UNIVERSITÀ
degli STUDI
di CATANIA

Report Subject:

Data Analysis Method on “Heart Disease” Dataset

Student:

Anahita Esfandiaryfard

1000023614

Professor:

Antonio Punzo

Academic Year 2020-2021

Dataset

The dataset that is used is “Heart Disease” dataset.You can find dataset from here:“https://www.kaggle.com /ronitf/heart-disease-uci”.This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. The “goal” field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.The dataset includes 303 objects of 14 variables: age : age of patients in the dataset sex : gender of paitnets which defined by 1 for male and 0 for female cp : chest pain type of each patient reported which takes integer values from 0 to 4 trestbps: The resting blood pressure of patients which is a real number chol: serum cholesterol in mg/dl fbs: the boolean value which shows if patients fasting blood sugar is greater than 120 mg/dl or not restecg: resting electrocardiographic results which categorized in 3 group from 0 to 3 thalach: maximum heart rate achieved exang: exercise induced angina oldpeak: ST depression induced by exercise relative to rest oldpeak: the slope of the peak exercise ST segment number of major vessels (0-3) colored by fluoroscopy thal: 3 = normal; 6 = fixed defect; 7 = reversible defect

```
heart <- read.csv("heart.csv")
str(heart)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

From the structure we can see that the dataset is not structured in a perfect way,as you can see “sex”,“cp”,“restecg”,“exang”, “slope”,“ca”,“thal” and “target” does not have clear data.

```
heart$sex <- factor(heart$sex)
heart$cp <- factor(heart$cp)
heart$fbs <- factor(heart$fbs)
heart$restecg <- factor(heart$restecg)
heart$exang <- factor(heart$exang)
heart$slope <- factor(heart$slope)
heart$ca <- factor(heart$ca)
heart$thal <- factor(heart$thal)
heart$target <- factor(heart$target)
str(heart)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 2 2 ...
##  $ cp      : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
##  $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 1 ...
##  $ restecg : Factor w/ 3 levels "0","1","2": 1 2 1 2 2 2 1 2 2 2 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
```

```
##  $ exang   : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : Factor w/ 3 levels "0","1","2": 1 1 3 3 3 2 2 3 3 3 ...
##  $ ca      : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ thal    : Factor w/ 4 levels "0","1","2","3": 2 3 3 3 3 2 3 4 4 3 ...
##  $ target  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

The result shows us a better view about our data.

```
head(heart)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  63   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2  37   1  2      130  250   0       1     187     0     3.5     0  0    2
## 3  41   0  1      130  204   0       0     172     0     1.4     2  0    2
## 4  56   1  1      120  236   0       1     178     0     0.8     2  0    2
## 5  57   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6  57   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

```
any(is.na(heart))
```

```
## [1] FALSE
```

There is no NA in data so we do not need to omit our data. The first type of analysis which is conducted is the univariate one. After this, a preliminary analysis is carried out, in order to understand if it would be useful to run a Principal Component Analysis and a Cluster one on this dataset. So, further on, it is runned the Principal Component Analysis and the Cluster Analysis, which results are verified through the Cluster Validation. Finally, Model Based Clustering is carried out.

Univariate Analysis

Age

Age is a continuous variable in range(0,),

```
length(heart$age)
```

```
## [1] 303
```

```
min(heart$age)
```

```
## [1] 29
```

```
max(heart$age)
```

```
## [1] 77
```

The length of sample is 303, and the values are in range 29-77.

```
summary(heart$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   29.00   47.50   55.00   54.37   61.00   77.00
```

```r
sd(heart$age)
```

```
## [1] 9.082101
```

```r
var(heart$age)
```

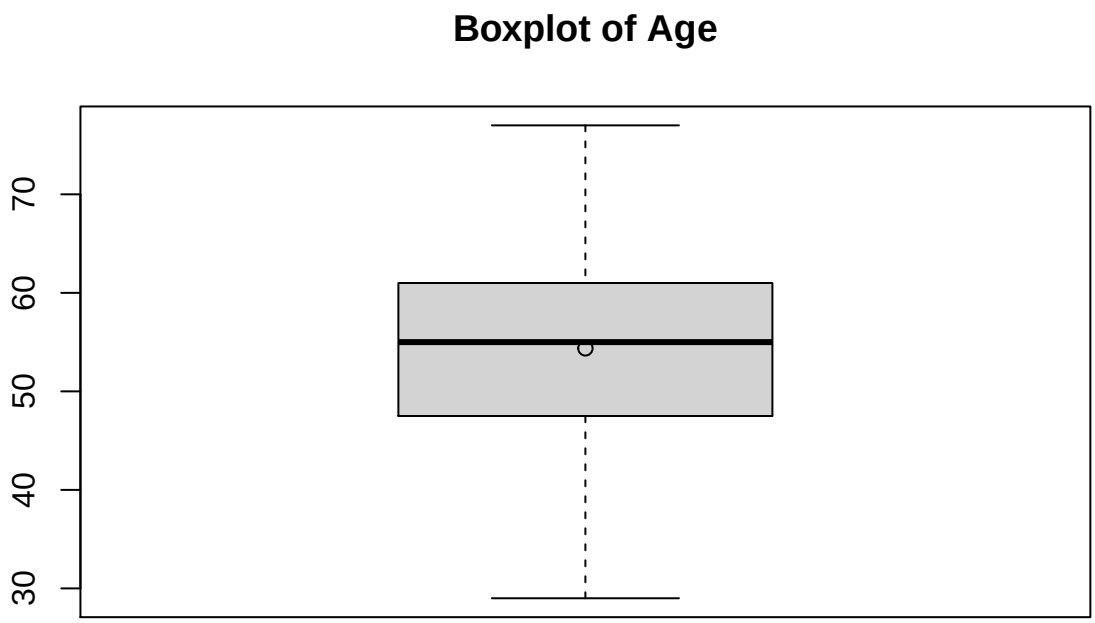
```
## [1] 82.48456
```

From the results we can see the Mean and Median are not equal so the distribution is asymmetrical or skewed. As we can see mean<median so the distribution should be negative.The skewness is minus so the prediction was right.

```r
library(e1071)
skewness(heart$age)
```

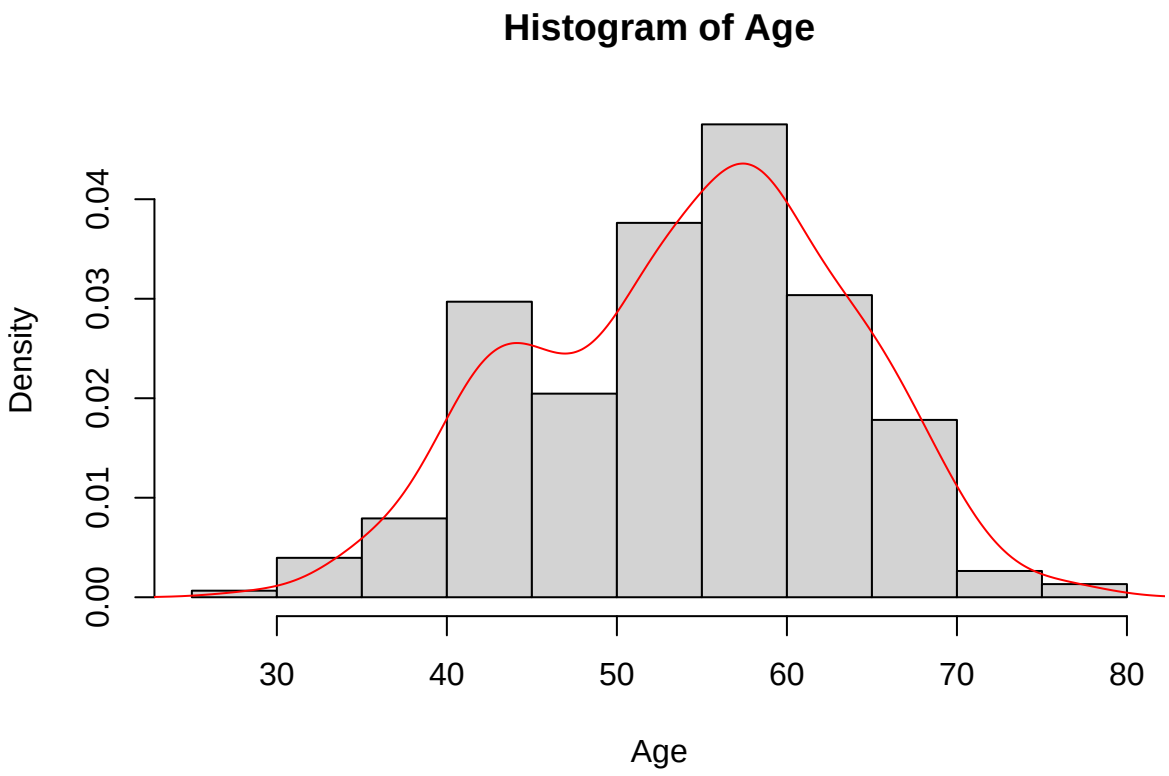
```
## [1] -0.2004632
```

```r
boxplot(heart$age, main = "Boxplot of Age")
points(mean(heart$age))
```



The box plot is a graphical representation used to describe the distribution of the variable which contains the first and third quartiles as the lower and upper ends of the plotted rectangle.The median divides the box into two parts. If we plot a box plot for our data the following plot appears. we can say that this variable has no outliers as there are no dots out of the line. Now, we can plot the histogram and look for the model that fits better the distribution:

```r
hist(heart$age, freq=FALSE, xlab= "Age", main = "Histogram of Age")
lines(density(heart$age), col="red")
```




```
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
```

As the dataset is related to heart disease from above plot we can realise that the distribution of data above mean is greater than below mean so we can assume that the range of heart disease in patients with age more than 54 is higher. If we look at the histogram of data we can see that the diagram has two peaks. Safe to say that the distribution is Bimodal.Also we can see that the left tail is larger which it was obvious for use as we saw below the skewness is negative.

Data fitting for Age

According to the domain of the variable, the distributions supported on the set of positive infinite real numbers are fitted on the data. Log-likelihood value, BIC and AIC are estimated in order to evaluate the fit of the distribution.

```
round(2*nrow(heart)**(1/3))
```

```
## [1] 13
```

```
library(gamlss)
```

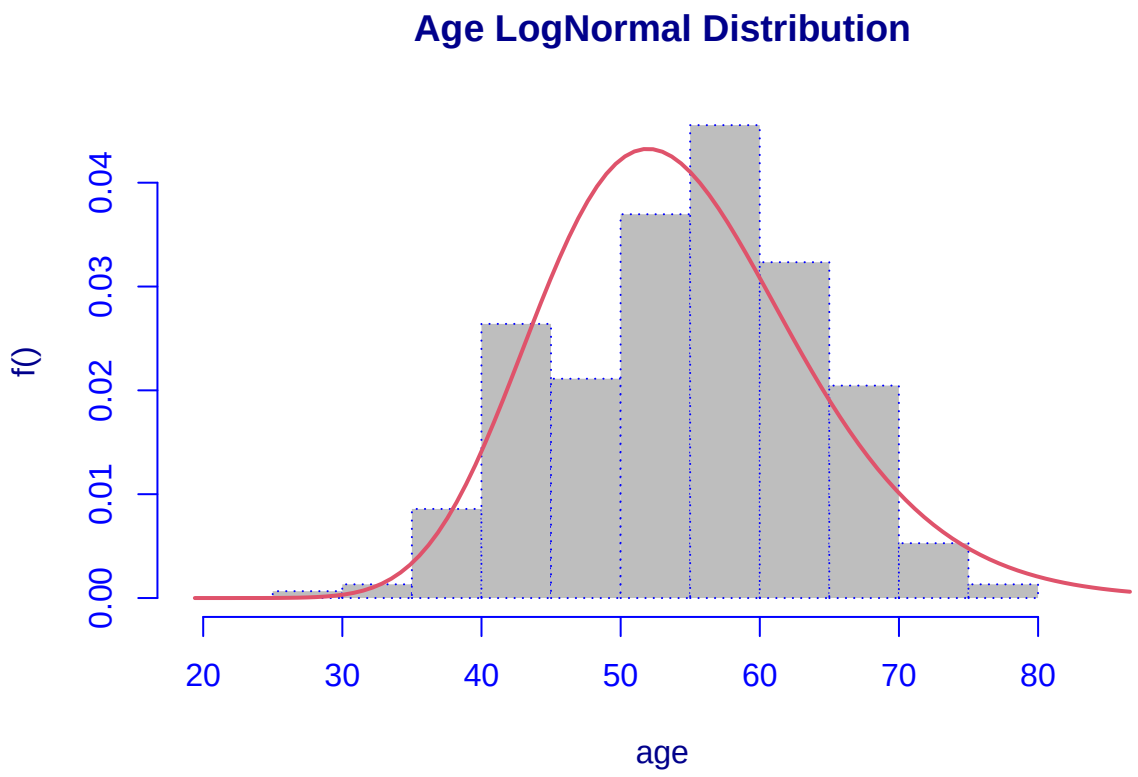
```
## Loading required package: splines
```

```
## Loading required package: gamlss.data
```

```
##
## Attaching package: 'gamlss.data'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep

## Loading required package: gamlss.dist

## Loading required package: MASS

## Loading required package: nlme

## Loading required package: parallel

##  **********   GAMLSS Version 5.2-0  **********

## For more on GAMLSS look at https://www.gamlss.com/

## Type gamlssNews() to see new features/changes/bug fixes.
```

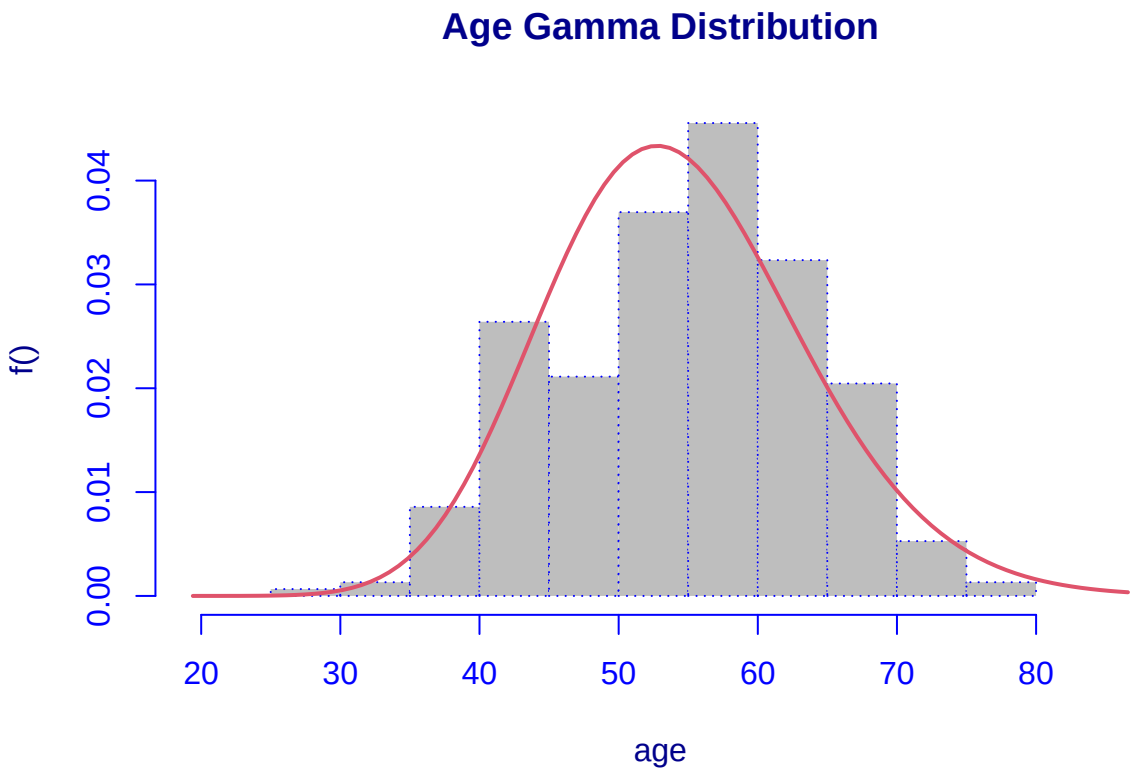
```
(age.LOGNO<- histDist(heart$age, xlab = "age", family=LOGNO, nbins=13,
         main = "Age LogNormal Distribution") )
```




