Class project document

Team member Mengxi Nie, Yi Tu, Cong Tang, Yibu Lin

Professor Kaisler

Intro. to Big Data & Analytics CSCI 6444 8

October 1, 2017

The George Washington University

a. Document of R function

1. Read the file

mr<- read.csv('agaricus-lepiota)

processeddata = read.csv('processedData.csv')

2. data format

summary(mr)

describe(mr)

str(mr)

mr.df<-as.data.frame(mr)

3. #Draw the attribution

pairs(class~sshape+sroot,data = mr)

plot(class~ sshape+sroot,data = mr)

4. #Choose the proper attribution

mrf[,c(1,23)]<-sapply(mrf[,c(1,23)],as.character)

5. #Preparing the training data

fitControl <- trainControl(method = "repeatedcv", number = 10, repeats = 10)

6. #KNN

knn(traiK=mrtrain70.df, test=mrtest70.df, cl=train70labels, 3)

ct70\_3<-CrossTable(test70labels, knn70.7)

7. #PCA

mrpca3<-principal(data, nfactors=3, rotate="none")

8. #Kmeans

mr.kc3<-kmeans(data,3)

9. #NaiveBayes & prediction

sms\_70\_classifier <- naiveBayes(mrtrain70.df,mrtrain70.df[,1])

sms\_70\_predictions<-predict(sms\_70\_classifier,mrtest70.df)

10. lm&glm&prediction

train70\_lm <- lm( class ~ cshape + csurface + ccolor + bruises + odor+

gattach + gspace + gsize + gcolor + sshape + sroot +

ssabove + ssbelow + scabove + scbelow + vcolor+

rnumber + rtype + spcolor + popnum + habitat, data = mrtrain70.df)

test70\_lm\_Pre <- predict(train70\_lm, data = mrtest70.df)

train70\_glm <- glm(class ~ cshape + csurface + ccolor + bruises + odor+

gattach + gspace + gsize + gcolor + sshape + sroot +

ssabove + ssbelow + scabove + scbelow + vcolor+

rnumber + rtype + spcolor + popnum + habitat, data = mrtrain70.df)

test70\_glm\_Pre <- predict(train70\_glm, data = mrtest70.df)

11. svm

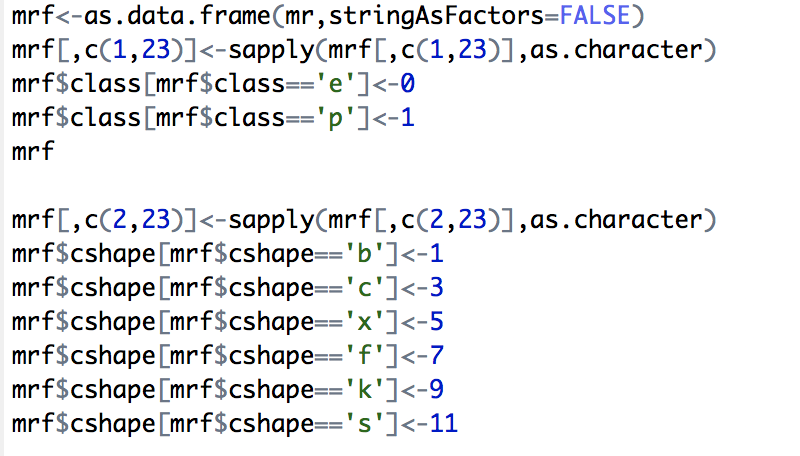
mrfsvm <- svm(class~ccolor+cshape, data = mrf,kernel="linear")

12.pam

data.pam3<-pam(mrf, k = 3, metric = "euclidean", stand = FALSE)

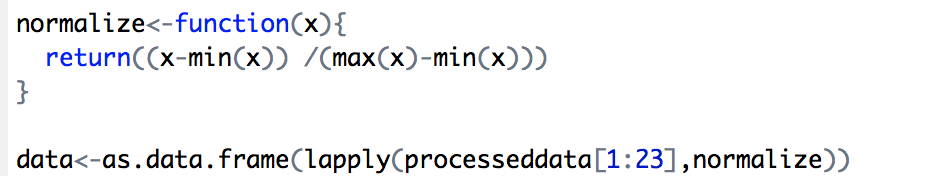
B. Table with results

As for the given data, we can’t use the original data directly to analyze. So we preprocessed the data first. Just like this:

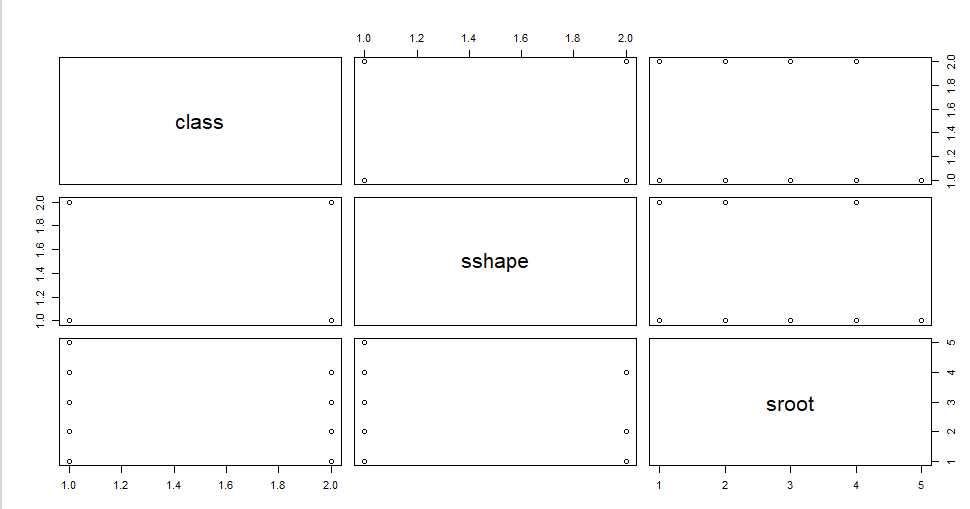


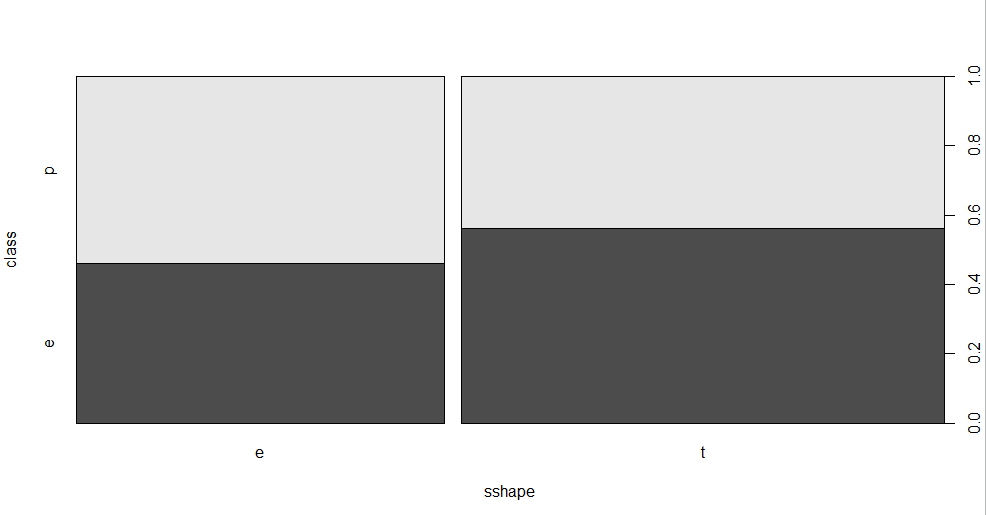
we do this substitutions for all the attributes, then we get a data with numeric attributes.

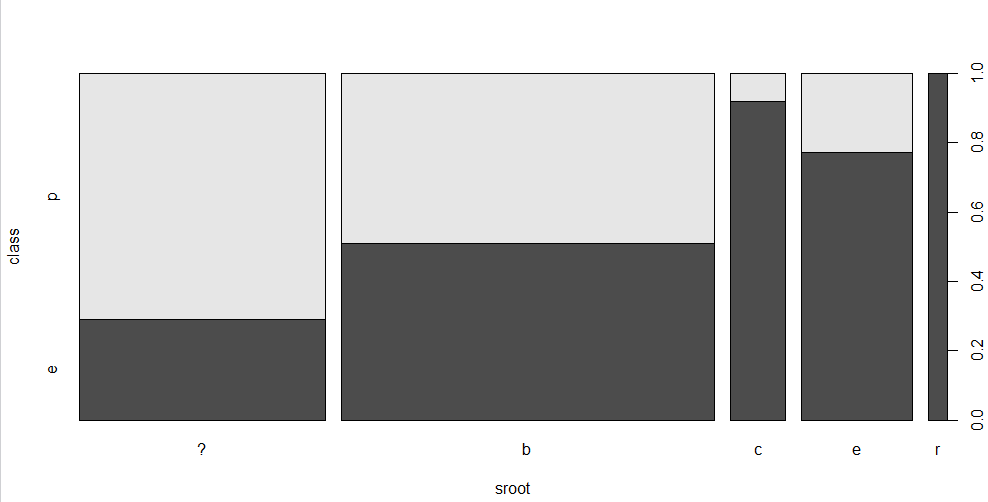
Then we normalize the data:



