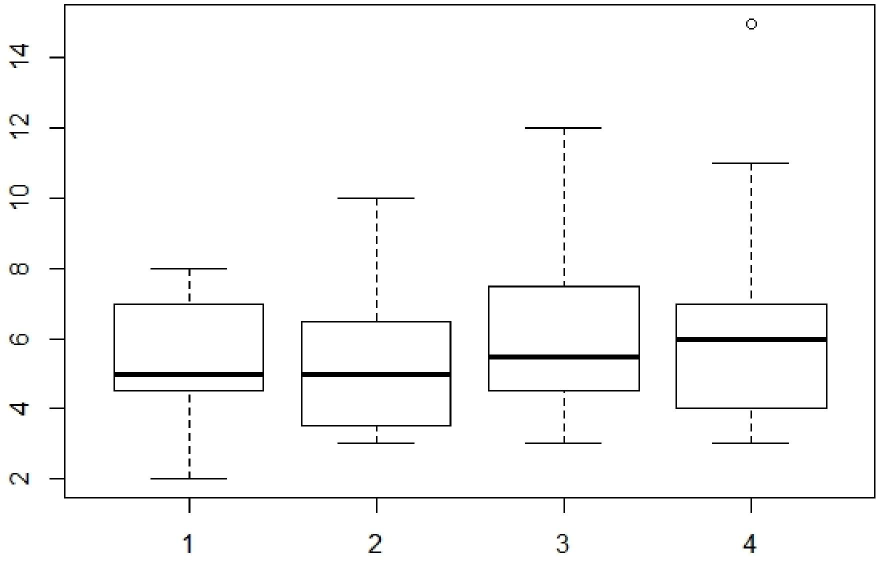
Plot of k versus Iteration count



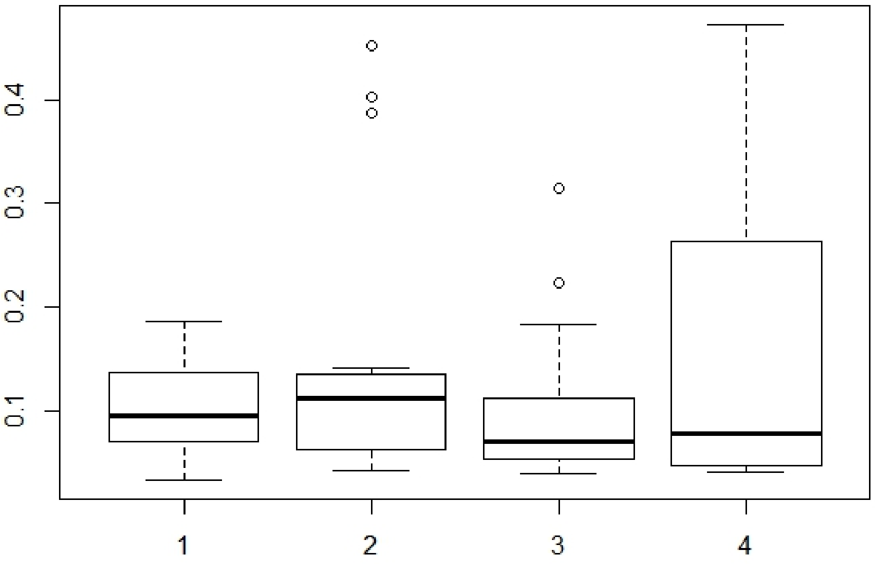
Below are the plots for Ionosphere dataset using C_k
Plot of k versus Error



