**Data Conversion:**

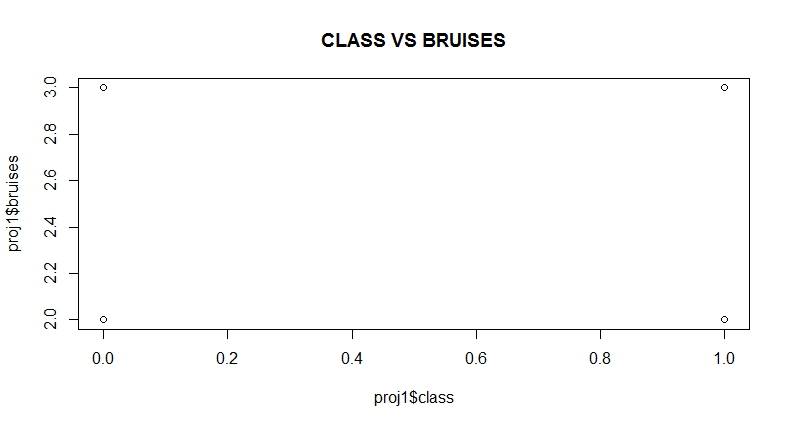
We did data conversion from character to numerical. It is required to plot.



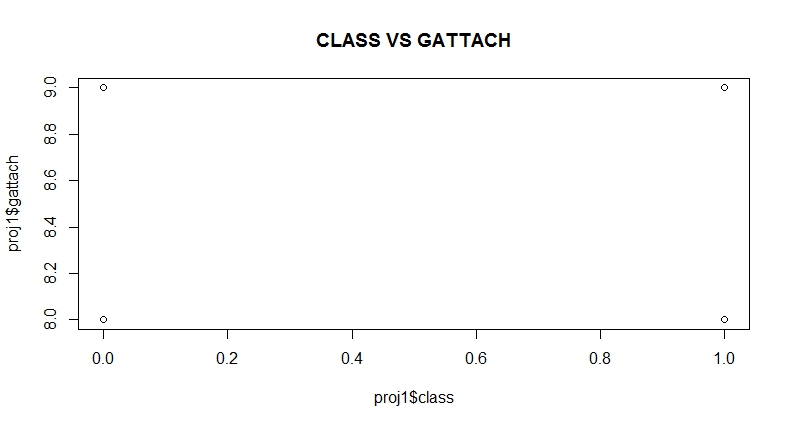
We ran the above commands to plot class column with all the other columns in order to judge which attributes help us identify for sure if the sample is poisonous or edible. One thing which is clear from the above screenshot is that NAs were generated for odor and stalk-root (sroot) columns, therefore, we cannot use those attributes while judging if the sample is poisonous, edible, or unsure.

Now, we concentrate on the other plots. From all the plots there were some that gave the exact same results, i.e., for these attributes, the mushrooms contained the same values. They are as follows:

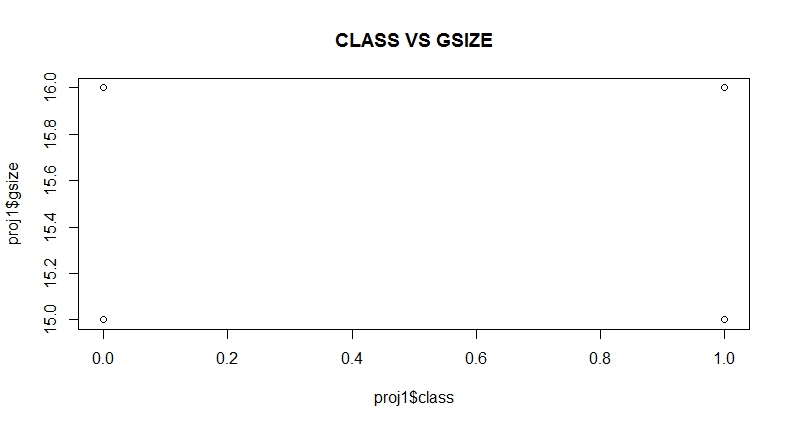
**Data Visualization :**



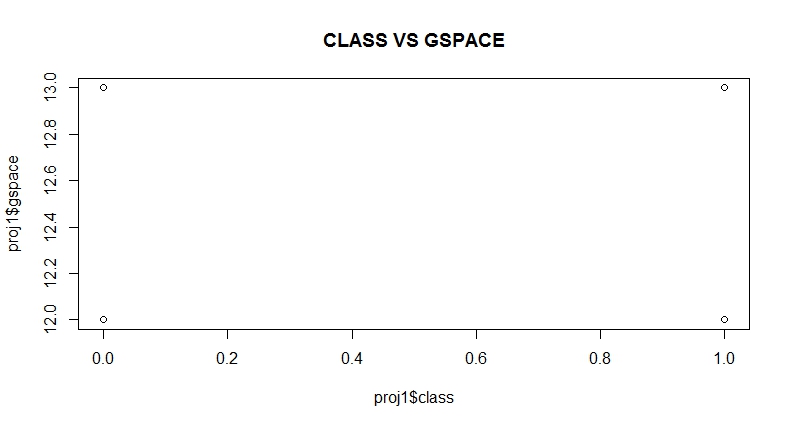
Now in the above plot, it is clear that both poisonous and edible have the same values for bruises. The value 2.0 tells us that the mushroom has bruises, and 3.0 tells us that the mushroom does not have bruises. This means that some edible mushrooms may have bruises and some may not have bruises, and same for the poisonous mushrooms. Therefore, this plot does not help us at all.



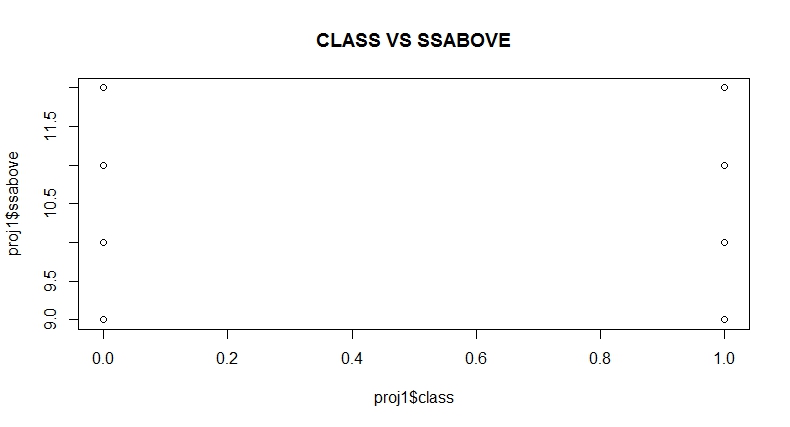
For the above plot of class vs gill-attach, it is again clear that the edible and poisonous mushrooms both may be free (8.0) or attached (9.0) to the gill. Hence, this plot also does not convey any important information.



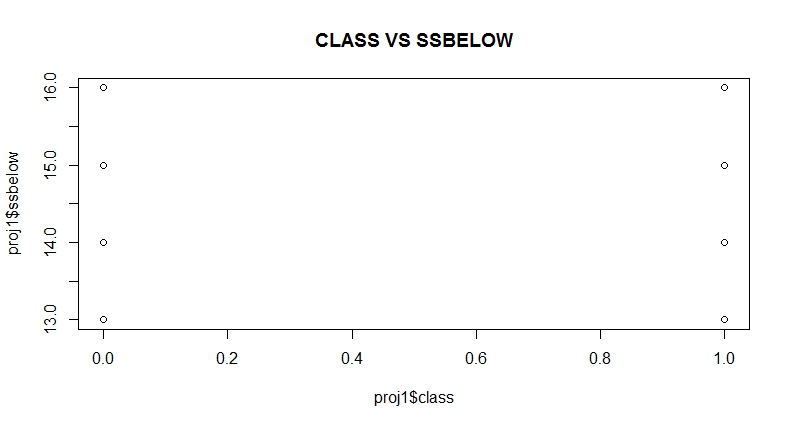
For the above plot of class vs gill-size, it is clear that the edible and poisonous mushrooms both may be broad (15.0) or narrow (16.0). Hence, this plot also does not convey any important information.



For the above plot of class vs gill-space, it is clear that the edible and poisonous mushrooms both may have a close (12.0) or crowded (13.0) gill-space. Hence, this plot also does not convey any important information.



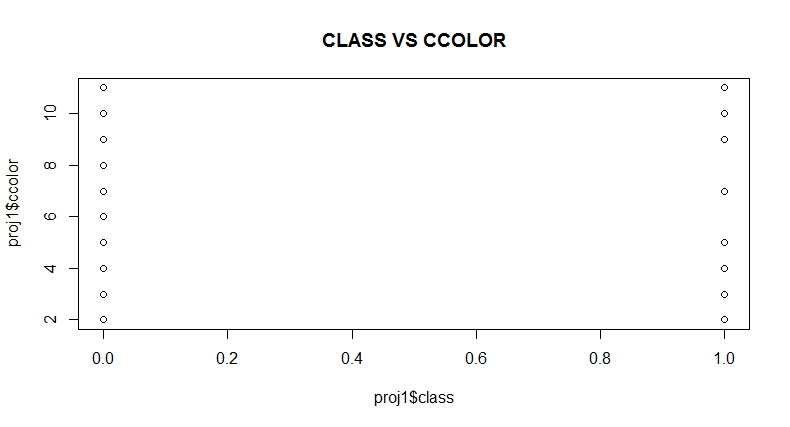
For the above plot of class vs stalk-surface-above-ring, it is clear that the edible and poisonous mushrooms both may have a fibrous (9.0), scaly (10.0), silky (11.0), or smooth (12.0) stalk surface above the mushroom ring. Hence, this plot also does not convey any important information. It is the exact same for stalk-surface-below-ring and plot is as follows:



Similar is the case with stalk shape and veil color.

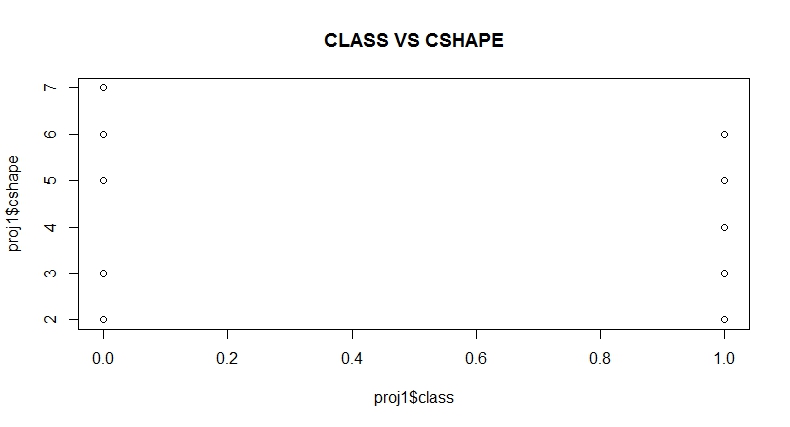
Therefore, there are 12 attributes that have different values for edible and poisonous, and help us determine for certain if the mushroom is edible or poisonous, and if the value is same for a certain value it means that those are unsure samples. The attributes with their plots is as follows:

1. Class vs Cap-color



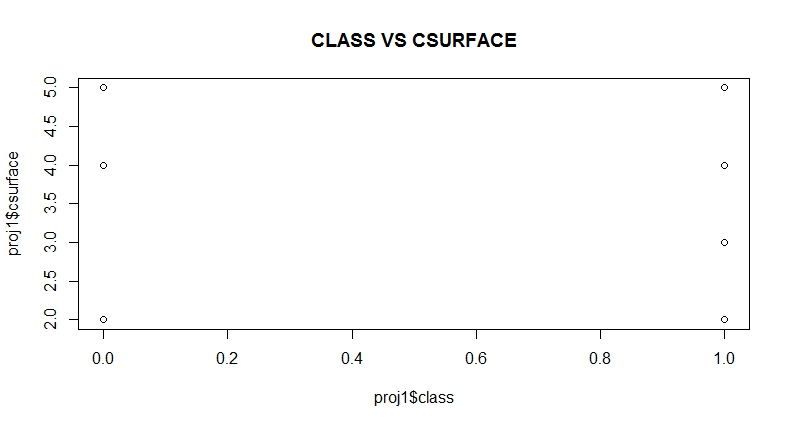
From the above plot we can say that if the color of the mushroom is either green (6) or purple (8) then the mushroom is surely edible. This rule can help us understand that only if the color of mushroom is green or purple then we can eat it, else it may be poisonous or unsure.

1. Class vs Cap-shape



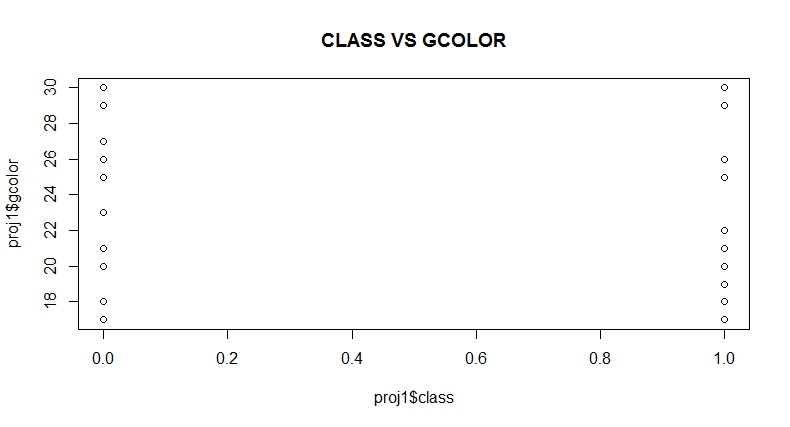
From the above plot we can say that if the shape of the mushroom is conical (4) then the mushroom is surely poisonous, and if the mushroom shape is sunken (7) then the mushroom is surely edible.

1. Class vs Cap-surface



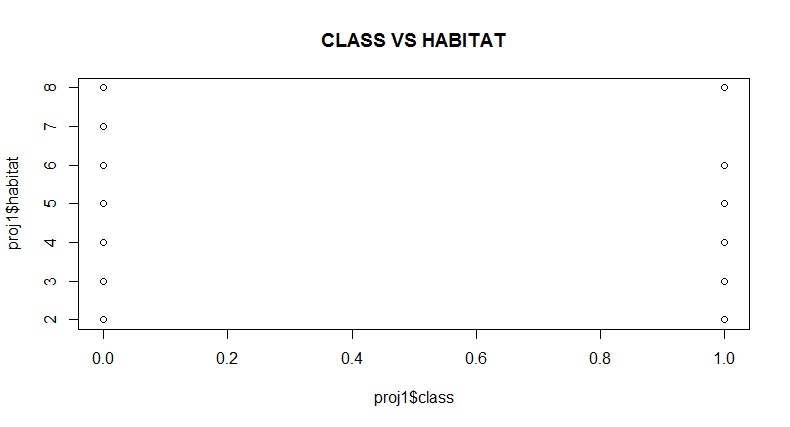
From the above plot we can that if the surface of the cap has grooves then the mushroom is surely poisonous.

