

**Diamonds**

**Data Analysis Report**

**R Language**

Sameer Afzal | 1000053143 | Data SCIENCE

Data Analysis and Statistical Learning

Professor Antonio Punzo

2023

Contents

[1. Introduction 3](#_Toc126104062)

[2. Preliminary Analysis 4](#_Toc126104063)

[2.1. Loading Data set 4](#_Toc126104064)

[2.2. Understanding of dataset 4](#_Toc126104065)

[2.3. Peek at your Data 5](#_Toc126104066)

[3. Univariate Data Analysis 8](#_Toc126104067)

[3.1. Cut 8](#_Toc126104068)

[3.2. Carat 9](#_Toc126104069)

[3.3. Depth 16](#_Toc126104070)

[3.4. Table 23](#_Toc126104071)

[3.5. X 30](#_Toc126104072)

[3.6. Y 37](#_Toc126104073)

[3.7. Z 44](#_Toc126104074)

[3.8. Price 51](#_Toc126104075)

[4. PRINCIPAL COMPONENT ANALYSIS 58](#_Toc126104076)

[4.1. Methods for selecting PC 62](#_Toc126104077)

[4.1.1. Cumulative proportion of variance explained (**CPVE**) 62](#_Toc126104078)

[4.1.2. Kaiser’s rule 62](#_Toc126104079)

[4.1.3. Scree plot 63](#_Toc126104080)

[4.2. PCA results 63](#_Toc126104081)

[5. CLUSTER ANALYSIS 64](#_Toc126104082)

[5.1. Euclidean distance 64](#_Toc126104083)

[5.2. Manhattan distance 65](#_Toc126104084)

[5.3. Agglomerative Hierarchical Clustering 66](#_Toc126104085)

[5.3.1. A.H.C. based on Euclidean distance and Ward’s linkage method 66](#_Toc126104086)

[5.3.2. A.H.C. based on Euclidean distance and Average linkage method 69](#_Toc126104087)

[5.3.3. A.H.C. based on Euclidean distance and Single linkage method 72](#_Toc126104088)

[5.3.4. A.H.C. based on Manhattan distance and Ward’s linkage method 75](#_Toc126104089)

[5.3.5. A.H.C. based on Manhattan distance and Average linkage method 78](#_Toc126104090)

[5.3.6. A.H.C. based on Manhattan distance and Single linkage method 81](#_Toc126104091)

[5.3.7. Concluding best algorithm 83](#_Toc126104092)

[5.4. Partitioning Clustering 84](#_Toc126104093)

[5.4.1. K-Means: k=2: 84](#_Toc126104094)

[5.4.2. K-Mean: k=4 85](#_Toc126104095)

[5.4.3. K-Medoids: k=2 87](#_Toc126104096)

[5.5. Cluster Validation 88](#_Toc126104097)

[5.5.1. Hopkins Statistic 88](#_Toc126104098)

[5.5.2. VAT Algorithm 89](#_Toc126104099)

[5.6. Cluster Statistics 92](#_Toc126104100)

[5.6.1. Silhouette Width 92](#_Toc126104101)

[5.6.2. Dunn Index 93](#_Toc126104102)

[5.7. Choosing the best clustering algorithms. 94](#_Toc126104103)

[5.7.1. Internal Measures 94](#_Toc126104104)

[5.7.2. Stability Measures 95](#_Toc126104105)

[5.8. Model Base Clustering 96](#_Toc126104106)

# Introduction

This classic dataset contains the prices and other attributes of 100 diamonds. It's a great dataset for beginners learning to work with data analysis. There are some factors that affects diamond price. This dataset contains diamond’s information like carat, cut, depth, table, x(length), y (width), z(depth).

This report is divided into 3 major parts.

* Univariate analysis
* PCA (principal component analysis)
* Cluster Analysis

# Preliminary Analysis

## **Loading Data set**

There are some data sets that are already pre-installed in R. Here, we shall be using external **Diamonds** data set that not comes built-in R Package.

While using any external data source, we can use the read command to load the files (Excel, CSV, HTML and text files etc.)

This data set is also available at [Kaggle](https://www.kaggle.com/datasets/shivam2503/diamonds). You may download the data set, both train and test files. In this analysis, we’d be just using the train data set.

## **Understanding of dataset**

We have used the diamond data set that contains records of 100 diamonds with different attributes. Below is a brief description of the 8 variables in the data set:

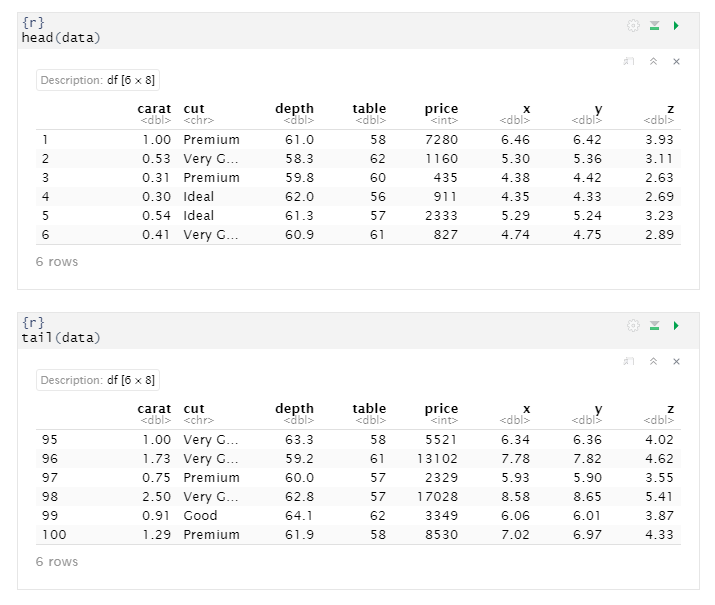
* **Price**: price in US dollars
* **Carat:** weight of the diamond
* **Cut:** quality of the cut
* **depth:** total depth percentage = z / mean (x, y) = 2 \* z / (x + y)
* **table:** width of top of diamond relative to widest point
* **x**: length in mm
* **y:** width in mm
* **z:** depth in mm

## Peek at your Data

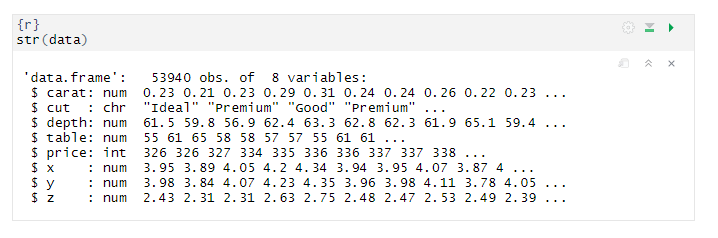
Before we begin working on the dataset, let’s have a good look at the raw data.

names() : This helps us in checking out all the variables in the data set.

In order to have a quick look at the data, we often use the head()/tail().



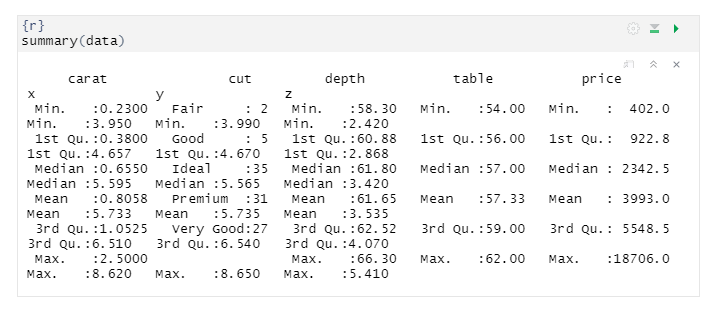
str() : This helps in understanding the structure of the data set, data type of each attribute and number of rows and columns present in the data.



summary(): is one of the most important functions that help in summarizing each attribute in the dataset. It gives a set of descriptive statistics, depending on the type of variable:

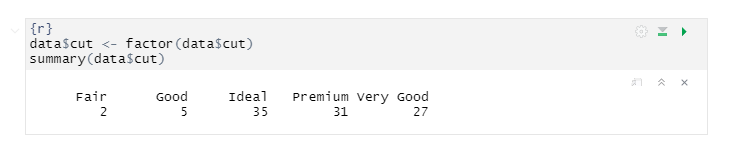
* In case of a Numerical Variable -> Gives Mean, Median, Mode, Range and Quartiles.
* In case of a Factor Variable -> Gives a table with the frequencies.
* In case of Factor + Numerical Variables -> Gives the number of missing values.
* In case of character variables -> Gives the length and the class.

In case we just need the summary statistic for a particular variable in the dataset, we can use summary(datasetName$VariableName)



as.factor(dataset$ColumnName)

There are times when some of the variables in the data set are factors but might get interpreted as numeric. For example, the cuts tales the values Fair, Good, Ideal, Premium and Very Good however, we know that these are not to be considered as numeric, as these are just types. In order to such variables treated as factors and not as numbers we need explicitly convert them to factors using the function as.factor()

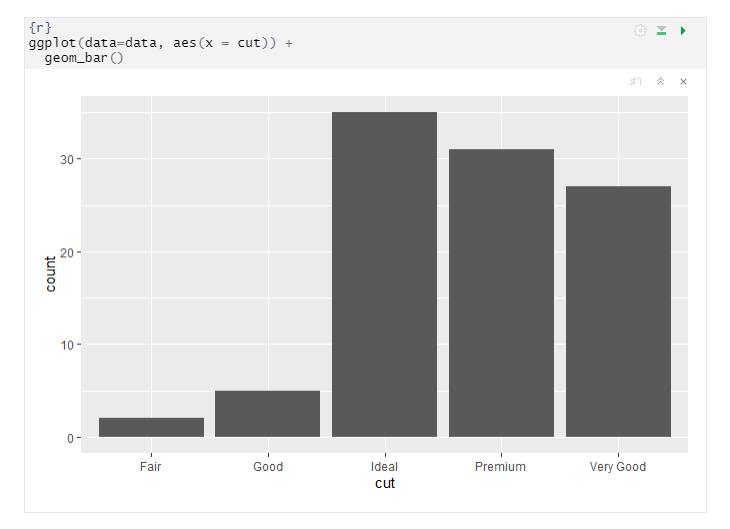


# Univariate Data Analysis

Univariate analysis is basically the simplest form to analyze data. Uni means one and this means that the data has only one kind of variable. The major reason for univariate analysis is to use the data to describe. Let’s do univariate analysis on each variable in depth.

## Cut

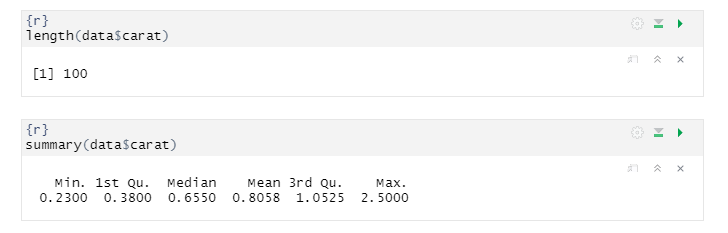
Cut is a categorical variable that can take 5 types Fair, Ideal, Good, Very Good and premium.



The above graph shows the counts of the diamonds with different type of cuts. So, based on the plot and summary, we can conclude that our data is unbalanced and the ideal type of cuts are more than others.

## Carat

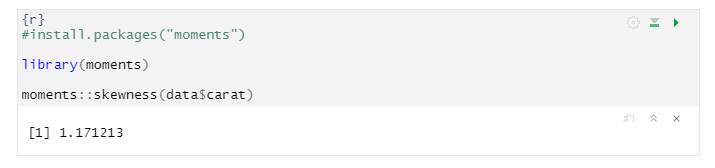
This is a numeric continuous variable. The basic statistical values are mentioned below.



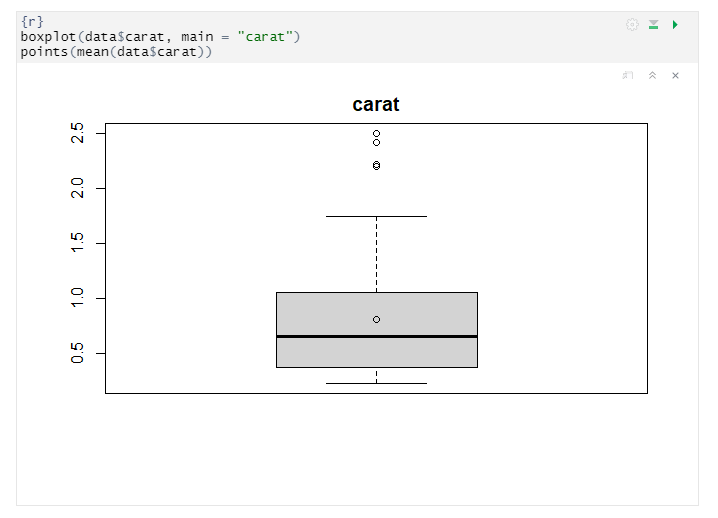
There is total 100 diamonds and weight range are between 0.23 to 2.5



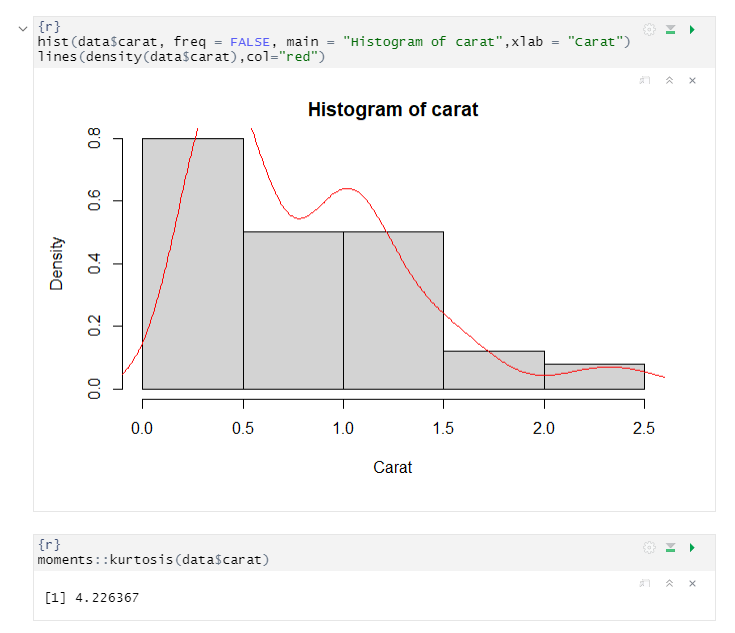
From above statistical computation we can observed that the mean and median are not identical, therefore the distribution is asymmetrical and skewed. And as mean is greater than median, hence the distribution should be positively skewed.



As the computed result of skewed is positive. Hence our prediction was correct.

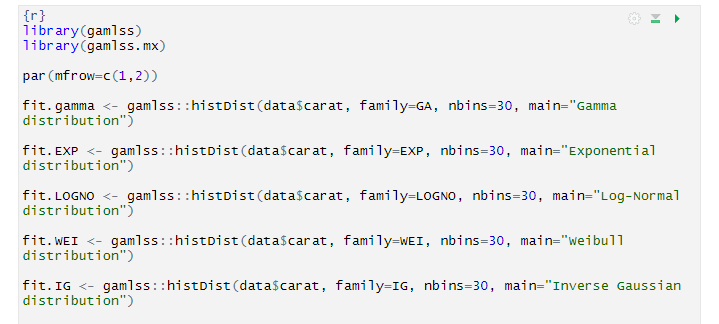


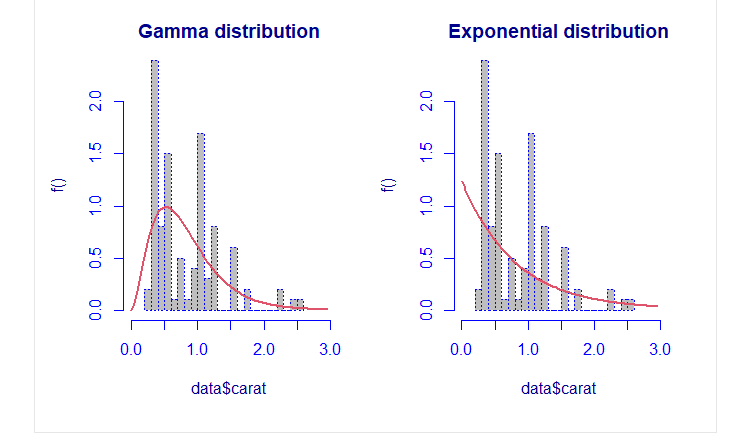
While reviewing the box plot, we can observe that there are outliers in this as we see some dots lie outside the upper whiskers of the boxplot, which means that there are some observations that is numerically distant from the rest of the data. Outliers are the data points located outside the whiskers of the boxplot. Now we can plot the histogram and look for the model that fits better the distribution

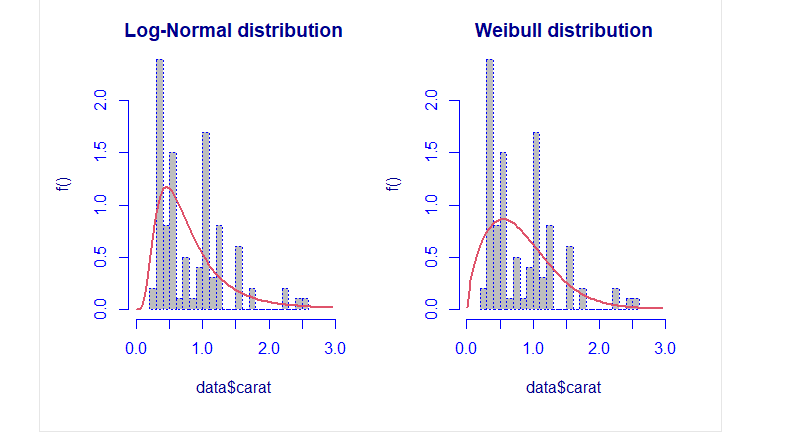


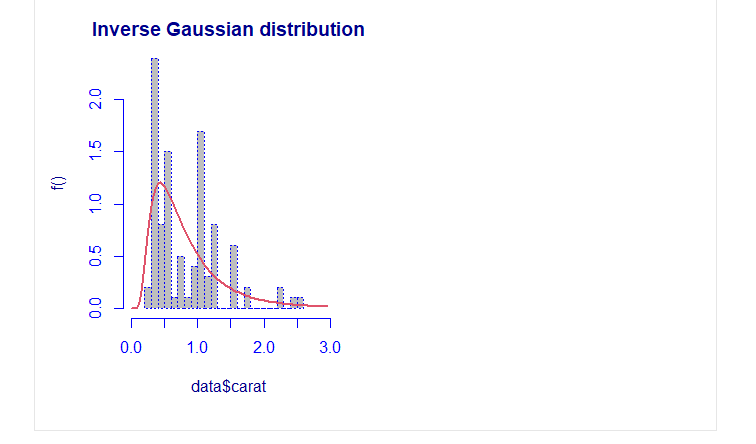
In the above boxplot, we can see so many outliers, then we see the histogram and the red line and then compute the kurtosis. kurtosis (4.2) indicates that the distribution has more outliers than a normal distribution it is also known as leptokurtic.

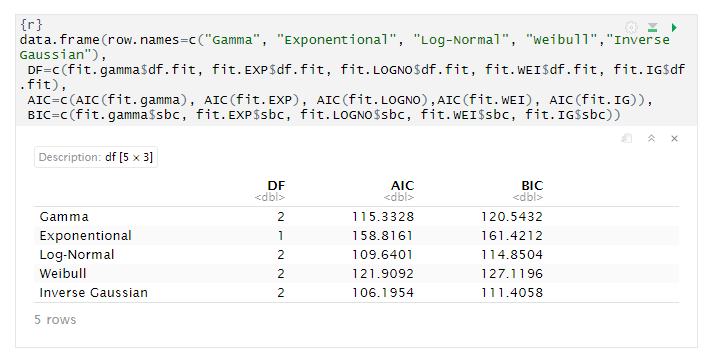
Data fitting for Carats, BIC and AIC are estimated in order to evaluate the fit of the distribution.







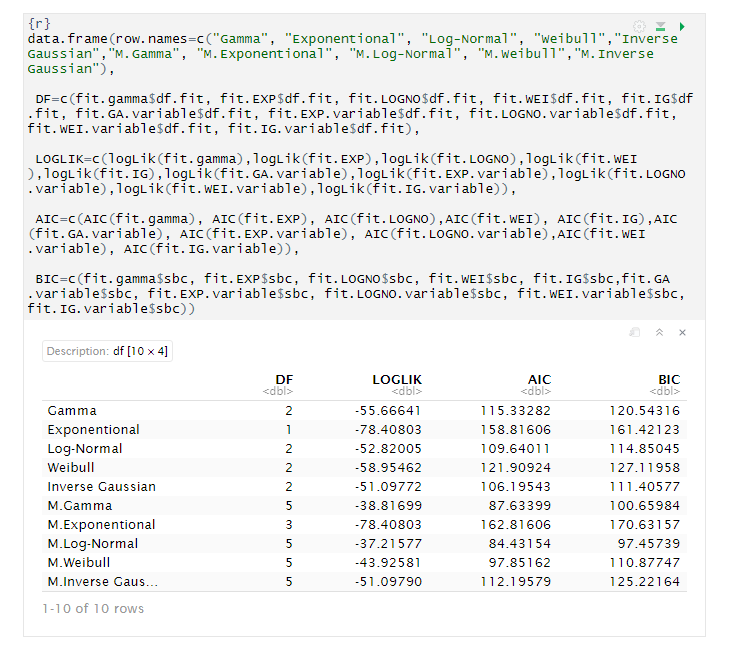




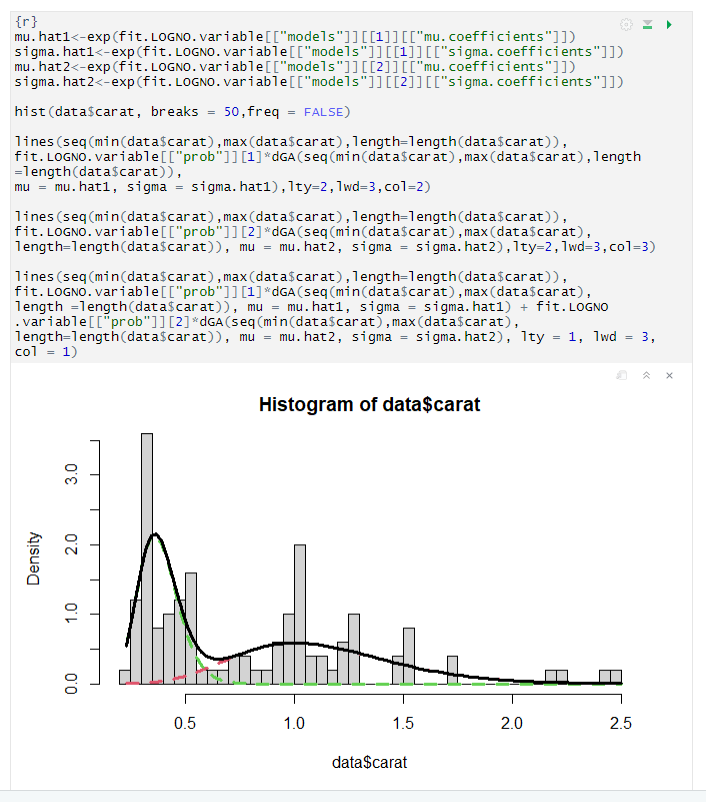
we can observe in the above summary that the model which has small value in AIC and BIC is “Inverse Gaussian distribution”.

The mixture of distribution: It is possible to compute a mixture of two distributions to find the best mixture, the algorithm is repeated five times.

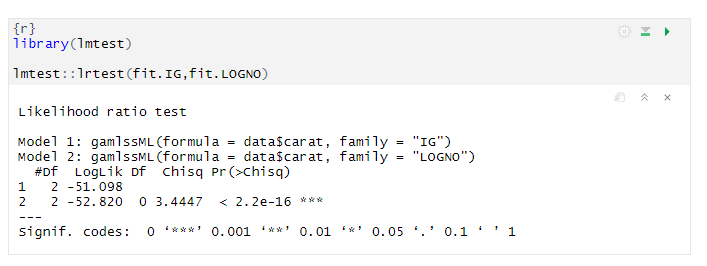




In the first column, the family with “M.” at the start indicate that it is a mixture. In the second column, it’s possible to look at the number of parameters. The best fitting is performed through a mixture distribution with the Log Normal family.



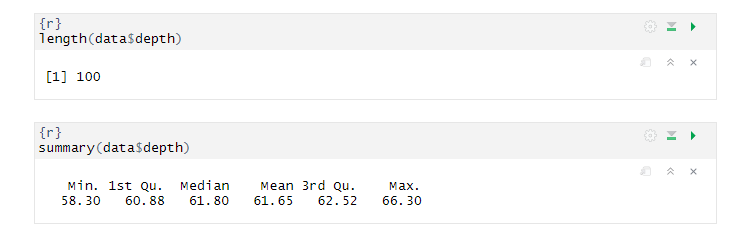
We could also compare two nested models by means of the Likelihood-Ratio Test, for example



The null hypothesis would be rejected at nearly every significance level. Thus, we know that we should definitely use the inverse Gaussian model as it increases the accuracy of our model by a substantial amount in terms of AIC and BIC values

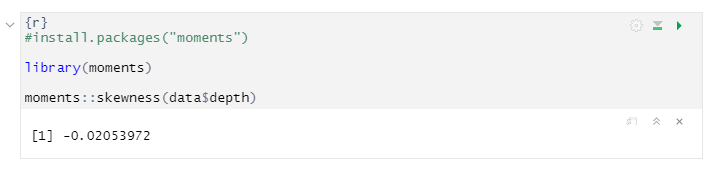
## Depth

This is a numeric continuous variable. Total depth is calculated from: (2 \* z / (x + y)). x, y and z are the other variables in this dataset. The basic statistical values are mentioned below.