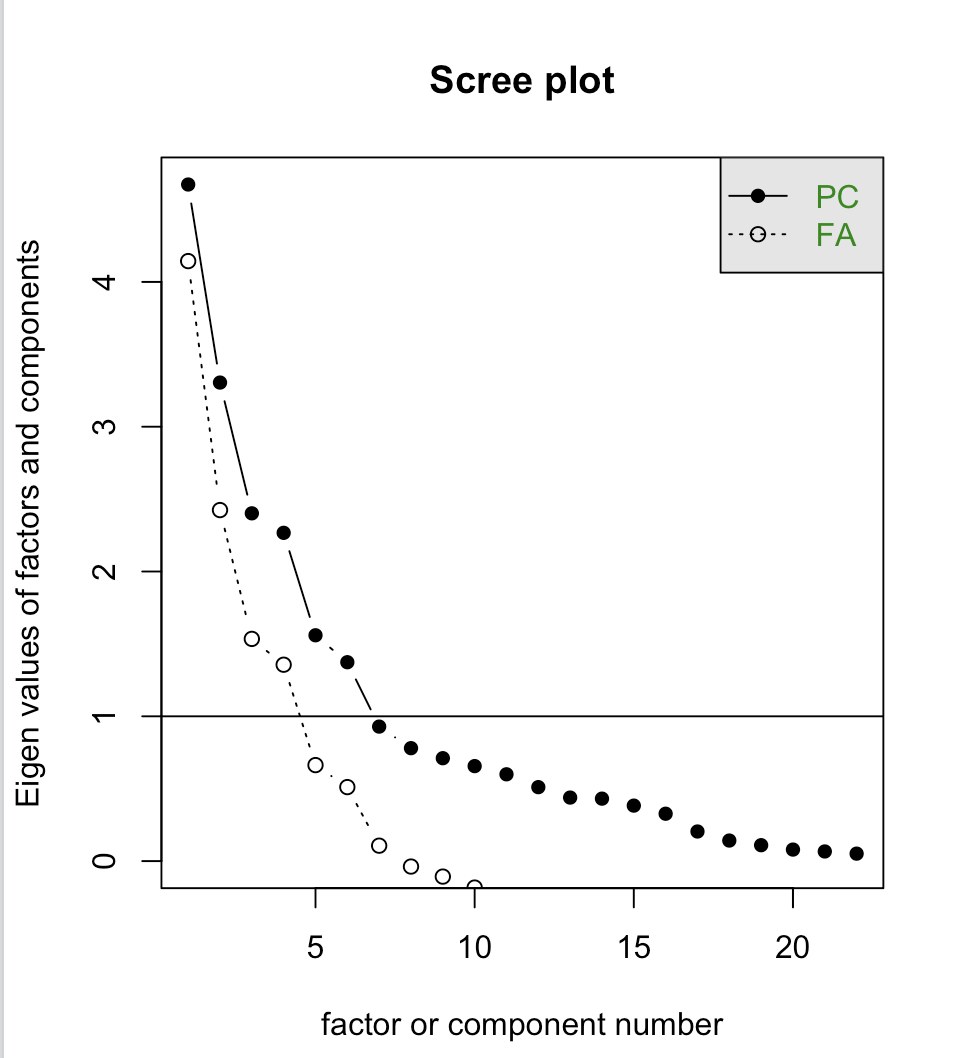
Plot data

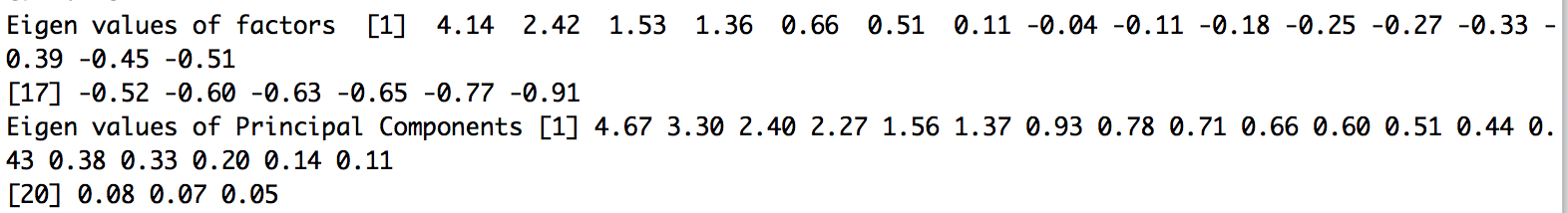


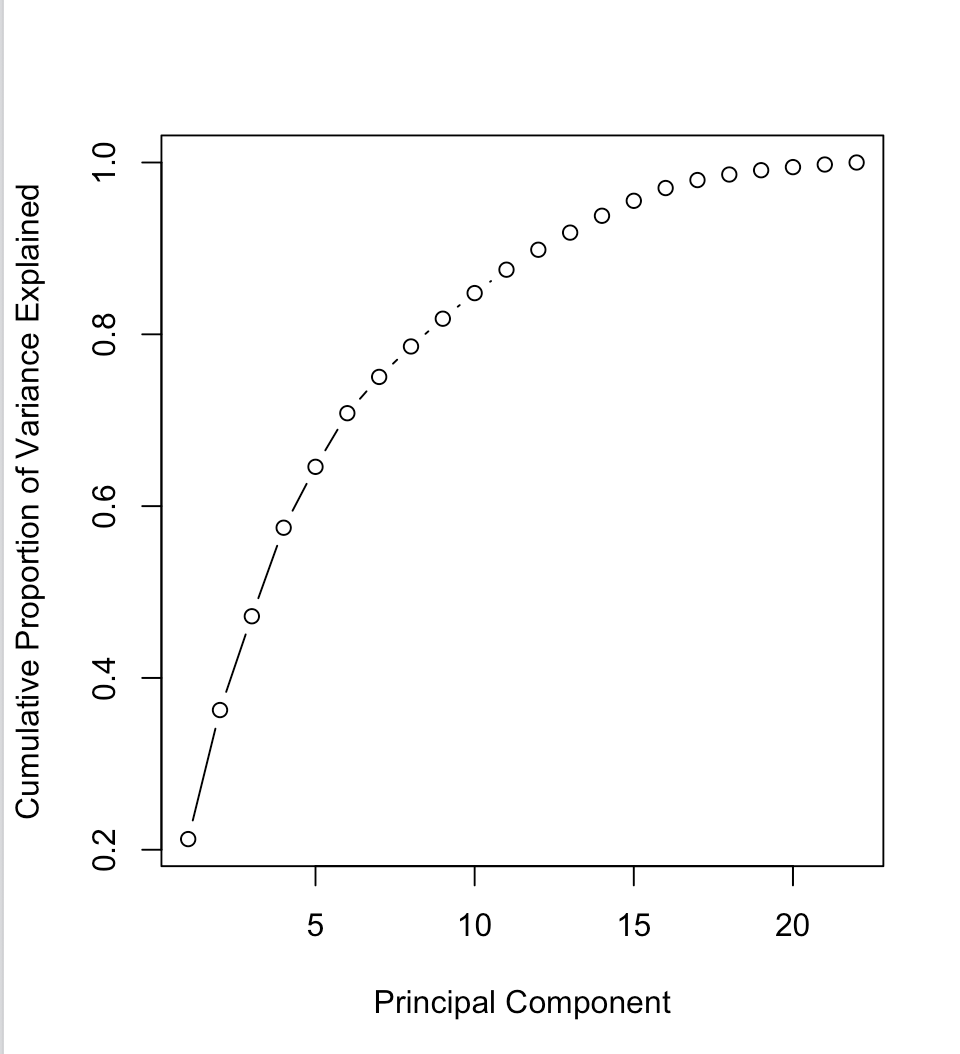




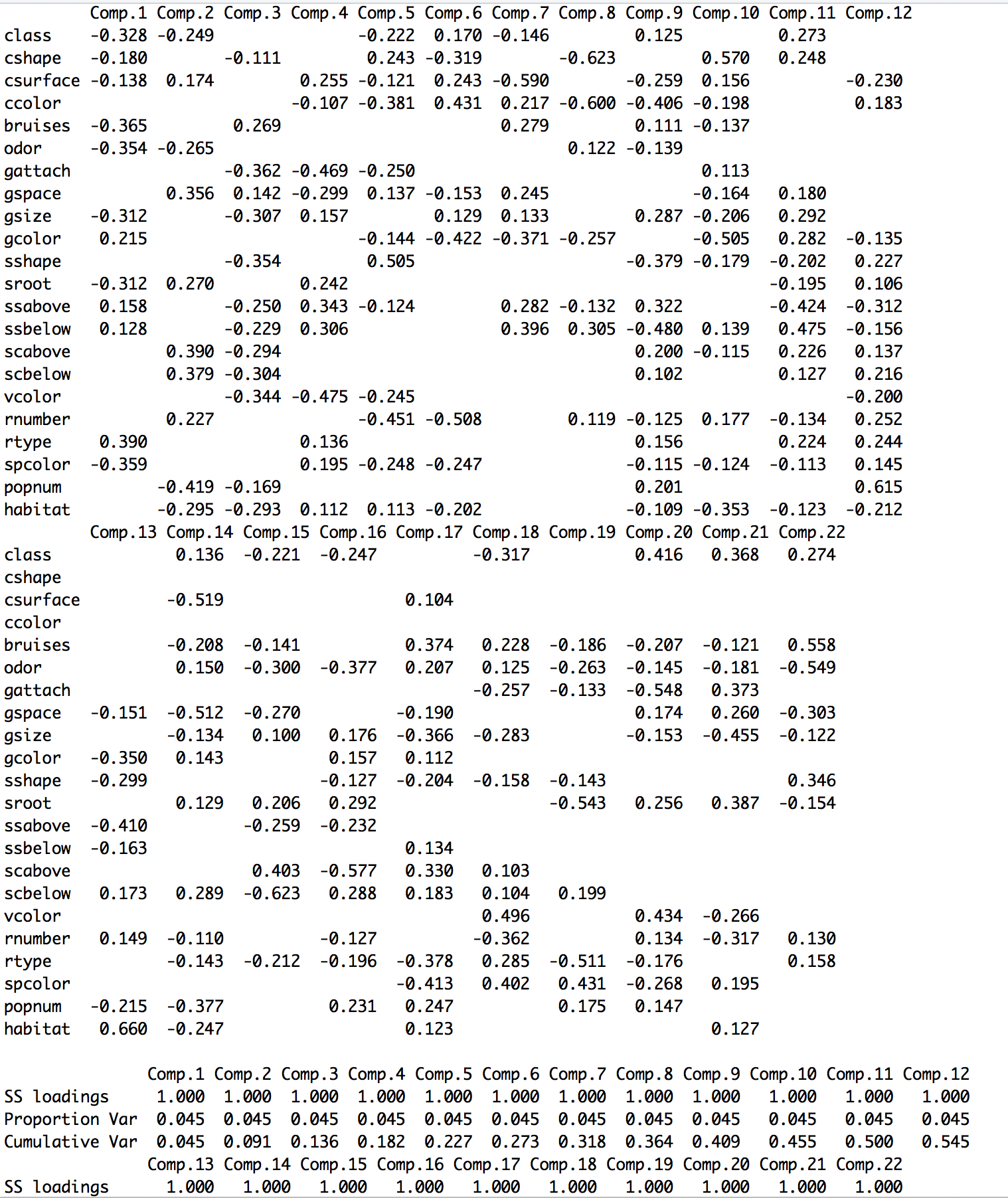
First, we use PCA (principal component analysis) to reduce the data to lower dimension.







the first 8 take up over 80% of the original data information. So we can check the first 8 components include which attributes:



So, we choose class, cshape, csurface, odor, gspace, gsize, gcolor, ssabove, ssbelow, habitat.

According to agaricus-lepiota.names, I choose these attributes:

* odor=NOT(almond.OR.anise.OR.none) odor=a OR *l* OR n
* spore-print-color=green spcolor=r
* odor=none.AND.stalk-surface-below-ring=scaly.AND. (stalk-color-above-ring=NOT.brown) odor = n AND ssbelow = y AND scabove != n
* habitat=leaves.AND.cap-color=white

OR population=clustered.AND.cap\_color=white

habitat = *l* AND ccolor =w OR popnum = c AND ccolor = w

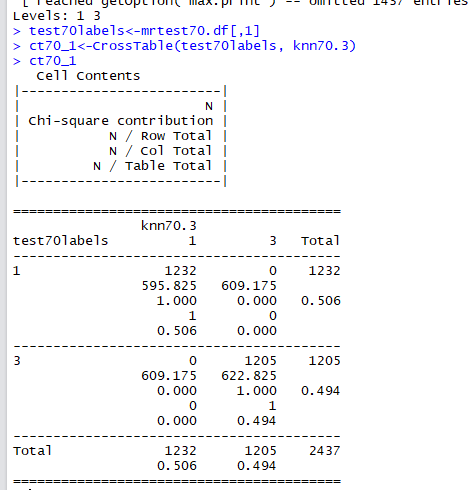


In this project, we use three methods to do the cluster analysis: KNN, K-means and Naive Bayes. Moreover, we do the Principal Components Analysis to do the data reduction to eliminate some attributes through. In each method, we set samples into K = 3, 5, 7 classes. Then, the result table got from different clustering method are showed as followed:

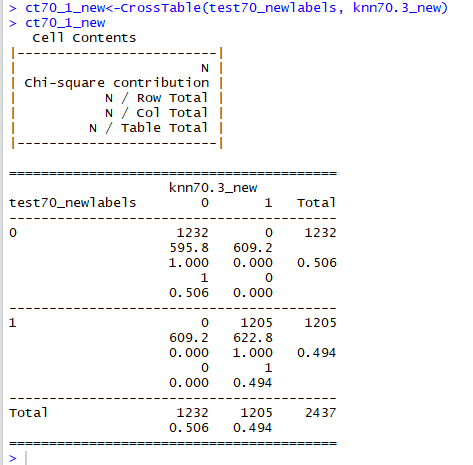
1. Knn(K-Nearest Neighbor)

When set Training data pair 70 - 30

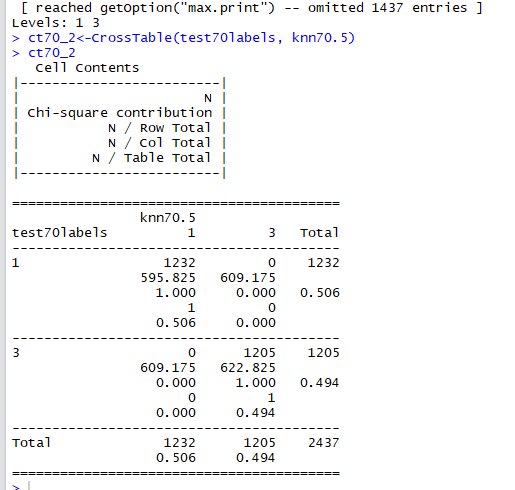
K = 3 (ALL Attributes)



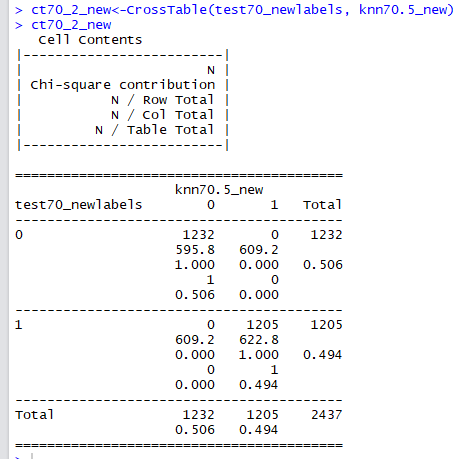
K = 3 (Removing meaningless attributes)



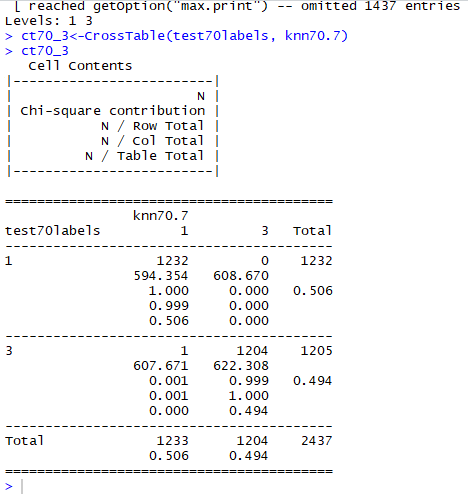
K = 5 (ALL Attributes)



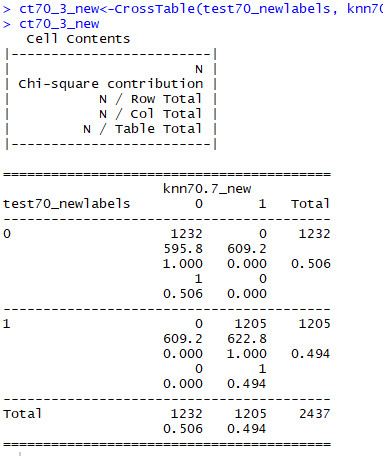
K = 5 (Removing meaningless attributes)



K=7 (ALL Attributes)

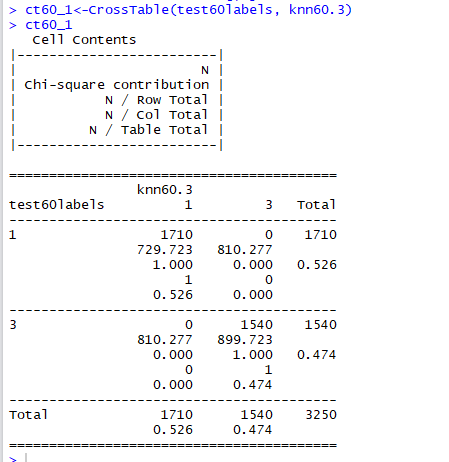


K=7 (Removing meaningless attributes)

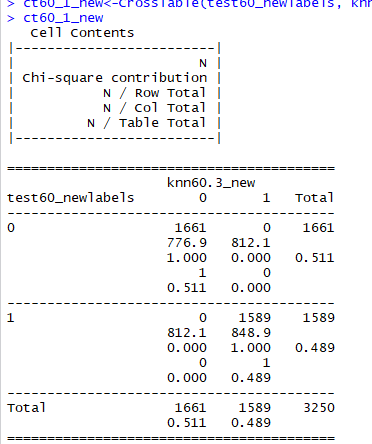


When set Training data pair 60 - 40

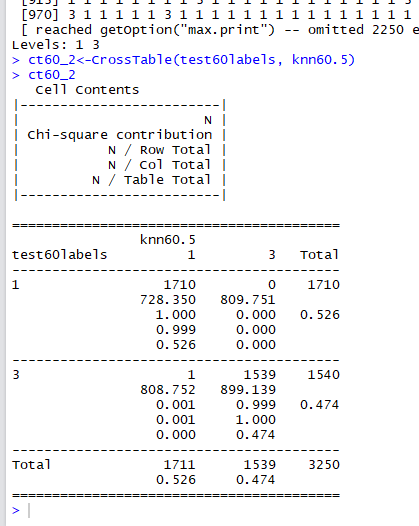
K = 3（ALL Attributes）



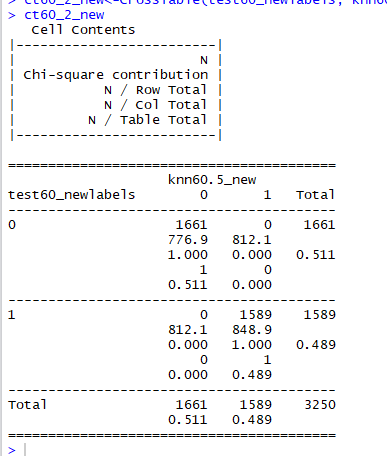
K = 3（Removing meaningless Attributes）



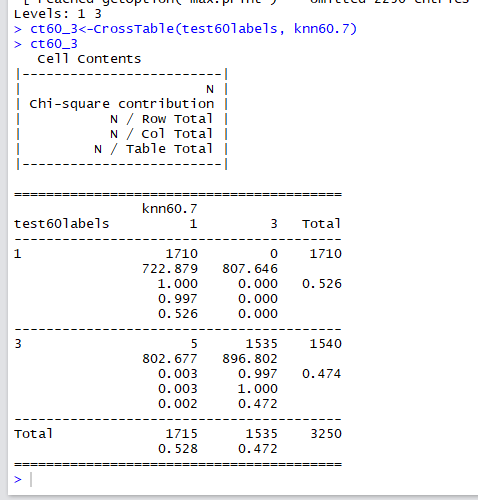
K = 5 (ALL Attributes)



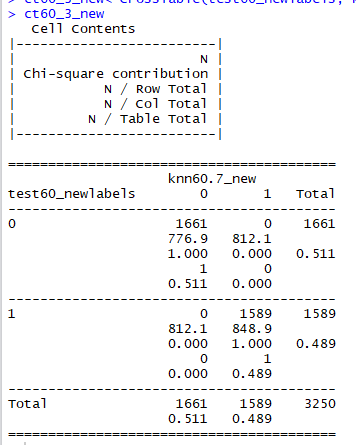
K = 5（Removing meaningless Attributes）



K = 7 (ALL Attributes)

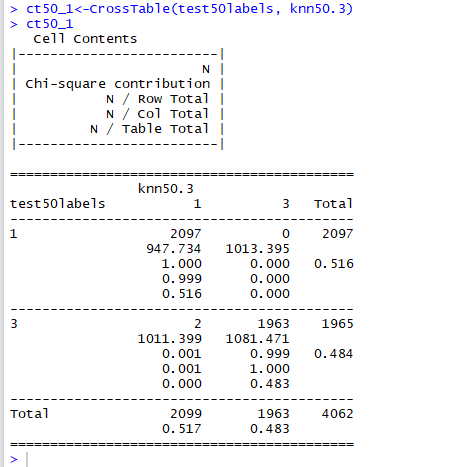


K = 7（Removing meaningless Attributes）

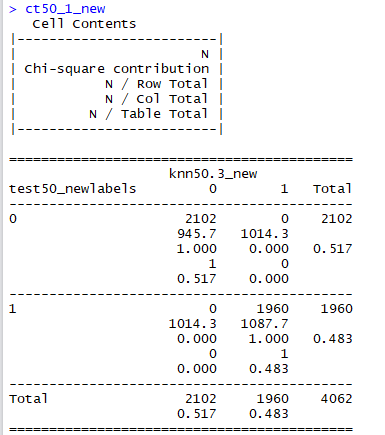


When set Training data pair 50 - 50

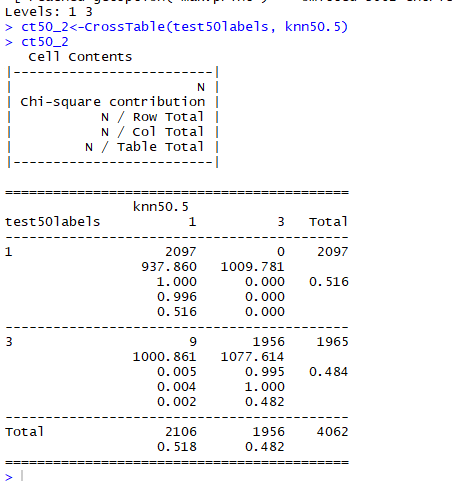
K = 3(ALL Attributes)