```
##
## Family:  c("LOGNO", "Log Normal")
## Fitting method: "nlminb"
##
## Call:  gamlssML(formula = heart$age, family = "LOGNO")
##
## Mu Coefficients:
## [1]  3.981
## Sigma Coefficients:
```

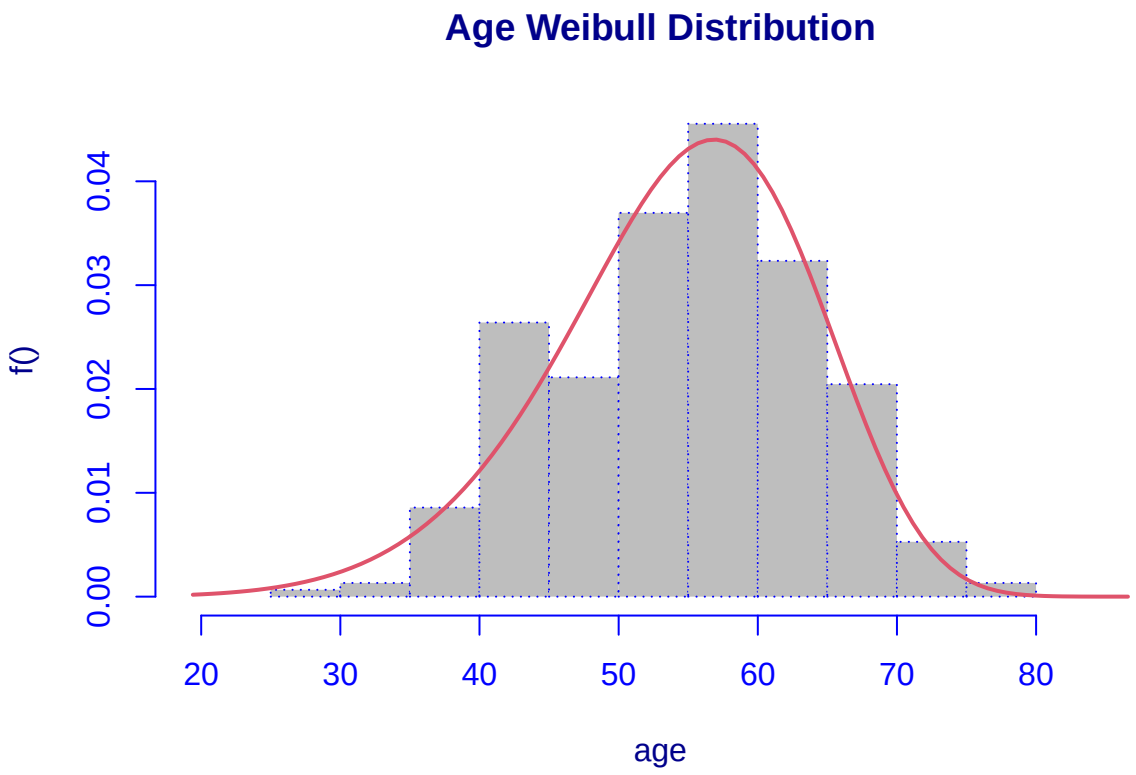
```
## [1]  -1.744
##
##  Degrees of Freedom for the fit: 2 Residual Deg. of Freedom   301
## Global Deviance:     2215.42
##             AIC:     2219.42
##             SBC:     2226.84
```

```
(age.GA<-histDist(heart$age, xlab = "age", family=GA, nbins=13,
         main = "Age Gamma Distribution"))
```

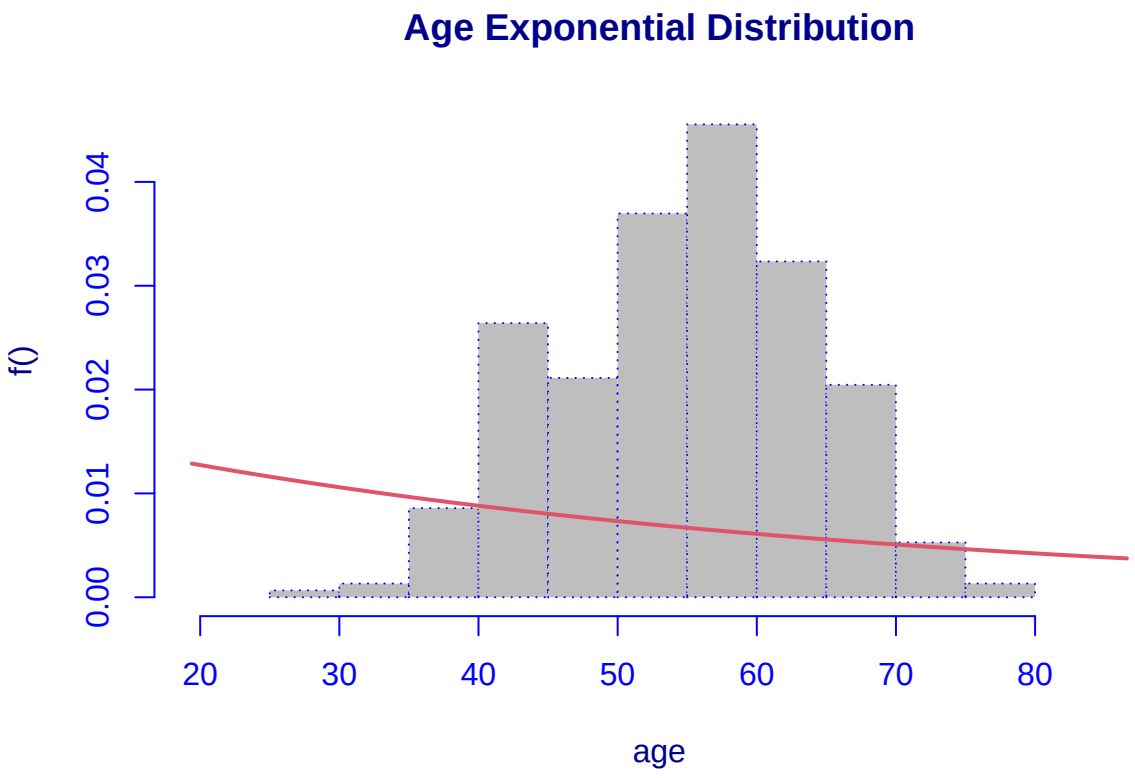



```
##
## Family:  c("GA", "Gamma")
## Fitting method: "nlminb"
##
## Call:  gamlssML(formula = heart$age, family = "GA")
##
## Mu Coefficients:
## [1]  3.996
## Sigma Coefficients:
## [1]  -1.764
##
##  Degrees of Freedom for the fit: 2 Residual Deg. of Freedom   301
## Global Deviance:     2206.4
##             AIC:     2210.4
##             SBC:     2217.82
```

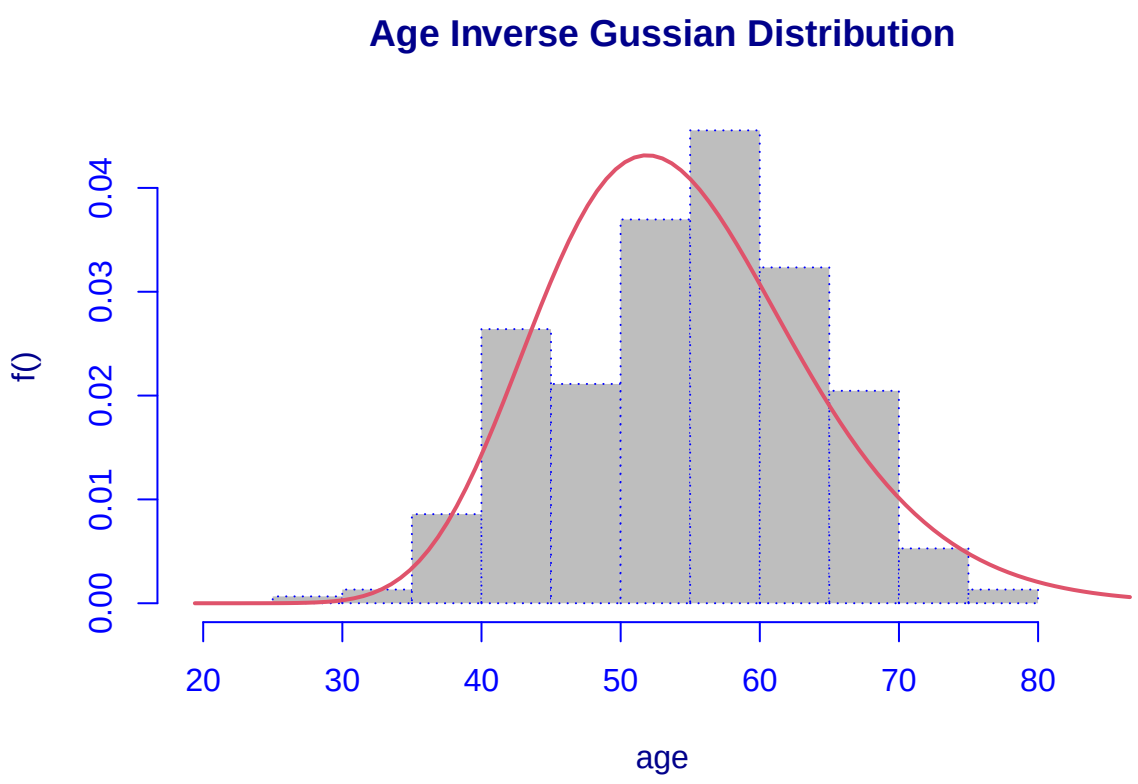
```
age.WEI<-histDist(heart$age, xlab = "age", family=WEI, nbins=13,
                  main = "Age Weibull Distribution")
```




```
age.EXP<- histDist(heart$age, xlab = "age", family=EXP, nbins=13,
                   main = "Age Exponential Distribution")
```




```
age.IG<-histDist(heart$age, xlab = "age", family=IG, nbins=13,
                 main = "Age Inverse Gussian Distribution")
```




```
age.matrix<-matrix(c(age.LOGNO$df.fit, logLik(age.LOGNO), AIC(age.LOGNO),
                     age.LOGNO$sbc, age.GA$df.fit, logLik(age.GA), AIC(age.GA),
                     age.GA$sbc,    age.WEI$df.fit, logLik(age.WEI),
                     AIC(age.WEI), age.WEI$sbc,
age.EXP$df.fit, logLik(age.EXP), AIC(age.EXP), age.EXP$sbc,
age.IG$df.fit, logLik(age.IG), AIC(age.IG), age.IG$sbc), ncol=4, byrow=TRUE)
colnames(age.matrix)<-c("df","LogLik", "AIC", "BIC")
rownames(age.matrix)<-c("LOGNO", "GA", "WEI", "EXP", "IG")
age.matrix<-as.table(age.matrix)
age.matrix
```

```
##              df    LogLik       AIC       BIC
## LOGNO     2.000 -1107.708  2219.415  2226.843
## GA        2.000 -1103.199  2210.398  2217.825
## WEI       2.000 -1096.332  2196.664  2204.091
## EXP       1.000 -1513.711  3029.422  3033.135
## IG        2.000 -1107.740  2219.480  2226.908
```

As we can see the model that has maximum value of log likelihood and less value in AIC and BIC is "Weibull distribution" so based on maximum likelihood method its safe to say that our data fits better in Weibull distribution.

Likelihood ratio test

The Likelihood-ratio test goodness of fit performed between the Exponential model (under the null hypothesis) and the Weibull model (under the alternative hypothesis).

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
lrtest(age.EXP, age.WEI)
```

```
## Likelihood ratio test
##
## Model 1: gamlssML(formula = heart$age, family = "EXP")
## Model 2: gamlssML(formula = heart$age, family = "WEI")
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   1 -1513.7
## 2   2 -1096.3  1 834.76  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis would be rejected at nearly every significance level. Thus, we know that we should definitely use the Weibull model as it increases the accuracy of our model by a substantial amount.

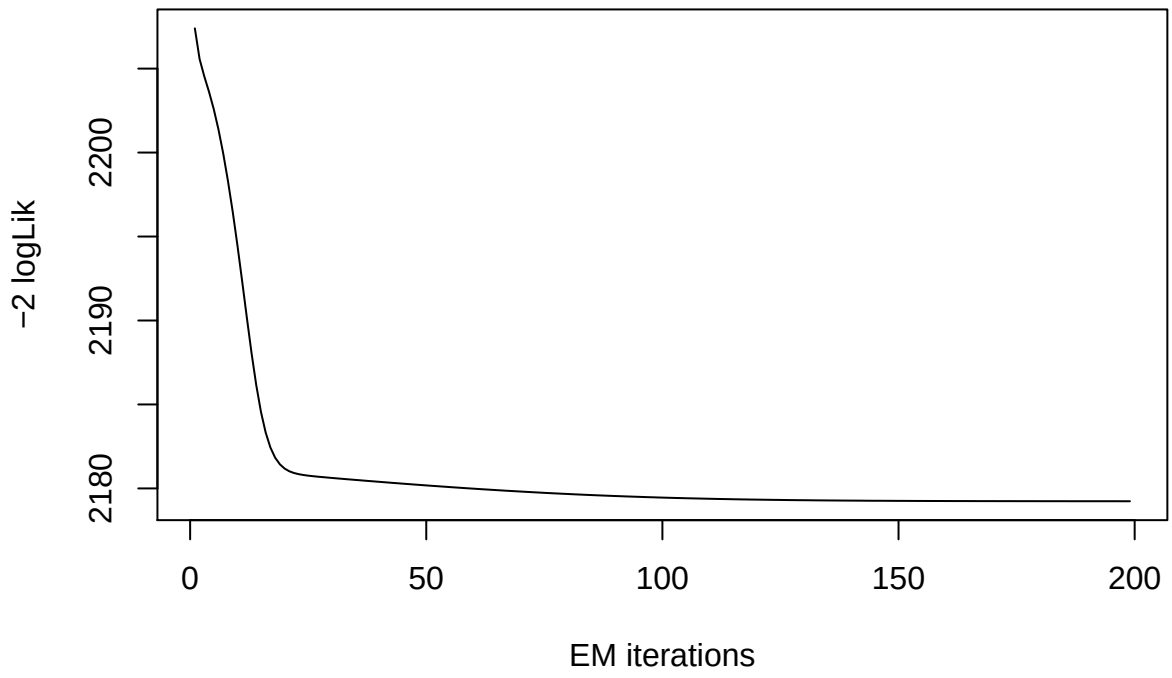
Mixture of distributions

It is possible to compute a mixture of two gamma distributions In order to find the best mixture,the algorithm is repeated five times.

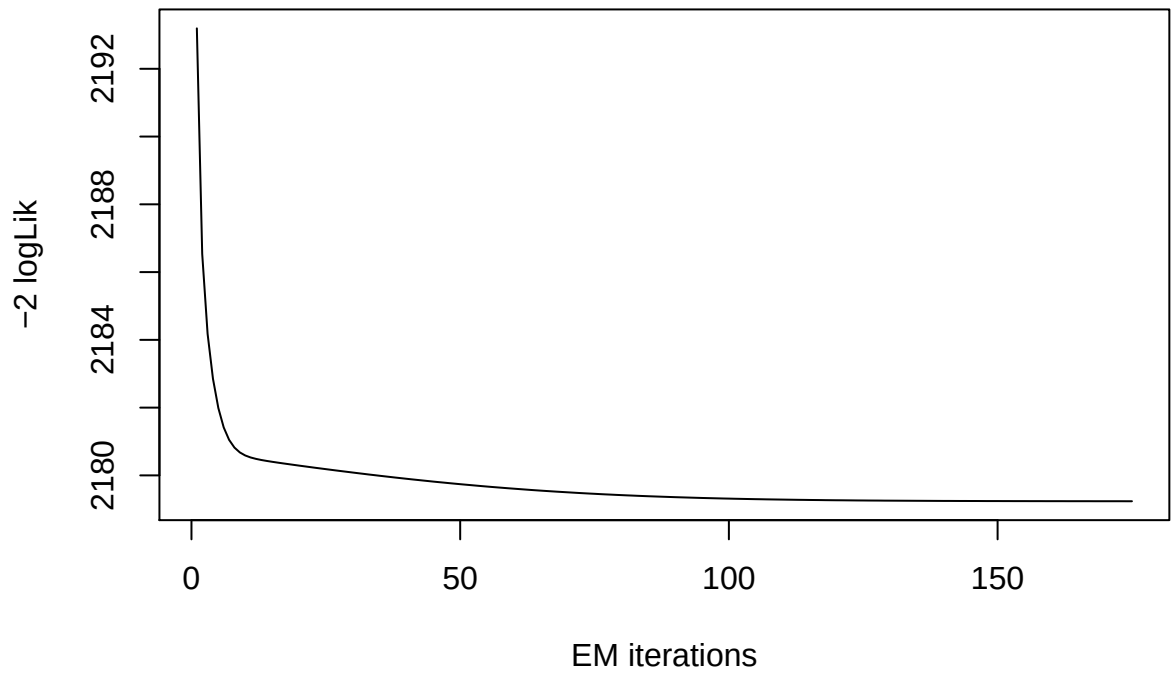
```
library(gamlss.mx)
```