1. Class vs gill-color



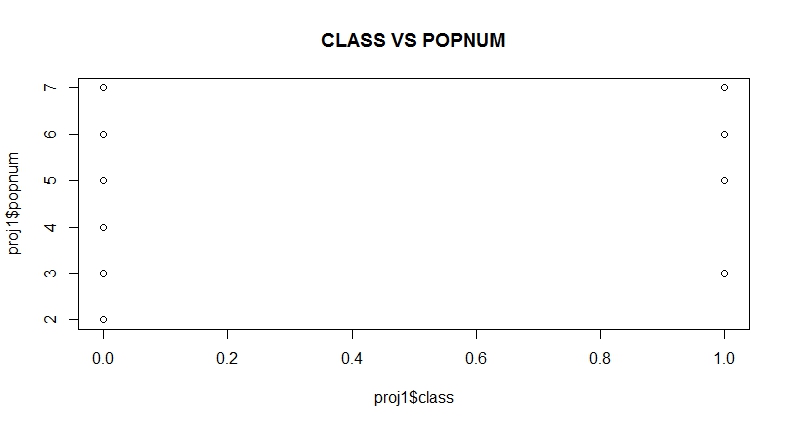
From the above plot we can say that if the gill-color of the mushroom is buff (19) or green (22) then the mushroom is poisonous, and if it is orange (23) or pink (25) then it is edible. The other colors represent the unsure class.

1. Class vs Habitat



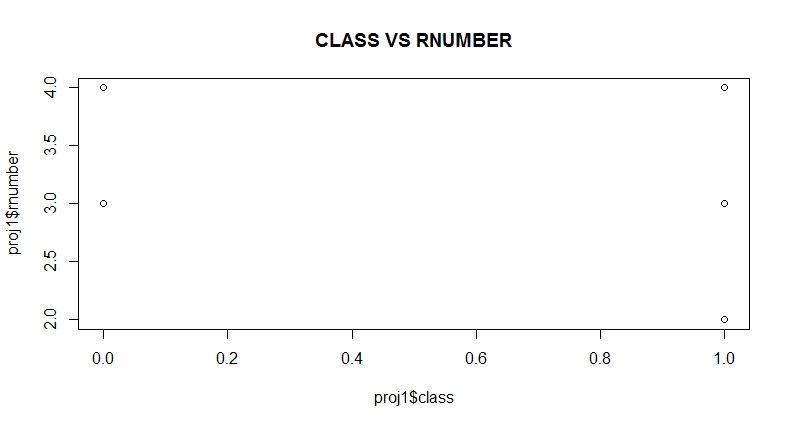
Form the above plot we can say that if the mushroom lives in paths (7) then it will always be poisonous.

1. Class vs Population



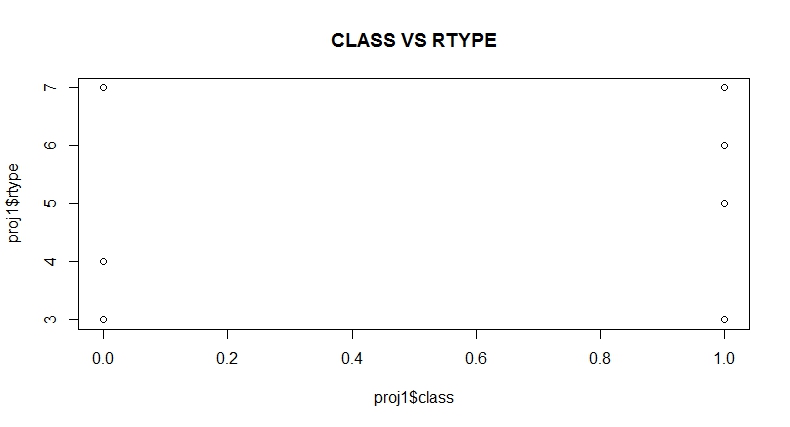
From the above plot we can say that if the population of mushroom is abundant (2) or numerous (4) then the mushroom is edible. This means that if the mushrooms are found together in abundance then these mushrooms are edible.

1. Class vs Ring-number



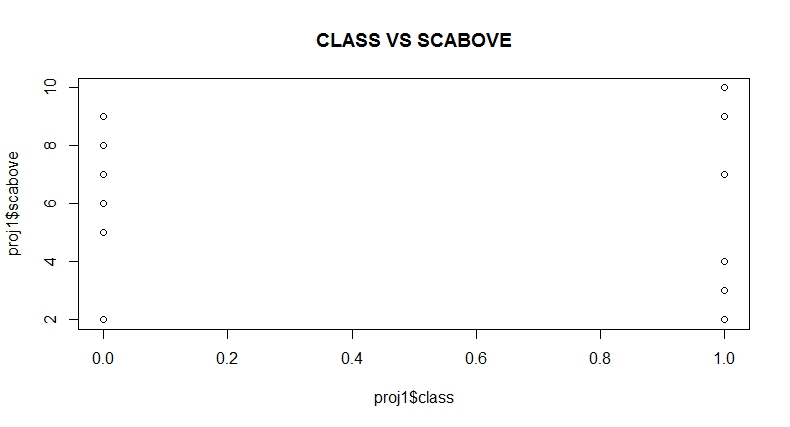
From the above plot, we can say that if the mushroom does not contain any rings (2) then it is poisonous.

1. Class vs Ring-type



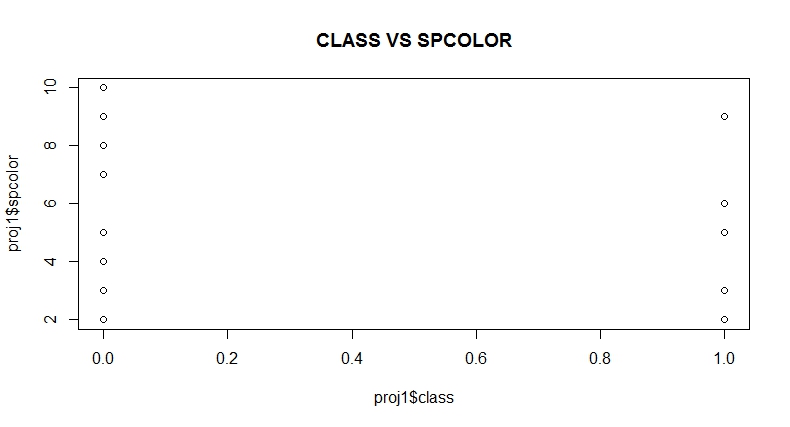
From the above plot we can say that if the mushroom has a flaring (4) ring type then it is edible. If the mushroom has a large (5) or no ring type at all (6) then it is poisonous.

1. Class vs Stalk-color-above-ring



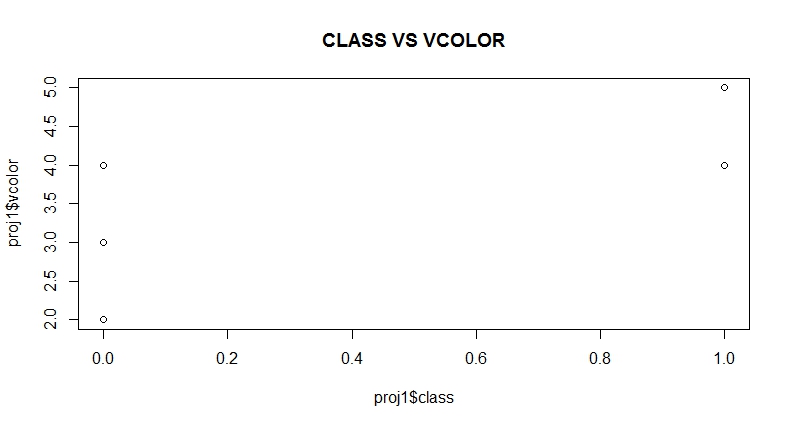
From the above plot we can say that if the color of the stalk above the ring is buff (3), cinnamon (4), or yellow (10) then the mushroom is poisonous. If the color of the stalk above the ring is gray (5), orange (6), or red (8) then the mushroom is edible. We get the exact same result for stalk-color-below-ring as well.

1. Class vs Stalk-print-color



From the above plot we can say that if the color of the stalk-print is buff (4), orange (7), purple (8), or yellow (10) then the mushroom is edible. If the color of stalk-print is green (6) then the mushroom is poisonous.

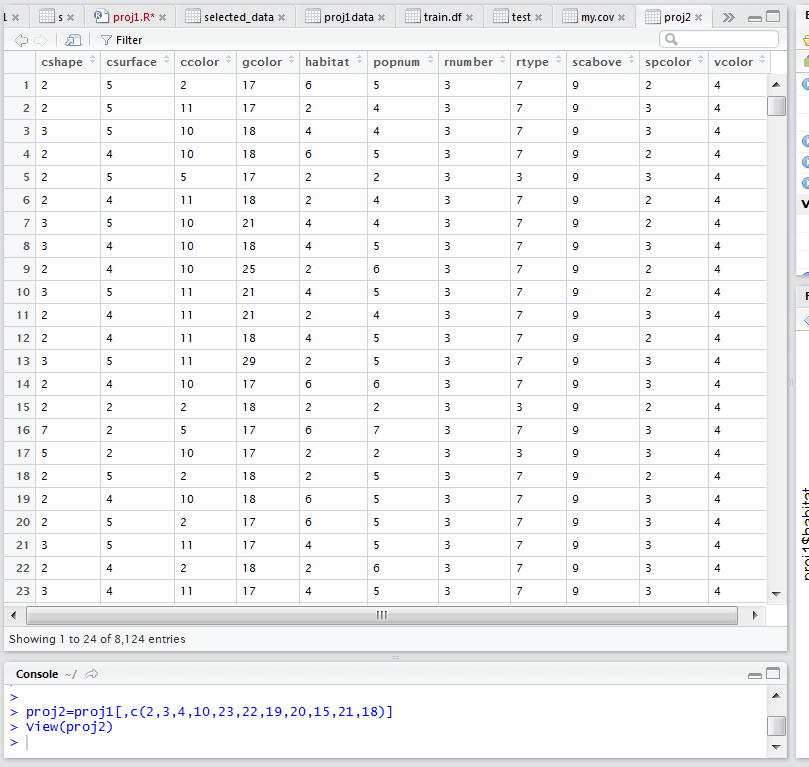
1. Class vs Veil-color



From the above plot we can say that if the color of the veil is brown (2) or orange (3) then the mushroom is edible, if it is yellow (5) then it is poisonous.

**Matrix Results:**

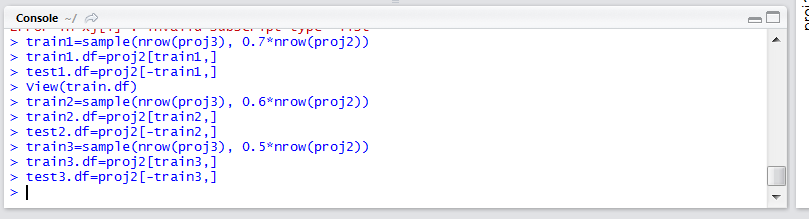
Now, we will remove these attributes from the original data frame and put it in a new data frame without the class attribute for clustering.



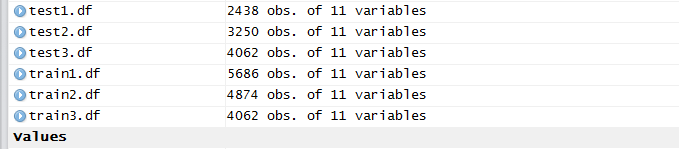
As we can see above, we have all the 11 attributes with its values and we have put it in a new data frame called proj2.

We apply k-means, k-nearest neighbor, and partitioning around medoids.

First, we divide the proj2 data frame in to two sets, training and test data sets. The division is done in three ways or three proportions: 70-30, 60-40, and 50-50.



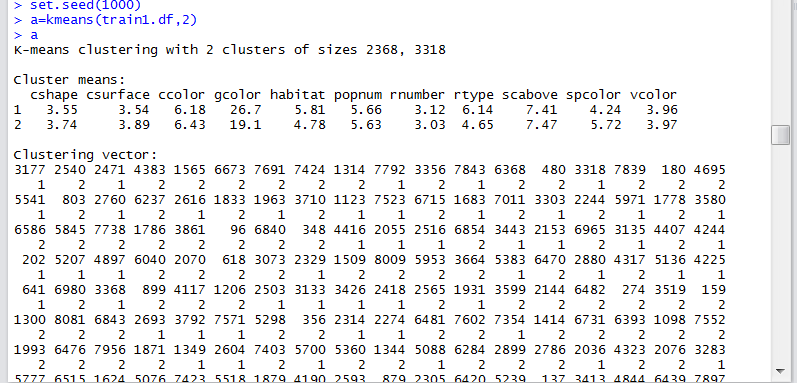
We get the training and test datasets as follows:



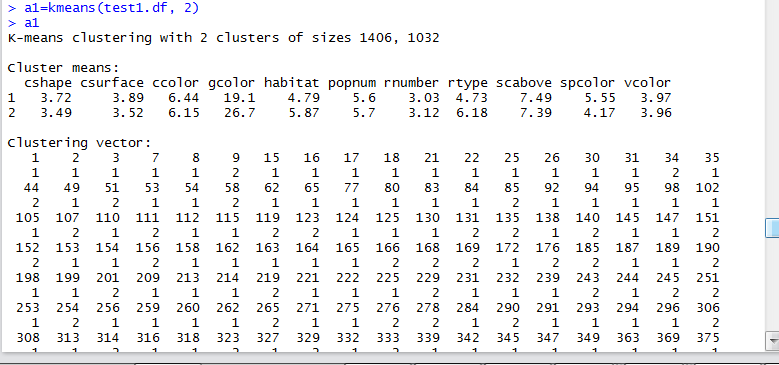
**Solution to question Number 3a)**

Now, we will perform k-means:

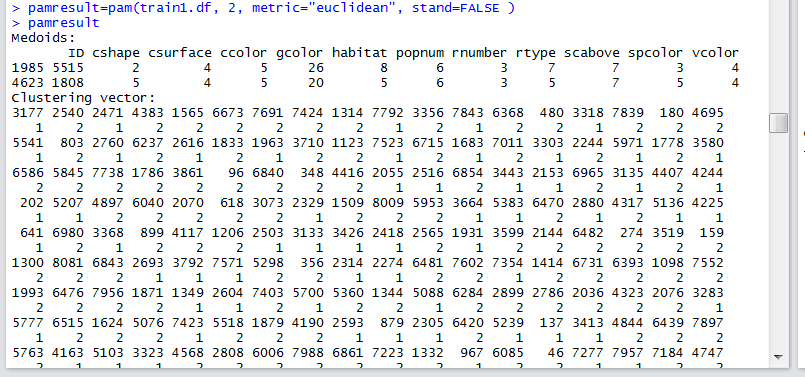
Kmeans with number of clusters (N)=2 on training data set with 70% of the values:



Kmeans with number of clusters (N)=2 on test data set with 30% of the values:

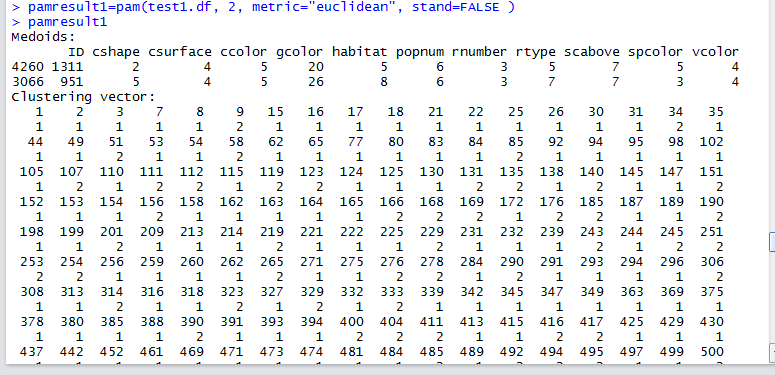


Partition around medoids with number of clusters (N)=2 on training data set with 70% of the values:

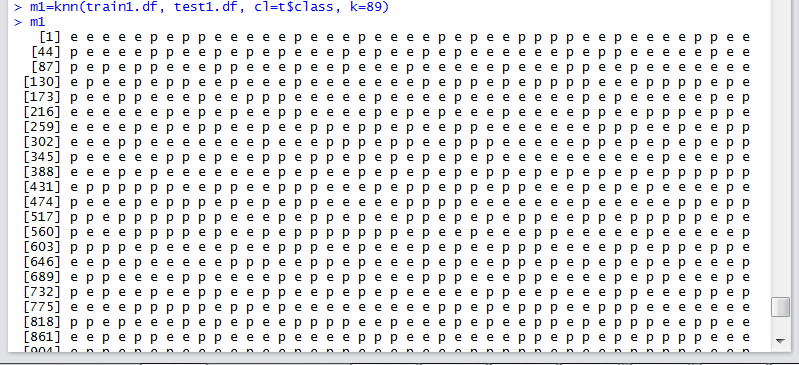


When we compare the PAM results to the Kmeans results we see that sample 3177 in both methods belongs to the same cluster 1. Also, 2540 in both methods belongs to cluster 2, and so on.