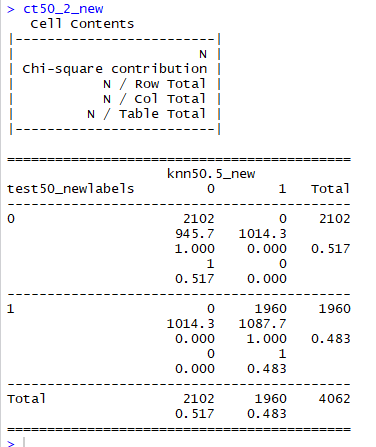
K = 3（Removing meaningless Attributes）



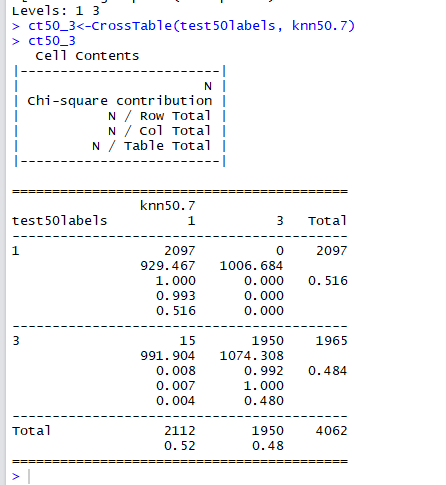
K = 5(ALL Attributes)



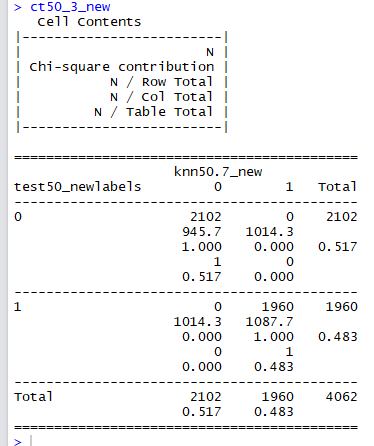
K = 5（Removing meaningless Attributes）



K = 7(ALL Attributes)



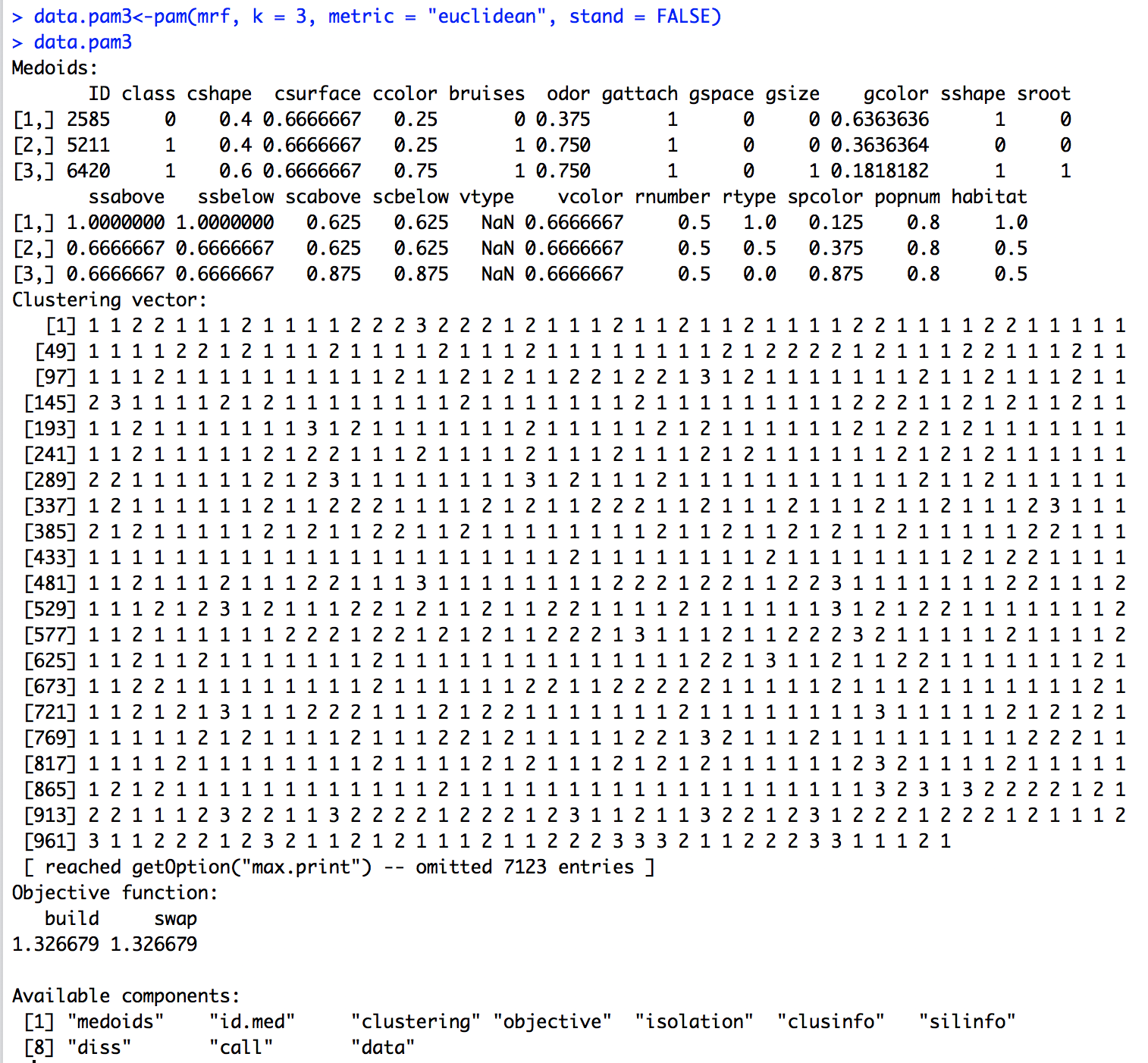
K = 7（Removing meaningless Attributes）



2. pam

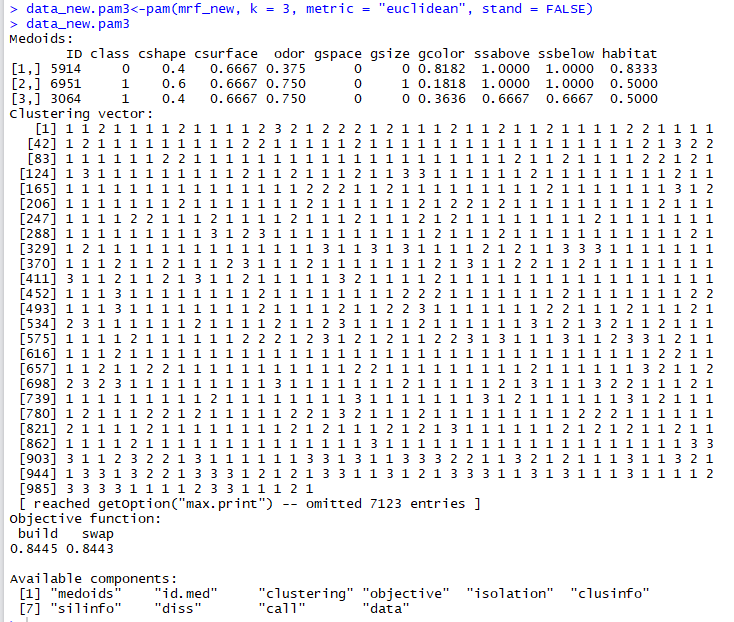
K = 3

data.pam3<-pam(mrf, k = 3, metric = "euclidean", stand = FALSE)

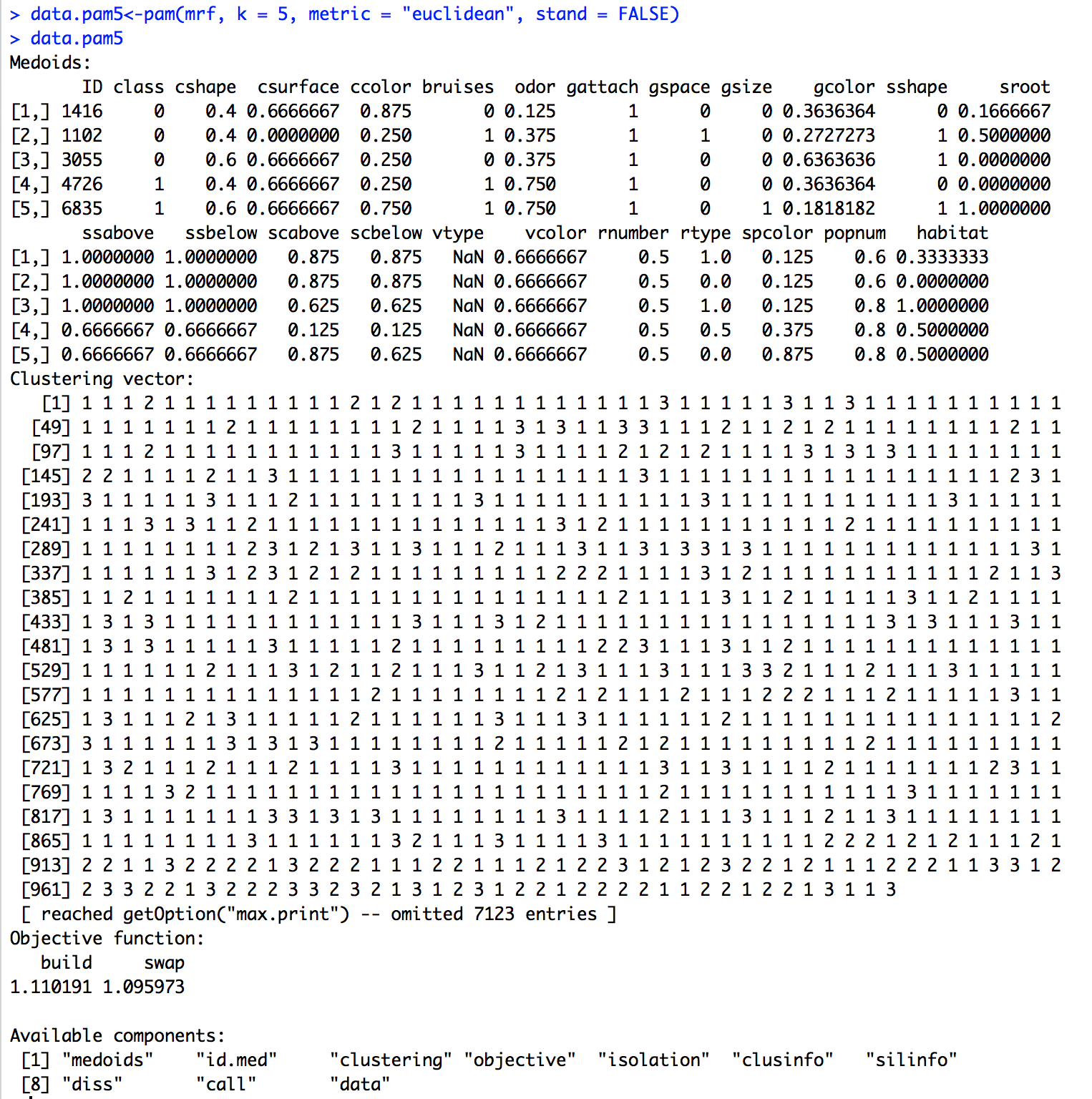


K=3 (removing the meaningless attributes)

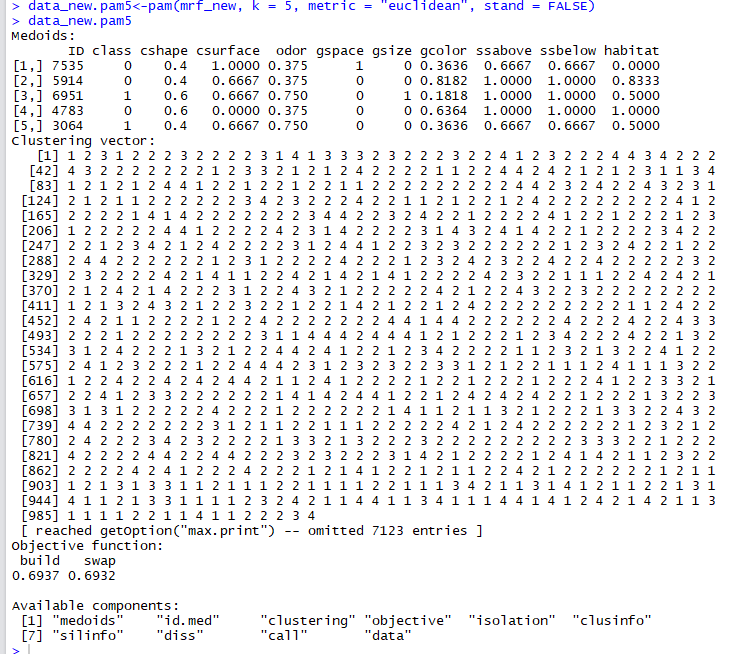
data\_new.pam3<-pam(mrf\_new, k = 3, metric = "euclidean", stand = FALSE)



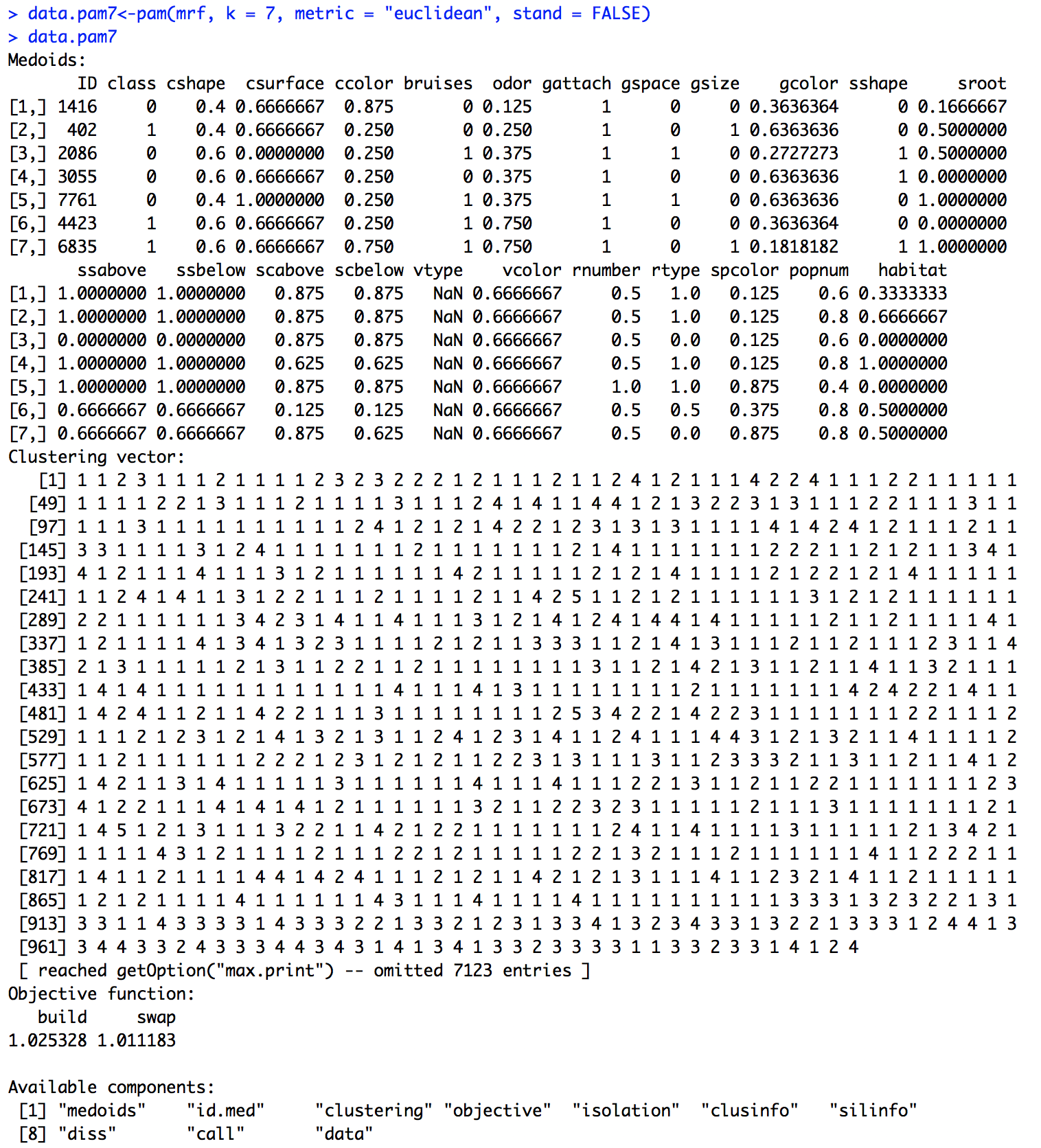
K = 5



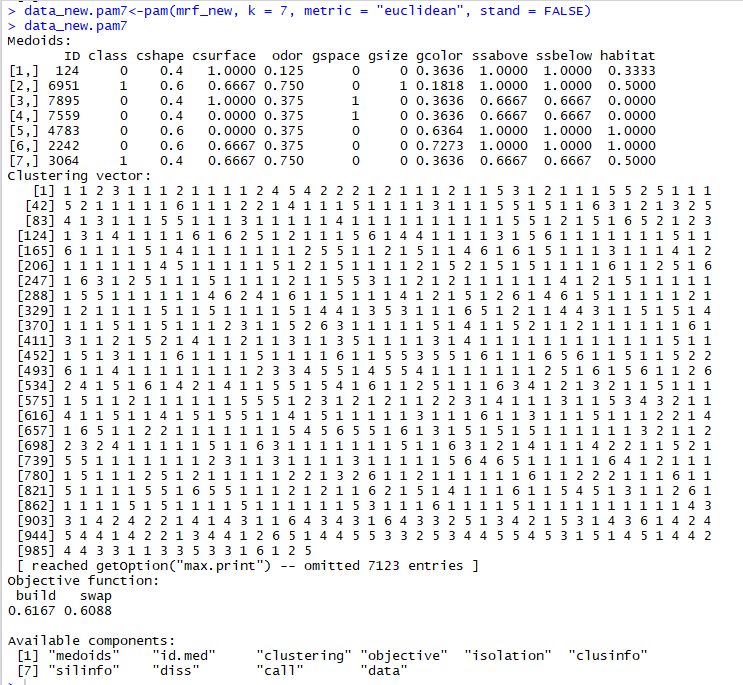
K = 5 (removing the meaningless attributes)