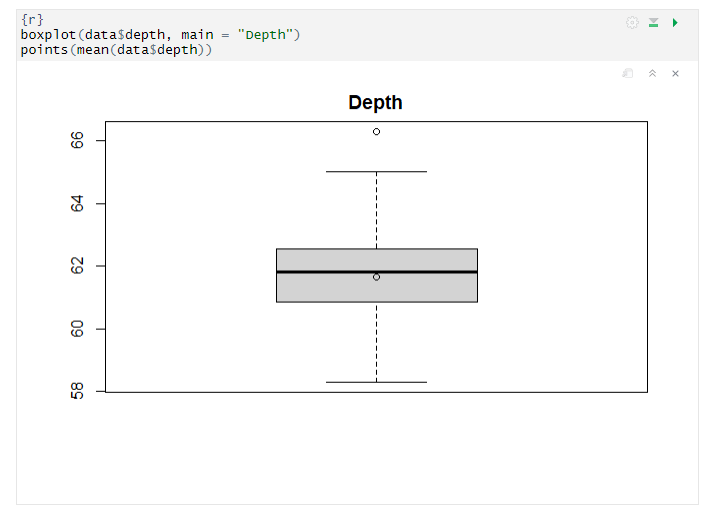


There is total 100 diamonds and the total depth range are between 58.3 to 66.3

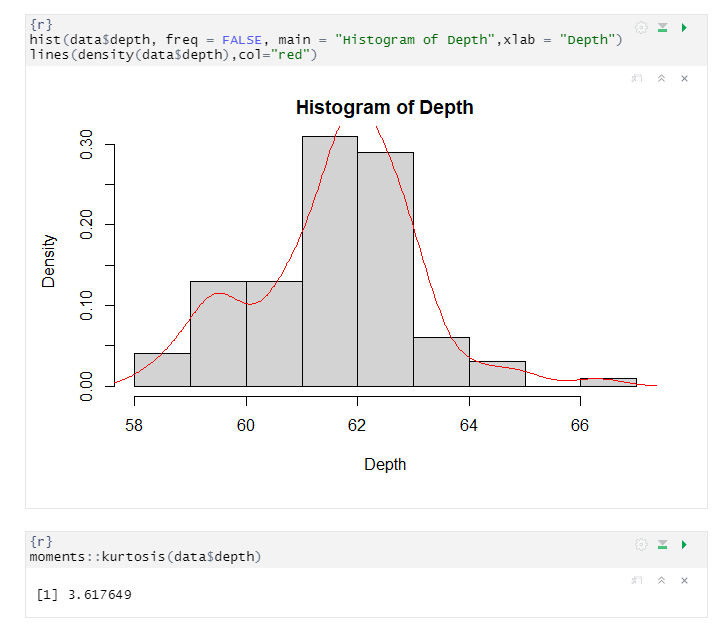
From above statistical computation we can observed that the mean and median are not identical, therefore the distribution is asymmetrical and skewed. And as median is greater than mean, hence the distribution should be negatively skewed.



As the computed result of skewed is negative. Hence our prediction was correct.



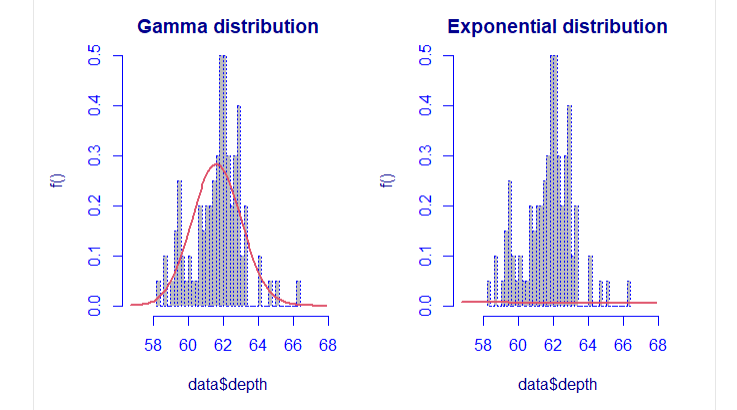
While reviewing the box plot, we can observe that there is only one outlier in this as we see one dot lie outside the upper whiskers of the boxplot. Outliers are the data points located outside the whiskers of the boxplot. Now we can plot the histogram and look for the model that fits better the distribution

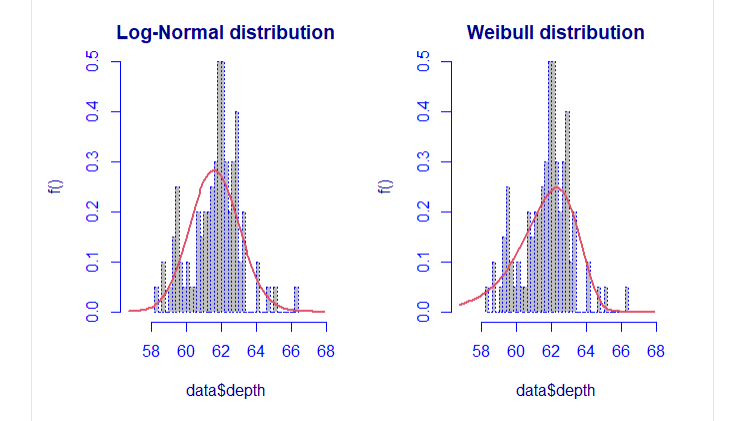


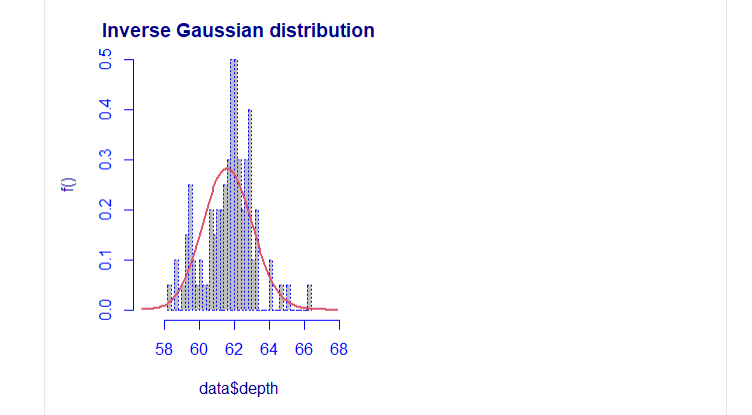
In the above box plot, we can see outlier, then we see the histogram and the red line and then compute the kurtosis. kurtosis (3.6) indicates that the distribution has some outliers than a normal distribution it is also known as leptokurtic.

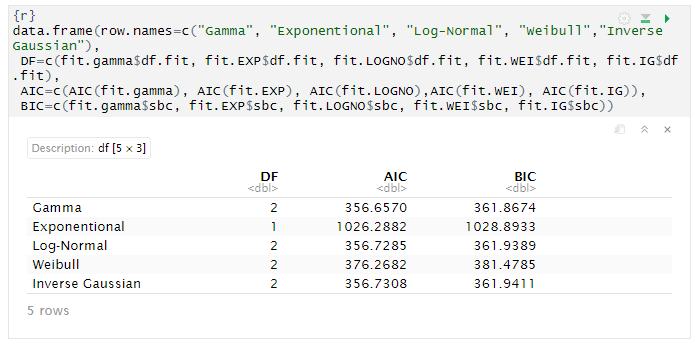
Data fitting for Total Depth, BIC and AIC are estimated in order to evaluate the fit of the distribution.





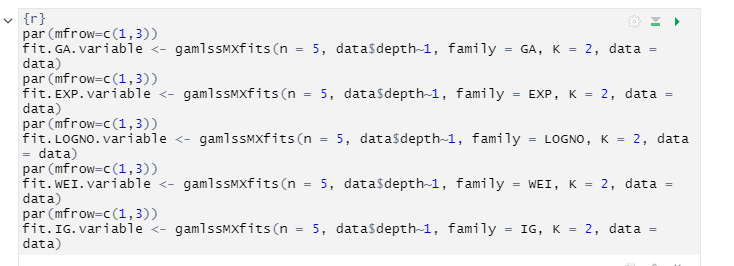


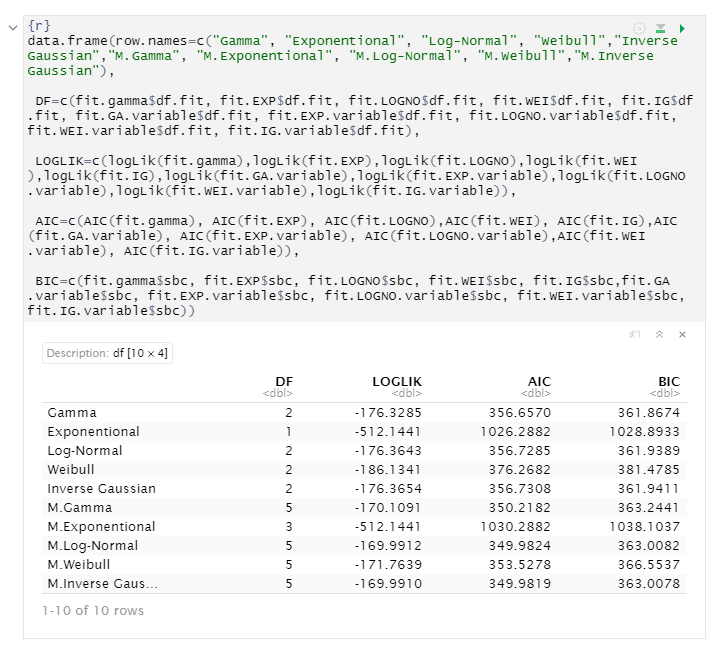




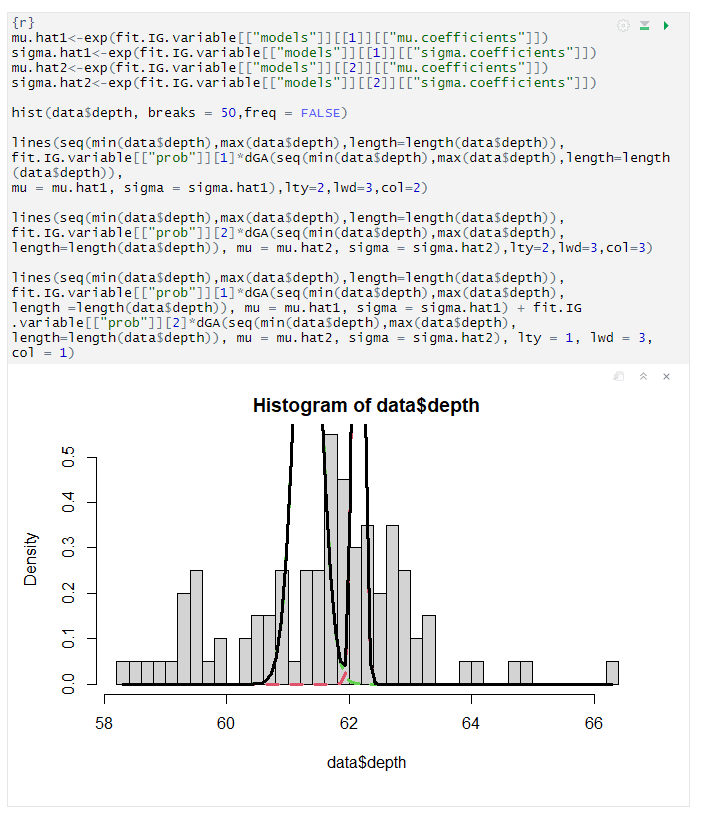
we can observe in the above summary that the model which has small value in AIC and BIC is “Gamma distribution”.

The mixture of distribution: It is possible to compute a mixture of two distributions to find the best mixture, the algorithm is repeated five times.



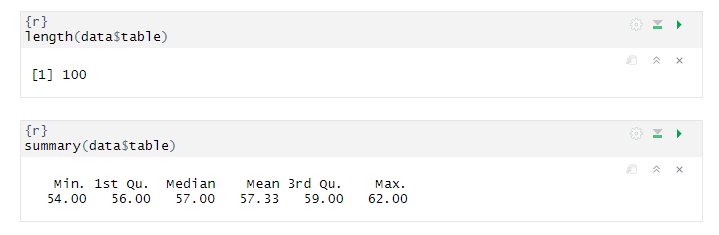


In the first column, the family with “M.” at the start indicate that it is a mixture. In the second column, it’s possible to look at the number of parameters. The best fitting is performed through a mixture distribution with the Inverse Gaussian family.



## Table

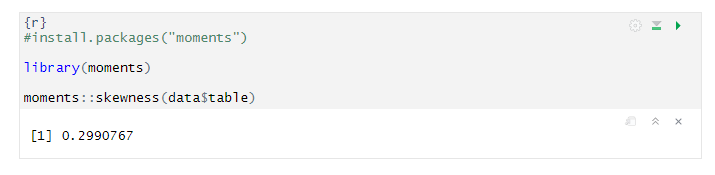
This is a numeric continuous variable. Table is the width of top of diamond relative to widest point. The basic statistical values are mentioned below.



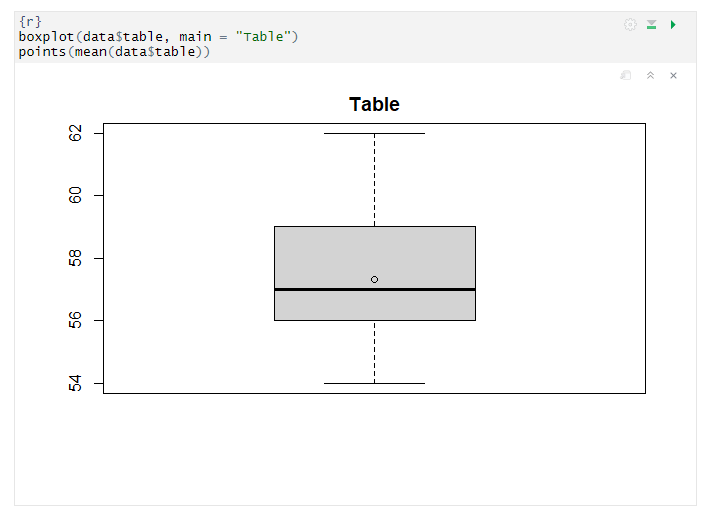
There is total 100 diamonds and the table range are between 54 to 62.



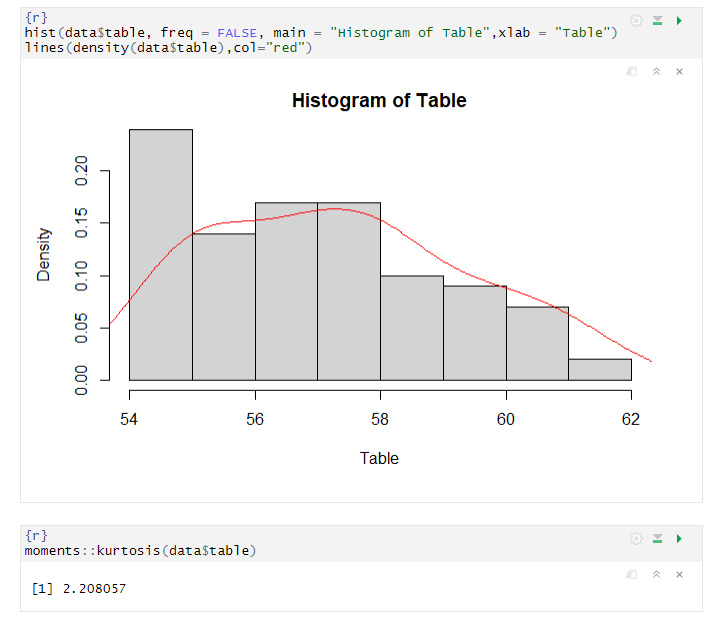
From above statistical computation we can observed that the mean and median are not identical, therefore the distribution is asymmetrical and skewed. And as mean is greater than median, hence the distribution should be positively skewed.



As the computed result of skewed is positive. Hence our prediction was correct.

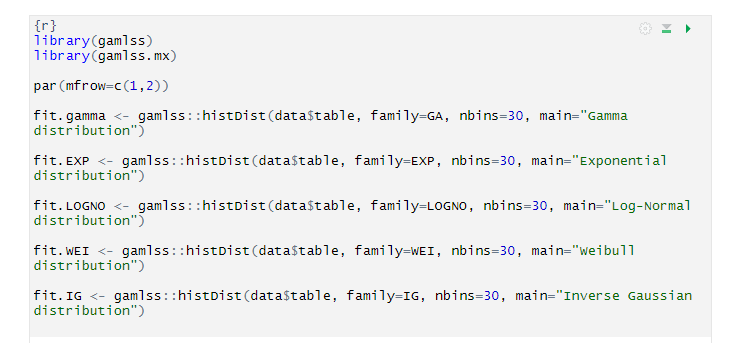


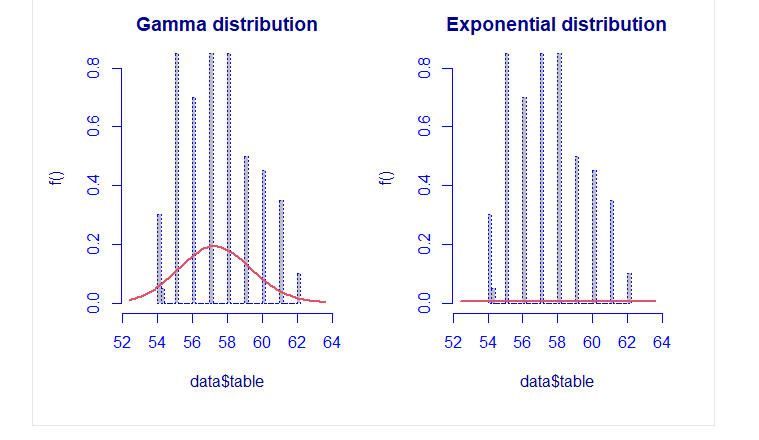
While reviewing the box plot, we can observe that there no outliers in this. Now we can plot the histogram and look for the model that fits better the distribution

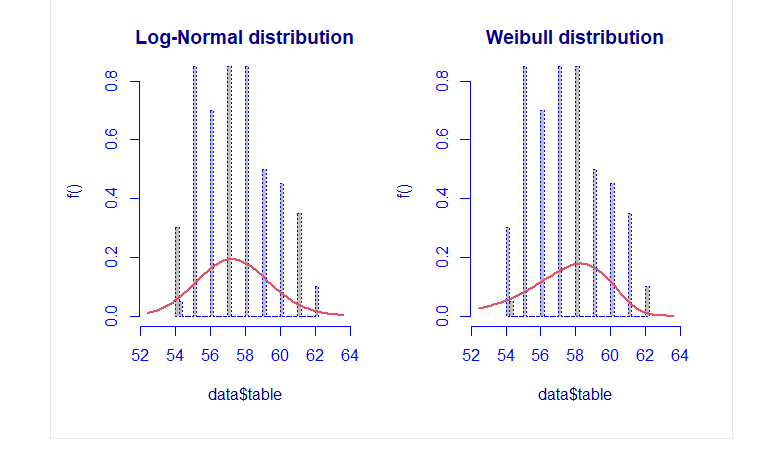


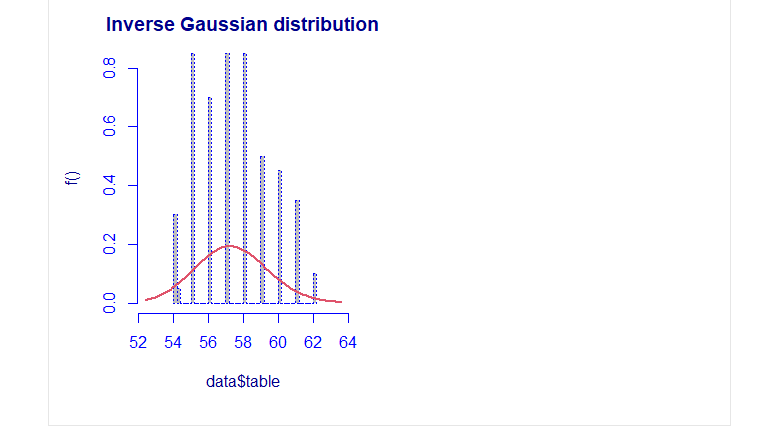
In the above box plot, we can see no outlier, then we see the histogram and the red line and then compute the kurtosis. kurtosis (2.2) indicates that your distribution is flatter than a normal curve.

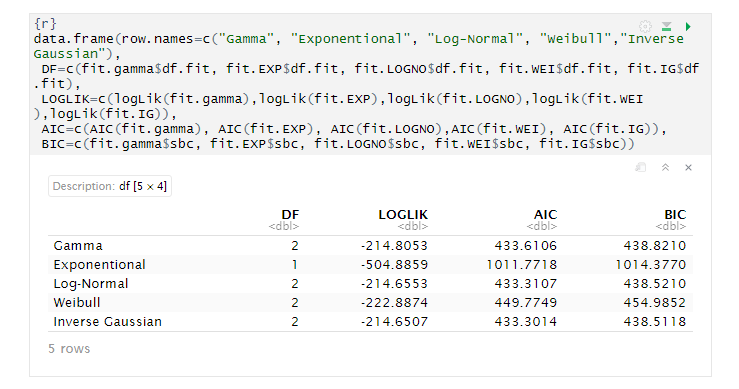
Data fitting for Table, BIC and AIC are estimated in order to evaluate the fit of the distribution.





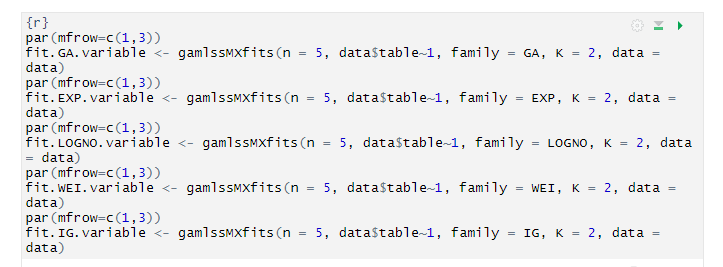


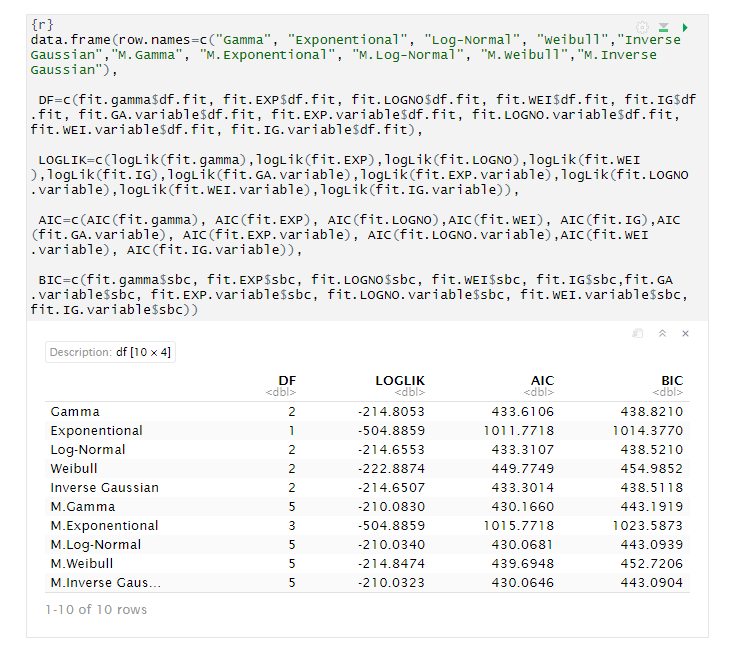




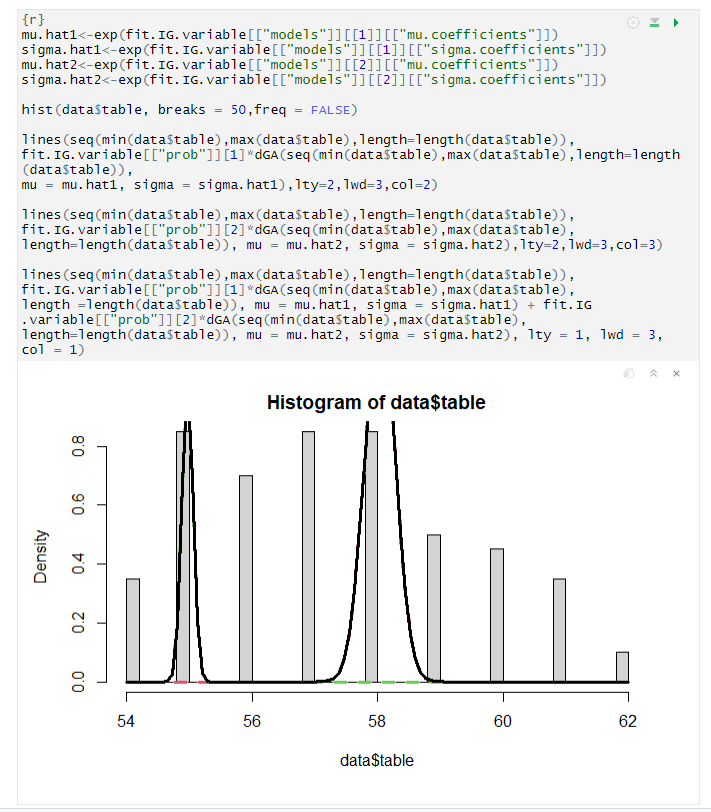
we can observe in the above summary that the model which has small value in AIC and BIC is “Inverse Gaussian distribution”.

