liverDisease <- read.csv("Liver\_Disease.csv")

names(liverDisease)[names(liverDisease) == "Dataset"] <- "Liver\_Disease"

liverDisease$Liver\_Disease[liverDisease$Liver\_Disease == "1"] <- 0  
  
liverDisease$Liver\_Disease[liverDisease$Liver\_Disease == "2"] <- 1

liverDisease$Gender[liverDisease$Gender == "Female"] <- 0  
  
liverDisease$Gender[liverDisease$Gender == "Male"] <- 1

liverDisease$Gender <- factor(liverDisease$Gender)  
  
liverDisease$Liver\_Disease <- as.numeric(liverDisease$Liver\_Disease)  
liverDisease$Liver\_Disease <- factor(liverDisease$Liver\_Disease)

liver <- na.omit(liverDisease)

# Principal Component Analysis

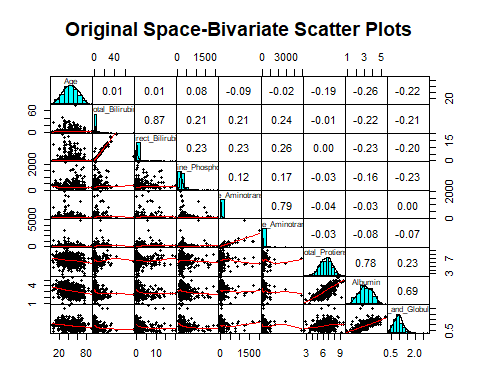
Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process. Before proceeding with the PCA, it is necessary to evaluate whether there is correlation between the numerical variables. To accomplish this, we’ll need a subset of the dataset that contains all continuous values.

liver\_sub <- liver[ -c(2,11) ]

library(psych)

## Warning: package 'psych' was built under R version 4.0.5

pairs.panels(liver\_sub, main= "Original Space-Bivariate Scatter Plots",  
 ellipses = FALSE, gap = 0)



This is a preliminary analysis of the data in the original space, which aim is to understand if it would be useful to run a Principal Component Analysis and a Cluster one on this dataset. In fact, in the upper triangle of the matrix there are the coefficients of correlation between variables, which are used to understand if PCA is useful or not, while in the lower triangle there are the scatterplots of data and on the main diagonal there is the non-parametric density of the data, both used to understand if Component analysis could be useful or not. Specifically, if we look at the plot, we can see that there is a high correlation between some variables, for example between Age and Total\_Bilirubin, which is 0.87 have a strongest positive correlation, Alamine\_Aminotransferase and Aspartate\_Aminotransferase, which is 0.79 having a strongest positive correlation, Total\_Protiens and Albumin having a strongest positive correlation of 0.78 and also Albumin and Albumin\_and\_Globulin\_Ratio having a correlation of 0.69. This means that a PCA in this dataset could really be useful, in fact we could create a linear combination between the variables and express them through a single variable. As regard to the diagonal panels, they are used to understand if the single variable is useful for clustering: if the density line is bi-model, the relative variable could be useful for clustering, otherwise not. In this case, each variable separately considered is not useful to see clusters, but, if we look at the pairwise of variables, it is useful. In fact, from the scatterplots of the data in the lower triangle we can see that there are clusters, because the data are grouped along the diagonal instead of remaining scattered in space.

## Prepare the Data

In order to evaluate the difference between the variables, the mean and the variance are computed for each variable.

apply(liver\_sub, 2, mean)

## Age Total\_Bilirubin   
## 44.7823834 3.3153713   
## Direct\_Bilirubin Alkaline\_Phosphotase   
## 1.4941278 291.3661485   
## Alamine\_Aminotransferase Aspartate\_Aminotransferase   
## 81.1260794 110.4145078   
## Total\_Protiens Albumin   
## 6.4816926 3.1385147   
## Albumin\_and\_Globulin\_Ratio   
## 0.9470639

apply(liver\_sub, 2, var)

## Age Total\_Bilirubin   
## 2.631463e+02 3.878445e+01   
## Direct\_Bilirubin Alkaline\_Phosphotase   
## 7.932664e+00 5.932238e+04   
## Alamine\_Aminotransferase Aspartate\_Aminotransferase   
## 3.355595e+04 8.401304e+04   
## Total\_Protiens Albumin   
## 1.176446e+00 6.311265e-01   
## Albumin\_and\_Globulin\_Ratio   
## 1.021391e-01

There is a great difference in the variables. It is preferable to normalize a variable in order to have a zero mean and uniform variance when working with homogeneous variables.

scaled\_liver <- apply(liver\_sub, 2, scale)  
head(scaled\_liver)

## Age Total\_Bilirubin Direct\_Bilirubin Alkaline\_Phosphotase  
## [1,] 1.24632497 -0.41995671 -0.4949862 -0.4284995  
## [2,] 1.06138848 1.21788279 1.4222880 1.6736358  
## [3,] 1.06138848 0.63982179 0.9252169 0.8155376  
## [4,] 0.81480651 -0.37178496 -0.3884709 -0.4490282  
## [5,] 1.67784343 0.09387529 0.1796103 -0.3956537  
## [6,] 0.07506058 -0.24332696 -0.2819557 -0.3422792  
## Alamine\_Aminotransferase Aspartate\_Aminotransferase Total\_Protiens  
## [1,] -0.35552499 -0.31883559 0.2934680  
## [2,] -0.09349172 -0.03593068 0.9388428  
## [3,] -0.11532783 -0.14633260 0.4778608  
## [4,] -0.36644304 -0.31193547 0.2934680  
## [5,] -0.29547570 -0.17738313 0.7544500  
## [6,] -0.33914791 -0.33263583 1.0310392  
## Albumin Albumin\_and\_Globulin\_Ratio  
## [1,] 0.20327072 -0.1472624  
## [2,] 0.07739506 -0.6479006  
## [3,] 0.20327072 -0.1785523  
## [4,] 0.32914639 0.1656364  
## [5,] -0.92961029 -1.7117566  
## [6,] 1.58790307 1.1043330

## Computing PCs

In order to find the Principal Components, the Eigen decomposition is applied to the covariance matrix of the standardized data.

liver\_cov <- cov(scaled\_liver)  
liver\_eigen<-eigen(liver\_cov)  
liver\_eigen$value

## [1] 2.75391157 2.02686232 1.36548027 0.95799983 0.84466046 0.66638940 0.20344650  
## [8] 0.12581520 0.05543444

The eigen vectors of the PCs are displayed as an example:

phi <- liver\_eigen$vectors[,1:3]  
phi <- -phi  
row.names(phi) <- c("Age", "Total\_Bilirubin", "Direct\_Bilirubin", "Alkaline\_Phosphotase", "Alamine\_Aminotransferase", "Aspartate\_Aminotransferase", "Total\_Protiens", "Albumin", "Albumin\_and\_Globulin\_Ratio")  
colnames(phi) <- c("PC1", "PC2", "PC3")  
phi

## PC1 PC2 PC3  
## Age 0.1401403 0.28574605 -0.01471618  
## Total\_Bilirubin 0.4134106 -0.25254722 0.45274491  
## Direct\_Bilirubin 0.4195284 -0.26370362 0.44042614  
## Alkaline\_Phosphotase 0.2530276 -0.05438827 0.09926171  
## Alamine\_Aminotransferase 0.2671770 -0.41670205 -0.50458906  
## Aspartate\_Aminotransferase 0.3005978 -0.39339220 -0.48188882  
## Total\_Protiens -0.2775673 -0.41943858 0.29778888  
## Albumin -0.4400519 -0.43099512 0.11470573  
## Albumin\_and\_Globulin\_Ratio -0.3701262 -0.30329504 -0.04811273

By examining the loading we note that first loading vector phi 1 puts most of its weight on Direct\_Bilirubin (0.419) and much less weight on Albumin (-0.440). The second loading vector phi 2 puts most of its weight on Age (0.285) and much less weight on Albumin (-0.430). The third loading vector phi 3 puts most of its weight on Total\_Bilirubin (0.452) and much less weight on Alamine\_Aminotransferase (-0.504).

## Principal Component Scores

PC1 <- scaled\_liver %\*% phi[,1]  
PC2 <- scaled\_liver %\*% phi[,2]  
PC3 <- scaled\_liver %\*% phi[,3]  
PC <- data.frame(ID = row.names(liver), PC1, PC2, PC3)  
head(PC)

## ID PC1 PC2 PC3  
## 1 1 -0.6222677 0.7235652 -0.01818250  
## 2 2 1.5817709 -0.6479197 1.71242776  
## 3 3 0.7769593 -0.2749009 1.06541563  
## 4 4 -0.7953810 0.4138056 0.05641998  
## 5 5 0.9501497 1.2261700 0.49260715  
## 6 6 -1.8792566 -1.0037263 0.49803940

library(ggplot2)

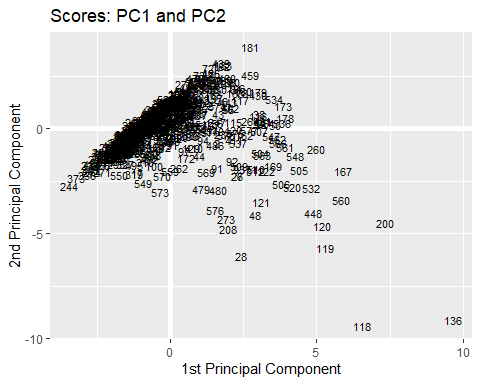
##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

library(modelr)

## Warning: package 'modelr' was built under R version 4.0.5

ggplot(PC, aes(PC1, PC2)) +  
modelr::geom\_ref\_line(h = 0) +  
modelr::geom\_ref\_line(v = 0) +  
geom\_text(aes(label = ID), size = 3) +  
xlab("1st Principal Component") +  
ylab("2nd Principal Component") +  
ggtitle("Scores: PC1 and PC2")



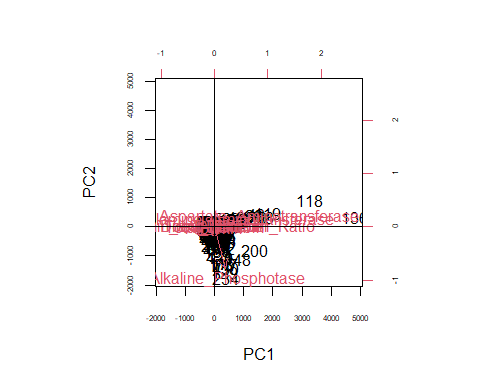
## Biplot

It is possible to visualize the scores and the original variable (represented by arrows) in the space spanned by the first two principal components. we set center= True to shift the variable into zero center as follows:

set.seed(123)  
liver\_pc <- prcomp(liver\_sub, center = TRUE, scale. = FALSE)  
biplot(liver\_pc, cex.axis = 0.5, scale=0)

## Warning in arrows(0, 0, y[, 1L] \* 0.8, y[, 2L] \* 0.8, col = col[2L], length =  
## arrow.len): zero-length arrow is of indeterminate angle and so skipped  
  
## Warning in arrows(0, 0, y[, 1L] \* 0.8, y[, 2L] \* 0.8, col = col[2L], length =  
## arrow.len): zero-length arrow is of indeterminate angle and so skipped  
  
## Warning in arrows(0, 0, y[, 1L] \* 0.8, y[, 2L] \* 0.8, col = col[2L], length =  
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

abline(h=0)  
abline(v=0)



The angle between the arrows gives information on the correlation between the two variables.

cor(liver\_sub)

## Age Total\_Bilirubin Direct\_Bilirubin  
## Age 1.000000000 0.011000374 6.784303e-03  
## Total\_Bilirubin 0.011000374 1.000000000 8.744810e-01  
## Direct\_Bilirubin 0.006784303 0.874480969 1.000000e+00  
## Alkaline\_Phosphotase 0.078878350 0.205739173 2.340076e-01  
## Alamine\_Aminotransferase -0.087799162 0.213375493 2.331801e-01  
## Aspartate\_Aminotransferase -0.020498946 0.237323055 2.570224e-01  
## Total\_Protiens -0.186248122 -0.007905923 3.270877e-05  
## Albumin -0.264210935 -0.222086570 -2.284092e-01  
## Albumin\_and\_Globulin\_Ratio -0.216408346 -0.206267186 -2.001247e-01  
## Alkaline\_Phosphotase Alamine\_Aminotransferase  
## Age 0.07887835 -0.08779916  
## Total\_Bilirubin 0.20573917 0.21337549  
## Direct\_Bilirubin 0.23400757 0.23318008  
## Alkaline\_Phosphotase 1.00000000 0.12477671  
## Alamine\_Aminotransferase 0.12477671 1.00000000  
## Aspartate\_Aminotransferase 0.16657999 0.79186215  
## Total\_Protiens -0.02706202 -0.04243210  
## Albumin -0.16341865 -0.02865750  
## Albumin\_and\_Globulin\_Ratio -0.23416650 -0.00237499  
## Aspartate\_Aminotransferase Total\_Protiens  
## Age -0.02049895 -1.862481e-01  
## Total\_Bilirubin 0.23732305 -7.905923e-03  
## Direct\_Bilirubin 0.25702239 3.270877e-05  
## Alkaline\_Phosphotase 0.16657999 -2.706202e-02  
## Alamine\_Aminotransferase 0.79186215 -4.243210e-02  
## Aspartate\_Aminotransferase 1.00000000 -2.575101e-02  
## Total\_Protiens -0.02575101 1.000000e+00  
## Albumin -0.08491457 7.831122e-01  
## Albumin\_and\_Globulin\_Ratio -0.07003983 2.348872e-01  
## Albumin Albumin\_and\_Globulin\_Ratio  
## Age -0.26421094 -0.21640835  
## Total\_Bilirubin -0.22208657 -0.20626719  
## Direct\_Bilirubin -0.22840915 -0.20012469  
## Alkaline\_Phosphotase -0.16341865 -0.23416650  
## Alamine\_Aminotransferase -0.02865750 -0.00237499  
## Aspartate\_Aminotransferase -0.08491457 -0.07003983  
## Total\_Protiens 0.78311217 0.23488718  
## Albumin 1.00000000 0.68963234  
## Albumin\_and\_Globulin\_Ratio 0.68963234 1.00000000

To select the number of principal components, three heuristic methods are proposed as follows:

### Cumulative Proportion of Variance Explained (CPVE)

According to this approach, the first q principal components that explain at least 80% of the total variance are retained.

(PVE <- liver\_eigen$values/sum(liver\_eigen$values))

## [1] 0.305990174 0.225206925 0.151720030 0.106444426 0.093851162 0.074043266  
## [7] 0.022605167 0.013979467 0.006159383

* The First PC explains 30.59% of the variability.
* The Second PC explains 22.52% of the variability.
* The Third PC explains 15.17% of the variability.
* The Fourth PC explains 10.64% of the variability.
* The Fifth PC explains 9.38% of the variability.
* The Sixth PC explains 7.40% of the variability.
* The Seven PC explains 2.26% of the variability.
* The Eight PC explains 1.39% of the variability.
* The Ninth PC explains 0.61% of the variability.

cumsum(PVE)

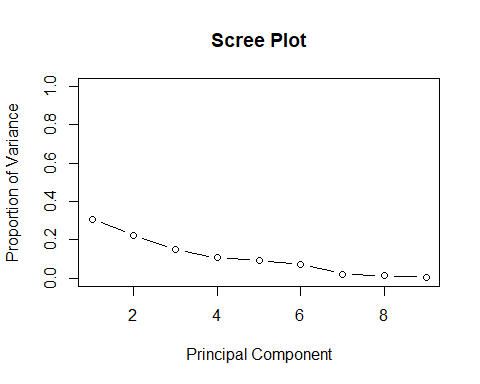
## [1] 0.3059902 0.5311971 0.6829171 0.7893616 0.8832127 0.9572560 0.9798612  
## [8] 0.9938406 1.0000000

According to this method, the first five principal components are retained because together they explain the 88% of the total variance.

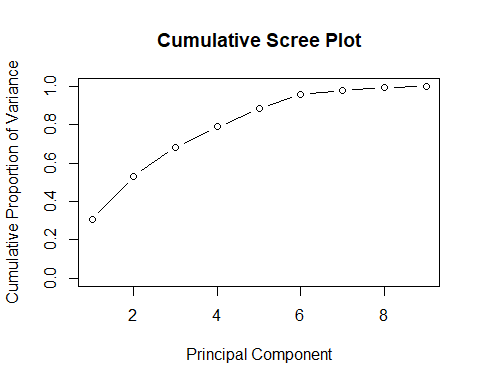
### Scree Plot

The scree plot shows the value of q that matches the value of m when the curve falls flat.

plot(PVE, xlab="Principal Component", ylab="Proportion of Variance", main="Scree Plot", ylim=c(0,1), type='b')



plot(cumsum(PVE), xlab="Principal Component", main="Cumulative Scree Plot", ylab="Cumulative Proportion of Variance", ylim=c(0,1),type='b')



In the “Proportion of variance Explained” plot, the elbow point in not so clear and it may be at q=2 or q=4 or q=6. However, according to this method, it seems reasonable to retain the first six principal components.

### Kaiser’s Rule

For standardized data, the principal components with a variance greater than one are chosen according to Kaiser’s rule.

liver\_eigen$values

## [1] 2.75391157 2.02686232 1.36548027 0.95799983 0.84466046 0.66638940 0.20344650  
## [8] 0.12581520 0.05543444

The rule of the Kaiser indicates that first three principal component should be maintained.

## PCA Result

I have achieved various results based on various methods. The ‘CPVE’ rule recommends that the first five components are retained, while the Scree plot provides result to retain the first six components but the kaiser’s rule implies that the first three components are maintained. I chose the results of Scree plot since more PCs will be obtained.

# Cluster Analysis

The purpose of the clustering is to locate homogeneous subgroups in the liver dataset and to accomplish this analysis, various methods can be useful. The analysis composed of many different steps. In the first step, a certain sort of distance is computed among pairs and the distance matrix is established. This is because in this kind of analysis, the concept of dissimilarity is important, since the unit most “similar” will be put into the same cluster, while there must be a large dissimilarity between the other clusters.