K = 7

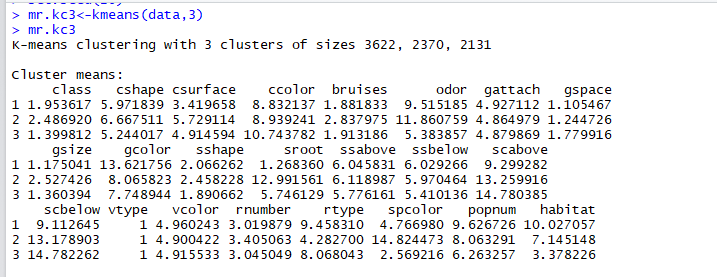


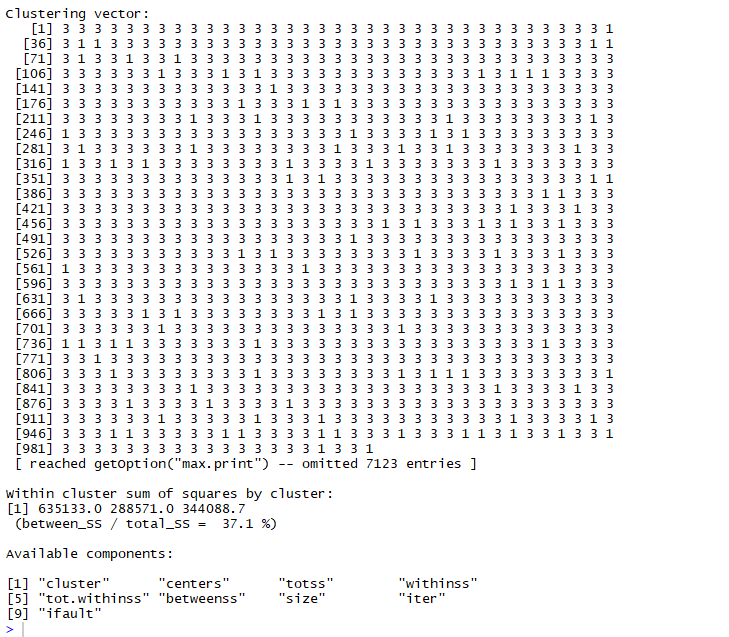
K = 7 (removing the meaningless attributes)



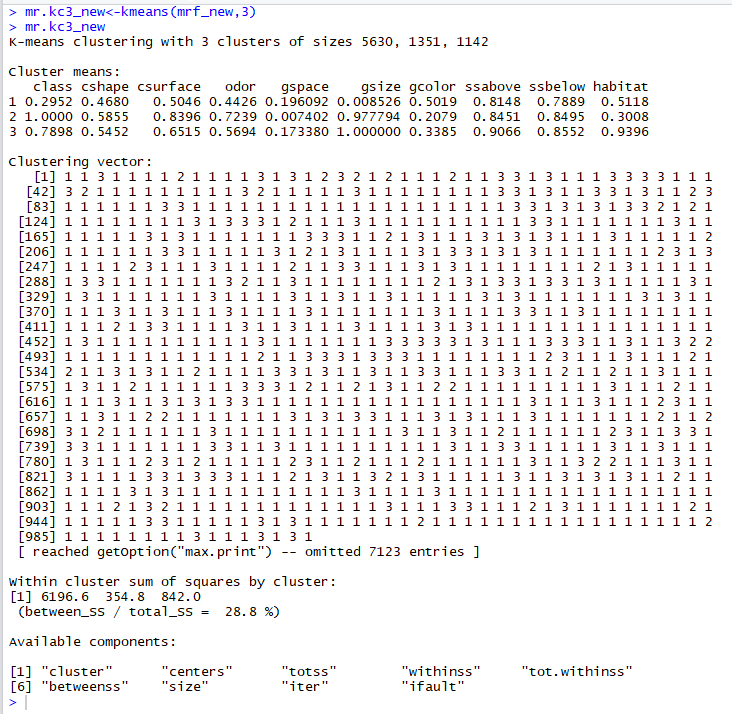
3.K-means

When we set K=3(ALL Attributes)

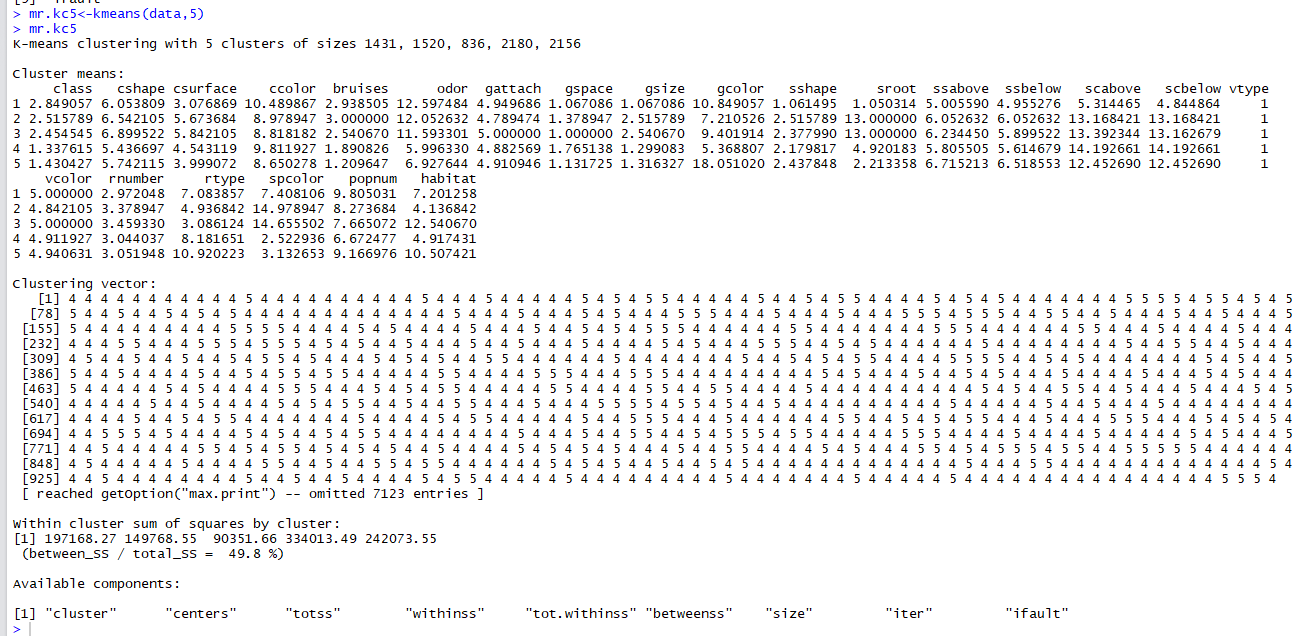




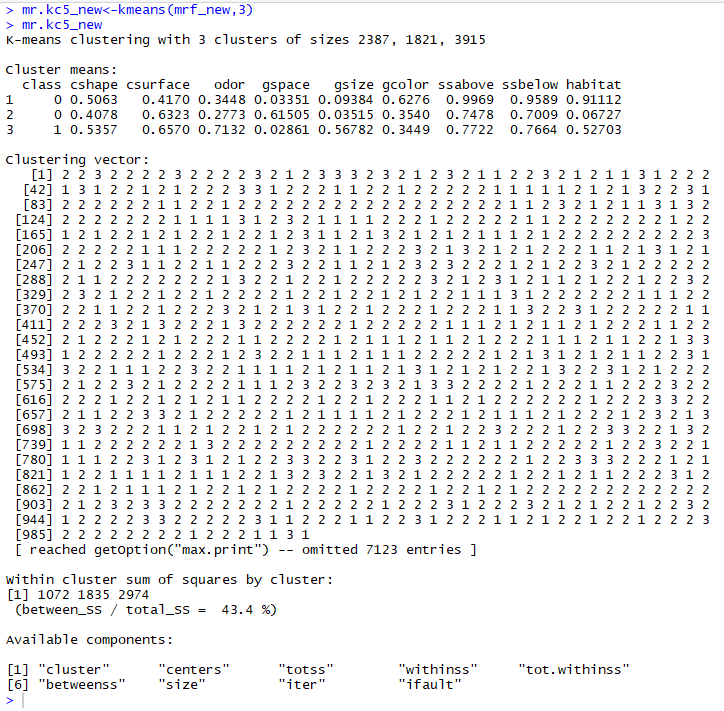
K = 3 (removing the meaningless attributes)



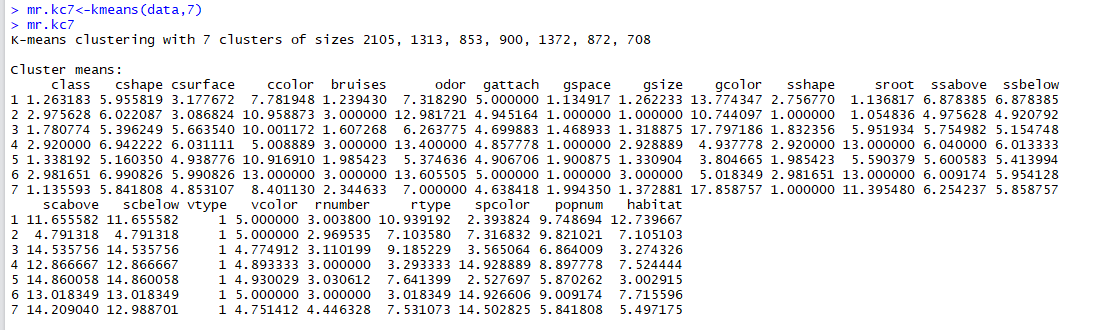
K = 5(ALL Attributes)

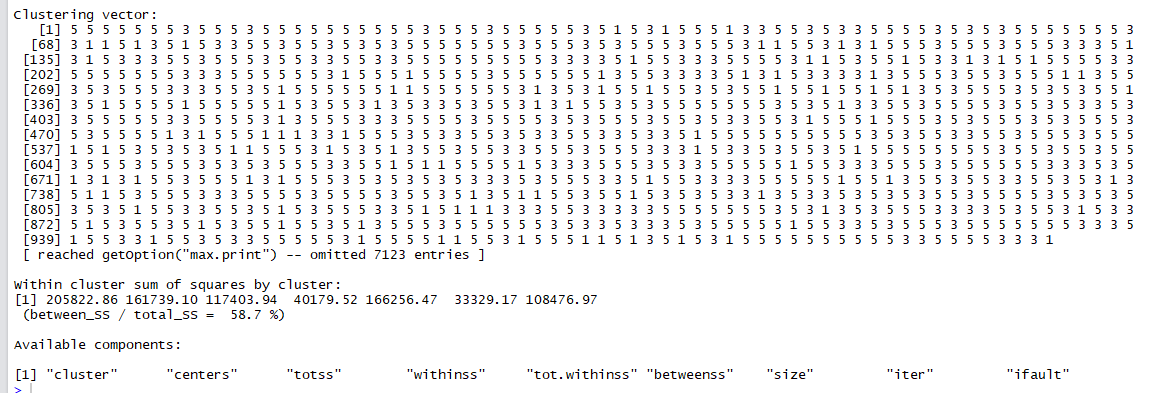


K = 5 (removing the meaningless attributes)

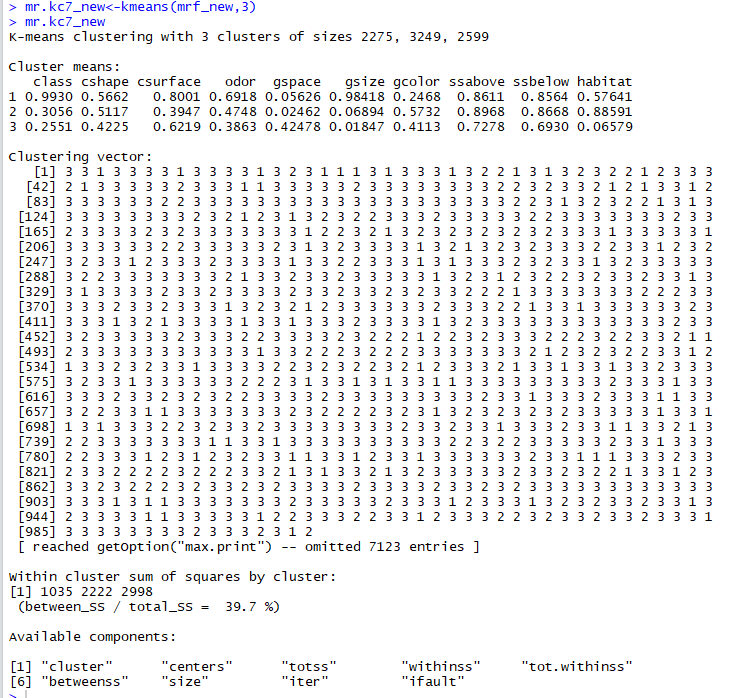


K=7(ALL Attributes)



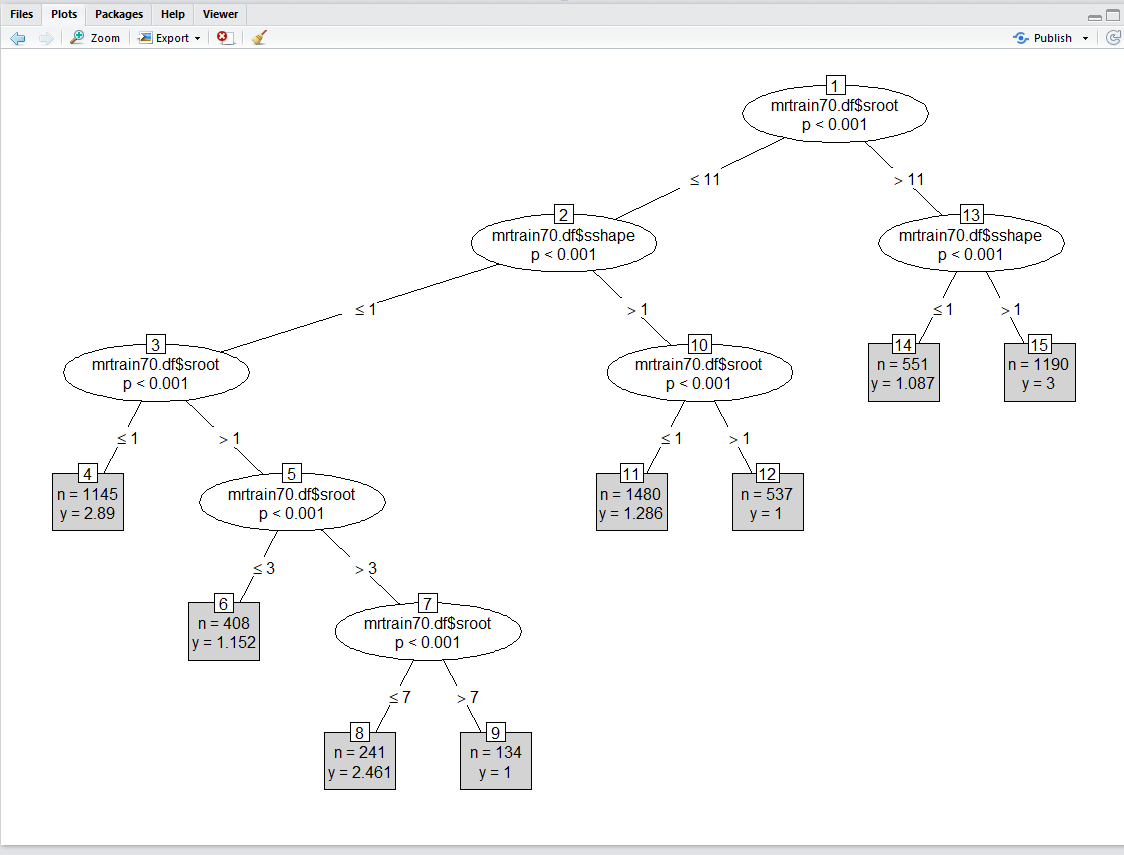


K = 7 (removing the meaningless attributes)