```
## Loading required package: nnet
```

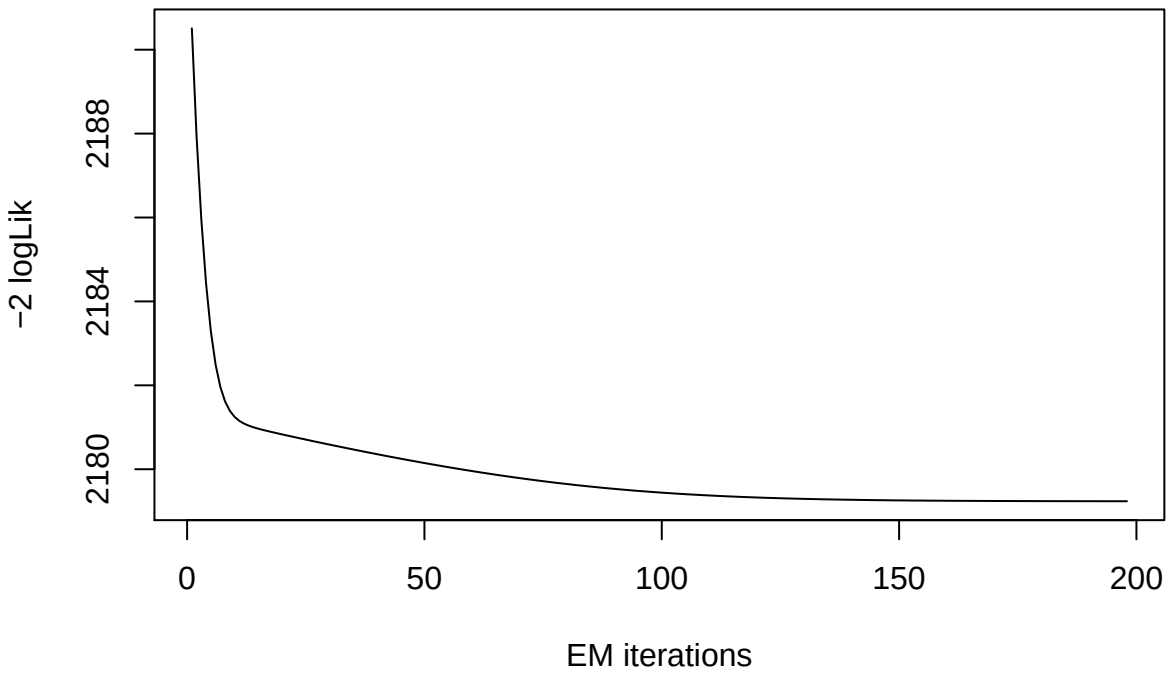
```
mxfit <- gamlssMXfits(n = 5, heart$age~1, family = GA, K = 2, data = heart)
```



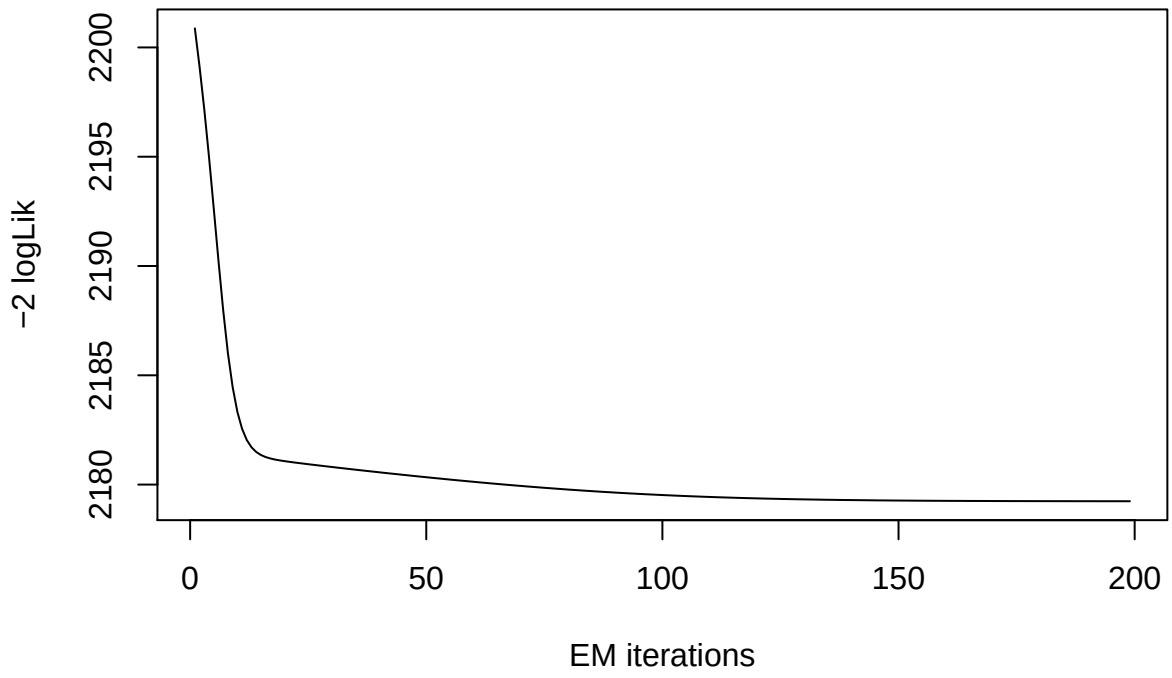

```
## model= 1
```



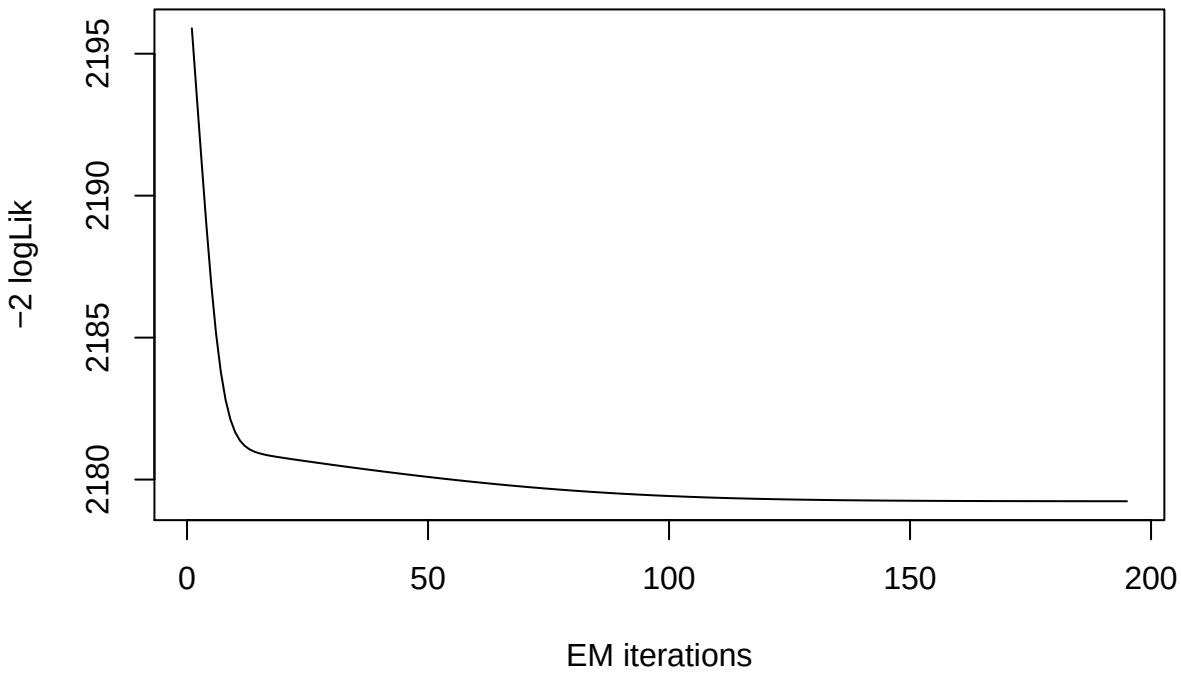

```
## model= 2
```




```
## model= 3
```


−2 logLik
2180
2185
2190
2195
2200
0
50
100
150
200
EM iterations

```
## model= 4
```




```
## model= 5
```

We can see that with the mixture of two Gamma distributions the AIC and BIC values have improved, as they are lower than those obtained with the single Gamma distribution: AIC is now equal to 2189.24, while the previous value was 2210.398, BIC is now 2207.81, which is lower than 2217.825 which was the previous value.

```
logLik(mxfit)
```

```
## 'log Lik.' -1089.618 (df=5)
```

```
mxfit$prob
```

```
## [1] 0.3290173 0.6709827
```

```
fitted(mxfit, "mu")[1]
```

```
## [1] 54.36778
```

```
fitted(mxfit, "sigma")[2]
```

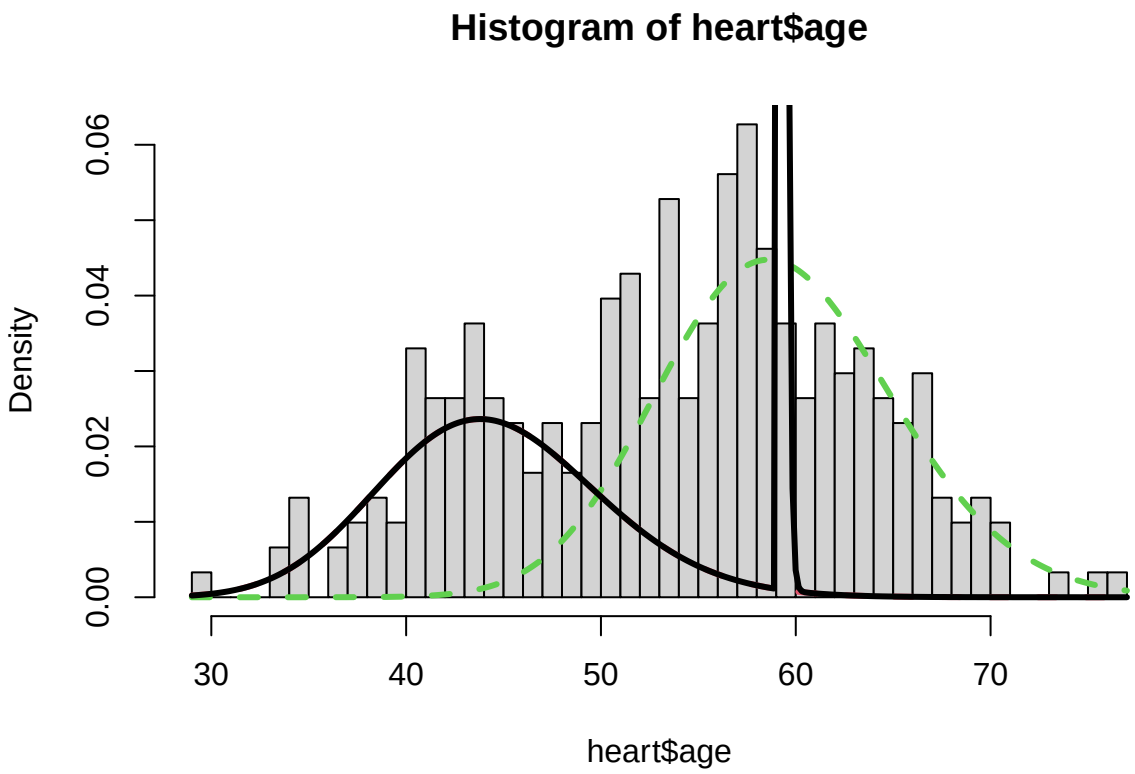
```
## [1] 54.36778
```

```
hist(heart$age, breaks = 50,freq = FALSE)
mu.hat1 <- exp(mxfit[["models"]][[1]][["mu.coefficients"]])
sigma.hat1 <- exp(mxfit[["models"]][[1]][["sigma.coefficients"]])
mu.hat2 <- exp(mxfit[["models"]][[2]][["mu.coefficients"]])
sigma.hat2 <- exp(mxfit[["models"]][[2]][["sigma.coefficients"]])
hist(heart$age, breaks = 50, freq = FALSE, xlab = "Age" ,
     main="Age-Mixture of two Lognormal distributions", plot = FALSE)
```

```
## Warning in hist.default(heart$age, breaks = 50, freq = FALSE, xlab = "Age", :
## arguments 'freq', 'main', 'xlab' are not made use of
```

```
## $breaks
##  [1] 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
## [26] 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
##
## $counts
##  [1]  1  0  0  0  2  4  0  2  3  4  3 10  8  8 11  8  7  5  7  5  7 12 13  8 16
## [26]  8 11 17 19 14 11  8 11  9 10  8  7  9  4  3  4  3  0  0  1  0  1  1
##
## $density
##  [1] 0.00330033 0.00000000 0.00000000 0.00000000 0.00660066 0.01320132
##  [7] 0.00000000 0.00660066 0.00990099 0.01320132 0.00990099 0.03300330
## [13] 0.02640264 0.02640264 0.03630363 0.02640264 0.02310231 0.01650165
## [19] 0.02310231 0.01650165 0.02310231 0.03960396 0.04290429 0.02640264
## [25] 0.05280528 0.02640264 0.03630363 0.05610561 0.06270627 0.04620462
## [31] 0.03630363 0.02640264 0.03630363 0.02970297 0.03300330 0.02640264
## [37] 0.02310231 0.02970297 0.01320132 0.00990099 0.01320132 0.00990099
## [43] 0.00000000 0.00000000 0.00330033 0.00000000 0.00330033 0.00330033
##
## $mids
##  [1] 29.5 30.5 31.5 32.5 33.5 34.5 35.5 36.5 37.5 38.5 39.5 40.5 41.5 42.5 43.5
## [16] 44.5 45.5 46.5 47.5 48.5 49.5 50.5 51.5 52.5 53.5 54.5 55.5 56.5 57.5 58.5
## [31] 59.5 60.5 61.5 62.5 63.5 64.5 65.5 66.5 67.5 68.5 69.5 70.5 71.5 72.5 73.5
## [46] 74.5 75.5 76.5
##
## $xname
## [1] "heart$age"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
lines(seq(min(heart$age),max(heart$age),length=length(heart$age)),
      mxfit[["prob"]][1]*dGA(seq(min(heart$age),max(heart$age),                    length=

lines(seq(min(heart$age),max(heart$age),length=length(heart$age)),
      mxfit[["prob"]][2]*dGA(seq(min(heart$age),max(heart$age),
length=length(heart$age)),mu=mu.hat2,sigma = sigma.hat2),lty=2,lwd=3,col=3)

lines(seq(min(heart$age),max(heart$age),length=length(heart$age)),
      mxfit[["prob"]][1]*dGA(seq(min(heart$age),max(heart$age),                    length=
        mxfit[["prob"]][2]*dRG(seq(min(heart$age),max(heart$age),
length=length(heart$trestbps)),mu= mu.hat2,sigma = sigma.hat2),
lty = 1, lwd = 3, col = 1)
```



Since we have selected K=2, we can see in the plot 2 peaks, each corresponding to a distribution.The dotted red line is relative to the first distribution, the dotted green one to the second distribution, while the black line is relative to the mixture, i.e. the overall model for all data.

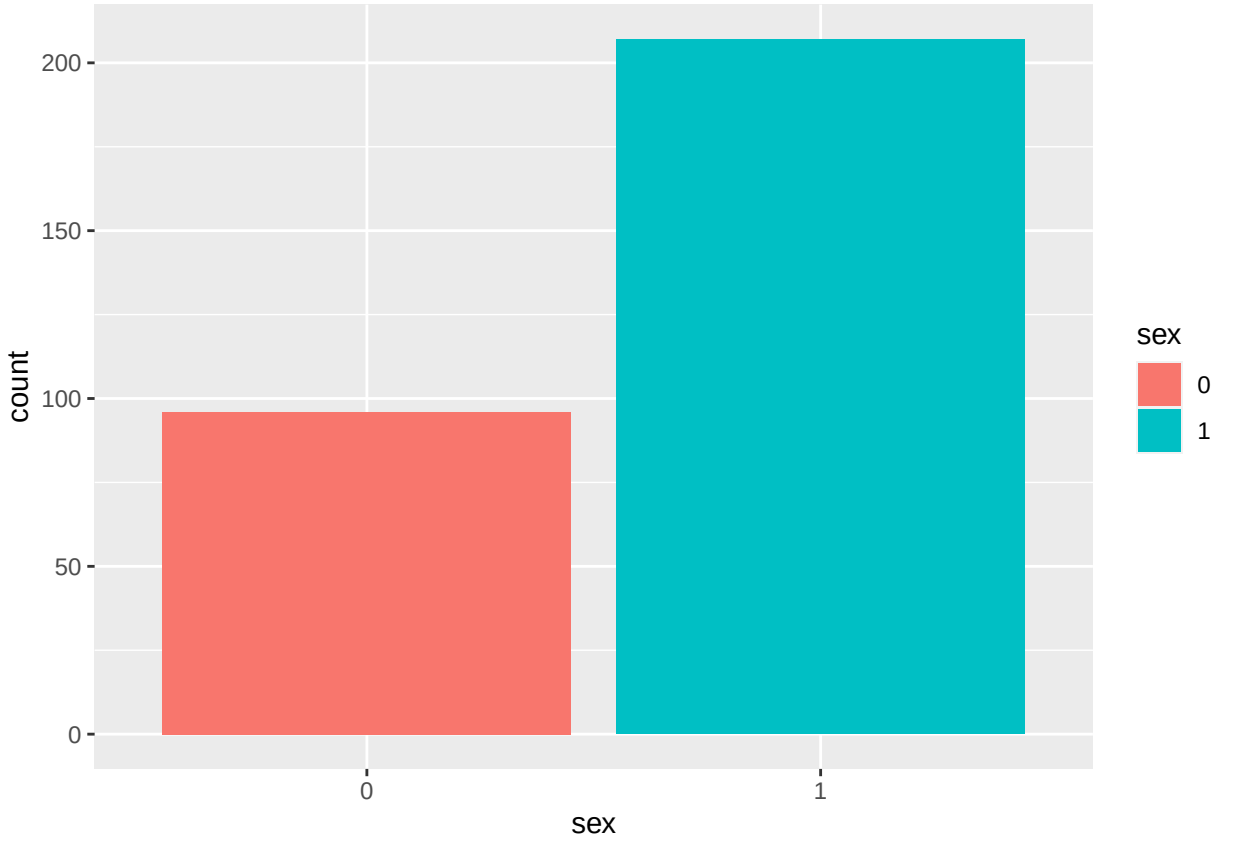
Sex

Sex is a categorical variable that can take two values 1 for male and 0 for females.