The mixture of distribution: It is possible to compute a mixture of two distributions to find the best mixture, the algorithm is repeated five times.



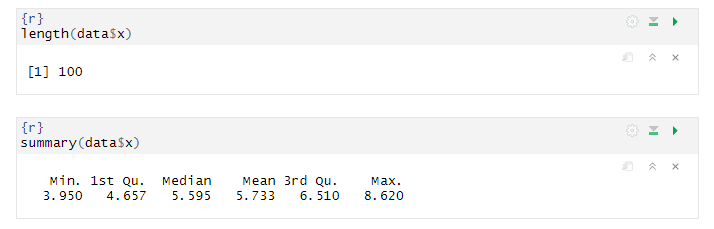


In the first column, the family with “M.” at the start indicate that it is a mixture. In the second column, it’s possible to look at the number of parameters. The best fitting is performed through a mixture distribution with the Inverse Gaussian family.



## X

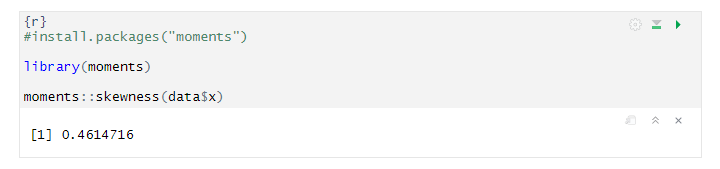
This is a numeric continuous variable. X is the length in mm. The basic statistical values are mentioned below.



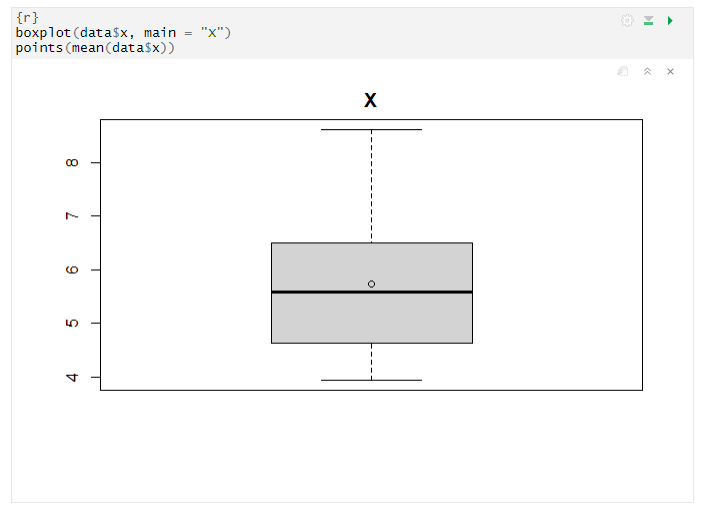
There is total 100 diamonds and x range are between 3.95 to 8.6



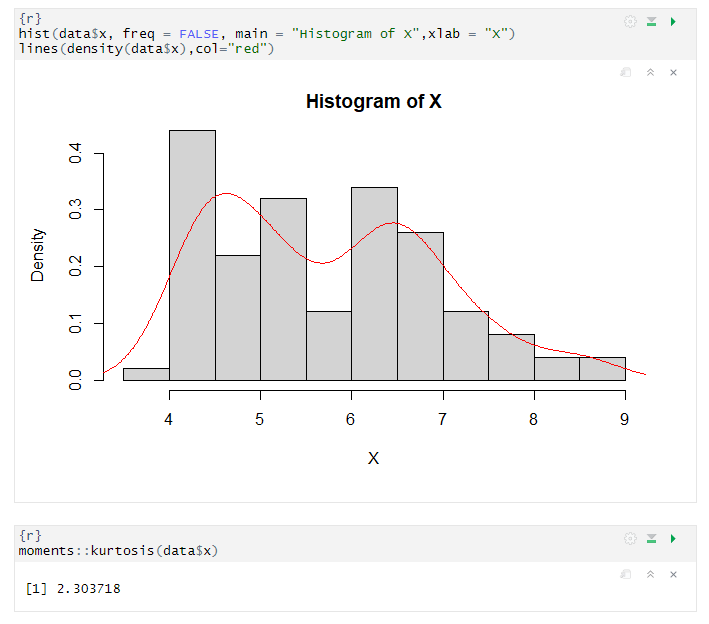
From above statistical computation we can observed that the mean and median are not identical, therefore the distribution is asymmetrical and skewed. And as mean is greater than median, hence the distribution should be positively skewed.



As the computed result of skewed is positive. Hence our prediction was correct.

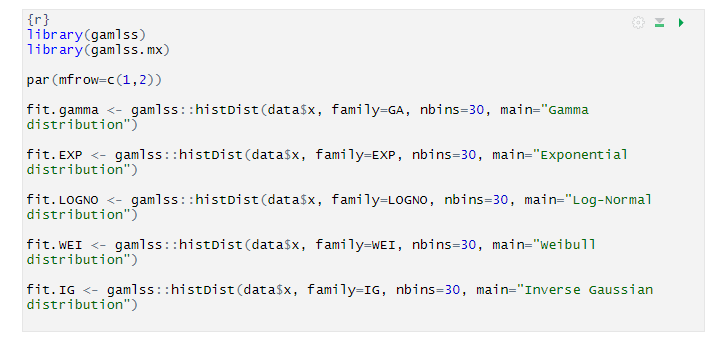


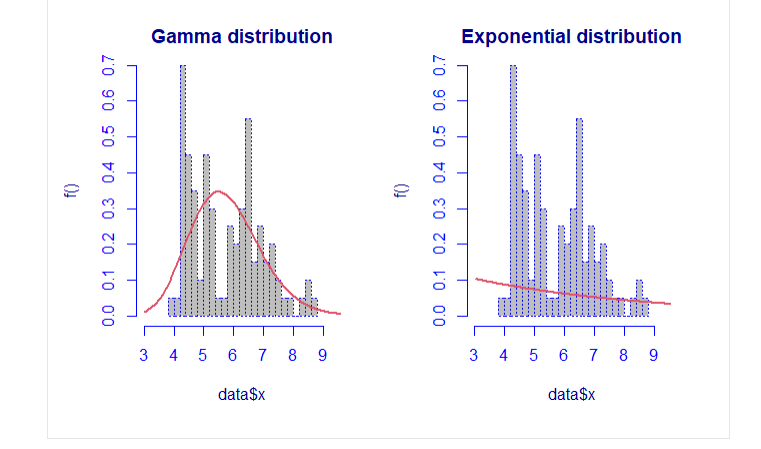
While reviewing the box plot, we can observe that there no outliers in this. Now we can plot the histogram and look for the model that fits better the distribution

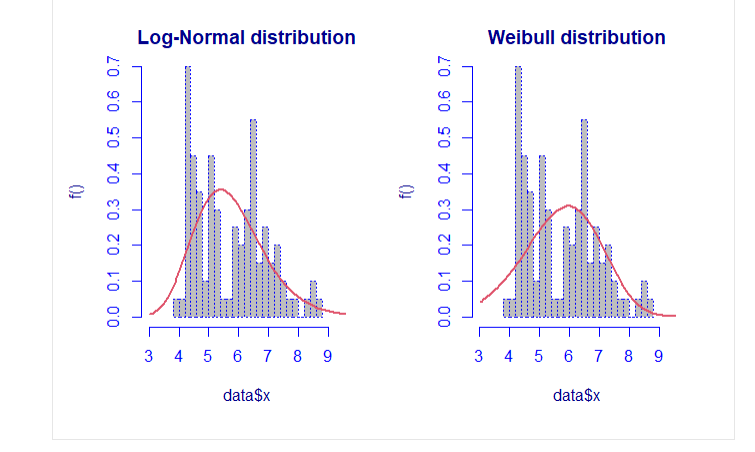


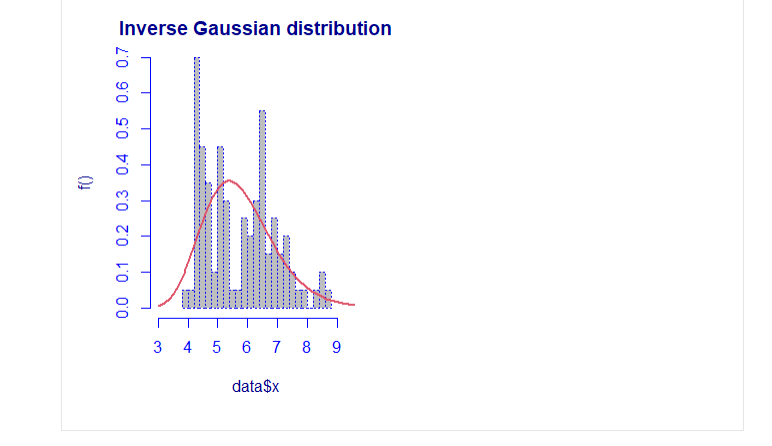
In the above box plot, we can see no outlier, then we see the histogram and the red line and then compute the kurtosis. kurtosis (2.3) indicates that your distribution is flatter than a normal curve.

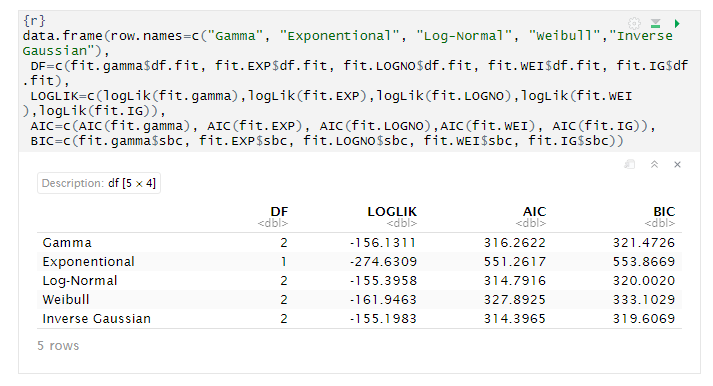
Data fitting for Table, BIC and AIC are estimated in order to evaluate the fit of the distribution.





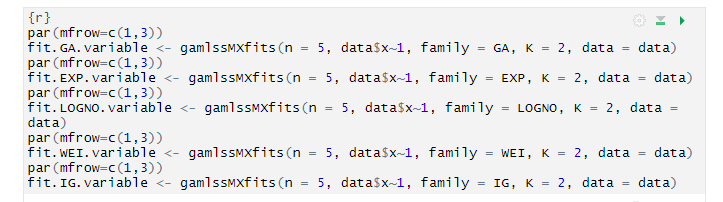


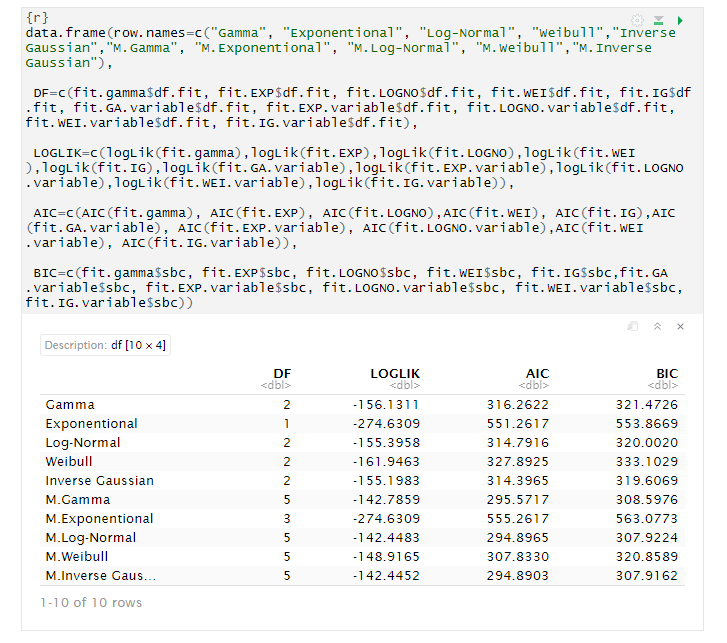




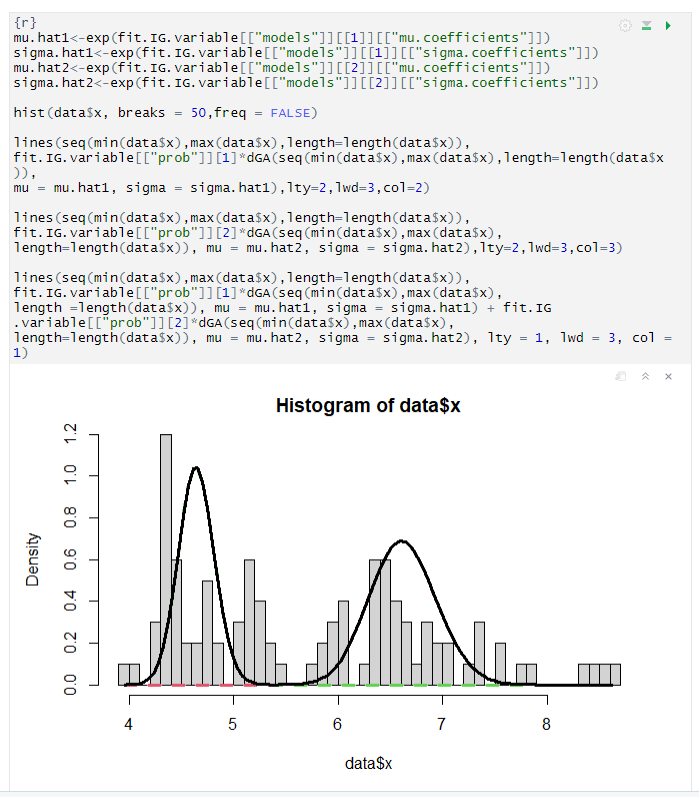
we can observe in the above summary that the model which has small value in AIC and BIC is “Inverse Gaussian distribution.

The mixture of distribution: It is possible to compute a mixture of two distributions to find the best mixture, the algorithm is repeated five times.



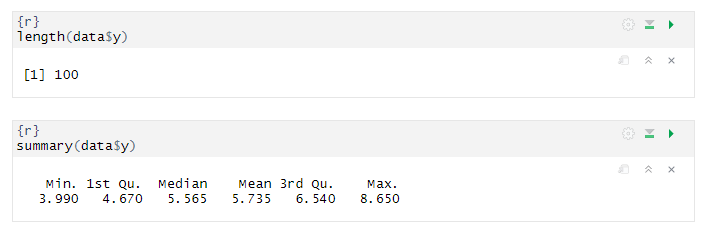


In the first column, the family with “M.” at the start indicate that it is a mixture. In the second column, it’s possible to look at the number of parameters. The best fitting is performed through a mixture distribution with the Inverse Gaussian family.



## Y

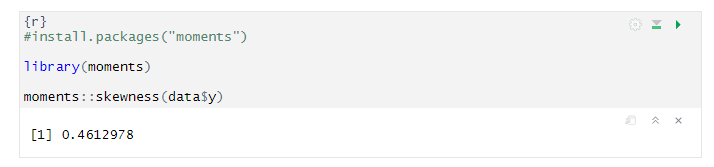
This is a numeric continuous variable. Y is the width in mm. The basic statistical values are mentioned below.



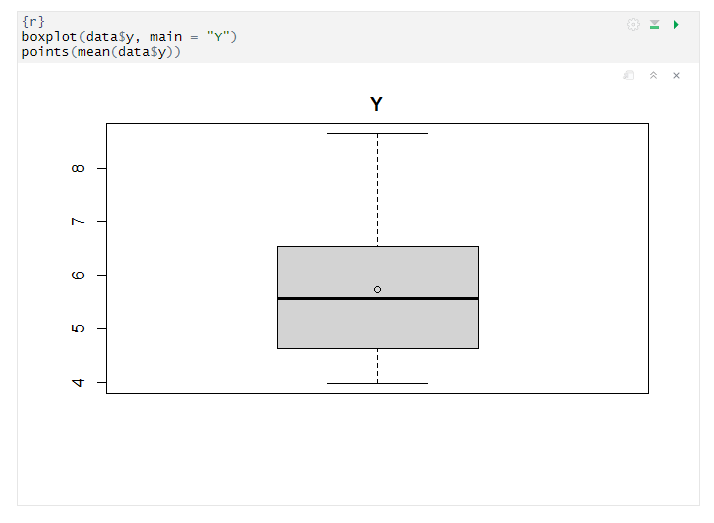
There is total 100 diamonds and y range are between 3.9 to 8.6



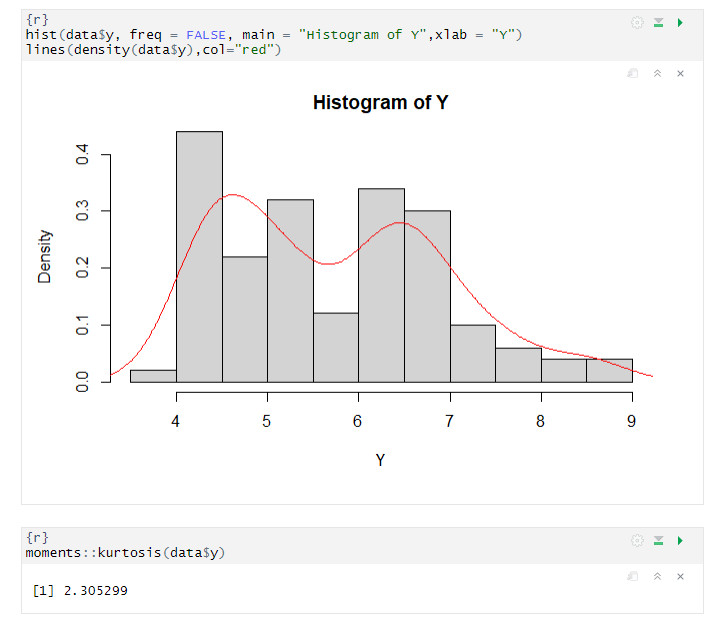
From above statistical computation we can observed that the mean and median are not identical, therefore the distribution is asymmetrical and skewed. And as mean is greater than median, hence the distribution should be positively skewed.



As the computed result of skewed is positive. Hence our prediction was correct.

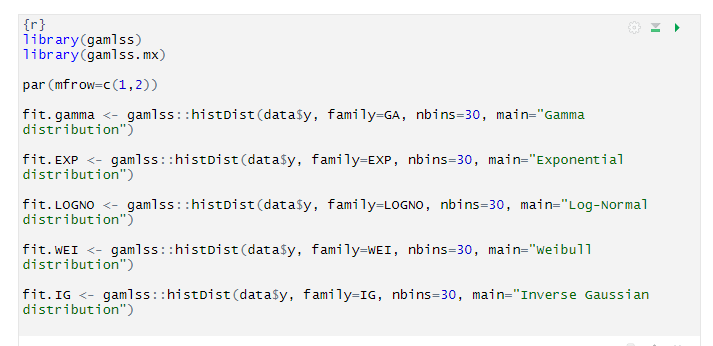


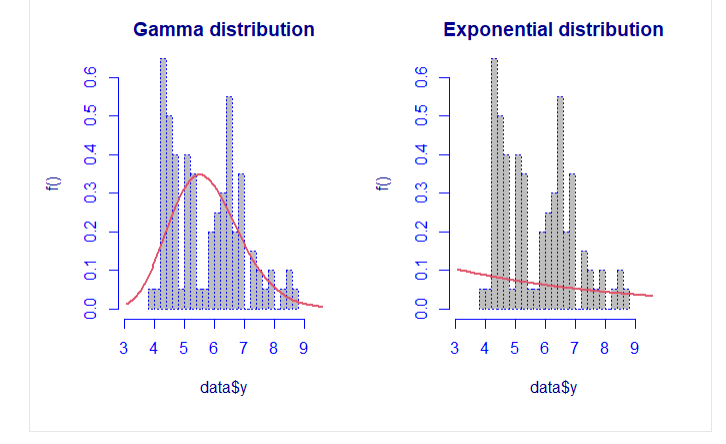
While reviewing the box plot, we can observe that there no outliers in this. Now we can plot the histogram and look for the model that fits better the distribution

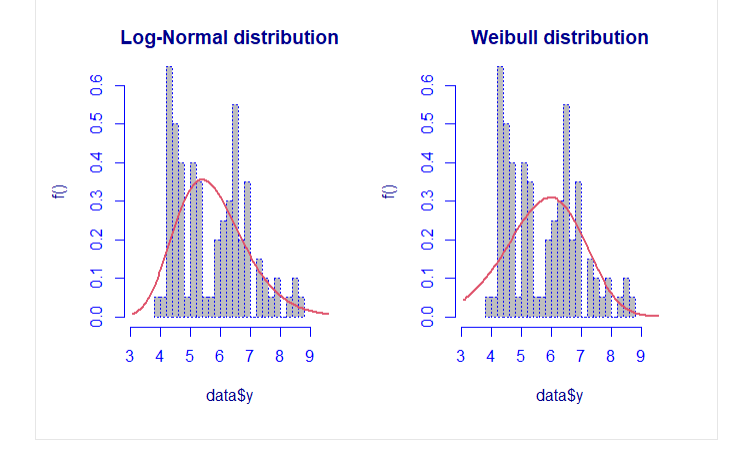


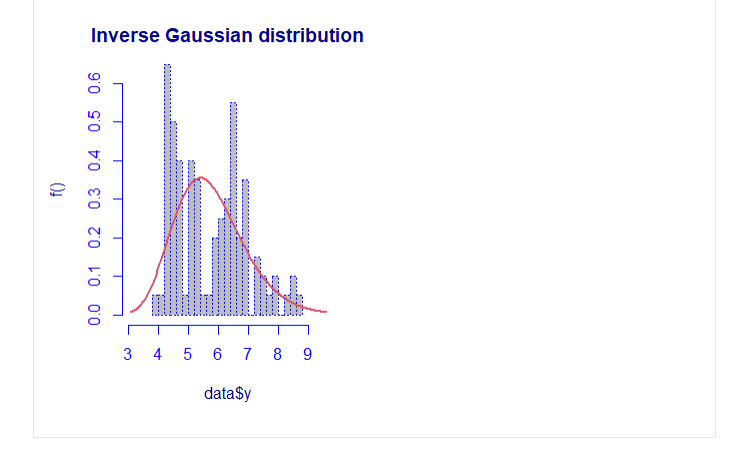
In the above box plot, we can see no outlier, then we see the histogram and the red line and then compute the kurtosis. kurtosis (2.3) indicates that your distribution is flatter than a normal curve.

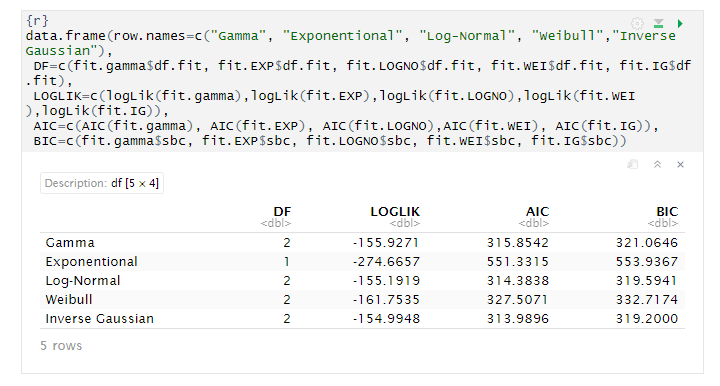
Data fitting for Table, BIC and AIC are estimated in order to evaluate the fit of the distribution.





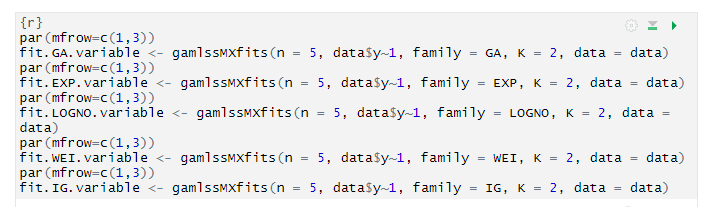


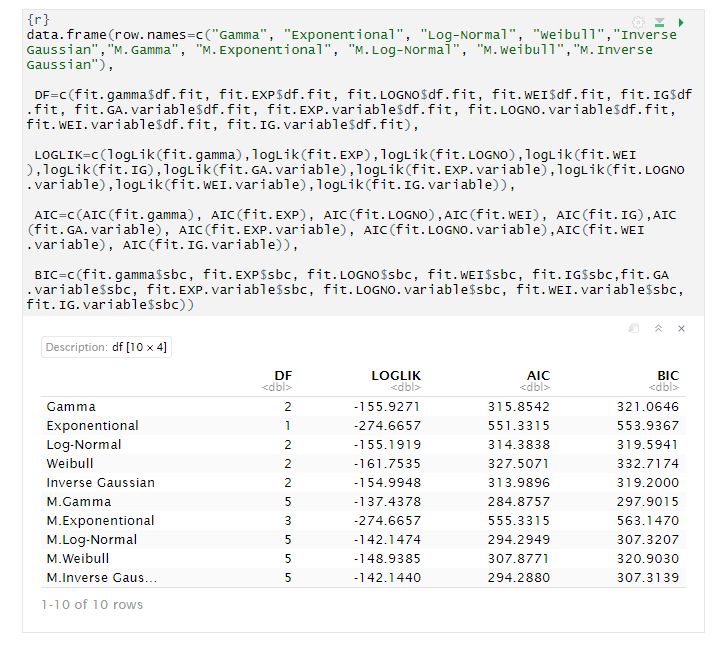




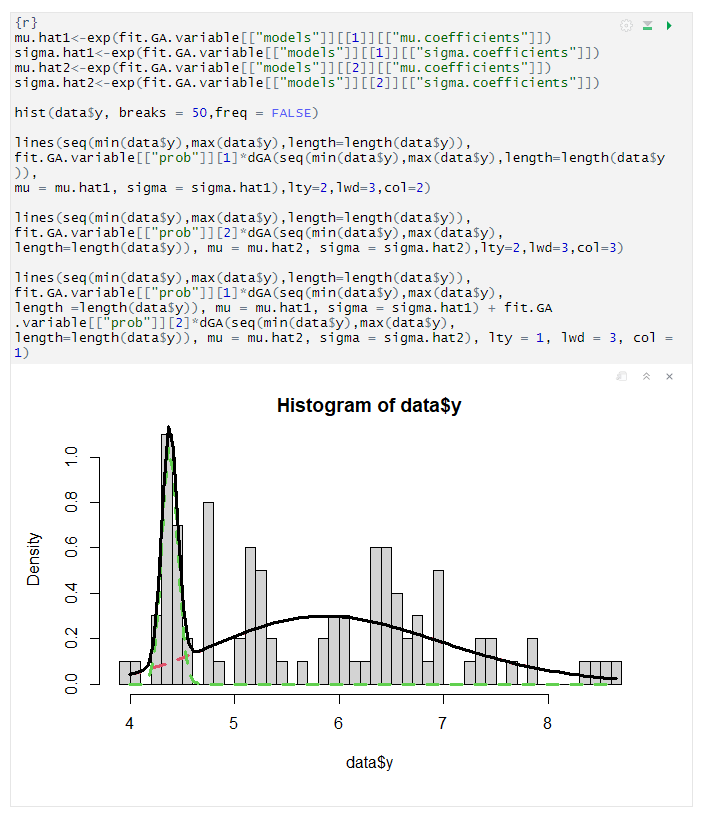
we can observe in the above summary that the model which has small value in AIC and BIC is “Inverse Gaussian distribution”.