## Hopkins statistic

liver\_scale <- scale(liver\_sub)

library(clustertend)

## Warning: package 'clustertend' was built under R version 4.0.5

hopkins(liver\_scale, n = nrow(liver\_scale)-1)

## $H  
## [1] 0.09085019

The statistical value of Hopkins is nearly 0. The outcome is clustered data, assuming that the uniform distribution is the configuration without cluster.

## Euclidean Distance

Let us compute the Euclidean distance as follows:

dist.eucl <- dist(liver\_scale, method = "euclidean")  
eucl <- round(as.matrix(dist.eucl)[1:9, 1:9], 2)  
row.names(eucl) <- c("Age", "Total\_Bilirubin", "Direct\_Bilirubin", "Alkaline\_Phosphotase", "Alamine\_Aminotransferase", "Aspartate\_Aminotransferase", "Total\_Protiens", "Albumin", "Albumin\_and\_Globulin\_Ratio")  
colnames(eucl) <- c("Age", "Total\_Bilirubin", "Direct\_Bilirubin", "Alkaline\_Phosphotase", "Alamine\_Aminotransferase", "Aspartate\_Aminotransferase", "Total\_Protiens", "Albumin", "Albumin\_and\_Globulin\_Ratio")  
eucl

## Age Total\_Bilirubin Direct\_Bilirubin  
## Age 0.00 3.41 2.20  
## Total\_Bilirubin 3.41 0.00 1.33  
## Direct\_Bilirubin 2.20 1.33 0.00  
## Alkaline\_Phosphotase 0.56 3.42 2.16  
## Alamine\_Aminotransferase 2.21 3.12 2.54  
## Aspartate\_Aminotransferase 2.34 3.95 2.92  
## Total\_Protiens 2.45 4.15 3.17  
## Albumin 2.34 4.07 3.03  
## Albumin\_and\_Globulin\_Ratio 3.31 4.62 3.76  
## Alkaline\_Phosphotase Alamine\_Aminotransferase  
## Age 0.56 2.21  
## Total\_Bilirubin 3.42 3.12  
## Direct\_Bilirubin 2.16 2.54  
## Alkaline\_Phosphotase 0.00 2.58  
## Alamine\_Aminotransferase 2.58 0.00  
## Aspartate\_Aminotransferase 1.90 4.16  
## Total\_Protiens 1.99 3.78  
## Albumin 1.84 3.88  
## Albumin\_and\_Globulin\_Ratio 2.81 4.79  
## Aspartate\_Aminotransferase Total\_Protiens Albumin  
## Age 2.34 2.45 2.34  
## Total\_Bilirubin 3.95 4.15 4.07  
## Direct\_Bilirubin 2.92 3.17 3.03  
## Alkaline\_Phosphotase 1.90 1.99 1.84  
## Alamine\_Aminotransferase 4.16 3.78 3.88  
## Aspartate\_Aminotransferase 0.00 2.02 1.80  
## Total\_Protiens 2.02 0.00 0.51  
## Albumin 1.80 0.51 0.00  
## Albumin\_and\_Globulin\_Ratio 1.87 1.20 1.21  
## Albumin\_and\_Globulin\_Ratio  
## Age 3.31  
## Total\_Bilirubin 4.62  
## Direct\_Bilirubin 3.76  
## Alkaline\_Phosphotase 2.81  
## Alamine\_Aminotransferase 4.79  
## Aspartate\_Aminotransferase 1.87  
## Total\_Protiens 1.20  
## Albumin 1.21  
## Albumin\_and\_Globulin\_Ratio 0.00

In this symmetric matrix, each value represents the distance between units. The values on the diagonal represent the distance between units and themselves (which is zero).

## Manhattan Distance

Let us compute the Manhattan distance as follows:

dist.man <- dist(liver\_scale, method = "manhattan")  
man <- round(as.matrix(dist.man)[1:9,1:9],2)  
row.names(man) <- c("Age", "Total\_Bilirubin", "Direct\_Bilirubin", "Alkaline\_Phosphotase", "Alamine\_Aminotransferase", "Aspartate\_Aminotransferase", "Total\_Protiens", "Albumin", "Albumin\_and\_Globulin\_Ratio")  
colnames(man) <- c("Age", "Total\_Bilirubin", "Direct\_Bilirubin", "Alkaline\_Phosphotase", "Alamine\_Aminotransferase", "Aspartate\_Aminotransferase", "Total\_Protiens", "Albumin", "Albumin\_and\_Globulin\_Ratio")  
man

## Age Total\_Bilirubin Direct\_Bilirubin  
## Age 0.00 7.66 4.54  
## Total\_Bilirubin 7.66 0.00 3.12  
## Direct\_Bilirubin 4.54 3.12 0.00  
## Alkaline\_Phosphotase 1.06 8.03 4.91  
## Alamine\_Aminotransferase 5.01 7.65 6.27  
## Aspartate\_Aminotransferase 5.05 10.06 7.87  
## Total\_Protiens 3.38 10.16 7.04  
## Albumin 3.51 10.47 7.35  
## Albumin\_and\_Globulin\_Ratio 5.66 11.44 9.06  
## Alkaline\_Phosphotase Alamine\_Aminotransferase  
## Age 1.06 5.01  
## Total\_Bilirubin 8.03 7.65  
## Direct\_Bilirubin 4.91 6.27  
## Alkaline\_Phosphotase 0.00 5.75  
## Alamine\_Aminotransferase 5.75 0.00  
## Aspartate\_Aminotransferase 4.06 8.26  
## Total\_Protiens 2.52 7.89  
## Albumin 2.61 8.26  
## Albumin\_and\_Globulin\_Ratio 4.77 9.41  
## Aspartate\_Aminotransferase Total\_Protiens Albumin  
## Age 5.05 3.38 3.51  
## Total\_Bilirubin 10.06 10.16 10.47  
## Direct\_Bilirubin 7.87 7.04 7.35  
## Alkaline\_Phosphotase 4.06 2.52 2.61  
## Alamine\_Aminotransferase 8.26 7.89 8.26  
## Aspartate\_Aminotransferase 0.00 4.42 3.86  
## Total\_Protiens 4.42 0.00 1.15  
## Albumin 3.86 1.15 0.00  
## Albumin\_and\_Globulin\_Ratio 3.01 2.59 2.40  
## Albumin\_and\_Globulin\_Ratio  
## Age 5.66  
## Total\_Bilirubin 11.44  
## Direct\_Bilirubin 9.06  
## Alkaline\_Phosphotase 4.77  
## Alamine\_Aminotransferase 9.41  
## Aspartate\_Aminotransferase 3.01  
## Total\_Protiens 2.59  
## Albumin 2.40  
## Albumin\_and\_Globulin\_Ratio 0.00

In this symmetric matrix, each value represents the distance between units. The values on the diagonal represent the distance between units and themselves (which is zero).

## Visualization

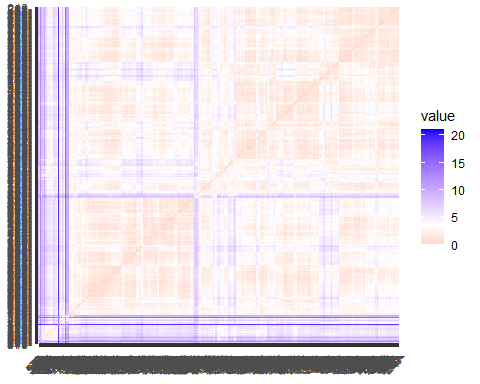
### Euclidean

library(factoextra)

## Warning: package 'factoextra' was built under R version 4.0.5

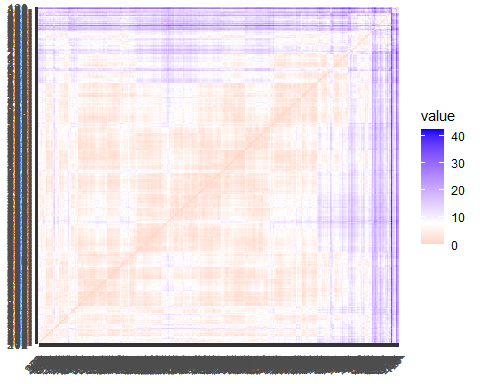
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

fviz\_dist(dist.eucl)



### Manhattan

fviz\_dist(dist.man)



These are the distance matrices, based on the Euclidean distance. The level of its colors is proportional to the value of the dissimilarity between observations: red stands for high similarity, blue indicates low similarity. Hence, in the diagonal the maximum of similarity is reached- namely the minimum of dissimilarity, in fact there are the pairs made up of each unit with itself, pairs with dissimilarity equal to zero.

## Maximum Number of Clusters

Using both Euclidean and Manhattan distance, according to different clustering methods, the maximum number of clusters will be computed as follows:

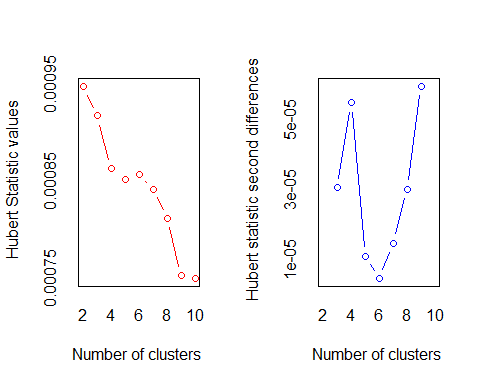
## Hierarchical Method

### Average Linkage Method & Euclidean Distance

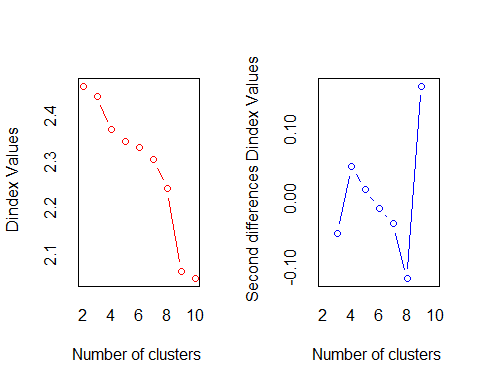
Average linkage method and Euclidean distance means the linkage methods work by calculating the distances or similarities between all objects. Then the closest pair of clusters are combined into a single cluster, reducing the number of clusters remaining. The process is then repeated until there is only a single cluster left.

library(NbClust)  
nb <- NbClust(liver\_scale, distance = "euclidean", min.nc = 2, max.nc = 10,  
method = "average")

## [1] "Frey index : No clustering structure in this data set"



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 9 proposed 2 as the best number of clusters   
## \* 5 proposed 3 as the best number of clusters   
## \* 8 proposed 9 as the best number of clusters   
## \* 1 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

library(factoextra)  
fviz\_nbclust(nb) +  
labs(subtitle = "H.C. - Average linkage Method & Euclidean Distance",  
 cex.sub= 0.5)

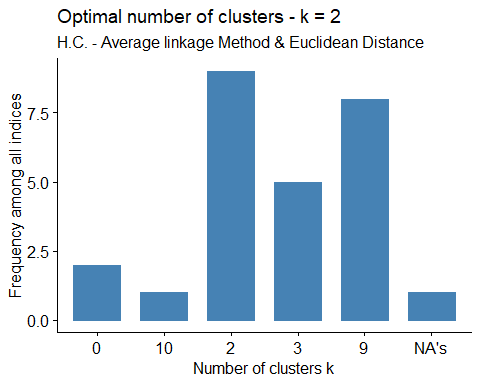
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") .viz\_NbClust(x, print.summary, : the  
## condition has length > 1 and only the first element will be used

## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") {: the condition has length > 1 and  
## only the first element will be used

## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 9 proposed 2 as the best number of clusters  
## \* 5 proposed 3 as the best number of clusters  
## \* 8 proposed 9 as the best number of clusters  
## \* 1 proposed 10 as the best number of clusters  
## \* 1 proposed NA's as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 2 .



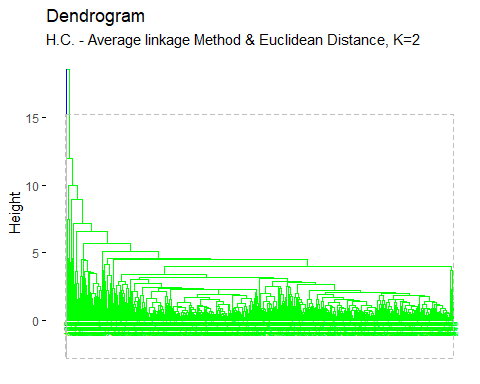
hc <- hclust(dist.eucl, method = "average")  
grp <- cutree(hc, k=2)  
table(grp)

## grp  
## 1 2   
## 578 1

head(grp)

## 1 2 3 4 5 6   
## 1 1 1 1 1 1

fviz\_dend(hc, k = 2, cex = 0.5, k\_colors = c("blue", "green"),  
 color\_labels\_by\_k = TRUE, rect = TRUE) +  
labs(title = "Dendrogram", subtitle = "H.C. - Average linkage Method & Euclidean Distance, K=2", cex.subtitle= 0.5)

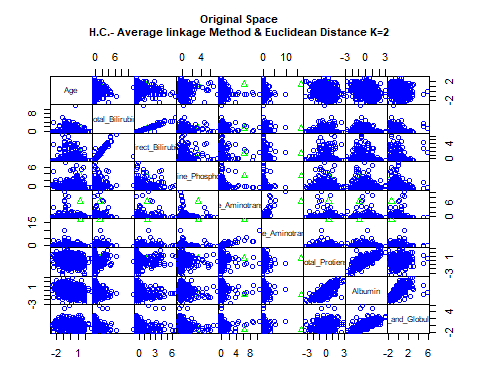


cor(dist.eucl, cophenetic(hc))

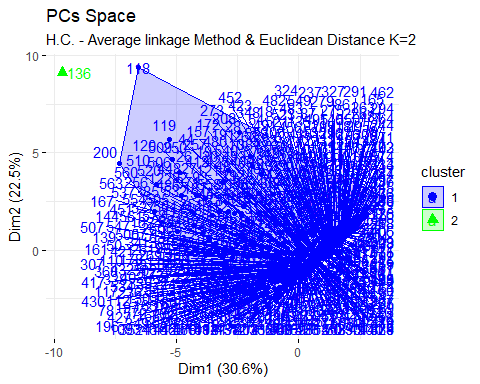
## [1] 0.8818356

According to the result of “NbClust”, the best number of clusters, applying hierarchical clustering using average linkage method and euclidean distance is 2.

pairs(liver\_scale, gap=0, pch=grp, cex.main= 0.7,  
 main="Original Space\nH.C.- Average linkage Method & Euclidean Distance K=2",  
 col=c("blue", "green")[grp])



options(ggrepel.max.overlaps = Inf)  
  
fviz\_cluster(list(data = liver\_scale, cluster = grp), palette = c("blue", "green"),  
 ellipse.type = "convex", main="PCs Space", repel = TRUE,  
 show.clust.aver = FALSE, ggtheme = theme\_minimal()) +  
 labs(subtitle = "H.C. - Average linkage Method & Euclidean Distance K=2",  
 cex.sub= 0.5)



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

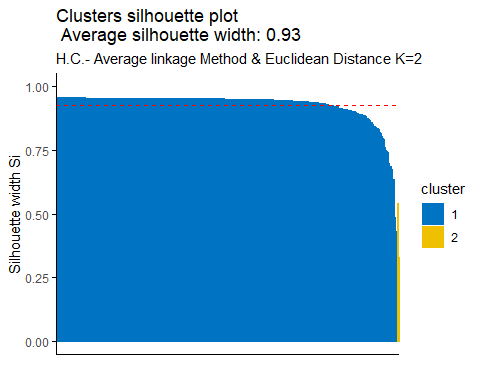
##### Silhouette Width

hclust<- eclust(liver\_sub, k=2, "hclust", hc\_method = "average", nboot = 50,  
 hc\_metric = "euclidean")  
silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.9252641

fviz\_silhouette(hclust, palette = "jco", ggtheme = theme\_classic()) +  
 labs(subtitle = "H.C.- Average linkage Method & Euclidean Distance K=2",  
 cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 577 0.93  
## 2 2 2 0.44



silinfo$clus.avg.widths

## [1] 0.9269589 0.4363246

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu <- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## [1] cluster neighbor sil\_width  
## <0 rows> (or 0-length row.names)

The value of average silhouette width indicates that in average the units are well enough clustered. As in particular, in cluster one the units are on average the same silhouette value with respect to the silhouette width but second cluster which has the lower silhouette value with respect to the silhouette width.

##### Dunn Index

library(fpc)

## Warning: package 'fpc' was built under R version 4.0.5

stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.3913763

Units are not clustered sufficiently according to the Dunn index.

#### External Validation Measures

##### Confusion Index

According to the Confusion matrix, the number of clusters is more than nominal values. The clusters found are 2 while the nominal variable can take 2 possible values.

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2  
## 0 412 2  
## 1 165 0

A large number of patients who doesn’t have the liver disease (n = 412) has been classified in cluster 1 while cluster 2 have only 2 values. The same happened for whom have the liver disease (n = 165) classified in cluster 1 while cluster 2 has 0 values.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats<- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.004118416

According to the Correct Rand Index, there is no agreement between the numerical value and the cluster solution. From -1 to +1, the agreement is very close to 0.