```
library(ggplot2)
summary(heart$sex)
```

```
##   0   1
##  96 207
```

```
ggplot(heart, aes(x = sex, fill=sex)) + geom_bar()
```



As the result shows our dataset contains data belonging to 96 females and 207 male.So from both plot and summary we can see that our data is not balanced and the study includes more male than females.

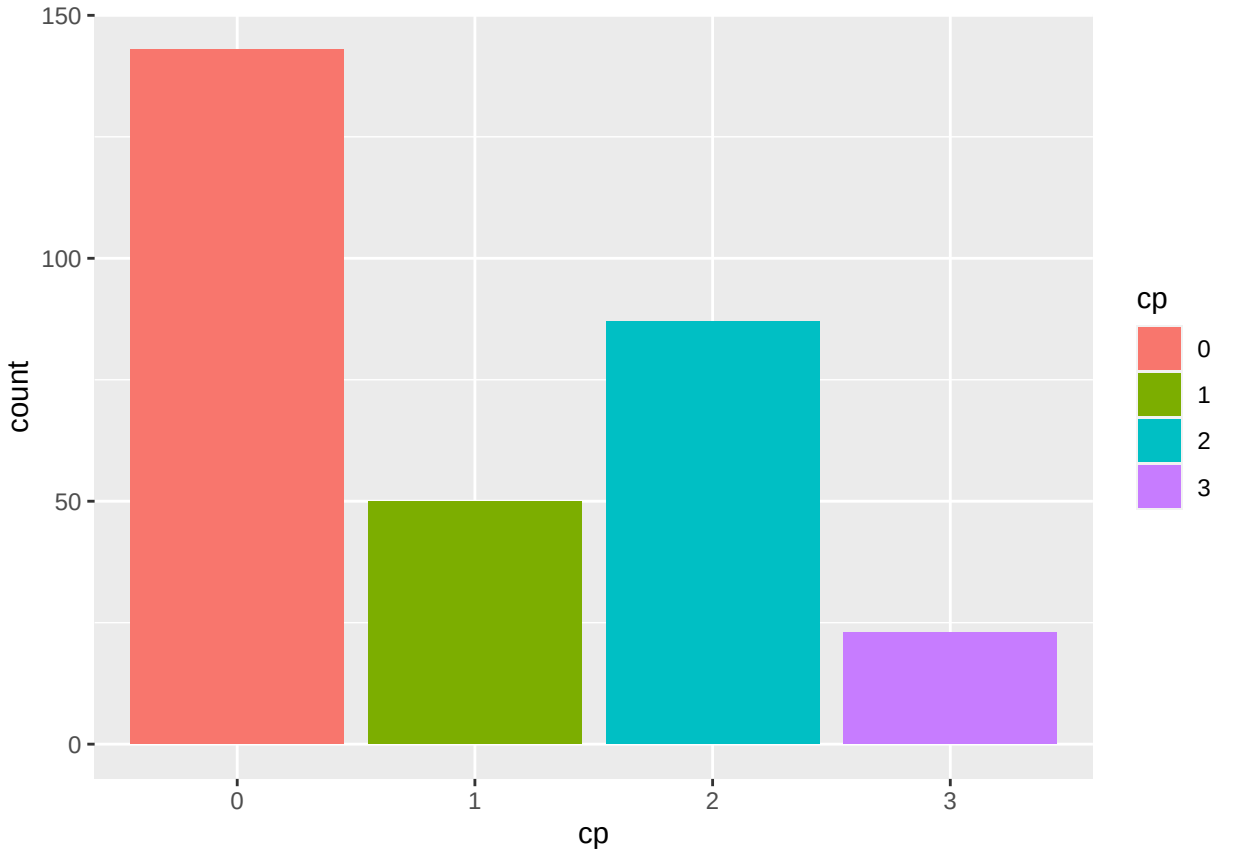
Chest Pain Type (CP)

Chest pain type (cp) is a categorical variable that takes four values from 0 to 4.

```
summary(heart$cp)
```

```
##   0   1   2   3
## 143  50  87  23
```

```
ggplot(heart, aes(x = cp, fill=cp)) + geom_bar()
```



As the result shows, the most common pain type is 0 and the least is 3.So from both plot and summary we can see that our data is not balanced.

The Resting Blood Pressure(trestbps)

The Resting Blood Pressure(trestbps) is a numerical continuous variable in rage (0,infinite).

```
length(heart$trestbps)
```

```
## [1] 303
```

```
min(heart$trestbps)
```

```
## [1] 94
```

```
max(heart$trestbps)
```

```
## [1] 200
```

The length of sample is 303, and the values are in range 94-200.

```
summary(heart$trestbps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    94.0   120.0   130.0   131.6   140.0   200.0
```

```
sd(heart$trestbps)
```

```
## [1] 17.53814
```

```r
var(heart$trestbps)
```

```
## [1] 307.5865
```

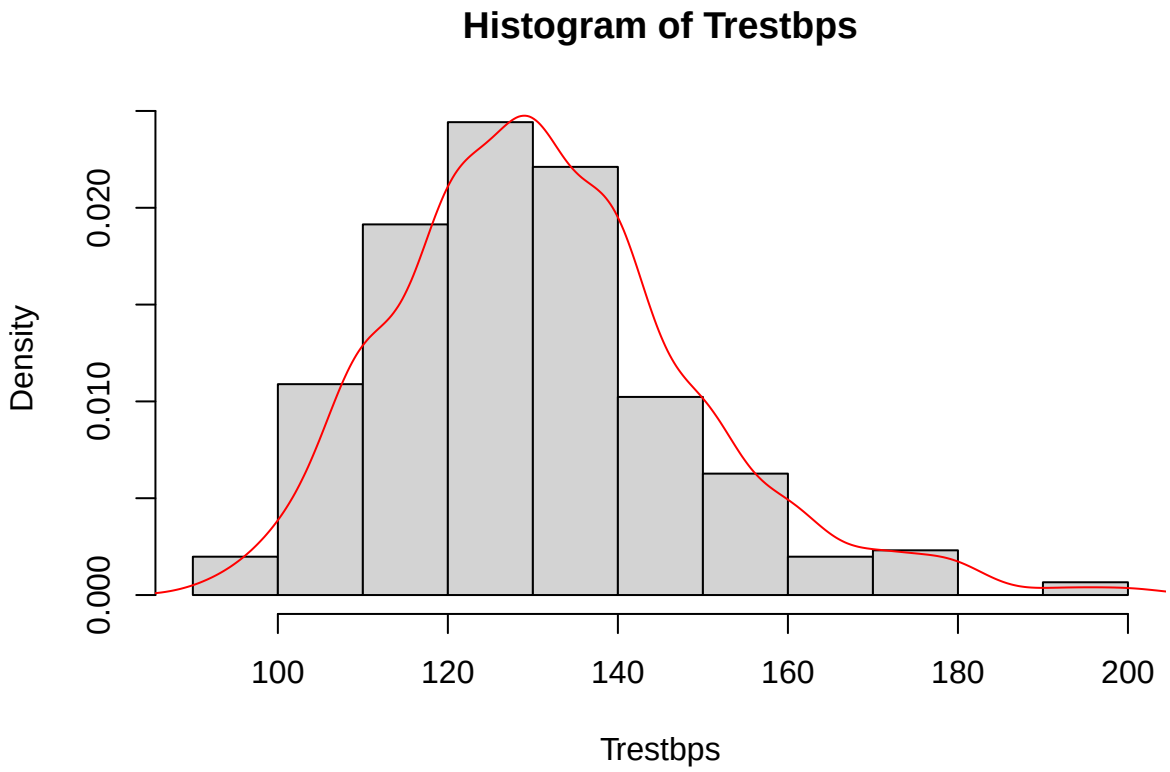
From the results we can see the Mean and Median are not equal so the distribution is asymmetrical or skewed. As we can see mean>median so the distribution should be positive and it appear that it is a unimodal distribution.

```r
skewness(heart$trestbps)
```

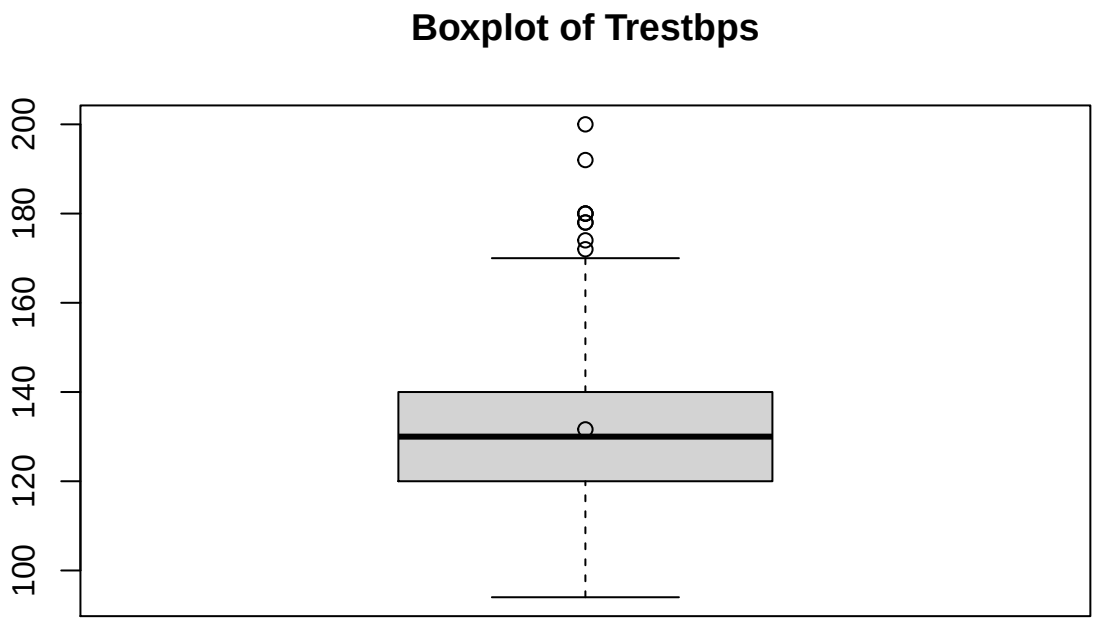
```
## [1] 0.706717
```

The skewness is positive so distribution is skewed towards the right, and many values are higher than the mean.

```r
hist(heart$trestbps, freq=FALSE, xlab= "Trestbps", main = "Histogram of Trestbps")
lines(density(heart$trestbps), col="red")
```




```r
boxplot(heart$trestbps, main = "Boxplot of Trestbps")
points(mean(heart$trestbps))
```



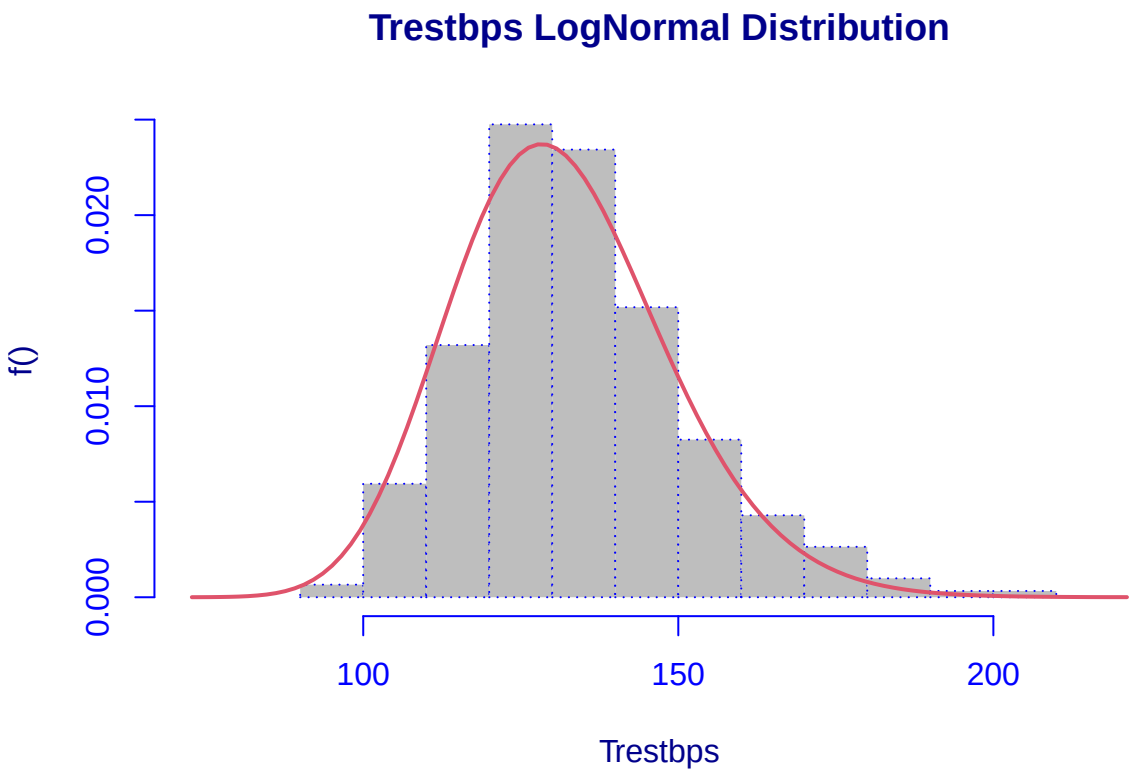
The box plot shows some noises but if we see the distribution of data below and above the mean line we can see that it's almost equal which it was predictable since mean and median have close values so we can understand the risk of heart disease is almost equal between people with different blood pressure. Even if we set a side noise from histogram the shape of histogram is close to normal distribution.

Data fitting for Trestbps

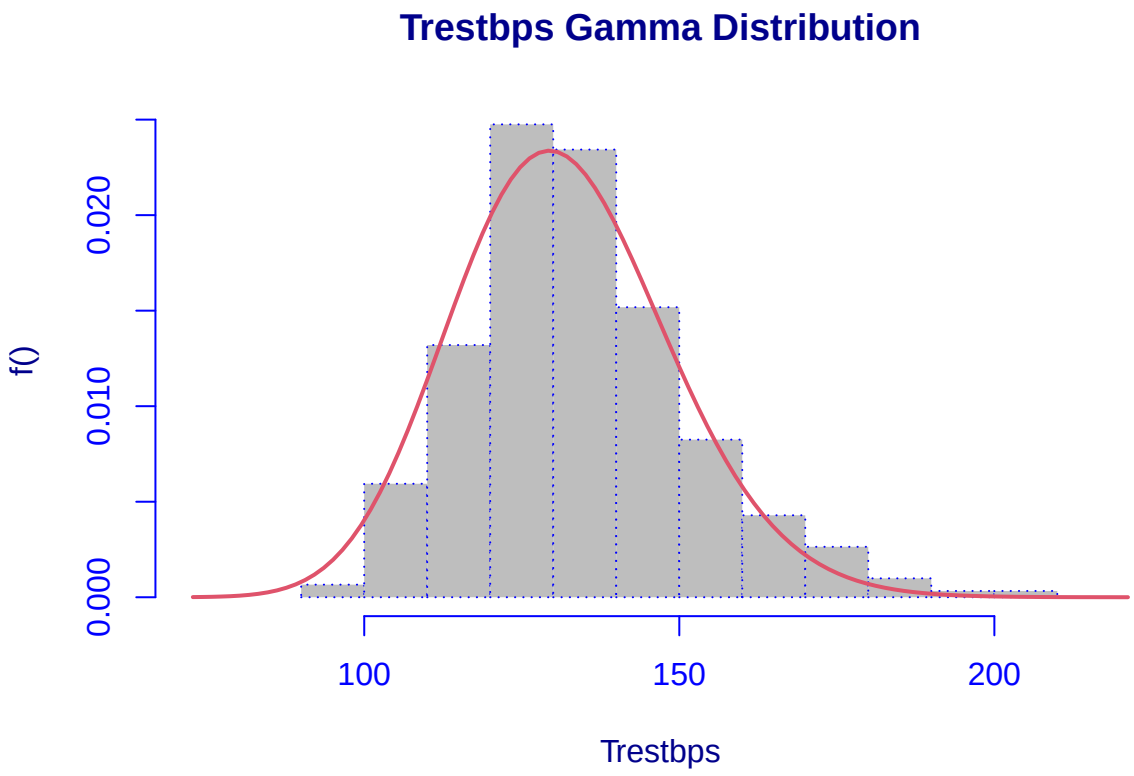
According to the domain of the variable, the distributions supported on the set of positive infinite real numbers are fitted on the data. Log-likelihood value, BIC and AIC are estimated in order to evaluate the fit of the distribution.