The mixture of distribution: It is possible to compute a mixture of two distributions to find the best mixture, the algorithm is repeated five times.



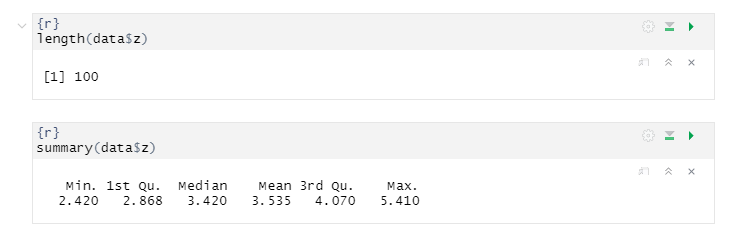


In the first column, the family with “M.” at the start indicate that it is a mixture. In the second column, it’s possible to look at the number of parameters. The best fitting is performed through a mixture distribution with the Gamma family.



## Z

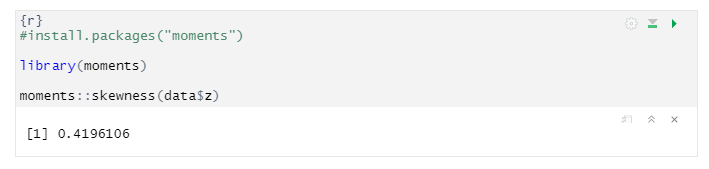
This is a numeric continuous variable. Z is the depth in mm. The basic statistical values are mentioned below.



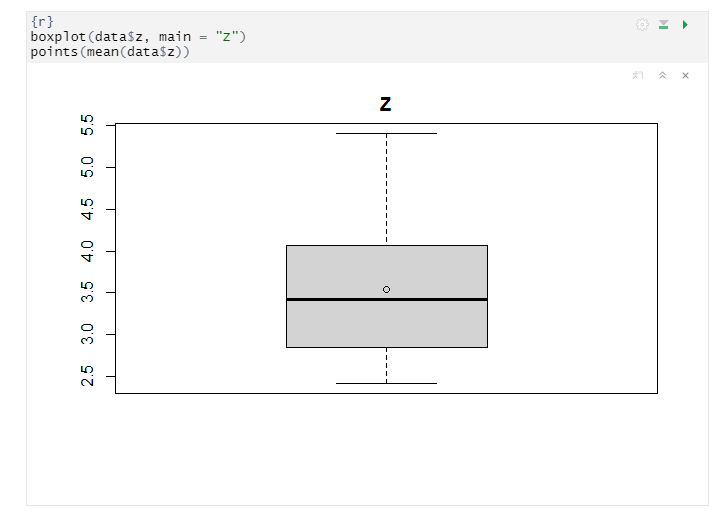
There is total 100 diamonds and z range are between 2.4 to 5.4



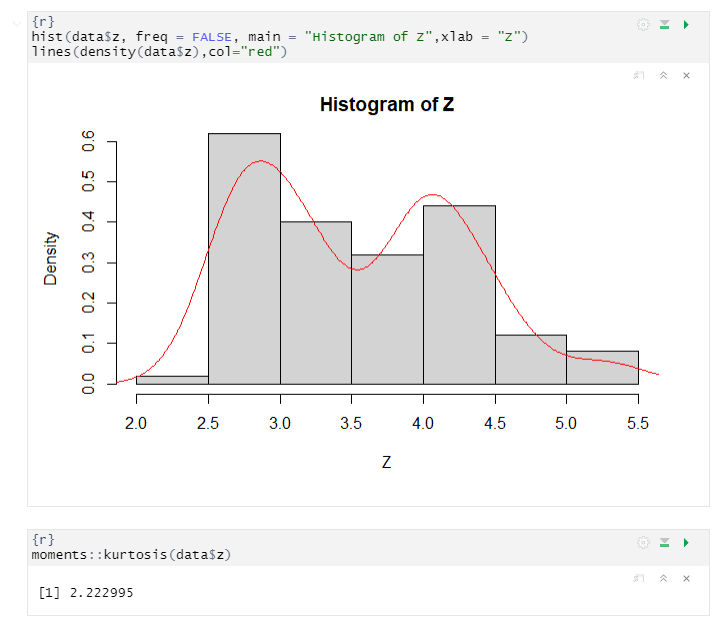
From above statistical computation we can observed that the mean and median are not identical, therefore the distribution is asymmetrical and skewed. And as mean is greater than median, hence the distribution should be positively skewed.



As the computed result of skewed is positive. Hence our prediction was correct.

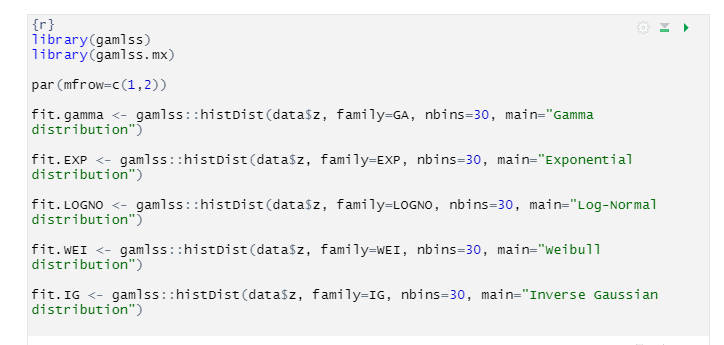


While reviewing the box plot, we can observe that there no outliers in this. Now we can plot the histogram and look for the model that fits better the distribution

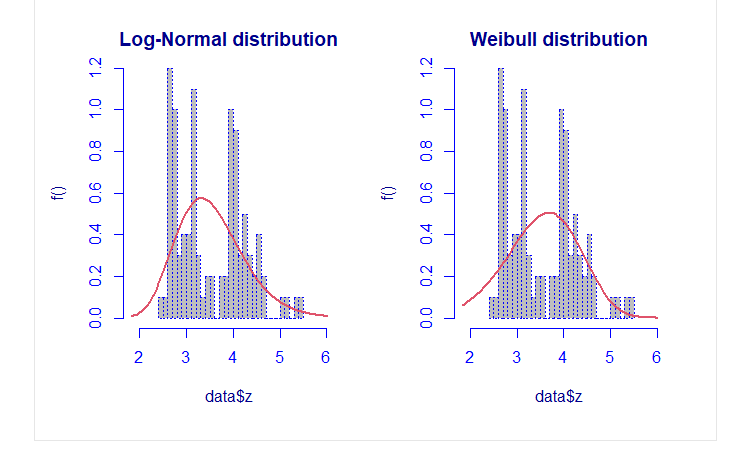


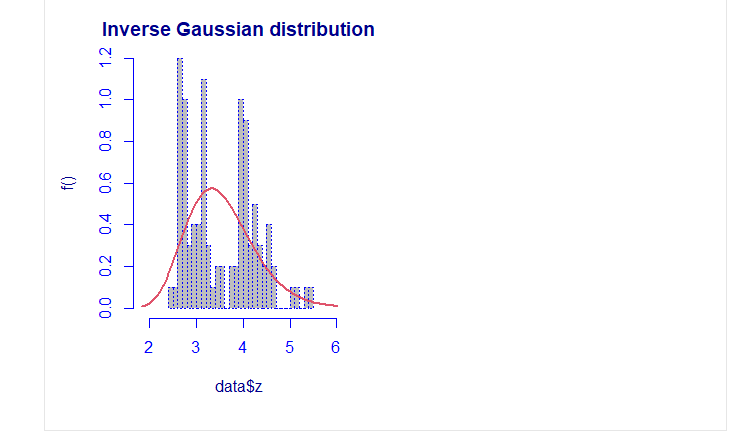
In the above box plot, we can see no outlier, then we see the histogram and the red line and then compute the kurtosis. kurtosis (2.2) indicates that your distribution is flatter than a normal curve.

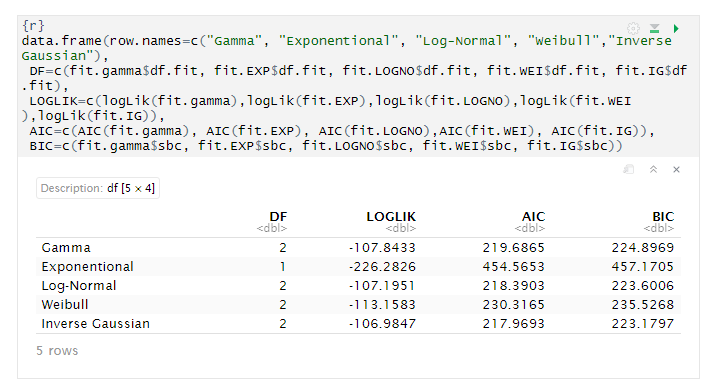
Data fitting for Table, Log-likelihood value, BIC and AIC are estimated in order to evaluate the fit of the distribution.





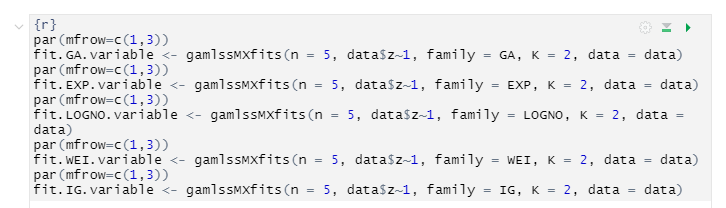


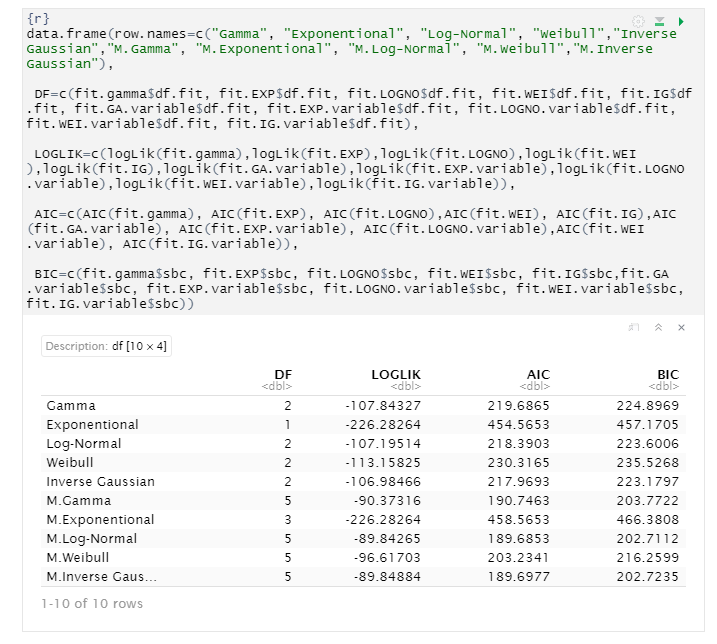




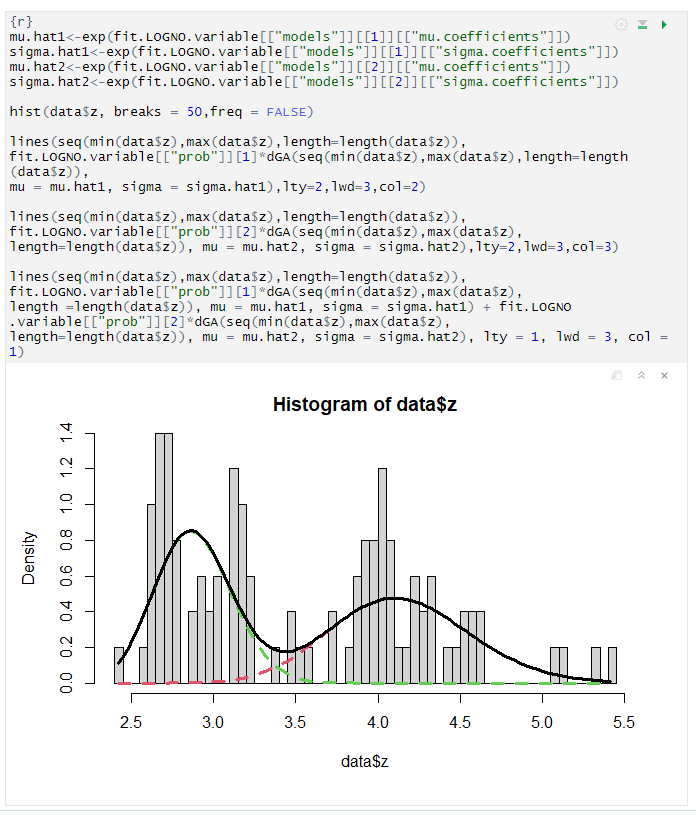
we can observe in the above summary that the model which has small value in AIC and BIC is “Inverse Gaussian distribution”.

The mixture of distribution: It is possible to compute a mixture of two distributions to find the best mixture, the algorithm is repeated five times.



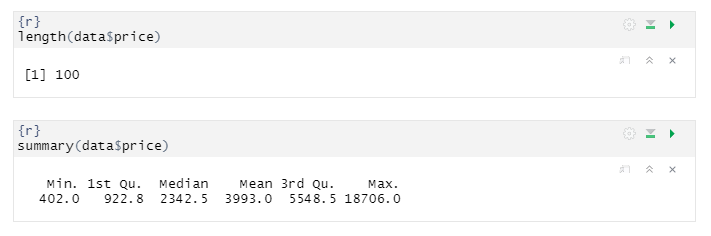


In the first column, the family with “M.” at the start indicate that it is a mixture. In the second column, it’s possible to look at the number of parameters. The best fitting is performed through a mixture distribution with the Log Normal family.



## Price

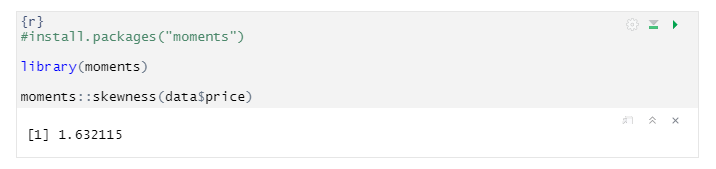
This is a numeric continuous variable. Price is in US Dollar. The basic statistical values are mentioned below.



There is total 100 diamonds and price range are between 402 to 18706 US Dollar



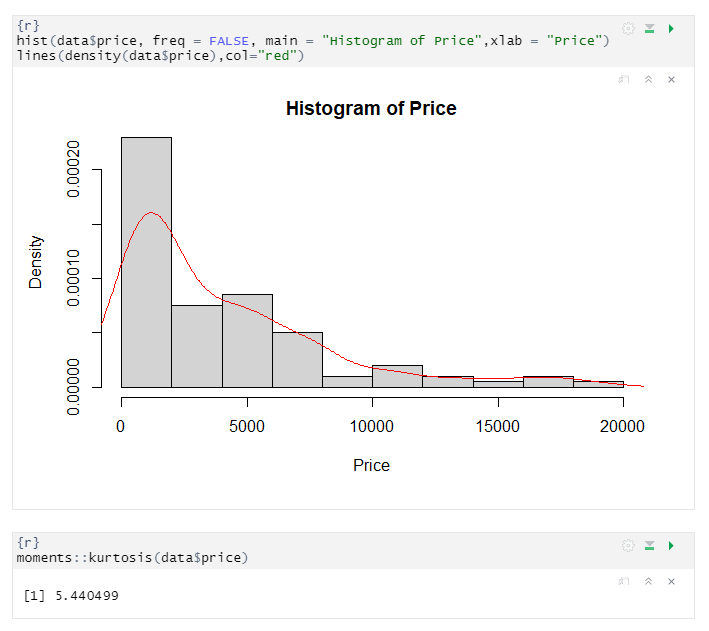
From above statistical computation we can observed that the mean and median are not identical, therefore the distribution is asymmetrical and skewed. And as mean is greater than median, hence the distribution should be positively skewed.



As the computed result of skewed is positive. Hence our prediction was correct.



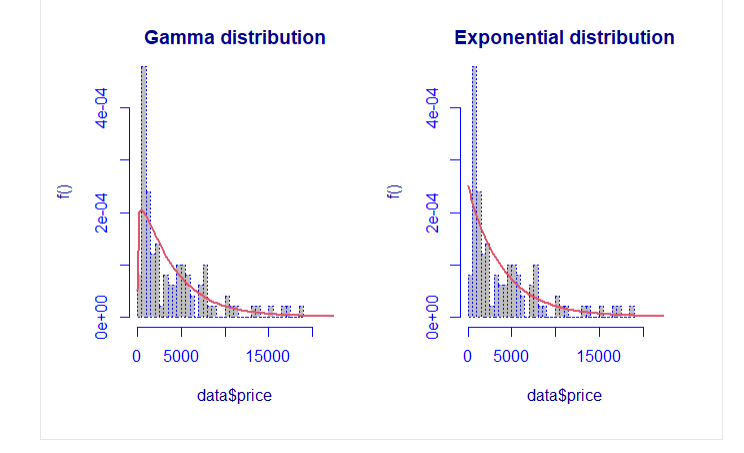
While reviewing the box plot, we can observe that there are outliers in this as we see some dots lie outside the upper whiskers of the boxplot, which means that there are some observations that is numerically distant from the rest of the data. Outliers are the data points located outside the whiskers of the boxplot. Now we can plot the histogram and look for the model that fits better the distribution

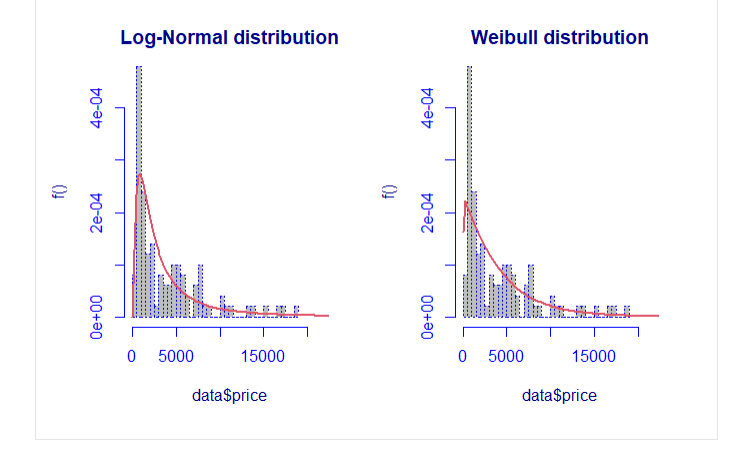


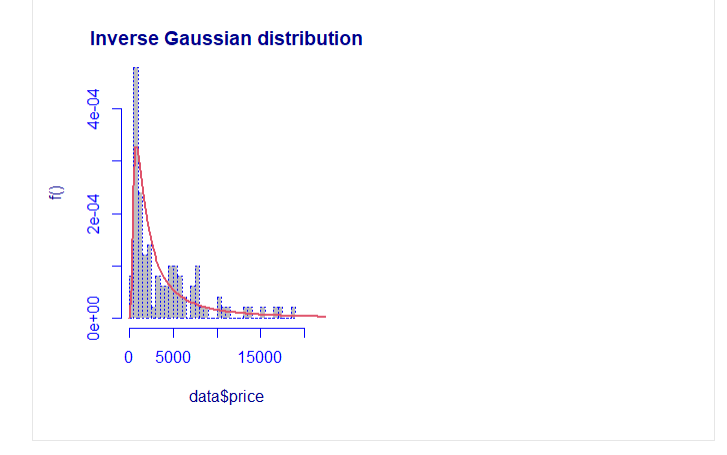
In the above box plot, we can see so many outliers, then we see the histogram and the red line and then compute the kurtosis. kurtosis (5.4) indicates that the distribution has more outliers than a normal distribution it is also known as leptokurtic.

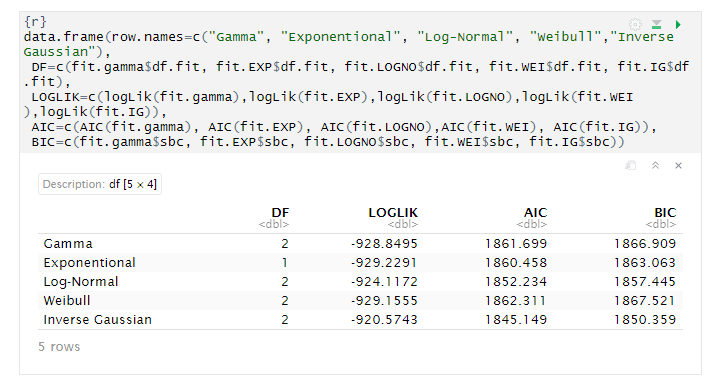
Data fitting for Carats, Log-likelihood value, BIC and AIC are estimated in order to evaluate the fit of the distribution.





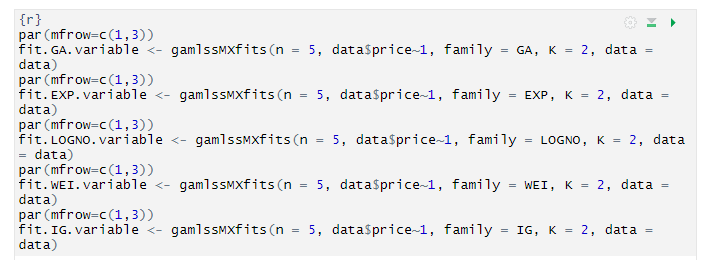


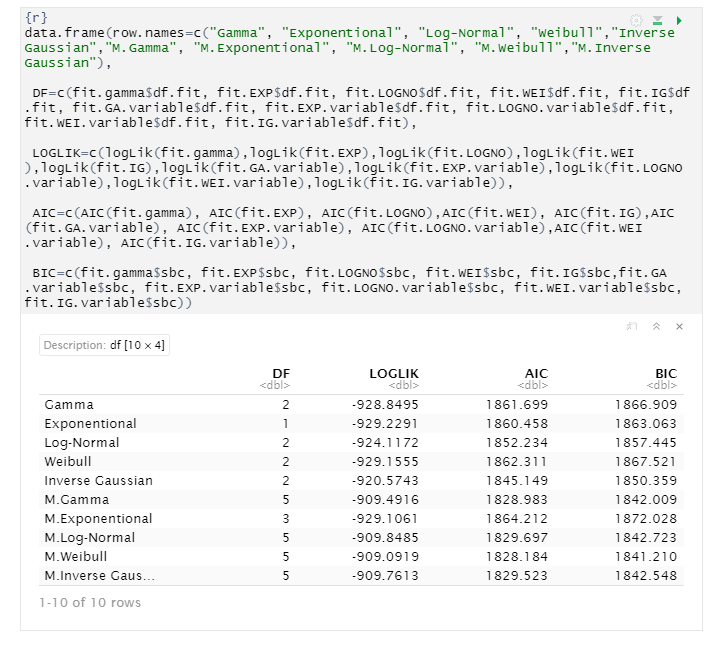




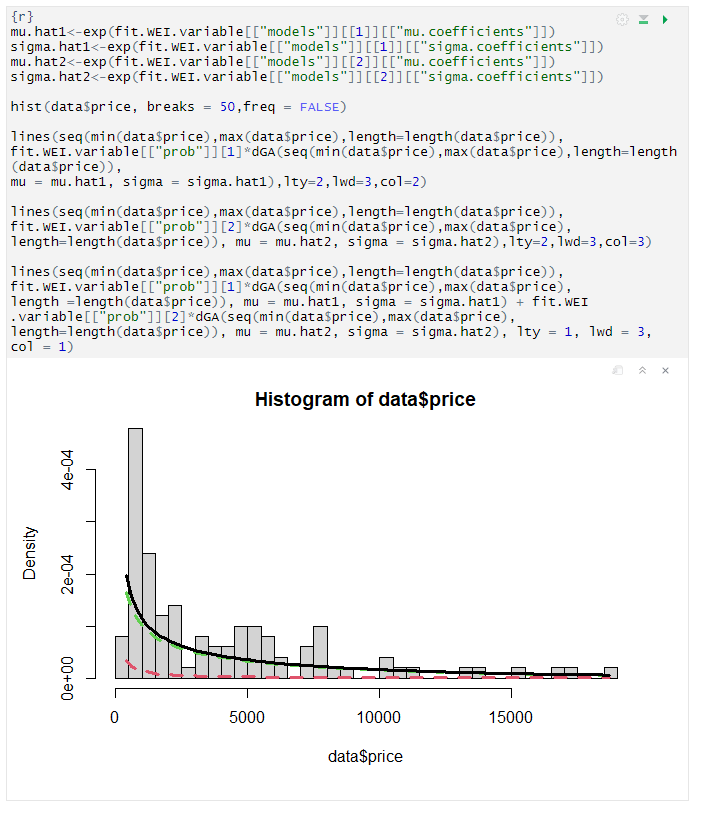
we can observe in the above summary that the model which has small value in AIC and BIC is “Inverse Gaussian distribution”

The mixture of distribution: It is possible to compute a mixture of two distributions to find the best mixture, the algorithm is repeated five times.