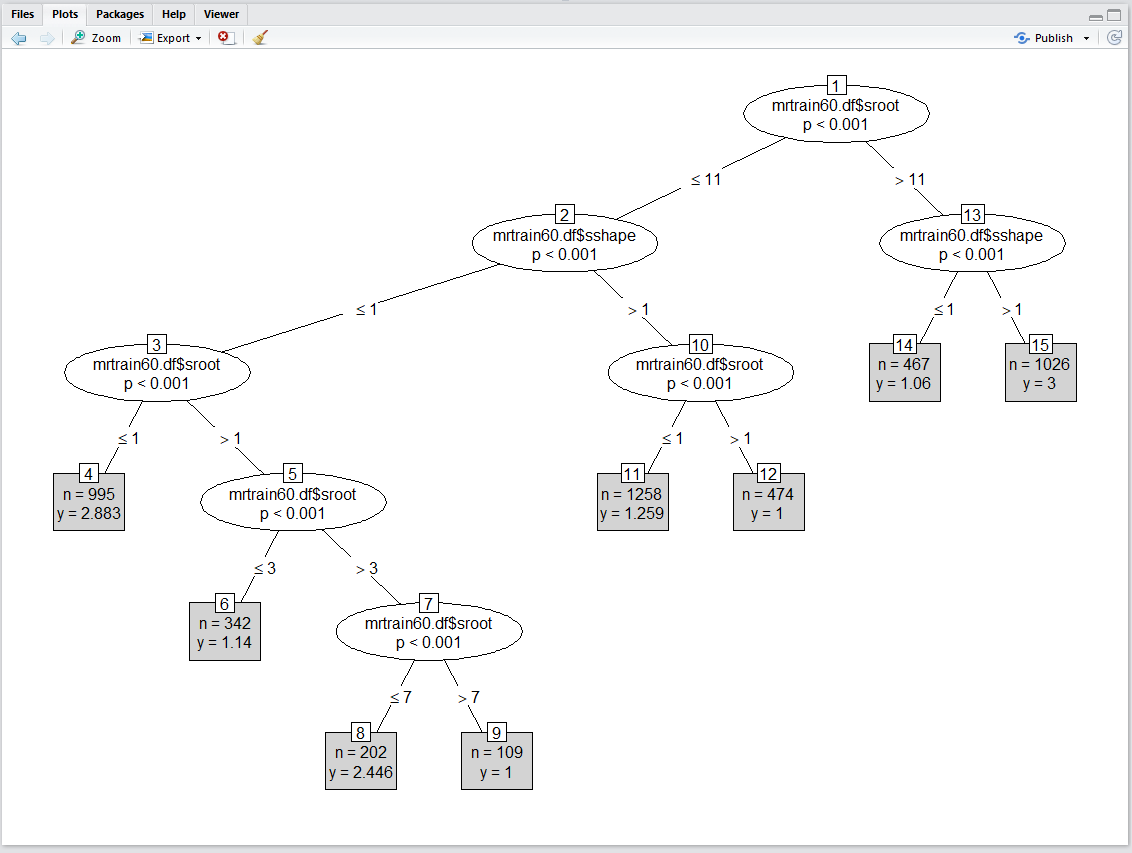


4. (Additional clustering method we choose) Decision Tree

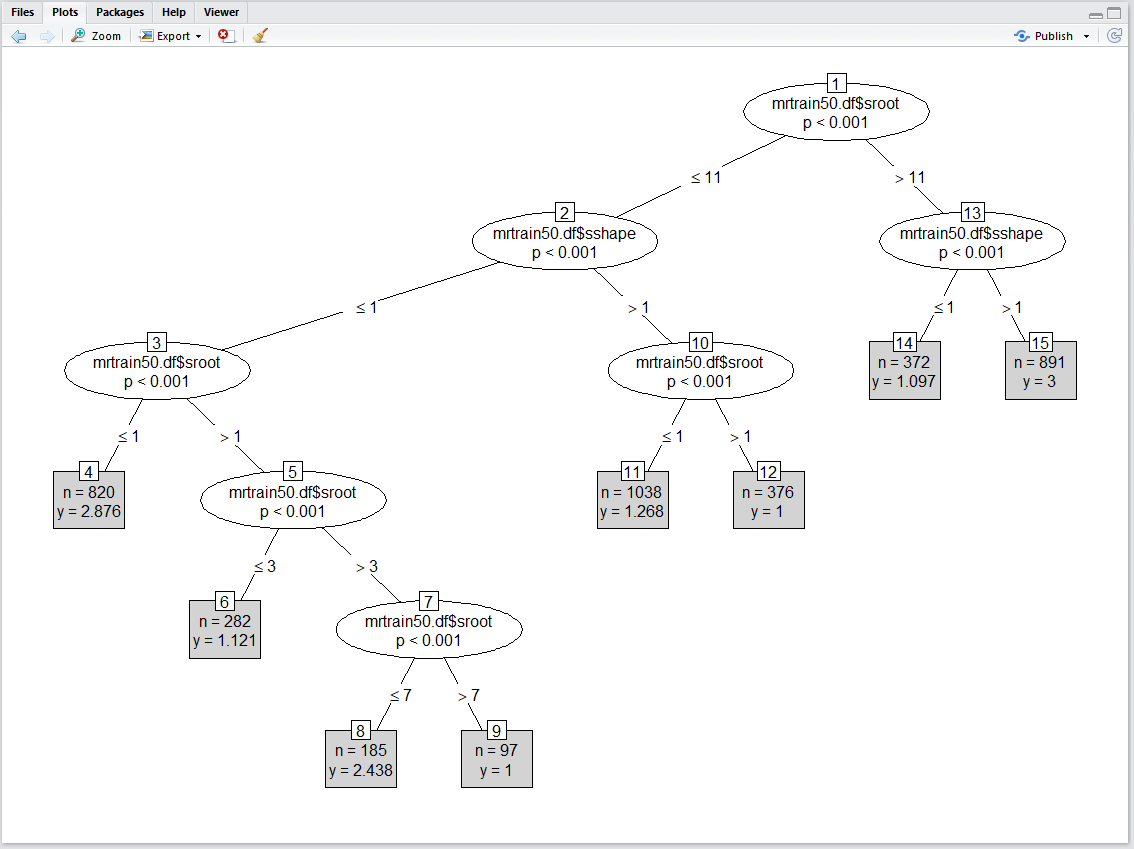
Training70%：



**Training60%：**



**Training50%：**



c. Result discussion

1. In method KNN

we set three pairs of data sets with the ratios, 70-30, 60-40,50-50. And in each pair, we set cluster into K = 3,5,7. So, we will discuss the result in each situation.

In 70-30, when k =3, we can see the true negative rate is 0.506, true positive is 0.494. So, the rate of correct is 100%. When k = 5, the true negative rate is 0.506, the true positive is 0.494, the rate of correct is 100%. When k = 7, the true negative rate is 0.506, the true positive is 0.494, the rate of correct is 100%.

In 60-40, when k = 3, the true negative rate is 0.526, the true positive is 0.474, the rate of correct is 100%. When k = 5, the true negative rate is 0.526, the true positive is 0.474, the rate of correct is 100%. When k = 7, the true negative rate is 0.526. the true positive is 0.472, the rate of correct is 99.8%

In 50-50 when k = 3, the true negative rate is 0.516, the true positive is 0.483, the rate of correct is 99.9%. When k = 5, the true negative rate is 0.516, the true positive is 0.482, the rate of correct is 99.8%. When k = 7, the true negative rate is 0.516, the true positive is 0.48, the rate of correct is 99.4%.

From these results, we can easily find out, As the right pair becomes bigger, the correct rate is lower. And when we pick a bigger cluster, the correct rate becomes lower. So, if we just want to the best prediction results, just pick the low cluster and high known data sets.

2. In method K-means

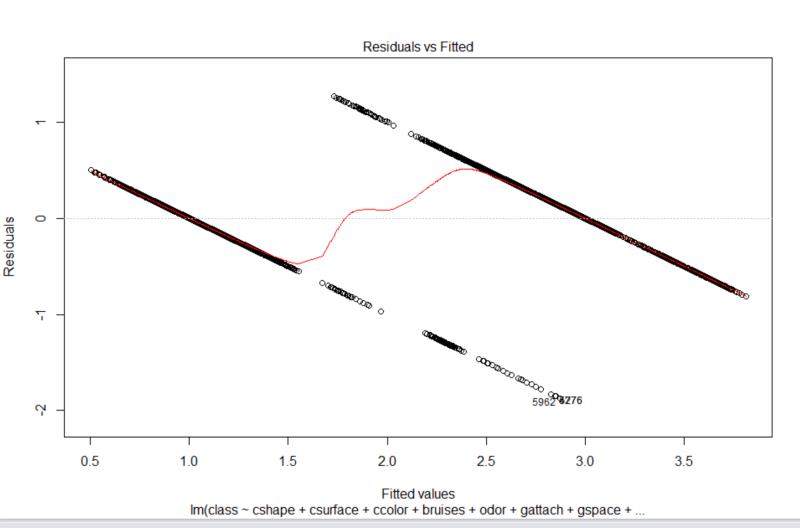
In method K-means, we use default algorithm Hartigan-Wong. And in the process, we also set three cluster K = 3,5,7. The result discussed as followed:

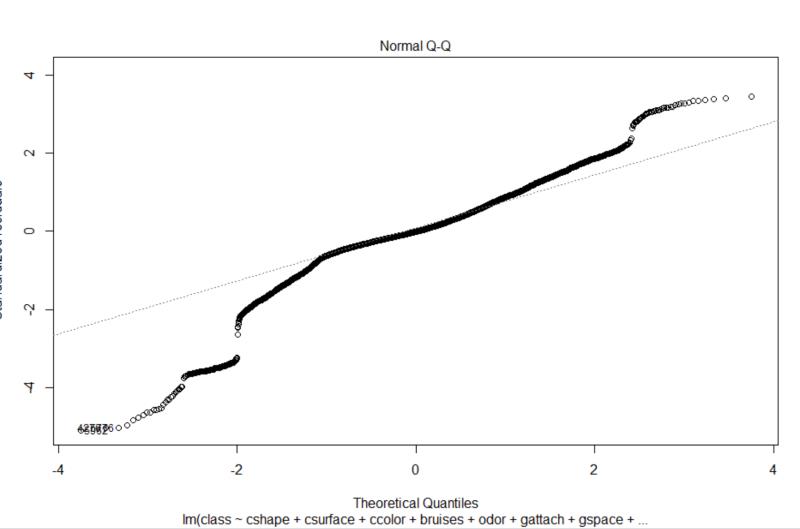
When K = 3, the ratio of between\_ss and total\_ss is 37.1. When K = 5, the ratio of between\_ss and total\_ss is 49.8%, When K = 7, the ratio of between\_ss and total\_ss is 58.7%. This number stands for the distance in a clustering. The larger the value, the smaller the distance in the clustering, which means a better clustering. As a result, it seems when K = 3, we have the best clustering and result.

**3, In method pam**

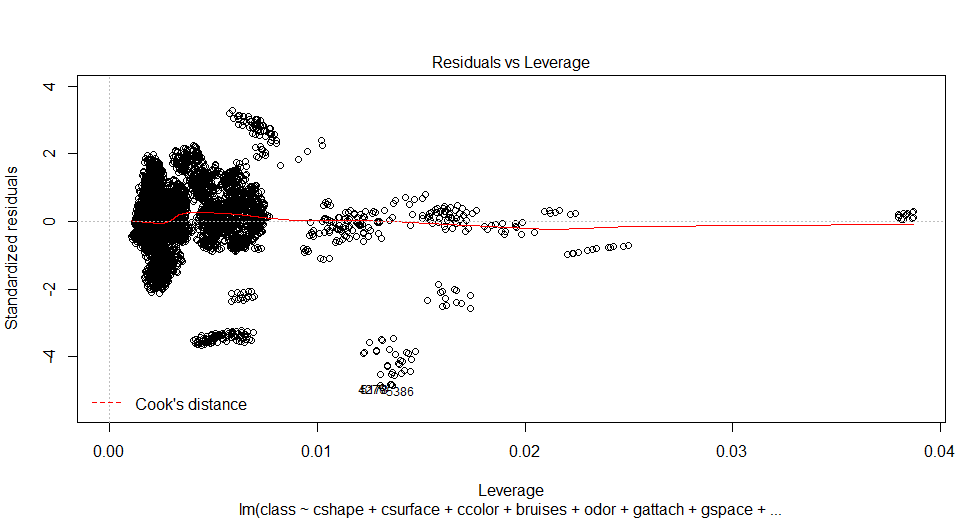
d. Table from LM and GLM

1. train70\_lm(All attributes)