```
trestbps.logno<- histDist(heart$trestbps, xlab = "Trestbps", family=LOGNO,
                          nbins=13, main = "Trestbps LogNormal Distribution")
```

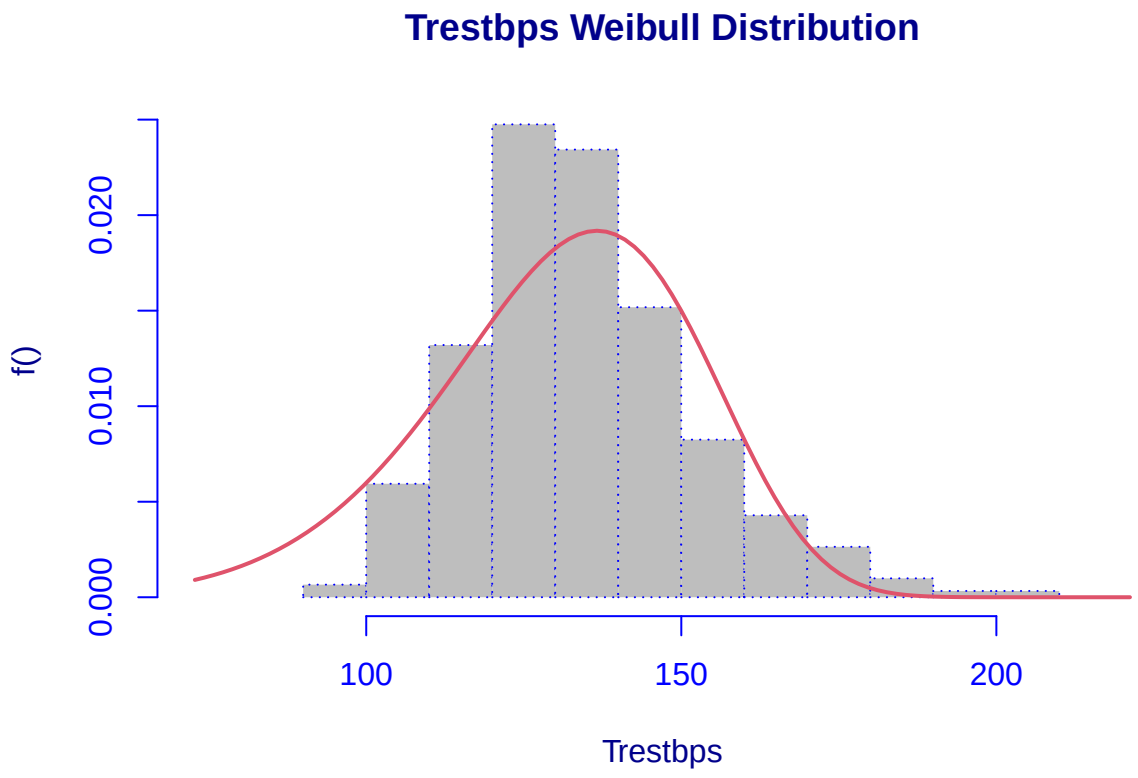
```
## Warning in MLE(ll2, start = list(eta.mu = eta.mu, eta.sigma = eta.sigma), :
## possible convergence problem: optim gave code=1 false convergence (8)
```



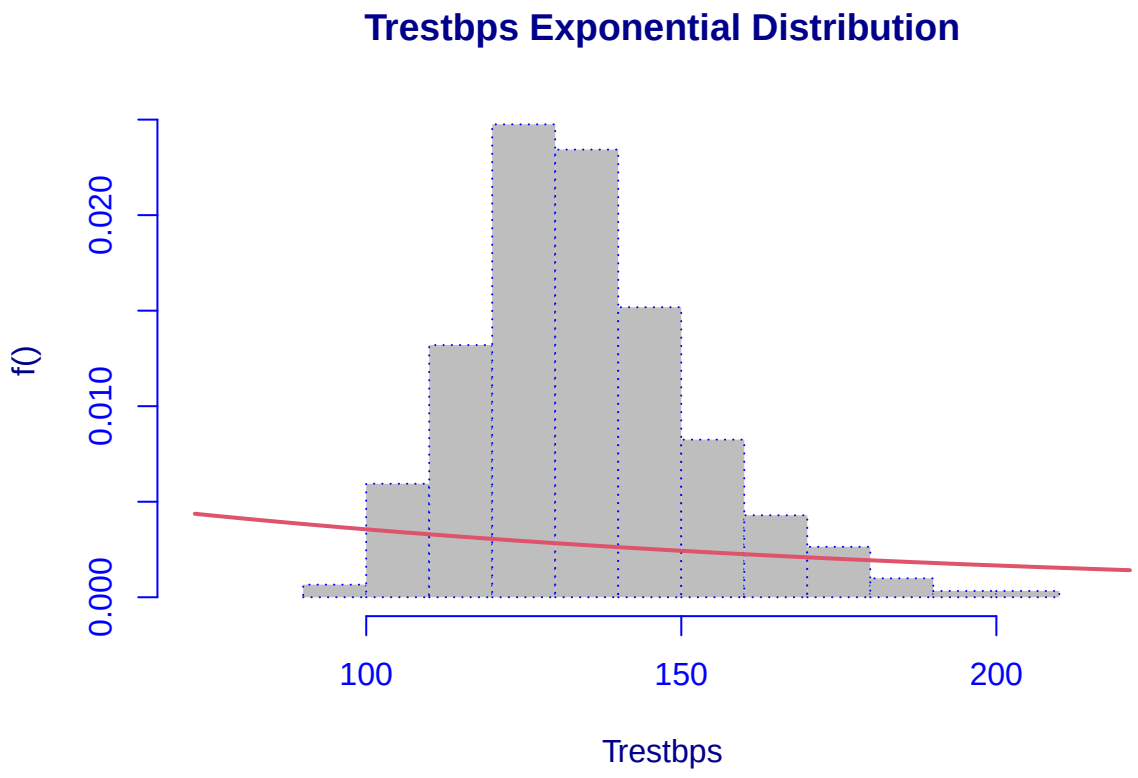

```
trestbps.ga<-histDist(heart$trestbps, xlab = "Trestbps", family=GA, nbins=13,
         main = "Trestbps Gamma Distribution")
```



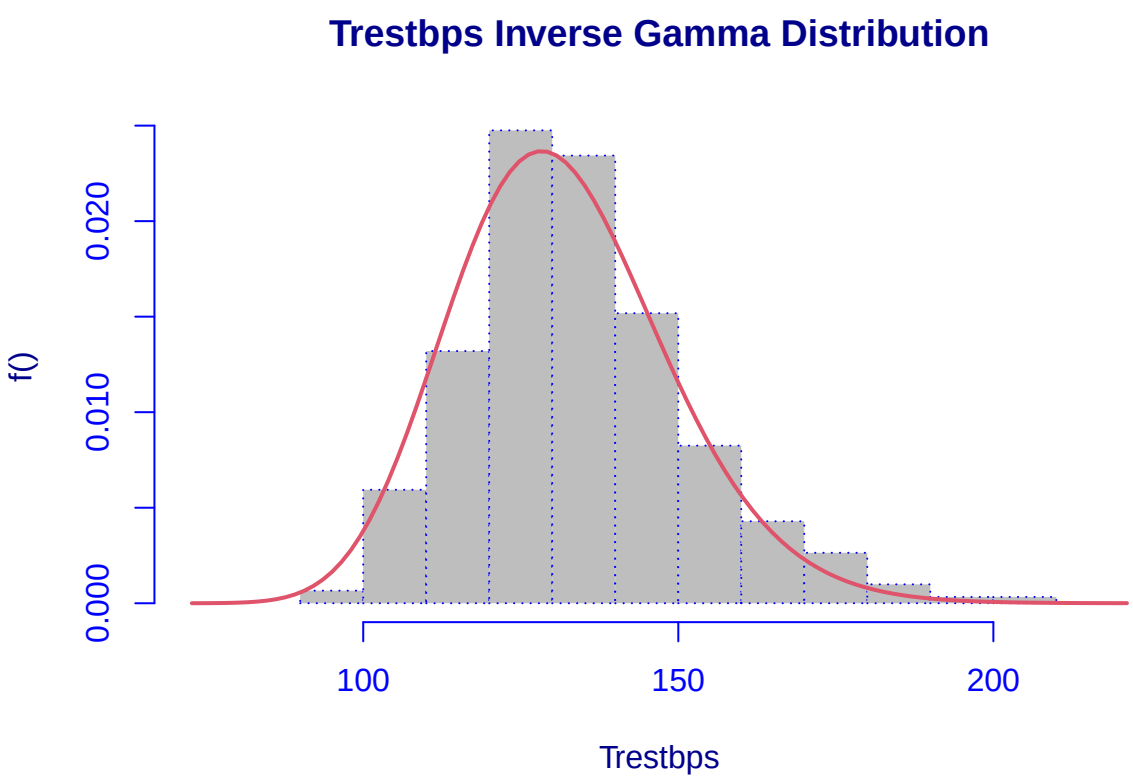

```
trestbps.wei<-histDist(heart$trestbps, xlab = "Trestbps", family=WEI, nbins=13,
          main = "Trestbps Weibull Distribution")
```




```
trestbps.exp<-histDist(heart$trestbps, xlab = "Trestbps", family=EXP, nbins=13,
          main = "Trestbps Exponential Distribution")
```




```
trestbps.ig<- histDist(heart$trestbps, xlab = "Trestbps" , family=IG, nbins=13,
          main = "Trestbps Inverse Gamma Distribution")
```




```
trestbps.matrix<-matrix(c(trestbps.logno$df.fit, logLik(trestbps.logno),
                          AIC(trestbps.logno), trestbps.logno$sbc,
                          trestbps.ga$df.fit, logLik(trestbps.ga),
                          AIC(trestbps.ga), trestbps.ga$sbc,
                          trestbps.wei$df.fit, logLik(trestbps.wei),
                          AIC(trestbps.wei), trestbps.wei$sbc,
                          trestbps.exp$df.fit, logLik(trestbps.exp),
                          AIC(trestbps.exp), trestbps.exp$sbc,
                          trestbps.ig$df.fit, logLik(trestbps.ig),
                          AIC(trestbps.ig), trestbps.ig$sbc), ncol=4,
                        byrow=TRUE)
colnames(trestbps.matrix)<-c("df","LogLik", "AIC", "BIC")
rownames(trestbps.matrix)<-c("EXP", "GA", "LOGNO", "IG", "WEI")
trestbps.matrix<-as.table(trestbps.matrix)
trestbps.matrix
```

```
##              df    LogLik       AIC       BIC
## EXP       2.000 -1287.728  2579.456  2586.883
## GA        2.000 -1290.029  2584.058  2591.486
## LOGNO     2.000 -1326.261  2656.523  2663.950
## IG        1.000 -1781.624  3565.248  3568.962
## WEI       2.000 -1287.756  2579.512  2586.939
```

As we can see the model that has maximum value of log likelihood and less value in AIC and BIC is "Exponential distribution" so based on maximum likelihood method its safe to say that our data fits better in Exponential distribution.

Likelihood ratio test

The Likelihood-ratio test goodness of fit performed between the Inverse Gamma model (under the null hypothesis) and the Exponential model (under the alternative hypothesis).

```
lrtest(trestbps.ig, trestbps.exp)
```

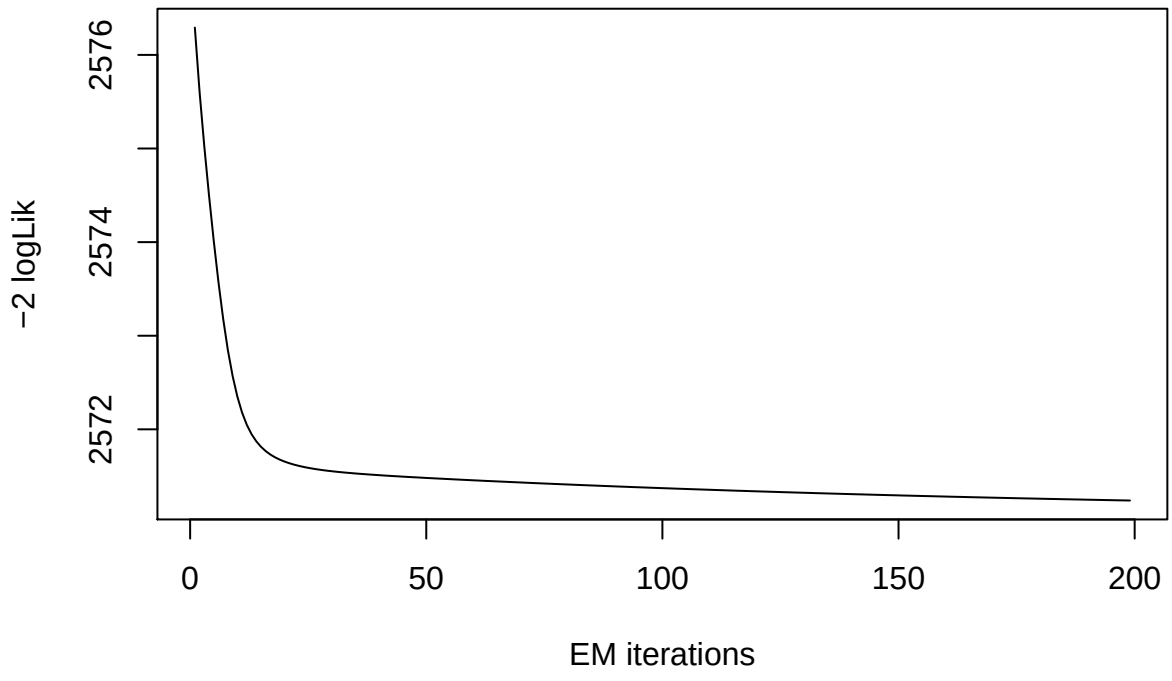
```
## Likelihood ratio test
##
## Model 1: gamlssML(formula = heart$trestbps, family = "IG")
## Model 2: gamlssML(formula = heart$trestbps, family = "EXP")
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   2 -1287.8
## 2   1 -1781.6 -1 987.74  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis would be rejected at nearly every significance level. Thus, we know that we should definitely use the Exponential model as it increases the accuracy of our model by a substantial amount.

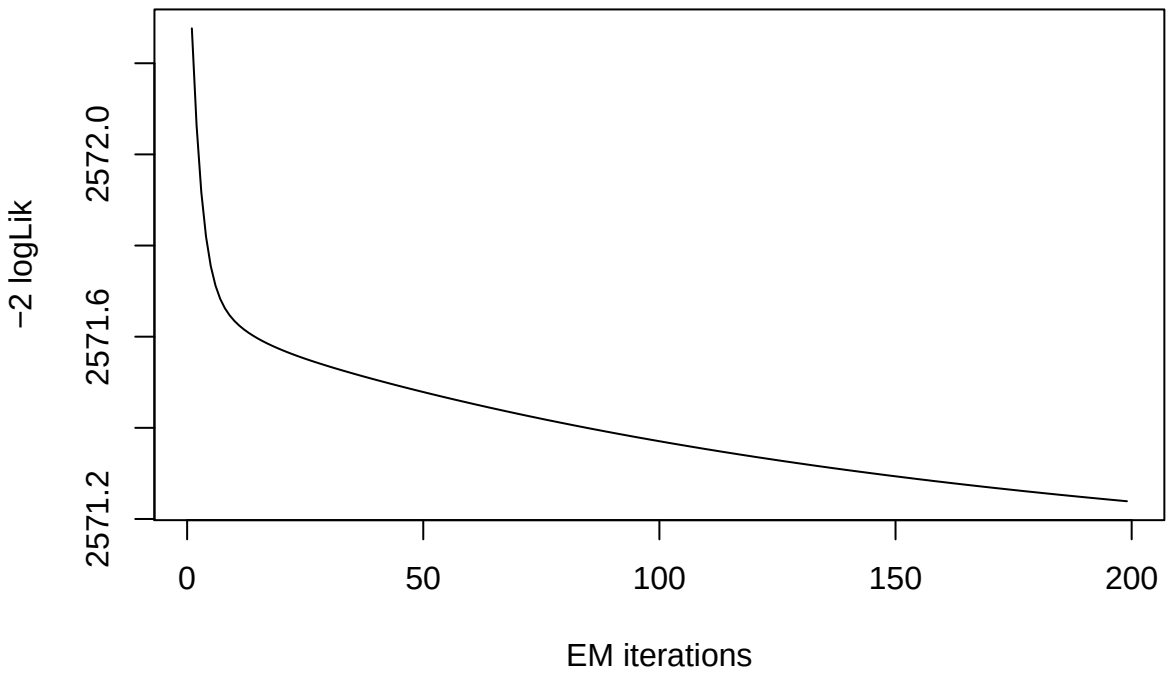
Mixture of distributions

It is possible to compute a mixture of two lognormal distributions In order to find the best mixture,the algorithm is repeated five times.

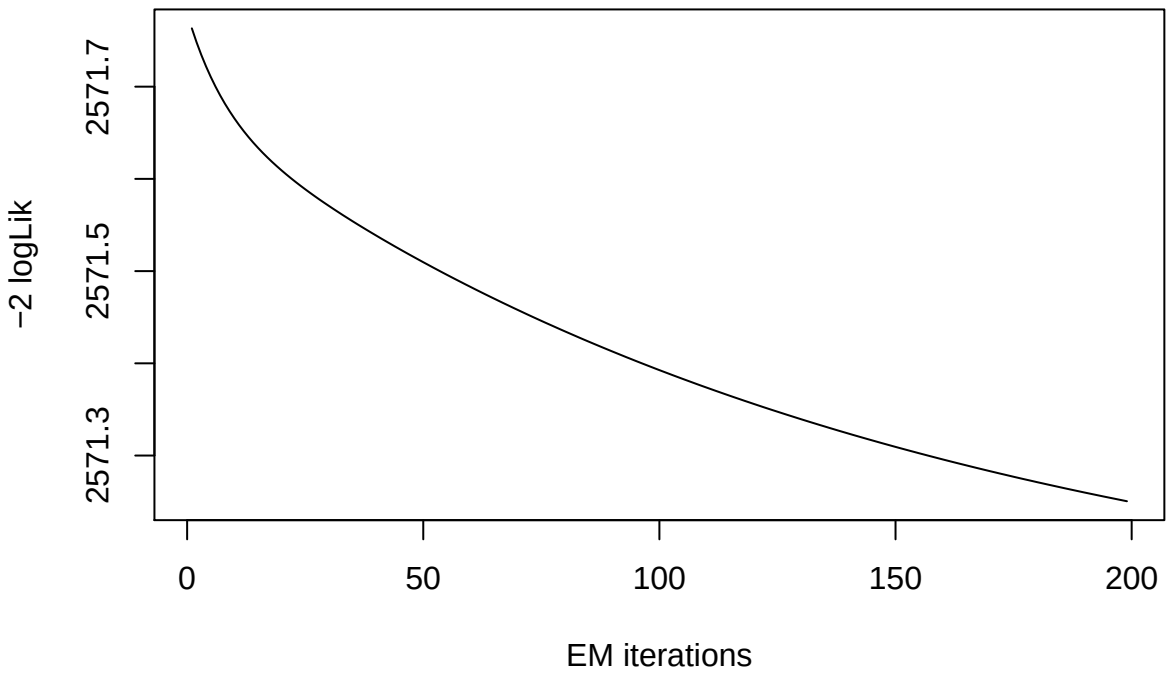
```
mxfit <- gamlssMXfits(n = 5, heart$trestbps~1, family = LOGNO, K = 2,
                      data = heart)
```