In the first column, the family with “M.” at the start indicate that it is a mixture. In the second column, it’s possible to look at the number of parameters. The best fitting is performed through a mixture distribution with the Weibull family.



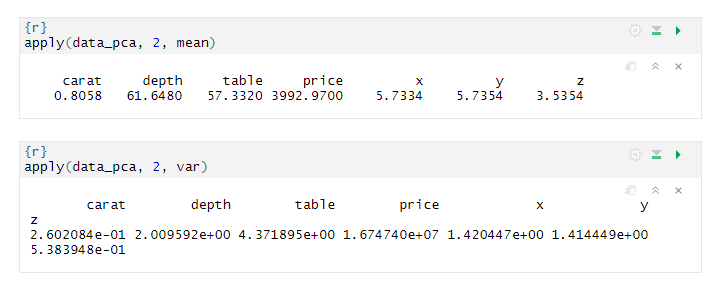
# PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of n observations of possibly correlated numerical variables, into a set of n values of new linearly uncorrelated variables called principal components (PCs). This transformation is defined in such a way that the first PC has the largest possible variance and each succeeding PC in turn has the highest variance possible under the constraint that it is orthogonal to the preceding PCs. The goal of PCA is to explain most of the variability in the data with a smaller number of variables than the original data set. Before proceeding with the PCA, it is necessary to evaluate whether there is a correlation between the numerical variables of this dataset, so first of all we create the (linear) correlation matrix.

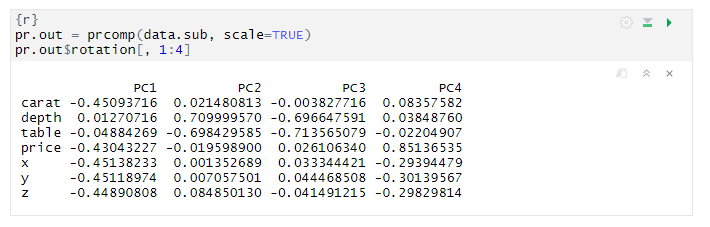


In the above code, first we remove the categorical data from the dataset and then run linear correlation matrix. We can see from the correlation matrix of diamond dataset that there is a high positive correlation between x and y (1), while the depth variable shows negative correlation with all of the other variables.

Now we have to standardize (i.e. scale) the data to make variables comparable, In order to evaluate the difference between the variables, the mean and the variance are computed for each variable.

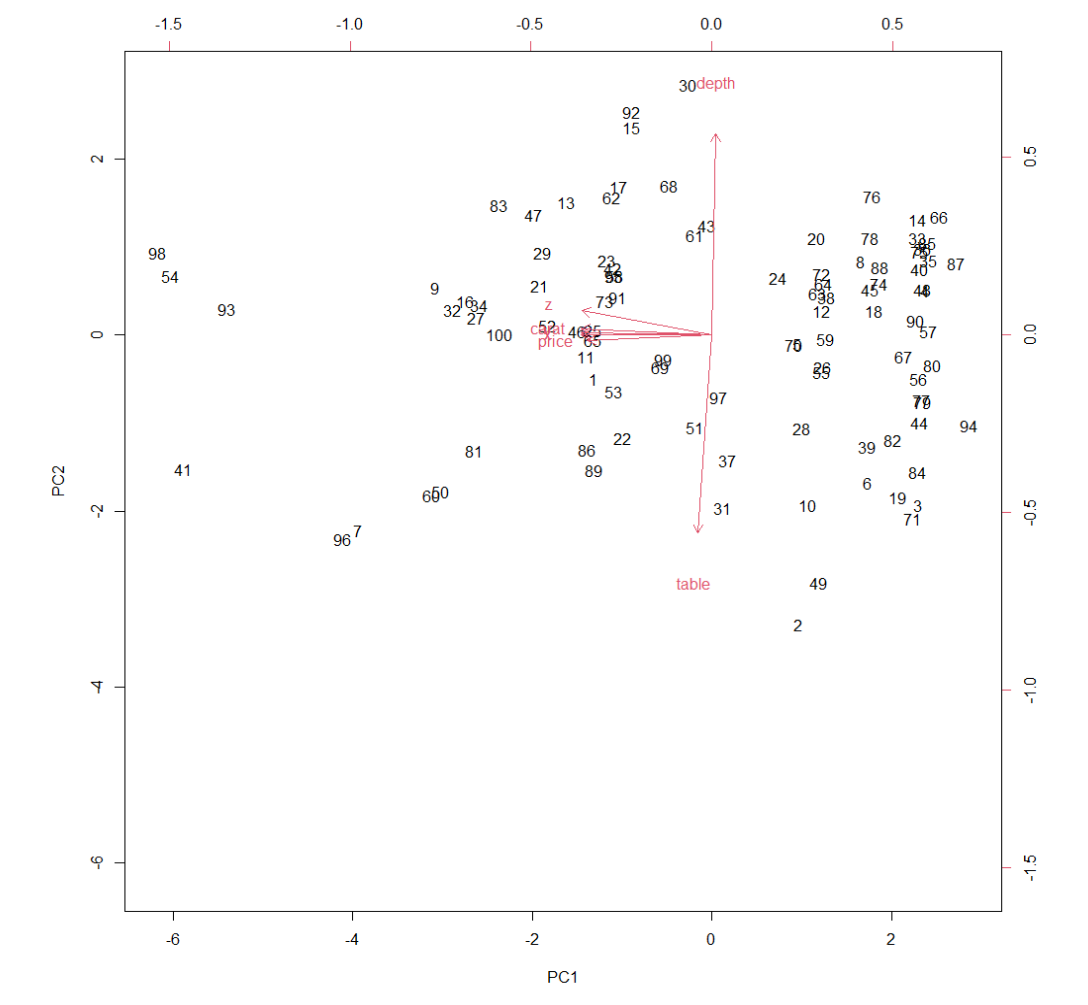


The variables are very different from each other. Now it is better to standardize the variable to have zero mean and unitary variance.



In the above code, we scaled our data for PCA using prcomp. If we look at above PCs, we can see that it indicates that we need minimum two PC and maximum three PC.

We can use the **biplot** to represent at the same time original variables, principal components and the sample units (PC scores):

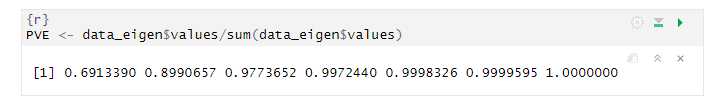


From this plot we can see that there is a high correlation between x and y and between y and price, because the angle between their relative arrows is close to zero and this means that the correlation is close to 1. 90° degrees angles imply the absence of correlation, as depth and carat indicated 90° angle and 180° degrees imply a perfect negative correlation as depth and table indicates. It is difficult to analyses the correlation from the above image. Overall, it looks like there are one variable near 180 degrees that imply a negative correlation and one variable has 90 degrees.

## Methods for selecting PC

We have analyzed the first two principal components, but are they good enough to explain the variability of the data? We can answer to this by means of different approaches:

## Cumulative proportion of variance explained (**CPVE**)

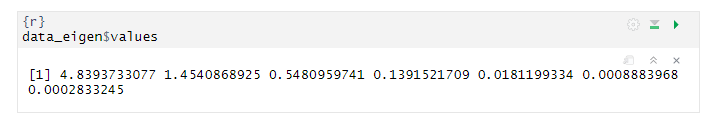


* The first PC explains approx. 69% of variability
* The second PC explains approx. (69 – 89 = 20) 20% of variability
* The third PC explains approx. (89 – 97 = 8) 8% of variability

And so, on

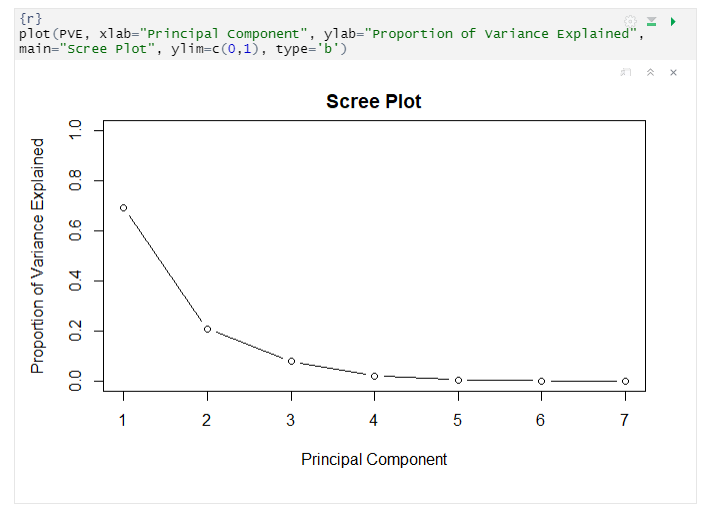
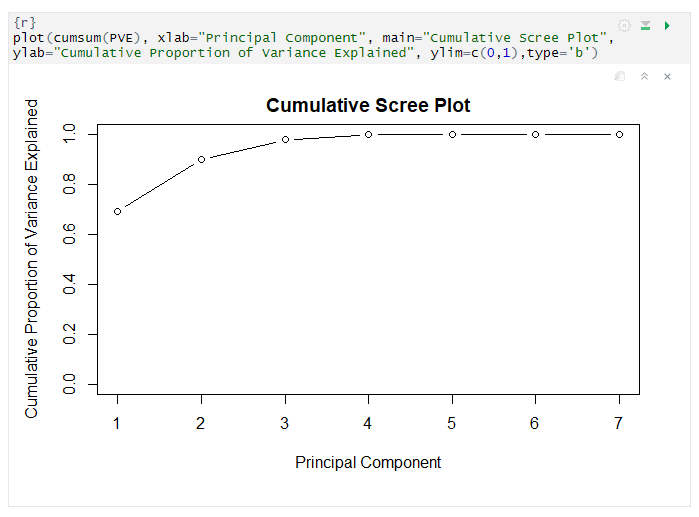
According to the CPVE we have to retain as many PCs as needed to explain at least the 80% of the total variance, hence we have to retain the first 2 PCs.

## Kaiser’s rule



The rule suggests retaining the principal components with variance greater than 1, reason why we can retain the first 2 PCs.

## Scree plot



This is the least useful method due to the difficulty in seeing an elbow, but as in this case elbow is not clear as much but it may be at 2 PC so we can state that we can select the first two PCs.

## PCA results

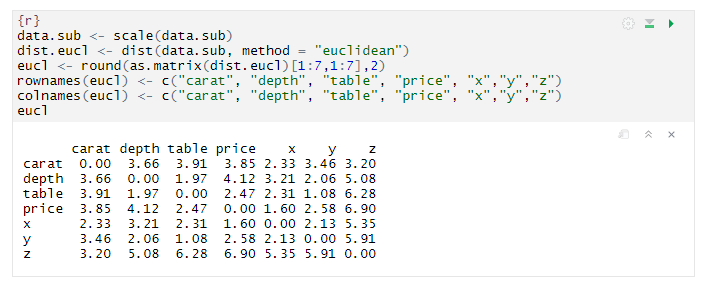
Based on different strategies I obtained almost same results. The cumulative PVE rule suggests retaining the first two principal components, the Kaiser’s rule suggests retaining the first two principal components while the Scree plot doesn’t provide a clear image but its also looks like first two principal components Therefore, I decided to choose CPVE and Kaiser’s rule results.

# CLUSTER ANALYSIS

The aim of the clustering is to find homogeneous subgroups among the observations and there are several methods to achieve this goal. However, the analysis consists of different steps.

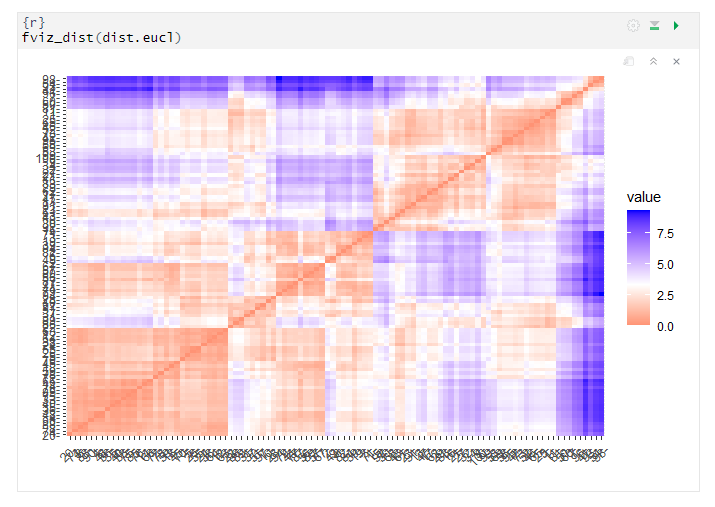
The first step is to calculate a certain type of distance between pairs of units and build the distance matrix.

## Euclidean distance

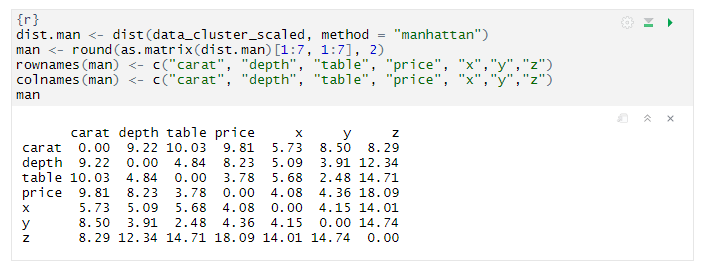


According to this distance matrix, relative to a subset of data, the most distant observations are (1.08) y and table , while the most similar ones are (6.90) price and z.

We can do the same observations by looking at the graphical visualization of the distance matrix

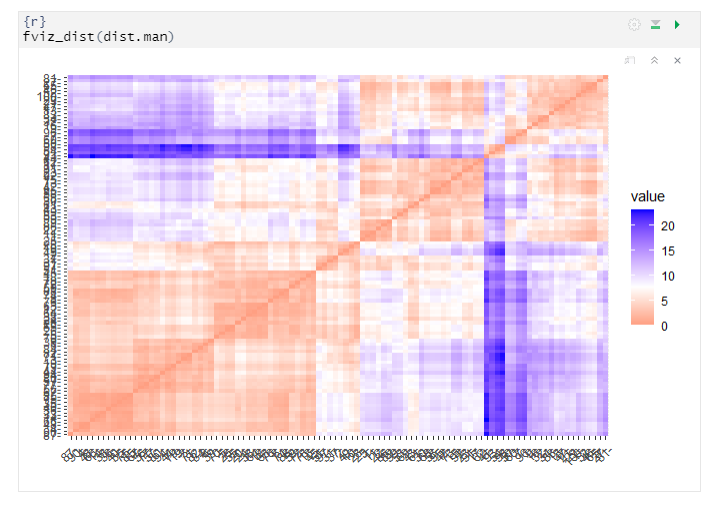


## Manhattan distance



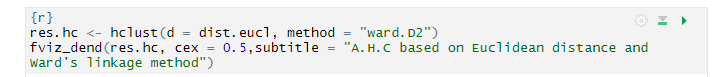
According to this distance matrix, relative to a subset of data, the most distant observations are (4.08) x and price, while the most similar ones are (18.09) prices and z.

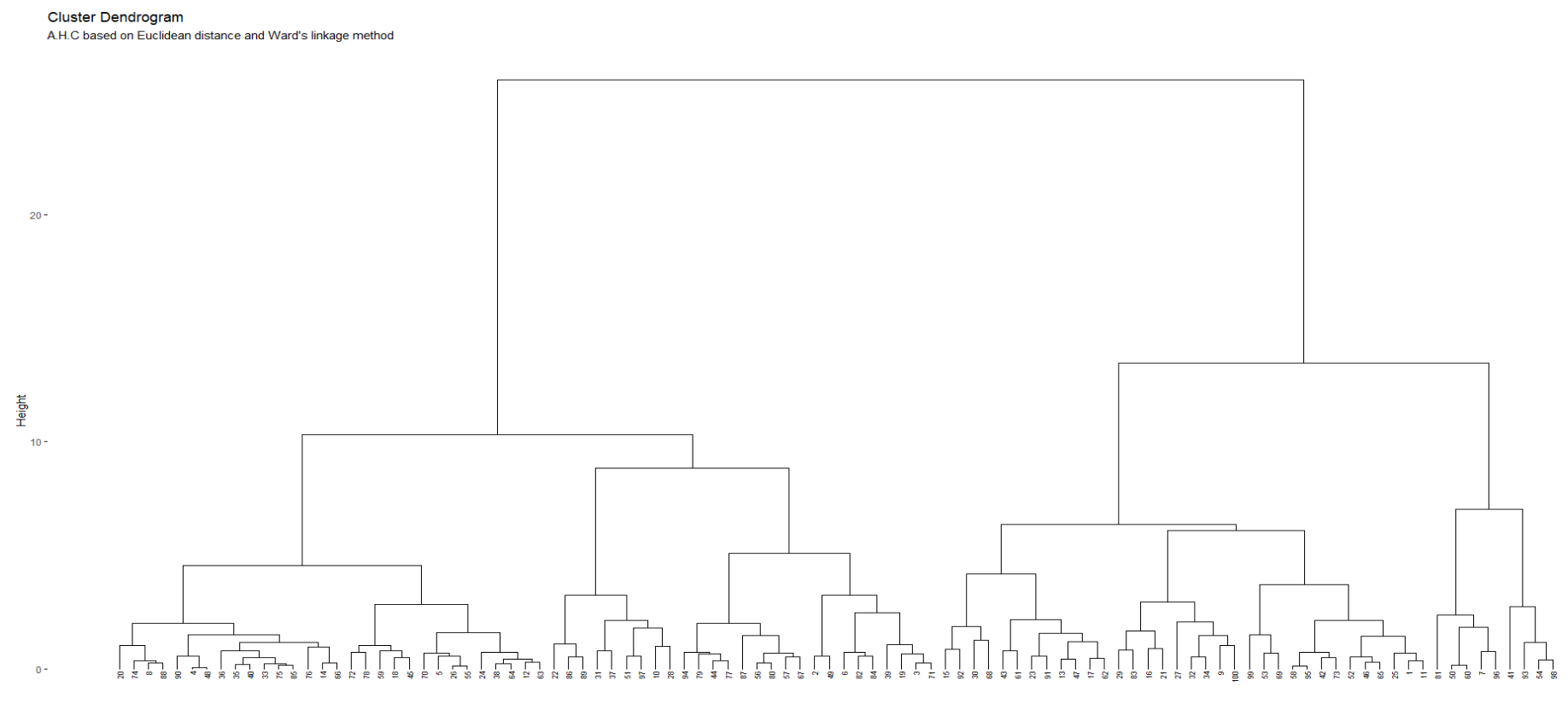
We can do the same observations by looking at the graphical visualization of the distance matrix



## Agglomerative Hierarchical Clustering

### A.H.C. based on Euclidean distance and Ward’s linkage method



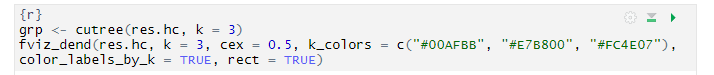


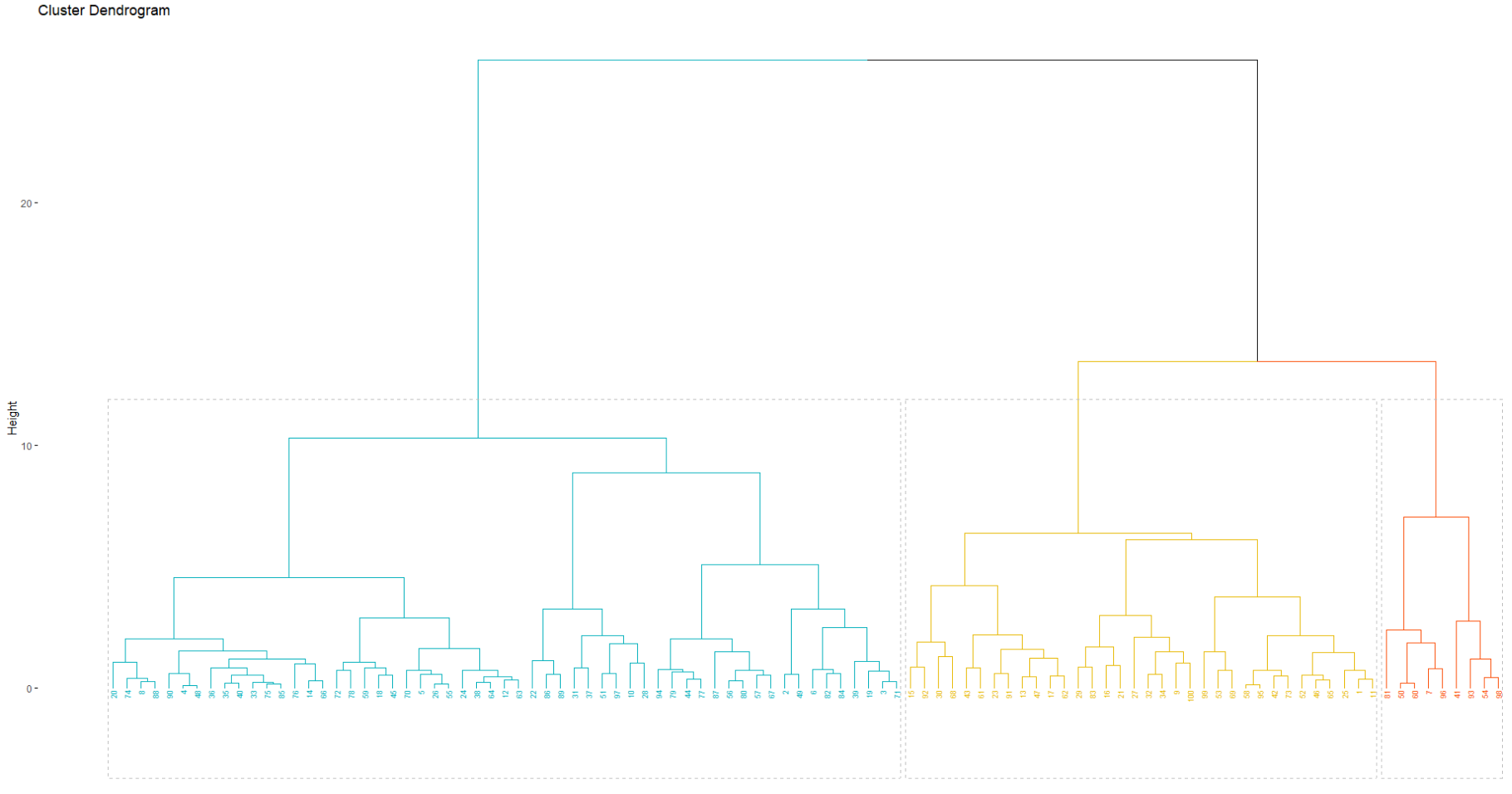
In order to understand if the clustering method is good or not: it is good if the cophenetic distance between units in the cluster tree is well correlated to the distance between units in the original distance matrix.



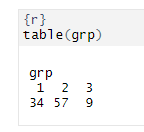
The result is 0.66, which is not such a good result. This means that this clustering method does not preserves the true original distances between units very well.

Now, if we want to see clusters, we have to cut the hierarchical tree, for example specifying the number of groups that we want.