Partition around medoids with number of clusters (N)=2 on test data set with 30% of the values:

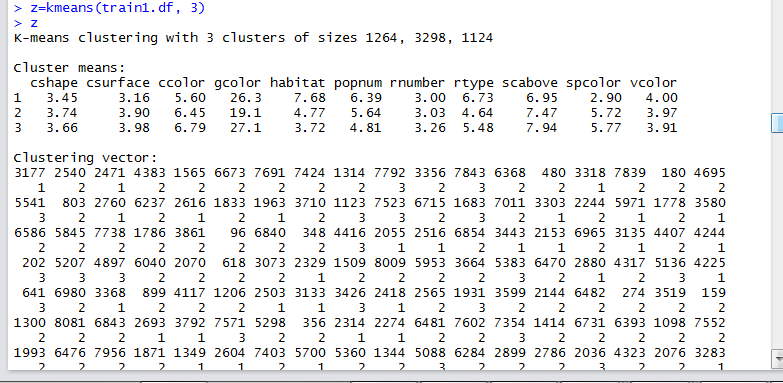


K nearest neighbor on training data set with 70% of the values and test data set with 30% of the values, with number of neighbors (k)=89:

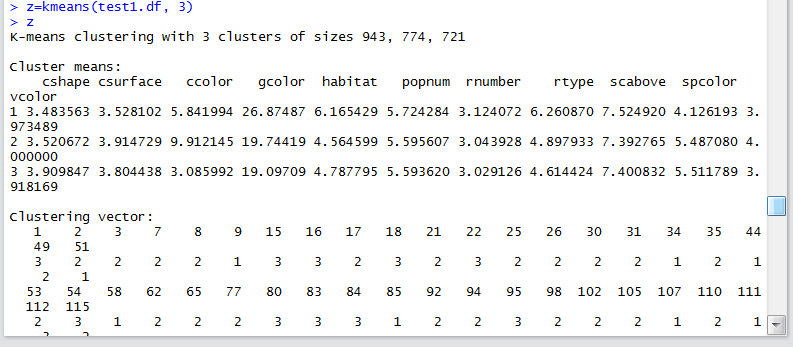


**Solution to question Number 3b)**

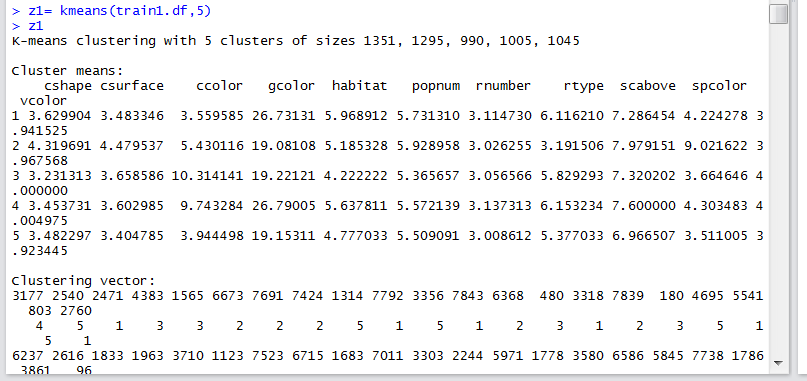
Kmeans with number of clusters (N)=3 on training data set with 70% of the values:



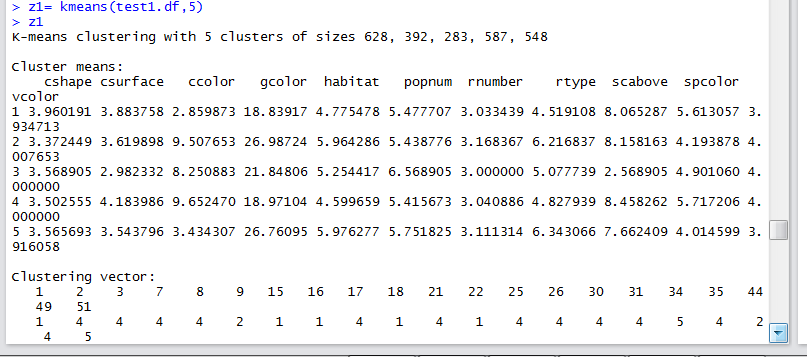
Kmeans with number of clusters (N)=3 on test data set with 30% of the values:



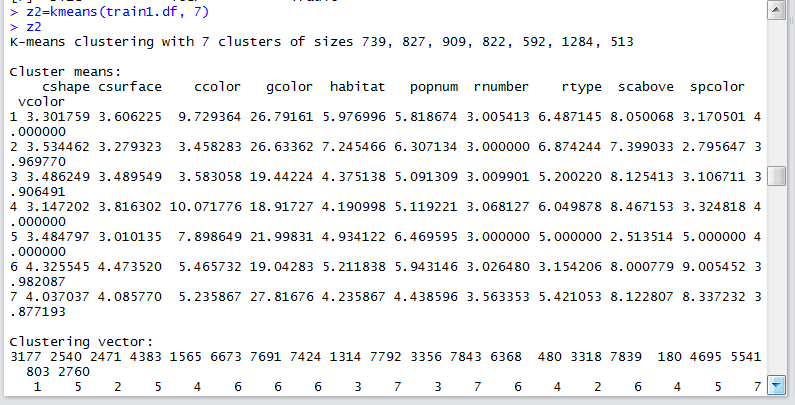
Kmeans with number of clusters (N)=5 on training data set with 70% of the values:



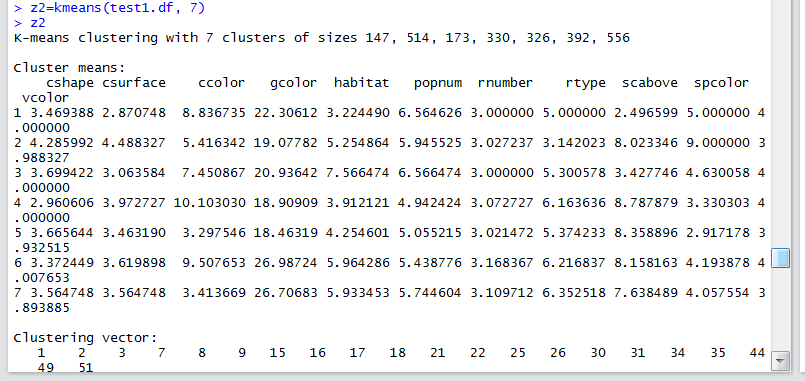
Kmeans with number of clusters (N)=5 on test data set with 30% of the values:



Kmeans with number of clusters (N)=7 on training data set with 70% of the values:

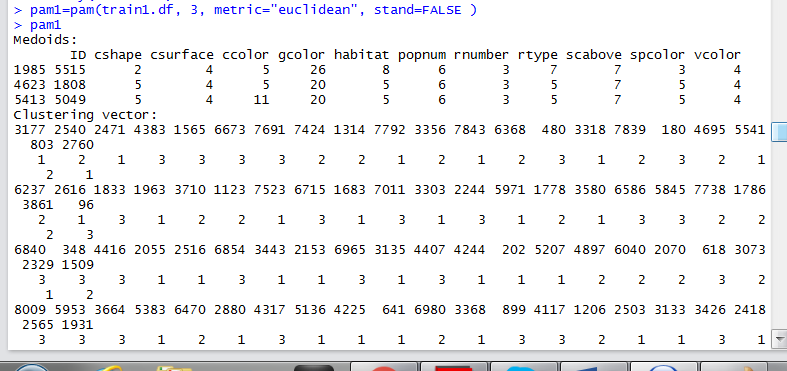


Kmeans with number of clusters (N)=7 on test data set with 30% of the values:

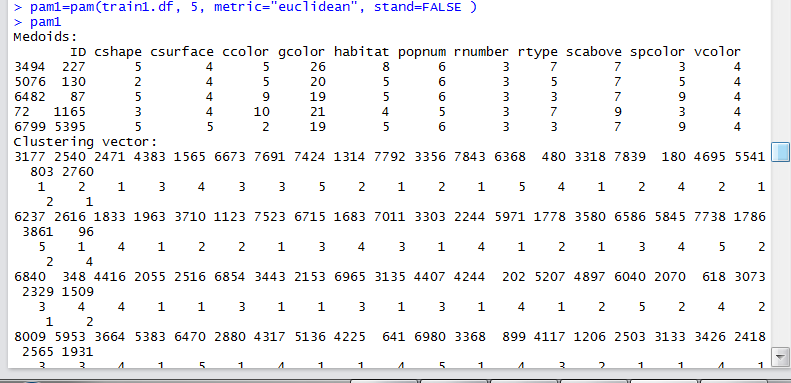


Similarly we can do the same with Partition around medoids.

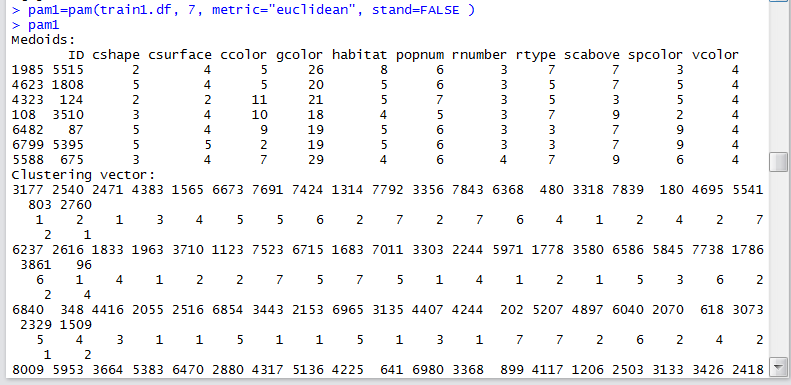
Partition around medoids with number of clusters (N)=3 on training data set with 70% of the values:



Partition around medoids with number of clusters (N)=5 on training data set with 70% of the values:

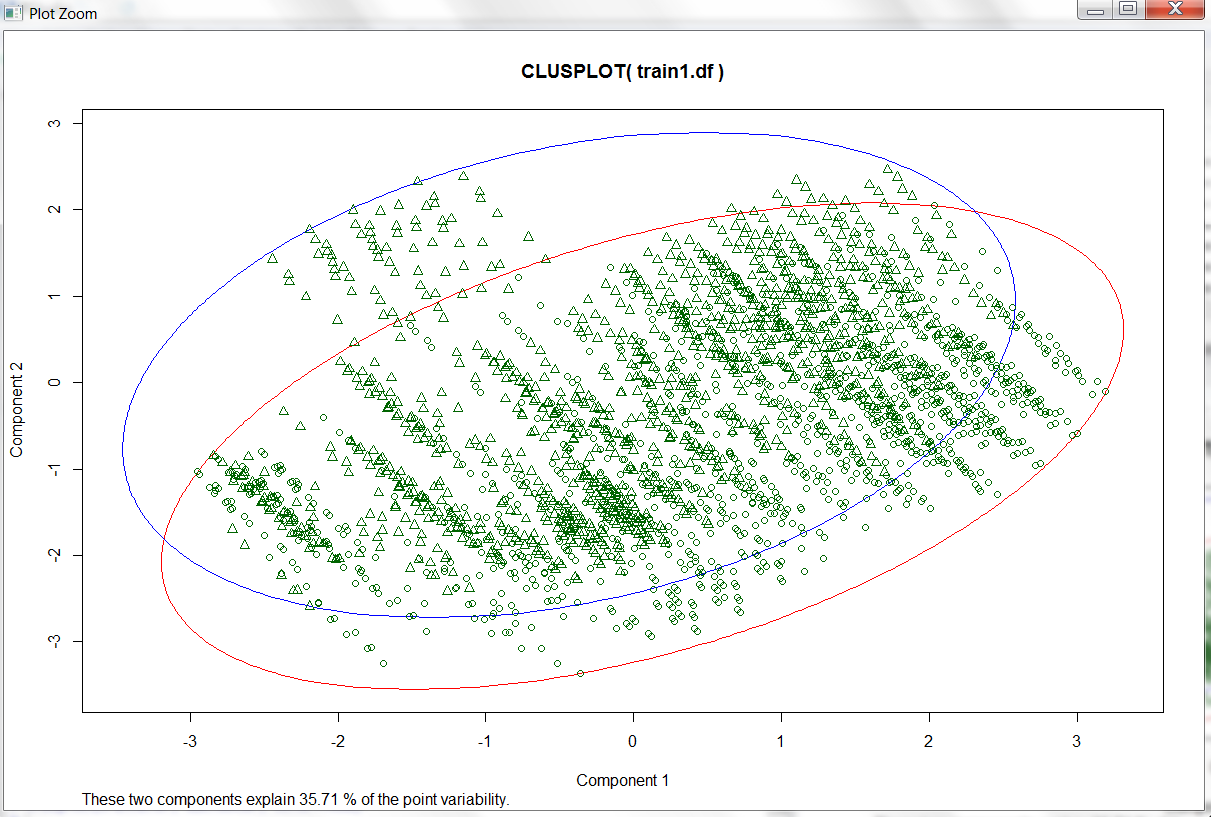


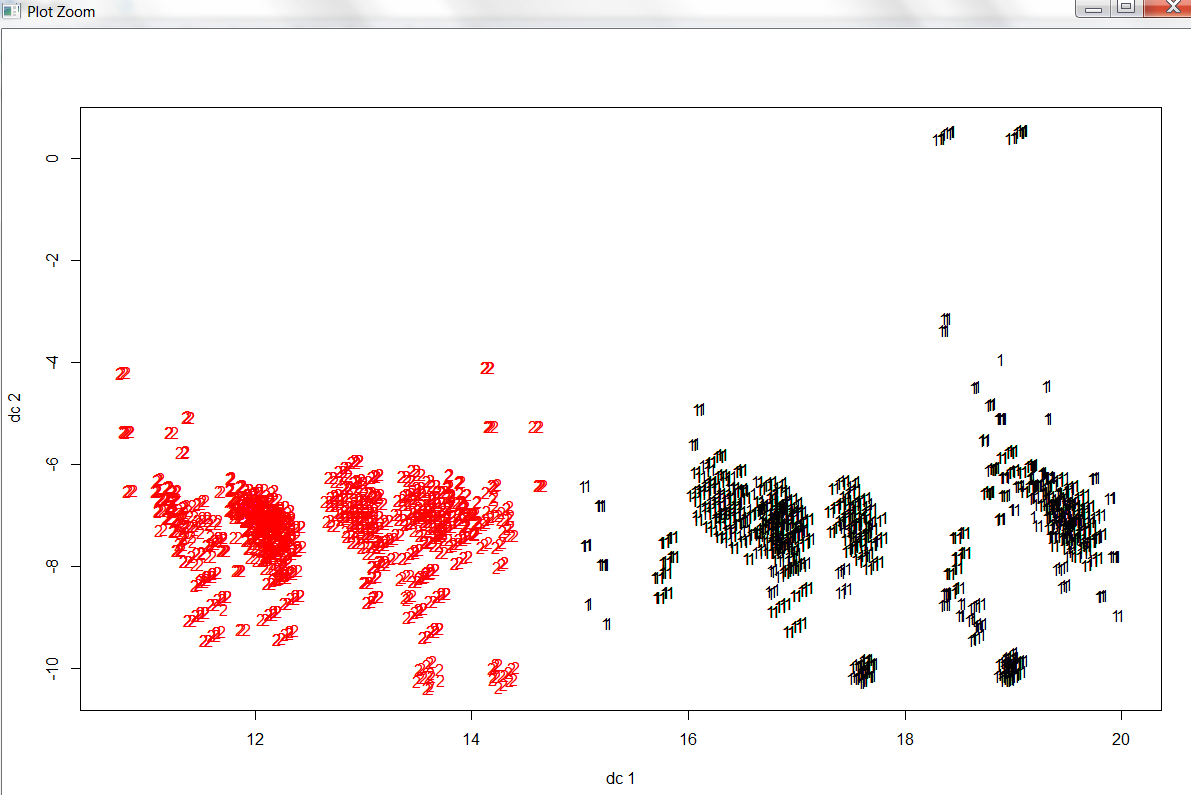
Partition around medoids with number of clusters (N)=7 on training data set with 70% of the values:



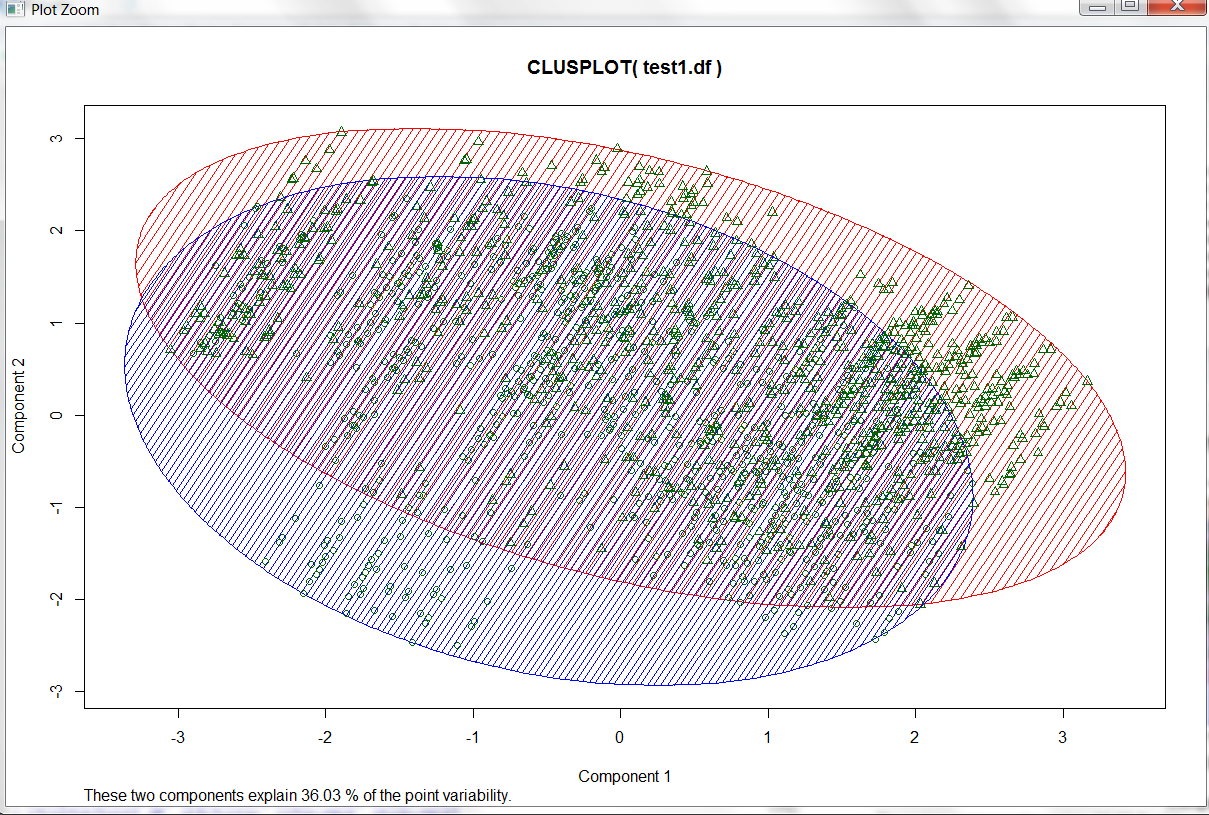
Plots are as follows:

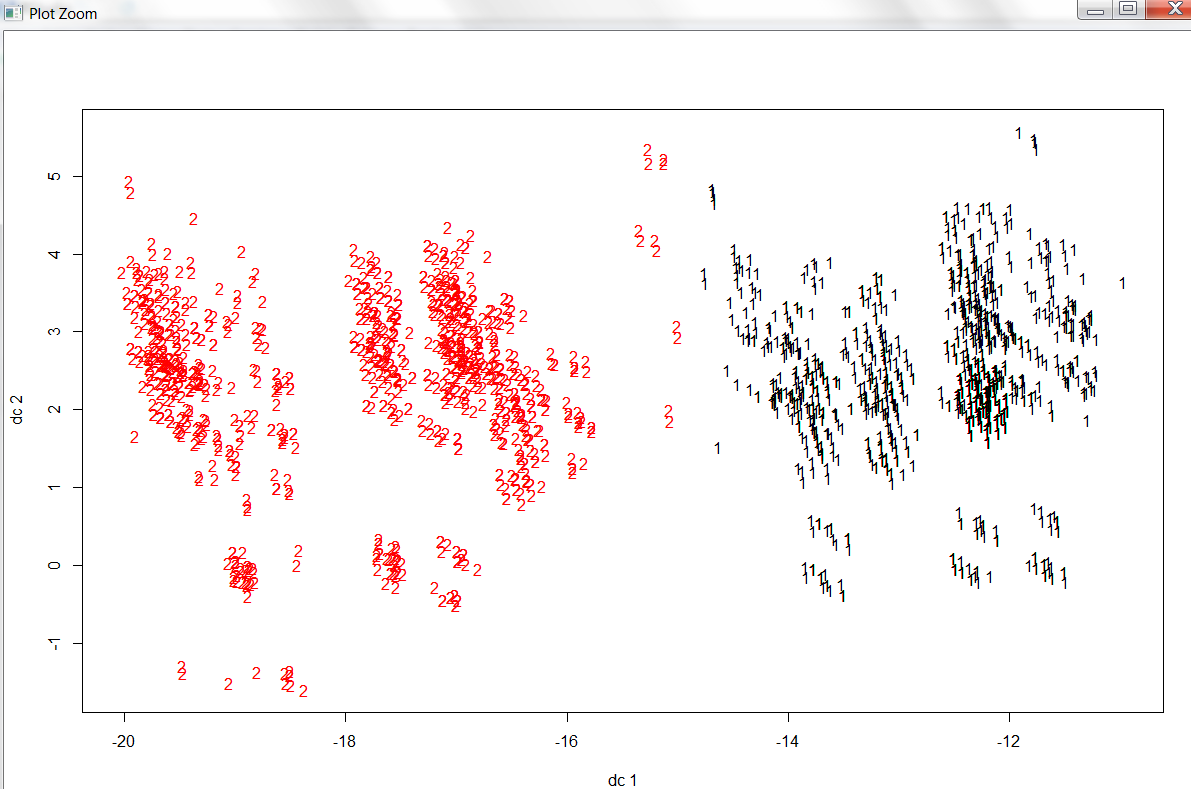
Kmeans with number of clusters (N)=2 on training data set with 70% of the values:



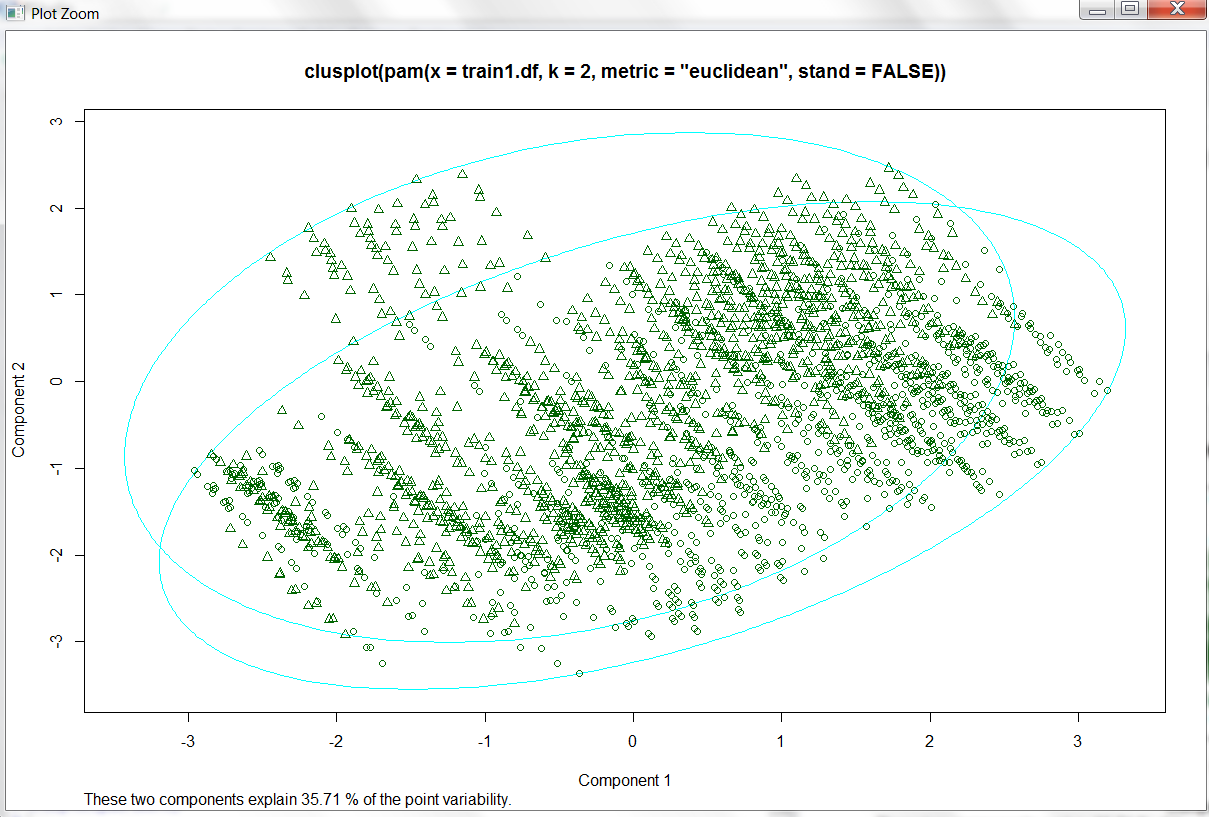


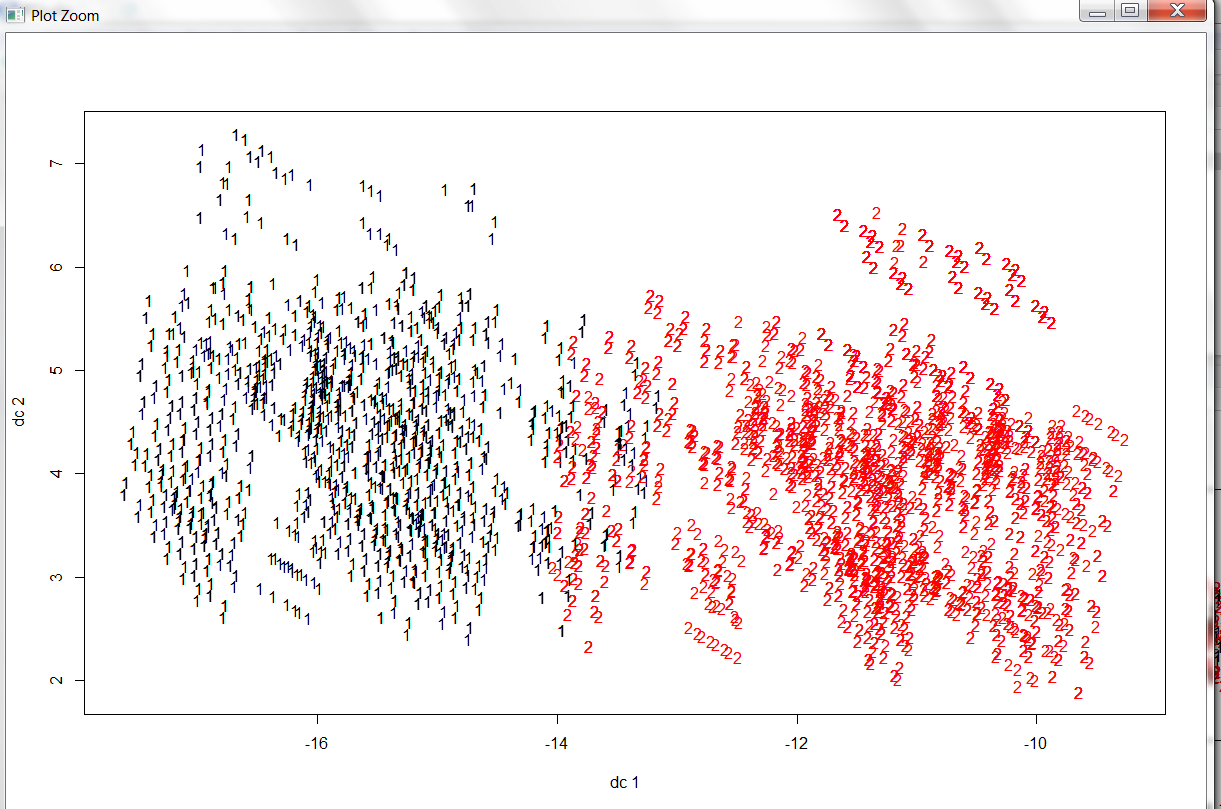
Kmeans with number of clusters (N)=2 on test data set with 30% of the values:



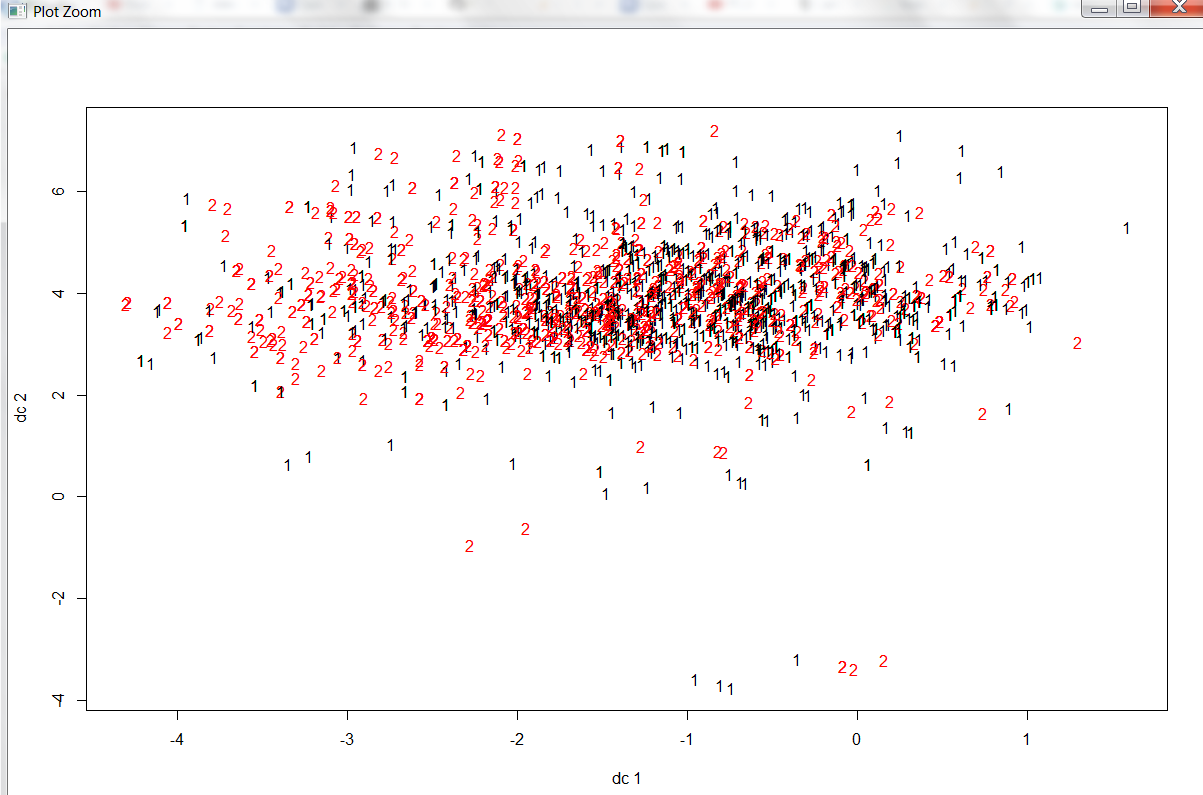


Partition around medoids with number of clusters (N)=2 on training data set with 70% of the values:

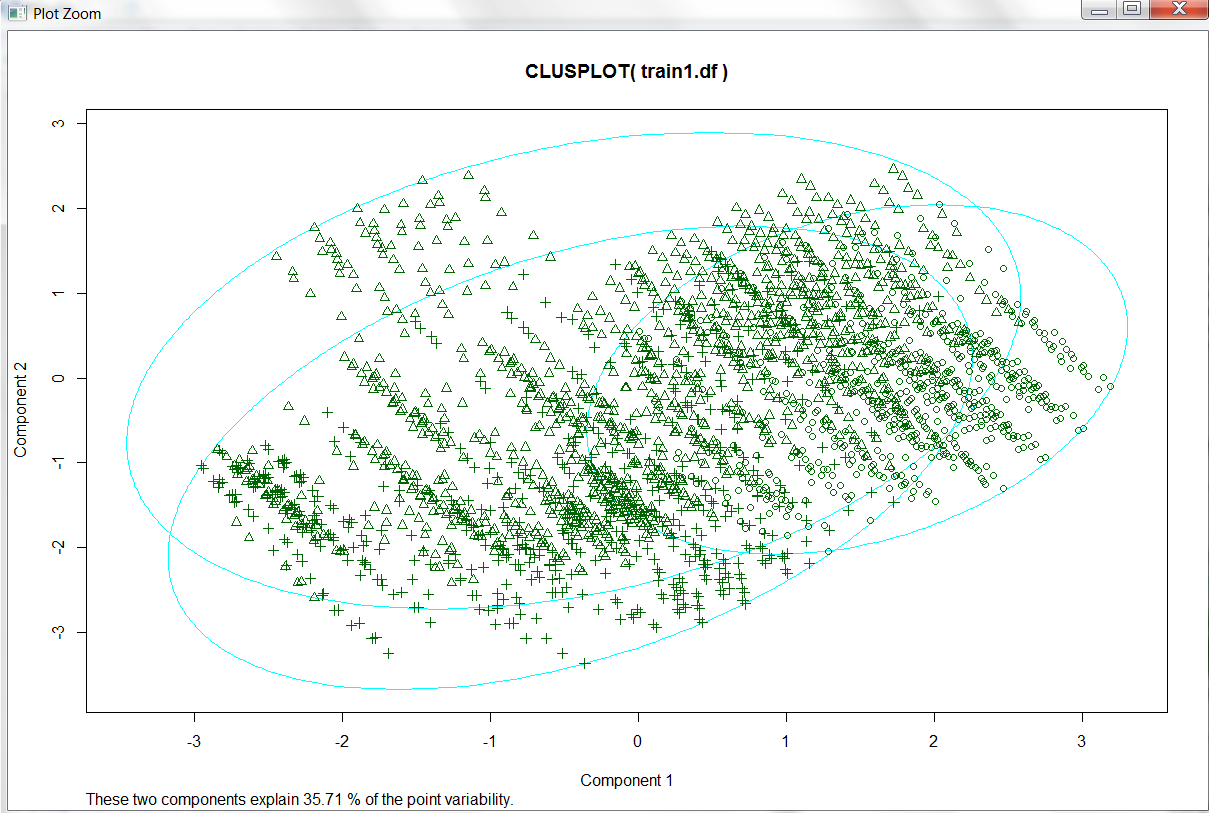
****

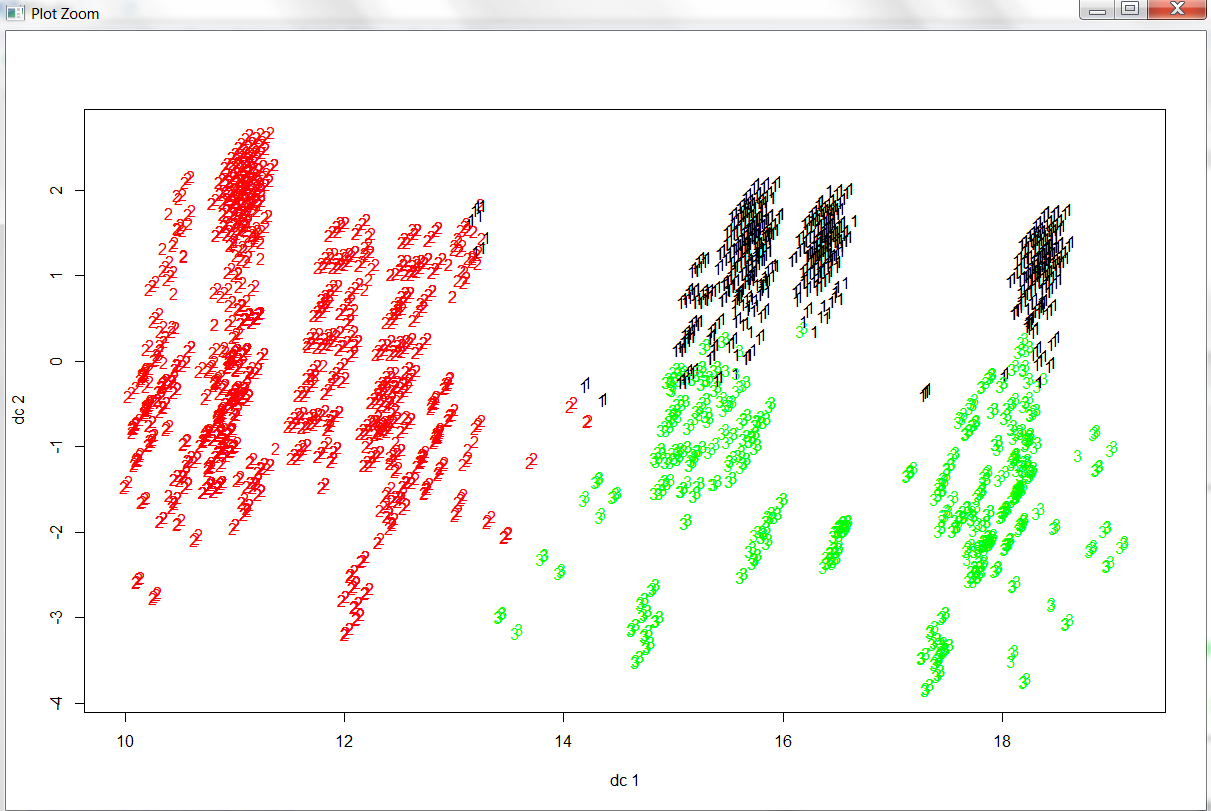
****

K nearest neighbor on training data set with 70% of the values and test data set with 30% of the values, with number of neighbors (k)=89:

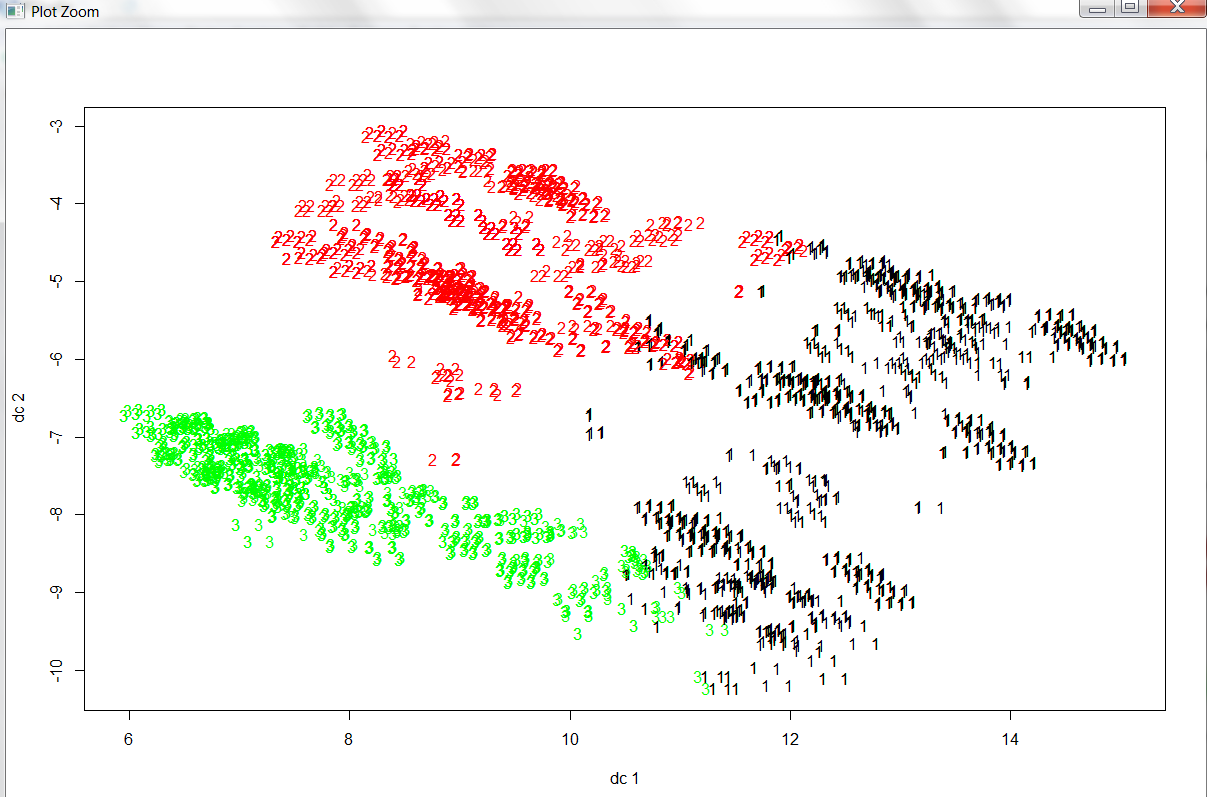


Kmeans with number of clusters (N)=3 on training data set with 70% of the values:



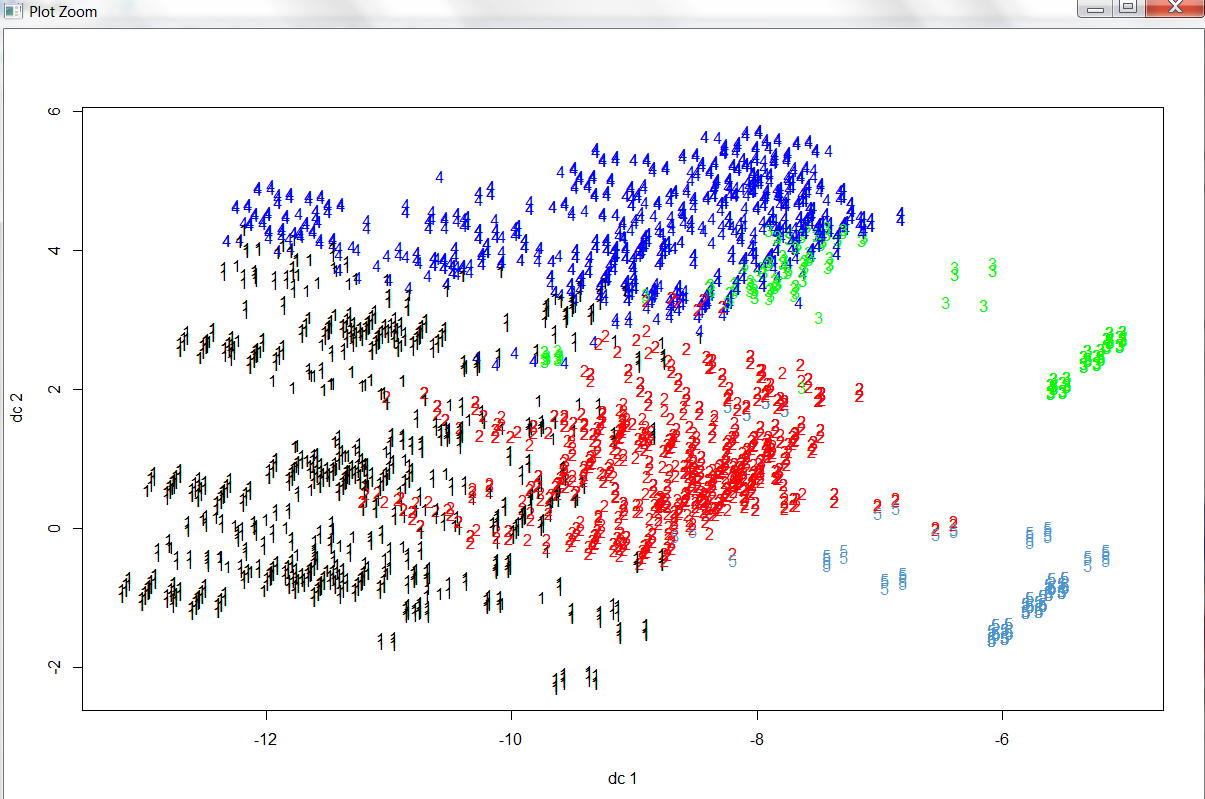


PAM with number of clusters (N)=3 on training data set with 70% of the values:



As we can see above, both the plots give similar results with minimum outliers. In K-means the intersecting data is lesser than in PAM. According to the results above, K-means gives a better result.