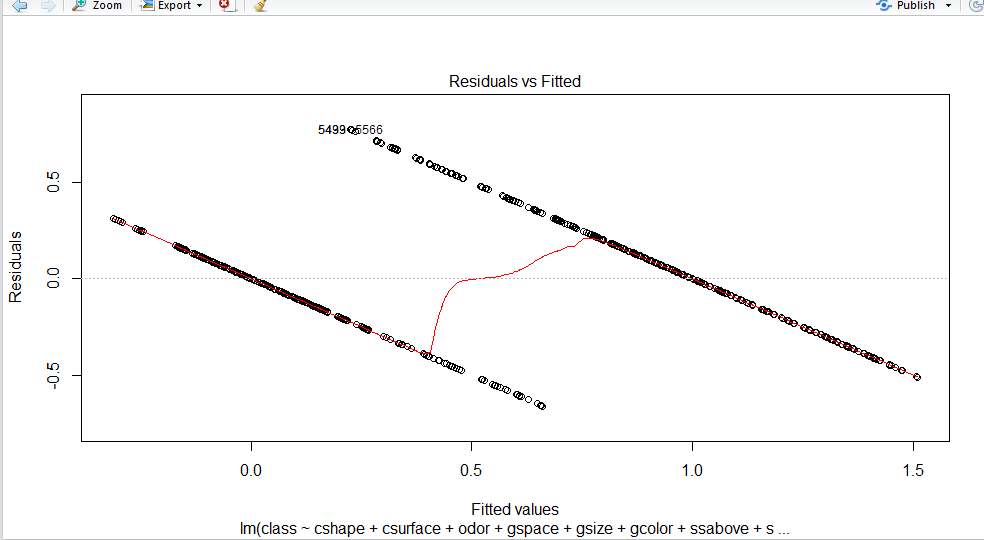


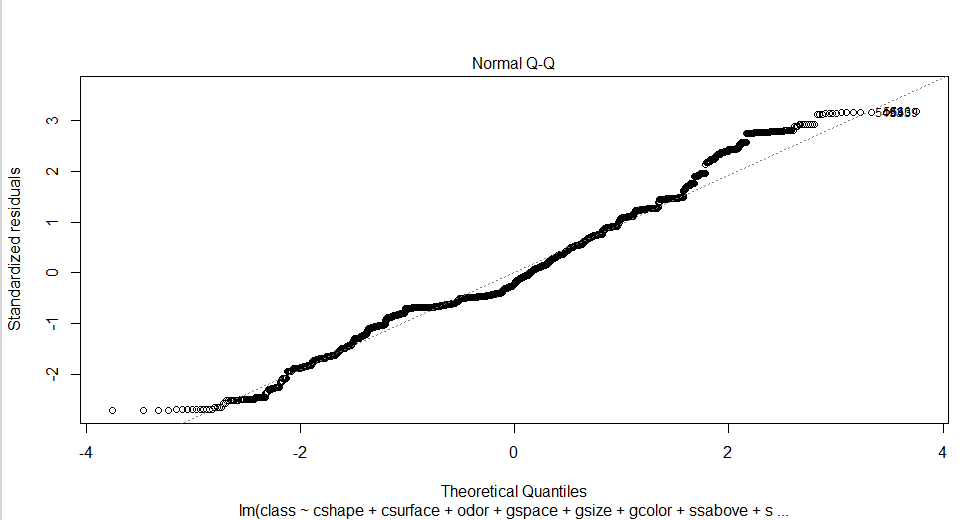


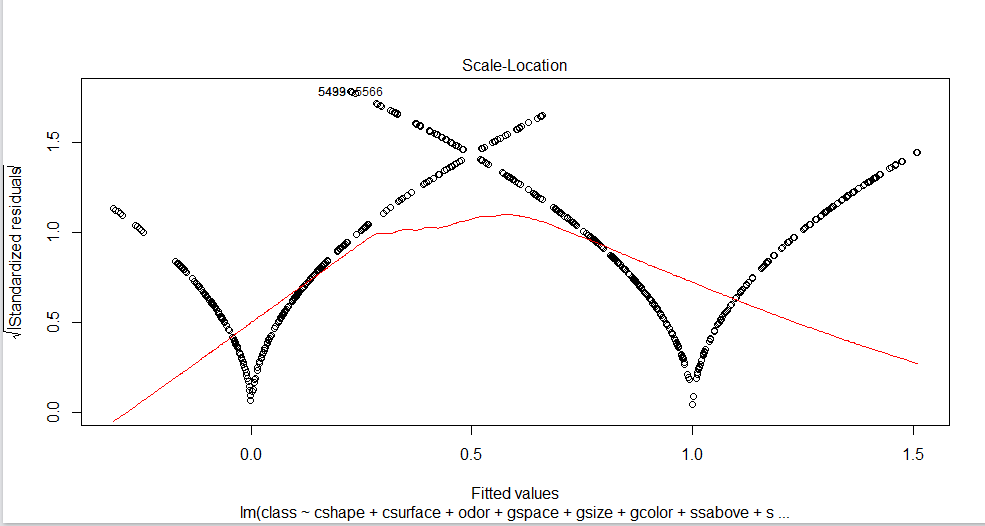


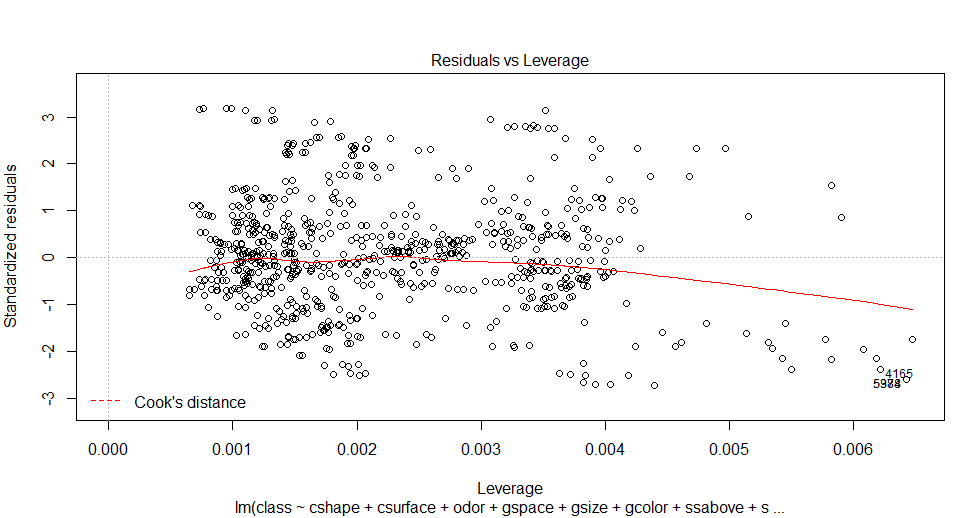


* 1. train70\_new\_lm (Removing meaningless attributes)

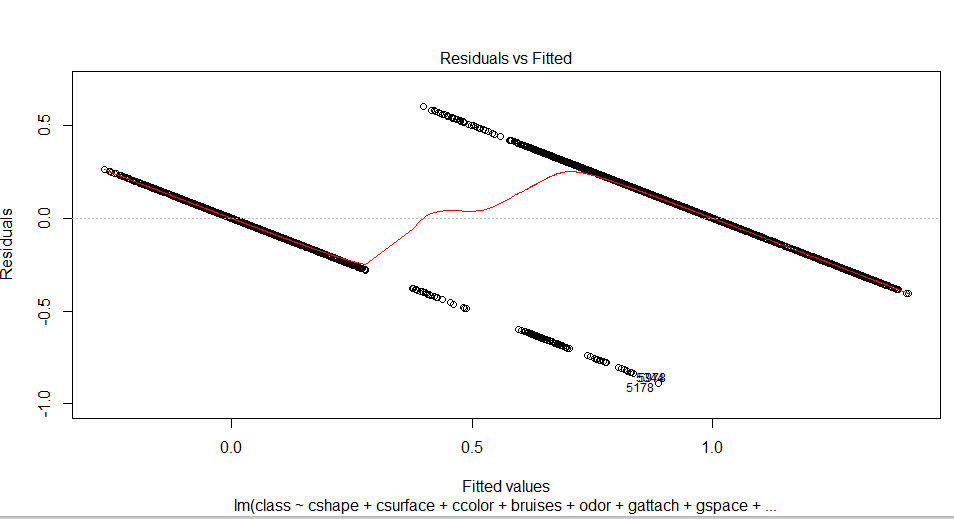


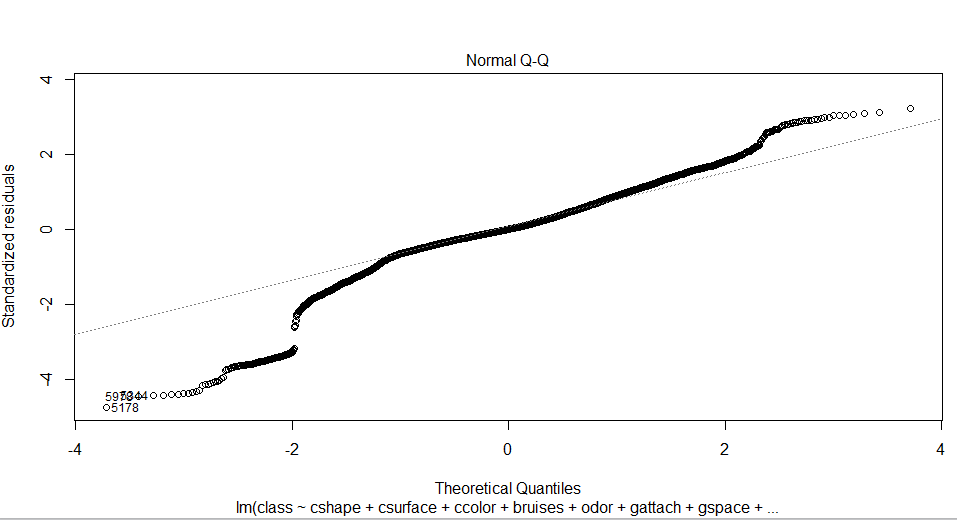


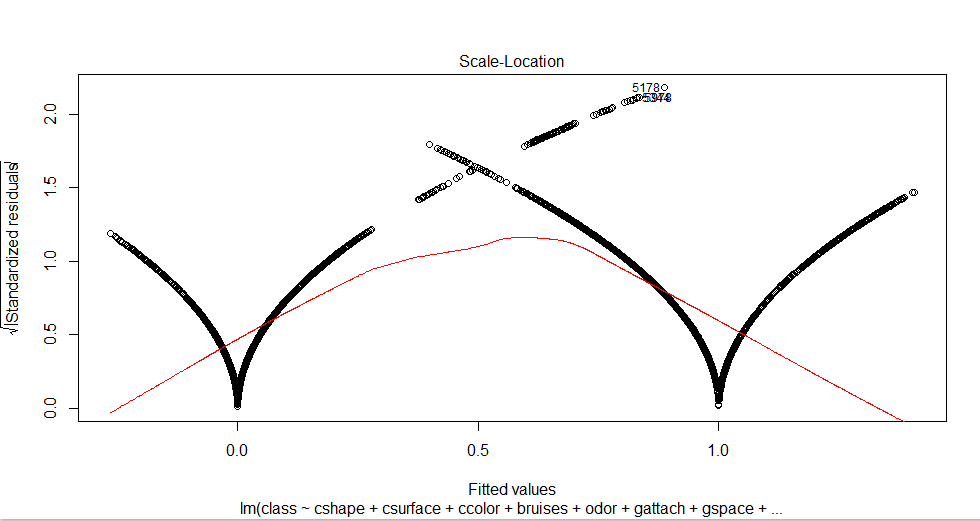


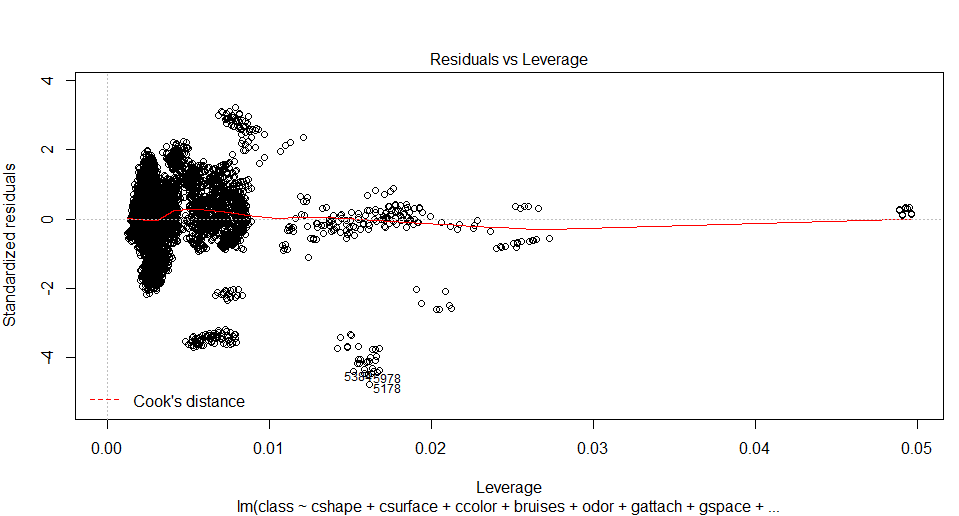


1. train60\_lm(All attributes)

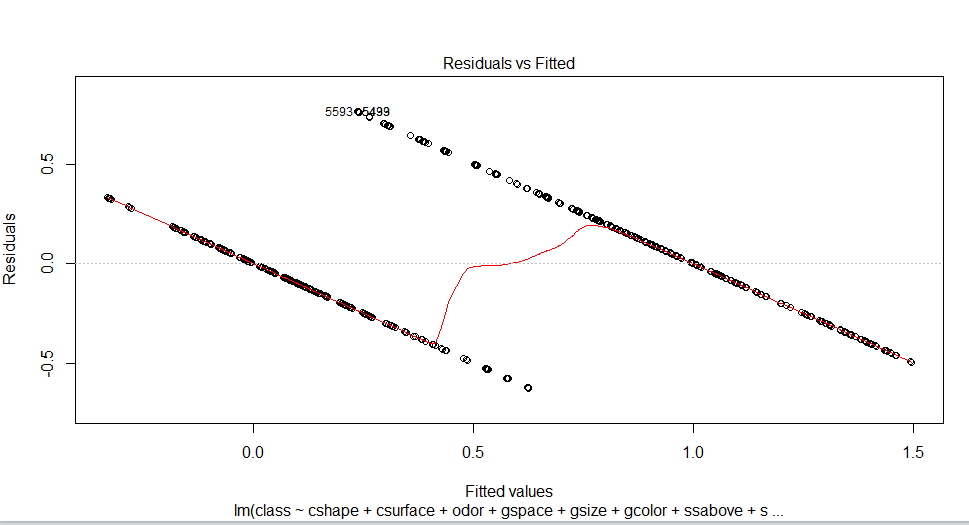


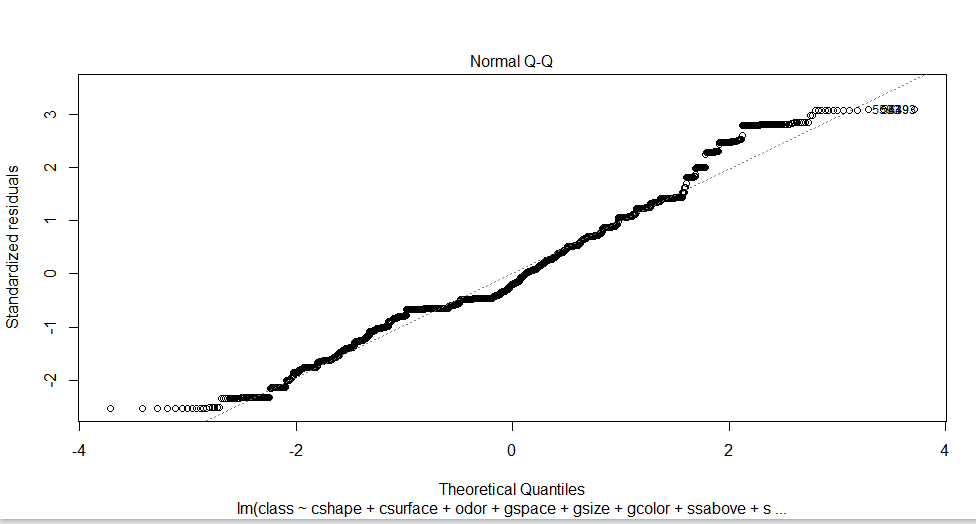


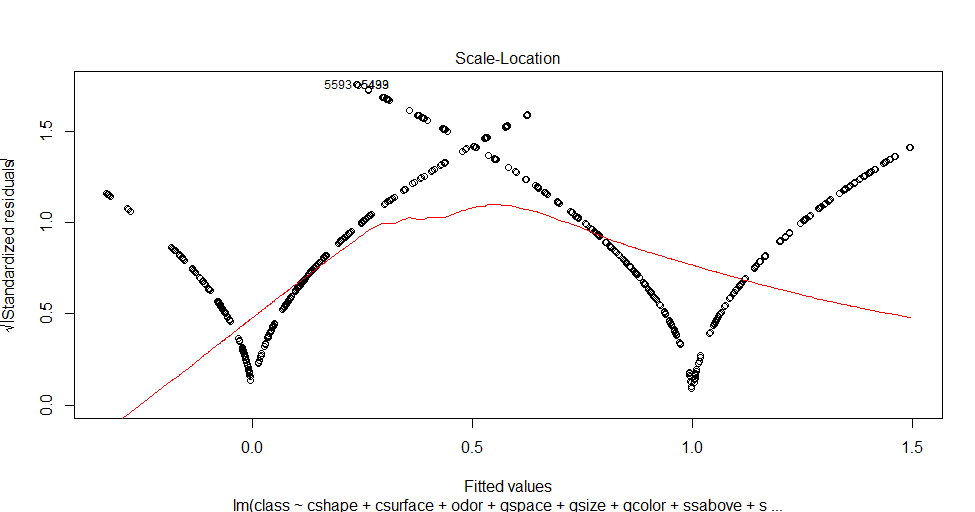


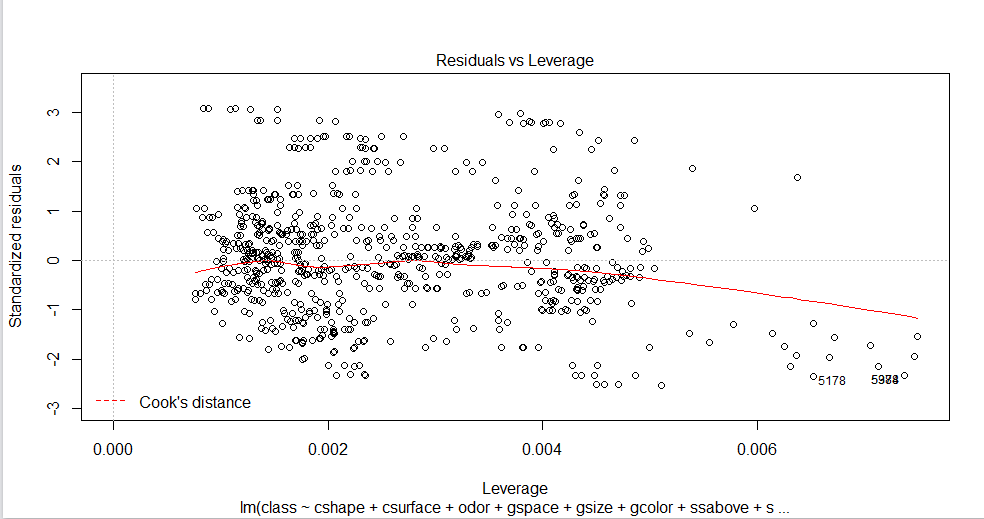


2.1 train60\_new\_lm(remove meaningless attributes)