##### Meila’s VI Index

stats$vi

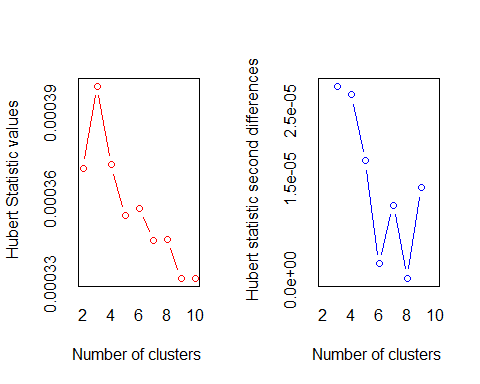
## [1] 0.6182953

### Average Linkage Method & Manhattan Distance

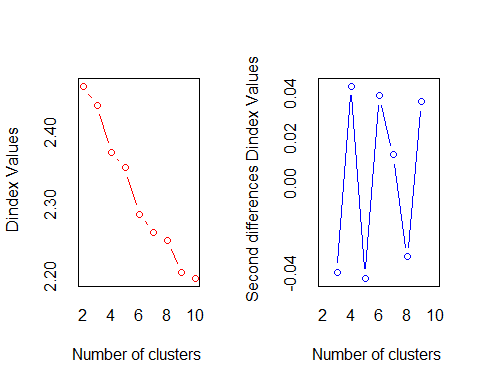
nb <- NbClust(liver\_scale, distance = "manhattan", min.nc = 2, max.nc = 10,  
method = "average")

## Warning in pf(beale, pp, df2): NaNs produced

## [1] "Frey index : No clustering structure in this data set"



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 9 proposed 2 as the best number of clusters   
## \* 3 proposed 3 as the best number of clusters   
## \* 3 proposed 4 as the best number of clusters   
## \* 2 proposed 5 as the best number of clusters   
## \* 4 proposed 6 as the best number of clusters   
## \* 1 proposed 8 as the best number of clusters   
## \* 1 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fviz\_nbclust(nb) +  
 labs(subtitle = "H.C. - Average linkage Method & Manhattab Distance",  
 cex.sub= 0.5)

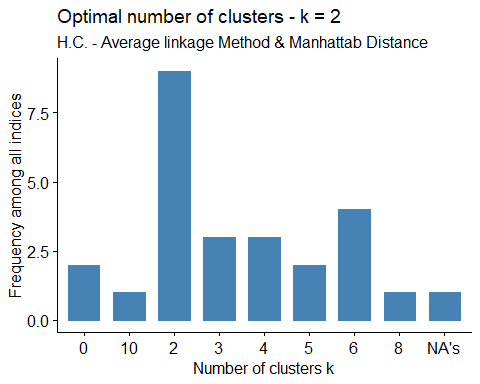
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") .viz\_NbClust(x, print.summary, : the  
## condition has length > 1 and only the first element will be used

## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") {: the condition has length > 1 and  
## only the first element will be used

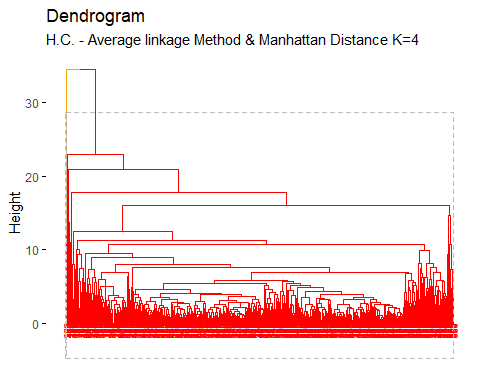
## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 9 proposed 2 as the best number of clusters  
## \* 3 proposed 3 as the best number of clusters  
## \* 3 proposed 4 as the best number of clusters  
## \* 2 proposed 5 as the best number of clusters  
## \* 4 proposed 6 as the best number of clusters  
## \* 1 proposed 8 as the best number of clusters  
## \* 1 proposed 10 as the best number of clusters  
## \* 1 proposed NA's as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 2 .



dist.man <- dist(liver\_scale, method = "manhattan")  
hc <- hclust(dist.man, method = "average")  
grp <- cutree(hc, k=2)  
table(grp)

## grp  
## 1 2   
## 578 1

fviz\_dend(hc, k = 2, cex = 0.5, k\_colors = c("orange", "red"),  
 color\_labels\_by\_k = TRUE, rect = TRUE) +  
labs(title = "Dendrogram",  
 subtitle = "H.C. - Average linkage Method & Manhattan Distance K=4",  
 cex.subtitle= 0.5)

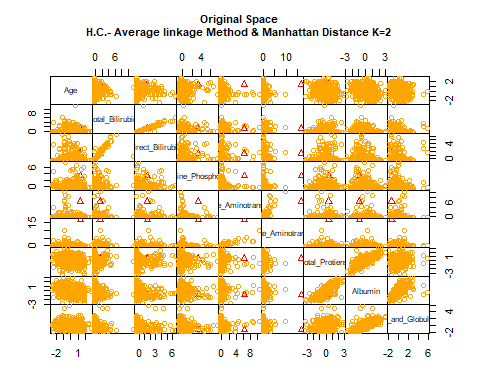


cor(dist.man, cophenetic(hc))

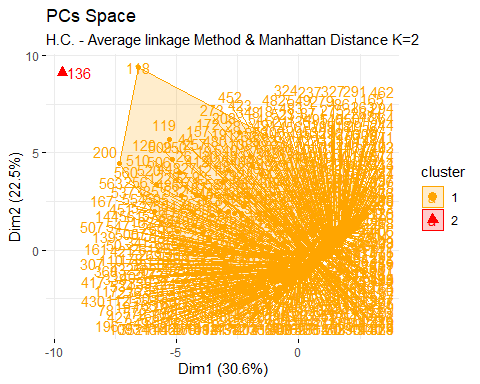
## [1] 0.8639619

According to the result of “NbClust”, the best number of clusters, applying hierarchical clustering using average linkage method and manhattan distance is 2.

pairs(liver\_scale, gap=0, pch=grp, cex.main= 0.7,  
 main="Original Space\nH.C.- Average linkage Method & Manhattan Distance K=2",  
 col=c("orange", "red")[grp])



options(ggrepel.max.overlaps = Inf)  
  
fviz\_cluster(list(data = liver\_scale, cluster = grp),  
palette = c("orange", "red"), ellipse.type = "convex",  
main="PCs Space", repel = TRUE, show.clust.aver = FALSE,  
ggtheme = theme\_minimal()) +   
labs(subtitle = "H.C. - Average linkage Method & Manhattan Distance K=2",  
cex.sub= 0.5)



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

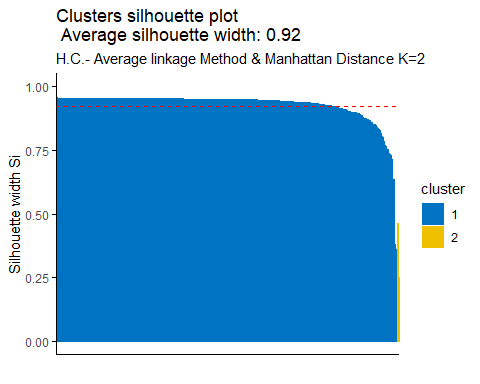
##### Silhouette width

hclust<- eclust(liver\_sub, k=2, "hclust", hc\_method = "average", nboot = 50,  
 hc\_metric = "manhattan")  
silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.9228996

fviz\_silhouette(hclust, palette = "jco", ggtheme = theme\_classic()) +  
labs(subtitle = "H.C.- Average linkage Method & Manhattan Distance K=2",  
cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 577 0.92  
## 2 2 2 0.36



silinfo$clus.avg.widths

## [1] 0.9248572 0.3581227

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu <- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## [1] cluster neighbor sil\_width  
## <0 rows> (or 0-length row.names)

The value of average silhouette width indicates that in average the units are well enough clustered. In particular, in cluster 1 (blue cluster) the units having the same silhouette value with respect to the silhouette width, in cluster 2 (the yellow one) the units are on average having the lower silhouette value with respect to silhouette width.

##### Dunn Index

library(fpc)  
stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.3913763

According to the Dunn index, the units are not clustered well enough.

#### External Validation Measures

##### Confusion Matrix

According to the Confusion matrix, the number of clusters is equal to nominal values.

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2  
## 0 412 2  
## 1 165 0

For the liver disease, data has been classified mostly in cluster 1, 577 units in cluster 1, and 2 in cluster 2. For the patients having liver disease classified mostly in cluster 1 while cluster 2 has 0 values. Safe to say data are not well balanced in both cluster.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats<- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.004118416

According to the Correct Rand Index, there is no agreement between the numerical values and the cluster solution. From -1 to +1, the agreement is very close to 0.

##### Meila’s VI Index

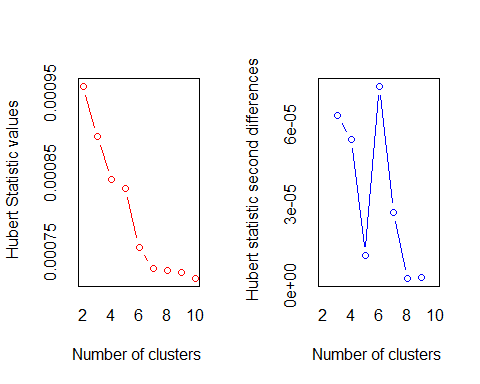
stats$vi

## [1] 0.6182953

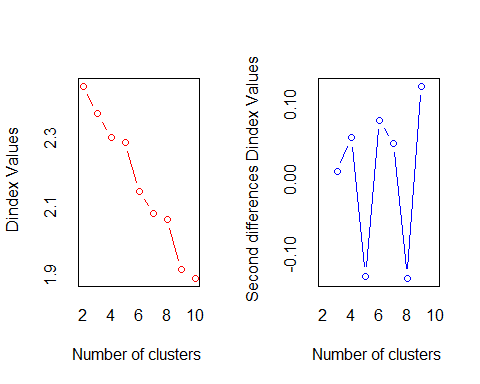
### Complete Linkage Method & Euclidean Distance

nb <- NbClust(liver\_scale, distance = "euclidean", min.nc = 2, max.nc = 10,  
 method = "complete")

## Warning in pf(beale, pp, df2): NaNs produced



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 8 proposed 2 as the best number of clusters   
## \* 2 proposed 3 as the best number of clusters   
## \* 3 proposed 5 as the best number of clusters   
## \* 2 proposed 6 as the best number of clusters   
## \* 2 proposed 8 as the best number of clusters   
## \* 5 proposed 9 as the best number of clusters   
## \* 2 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fviz\_nbclust(nb) +  
 labs(subtitle = "H.C. - Complete linkage Method & Euclidean Distance",  
 cex.sub= 0.5)

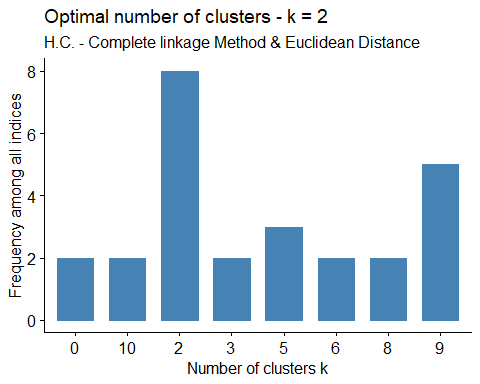
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") .viz\_NbClust(x, print.summary, : the  
## condition has length > 1 and only the first element will be used

## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") {: the condition has length > 1 and  
## only the first element will be used

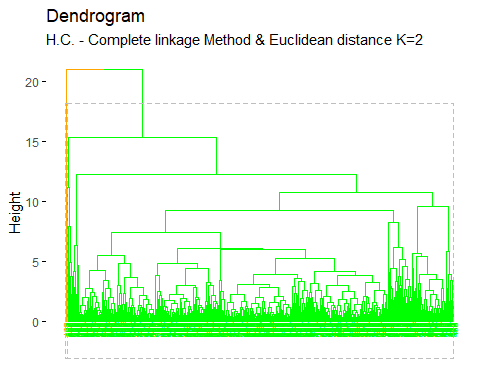
## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 8 proposed 2 as the best number of clusters  
## \* 2 proposed 3 as the best number of clusters  
## \* 3 proposed 5 as the best number of clusters  
## \* 2 proposed 6 as the best number of clusters  
## \* 2 proposed 8 as the best number of clusters  
## \* 5 proposed 9 as the best number of clusters  
## \* 2 proposed 10 as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 2 .



hc <- hclust(dist.eucl, method = "complete")  
grp <- cutree(hc, k=2)  
table(grp)

## grp  
## 1 2   
## 577 2

fviz\_dend(hc, k = 2, cex = 0.5, k\_colors = c("orange", "green"),  
 color\_labels\_by\_k = TRUE, rect = TRUE) + labs(title = "Dendrogram",  
 subtitle = "H.C. - Complete linkage Method & Euclidean distance K=2",  
 cex.subtitle= 0.5)

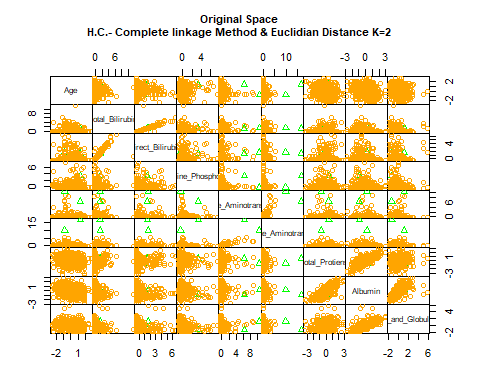


cor(dist.eucl, cophenetic(hc))

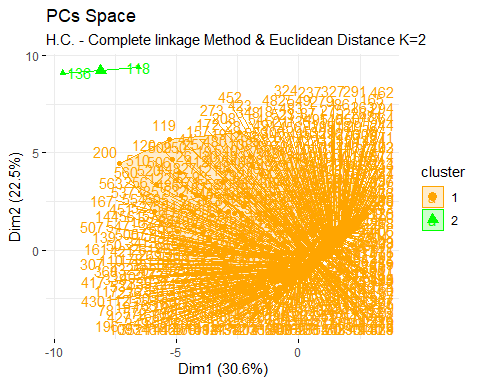
## [1] 0.76228

According to the result of “NbClust”, the best number of clusters, applying hierarchical clustering using complete linkage method and euclidean distance is 2.

pairs(liver\_scale, gap=0, pch=grp, cex.main= 0.7,  
main="Original Space\nH.C.- Complete linkage Method & Euclidian Distance K=2",  
col=c("orange", "green")[grp])



fviz\_cluster(list(data = liver\_scale, cluster = grp),  
 palette = c("orange", "green"), ellipse.type = "convex",  
 main="PCs Space", repel = TRUE, show.clust.aver = FALSE,  
 ggtheme = theme\_minimal()) +  
 labs(subtitle = "H.C. - Complete linkage Method & Euclidean Distance K=2",  
 cex.sub= 0.5)



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal validation measures

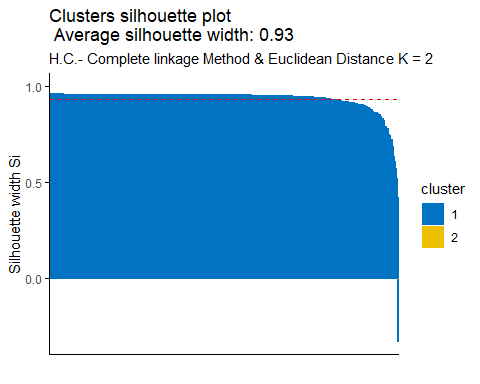
##### Silhouette width

hclust<- eclust(liver\_sub, k = 2, "hclust", hc\_method = "complete",  
 nboot = 50, hc\_metric = "euclidean")  
silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.9341632

fviz\_silhouette(hclust, palette = "jco", ggtheme = theme\_classic()) +  
labs(subtitle = "H.C.- Complete linkage Method & Euclidean Distance K = 2",  
cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 578 0.94  
## 2 2 1 0.00



silinfo$clus.avg.widths

## [1] 0.9357794 0.0000000

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu<- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## cluster neighbor sil\_width  
## 118 1 2 -0.3302797

The value of complete silhouette width indicates that on average the units are well enough clustered. In cluster 1 (blue cluster) the units are on average the same silhouette value with respect to the silhouette width, in cluster 2 (the yellow one) the units are on average having the lower silhouette value with respect to silhouette width. According to the index, unit that belong to cluster 1 are not well clustered, it should belong to the neighbor cluster 2.

##### Dunn Index

stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.5317513

According to the Dunn index, the units are not clustered well enough.

#### External Validation Measures

##### Confusion Matrix

According to the Confusion matrix, the number of clusters is equal to the nominal values.

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2  
## 0 413 1  
## 1 165 0

For the liver disease, data has been classified mostly in cluster 1, 578 units in cluster 1, and 1 in cluster 2. For the patients having liver disease classified mostly in cluster 1 while cluster 2 has 0 values. Safe to say data are not well balanced in both cluster.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats<- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.002074324

According to the Correct Rand Index, there is no agreement between the numerical values and the cluster solution. From -1 to +1, the agreement is very close to 0.