PAM with number of clusters (N)=5 on training data set with 70% of the values:



PAM with number of clusters (N)=7 on training data set with 70% of the values:

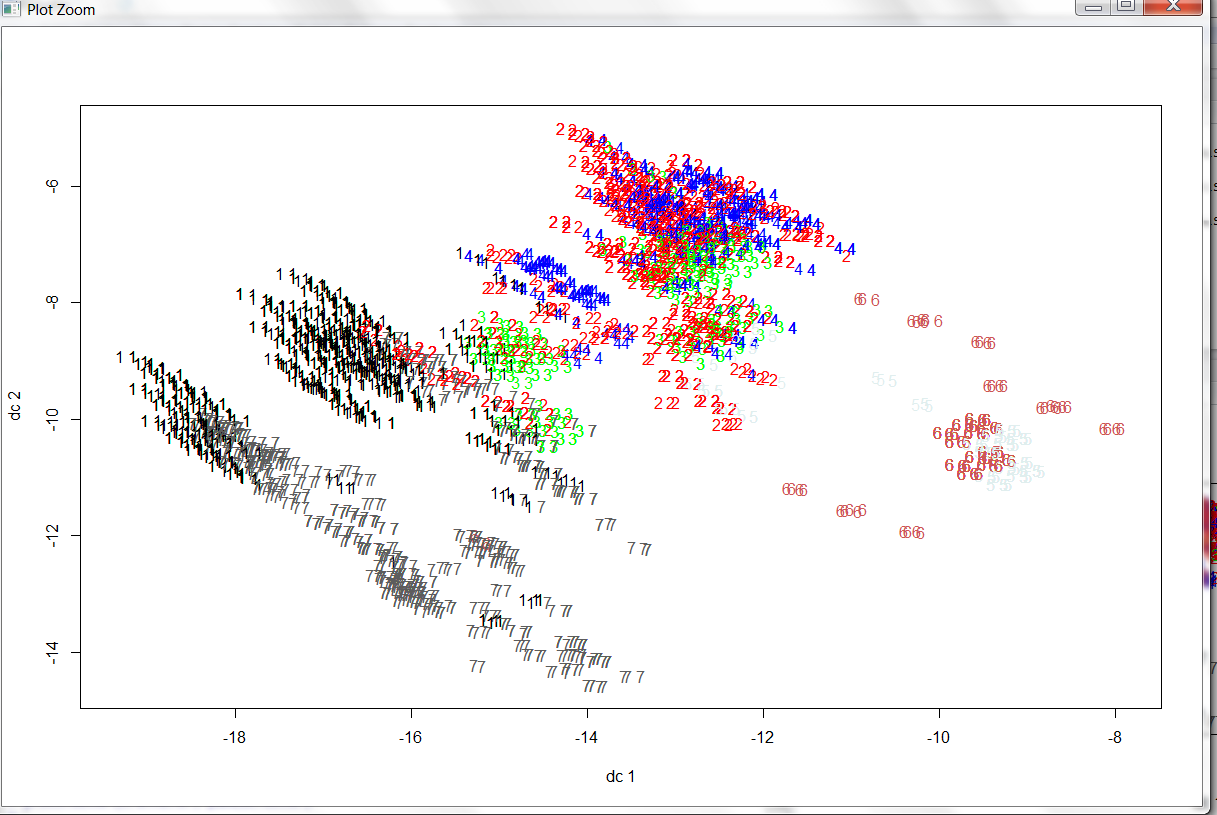
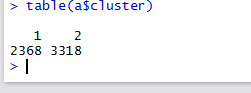
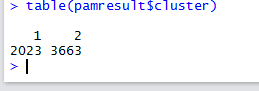


Table matrix is as follows:

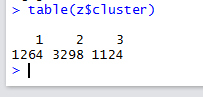
Kmeans with number of clusters (N)=2 on training data set with 70% of the values:



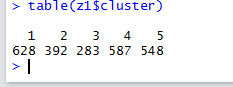
PAM with number of clusters (N)=2 on training data set with 70% of the values:



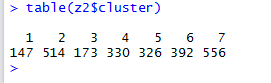
Kmeans with number of clusters (N)=3 on training data set with 70% of the values:



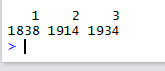
Kmeans with number of clusters (N)=5 on test data set with 30% of the values:



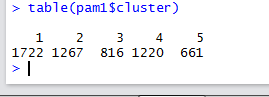
Kmeans with number of clusters (N)=7 on test data set with 30% of the values:



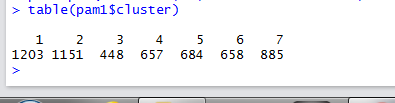
PAM with number of clusters (N)=3 on training data set with 70% of the values:



PAM with number of clusters (N)=5 on training data set with 70% of the values:



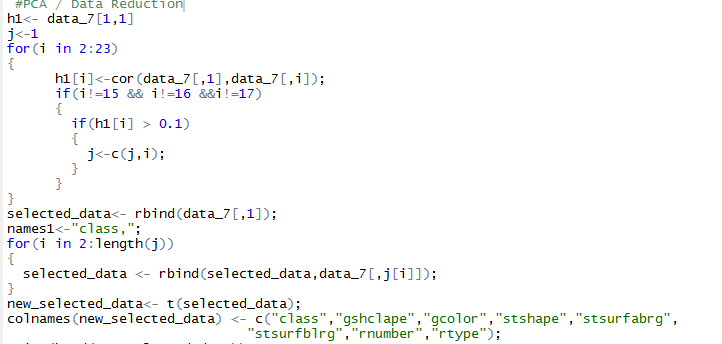
PAM with number of clusters (N)=7 on training data set with 70% of the values:



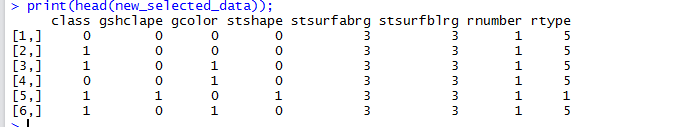
**Data Reduction / Principal Component Analysis:**

For reducing number of attributes, we found attributes that are positively correlated and which would help determining class of item.

**Program:**

****

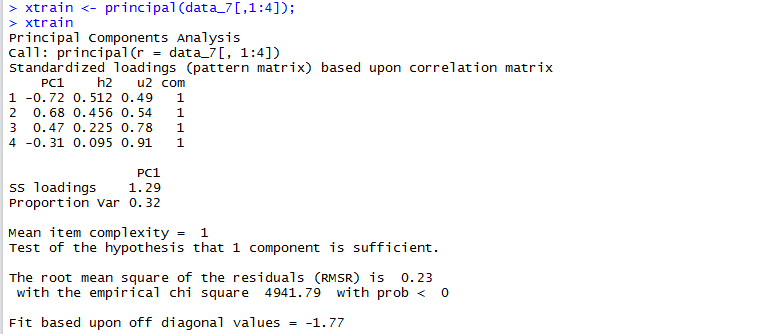
**Results:**

****

**Principal Component Analysis:**

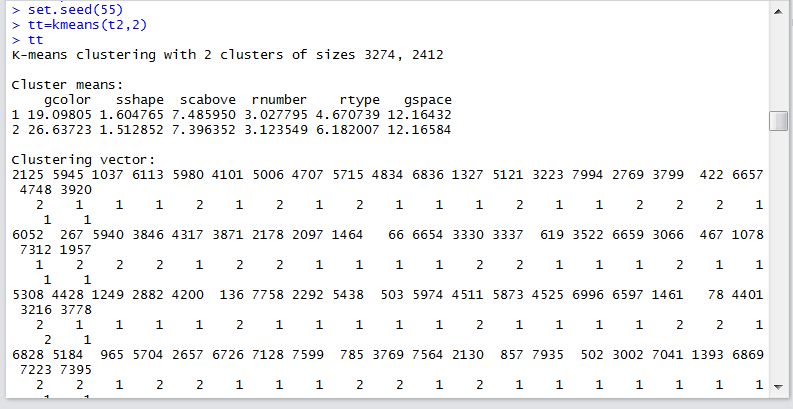
**PCA:**

Principal Component Analysis:

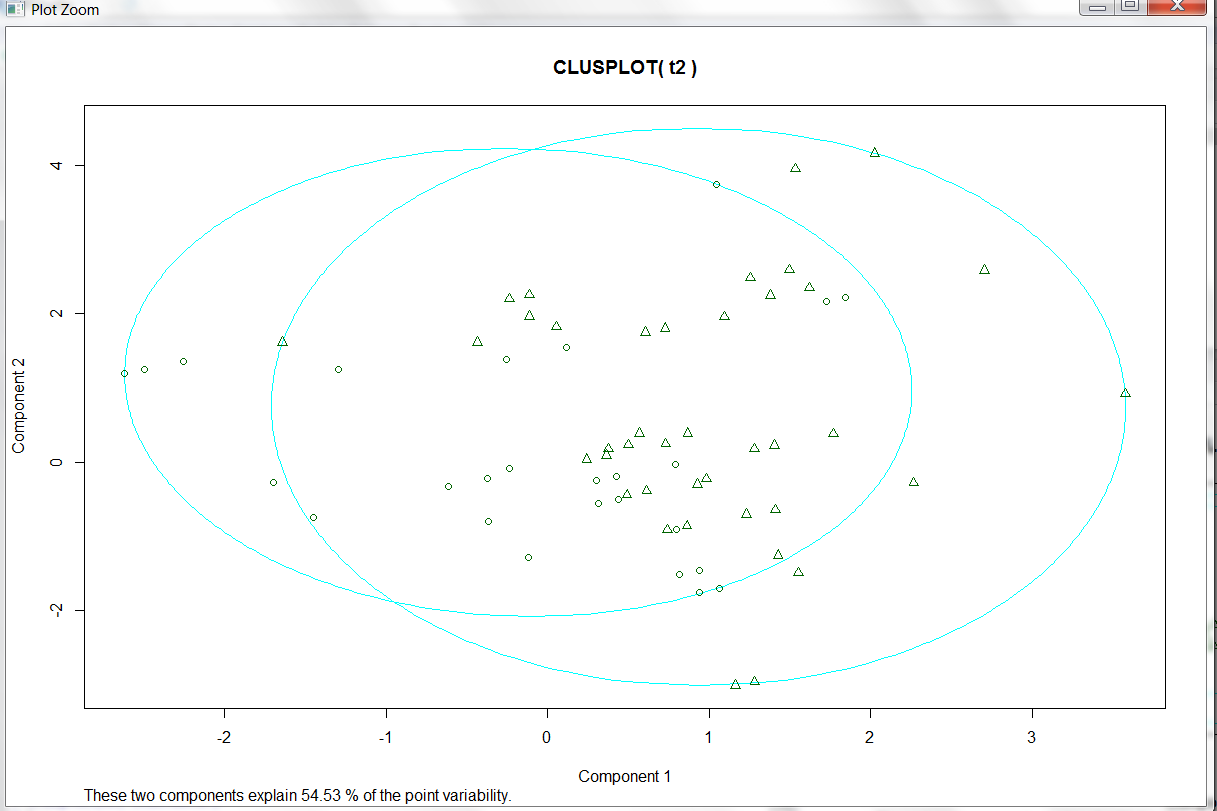


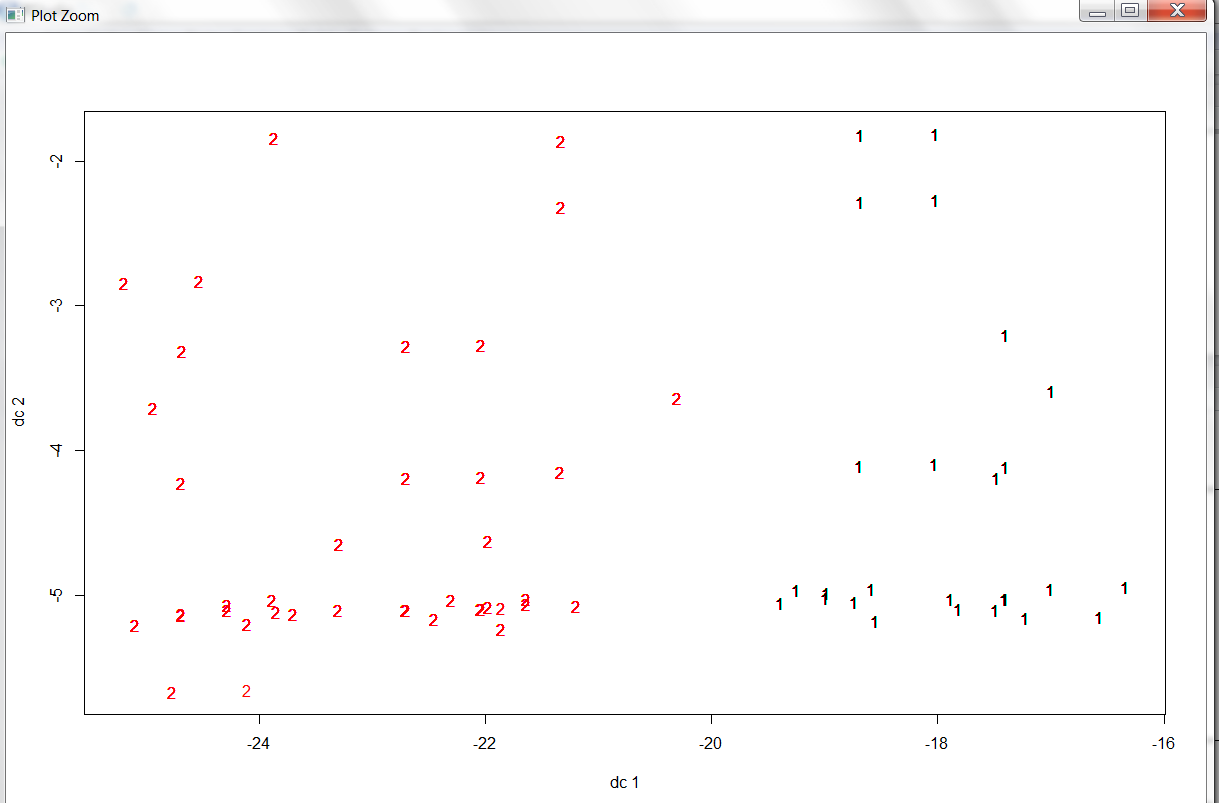
We performed Kmeans, PAM, and KNN when considering the above 7 attributes.