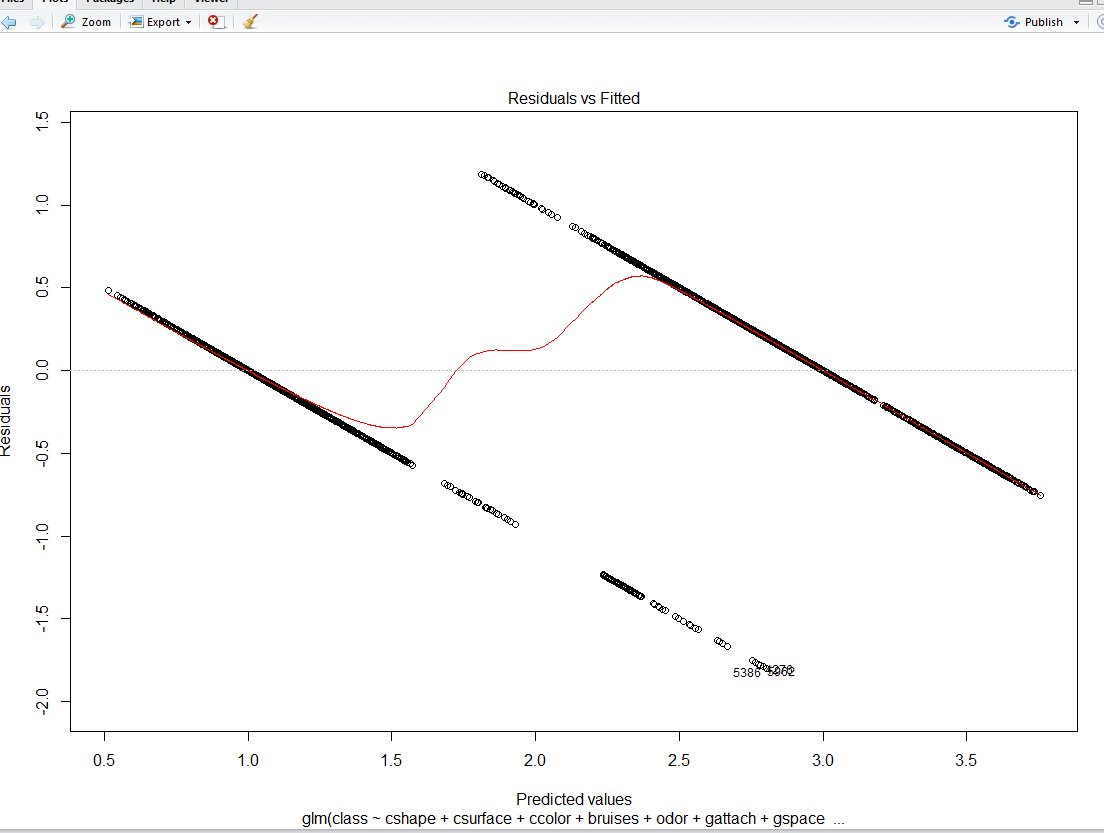


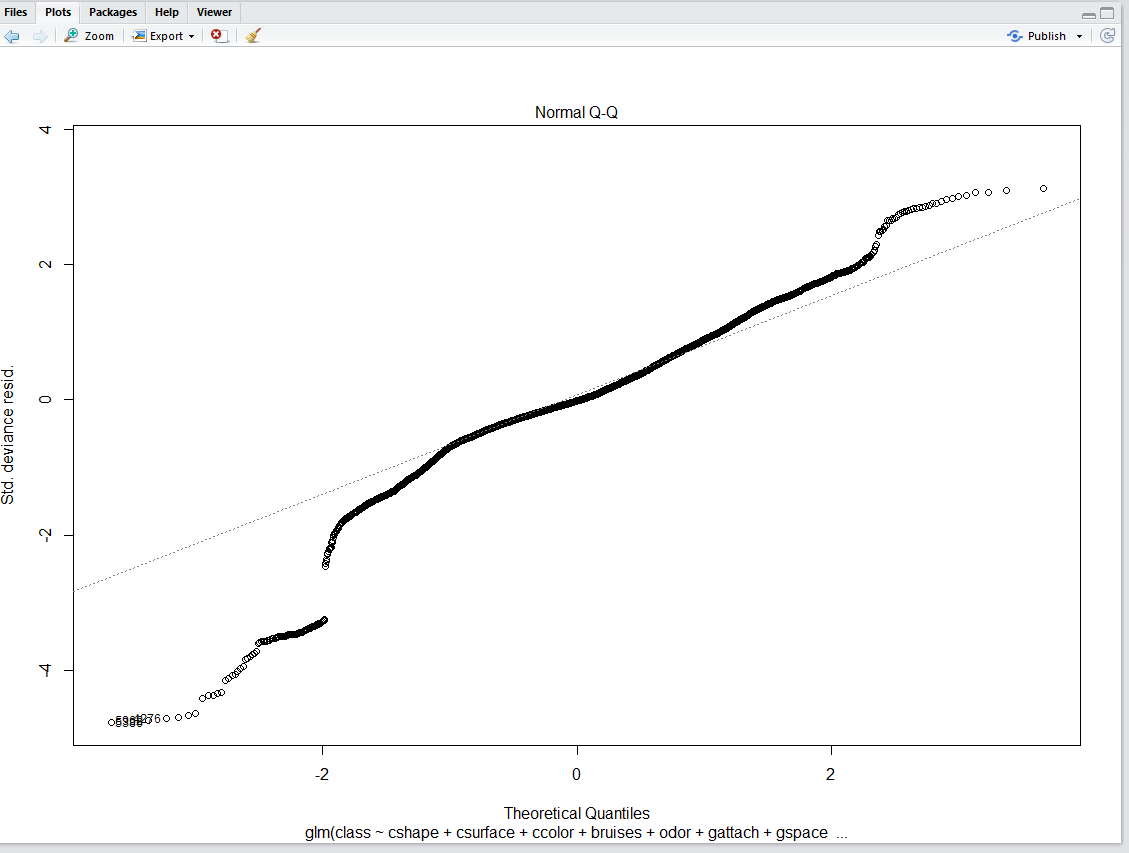


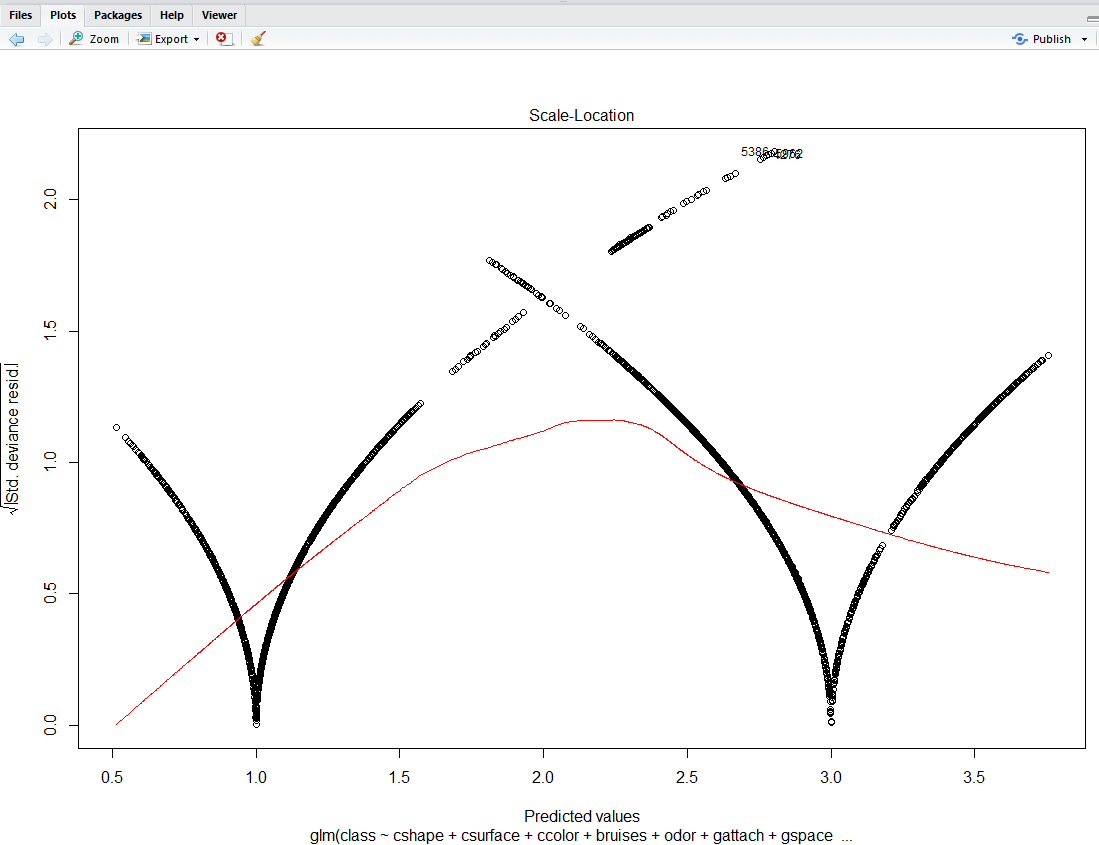


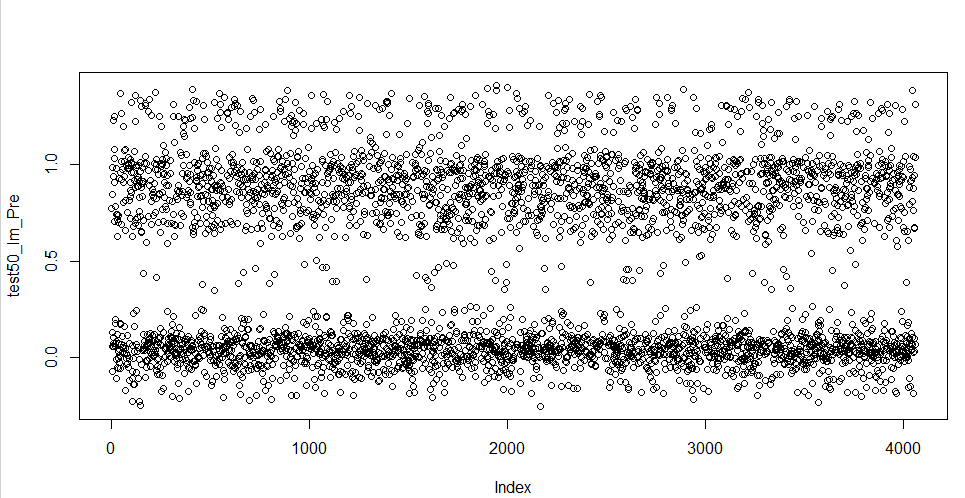


1. train50\_lm(All attributes)

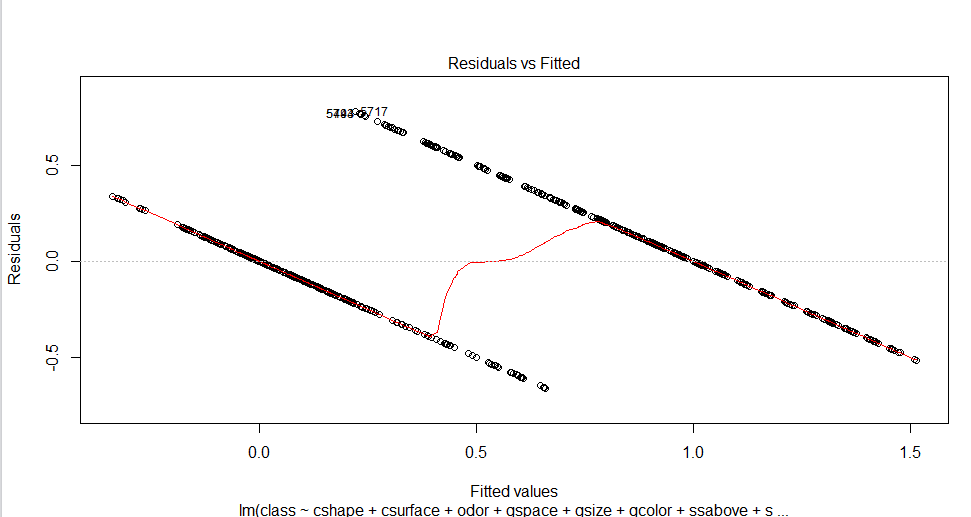


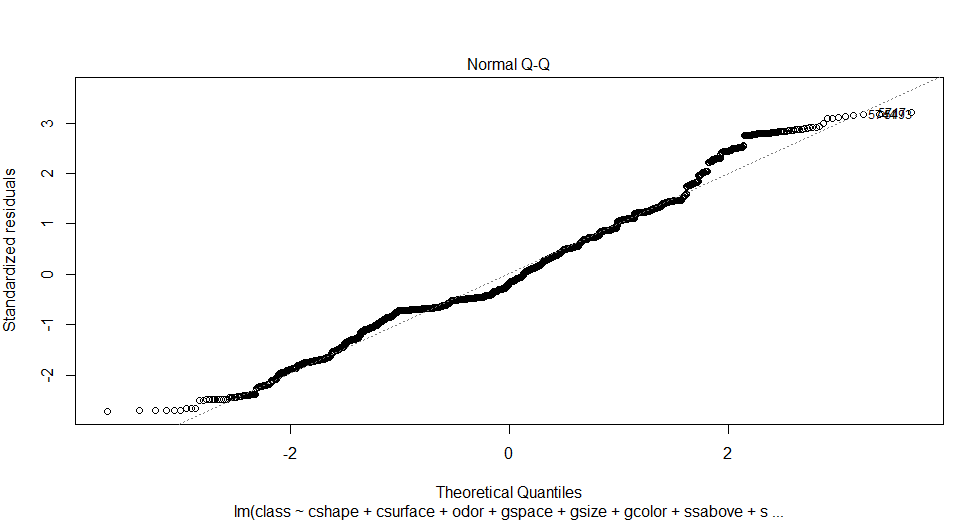


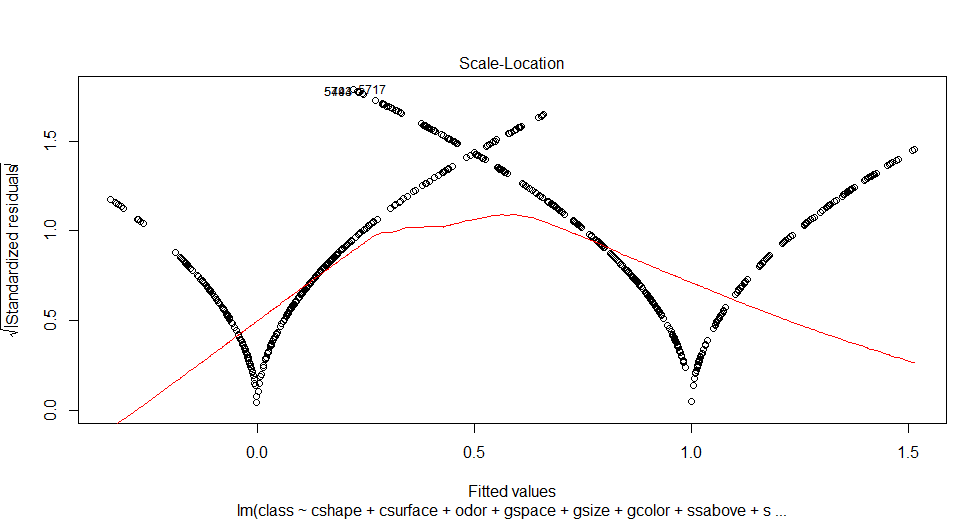


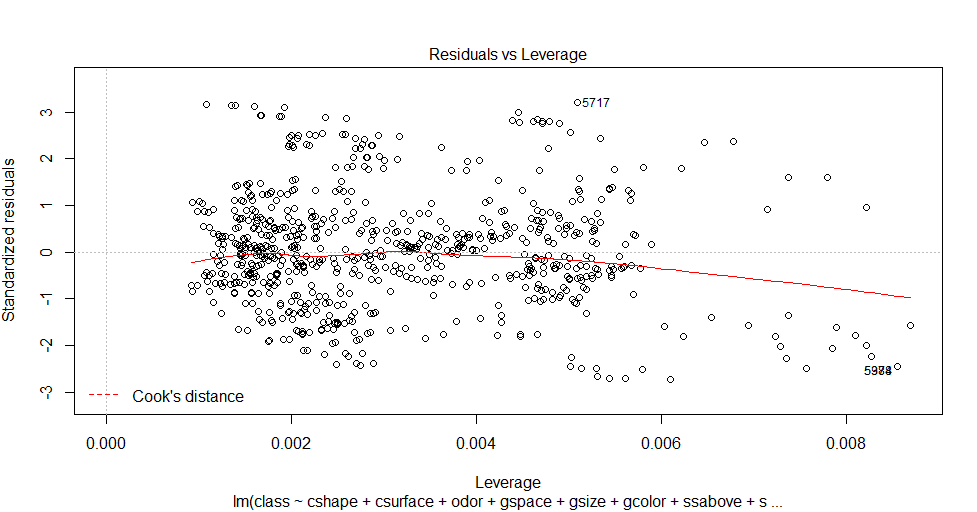


3.1 train50\_new\_lm (removing meaningless attributes)