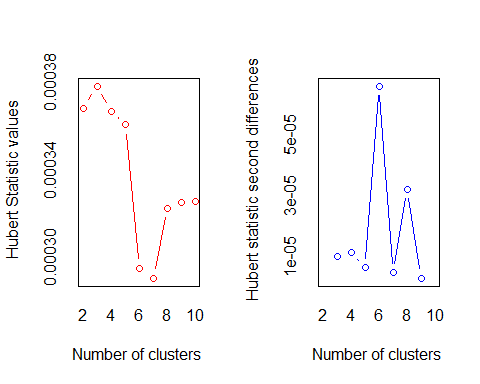
##### Meila’s VI Index

stats$vi

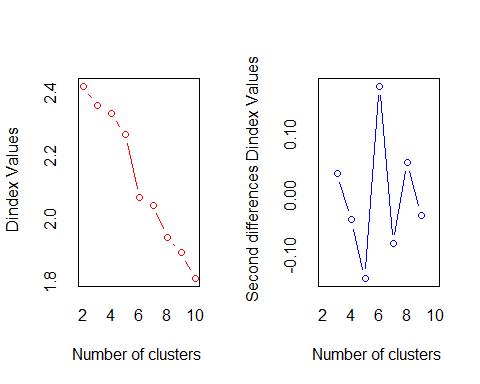
## [1] 0.6091425

### Complete linkage Method & Manhattan Distance

nb <- NbClust(liver\_scale, distance = "manhattan", min.nc = 2, max.nc = 10,  
 method = "complete")



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 9 proposed 2 as the best number of clusters   
## \* 1 proposed 3 as the best number of clusters   
## \* 5 proposed 4 as the best number of clusters   
## \* 1 proposed 5 as the best number of clusters   
## \* 6 proposed 6 as the best number of clusters   
## \* 2 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fviz\_nbclust(nb) +  
 labs(subtitle = "H.C. - Complete linkage Method & Manhattan Distance",  
 cex.sub= 0.5)

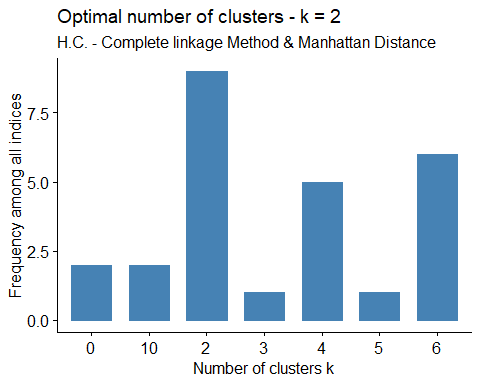
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") .viz\_NbClust(x, print.summary, : the  
## condition has length > 1 and only the first element will be used

## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") {: the condition has length > 1 and  
## only the first element will be used

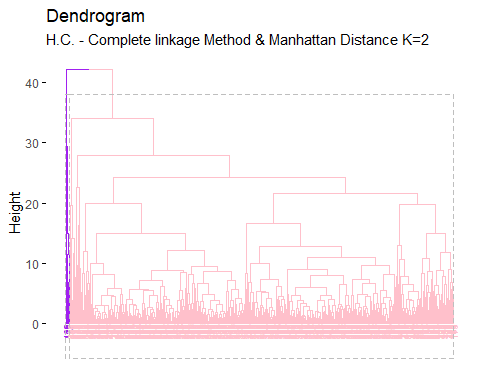
## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 9 proposed 2 as the best number of clusters  
## \* 1 proposed 3 as the best number of clusters  
## \* 5 proposed 4 as the best number of clusters  
## \* 1 proposed 5 as the best number of clusters  
## \* 6 proposed 6 as the best number of clusters  
## \* 2 proposed 10 as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 2 .



dist.man <- dist(liver\_scale, method = "manhattan")  
hc <- hclust(dist.man, method = "complete")  
grp <- cutree(hc, k=2)  
table(grp)

## grp  
## 1 2   
## 573 6

fviz\_dend(hc, k = 2, cex = 0.5, k\_colors = c("purple", "pink"),  
 r\_labels\_by\_k = TRUE, rect = TRUE) + labs(title = "Dendrogram",  
 subtitle = "H.C. - Complete linkage Method & Manhattan Distance K=2",  
 cex.subtitle= 0.5)

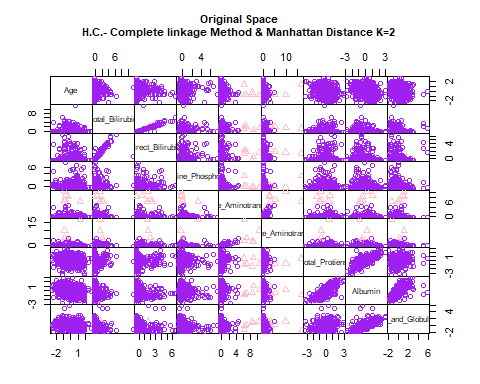


cor(dist.man, cophenetic(hc))

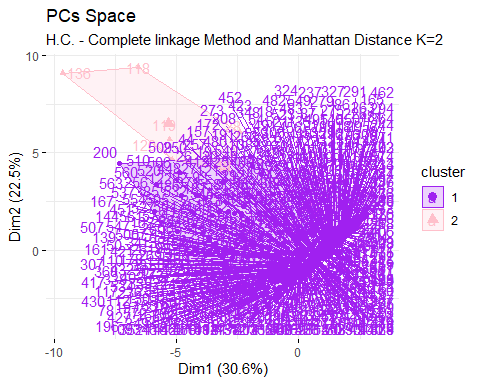
## [1] 0.6077513

According to the result of “NbClust”, the best number of clusters, applying hierarchical clustering using complete linkage method and manhattan distance is 2.

pairs(liver\_scale, gap=0, pch=grp, cex.main= 0.7,  
 main="Original Space\nH.C.- Complete linkage Method & Manhattan Distance K=2",  
 col=c("purple", "pink")[grp])



fviz\_cluster(list(data = liver\_scale, cluster = grp), palette = c("purple", "pink"),  
 ellipse.type = "convex", main="PCs Space", repel = TRUE,  
 show.clust.aver = FALSE, ggtheme = theme\_minimal()) +  
 labs(subtitle = "H.C. - Complete linkage Method and Manhattan Distance K=2",  
 cex.sub= 0.5)



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

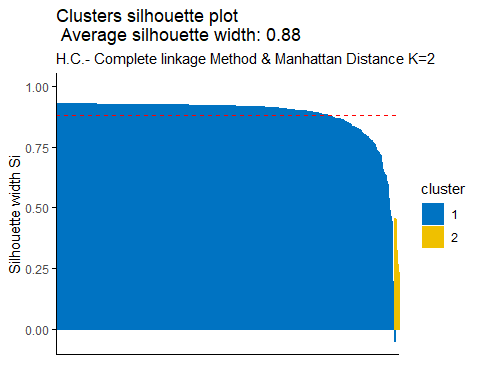
##### Silhouette width

hclust<- eclust(liver\_sub, k=2, "hclust", hc\_method = "complete",  
 nboot = 50, hc\_metric = "manhattan")  
silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.8807343

fviz\_silhouette(hclust, palette = "jco", ggtheme = theme\_classic()) +  
labs(subtitle = "H.C.- Complete linkage Method & Manhattan Distance K=2",  
cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 572 0.89  
## 2 2 7 0.33



silinfo$clus.avg.widths

## [1] 0.8875222 0.3260695

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu<- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## cluster neighbor sil\_width  
## 200 1 2 -0.05135232

The value of complete silhouette width indicates that on average the units are well enough clustered. In cluster 1 (blue cluster) the units are on average the same silhouette value with respect to the silhouette width, in cluster 2 (the yellow one) the units are on average having the lower silhouette value with respect to silhouette width. According to the index, unit that belong to cluster 1 are not well clustered, it should belong to the neighbor cluster 2.

##### Dunn Index

stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.2188165

According to the Dunn index, the units are not clustered well enough.

#### External Validation Measures

##### Confusion Matrix

According to the Confusion matrix, the number of clusters is equal to nominal values. The clusters found are 2 and the nominal variable can take 2 possible values.

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2  
## 0 407 7  
## 1 165 0

For the liver disease, data has been classified mostly in cluster 1, 572 units in cluster 1, and 7 in cluster 2. For the patients having liver disease classified mostly in cluster 1 while cluster 2 has 0 values. Safe to say data are not well balanced in both cluster.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats <- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.01389137

According to the Correct Rand Index, there is no agreement between the numerical value and the cluster. solution. From -1 to +1, the agreement is very close to 0.

##### Meila’s VI Index

stats$vi

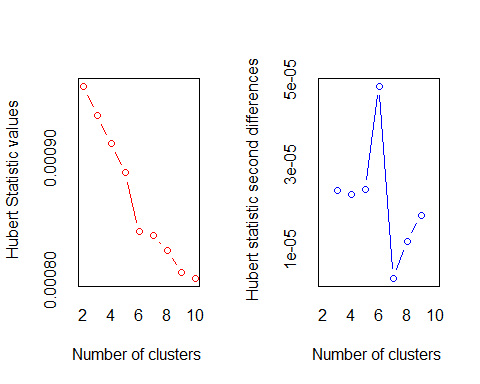
## [1] 0.6548182

### Centroid Linkage Method & Euclidean Distance

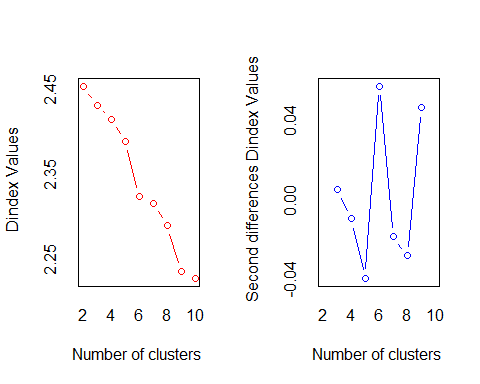
library(NbClust)  
nb <- NbClust(liver\_scale, distance = "euclidean", min.nc = 2, max.nc = 10,  
 method = "centroid")

## Warning in pf(beale, pp, df2): NaNs produced  
  
## Warning in pf(beale, pp, df2): NaNs produced

## [1] "Frey index : No clustering structure in this data set"



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 8 proposed 2 as the best number of clusters   
## \* 5 proposed 3 as the best number of clusters   
## \* 3 proposed 4 as the best number of clusters   
## \* 3 proposed 6 as the best number of clusters   
## \* 1 proposed 7 as the best number of clusters   
## \* 3 proposed 9 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

library(factoextra)  
fviz\_nbclust(nb) +  
 labs(subtitle = "H.C. - Centroid linkage Method & Euclidian Distance",  
 cex.sub= 0.5)

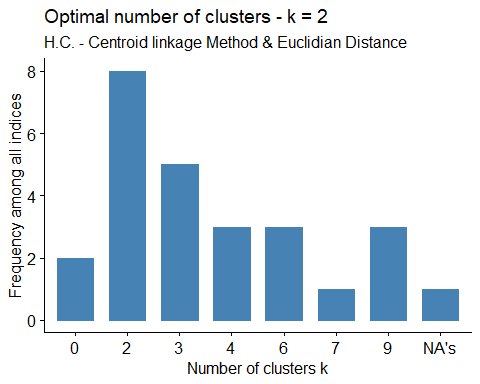
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") .viz\_NbClust(x, print.summary, : the  
## condition has length > 1 and only the first element will be used

## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") {: the condition has length > 1 and  
## only the first element will be used

## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 8 proposed 2 as the best number of clusters  
## \* 5 proposed 3 as the best number of clusters  
## \* 3 proposed 4 as the best number of clusters  
## \* 3 proposed 6 as the best number of clusters  
## \* 1 proposed 7 as the best number of clusters  
## \* 3 proposed 9 as the best number of clusters  
## \* 1 proposed NA's as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 2 .

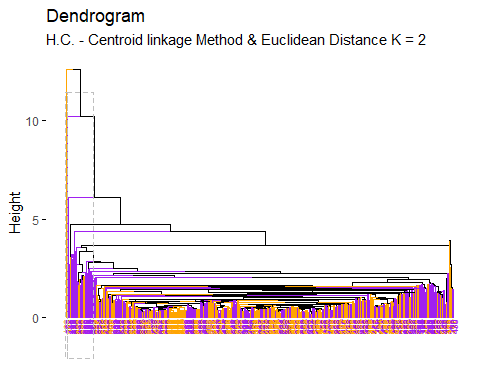


dist.eucl <- dist(liver\_scale, method = "euclidian")  
hc <- hclust(dist.eucl, method = "centroid")  
grp <- cutree(hc, k=2)  
table(grp)

## grp  
## 1 2   
## 577 2

fviz\_dend(hc, k = 2, cex = 0.5, k\_colors = c("orange", "purple"),  
 color\_labels\_by\_k = TRUE, rect = TRUE) + labs(title = "Dendrogram",  
 subtitle = "H.C. - Centroid linkage Method & Euclidean Distance K = 2",  
 cex.subtitle= 0.5)

## Warning in get\_col(col, k): Length of color vector was shorter than the number  
## of clusters - color vector was recycled

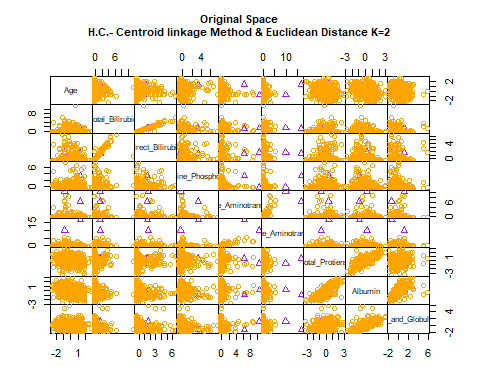


cor(dist.eucl, cophenetic(hc))

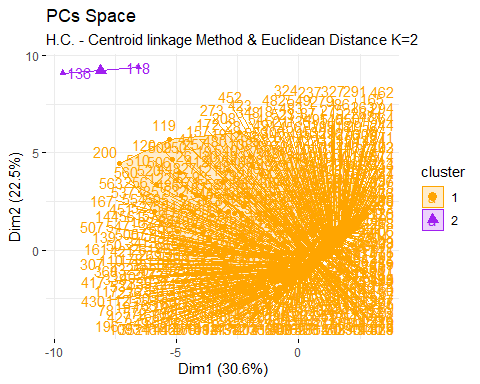
## [1] 0.8727531

According to the result of “NbClust”, the best number of clusters, applying hierarchical clustering using the Centroid linkage method and euclidean distance is 2.

pairs(liver\_scale, gap=0, pch=grp, cex.main= 0.7,  
 main="Original Space\nH.C.- Centroid linkage Method & Euclidean Distance K=2",  
 col=c("orange", "purple")[grp])



fviz\_cluster(list(data = liver\_scale, cluster = grp),  
 palette = c("orange", "purple"), ellipse.type = "convex",  
 main="PCs Space", repel = TRUE, show.clust.aver = FALSE,  
 ggtheme = theme\_minimal()) +  
 labs(subtitle = "H.C. - Centroid linkage Method & Euclidean Distance K=2",  
 cex.sub= 0.5)



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

##### Silhouette width

hclust<- eclust(liver\_sub, k = 2, "hclust", hc\_method = "centroid",  
 nboot = 50, hc\_metric = "euclidean")

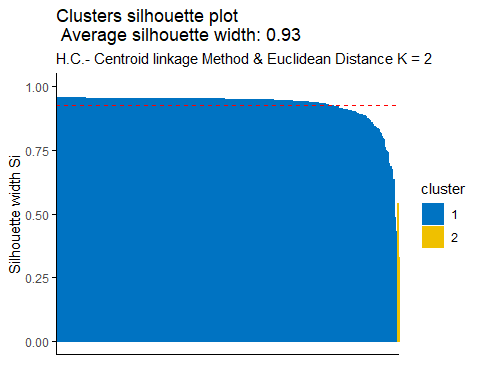
## Warning in get\_col(col, k): Length of color vector was shorter than the number  
## of clusters - color vector was recycled

silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.9252641

fviz\_silhouette(hclust, palette = "jco", ggtheme = theme\_classic()) +  
 labs(subtitle = "H.C.- Centroid linkage Method & Euclidean Distance K = 2",  
 cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 577 0.93  
## 2 2 2 0.44



silinfo$clus.avg.widths

## [1] 0.9269589 0.4363246

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu<- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## [1] cluster neighbor sil\_width  
## <0 rows> (or 0-length row.names)

The value of complete silhouette width indicates that on average the units are well enough clustered. In cluster 1 (blue cluster) the units are on average the same silhouette value with respect to the silhouette width, in cluster 2 (the yellow one) the units are on average having the lower silhouette value with respect to silhouette width.

##### Dunn Index

library(fpc)  
stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.3913763

According to the Dunn index, the units are not clustered well enough.

#### External Validation Measures

##### Confusion Matrix

According to the Confusion matrix, the number of clusters is equal to the nominal values.

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2  
## 0 412 2  
## 1 165 0

For the liver disease, data has been classified mostly in cluster 1, 577 units in cluster 1, and 2 in cluster 2. For the patients having liver disease classified mostly in cluster 1 while cluster 2 has 0 values. Safe to say data are not well balanced in both cluster.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats<- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.004118416

According to the Correct Rand Index, there is no agreement between the numerical values and the cluster. Solution. From -1 to +1, the agreement is very close to 0.

##### Meila’s VI Index

stats$vi

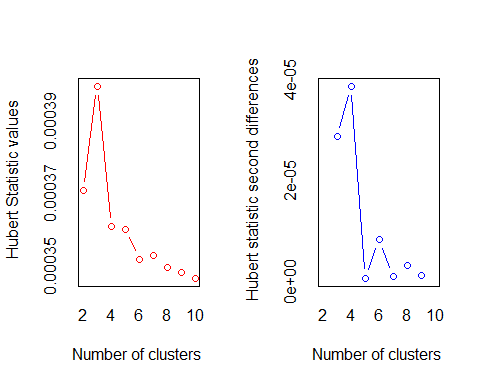
## [1] 0.6182953

### Centroid linkage Method & Manhattan Distance

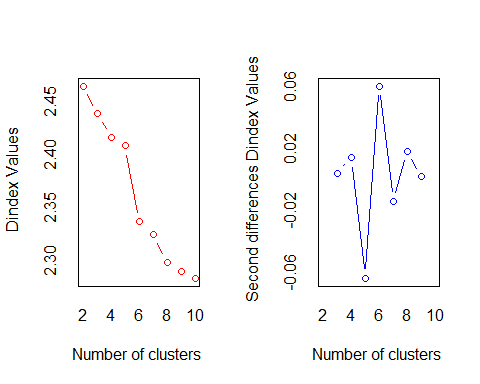
nb <- NbClust(liver\_scale, distance = "manhattan", min.nc = 2, max.nc = 10,  
 method = "centroid")

## Warning in pf(beale, pp, df2): NaNs produced

## [1] "Frey index : No clustering structure in this data set"



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 10 proposed 2 as the best number of clusters   
## \* 2 proposed 3 as the best number of clusters   
## \* 3 proposed 4 as the best number of clusters   
## \* 2 proposed 5 as the best number of clusters   
## \* 4 proposed 6 as the best number of clusters   
## \* 1 proposed 7 as the best number of clusters   
## \* 1 proposed 8 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fviz\_nbclust(nb) +  
 labs(subtitle = "H.C. - Centroid linkage Method & Manhattan Distance",  
 cex.sub= 0.5)

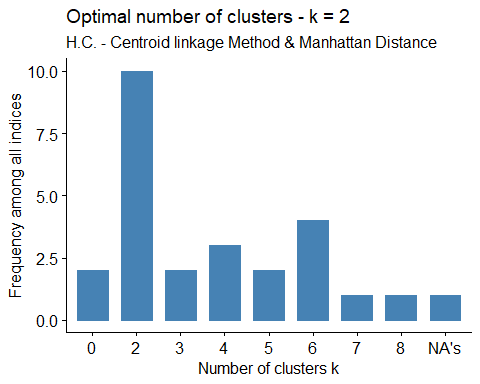
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") .viz\_NbClust(x, print.summary, : the  
## condition has length > 1 and only the first element will be used

## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") {: the condition has length > 1 and  
## only the first element will be used

## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 10 proposed 2 as the best number of clusters  
## \* 2 proposed 3 as the best number of clusters  
## \* 3 proposed 4 as the best number of clusters  
## \* 2 proposed 5 as the best number of clusters  
## \* 4 proposed 6 as the best number of clusters  
## \* 1 proposed 7 as the best number of clusters  
## \* 1 proposed 8 as the best number of clusters  
## \* 1 proposed NA's as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 2 .



dist.man <- dist(liver\_scale, method = "manhattan")  
hc <- hclust(dist.man, method = "centroid")  
grp <- cutree(hc, k = 2)  
table(grp)

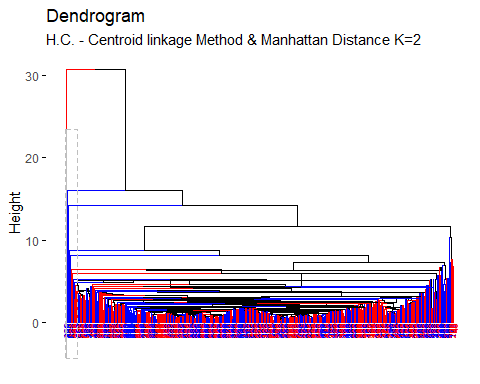
## grp  
## 1 2   
## 578 1

head(grp)

## 1 2 3 4 5 6   
## 1 1 1 1 1 1

fviz\_dend(hc, k = 2, cex = 0.5, k\_colors = c("red", "blue"),  
 color\_labels\_by\_k = TRUE, rect = TRUE) + labs(title = "Dendrogram",  
 subtitle = "H.C. - Centroid linkage Method & Manhattan Distance K=2",  
 cex.subtitle= 0.5)

## Warning in get\_col(col, k): Length of color vector was shorter than the number  
## of clusters - color vector was recycled

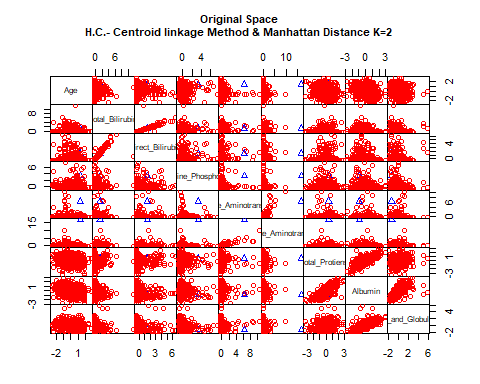


cor(dist.eucl, cophenetic(hc))

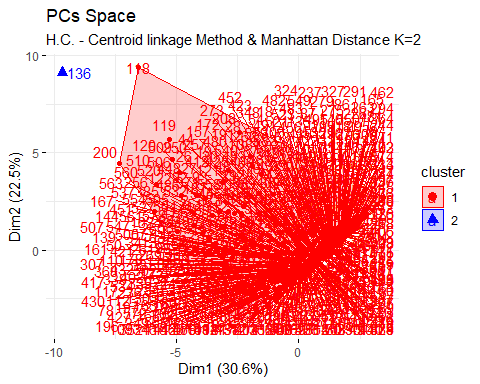
## [1] 0.8532441

According to the result of NbClust, the best number of clusters, applying hierarchical clustering using centroid linkage method and manhattan distance is 2.

pairs(liver\_scale, gap=0, pch = grp, cex.main = 0.7,  
 main = "Original Space\nH.C.- Centroid linkage Method & Manhattan Distance K=2",  
 col=c("red", "blue")[grp])



fviz\_cluster(list(data = liver\_scale, cluster = grp),  
 palette = c("red", "blue"), ellipse.type = "convex",  
 main = "PCs Space", repel = TRUE, show.clust.aver = FALSE,  
 ggtheme = theme\_minimal()) +  
 labs(subtitle = "H.C. - Centroid linkage Method & Manhattan Distance K=2",  
 cex.sub= 0.5)



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

##### Silhouette width

hclust<- eclust(liver\_sub, k = 2, "hclust", hc\_method = "centroid",  
 nboot = 50, hc\_metric = "manhattan")

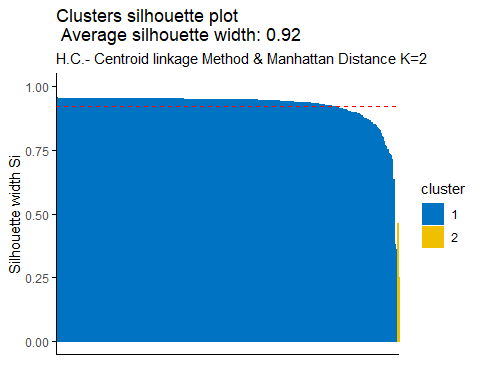
## Warning in get\_col(col, k): Length of color vector was shorter than the number  
## of clusters - color vector was recycled

silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.9228996

fviz\_silhouette(hclust, palette = "jco", ggtheme = theme\_classic()) +  
 labs(subtitle = "H.C.- Centroid linkage Method & Manhattan Distance K=2",  
cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 577 0.92  
## 2 2 2 0.36



silinfo$clus.avg.widths

## [1] 0.9248572 0.3581227

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu <- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## [1] cluster neighbor sil\_width  
## <0 rows> (or 0-length row.names)

The value of complete silhouette width indicates that on average the units are well enough clustered. In cluster 1 (blue cluster) the units are on average the same silhouette value with respect to the silhouette width, in cluster 2 (the yellow one) the units are on average having the lower silhouette value with respect to silhouette width.

##### Dunn index

stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.3913763

#### External Validation Measures

##### Confusion Matrix

According to the Confusion matrix, the number of clusters is equal to nominal values. The clusters found are 2 and the nominal variable can take 2 possible values.

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2  
## 0 412 2  
## 1 165 0

For the liver disease, data has been classified mostly in cluster 1, 577 units in cluster 1, and 2 in cluster 2. For the patients having liver disease classified mostly in cluster 1 while cluster 2 has 0 values. Safe to say data are not well balanced in both cluster.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats <- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.004118416

According to the Correct Rand Index, there is no agreement between the numerical value and the cluster. solution. From -1 to +1, the agreement is very close to 0.

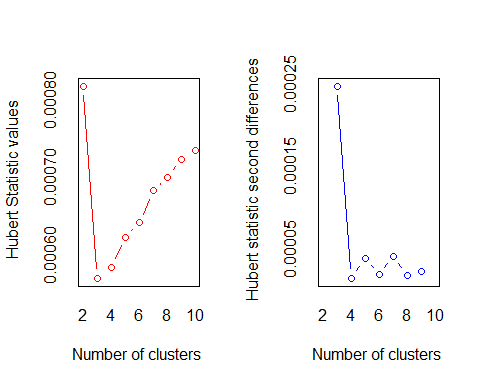
##### Meila’s VI Index

stats$vi

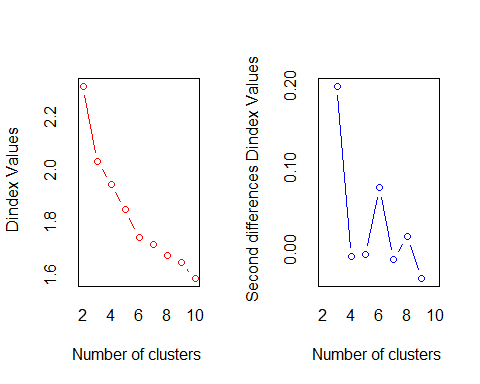
## [1] 0.6182953

### Ward’s Method - Minimum Deviance

nb <- NbClust(liver\_scale, distance = "euclidean", min.nc = 2, max.nc = 10,  
 method = "ward.D2")



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 5 proposed 2 as the best number of clusters   
## \* 6 proposed 3 as the best number of clusters   
## \* 6 proposed 4 as the best number of clusters   
## \* 2 proposed 6 as the best number of clusters   
## \* 1 proposed 7 as the best number of clusters   
## \* 4 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 3   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fviz\_nbclust(nb) +  
 labs(subtitle = "H.C. - Wards's Method", cex.sub= 0.5)

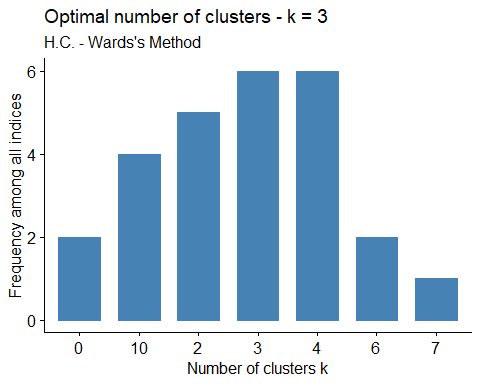
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") .viz\_NbClust(x, print.summary, : the  
## condition has length > 1 and only the first element will be used

## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") {: the condition has length > 1 and  
## only the first element will be used

## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 5 proposed 2 as the best number of clusters  
## \* 6 proposed 3 as the best number of clusters  
## \* 6 proposed 4 as the best number of clusters  
## \* 2 proposed 6 as the best number of clusters  
## \* 1 proposed 7 as the best number of clusters  
## \* 4 proposed 10 as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 3 .



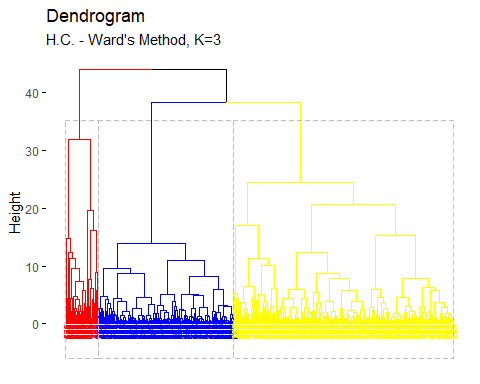
hc <- hclust(dist.eucl, method = "ward.D2")  
grp <- cutree(hc, k=3)  
table(grp)

## grp  
## 1 2 3   
## 328 202 49

head(grp)

## 1 2 3 4 5 6   
## 1 1 1 1 1 2

fviz\_dend(hc, k = 3, cex = 0.5, k\_colors = c("red", "blue","yellow"),  
 color\_labels\_by\_k = TRUE, rect = TRUE) +  
 labs(title = "Dendrogram", subtitle = "H.C. - Ward's Method, K=3",  
 cex.subtitle= 0.5)

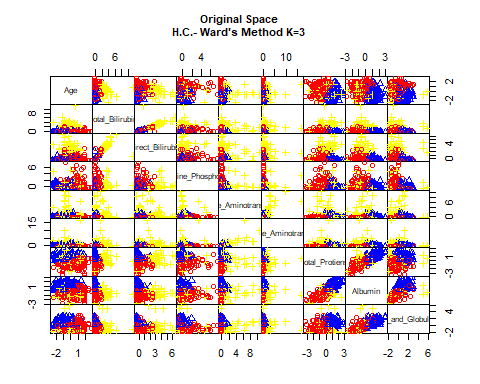


cor(dist.eucl, cophenetic(hc))

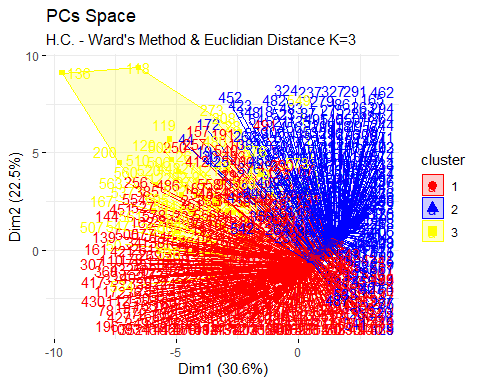
## [1] 0.5078094

According to the function NbClust, the best number of clusters, applying hierarchical clustering using Ward’s method and euclidean distance is 3.

pairs(liver\_scale, gap=0, pch=grp, cex.main= 0.7,  
 main="Original Space\nH.C.- Ward's Method K=3",  
 col=c("red", "blue", "yellow")[grp])



fviz\_cluster(list(data = liver\_scale, cluster = grp),  
 palette = c("red", "blue", "yellow"), ellipse.type = "convex",  
 main="PCs Space", repel = TRUE, show.clust.aver = FALSE,  
 ggtheme = theme\_minimal()) +  
 labs(subtitle = "H.C. - Ward's Method & Euclidian Distance K=3", cex.sub= 0.5)



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

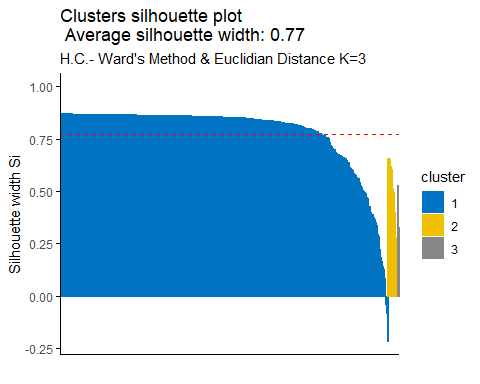
##### Silhouette Width

hclust<- eclust(liver\_sub, k=3, "hclust", hc\_method = "ward.D2", nboot = 50)  
silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.7738898

fviz\_silhouette(hclust, palette = "jco", ggtheme = theme\_classic()) +  
 labs(subtitle = "H.C.- Ward's Method & Euclidian Distance K=3", cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 560 0.78  
## 2 2 17 0.48  
## 3 3 2 0.43



silinfo$clus.avg.widths

## [1] 0.7840013 0.4812584 0.4300389

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu <- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## cluster neighbor sil\_width  
## 486 1 2 -0.04182439  
## 256 1 2 -0.08167575  
## 417 1 2 -0.08297610  
## 81 1 2 -0.21688832

The value of complete silhouette width indicates that on average the units are well enough clustered. As in particular, in cluster 1 the units are on above average the silhouette value with respect to the silhouette width, in cluster 2 the units are on average having the lower silhouette value with respect to the silhouette width, and in cluster 3 on average having the lower silhouette value with respect to the silhouette width. According to above index, units that belong to cluster 1 are not well clustered, they should belong to the neighbor cluster 2.

##### Dunn Index

stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.07148566

According to the Dunn index, the units are not clustered well enough.

#### External Validation Measures

##### Confusion Matrix

According to the Confusion matrix, the number of clusters is equal to nominal values. The clusters found are 2 and the nominal variable can take 2 possible values.

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2 3  
## 0 396 16 2  
## 1 164 1 0

For the liver disease, data has been classified mostly in cluster 1, 560 units in cluster 1, 17 in cluster 2, and 2 in cluster 3. For the patients having liver disease classified mostly in cluster 1 while cluster 2 has 1 value and cluster 3 has 0 value. Safe to say data are not well balanced in both cluster.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats<- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.02867484

According to the Correct Rand Index, there is no agreement between the numerical value and the cluster solution. From -1 to +1, the agreement is very close to 0.

##### Meila’s VI Index

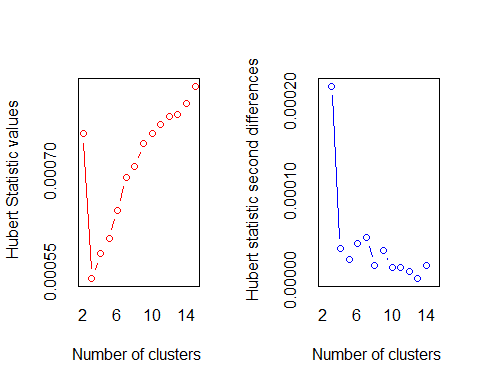
stats$vi

## [1] 0.7406794

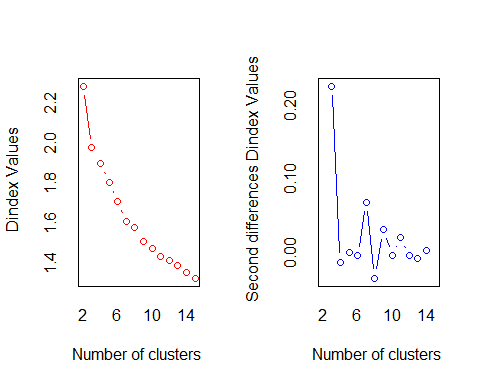
## Partitional Method

### K-Means

library(ggplot2)  
nb <- NbClust(liver\_scale, min.nc=2, max.nc=15, method="kmeans")



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 8 proposed 2 as the best number of clusters   
## \* 5 proposed 3 as the best number of clusters   
## \* 3 proposed 5 as the best number of clusters   
## \* 3 proposed 6 as the best number of clusters   
## \* 1 proposed 14 as the best number of clusters   
## \* 4 proposed 15 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fviz\_nbclust(nb) +  
 labs(subtitle = " Partitional Clustering - K-Means")

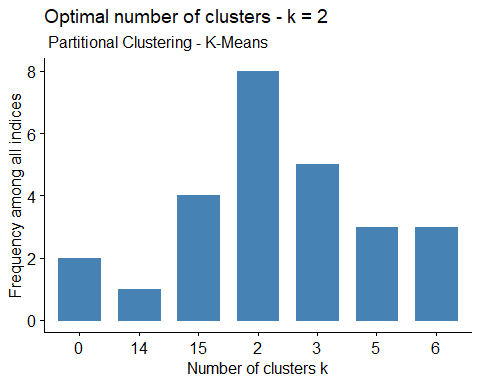
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") .viz\_NbClust(x, print.summary, : the  
## condition has length > 1 and only the first element will be used

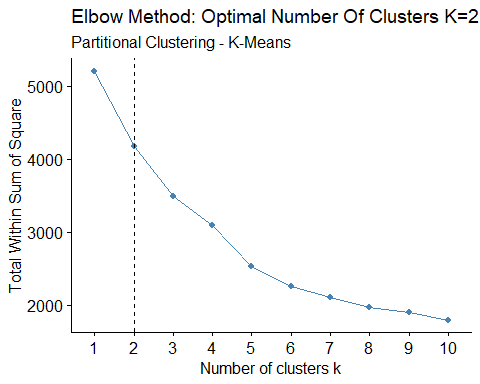
## Warning in if (class(best\_nc) == "numeric") print(best\_nc) else if  
## (class(best\_nc) == : the condition has length > 1 and only the first element  
## will be used

## Warning in if (class(best\_nc) == "matrix") {: the condition has length > 1 and  
## only the first element will be used