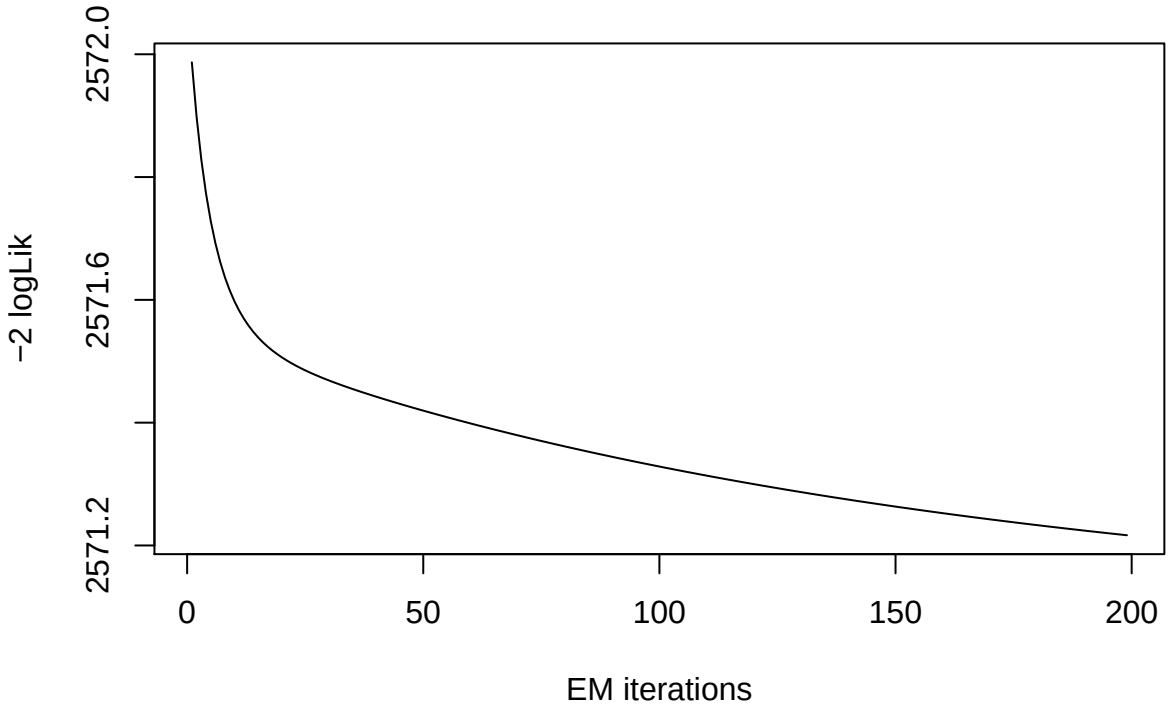

```
## model= 1
```



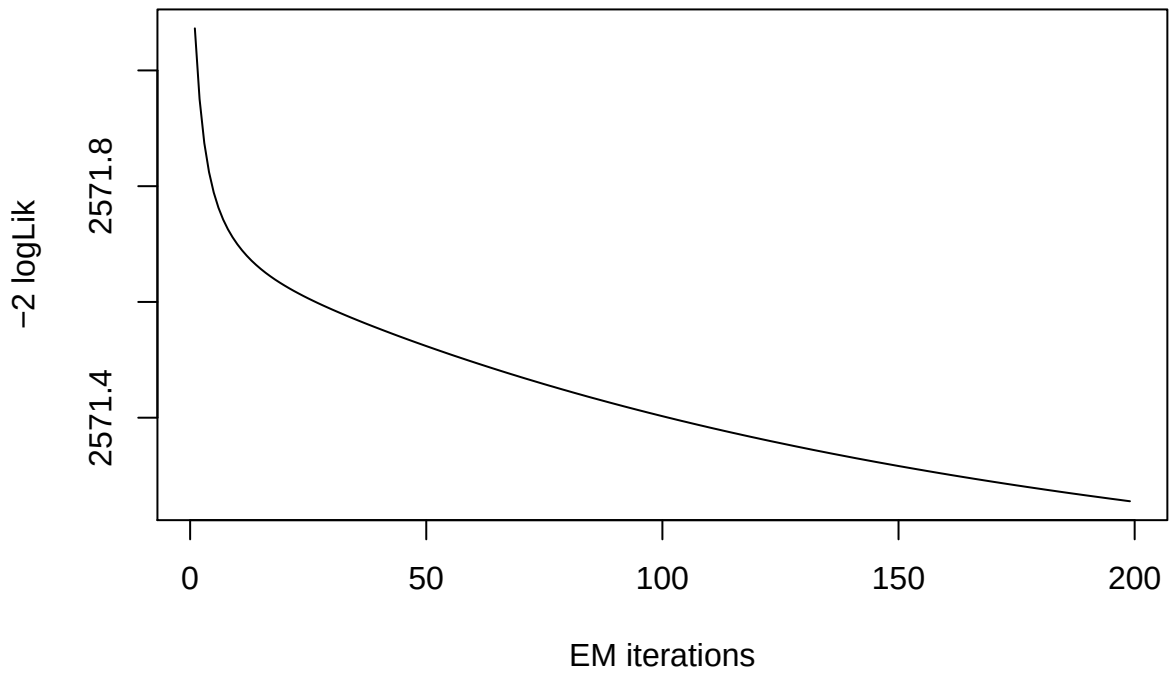

```
## model= 2
```




```
## model= 3
```



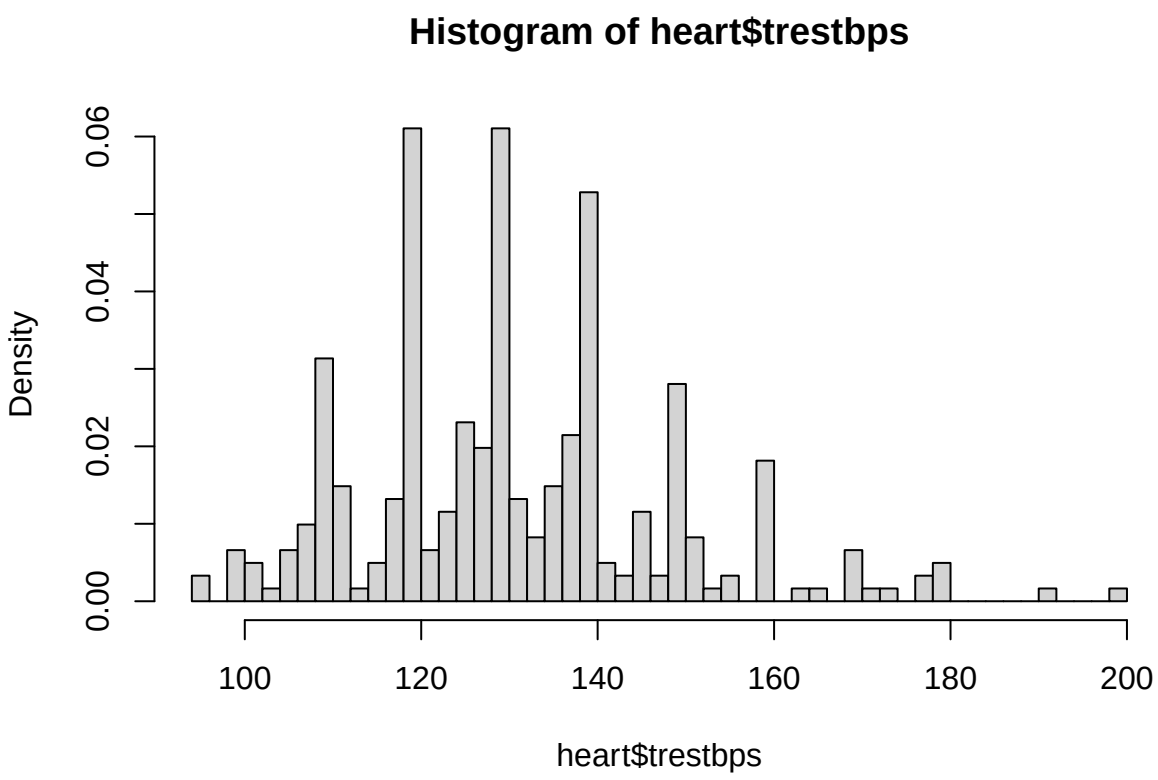

```
## model= 4
```



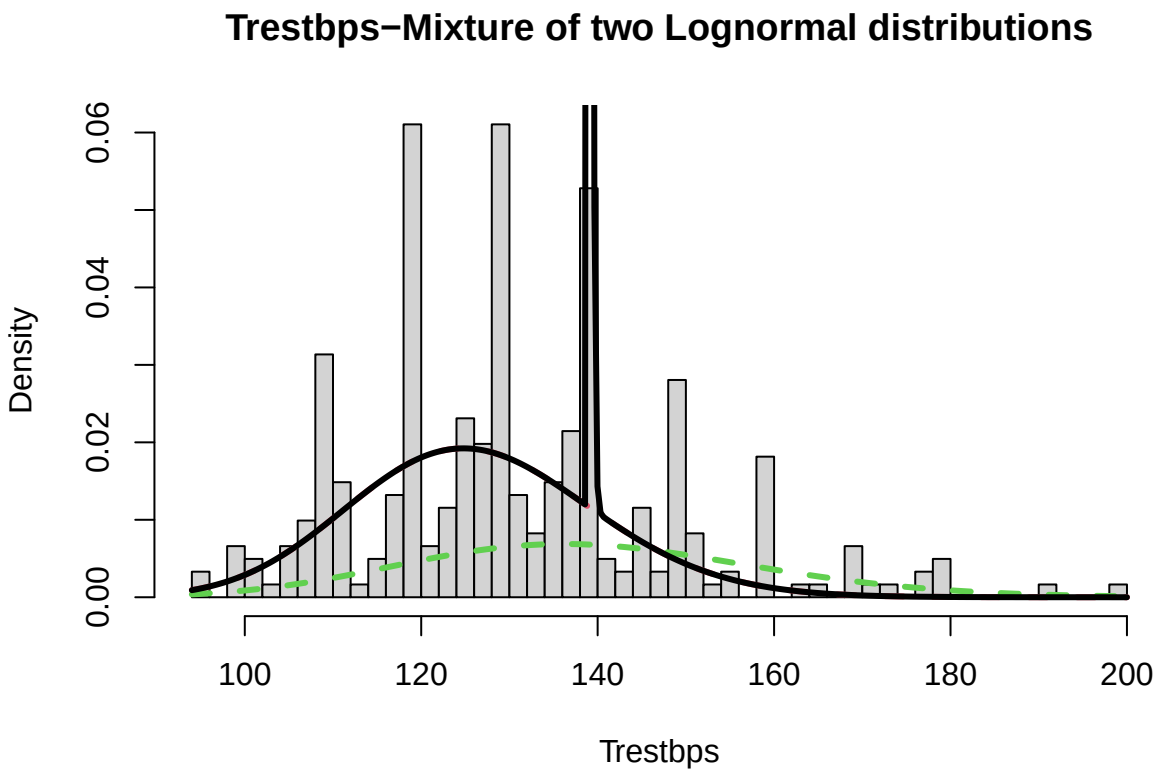

```
## model= 5
```

We can see that with the mixture of two Gamma distributions the AIC and BIC values have improved, as they are lower than those obtained with the single Gamma distribution: AIC is now equal to 2581.22, while the previous value was 2656.523, BIC is now 2599.79, which is lower than 2663.950 which was the previous value.

```
hist(heart$trestbps, breaks = 50,freq = FALSE)
```




```
mu.hat1 <- exp(mxfit[["models"]][[1]][["mu.coefficients"]])
sigma.hat1 <- exp(mxfit[["models"]][[1]][["sigma.coefficients"]])
mu.hat2 <- exp(mxfit[["models"]][[2]][["mu.coefficients"]])
sigma.hat2 <- exp(mxfit[["models"]][[2]][["sigma.coefficients"]])
hist(heart$trestbps, breaks = 50, freq = FALSE, xlab = "Trestbps" ,
     main="Trestbps-Mixture of two Lognormal distributions")
lines(seq(min(heart$trestbps),max(heart$trestbps),
          length=length(heart$trestbps)),
      mxfit[["prob"]][1]*dGA(seq(min(heart$trestbps),max(heart$trestbps),
         length=length(heart$trestbps)),mu=mu.hat1,sigma=sigma.hat1),
      lty=2,lwd=3,col=2)
lines(seq(min(heart$trestbps),max(heart$trestbps),
          length=length(heart$trestbps)),
      mxfit[["prob"]][2]*dGA(seq(min(heart$trestbps),max(heart$trestbps),
length=length(heart$trestbps)),mu=mu.hat2,sigma = sigma.hat2),lty=2,lwd=3,col=3)
lines(seq(min(heart$trestbps),max(heart$trestbps),
          length=length(heart$trestbps)),
      mxfit[["prob"]][1]*dGA(seq(min(heart$trestbps),max(heart$trestbps),
mu= mu.hat2,sigma = sigma.hat2), lty = 1, lwd = 3, col = 1)
```



Since we have selected K=2, we can see in the plot 2 peaks, each corresponding to a distribution.The dotted red line is relative to the first distribution, the dotted green one to the second distribution, while the black line is relative to the mixture, i.e. the overall model for all data.

Serum cholesterol in mg/dl(chol)

Serum cholesterol in mg/dl(chol) is a numerical continuous variable in rage (0,infinite).

```
length(heart$chol)
```

```
## [1] 303
```

```
min(heart$chol)
```

```
## [1] 126
```

```
max(heart$chol)
```

```
## [1] 564
```

The length of sample is 303, and the values are in range 126-564.

```
summary(heart$chol)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   126.0   211.0   240.0   246.3   274.5   564.0
```

```
sd(heart$chol)
```

```
## [1] 51.83075
```

```r
var(heart$chol)
```

```
## [1] 2686.427
```

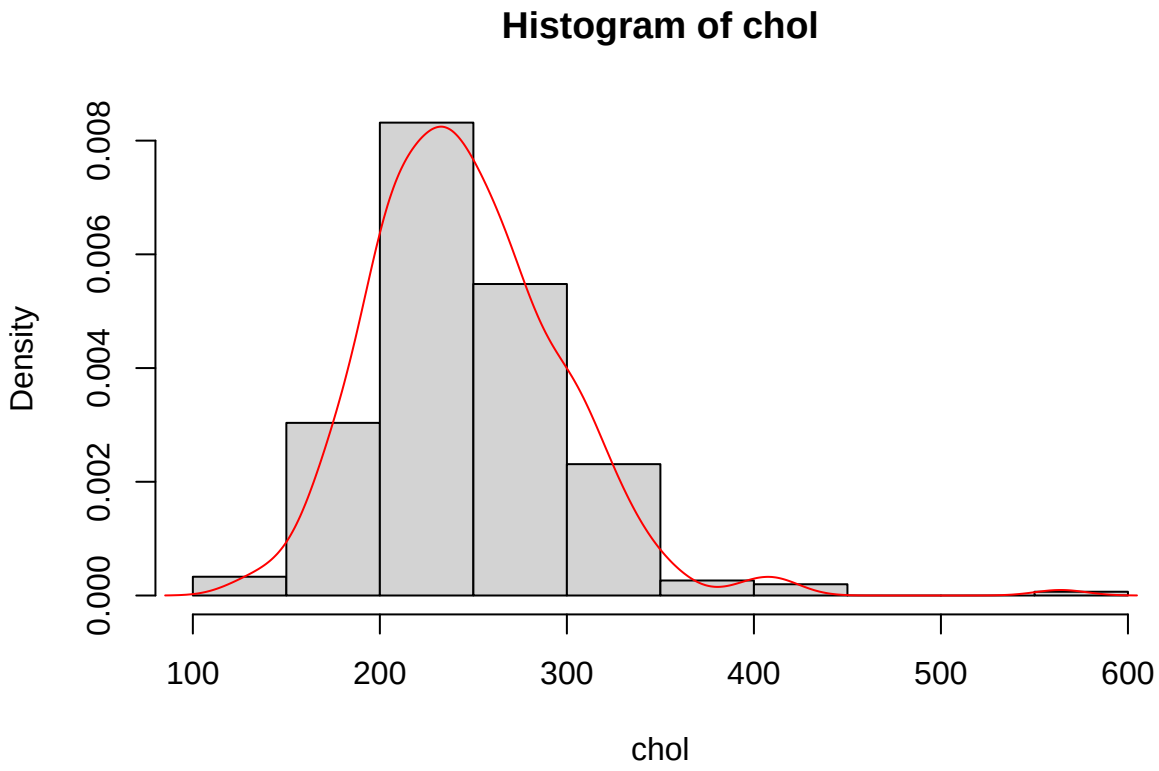
From the results we can see the Mean and Median are not equal so the distribution is asymmetrical or skewed. As we can see mean>median so the distribution should be positive and from histogram it appears to be a unimodal distribution.

```r
skewness(heart$chol)
```

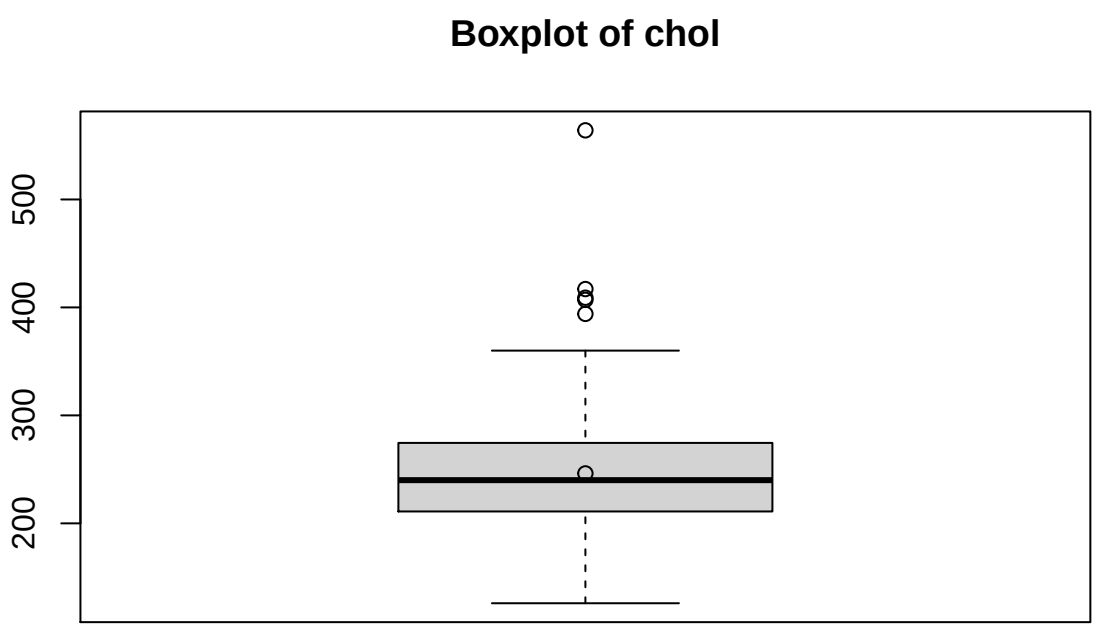
```
## [1] 1.132105
```

The skewness is positive so distribution is skewed towards the right, and many alues are higher than the mean.

```r
hist(heart$chol, freq=FALSE, xlab= "chol", main = "Histogram of chol")
lines(density(heart$chol), col="red")
```




```r
boxplot(heart$chol, main = "Boxplot of chol")
points(mean(heart$chol))
```