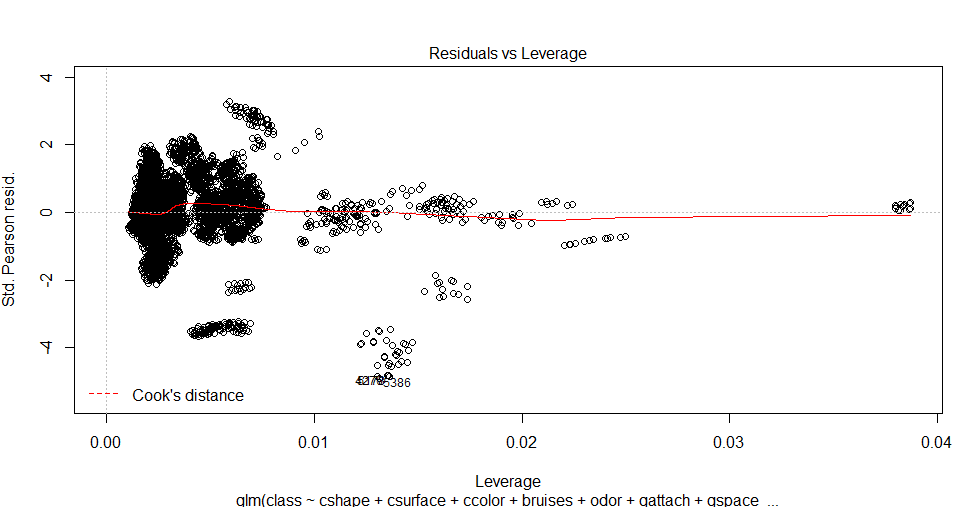




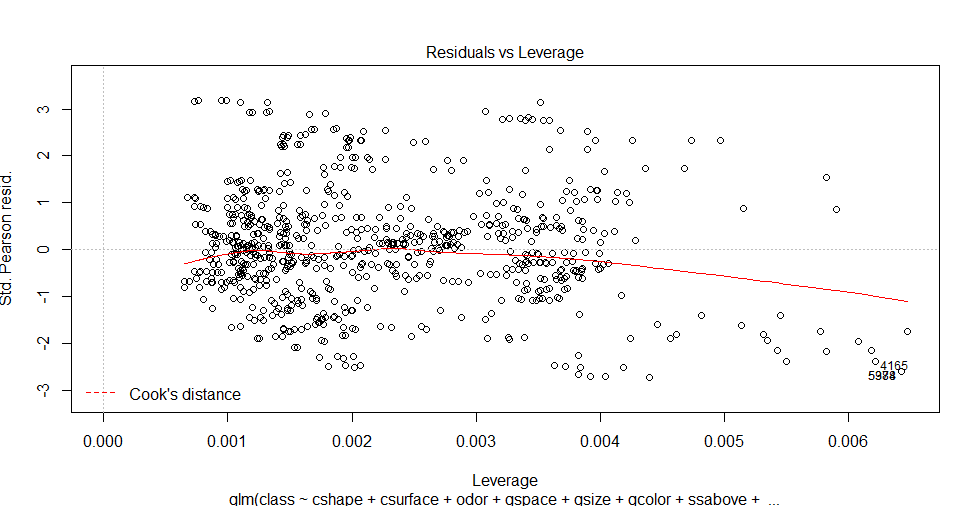




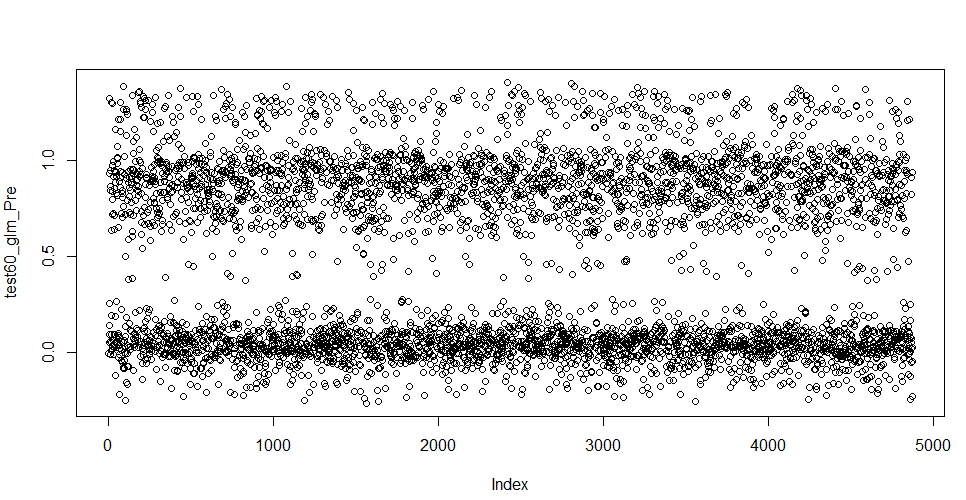
1. train70\_glm(All attributes)



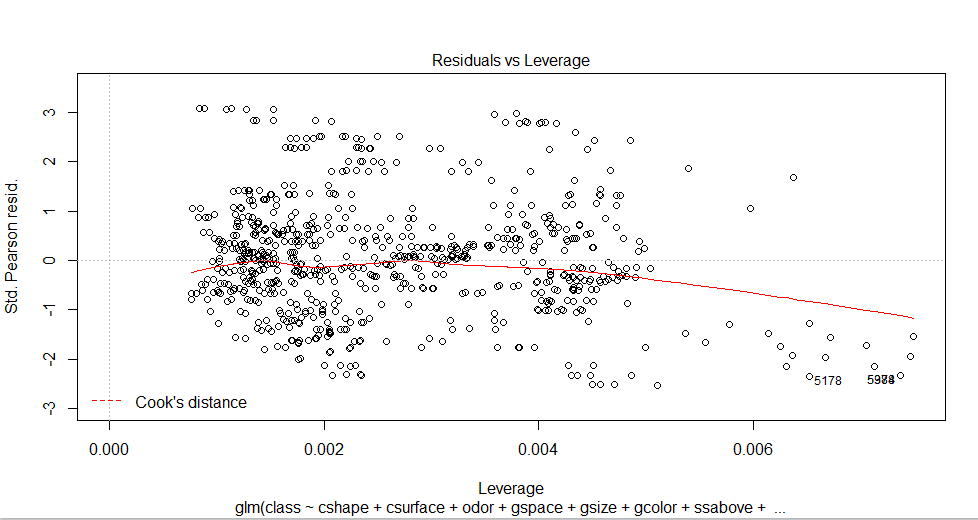
* 1. train70\_new\_glm(removing meaningless attributes)



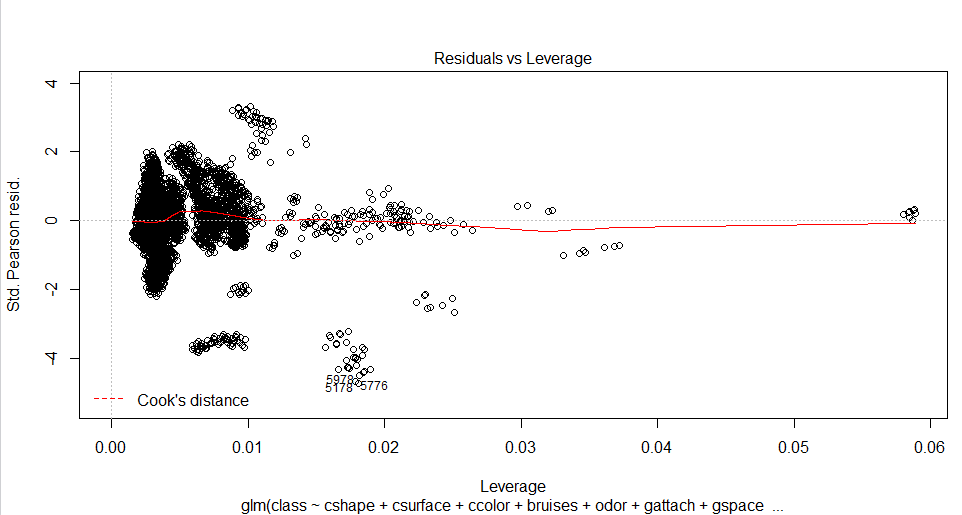
1. train60\_glm(All attributes)



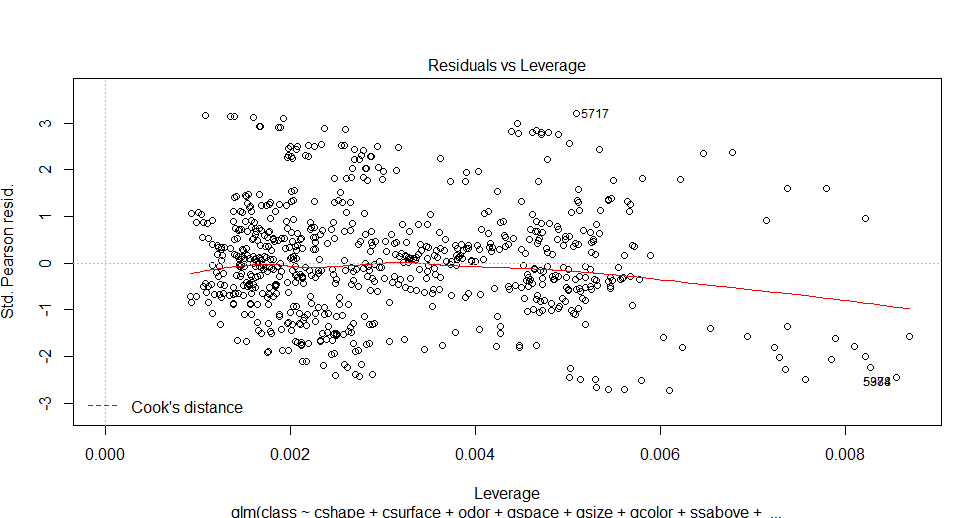
5.1 train60\_new\_glm(removing meaningless attributes)



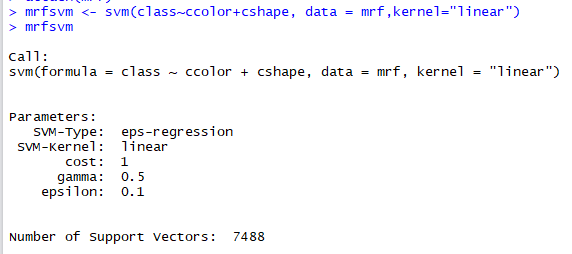
1. train50\_glm(All attributes)

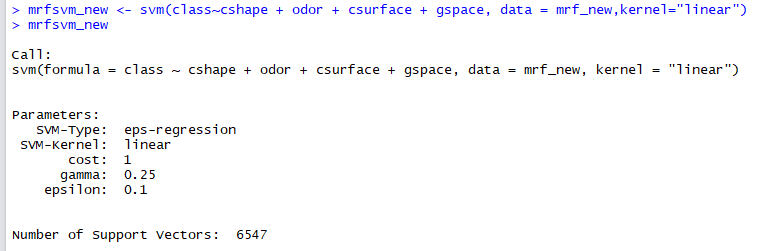


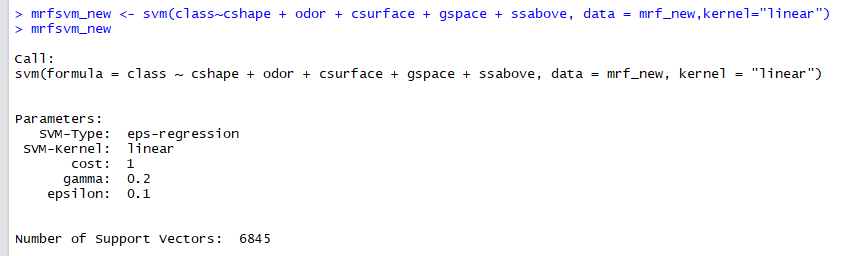
6.1 train50\_new\_glm(removing meaningless attributes)



1. SVM







e. Plots with discussion of results

Lm, GLM, and SVM methods help us understand best. I will introduce them respectively.

LM method, called Levenberg-Marquardt, which can be used to solve the nonlinear least squares problem, and more for curve fitting and other occasions.

The realization of the LM algorithm is not difficult. The key of the algorithm is to use the model function f to do the linear approximation in the field of the parameter p, and to ignore the second-order derivative term, which is transformed into the linear least squares problem. Convergence speed and so on. LM algorithm belongs to a "trust domain method" - the so-called trust region method, here a little explanation: in the optimization algorithm, are required a function of the minimum value, each step iterations, require the objective function value Is the decline of the law, the name of the law, as the name suggests, that is, starting from the initial point, the first assumption that a maximum displacement can be trusted s, and then in the current point as the center, to s for the radius of the region, by looking for an approximation of the objective function(quadratic) of the optimal point to solve the true displacement. After the displacement is obtained, the objective function value is recalculated, and if it declines that the displacement of the objective function value satisfies certain conditions, then the displacement is reliable, then the iteration is continued according to this rule; if it can’t make the objective function value of the decline to meet certain conditions, it should reduce the scope of the trust area, and then solve it again.

In fact, you can find a description of the LM algorithm seen from all the information that can be found, similar to "if the objective function value increases, then adjust a coefficient to continue to solve; if the objective function value decreases, Coefficient and then continue to solve the "iterative process, this process and the above-mentioned trust region method is very similar, so that LM algorithm is a trusted domain method.

Generalized linear model (GLM) is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value.

Generalized linear models includes linear regression, logistic regression, logistic regression and Poisson regression. They proposed an iteratively reweighted least squares method for maximum likelihood estimation of the model parameters. Maximum-likelihood estimation remains popular and is the default method on many statistical computing packages. Other approaches, including Bayesian approaches and least squares fits to variance stabilized responses, have been developed.

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

SVM has three main advantages. First, it can be used for linear / non-linear classification, can also be used for regression, generalization error rate is low, the calculation is not expensive, the results easy to explain; Second, it can solve the problem of machine learning in the case of small sample, can solve the high-dimensional problem can avoid the neural network structure selection and local minimal point problem. Third, SVM is the best off-the-shelf classifier, ready-made means that no modification can be used directly. And can get a lower error rate, SVM can be outside the training set of data points do a good classification decision. On the other hand, it also has this disadvantage. It’s sensitive to parameter selection and the selection of functions, the original classifier is not modified to apply only to the processing of two classification problems.

f. Self-Anlysis

When we deal with a data set with multiple features, the first thing we should try is to do data reduction by using PCA (principle component analysis) method to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables.

“PCA is mostly used as a tool in exploratory data analysis and for making predictive models. It's often used to visualize genetic distance and relatedness between populations. PCA can be done by eigenvalue decomposition of a data covariance (or correlation) matrix or singular value decomposition of a data matrix, usually after mean centering (and normalizing or using Z-sores) the data matrix for each attribute. The results of a PCA are usually discussed in terms of *component scores*, sometimes called *factor scores* (the transformed variable values corresponding to a particular data point), and *loadings* (the weight by which each standardized original variable should be multiplied to get the component score).

PCA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data. If a multivariate dataset is visualised as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA can supply the user with a lower-dimensional picture, a projection of this object when viewed from its most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.”

As for a data set with much complexity, a certain model such as SVM, LM, GLM can’t give a perfect result. Thus, we may need to attempt several potential models. Model selection is pretty important. It determines the accuracy of our data analysis. So, we need to compare different methods applied in one data set to choose a better one.

As for a data set with attributes expressed in letter values, there is no doubt that we need to convert letter values to numbers, because the clustering analysis methods don’t accept letter. If not, the clustering methods will not work. In the clustering algorithms, the comparison is between records, rather than between attributes.

By doing this job, we find that analyzing data is very interesting, and we need to focus, and devote much more energy to master the techniques of big data.