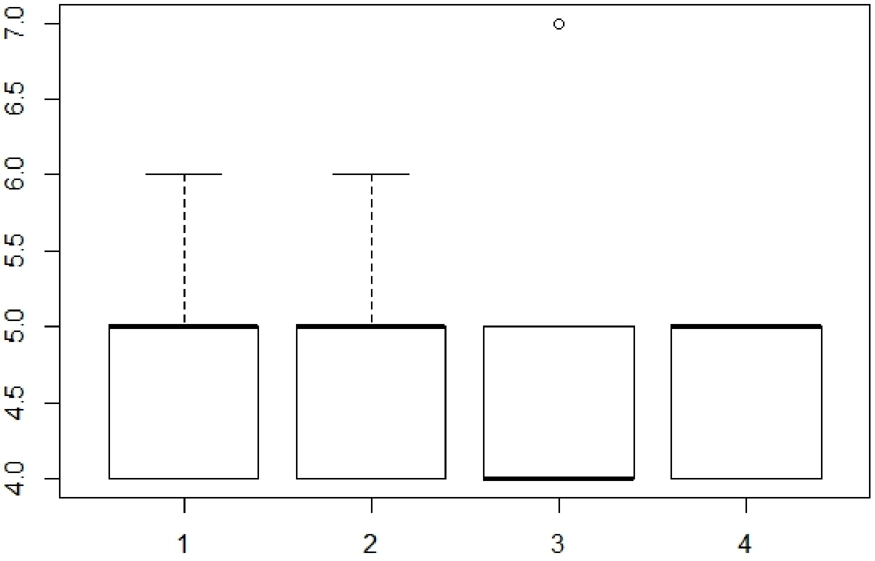
Plot of k versus Iteration count



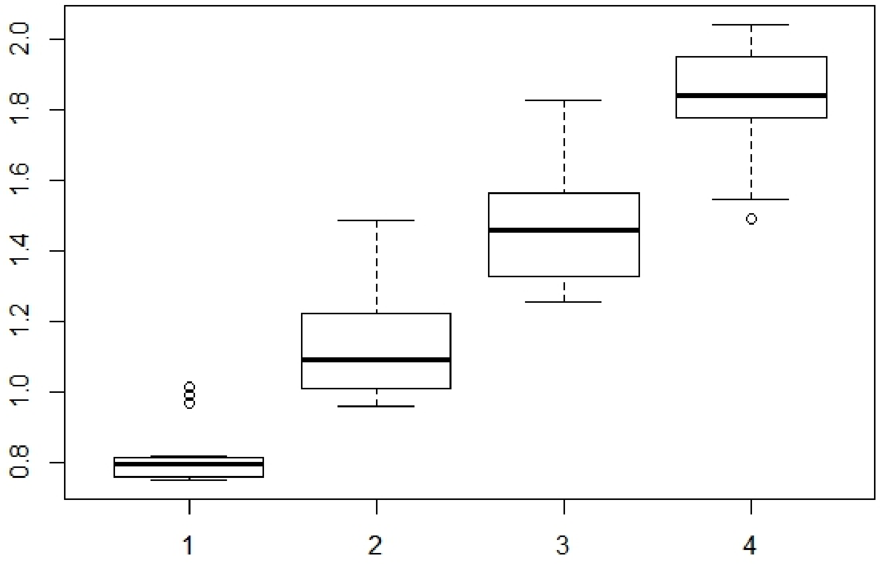
Below are the plots for Ringnorm dataset using G_k
Plot of k versus Error