## Among all indices:   
## ===================  
## \* 2 proposed 0 as the best number of clusters  
## \* 8 proposed 2 as the best number of clusters  
## \* 5 proposed 3 as the best number of clusters  
## \* 3 proposed 5 as the best number of clusters  
## \* 3 proposed 6 as the best number of clusters  
## \* 1 proposed 14 as the best number of clusters  
## \* 4 proposed 15 as the best number of clusters  
##   
## Conclusion  
## =========================  
## \* According to the majority rule, the best number of clusters is 2 .



fviz\_nbclust(liver\_scale, kmeans, method = "wss") +  
 geom\_vline(xintercept = 2, linetype = 2) +  
 labs(title= "Elbow Method: Optimal Number Of Clusters K=2",  
 subtitle = "Partitional Clustering - K-Means")



set.seed(123)  
(km.res<- kmeans(liver\_scale, 2, nstart = 25))

## K-means clustering with 2 clusters of sizes 517, 62  
##   
## Cluster means:  
## Age Total\_Bilirubin Direct\_Bilirubin Alkaline\_Phosphotase  
## 1 -0.002204957 -0.2651922 -0.2757749 -0.08439623  
## 2 0.018386492 2.2113604 2.2996070 0.70375568  
## Alamine\_Aminotransferase Aspartate\_Aminotransferase Total\_Protiens Albumin  
## 1 -0.1661591 -0.1540735 0.01545215 0.0715517  
## 2 1.3855523 1.2847745 -0.12885100 -0.5966488  
## Albumin\_and\_Globulin\_Ratio  
## 1 0.0632937  
## 2 -0.5277878  
##   
## Clustering vector:  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40   
## 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1   
## 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60   
## 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1   
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100   
## 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 2 1 1   
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2   
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140   
## 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1   
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180   
## 2 2 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 2 2 1   
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2   
## 201 202 203 204 205 206 207 208 209 211 212 213 214 215 216 217 218 219 220 221   
## 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1   
## 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 243 244 245 246 247 248 249 250 251 252 253 255 256 257 258 259 260 261 262 263   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1   
## 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283   
## 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1   
## 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 304 305 306 307 308 309 310 311 312 314 315 316 317 318 319 320 321 322 323 324   
## 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1   
## 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384   
## 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1   
## 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464   
## 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1   
## 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2   
## 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524   
## 2 2 2 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1   
## 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544   
## 1 1 1 1 1 1 1 2 1 2 1 2 2 1 1 1 1 1 1 1   
## 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564   
## 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 1   
## 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583   
## 1 2 1 1 2 1 1 1 2 1 2 2 2 2 1 1 1 1 1   
##   
## Within cluster sum of squares by cluster:  
## [1] 2610.734 1556.996  
## (between\_SS / total\_SS = 19.9 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

aggregate(liver\_sub, by=list(cluster=km.res$cluster), mean)

## cluster Age Total\_Bilirubin Direct\_Bilirubin Alkaline\_Phosphotase  
## 1 1 44.74662 1.66383 0.7174081 270.8104  
## 2 2 45.08065 17.08710 7.9709677 462.7742  
## Alamine\_Aminotransferase Aspartate\_Aminotransferase Total\_Protiens Albumin  
## 1 50.68859 65.75629 6.498453 3.195358  
## 2 334.93548 482.80645 6.341935 2.664516  
## Albumin\_and\_Globulin\_Ratio  
## 1 0.9672921  
## 2 0.7783871

As the results shows first cluster contains lower units of variables.

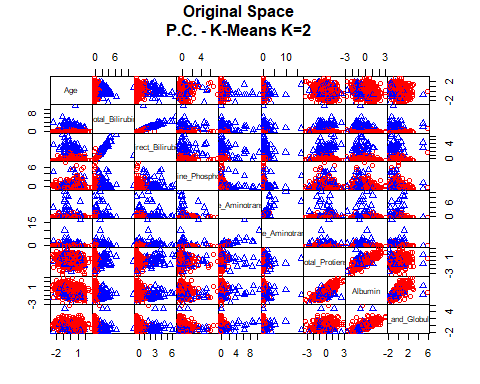
dd <- cbind(liver\_sub, cluster = km.res$cluster)  
head(dd)

## Age Total\_Bilirubin Direct\_Bilirubin Alkaline\_Phosphotase  
## 1 65 0.7 0.1 187  
## 2 62 10.9 5.5 699  
## 3 62 7.3 4.1 490  
## 4 58 1.0 0.4 182  
## 5 72 3.9 2.0 195  
## 6 46 1.8 0.7 208  
## Alamine\_Aminotransferase Aspartate\_Aminotransferase Total\_Protiens Albumin  
## 1 16 18 6.8 3.3  
## 2 64 100 7.5 3.2  
## 3 60 68 7.0 3.3  
## 4 14 20 6.8 3.4  
## 5 27 59 7.3 2.4  
## 6 19 14 7.6 4.4  
## Albumin\_and\_Globulin\_Ratio cluster  
## 1 0.90 1  
## 2 0.74 2  
## 3 0.89 1  
## 4 1.00 1  
## 5 0.40 1  
## 6 1.30 1

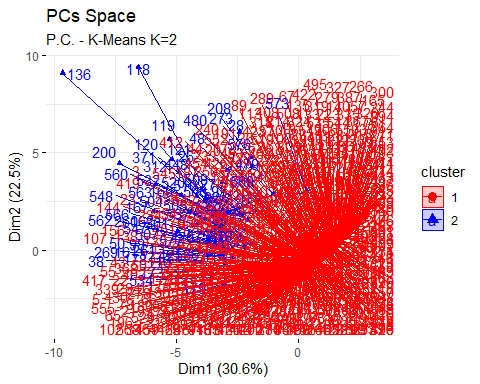
cl <- km.res$cluster  
table(cl)

## cl  
## 1 2   
## 517 62

pairs(liver\_scale, gap=0, pch=cl, main="Original Space\nP.C. - K-Means K=2",  
 cex.main= 1, col=c("red", "blue")[cl])



fviz\_cluster(km.res, data = liver\_scale, palette = c("red", "blue"),  
 ellipse.type = "euclid", star.plot = TRUE, repel = TRUE,  
 main= "PCs Space", ggtheme = theme\_minimal()) +  
 labs(subtitle = "P.C. - K-Means K=2")



According to the function NbClust, the best number of clusters, applying partitional clustering using K-means method is 2. K-means clustering suggests 2 clusters of sizes 517 in cluster 1, and 62 in cluster 2. The quality between different clusters is 19.9 % of the total variability is explained by the separation between clusters. In the PCs space, there is no separation between clusters. To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

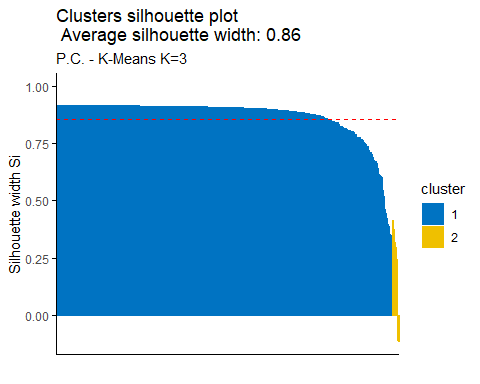
##### Silhouette width

hclust<- eclust(liver\_sub, k=2, "kmeans", nstart=25, graph = FALSE)  
silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.8566261

fviz\_silhouette(hclust, palette = "jco", ggtheme = theme\_classic()) +  
 labs(subtitle = "P.C. - K-Means K=3")

## cluster size ave.sil.width  
## 1 1 568 0.87  
## 2 2 11 0.18



silinfo$clus.avg.widths

## [1] 0.8697741 0.1777092

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu <- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## cluster neighbor sil\_width  
## 200 2 1 -0.003525611  
## 26 2 1 -0.108790427  
## 27 2 1 -0.108790427  
## 480 2 1 -0.115812579

The value of average silhouette width indicates that in average the units are well enough clustered. In particular, in cluster 1 (blue cluster) the units are on average the same silhouette value with respect to the silhouette width, in cluster 2 (the yellow one) the units are below average the silhouette value with respect to silhouette width. According to the above index, 4 units that belong to cluster 2 are not well clustered, they should belong to the cluster 1.

##### Dunn index

stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.1439413

According to the Dunn index, the units are not clustered well enough.

#### External Validation Measures

##### Confusion matrix

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2  
## 0 403 11  
## 1 165 0

According to the Confusion matrix, there is not a perfect agreement between the nominal variable Liver\_Disease and the cluster solution. For the liver disease, data has been classified mostly in cluster 1, 568 units in cluster 1, and 11 in cluster 2. For the patients having liver disease classified mostly in cluster 1 while cluster 2 has 0 value. Safe to say data are not well balanced in both cluster.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)   
stats<- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.02118264

According to the Correct Rand Index, there is no agreement between the numerical values and the cluster solution. From -1 to +1, the agreement is very close to 0.

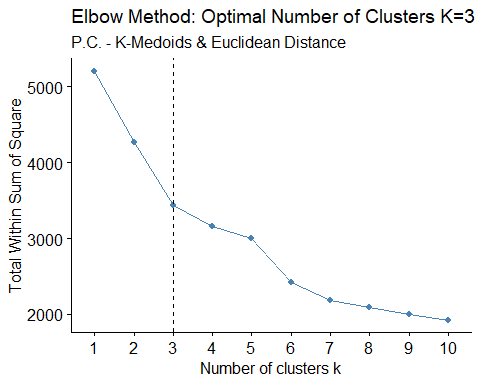
##### Meila’s VI Index

stats$vi

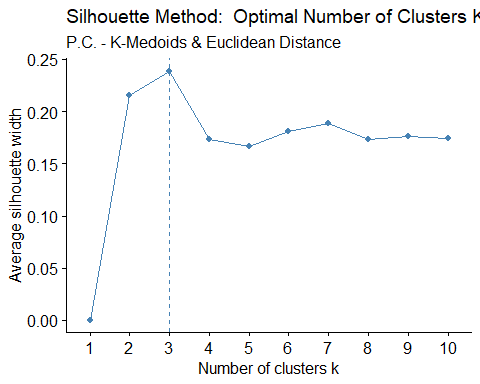
## [1] 0.6788131

### Partitioning Around Medoids (PAM) & Euclidean Distance Method

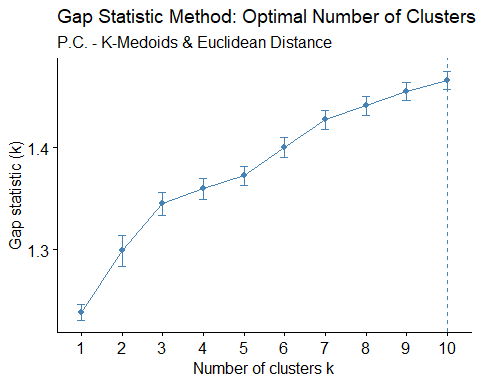
fviz\_nbclust(liver\_scale, cluster::pam, method = "wss") +  
 geom\_vline(xintercept = 3, linetype = 2) +  
 labs(title = "Elbow Method: Optimal Number of Clusters K=3",  
 subtitle="P.C. - K-Medoids & Euclidean Distance", cex.sub= 0.5)



fviz\_nbclust(liver\_scale, cluster::pam, method = "silhouette") +  
 labs(title = "Silhouette Method: Optimal Number of Clusters K=3",  
 subtitle="P.C. - K-Medoids & Euclidean Distance", cex.sub= 0.5)



fviz\_nbclust(liver\_scale, cluster::pam, method = "gap\_stat", nboot = 500) +  
 labs(title = "Gap Statistic Method: Optimal Number of Clusters K",  
 subtitle="P.C. - K-Medoids & Euclidean Distance", cex.sub= 0.5)



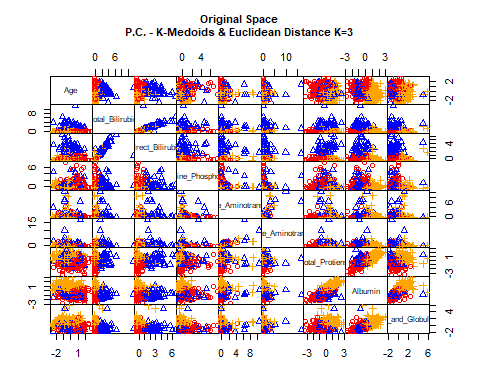
library(cluster)  
set.seed(123)  
(pam.res <- pam(liver\_scale, 3, metric = "euclidean"))

## Medoids:  
## ID Age Total\_Bilirubin Direct\_Bilirubin Alkaline\_Phosphotase  
## 246 244 0.56822453 -0.3878422 -0.4594811 -0.005609041  
## 537 533 0.07506058 2.0046880 2.0258743 -0.264270225  
## 323 319 -0.54139437 -0.2593842 -0.3529658 -0.354596353  
## Alamine\_Aminotransferase Aspartate\_Aminotransferase Total\_Protiens  
## 246 -0.36098402 -0.3188356 -0.3519068  
## 537 -0.07711464 0.3780765 0.3856644  
## 323 -0.24634446 -0.2636346 0.5700572  
## Albumin Albumin\_and\_Globulin\_Ratio  
## 246 -0.4261076 -0.4601612  
## 537 -0.6778590 -1.0859589  
## 323 0.9585247 0.7914341  
## Clustering vector:  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 1 2 1 1 1 3 3 3 3 1 1 1 1 3 1 1 3 3 1 1   
## 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40   
## 1 1 1 1 3 1 1 3 1 1 1 1 1 1 1 3 3 2 1 1   
## 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60   
## 1 1 1 1 1 1 3 3 1 2 1 3 3 3 2 2 3 1 1 1   
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80   
## 1 3 3 1 1 1 3 3 3 1 3 1 1 3 1 1 1 1 3 1   
## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100   
## 1 3 1 3 3 3 3 1 3 1 2 2 2 1 1 1 3 1 3 3   
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120   
## 1 1 3 1 1 1 1 1 1 1 1 1 3 1 1 1 1 2 2 2   
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140   
## 2 2 3 1 3 1 1 1 1 1 1 1 3 1 3 2 1 1 1 3   
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160   
## 1 1 1 1 1 1 1 1 3 3 3 1 1 3 1 2 1 1 1 3   
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180   
## 2 2 1 3 3 1 2 1 2 1 1 3 2 3 3 3 1 2 2 1   
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200   
## 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 2   
## 201 202 203 204 205 206 207 208 209 211 212 213 214 215 216 217 218 219 220 221   
## 1 1 1 3 3 3 1 3 1 1 1 3 3 3 3 1 3 1 3 1   
## 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241   
## 3 3 3 1 3 3 3 1 1 3 3 1 1 3 1 3 3 3 1 3   
## 243 244 245 246 247 248 249 250 251 252 253 255 256 257 258 259 260 261 262 263   
## 3 3 1 1 2 1 1 1 1 1 3 3 1 1 1 3 2 2 3 3   
## 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283   
## 1 3 3 3 3 2 1 3 3 2 3 3 3 3 1 3 3 1 3 1   
## 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303   
## 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1   
## 304 305 306 307 308 309 310 311 312 314 315 316 317 318 319 320 321 322 323 324   
## 3 3 3 1 1 3 3 3 2 3 3 3 3 3 3 3 3 1 3 3   
## 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344   
## 1 1 3 3 1 3 1 1 1 3 3 3 1 1 1 1 1 1 3 1   
## 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364   
## 3 1 3 3 1 3 1 1 3 3 3 3 3 1 1 3 3 1 1 3   
## 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384   
## 3 1 3 1 3 3 2 3 3 3 3 3 3 3 1 1 1 3 1 1   
## 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404   
## 1 3 3 3 3 1 1 1 1 3 1 1 1 1 3 3 1 2 3 3   
## 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424   
## 3 3 3 3 3 1 1 1 1 1 1 1 1 3 1 1 2 3 3 3   
## 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444   
## 1 1 1 1 1 1 1 1 3 3 3 3 3 1 1 1 3 1 1 1   
## 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464   
## 1 3 1 2 1 1 1 3 1 1 3 3 3 1 1 1 1 3 1 3   
## 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484   
## 1 1 1 3 1 1 3 1 1 3 1 3 1 1 3 3 3 3 3 3   
## 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504   
## 1 1 1 1 1 1 3 3 1 1 3 1 3 1 3 1 1 1 1 2   
## 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524   
## 2 2 2 3 2 2 3 3 1 1 3 1 1 3 3 2 3 1 3 1   
## 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544   
## 3 3 2 3 1 1 3 2 1 1 3 2 2 3 1 1 1 1 3 3   
## 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564   
## 1 1 2 2 3 3 1 1 1 2 1 1 1 1 3 2 2 2 2 1   
## 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583   
## 1 2 3 1 2 3 1 1 3 1 2 2 2 2 1 3 1 3 3   
## Objective function:  
## build swap   
## 2.066652 2.005459   
##   
## Available components:  
## [1] "medoids" "id.med" "clustering" "objective" "isolation"   
## [6] "clusinfo" "silinfo" "diss" "call" "data"

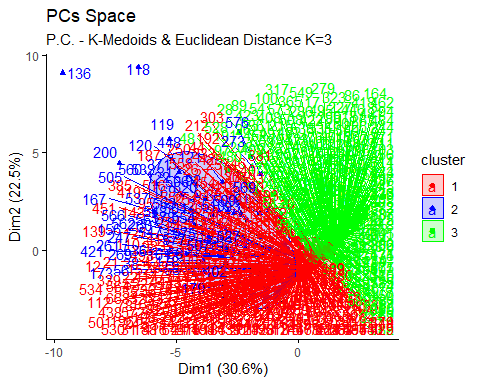
pam.res$clusinfo

## size max\_diss av\_diss diameter separation  
## [1,] 290 7.614993 1.889974 9.601375 0.2933832  
## [2,] 57 17.925607 3.792078 21.013509 0.9263908  
## [3,] 232 9.598995 1.710860 10.751924 0.2933832

cc <- pam.res$cluster  
pairs(liver\_scale, gap=0, pch=cc,  
 main="Original Space\nP.C. - K-Medoids & Euclidean Distance K=3",  
 cex.main= 0.7, col=c("red", "blue", "orange")[cc])



fviz\_cluster(pam.res, palette = c("red", "blue", "green"), ellipse.type = "t",  
 repel = TRUE, main= "PCs Space", ggtheme = theme\_classic()) +  
 labs(subtitle = "P.C. - K-Medoids & Euclidean Distance K=3", cex.sub= 0.5)



table(cc)