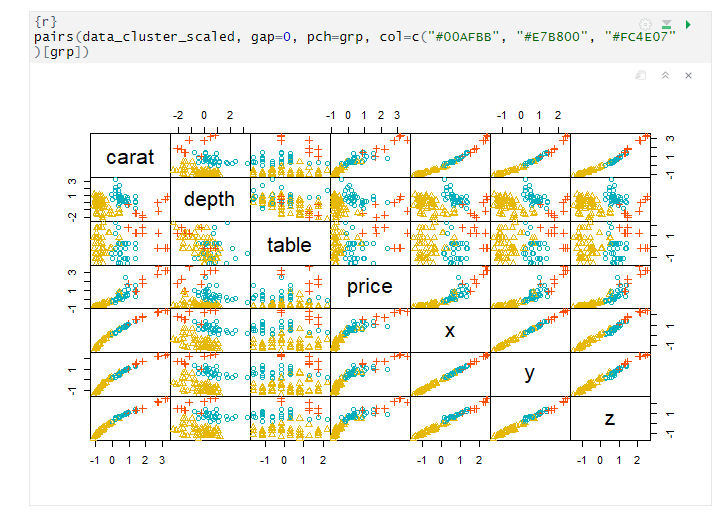




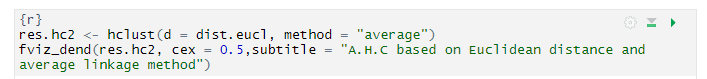
We can see that each cluster is identified by a specific color. Moreover, the second cluster is the bigger one, as we can see from this R code below

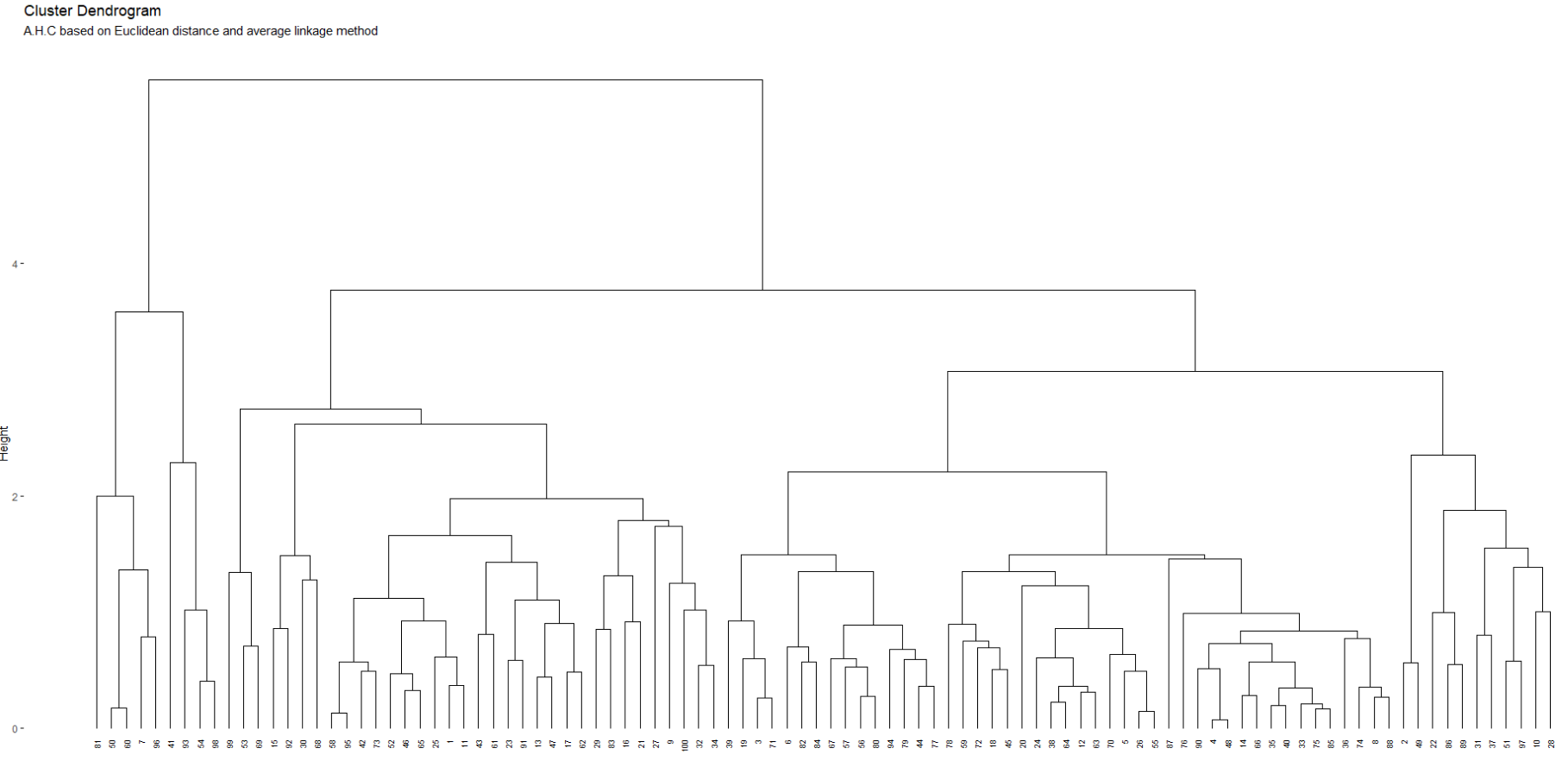


We can visualize the clustering results in the original space via the matrix of pairwise scatterplots



### A.H.C. based on Euclidean distance and Average linkage method



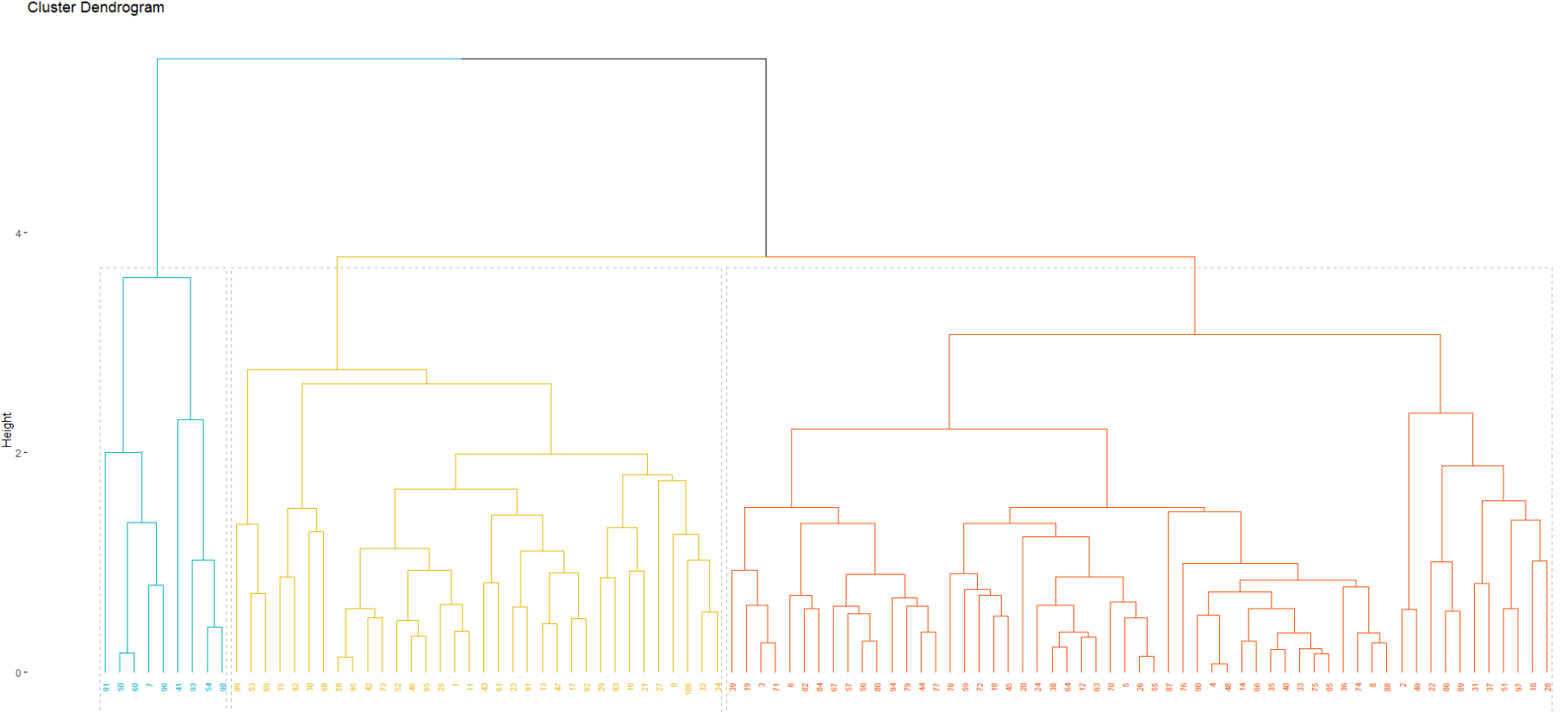


Also, in this case we have to compare the cophenetic distance to the original one, in order to understand if the clustering method is good or not



The result is 0.797, which is good enough. So, this clustering method preserves the true original distances between units quite well, however, better than the previous one, with Ward’s linkage method.

Now, in order to see clusters, we have to cut the hierarchical tree, for example specifying the number of groups that we want.

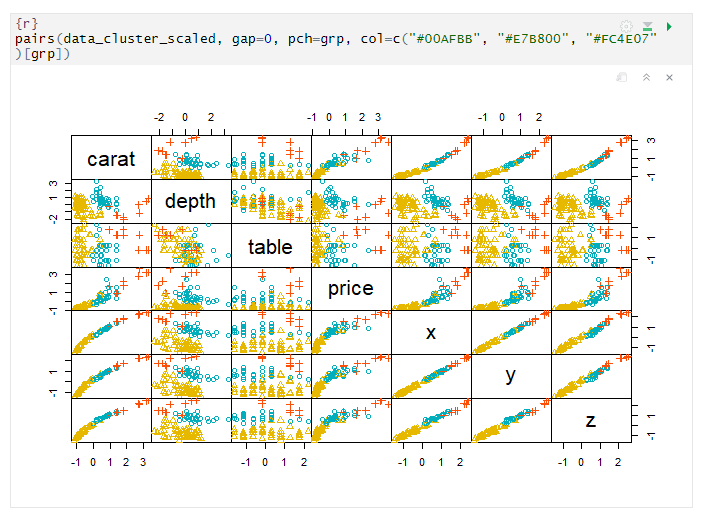


In this case the last cluster is the bigger one, as we can see from this R code:

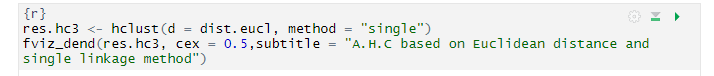


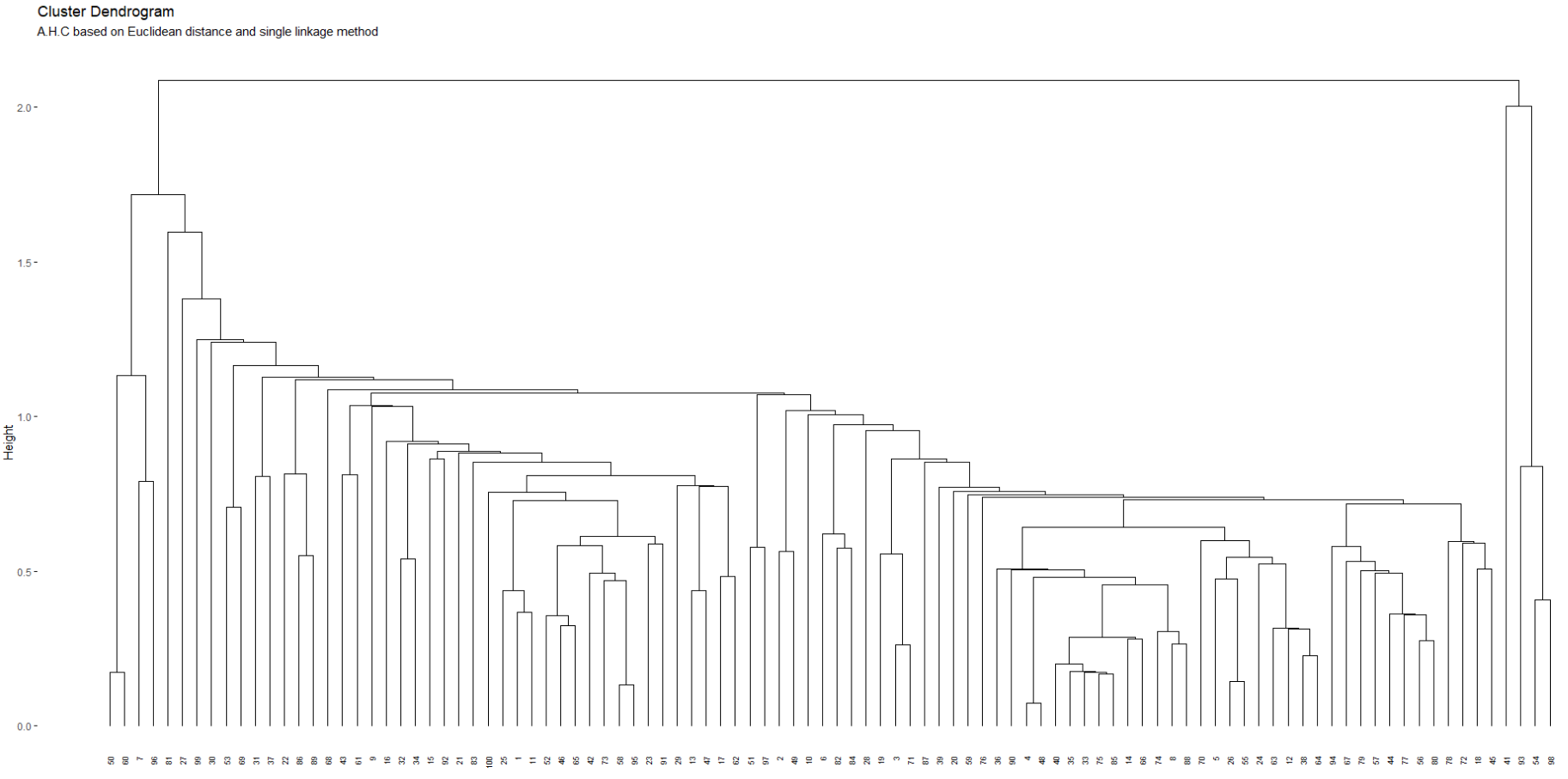
Cluster 2 contains more units than cluster 1 and 3, which contain 34 and 9 units respectively

We can visualize the clustering results in the original space via the matrix of pairwise scatterplots

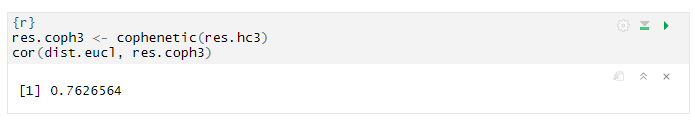


### A.H.C. based on Euclidean distance and Single linkage method



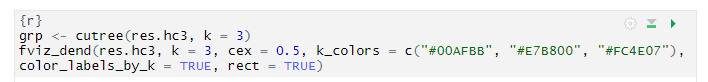


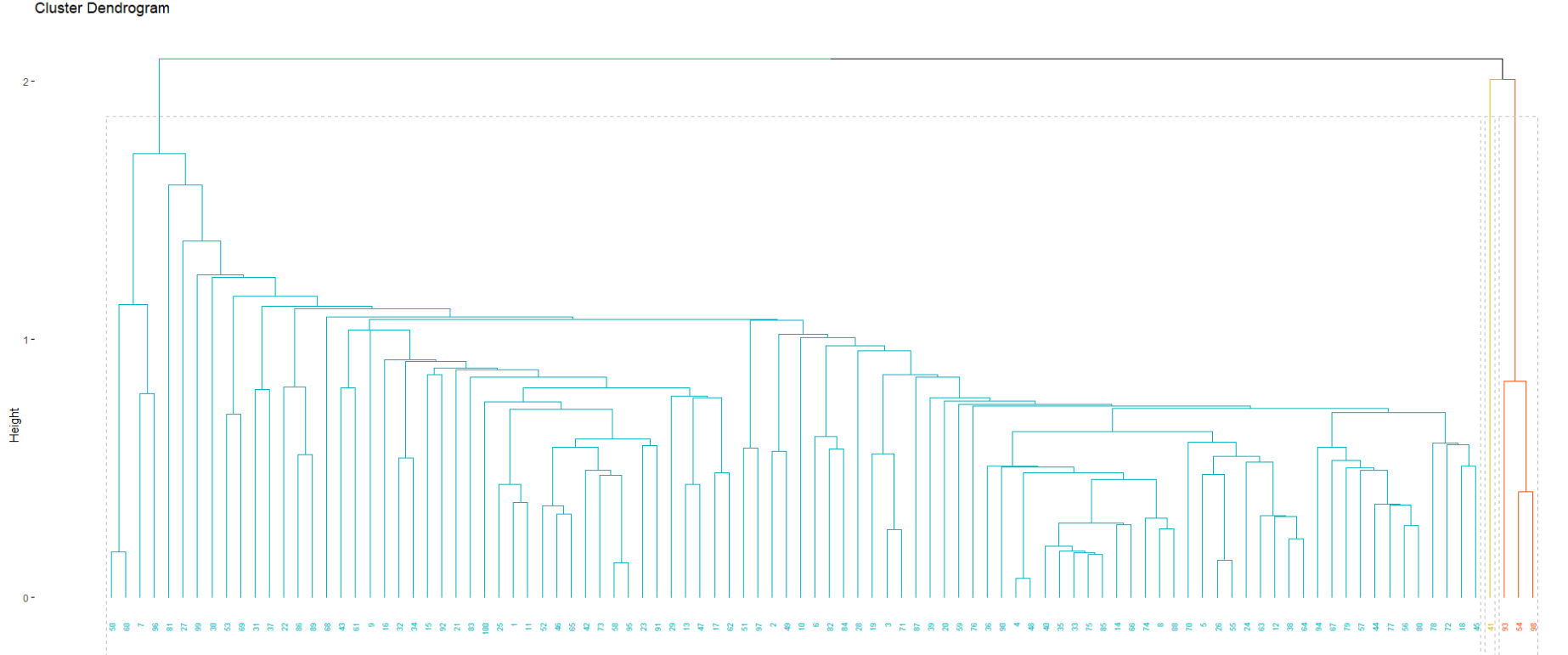
Even just looking at the dendrogram we understand that the distribution of data is fairly ambiguous. So, let’s check for the correlation between the cophenetic distance and the original one



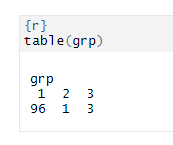
The result is 0.76, which is good enough. So, this clustering method preserves the true original distances between units quite well, however, it’s not better than the previous one, with Average linkage method.

We can visualize the clustering results in the original space via the matrix of pairwise scatterplots

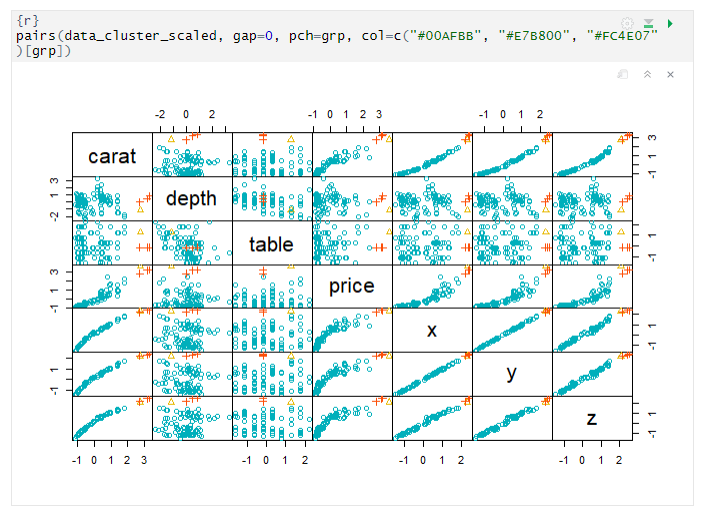




As we can see, almost all units are grouped in a single cluster and we can check this also through the R code which gives us the clusters size

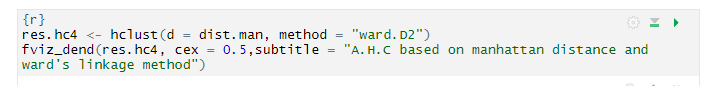


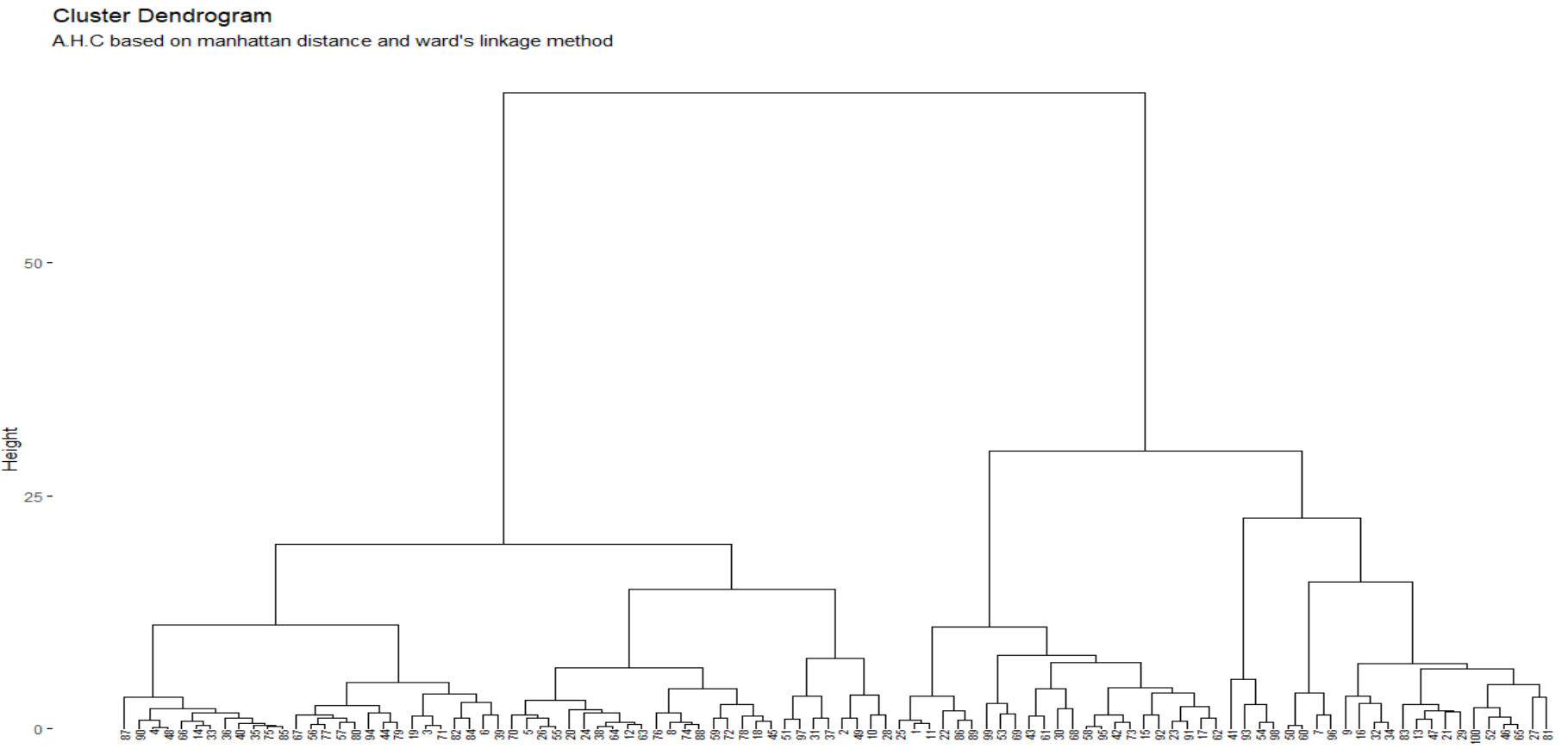
This is evident also by the clustering results in the original space



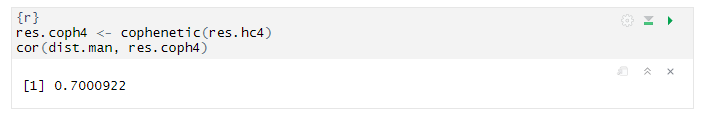
In each pairwise of the scatterplot all the observations are the ones of the first cluster, plus the two of the other two clusters.