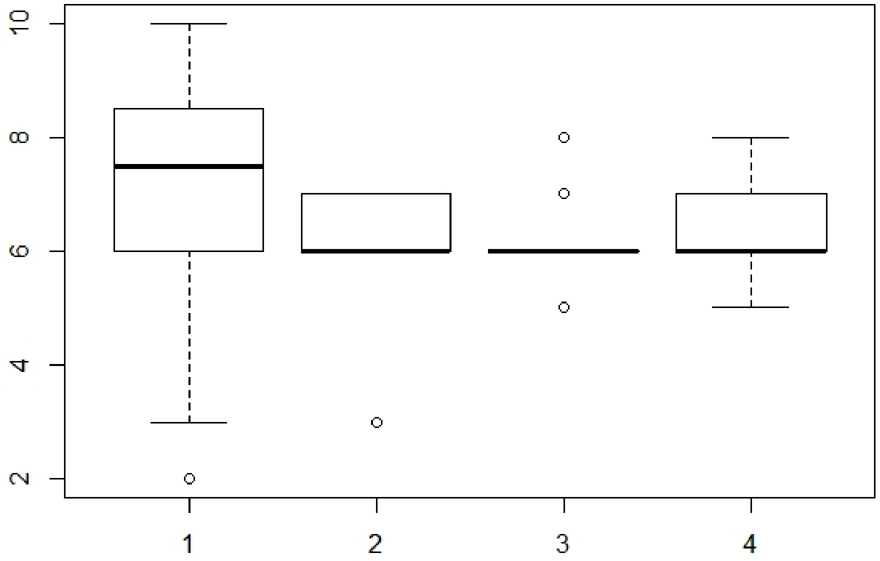
Plot of k versus Iteration count



Below are the plots for Ringnorm dataset using C_k
Plot of k versus Error



Plot of k versus Iteration count



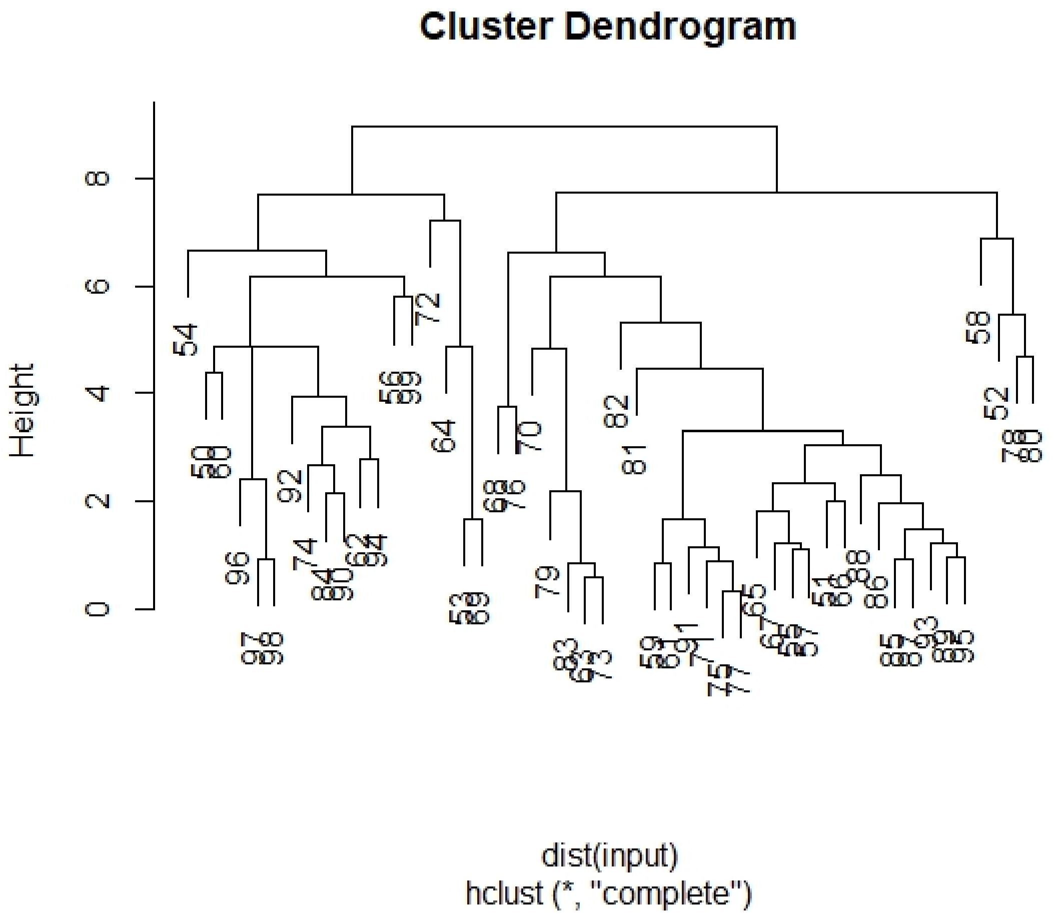
4.5 Discuss that how PCA affects the performance of C_k and G_k.

From the plots we can observe that the error for C_k and G_k have reduced to some extent. But most importantly, both C_k and G_k have converged faster. This is because we have taken only the components which have 90*percent* variance. This leaves us with less features for computation. Hence the faster convergence.