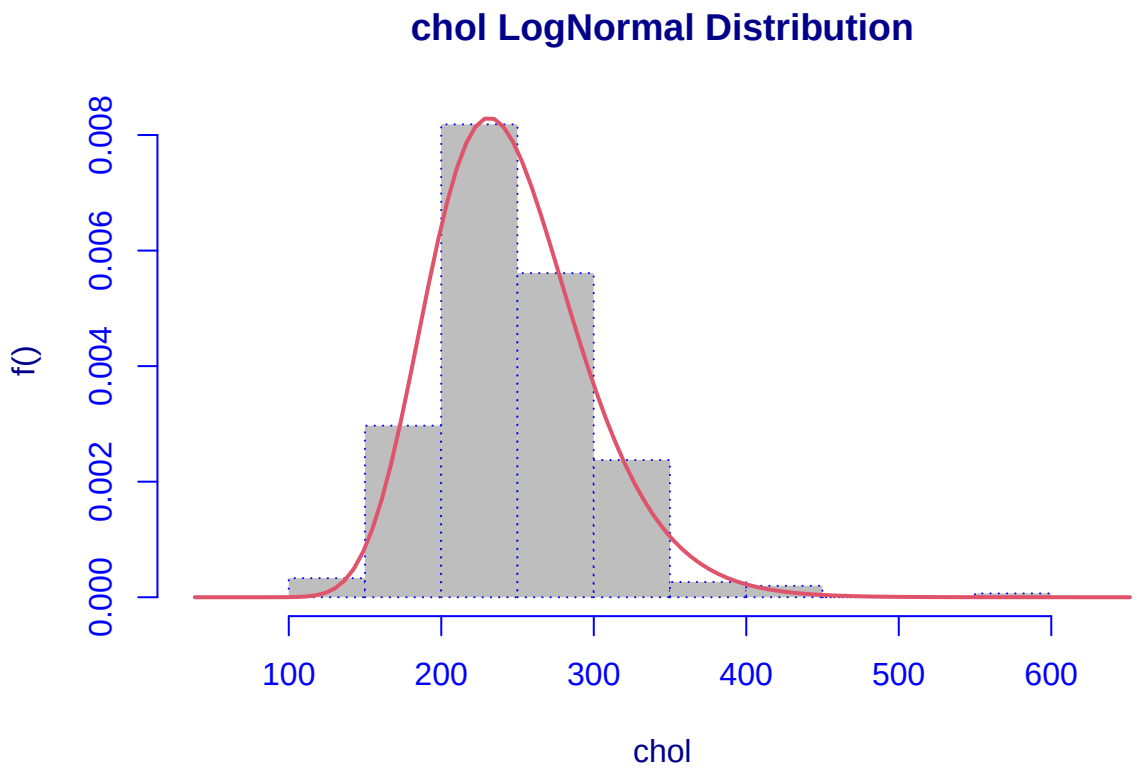


The box plot shows some noises but if we see the distribution of data below and above the mean line we can see that it's almost equal which is predictable since mean and median have close values so we can understand the risk of heart disease is almost equal between people with different cholesterol levels. Even if we set a side noise from histogram the shape of histogram is close to normal distribution.

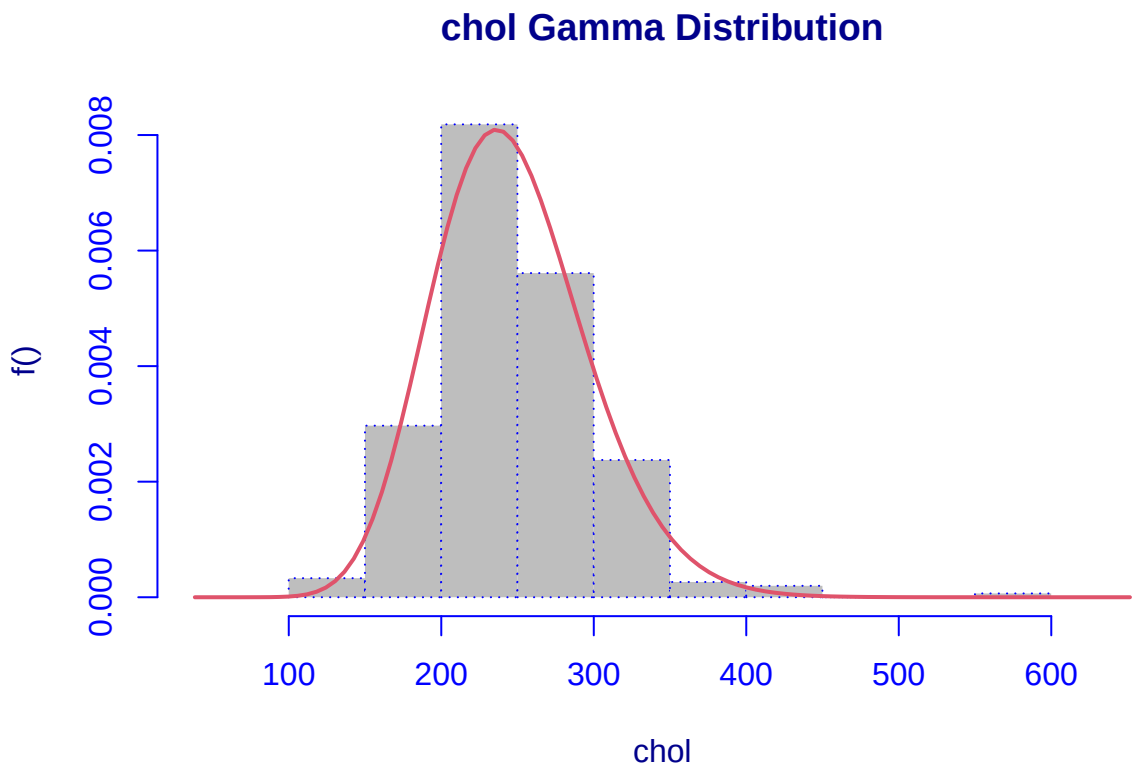
Data fitting for Chol

According to the domain of the variable, the distributions supported on the set of positive infinite real numbers are fitted on the data. Log-likelihood value, BIC and AIC are estimated in order to evaluate the fit of the distribution.

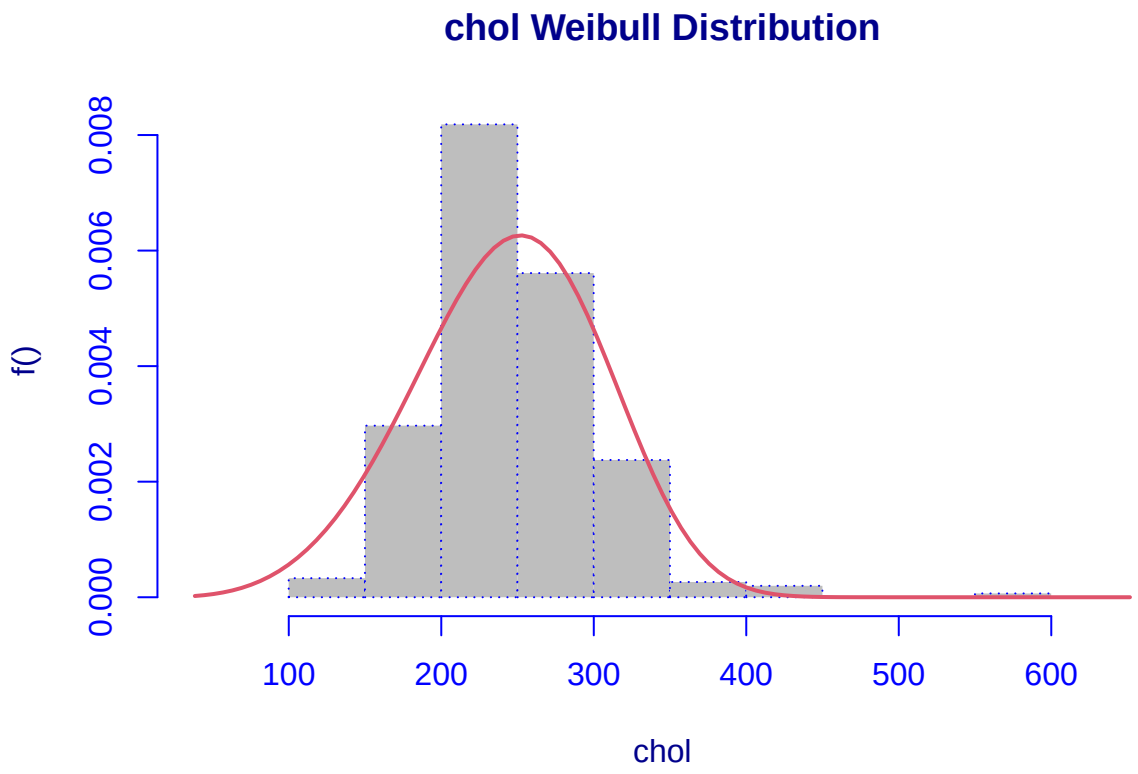
```
chol.logno<-histDist(heart$chol, xlab = "chol" , family=LOGNO, nbins=13,
                     main = "chol LogNormal Distribution")
```



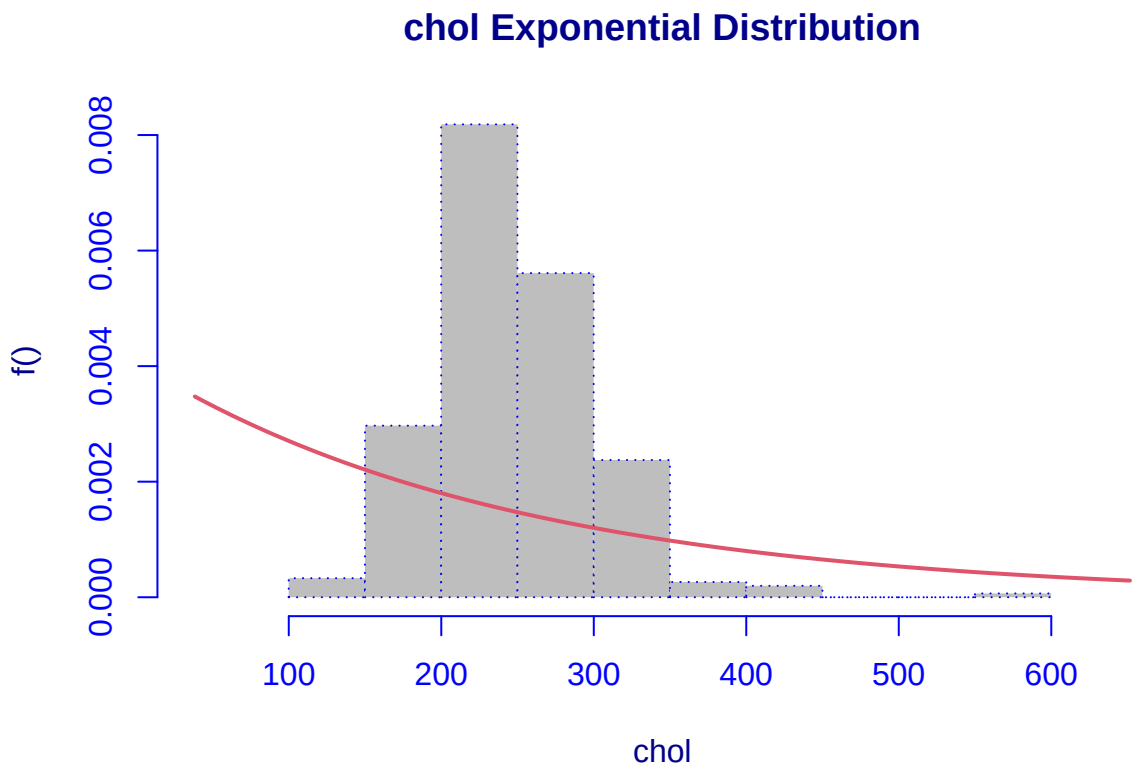

```
chol.ga<-histDist(heart$chol, xlab = "chol" , family=GA, nbins=13,
                  main = "chol Gamma Distribution")
```



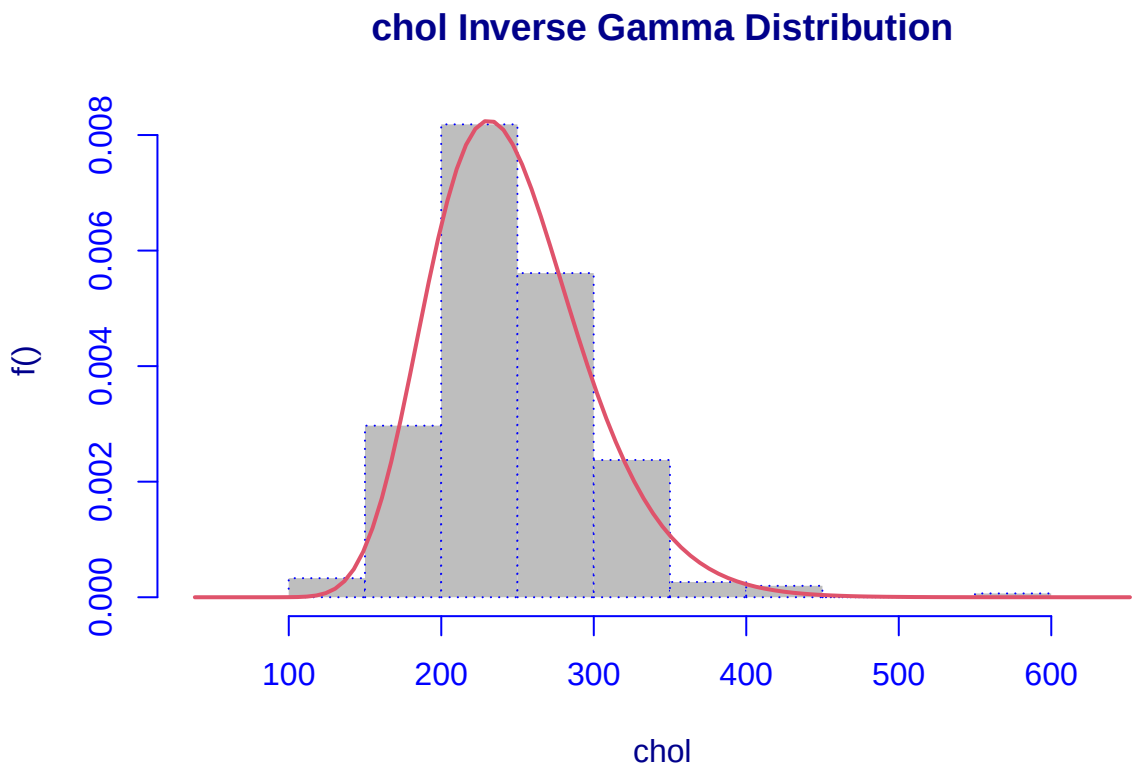

```
chol.wei<-histDist(heart$chol, xlab = "chol" ,family=WEI, nbins=13,
                   main = "chol Weibull Distribution")
```




```
chol.exp <- histDist(heart$chol, xlab = "chol" , family=EXP, nbins=13,
                     main = "chol Exponential Distribution")
```




```
chol.ig <- histDist(heart$chol, xlab = "chol" , family=IG, nbins=13,
                    main = "chol Inverse Gamma Distribution")
```




```
chol.matrix<-matrix(c(chol.logno$df.fit, logLik(chol.logno), AIC(chol.logno),
                      chol.logno$sbc, chol.ga$df.fit, logLik(chol.ga),
                      AIC(chol.ga), chol.ga$sbc, chol.wei$df.fit,
                      logLik(chol.wei), AIC(chol.wei), chol.wei$sbc,
                      chol.exp$df.fit, logLik(chol.exp), AIC(chol.exp),
                      chol.exp$sbc, chol.ig$df.fit, logLik(chol.ig),
                      AIC(chol.ig), chol.ig$sbc), ncol=4, byrow=TRUE
)
colnames(chol.matrix)<-c("df","LogLik", "AIC", "BIC")
rownames(chol.matrix)<-c("EXP", "GA", "LOGNO", "IG", "WEI")
chol.matrix<-as.table(chol.matrix)
chol.matrix
```

```
##              df    LogLik       AIC       BIC
## EXP       2.000 -1609.604  3223.207  3230.635
## GA        2.000 -1612.042  3228.084  3235.511
## LOGNO     2.000 -1651.485  3306.970  3314.398
## IG        1.000 -1971.440  3944.881  3948.595
## WEI       2.000 -1610.063  3224.126  3231.554
```

As we can see the model that has maximum value of log likelihood and less value in AIC and BIC is "Exponential distribution" so based on maximum likelihood method its safe to say that our data fits better in Exponential distribution.