### A.H.C. based on Manhattan distance and Ward’s linkage method

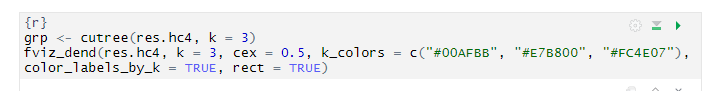


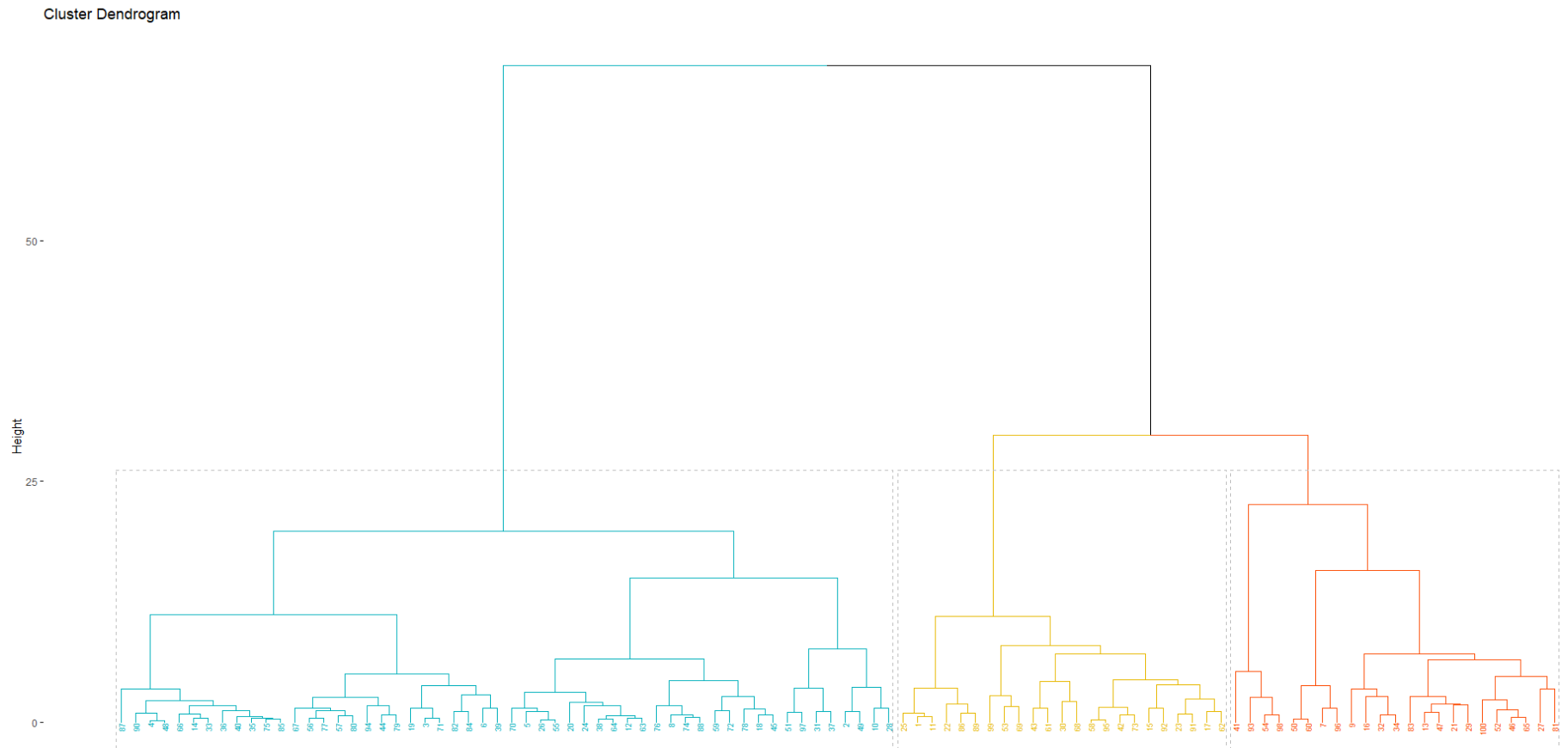


we have to compare the cophenetic distance to the original one



The correlation between the two distances is near average: 0.70



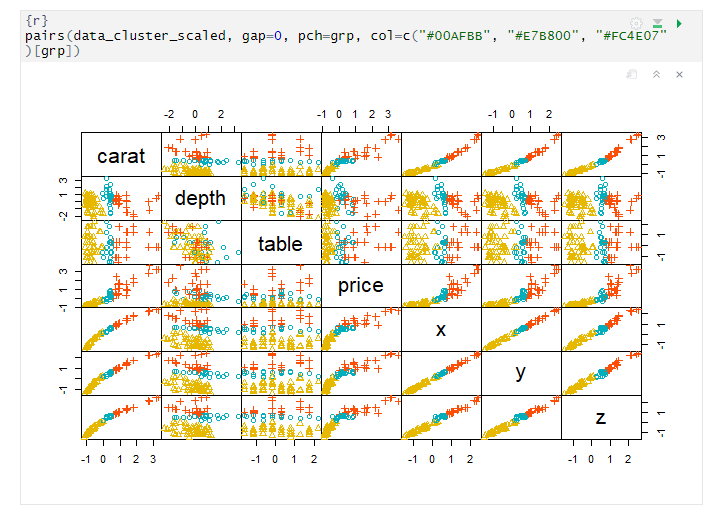


As regard to the clusters size:

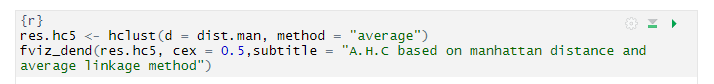


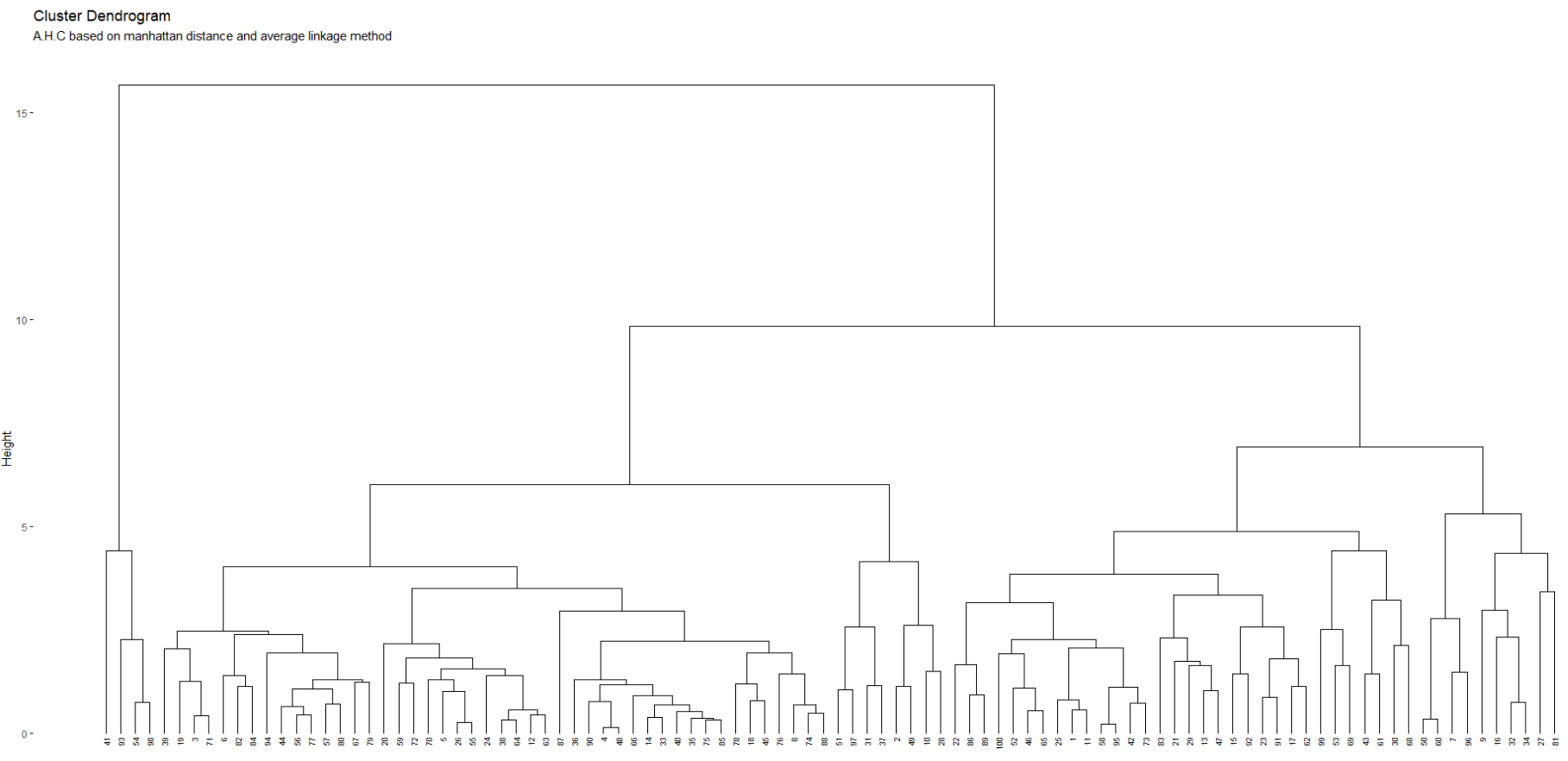
In this case the biggest cluster is the second one, which contains 54 units.

This is evident also by the clustering results in the original space

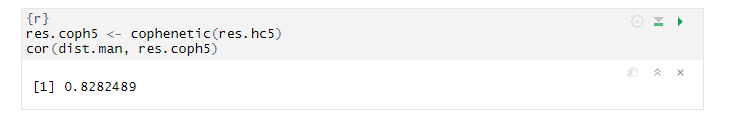


### A.H.C. based on Manhattan distance and Average linkage method

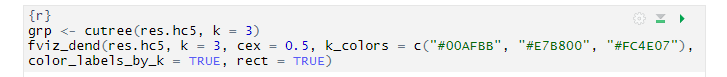


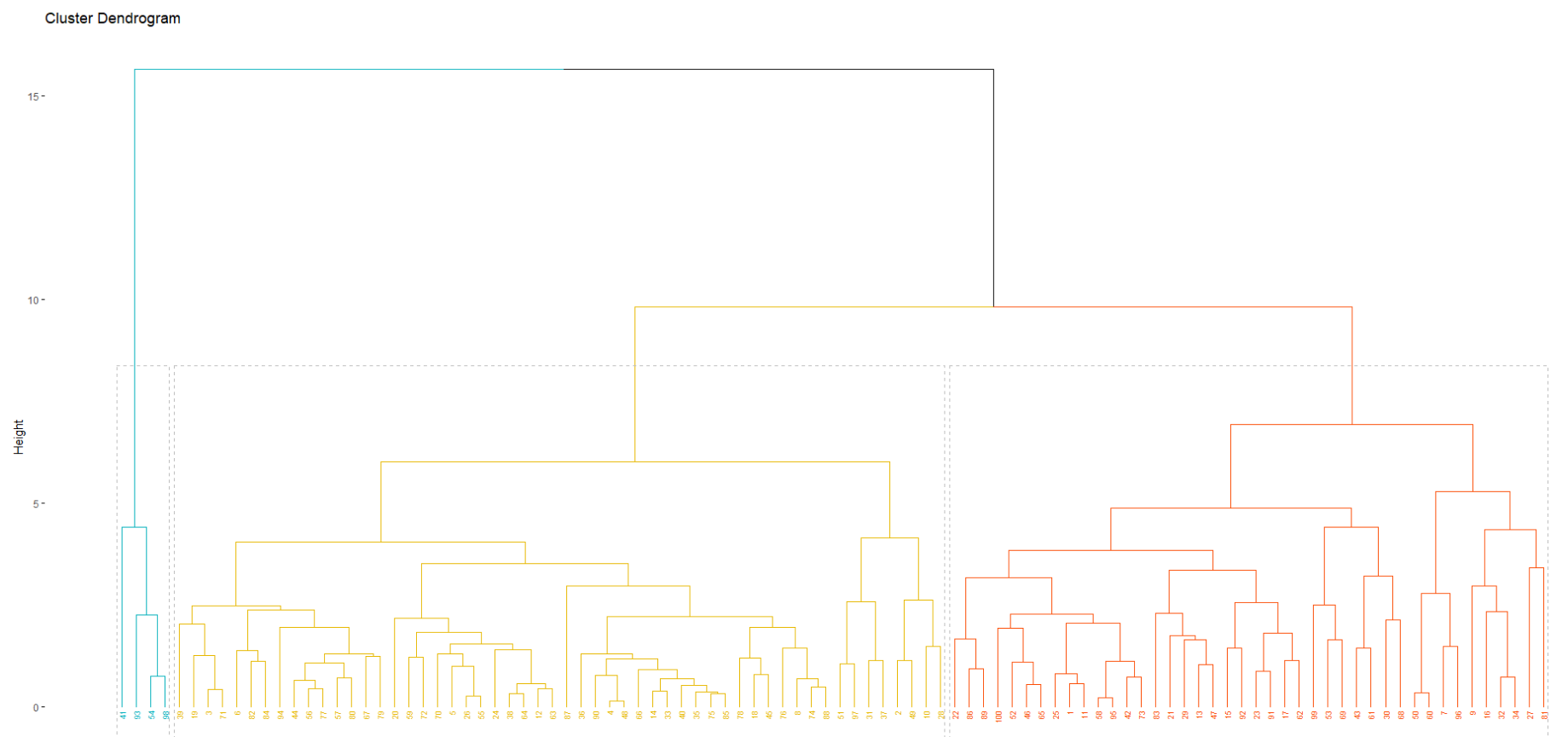


Let’s check the correlation between the cophenetic distance and the original one:

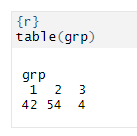


The result is 0.82, which is above average. So, this clustering method preserves the true original distances between units quite well.



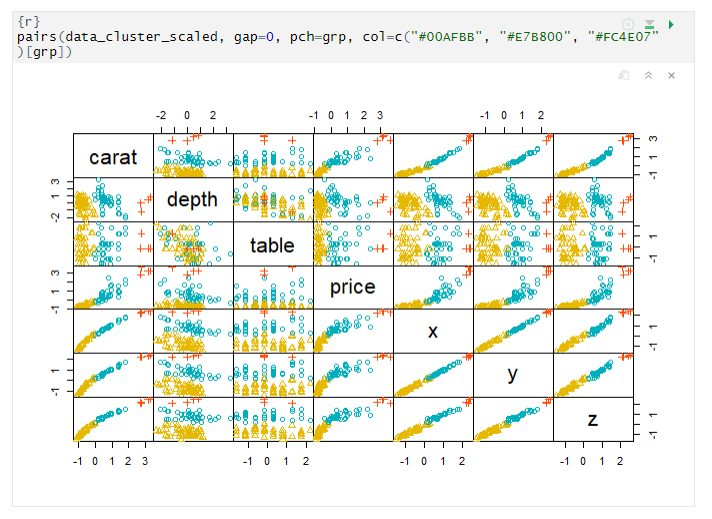


In this case the second cluster is the bigger one, as we can see from this R code

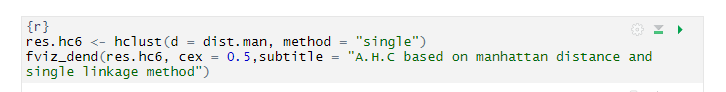


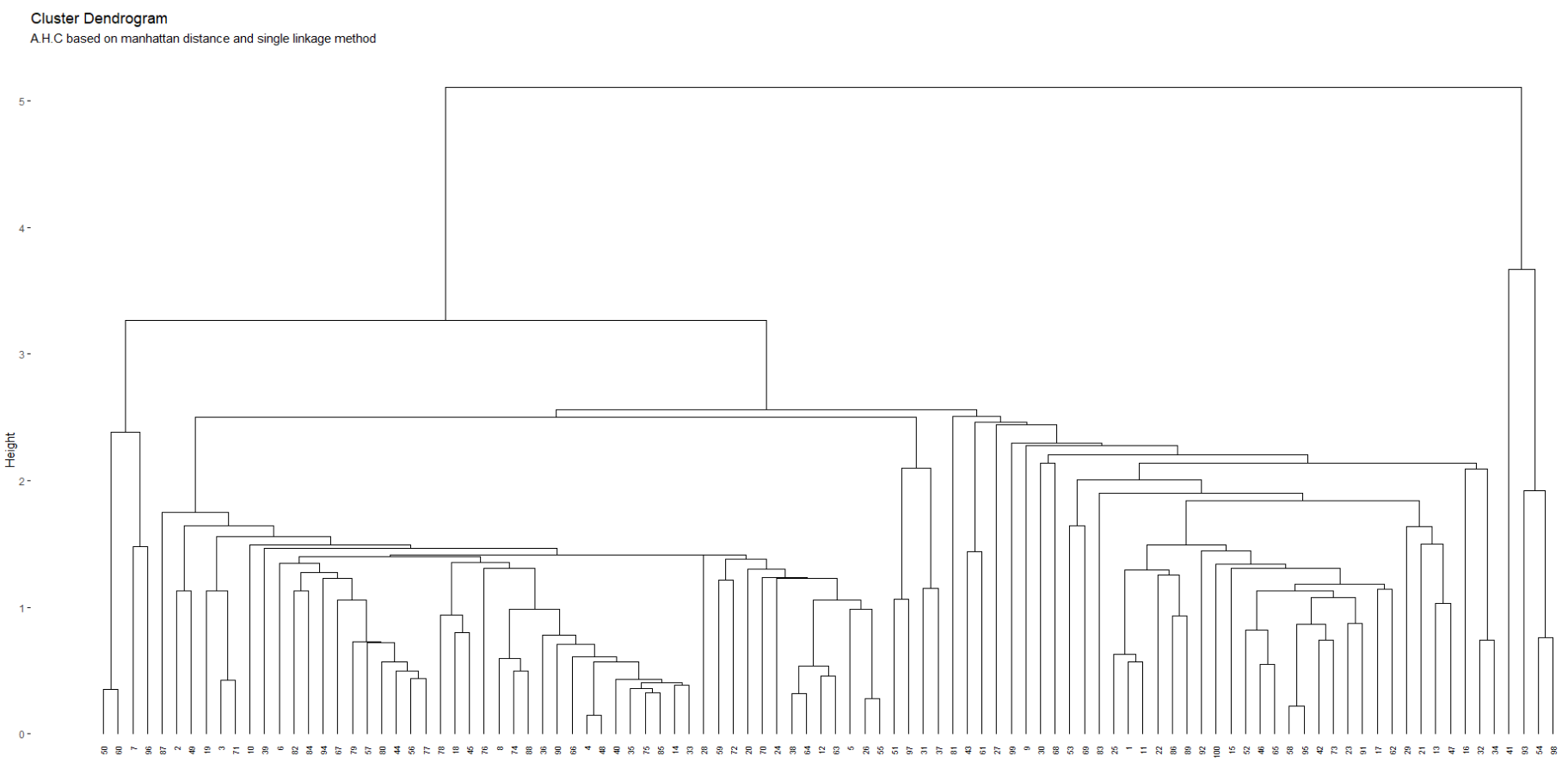
Cluster 2 contains 54 units; cluster 1 contains 42 unit and cluster 3 contains 4 units.

This is evident also by the clustering results in the original space

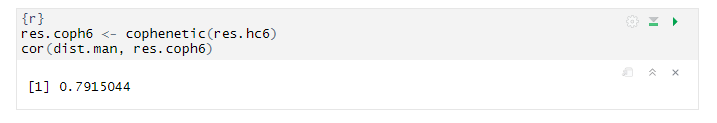


### A.H.C. based on Manhattan distance and Single linkage method



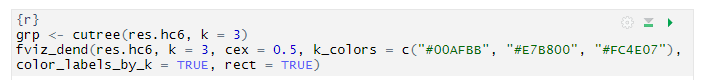


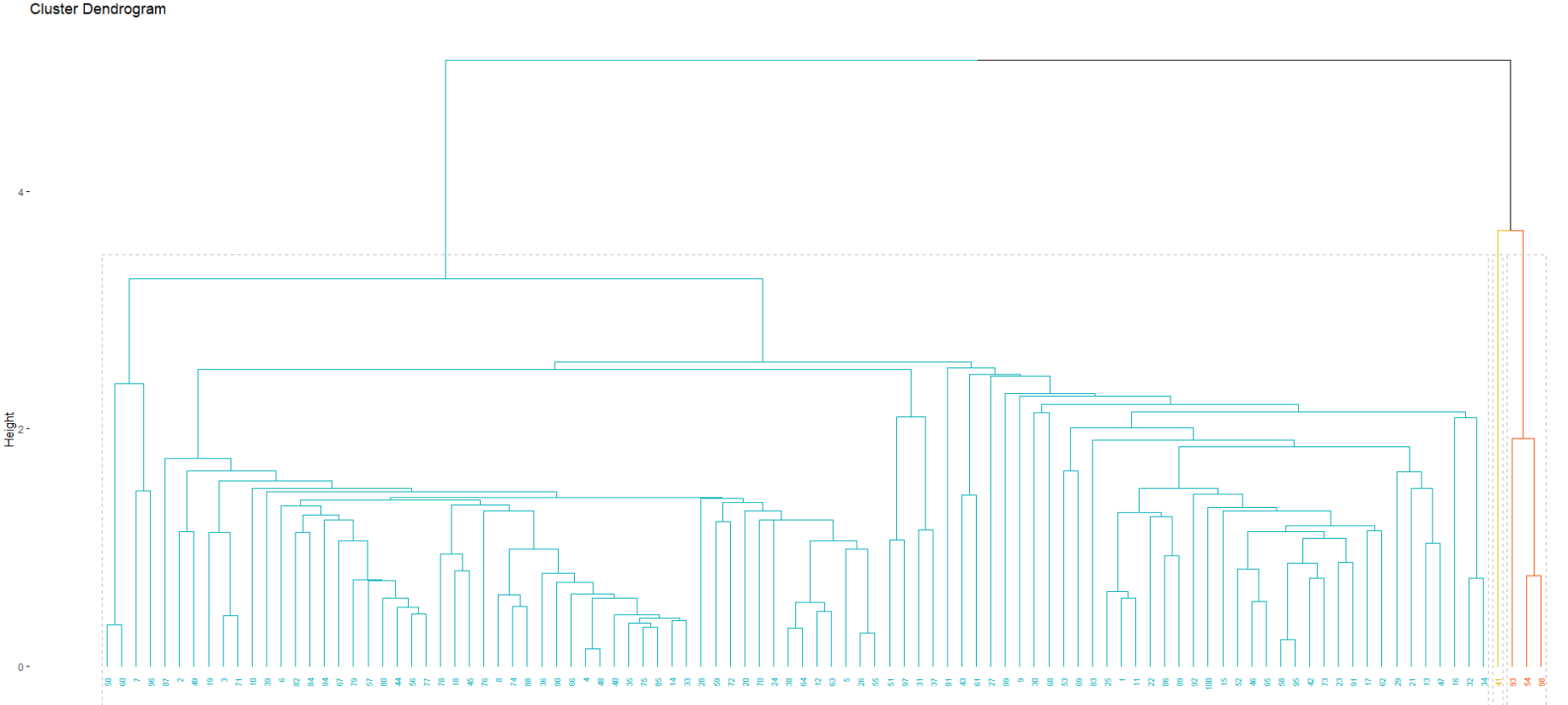
we have to use the cophenetic distance e we have to compare it to the original one, in order to understand if the algorithm is good.



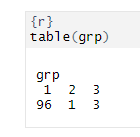
The result is 0.79, which is good enough. So, this clustering method preserves the true original distances between units quite well.

Now Checking for clusters

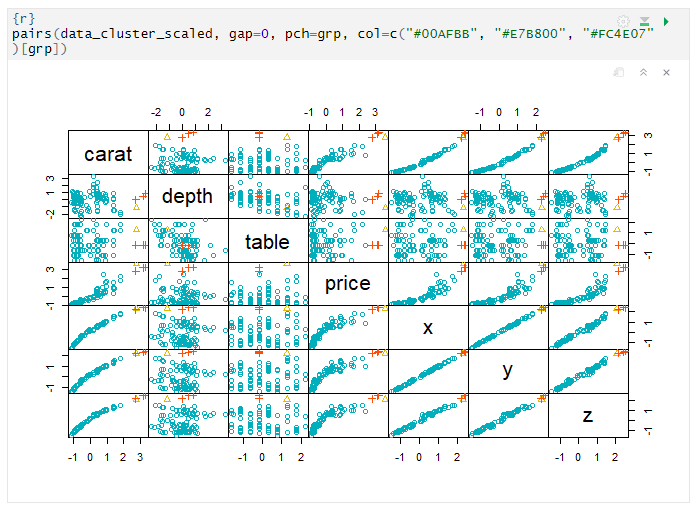




Almost all units are grouped in a single cluster and we can see this also through the R code which gives us the clusters size:



Moreover, if we look at the clustering results in the original space, we can see that in each pairwise of the scatterplot all the observations are the ones of the first cluster, plus the two of the other two clusters

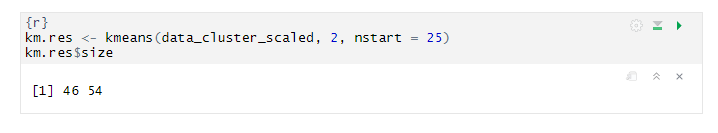


### Concluding best algorithm

After trying all these different combinations, we can conclude that the best combination is the one with the Manhattan distance and the Average linkage method. In fact, if we look at the results of the correlation between the cophenetic distance and the original one for each combination, we see that the highest result was obtained precisely for the above combination.

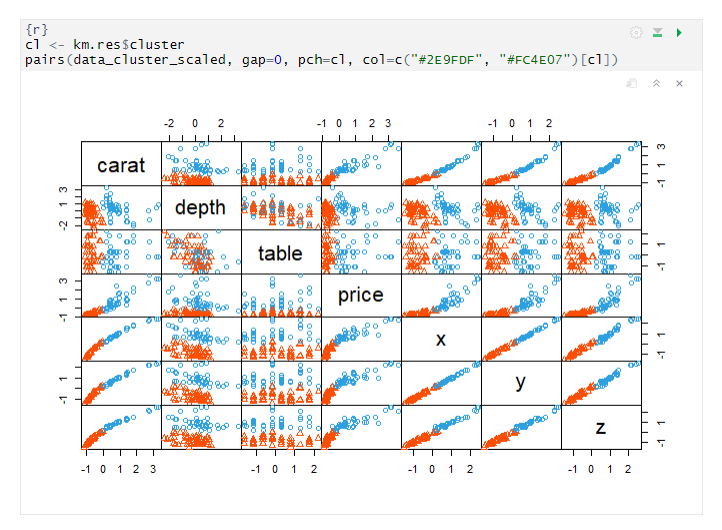
## Partitioning Clustering

### K-Means: k=2:



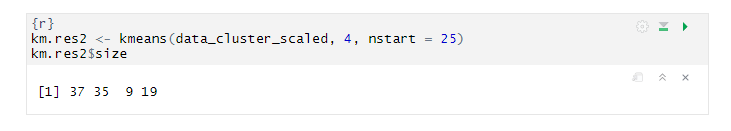
Running this k-means algorithm we obtain two clusters: the second is the bigger one, containing 54 units, while the first is the smallest with 46 units.

We can visualize the cluster results in the original space



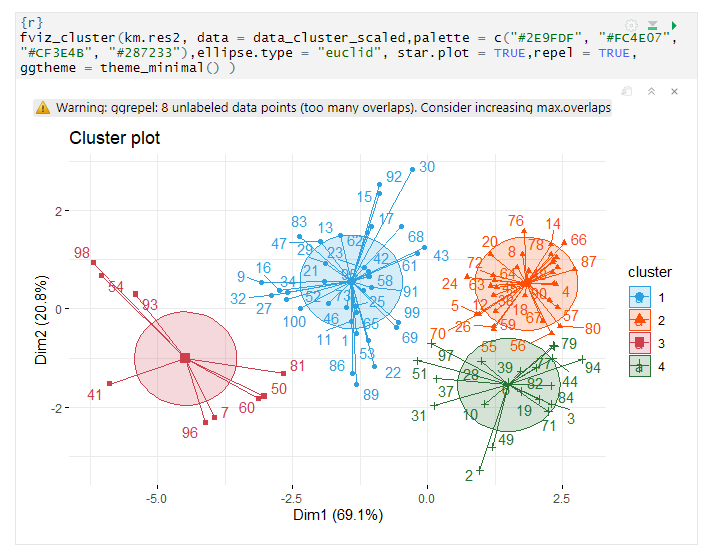
From this plot we can understand what variables are useful to find clusters. For example, looking at the scatterplot between depth and table, we see a high overlap between clusters. We also see that most of the variable are good for the clustering because we can see the separation between clusters.