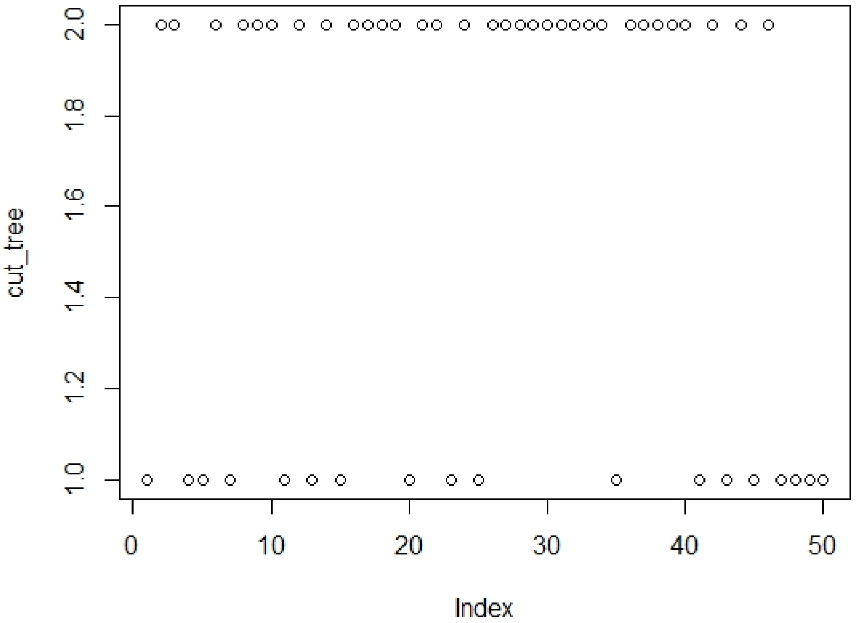
Problem 5 [50 points]

Randomly choose 50 points from Ionosphere data set (call this data set I_{50}) and perform hierarchical clustering. You are allowed to use R packages for this question. (Ignore the class variable while performing hierarchical clustering.)