## cc  
## 1 2 3   
## 290 57 232

Applying partitioning clustering method and using PAM algorithm and euclidean distance, clusters are composed in this way: cluster 1 with 290 units, cluster 2 with 57 units and cluster 3 with 232 units. To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

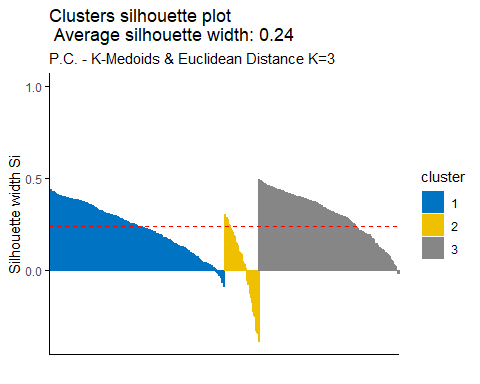
##### Silhouette width

hclust <- eclust(liver\_sub, k = 3, "pam", graph = FALSE, hc\_metric = "euclidean")  
silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.6563575

fviz\_silhouette(pam.res, palette = "jco", ggtheme = theme\_classic()) +  
 labs(subtitle = "P.C. - K-Medoids & Euclidean Distance K=3",  
 cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 290 0.23  
## 2 2 57 0.03  
## 3 3 232 0.30



silinfo$clus.avg.widths

## [1] 0.79842949 0.17747402 -0.06541546

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu <- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## cluster neighbor sil\_width  
## 385 2 1 -0.03525340  
## 579 2 1 -0.04082789  
## 250 2 1 -0.04883615  
## 3 2 1 -0.05143776  
## 177 2 1 -0.07069781  
## 36 2 1 -0.07642448  
## 110 2 1 -0.09948974  
## 22 2 1 -0.11948583  
## 251 2 1 -0.12733909  
## 272 2 1 -0.12901159  
## 66 2 1 -0.14092109  
## 337 2 1 -0.14469730  
## 509 2 1 -0.17036132  
## 536 2 1 -0.17576401  
## 157 2 1 -0.17670219  
## 419 2 1 -0.24048823  
## 548 2 1 -0.29425858  
## 303 2 1 -0.34485285  
## 367 2 1 -0.39104500  
## 34 2 1 -0.41151262  
## 35 2 1 -0.41151262  
## 339 2 1 -0.42249684  
## 200 3 2 -0.02551749  
## 561 3 1 -0.06630383  
## 273 3 1 -0.09078114  
## 520 3 1 -0.18045397  
## 97 3 1 -0.19330062  
## 510 3 1 -0.28699184  
## 100 3 1 -0.32409255  
## 121 3 1 -0.33688590  
## 559 3 1 -0.37265836  
## 17 3 1 -0.38860134  
## 562 3 1 -0.42656660  
## 94 3 1 -0.44157478  
## 495 3 1 -0.46335807  
## 563 3 1 -0.47797999  
## 71 3 1 -0.49335725  
## 53 3 1 -0.52732712  
## 236 3 1 -0.53899193  
## 482 3 1 -0.54486317

The value of average silhouette width indicates that in average the units are not well enough clustered. In particular, in cluster 1 the units are in average having the same silhouette value with respect to the silhouette width, in cluster 2 the units have in average a lowest value with respect to the silhouette width while in cluster 3 the units are in average having the higher value with respect to the silhouette width. According to the index, 40 units are not well clustered, units that belong to cluster 2 and 3, should belong to cluster 1.

##### Dunn Index

stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.02673178

According to the Dunn index, the units are not clustered well enough.

#### External validation Measures

##### Confusion Matrix

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2 3  
## 0 307 70 37  
## 1 154 11 0

According to the Confusion matrix, there is not a perfect agreement between the nominal variable Liver\_Disease and the cluster solution. A large number of patients not having liver disease 461 has been classified in cluster 1, 81 in cluster 2, and 37 in cluster 3. For the patients who have liver disease mostly classified in cluster 1 with 154 units and 11 units in cluster 2 but none in cluster 3.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats<- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.08037447

According to the Correct Rand Index, there is no agreement between the numerical values and the cluster solution. From -1 to +1, the agreement is very close to 0.

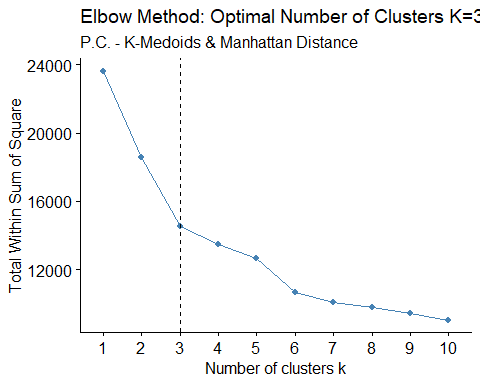
##### Meila’s VI Index

stats$vi

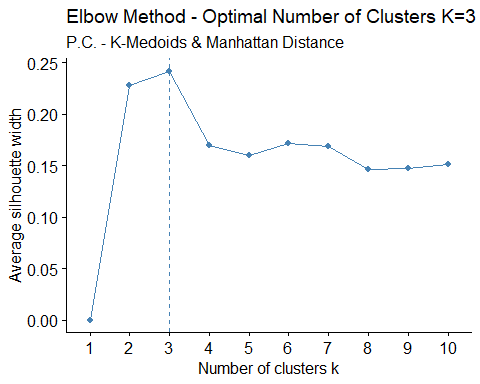
## [1] 1.160317

### Partitioning Around Medoids (PAM) & Manhattan Distance

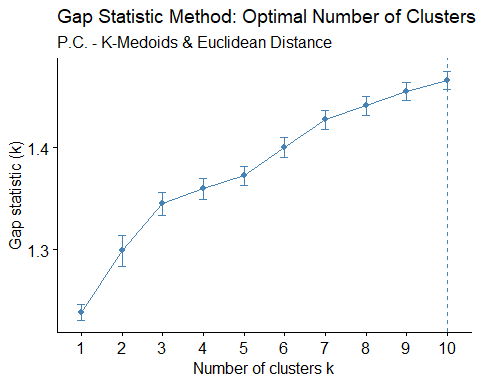
set.seed(123)  
fviz\_nbclust(liver\_scale, cluster::pam, method = "wss",  
 diss = dist(liver\_scale, method = "manhattan")) +  
 geom\_vline(xintercept = 3, linetype = 2) +  
 labs(title = "Elbow Method: Optimal Number of Clusters K=3",  
 subtitle="P.C. - K-Medoids & Manhattan Distance", cex.sub= 0.5)



fviz\_nbclust(liver\_scale, cluster::pam, method = "silhouette",  
 diss = dist(liver\_scale, method = "manhattan")) +  
 labs(title = "Elbow Method - Optimal Number of Clusters K=3",  
 subtitle="P.C. - K-Medoids & Manhattan Distance", cex.sub= 0.5)



set.seed(123)  
fviz\_nbclust(liver\_scale, cluster::pam, method = "gap\_stat", nboot = 500,  
 diss=dist(liver\_scale, method = "manhattan")) +  
 labs(title = "Gap Statistic Method: Optimal Number of Clusters K",  
 subtitle="P.C. - K-Medoids & Euclidean Distance", cex.sub= 0.5)



In order to find the optimal number of clusters, three indices were used. The elbow method seem suggest k=3, Silhouette method suggests 3 clusters, Gap statistics 10 clusters. It was decided to proceed by identifying 3 clusters.

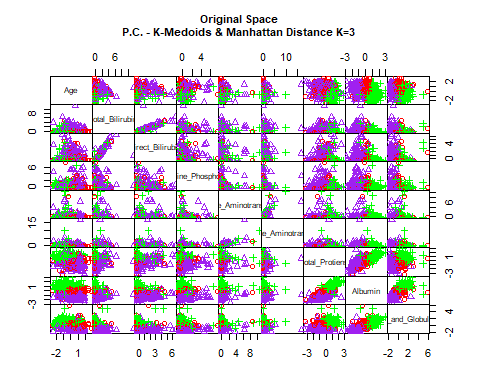
library(cluster)  
set.seed(123)  
(pam.res <- pam(liver\_scale, 3, metric="manhattan"))

## Medoids:  
## ID Age Total\_Bilirubin Direct\_Bilirubin Alkaline\_Phosphotase  
## 450 446 0.1983516 -0.38784221 -0.4594811 -0.4859798  
## 112 112 0.1983516 -0.01852546 0.0375900 -0.1410982  
## 176 176 -0.8496218 -0.40389946 -0.4594811 -0.3833365  
## Alamine\_Aminotransferase Aspartate\_Aminotransferase Total\_Protiens  
## 450 -0.3009347 -0.28778505 -0.2597104  
## 112 -0.2627215 0.01927028 -0.7206924  
## 176 -0.2081313 -0.27398481 0.7544500  
## Albumin Albumin\_and\_Globulin\_Ratio  
## 450 -0.04848061 0.1656364  
## 112 -1.18136162 -1.0233792  
## 176 1.08440040 0.7914341  
## Clustering vector:  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 1 2 1 1 2 3 3 3 3 1 1 2 1 3 1 2 3 3 1 1   
## 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40   
## 2 2 1 2 1 2 2 3 1 1 2 1 1 2 2 3 3 2 2 1   
## 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60   
## 1 1 2 1 1 1 3 1 1 2 1 1 3 3 2 2 3 1 2 1   
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80   
## 1 1 1 2 2 2 1 3 1 1 1 2 2 1 1 1 2 2 1 2   
## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100   
## 2 1 1 1 3 3 3 1 3 1 1 2 2 2 2 1 1 2 1 3   
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120   
## 2 2 3 1 1 2 2 2 2 1 2 2 1 2 2 2 2 3 2 2   
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140   
## 3 2 1 1 3 2 2 1 3 2 2 2 3 1 3 2 2 2 2 3   
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160   
## 1 2 1 1 1 1 2 1 3 3 3 1 1 1 2 2 1 2 2 3   
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180   
## 2 2 2 3 3 2 2 2 1 1 1 1 2 3 3 3 2 2 2 2   
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200   
## 2 2 2 1 3 1 2 2 2 1 1 1 1 1 1 2 2 1 1 2   
## 201 202 203 204 205 206 207 208 209 211 212 213 214 215 216 217 218 219 220 221   
## 1 1 1 3 1 3 2 3 1 1 1 1 3 3 1 1 3 2 1 1   
## 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241   
## 3 3 3 1 3 3 1 1 1 3 3 2 1 3 2 3 3 3 1 3   
## 243 244 245 246 247 248 249 250 251 252 253 255 256 257 258 259 260 261 262 263   
## 3 3 1 1 2 1 1 1 1 2 3 3 1 1 1 3 2 2 3 3   
## 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283   
## 1 3 3 1 3 2 1 3 1 3 3 3 3 3 1 3 3 1 3 1   
## 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303   
## 2 3 3 3 3 1 3 3 3 1 3 3 3 3 3 3 3 3 1 1   
## 304 305 306 307 308 309 310 311 312 314 315 316 317 318 319 320 321 322 323 324   
## 3 3 1 2 1 3 3 3 1 3 3 3 1 1 3 3 3 1 3 3   
## 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344   
## 1 1 3 3 1 3 1 2 2 1 3 3 1 1 1 2 1 1 3 2   
## 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364   
## 3 2 1 3 1 1 1 2 3 3 3 3 3 1 1 3 1 1 1 3   
## 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384   
## 3 1 3 1 3 3 1 3 3 3 3 1 3 1 1 1 1 1 1 1   
## 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404   
## 2 3 3 1 3 1 1 1 1 3 2 1 1 1 3 1 1 2 3 1   
## 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424   
## 3 3 3 3 1 1 2 1 2 2 2 2 1 3 2 2 2 3 1 3   
## 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444   
## 1 1 1 1 1 2 2 1 3 3 3 3 1 1 2 1 3 2 2 1   
## 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464   
## 1 3 1 2 1 1 1 1 1 1 3 3 3 1 2 2 2 3 1 3   
## 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484   
## 2 2 2 1 2 2 3 1 2 3 2 3 1 1 3 3 3 3 3 3   
## 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504   
## 2 2 1 1 1 1 1 3 3 2 3 1 3 1 3 2 2 2 1 2   
## 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524   
## 2 1 2 3 2 1 3 1 1 1 3 2 1 3 1 2 3 2 1 1   
## 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544   
## 3 3 2 1 2 2 3 2 1 2 3 2 2 3 1 1 1 1 3 3   
## 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564   
## 1 1 2 2 1 3 2 1 1 2 1 1 1 1 3 2 2 2 2 2   
## 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583   
## 1 2 3 2 3 3 2 1 3 2 2 3 2 2 2 1 1 3 3   
## Objective function:  
## build swap   
## 4.145986 4.145986   
##   
## Available components:  
## [1] "medoids" "id.med" "clustering" "objective" "isolation"   
## [6] "clusinfo" "silinfo" "diss" "call" "data"

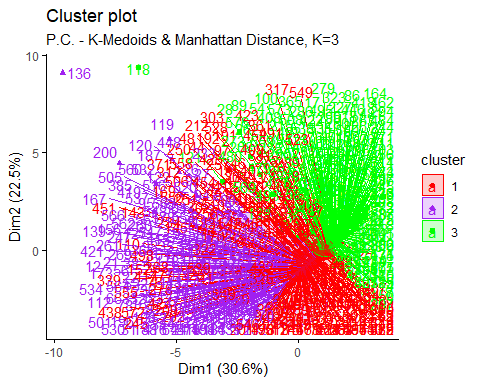
pam.res$clusinfo

## size max\_diss av\_diss diameter separation  
## [1,] 229 15.28487 3.307738 27.59425 0.5847884  
## [2,] 164 32.40721 5.989488 39.37034 0.8141526  
## [3,] 186 27.52860 3.552567 36.70780 0.5847884

cm <- pam.res$cluster  
pairs(liver\_scale, gap=0,  
 main="Original Space\nP.C. - K-Medoids & Manhattan Distance K=3",  
 cex.main= 0.7, pch = cm, col = c("red", "purple", "green")[cm])



fviz\_cluster(pam.res, palette = c("red", "purple", "green"), ellipse.type = "t",  
 repel = TRUE, ggtheme = theme\_classic()) +  
 labs(subtitle = "P.C. - K-Medoids & Manhattan Distance, K=3", cex.sub= 0.5)



table(cm)

## cm  
## 1 2 3   
## 229 164 186

Applying partitioning clustering using PAM algorithm and euclidean distance, three clusters are composed in this way: cluster 1 with 290 units, cluster 2 with 57 units and cluster 3 with 232 units. To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analyzed as follows:

#### Internal Validation Measures

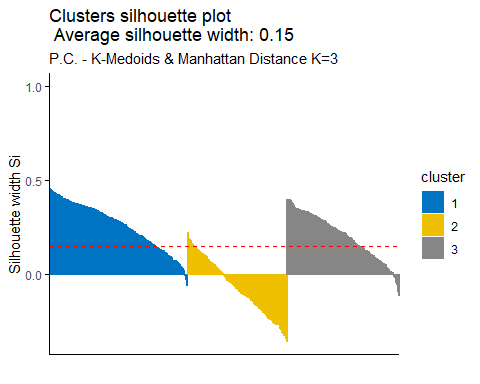
##### Silhouette width

hclust<- eclust(liver\_sub, k=3, "pam", graph = FALSE, hc\_metric = "manhattan")  
silinfo <- hclust$silinfo  
silinfo$avg.width

## [1] 0.6563575

fviz\_silhouette(pam.res, palette = "jco", ggtheme = theme\_classic()) +  
 labs(subtitle = "P.C. - K-Medoids & Manhattan Distance K=3", cex.sub= 0.5)

## cluster size ave.sil.width  
## 1 1 229 0.26  
## 2 2 164 -0.07  
## 3 3 186 0.20



silinfo$clus.avg.widths

## [1] 0.79842949 0.17747402 -0.06541546

sil <- hclust$silinfo$widths[, 1:3]  
neg\_sil\_index\_aver.eu <- which(sil[, "sil\_width"] < 0)  
sil[neg\_sil\_index\_aver.eu, , drop = FALSE]

## cluster neighbor sil\_width  
## 385 2 1 -0.03525340  
## 579 2 1 -0.04082789  
## 250 2 1 -0.04883615  
## 3 2 1 -0.05143776  
## 177 2 1 -0.07069781  
## 36 2 1 -0.07642448  
## 110 2 1 -0.09948974  
## 22 2 1 -0.11948583  
## 251 2 1 -0.12733909  
## 272 2 1 -0.12901159  
## 66 2 1 -0.14092109  
## 337 2 1 -0.14469730  
## 509 2 1 -0.17036132  
## 536 2 1 -0.17576401  
## 157 2 1 -0.17670219  
## 419 2 1 -0.24048823  
## 548 2 1 -0.29425858  
## 303 2 1 -0.34485285  
## 367 2 1 -0.39104500  
## 34 2 1 -0.41151262  
## 35 2 1 -0.41151262  
## 339 2 1 -0.42249684  
## 200 3 2 -0.02551749  
## 561 3 1 -0.06630383  
## 273 3 1 -0.09078114  
## 520 3 1 -0.18045397  
## 97 3 1 -0.19330062  
## 510 3 1 -0.28699184  
## 100 3 1 -0.32409255  
## 121 3 1 -0.33688590  
## 559 3 1 -0.37265836  
## 17 3 1 -0.38860134  
## 562 3 1 -0.42656660  
## 94 3 1 -0.44157478  
## 495 3 1 -0.46335807  
## 563 3 1 -0.47797999  
## 71 3 1 -0.49335725  
## 53 3 1 -0.52732712  
## 236 3 1 -0.53899193  
## 482 3 1 -0.54486317

The value of average silhouette width indicates that in average the units are not well enough clustered. In particular, in cluster 1 the units are in average having the same silhouette value with respect to the silhouette width, in cluster 2 the units have in average a lowest value with respect to the silhouette width while in cluster 3 the units are in average having the higher value with respect to the silhouette width. According to the index, 40 units are not well clustered, units that belong to cluster 2 and 3, should belong to cluster 1.