### K-Mean: k=4

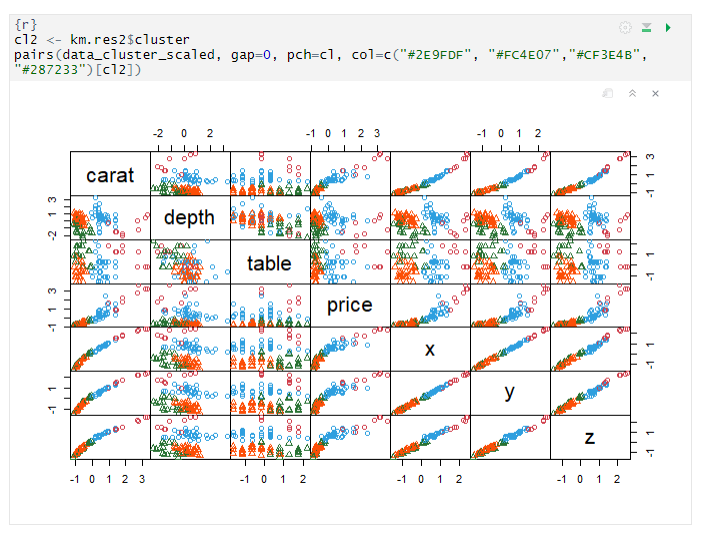


This time we have 4 clusters, the largest being the first one, as we can see by the R code above.

Let’s visualize clustering results in the space spanned by the first two principal components

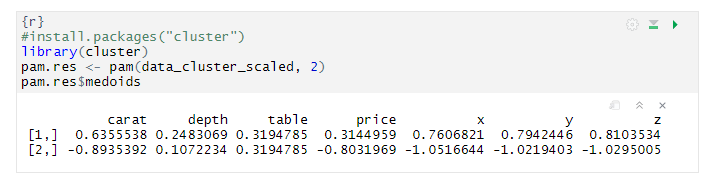


We can see that separation between clusters is not so clear in some of the case, because some units are closer to a different cluster than the one, they belong to – as in the case of 55, 56 which is closer to cluster 4. This is due, precisely, to the fact that the clustering results are visualized in the PCA space, but clusters were searched in the original space, hence it is better to look at the results in the latter (as the previous case)

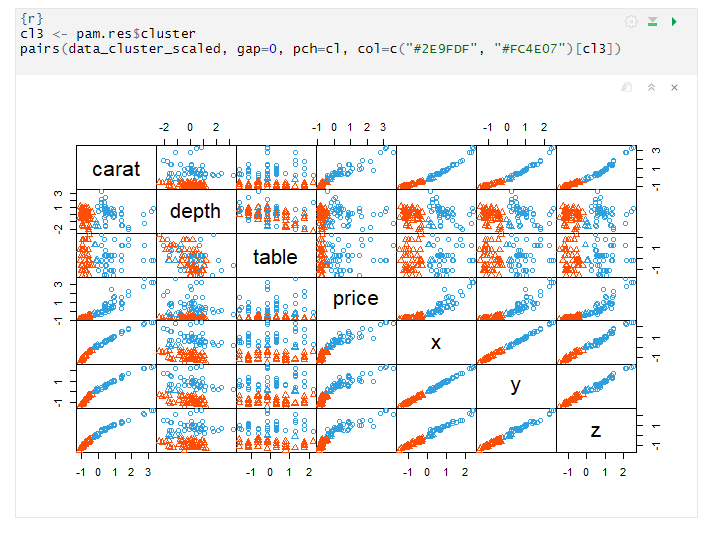


As regard to this plot, the considerations are more or less the same as for the algorithm with 2 clusters. However, in this case, with 4 clusters, it seems a little bit difficult to see separation between clusters, which appear mostly overlapped each other. In fact, as we have already seen, in the space spanned by the first 2 PCs several units are closer to a different cluster than the one they belong to.

### K-Medoids: k=2



Here we can see the medoids, namely the representative point, of both clusters: each column represents their both clusters.



This is the graphical visualization of clustering results in the original space. Looking at the scatterplot between pairs of variables we can understand which are more useful to identify clusters. For example, looking at the scatterplot between depth and table, we see a high overlap between clusters. We also see that most of the variable are good for the clustering because we can see the separation between clusters.

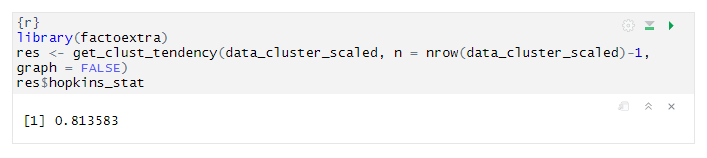
## Cluster Validation

Before applying any clustering method on your data, it’s important to evaluate whether the data sets contain meaningful clusters (i.e.: non-random structures) or not. If yes, then how many clusters are there. This process is defined as the assessing of clustering tendency or the feasibility of the clustering analysis.

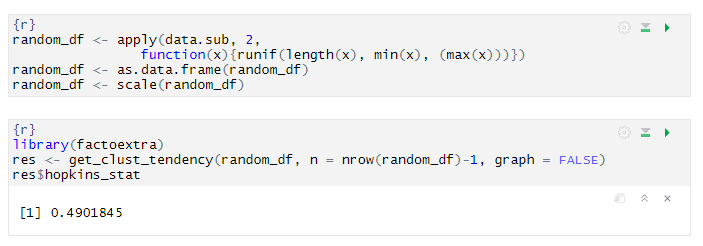
A big issue, in cluster analysis, is that clustering methods will return clusters even if the data does not contain any clusters. In other words, if you blindly apply a clustering method on a data set, it will divide the data into clusters because that is what it supposed to do.

So, let’s check for the clustering tendency. We can assess if there are clusters from both a statistical point of view and from a graphical one, using, respectively, the Hopkins statistic and the VAT algorithm.

### Hopkins Statistic



Computing Hopkins statistic for the random dataset



It can be seen that the data set is highly cluster able (the H value = 0.81 which is far above the threshold 0.5). However, the random\_df data set is not cluster able (H = 0.49)

### VAT Algorithm

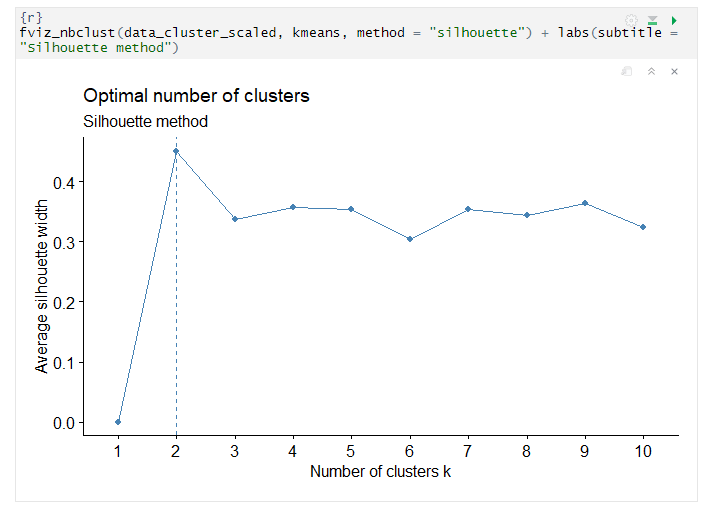
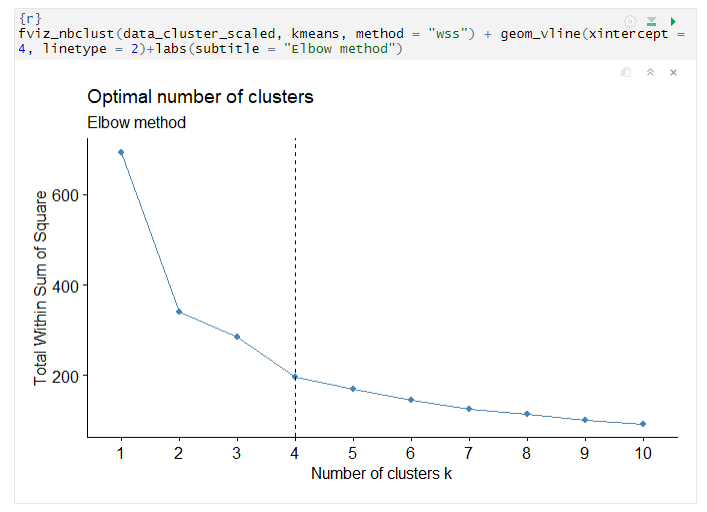


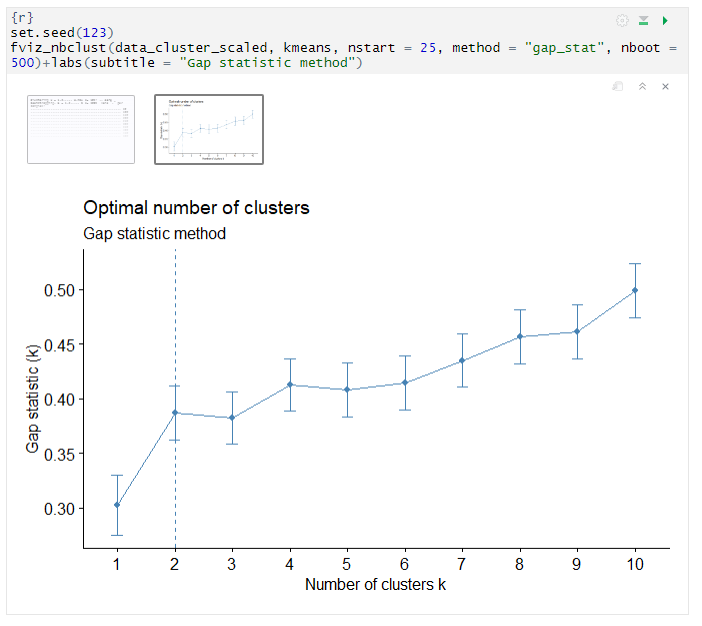


The dissimilarity matrix image confirms that there is a cluster structure in the standardized dataset, but not in the randomized one.

**Determining the optimal number of clusters:**

There are Several methods can be used for determining the optimal number of clusters. Using the fviz\_nbclust function we obtain the following results





In order to find the optimal number of clusters, three indices were used. The elbow method suggested 3 or 4 numbers of the cluster. Silhouette method suggests 2 clusters. Gap statistics 2 clusters, i.e., there are two clusters in the data. It was decided to proceed by identifying 2 clusters.

Using the NbClust function we obtain the following results:

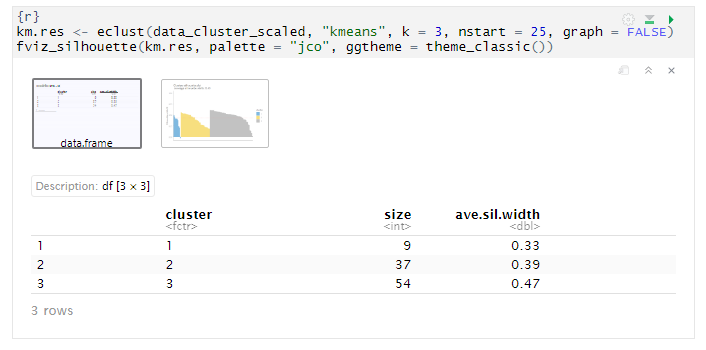


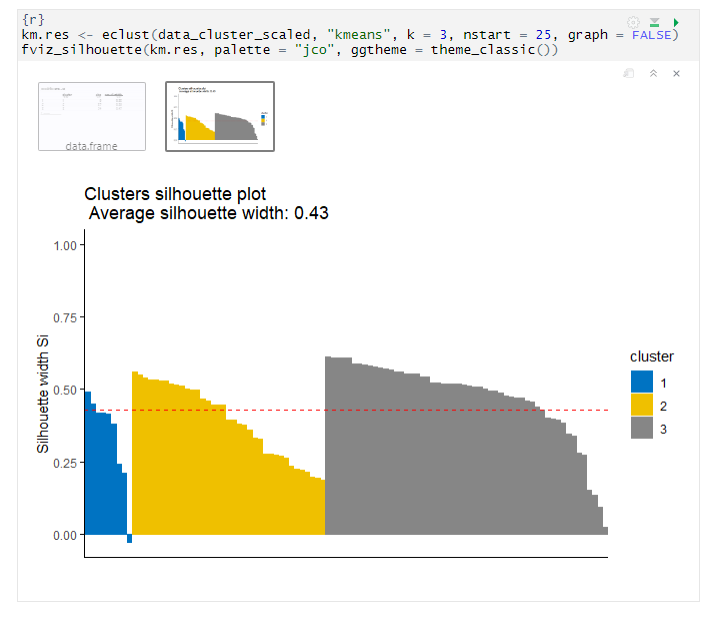
In this case the best solution, according to 9 indices, is K=3; the second-best solution, according to 7 indices, is K=2.

## Cluster Statistics

Cluster statistics are used to evaluate the goodness of clustering results. It can be categorized into 3 classes: internal, external and relative cluster validation. The internal cluster validation allows us to estimate the optimal number of clusters and to select the appropriate clustering algorithm, using two indices: the Silhouette Width and the Dunn index.

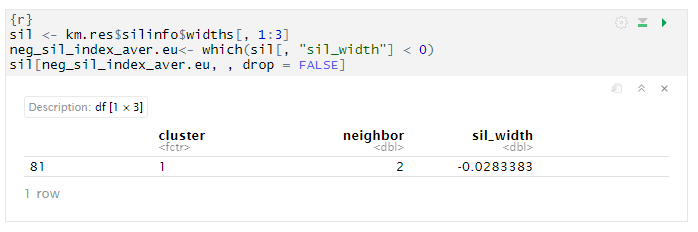
### Silhouette Width





The average silhouette width is 0.43. The third cluster participates in the overall silhouette width with a greater weight because there are more units in it. Looking at the plot, we can see that in cluster 3 most of the units have a silhouette width higher than the average one; in cluster 1 there are few units but stills its higher than the average and some have a negative silhouette width; in cluster 2 we have lesser unit if we compare with 3 cluster.

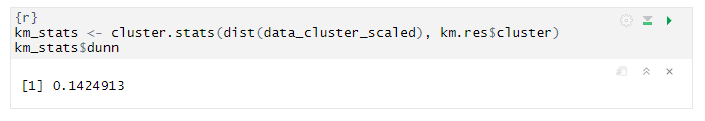
As regards the units with a negative silhouette value, this means that they are not in the right cluster, so we have to find their neighbor clusters:



According to the silhouette method, 81 should be in cluster 2.

### Dunn Index

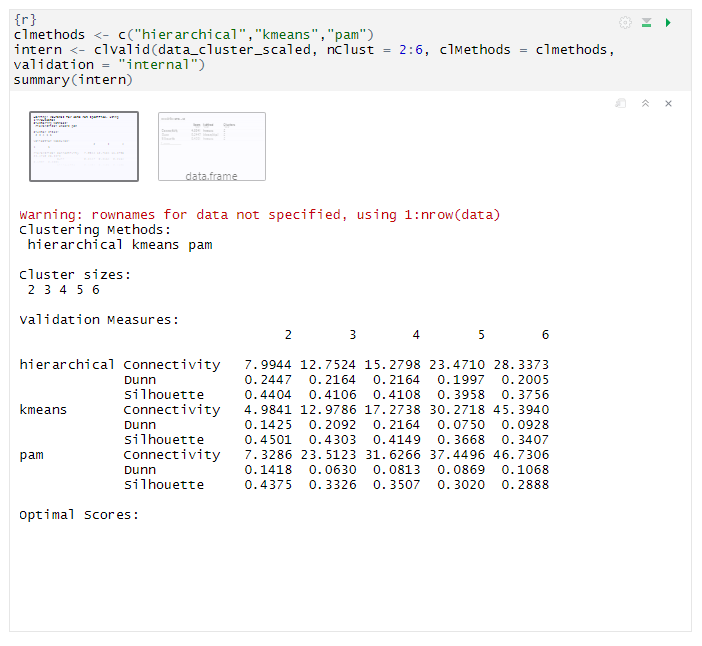
The Dunn index aims to identify sets of clusters that are compact, with a small variance between members of the cluster, and well separated. A higher Dunn index indicates better clustering, so it should be maximized.

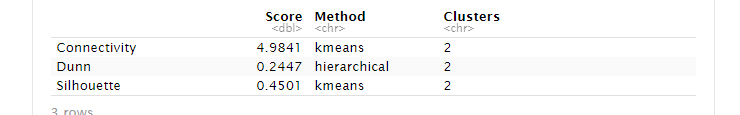


## Choosing the best clustering algorithms.

To choose the best pair of clustering algorithms and an optimal number of clusters, we can use both internal measures and stability ones.

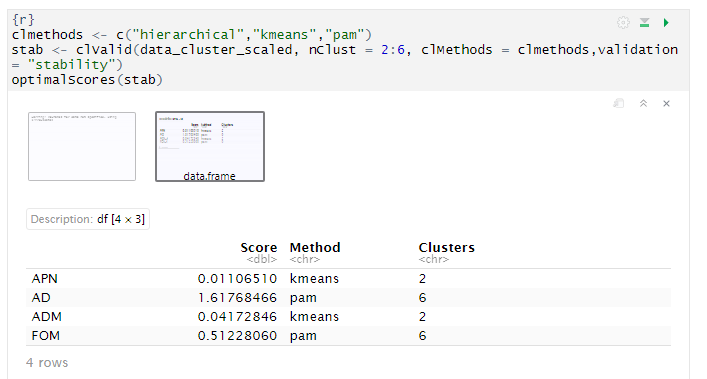
### Internal Measures





We are comparing 3 clustering methods, hierarchical, k-means and pam, using 3 indices as connectivity, Dunn and silhouette width. All of methods having the same cluster count which is k = 2.

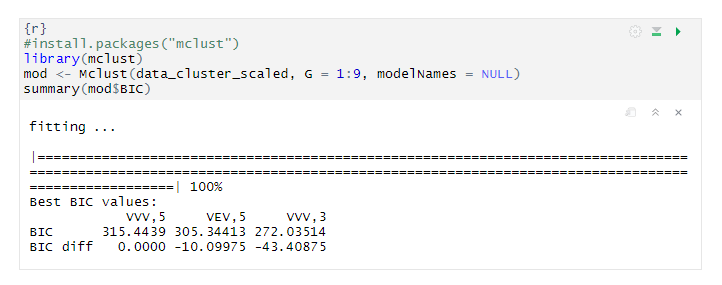
### Stability Measures



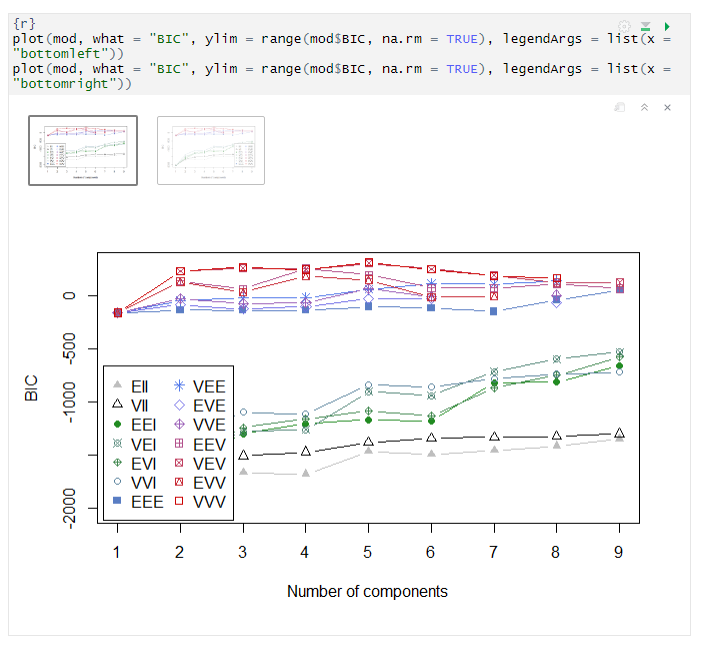
For APN and ADM measures, K means clustering with K = 2 clusters again gives  
the best score. For the other measures, PAM with K = 6 clusters has the best score.

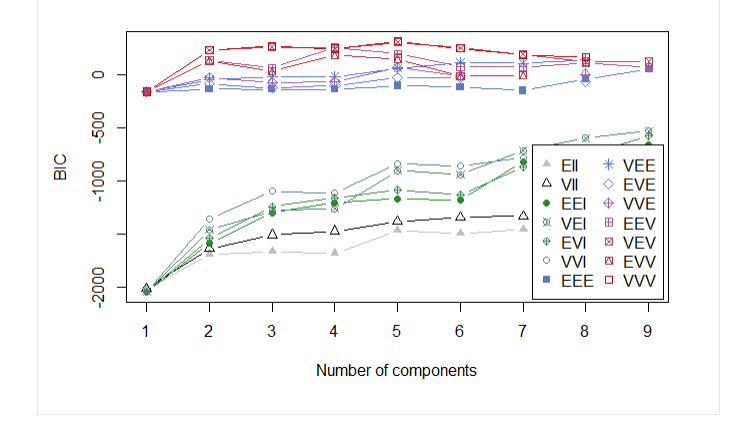
## Model Base Clustering

The model-based clustering is based on statistical models, thus trying to solve the disadvantage of traditional clustering methods, which are heuristic and not based on formal models. Now, we are going to implement the Model-Based Clustering via parsimonious Gaussian mixtures models. Gaussian mixtures are the ones most used, because they can model several density functions, by means of the suitable parameters, which are automatically estimated by the function used to fit the model. Moreover, we are referring to "parsimonious configurations" in the sense that we want to explain the reality by using a simple model, namely a model with a low number of parameters, according to the parsimony criterion. However, mixture models that use Gaussian distribution are complex ones, so we are in need to define a parsimonious version, reducing the number of parameters by means of the eigen-decomposition. Basically, we must find the right compromise between the fit of the model and the number of parameters (that is, the complexity of the model): comparing two criteria, if the increase in the fit is small, it is not worth choosing the more complex model. It is something related to the concept of penalization: the better the model fits the data, the higher the number of parameters it uses, which does not respect the parsimony criterion. Therefore, in some sense, the fitting of the model has to be penalized, taking into account the number of parameters, until, as already said, the right compromise is found. Finally, we will select the number of clusters and the best parsimonious configuration, by means of a selection criterion, for example the Bayesian Information Criterion (BIC):

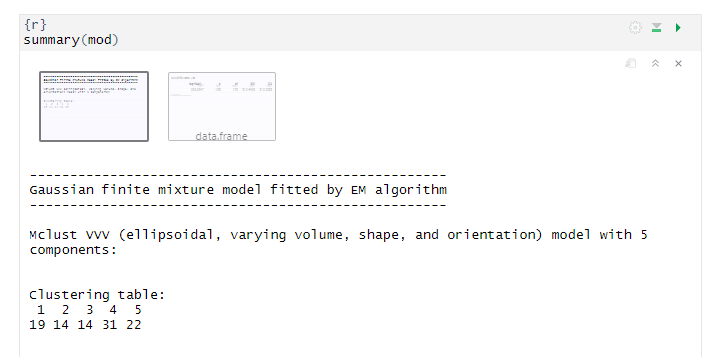


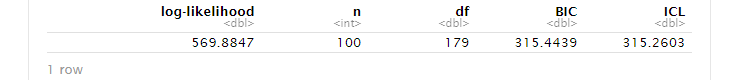
Here we have the best 3 models according to the Bayesian Information Criterion (BIC): the best model is VVV with 5 clusters; the second best is VEV with 5 clusters and the last model is VVV with 3 clusters.





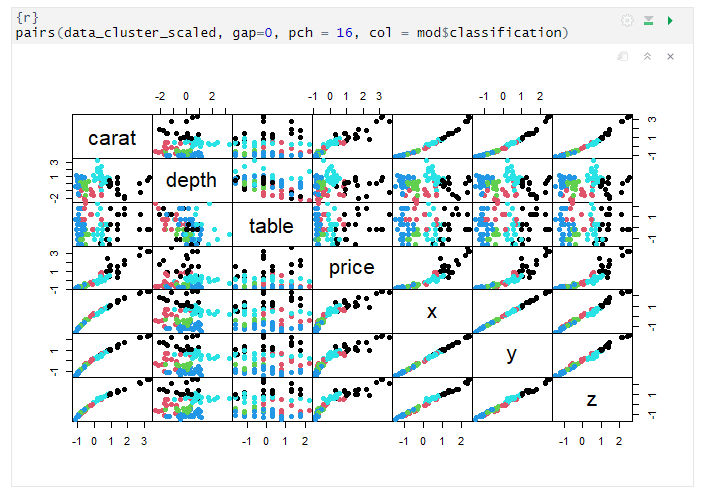
In this plot, which is the graphical counterpart of the results of the previous code, we have the BIC curves for the penalized models; we have a different curve for each number of clusters and for each parsimonious configuration we have a different symbol. The maximum in this configuration is related to 5 clusters and VVV model.





Here we can see that BIC = 315.4439 is the best value for BIC, which must be maximized. Moreover, according to this best model, VVV, the first cluster is characterized by 19 observations, the second one by 14 then 14 then 31 and the last is 22.

Finally, we can graphically visualize the clustering results:



Colors arise by the best model-based clustering model, VVV with 5 clusters. So, we can see that there is a clustering structure also by the result of Model-Based Clustering.