5.1 Using hierarchical clustering with complete linkage and Euclidean distance cluster I_{50}. Plot the dendrogram.

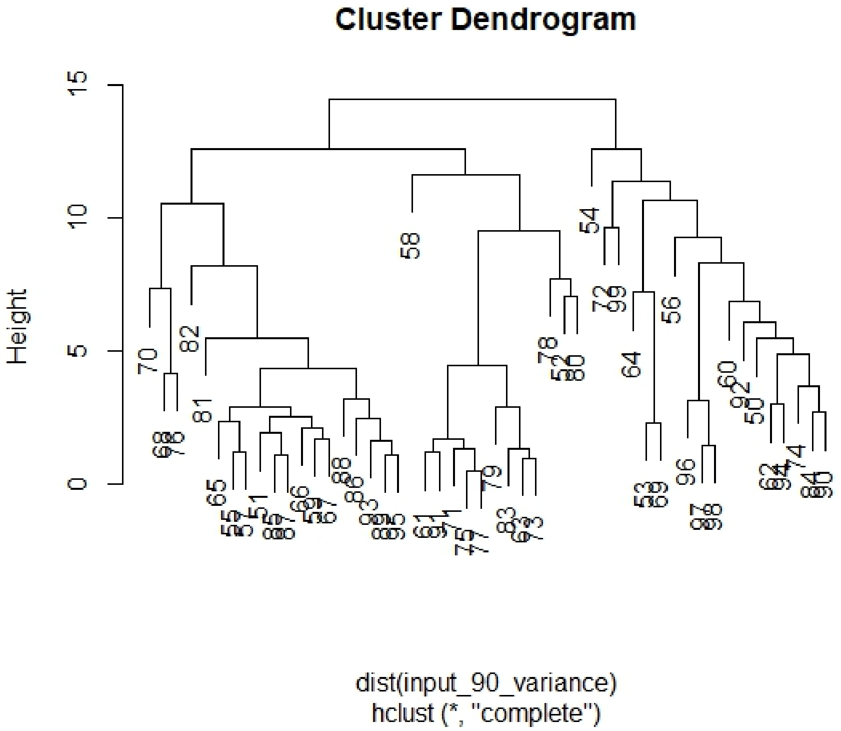


5.2 Cut the dendrogram at a height that results in two distinct clusters. Calculate an error rate.

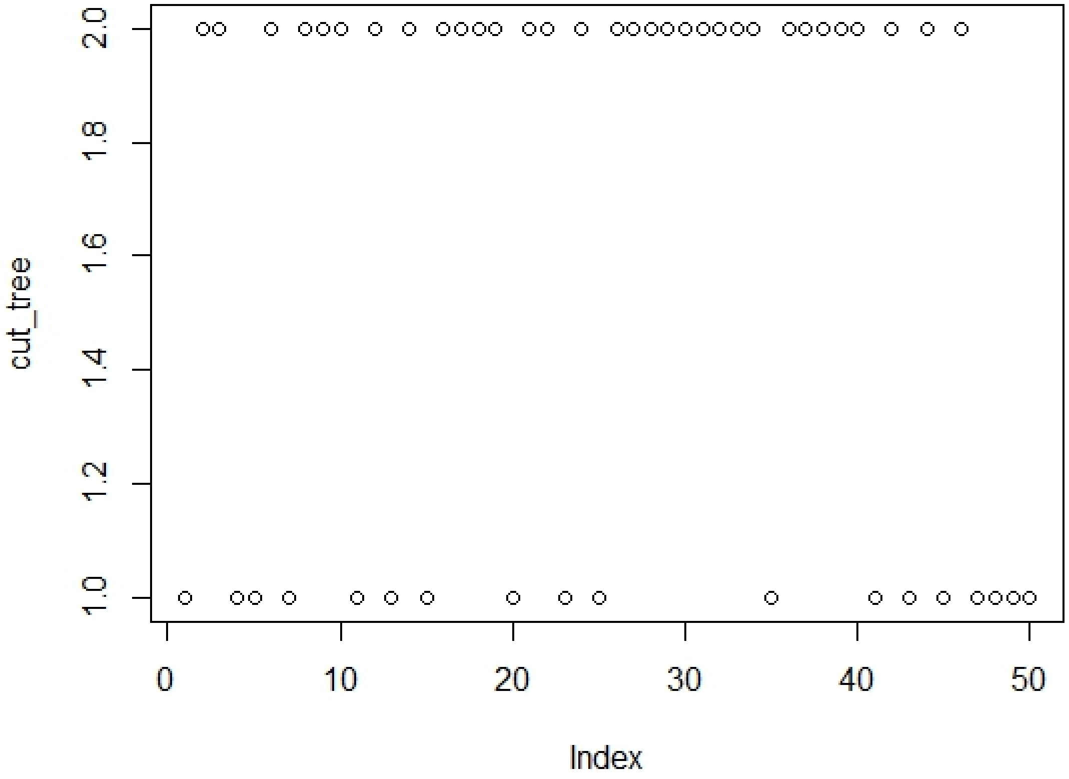


The error rate for each cluster is done in the following way:
$(Number - of - bad - points)/(Total - points - in - the - cluster)$
This is calculated for both clusters and the two values are summed up.
The error before doing PCA is - 1.12152**.**

5.3 First, perform PCA on I_{50} (Keep 90% of variance). Then hierarchically cluster the reduced data using complete linkage and Euclidean distance. Plot the dendrogram



5.4 Cut the dendrogram at a height that results in two distinct clusters. Calculate an error rate. How did PCA affect hierarchical clustering?



The error rate for each cluster is done in the following way:
$(Number - of - bad - points)/(Total - points - in - the - cluster)$
This is calculated for both clusters and the two values are summed up.
The error after doing PCA is - 1.12152**.**

From the errors we see that there is no difference in it. In this case, the PCA did not affect the error rate of the dendogram. This is because we considered the components which accounted for $90 percent$ variance. Since $90 percent$ variance is high, this is as good as most part of the entire data.

Extra credit [60 points]

This part is optional.

1 Improve the EM algorithm through initialization. k-means ++ is an extended k-means clustering algorithm and induces non-uniform distributions over the data that serve as the initial centroids. Read the paper and implement this idea to improve your G_k program (from question 3.1). Run your new G_k and old one (question 3.1) for $k = 2, \ldots, 5$ and compare the results using whisker plots. [30 points]

2 Run the EM algorithm for different mixture models, i.e., Poisson, and against different data sets. [30 points]