##### Dunn index

stats <- cluster.stats(dist(liver\_scale), hclust$cluster)  
stats$dunn

## [1] 0.02673178

According to the Dunn index, the units are not clustered well enough.

#### External Validation Measures

##### Confusion Matrix

table(liver$Liver\_Disease, hclust$cluster)

##   
## 1 2 3  
## 0 307 70 37  
## 1 154 11 0

According to the Confusion matrix, there is not a perfect agreement between the nominal variable Liver\_disease and the cluster solution. A large number of patients not having liver disease has been classified in cluster 1, 70 in cluster 2, and 37 have been classified in cluster 3. For patients having a liver disease, large number of data - 154 units has been classified in cluster 1, 11 in cluster 2, and 0 in cluster 3.

##### Correct Rand Index

liver.disease <- as.numeric(liver$Liver\_Disease)  
stats<- cluster.stats(d = dist(liver\_scale), liver.disease, hclust$cluster)  
stats$corrected.rand

## [1] -0.08037447

According to the Correct Rand Index, there is no agreement between the numerical values and the cluster solution. From -1 to +1, the agreement is very close to 0.

##### Meila’s VI Index

stats$vi

## [1] 1.160317

## Soft Clustering Approach

### Model-Based Clustering

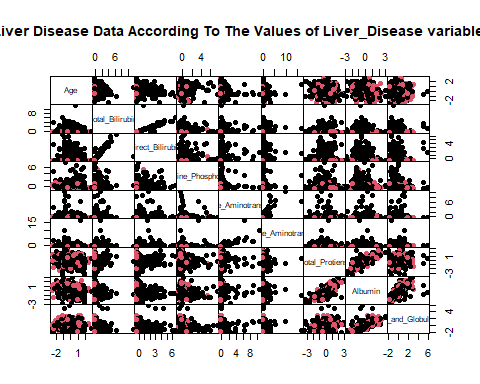
summary(liver$Liver\_Disease)

## 0 1   
## 414 165

head(liver)

## Age Gender Total\_Bilirubin Direct\_Bilirubin Alkaline\_Phosphotase  
## 1 65 0 0.7 0.1 187  
## 2 62 1 10.9 5.5 699  
## 3 62 1 7.3 4.1 490  
## 4 58 1 1.0 0.4 182  
## 5 72 1 3.9 2.0 195  
## 6 46 1 1.8 0.7 208  
## Alamine\_Aminotransferase Aspartate\_Aminotransferase Total\_Protiens Albumin  
## 1 16 18 6.8 3.3  
## 2 64 100 7.5 3.2  
## 3 60 68 7.0 3.3  
## 4 14 20 6.8 3.4  
## 5 27 59 7.3 2.4  
## 6 19 14 7.6 4.4  
## Albumin\_and\_Globulin\_Ratio Liver\_Disease  
## 1 0.90 0  
## 2 0.74 0  
## 3 0.89 0  
## 4 1.00 0  
## 5 0.40 0  
## 6 1.30 0

X <- data.matrix(liver\_sub)  
sX <- scale(X)  
pairs(sX, gap=0, pch = 16, col = as.numeric(liver$Liver\_Disease), cex.main = 0.9,  
 main="Liver Disease Data According To The Values of Liver\_Disease variable")



To evaluate if there is a relation between the categorical variable Liver\_Disease and the underlying clustering, the variable is deleted and the data are standardized. The data are visualized by pairwise scatterplots, in which the colors represent the two possible values of Liver\_Disease: 0 or 1. It seems difficult to distinguish separate groups. Different Parsimonious Gaussian mixtures are fitted on the standardized data by using the function Mclust() in R.

library(mclust)

## Warning: package 'mclust' was built under R version 4.0.5

## Package 'mclust' version 5.4.7  
## Type 'citation("mclust")' for citing this R package in publications.

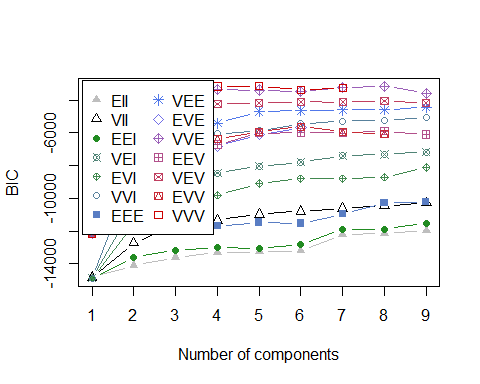
##   
## Attaching package: 'mclust'

## The following object is masked from 'package:psych':  
##   
## sim

library(psych)  
mod <- Mclust(liver\_scale)  
summary(mod$BIC)

## Best BIC values:  
## VVV,5 VVE,8 VVV,4  
## BIC -3155.656 -3157.139575 -3162.055214  
## BIC diff 0.000 -1.484033 -6.399673

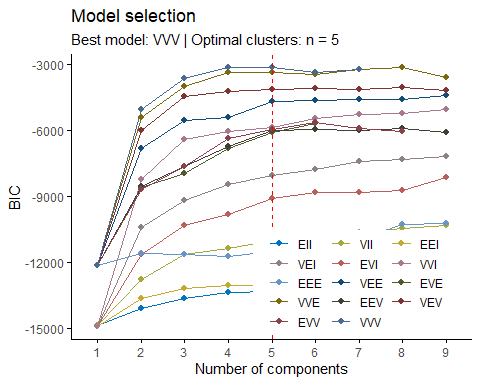
plot(mod, what = "BIC", ylim = range(mod$BIC, na.rm = TRUE),  
 legendArgs = list(x = "topleft"))



summary(mod)

## ----------------------------------------------------   
## Gaussian finite mixture model fitted by EM algorithm   
## ----------------------------------------------------   
##   
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 5  
## components:   
##   
## log-likelihood n df BIC ICL  
## -706.3293 579 274 -3155.656 -3206.669  
##   
## Clustering table:  
## 1 2 3 4 5   
## 91 48 106 235 99

fviz\_mclust(mod, "BIC", palette = "jco")

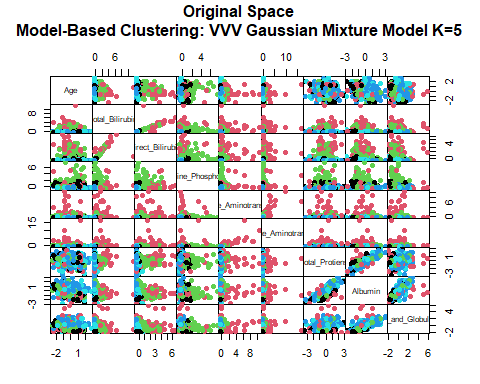


head(round(mod$z, 6), 20)

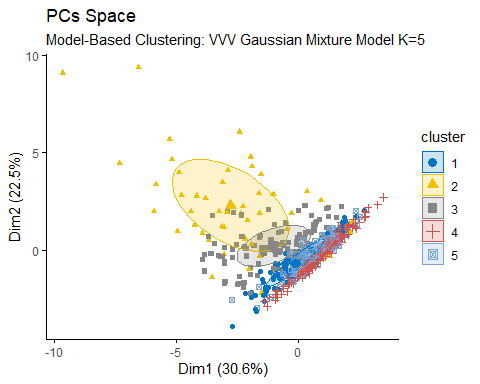
## [,1] [,2] [,3] [,4] [,5]  
## 1 0.001855 0.0e+00 0.000001 0.994983 0.003161  
## 2 0.000000 1.1e-05 0.999989 0.000000 0.000000  
## 3 0.000002 1.5e-05 0.999983 0.000000 0.000000  
## 4 0.074006 0.0e+00 0.000041 0.542297 0.383656  
## 5 0.999995 0.0e+00 0.000005 0.000000 0.000000  
## 6 0.294785 0.0e+00 0.000135 0.000000 0.705080  
## 7 0.002809 0.0e+00 0.000002 0.997005 0.000185  
## 8 0.005433 0.0e+00 0.000002 0.989533 0.005032  
## 9 0.004158 0.0e+00 0.000000 0.994037 0.001805  
## 10 0.056921 0.0e+00 0.000146 0.906932 0.036001  
## 11 0.025260 0.0e+00 0.000043 0.803215 0.171483  
## 12 0.999266 0.0e+00 0.000393 0.000000 0.000342  
## 13 0.667823 0.0e+00 0.001682 0.255157 0.075338  
## 14 0.003366 0.0e+00 0.000004 0.841177 0.155453  
## 15 0.003446 0.0e+00 0.000024 0.923137 0.073393  
## 16 0.000000 0.0e+00 0.003354 0.000001 0.996646  
## 17 0.000000 1.3e-04 0.999870 0.000000 0.000000  
## 18 0.997169 0.0e+00 0.000368 0.000000 0.002463  
## 19 0.000000 2.8e-05 0.999972 0.000000 0.000000  
## 20 0.000000 2.8e-05 0.999972 0.000000 0.000000

According to the penalized selection criterion called “BIC” (Bayesian Information Criterion), the three best Gaussian mixture models are: VVV with 5 clusters, VVE with 8 clusters, and VVV with 4 clusters. The number of clusters that maximizes the BIC of this model is 5, cluster 1 with 91 units, cluster 2 with 48 units, cluster 3 with 106 units, cluster 4 with 235 units and cluster 5 with 99 units.

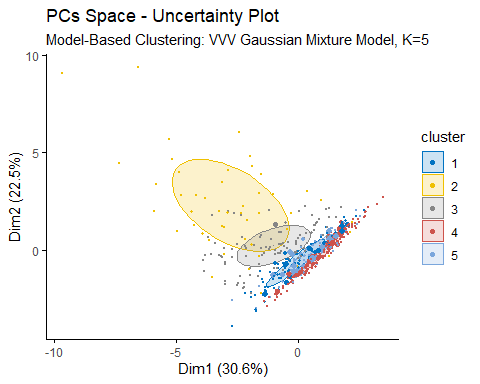
pairs(sX, gap=0, pch = 16, col = mod$classification, cex.main = 1,  
 main="Original Space\nModel-Based Clustering: VVV Gaussian Mixture Model K=5")



fviz\_mclust(mod, "classification", geom = "point", pointsize = 1.5,  
palette = "jco", main = "PCs Space") +  
labs(subtitle= "Model-Based Clustering: VVV Gaussian Mixture Model K=5")



fviz\_mclust(mod, "uncertainty", palette = "jco",  
 main = "PCs Space - Uncertainty Plot") +  
 labs(subtitle= "Model-Based Clustering: VVV Gaussian Mixture Model, K=5")



Both in the original space and in the PCs space there is a great separation between clusters. Furthermore, from the Uncertainty plot, it is noted that some units (big points) are problematic for the soft approach because they belong to different clusters with the same (or similar) probability.

#### External Validation Measures

##### Confusion Matrix

table(liver$Liver\_Disease, mod$classification)

##   
## 1 2 3 4 5  
## 0 68 46 100 131 69  
## 1 23 2 6 104 30

According to the Confusion matrix, there is a good agreement between the nominal variable Liver\_Disease and the cluster solution. A large number of patients not having liver disease (n = 131) has been classified in cluster 4. A large number of patients having liver disease (n=104) have also been classified in the same cluster 4.

##### Correct Rand Index

adjustedRandIndex(liver$Liver\_Disease, mod$classification)

## [1] -0.007492218

According the Correct Rand Index, there is a good agreement between the Liver\_Disease nominal variable and the cluster solution.

## The Best Clustering Algorithm

In order to choose the best clustering algorithm among those proposed, the clValid package of R. is used. The clValid function enables to compare clustering algorithms using two cluster validation measures is Internal Measures (Connectivity, Silhouette coefficient and Dunn index). The methods considered are: hierarchical method (Euclidean- Manhattan), k-Means, PAM (Euclidean-Manhattan) and Model-Based Clustering.

library(clValid)

## Warning: package 'clValid' was built under R version 4.0.5

clmethods <- c ("hierarchical", "kmeans", "pam")  
V\_eucl<-clValid(liver\_scale, nClust=2:6, clMethods= clmethods,  
 metric="euclidean", validation="internal")  
summary(V\_eucl)

##   
## Clustering Methods:  
## hierarchical kmeans pam   
##   
## Cluster sizes:  
## 2 3 4 5 6   
##   
## Validation Measures:  
## 2 3 4 5 6  
##   
## hierarchical Connectivity 3.0956 6.0246 14.5020 18.3171 20.7254  
## Dunn 0.5318 0.4789 0.2824 0.2824 0.2824  
## Silhouette 0.8040 0.6973 0.6493 0.6061 0.5948  
## kmeans Connectivity 10.5190 40.0226 49.2111 129.9921 132.8639  
## Dunn 0.1439 0.0599 0.0769 0.0291 0.0291  
## Silhouette 0.6403 0.4761 0.4769 0.2643 0.2664  
## pam Connectivity 138.1683 147.4345 228.6667 271.0968 273.3825  
## Dunn 0.0128 0.0140 0.0083 0.0102 0.0138  
## Silhouette 0.2155 0.2386 0.1730 0.1666 0.1814  
##   
## Optimal Scores:  
##   
## Score Method Clusters  
## Connectivity 3.0956 hierarchical 2   
## Dunn 0.5318 hierarchical 2   
## Silhouette 0.8040 hierarchical 2

According to the result, the best clustering algorithm using the Euclidean Distance is the Hierarchical Clustering method with 2 clusters.

clmethods <- c ("hierarchical", "kmeans", "pam")  
V\_eucl <- clValid(liver\_scale, nClust=2:6, clMethods= clmethods,  
 metric="manhattan", validation="internal")  
summary(V\_eucl)

##   
## Clustering Methods:  
## hierarchical kmeans pam   
##   
## Cluster sizes:  
## 2 3 4 5 6   
##   
## Validation Measures:  
## 2 3 4 5 6  
##   
## hierarchical Connectivity 3.0290 7.7619 15.3579 18.2869 31.9294  
## Dunn 0.5931 0.2270 0.2672 0.2672 0.1777  
## Silhouette 0.7722 0.6542 0.6251 0.5714 0.5384  
## kmeans Connectivity 11.6222 41.8865 54.5698 128.9794 134.8909  
## Dunn 0.1647 0.0600 0.0658 0.0249 0.0298  
## Silhouette 0.6274 0.5025 0.4998 0.2662 0.2681  
## pam Connectivity 154.1667 254.6214 256.1413 277.3071 275.6333  
## Dunn 0.0120 0.0149 0.0090 0.0062 0.0081  
## Silhouette 0.2438 0.1482 0.1719 0.1680 0.1841  
##   
## Optimal Scores:  
##   
## Score Method Clusters  
## Connectivity 3.0290 hierarchical 2   
## Dunn 0.5931 hierarchical 2   
## Silhouette 0.7722 hierarchical 2

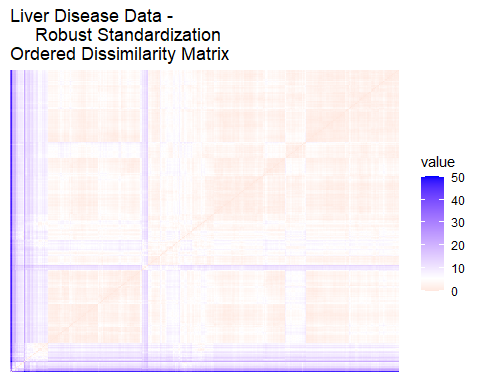
According to the result, the best clustering algorithm using the Manhattan Distance is the Hierarchical Clustering method with 2 clusters.

clmethods <- c ("hierarchical", "kmeans", "pam")  
V\_eucl <- clValid(liver\_scale, nClust=2:6, clMethods= clmethods,  
 metric="manhattan", validation="stability")  
summary(V\_eucl)

##   
## Clustering Methods:  
## hierarchical kmeans pam   
##   
## Cluster sizes:  
## 2 3 4 5 6   
##   
## Validation Measures:  
## 2 3 4 5 6  
##   
## hierarchical APN 0.0011 0.0029 0.0058 0.0079 0.0124  
## AD 7.6286 7.5371 7.3617 7.3082 7.1719  
## ADM 0.0158 0.0376 0.0770 0.0831 0.1713  
## FOM 0.9686 0.9618 0.9598 0.9419 0.9405  
## kmeans APN 0.0176 0.1602 0.2178 0.0967 0.1352  
## AD 7.4381 6.8605 6.7396 5.7955 5.9327  
## ADM 0.1187 0.5450 0.6793 0.3377 0.6452  
## FOM 0.9770 0.9632 0.9142 0.8012 0.8510  
## pam APN 0.1497 0.2030 0.1966 0.3021 0.4065  
## AD 6.8566 6.5238 5.9394 5.8733 5.7696  
## ADM 0.4090 0.6878 0.5010 0.7500 0.9540  
## FOM 0.9528 0.9346 0.9075 0.9025 0.9026  
##   
## Optimal Scores:  
##   
## Score Method Clusters  
## APN 0.0011 hierarchical 2   
## AD 5.7696 pam 6   
## ADM 0.0158 hierarchical 2   
## FOM 0.8012 kmeans 5

In this measure according to the APN and ADM the best method is Hierarchical with 2 clusters, according to AD the best method is pam with 6 clusters and according to FOM the best method is kmeans with 5 number of clusters.

scale\_rob <- scale(liver\_sub, center = apply(liver\_sub, 2, median),  
 scale = apply(liver\_sub, 2, meanabsdev))  
rownames(scale\_rob) <- rownames(liver)  
fviz\_dist(dist(scale\_rob), show\_labels = FALSE) +  
labs(title = "Liver Disease Data -   
 Robust Standardization\nOrdered Dissimilarity Matrix",  
 gradient = list(low = "blue", mid = "green", high = "red"))



hopkins(scale\_rob, n = nrow(scale\_rob)-1)

## $H  
## [1] 0.0817502

According to the Hopkins statistic (H) the data set is uniformly distributed because the values is close to 0, also the dissimilarity matrix image the data contain a clusters structure.

## Clustering Result

According to the proposed indices, among the clustering algorithms adopted, the Hierarchical Method, computed using the Euclidean Distance with 2 clusters seems to be the most suitable.