Kmeans with N=2 on 70% of data:

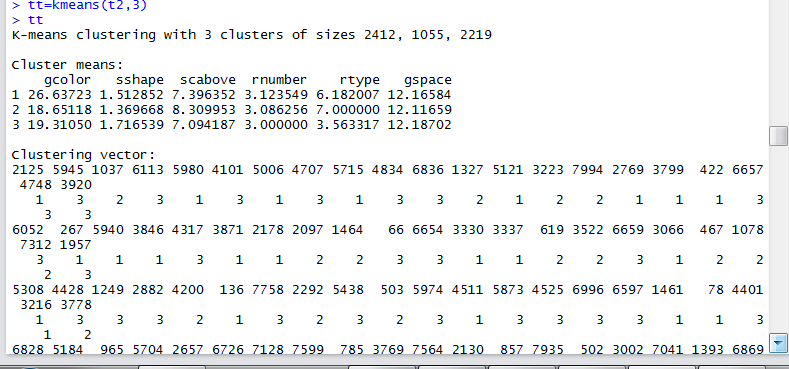


Plots are as follows:

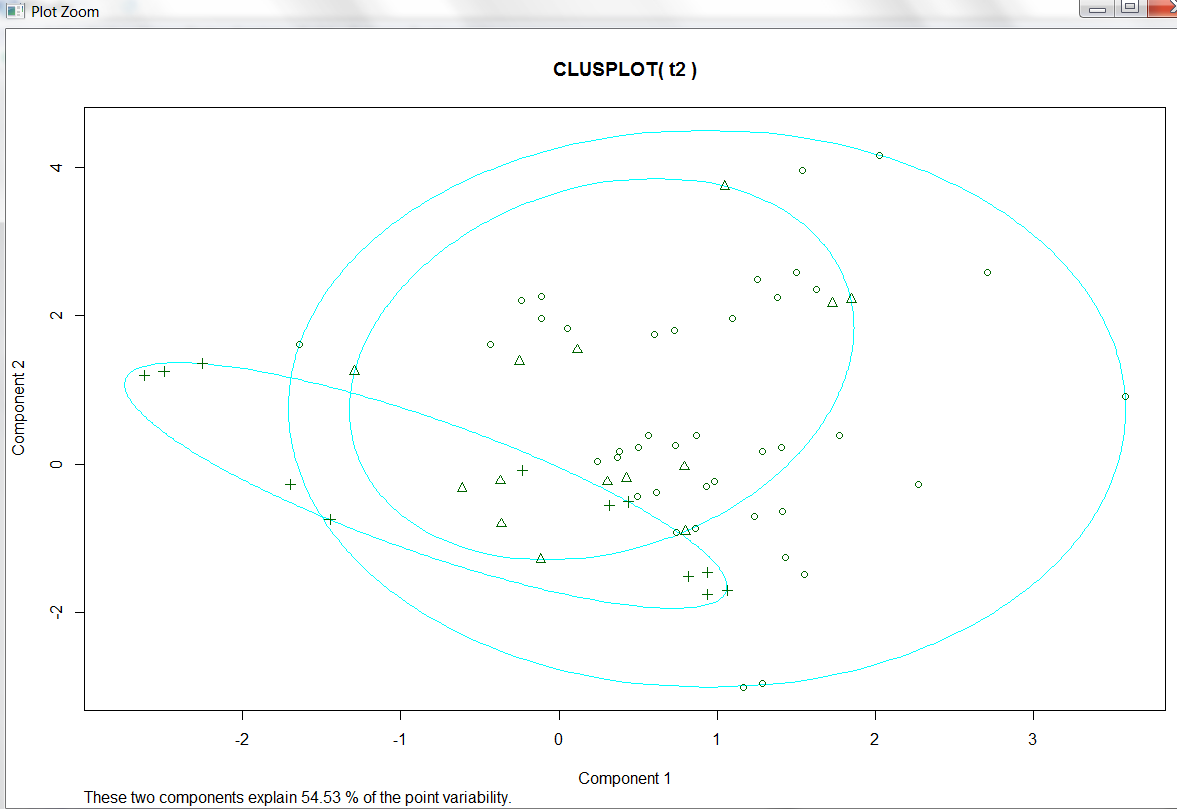


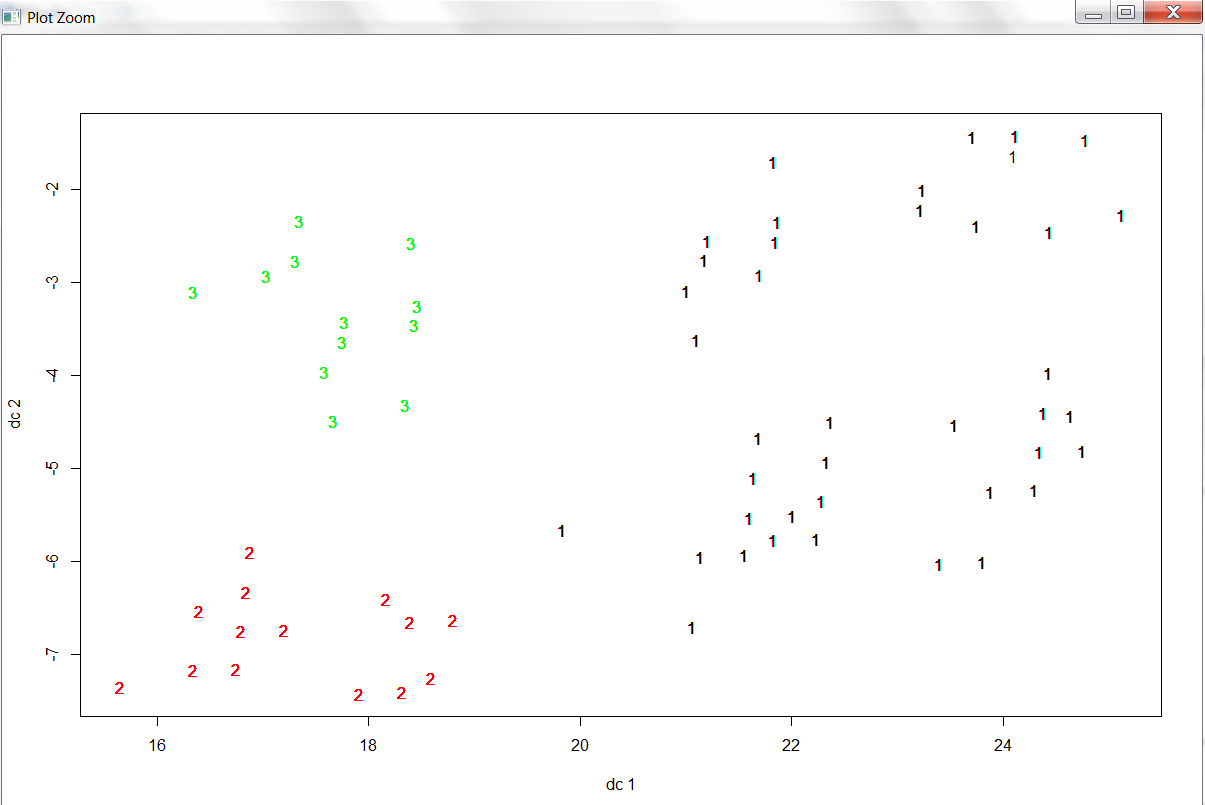


Kmeans with N=3 on 70% of data:

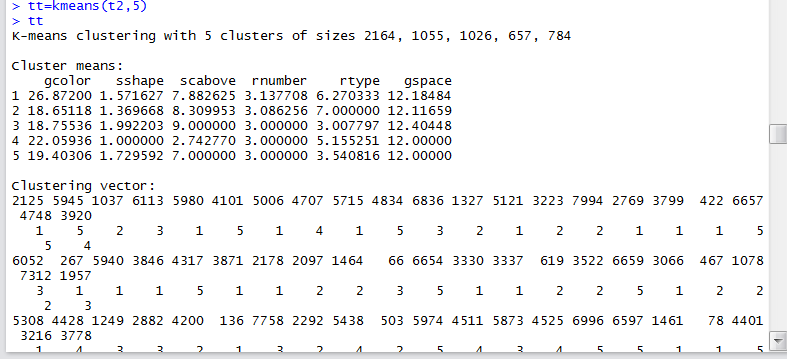


Plots are as follows:

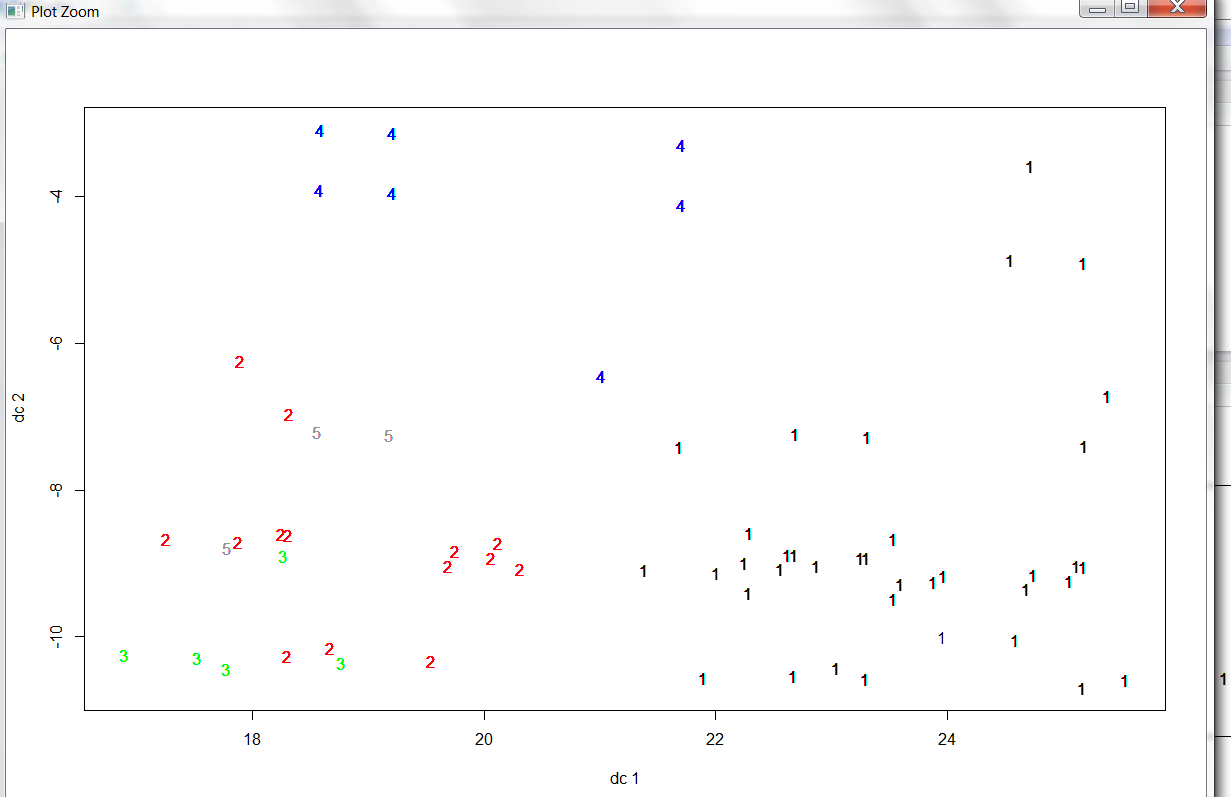


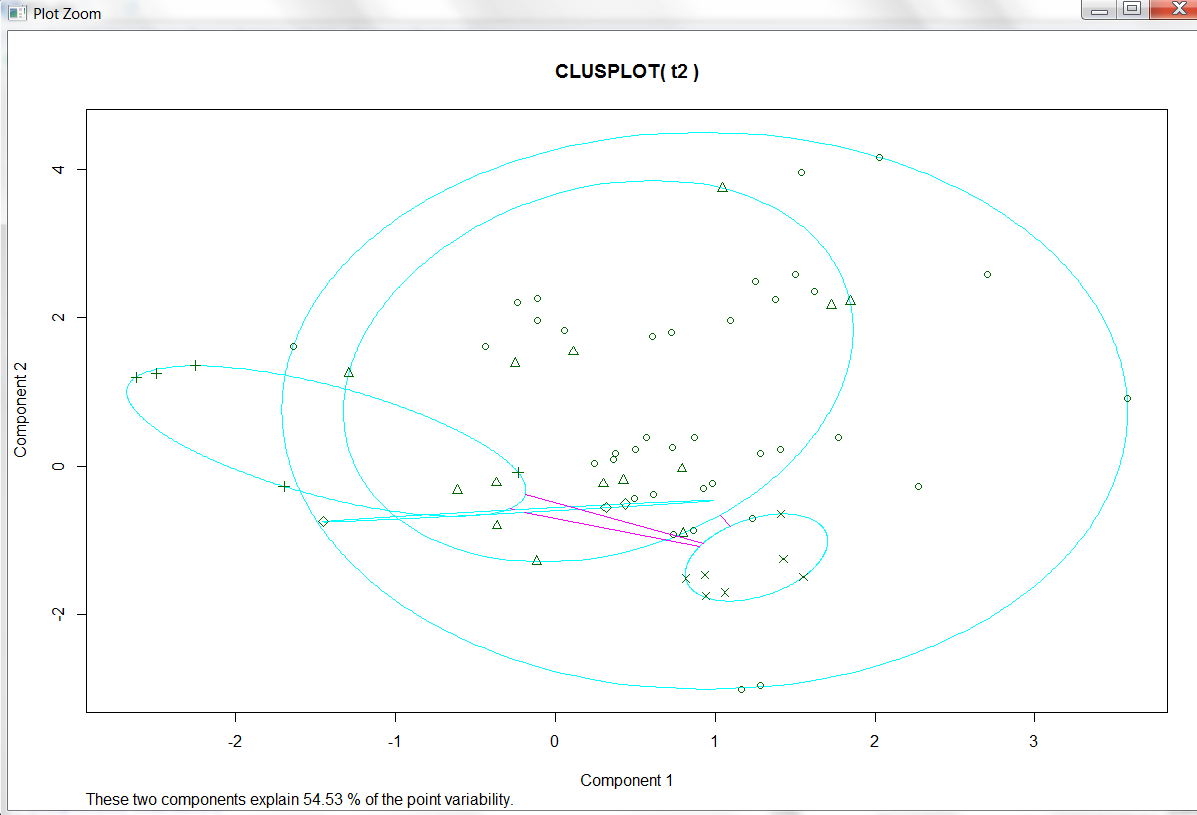


Kmeans with N=5 on 70% of data:



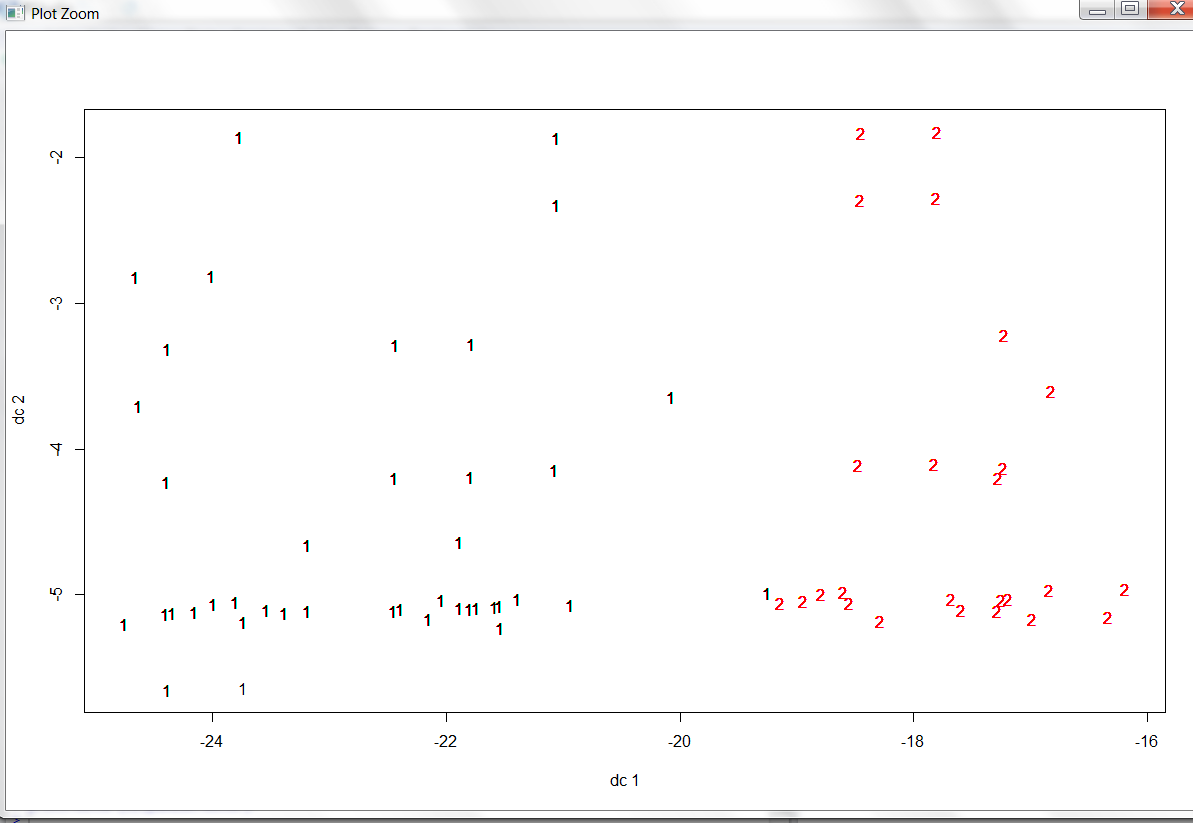
Plots are as follows:

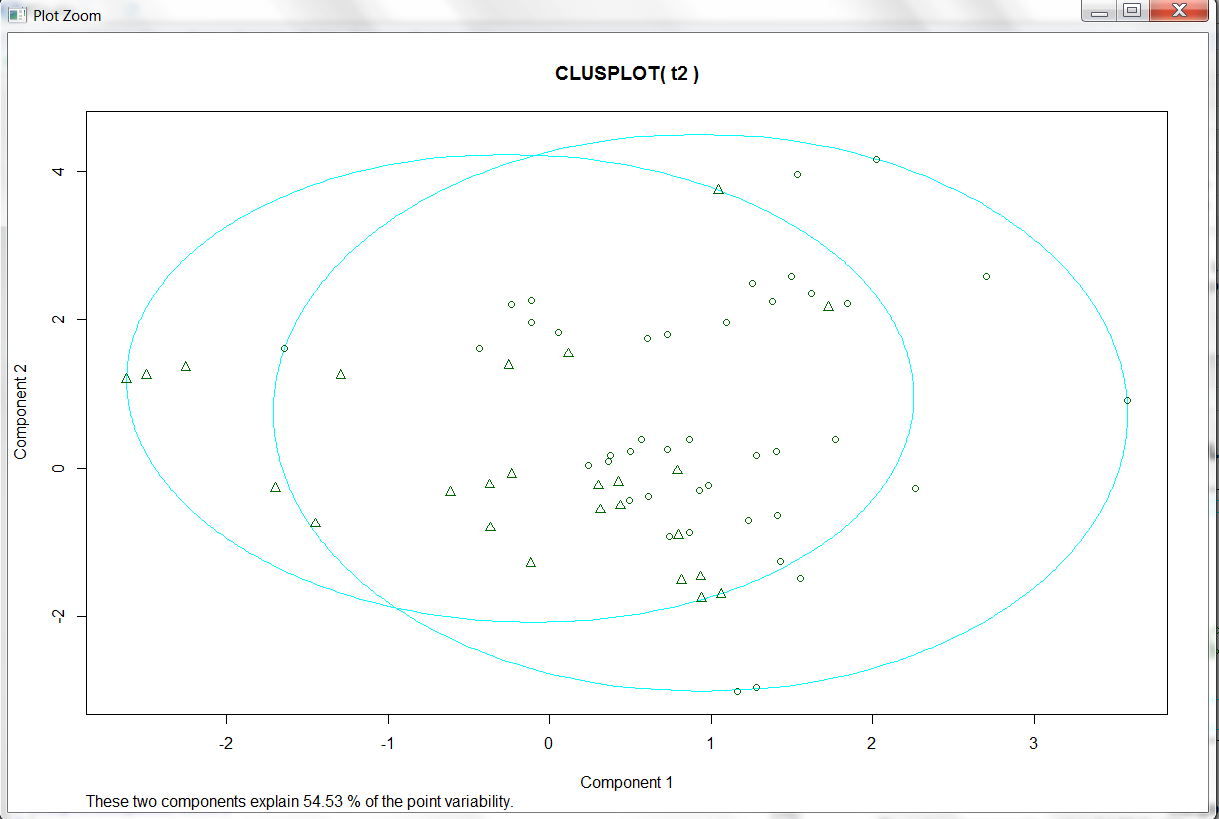




PAM with N=2 on 70% of data:

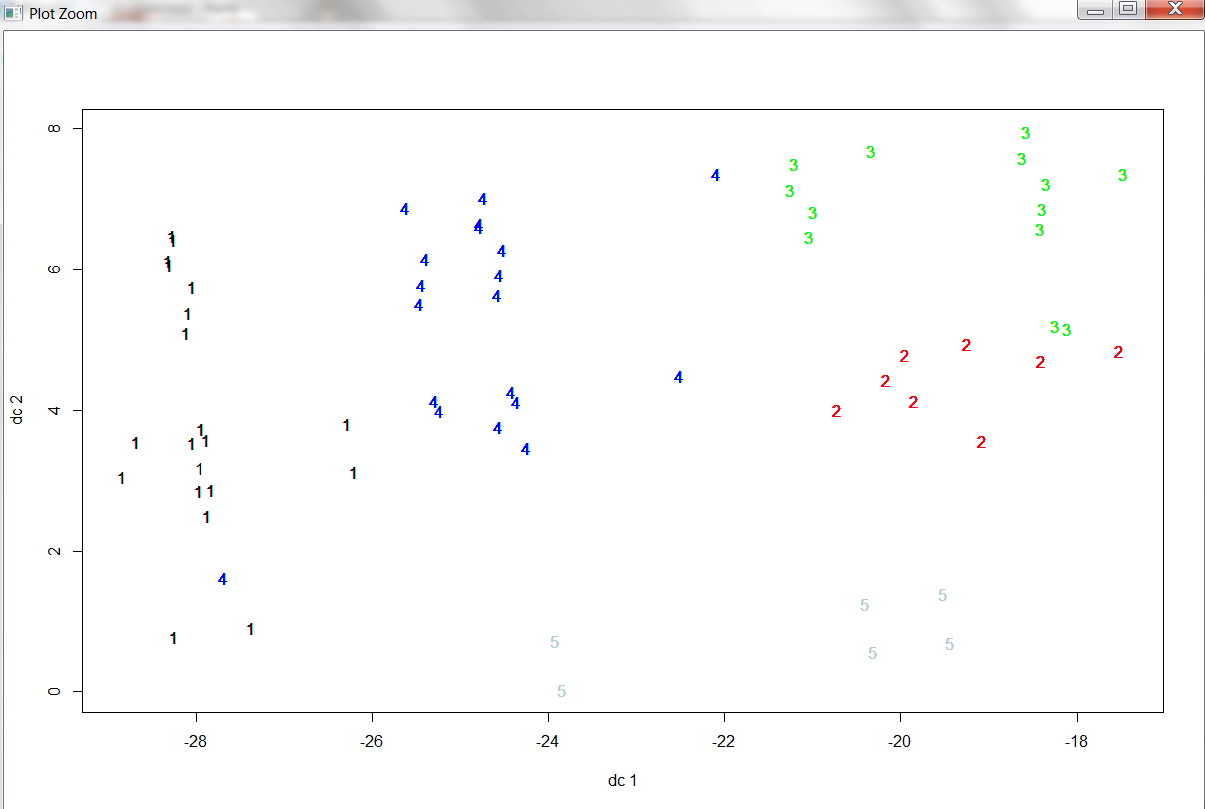
Plots are as follows:

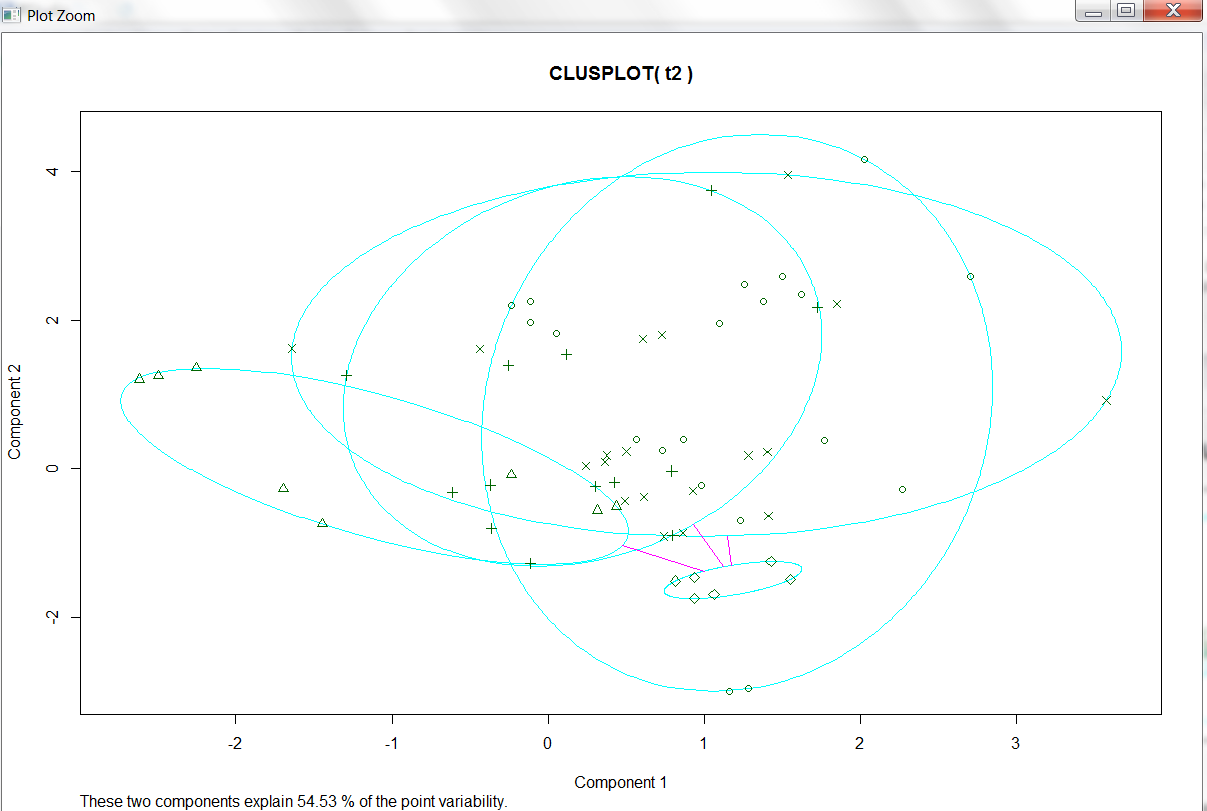




PAM with N=5 on 70% of data:

Plots are as follows:





**Missing value:**

After PCA, we reduced data and this new data did not included parameter stalk-root. Thus, we do not need to calculate missing data.

**Monte carlo predictive analysis:**

Algorithm:

1. Created decision tree using 70% as training data set.

2. Visualized results.

3. Tested results for remaining 30% data.

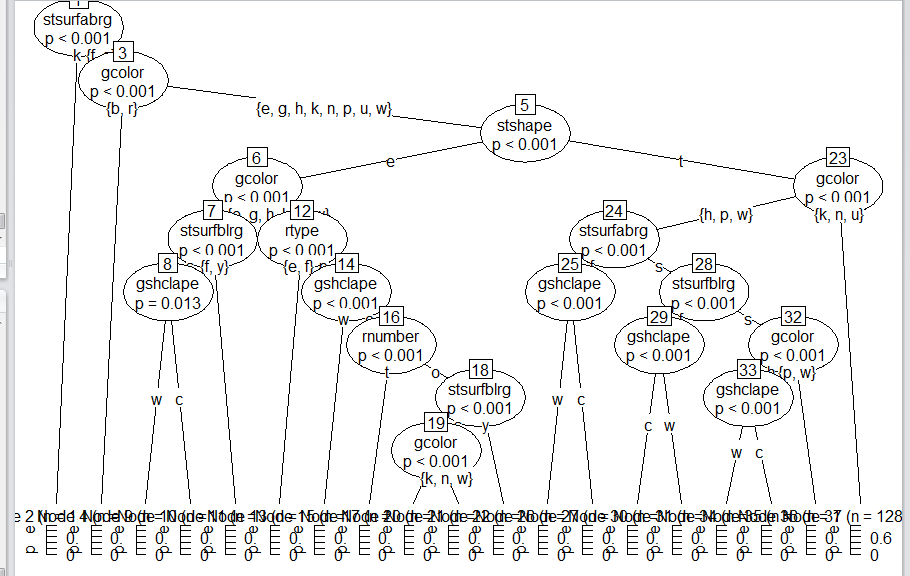
4. Applied decision tree on 30% data and if out come of decision tree is “edible” and it is marked as “poisonous” then changed it’s class to “unsure”.

**Program:**

****

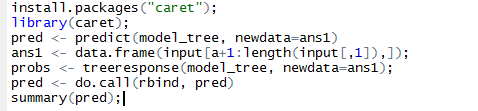
**Results:**

****

****

Predictive analysis :

Program :

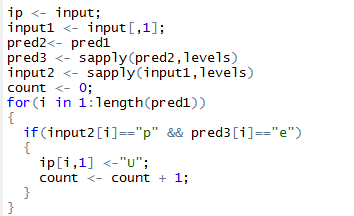


**Results:**



Finding out unsure class:

Program:



Result:

Changed state of “p” items to “unsure” class which show edible properties.

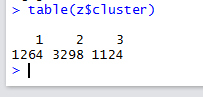
Most important factors required to decide class are as follows:

gcolor+ stshape + stsurfabrg

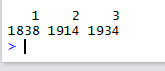
**Question 5.b)**

The results for N=3, 5, 7 are as follows:

The following clusters are for N=3 with 70% of values by K-means:



The following clusters are for N=3 with 70% of values by PAM:



From the results above we can see that PAM has divided data more evenly across the three clusters than K-means. This is why we think PAM gave a better result.

5c has been discussed with the plots.

**e. Learning of Data Science:**

We learnt about process of Data Science. As this data was large and having more than 23 attributes, first we reduced number of attributes by Principal component analysis. Then we explored clustering methods and visualized data. Then we implemented decision trees method for classification for and finding out unsure class.

We have learnt about predictive analysis, principal component analysis, clustering and classification methods in this Project. Visualization of results was another important feature we learnt in this project.

PCA is used to reduce attributes that act on data, so that we can play around with our data even better and more easily. Data science lets us scientifically study the data. We used a lot of functions in R to study and manipulate the mushroom data set. We studied and used various data mining (clustering) algorithms which is a major part of data science. We also used different plotting techniques to visualize our results and understand them better. We looked into the correlation function in order to reduce the number of attributes we need to work with.