Likelihood ratio test

The Likelihood-ratio test goodness of fit performed between the Inverse Gamma model (under the null hypothesis) and the Exponential model (under the alternative hypothesis).

```
lrtest(chol.ig, chol.exp)
```

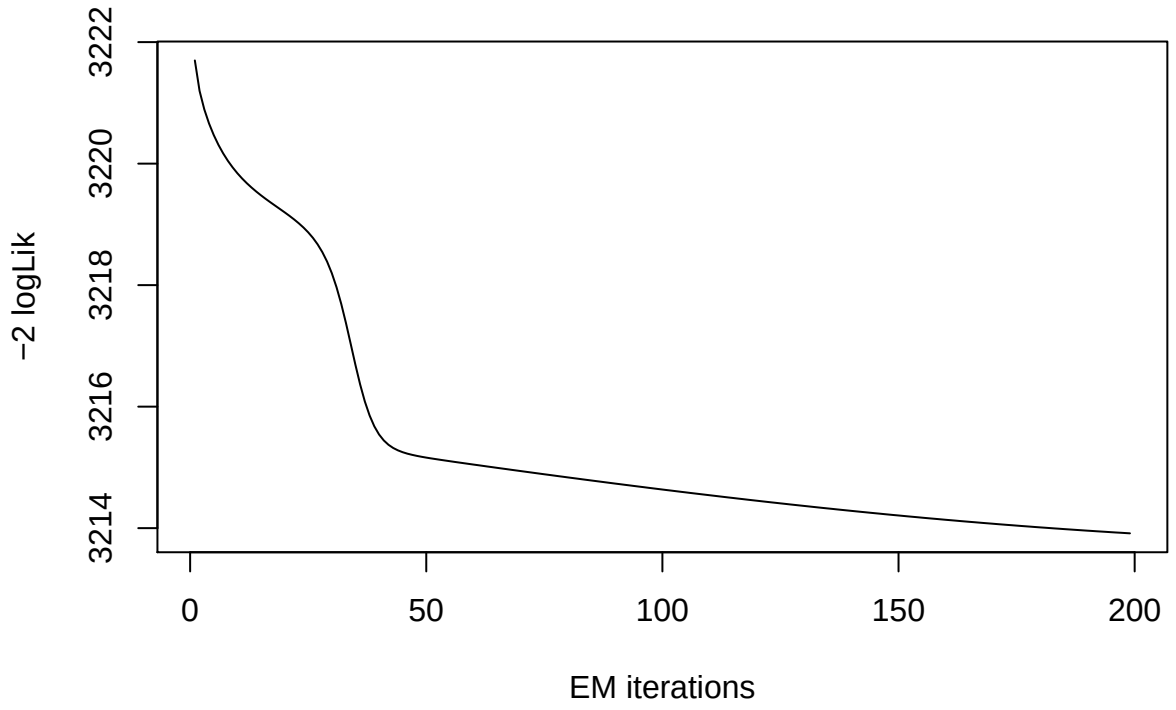
```
## Likelihood ratio test
##
## Model 1: gamlssML(formula = heart$chol, family = "IG")
## Model 2: gamlssML(formula = heart$chol, family = "EXP")
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   2 -1610.1
## 2   1 -1971.4 -1 722.75  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis would be rejected at nearly every significance level. Thus, we know that we should definitely use the Exponential model as it increases the accuracy of our model by a substantial amount.

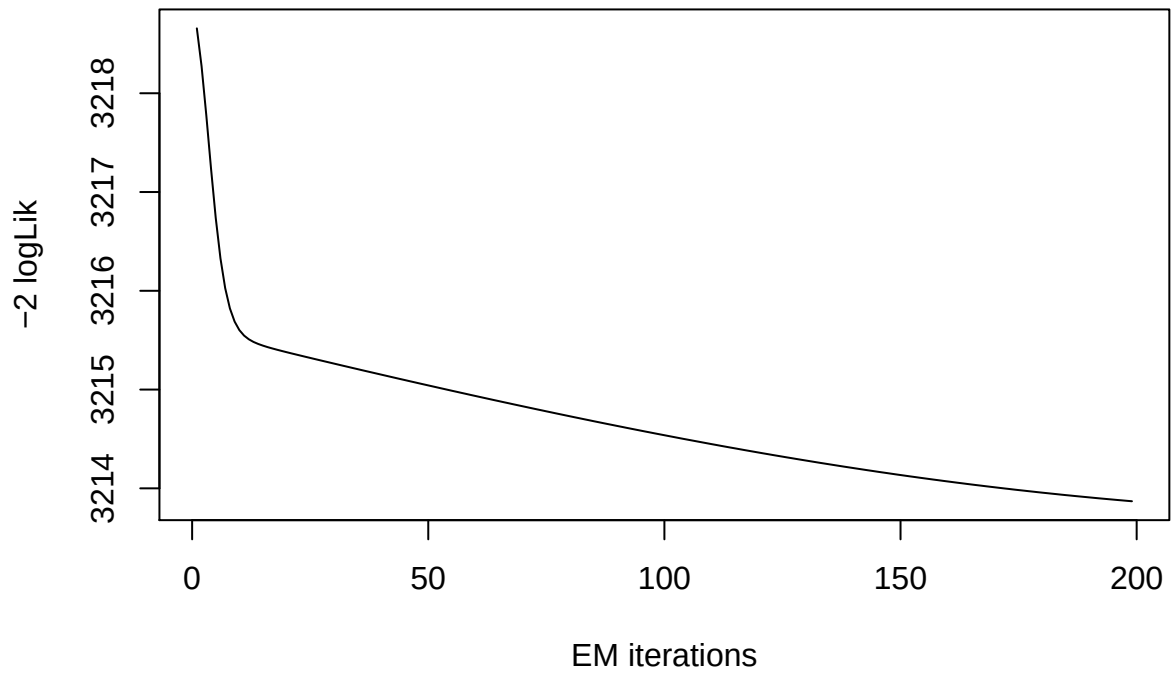
Mixture of distributions

It is possible to compute a mixture of two lognormal distributions In order to find the best mixture,the algorithm is repeated five times.

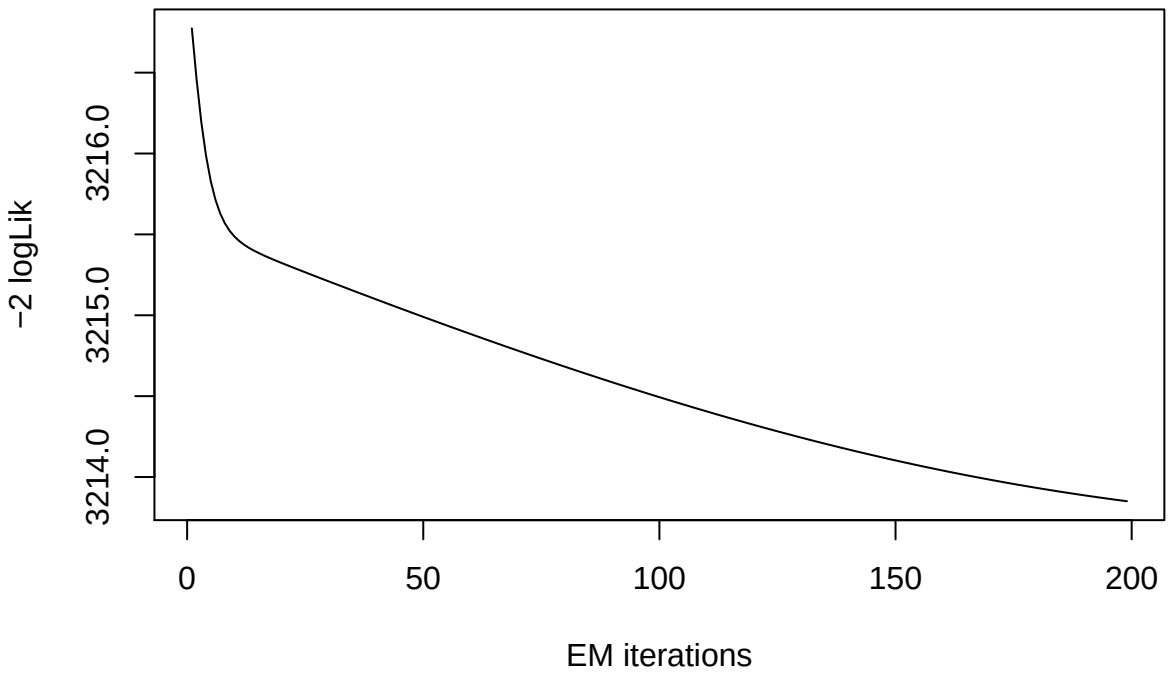
```
mxfit <- gamlssMXfits(n = 5, heart$chol~1, family = LOGNO, K = 2, data = heart)
```



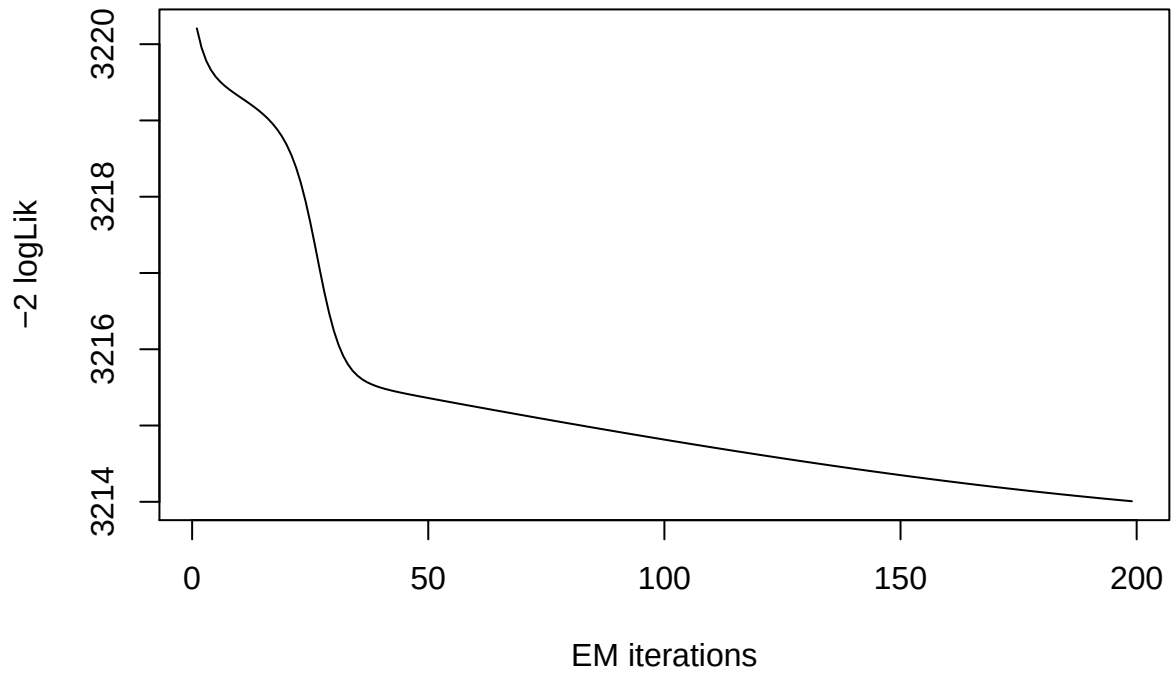

```
## model= 1
```


−2 logLik
3214
3215
3216
3217
3218
0
50
100
150
200
EM iterations

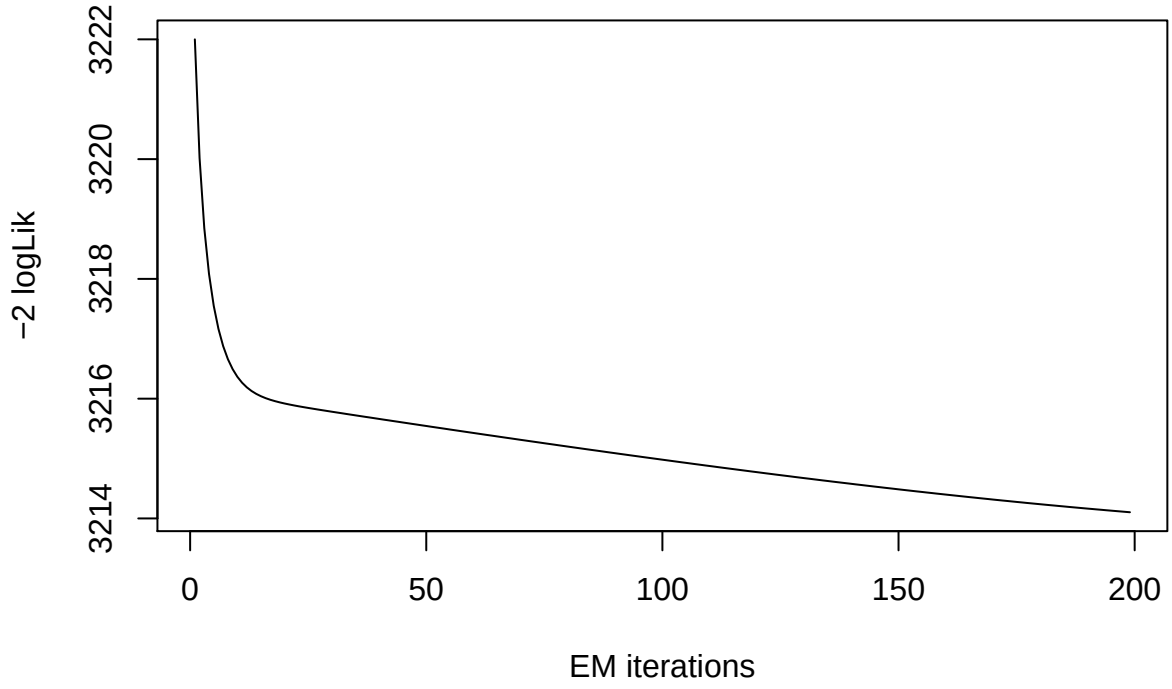
```
## model= 2
```




```
## model= 3
```




```
## model= 4
```




```
## model= 5
```

We can see that with the mixture of two Gamma distributions the AIC and BIC values have improved, as they are lower than those obtained with the single Gamma distribution: AIC is now equal to 3223.84, while the previous value was 3306.970 , BIC is now 3242.41, which is lower than 3314.398 which was the previous value.

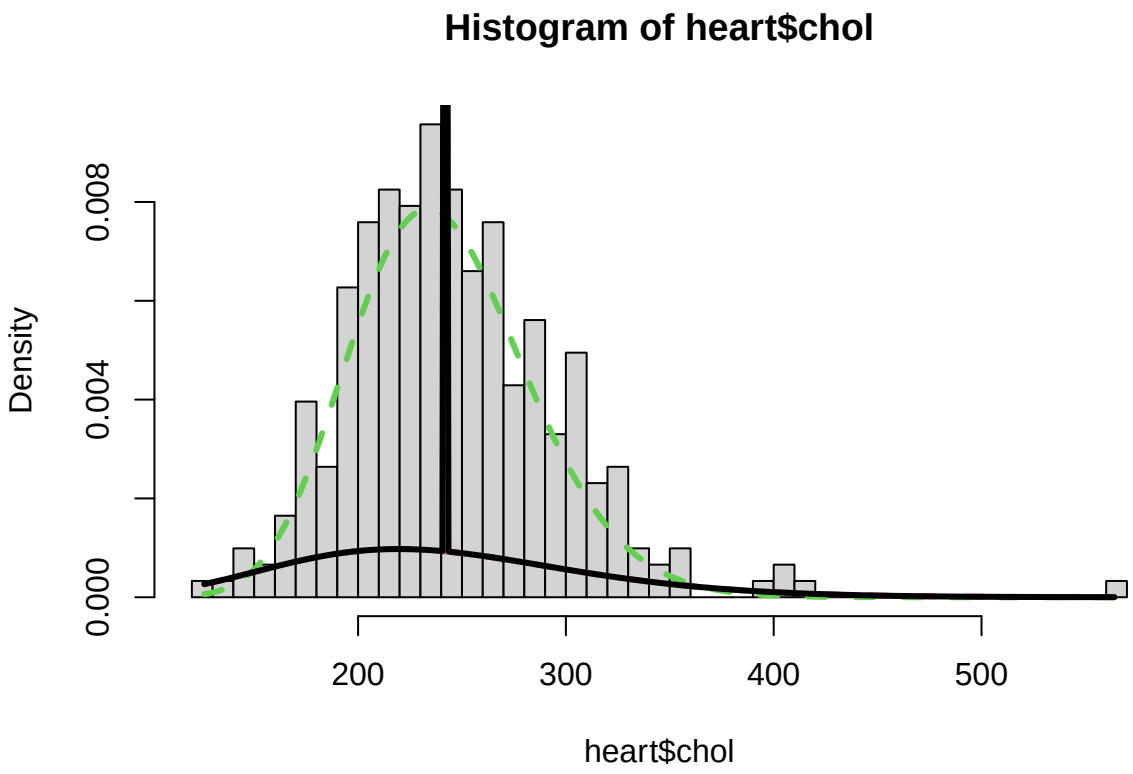
```
hist(heart$chol, breaks = 50,freq = FALSE)
mu.hat1 <- exp(mxfit[["models"]][[1]][["mu.coefficients"]])
sigma.hat1 <- exp(mxfit[["models"]][[1]][["sigma.coefficients"]])
mu.hat2 <- exp(mxfit[["models"]][[2]][["mu.coefficients"]])
sigma.hat2 <- exp(mxfit[["models"]][[2]][["sigma.coefficients"]])
hist(heart$chol, breaks = 50, freq = FALSE, xlab = "Chol",
     main="Chol-Mixture of two Lognormal distributions", plot = FALSE)
```

```
## Warning in hist.default(heart$chol, breaks = 50, freq = FALSE, xlab = "Chol", :
## arguments 'freq', 'main', 'xlab' are not made use of
```

```
## $breaks
##  [1] 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300
## [20] 310 320 330 340 350 360 370 380 390 400 410 420 430 440 450 460 470 480 490
## [39] 500 510 520 530 540 550 560 570
##
## $counts
##  [1]  1  1  3  2  5 12  8 19 23 25 24 29 25 20 23 13 17 10 15  7  8  3  2  3  0
## [26]  0  0  1  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
##
## $density
```

```
##  [1] 0.000330033 0.000330033 0.000990099 0.000660066 0.001650165 0.003960396
##  [7] 0.002640264 0.006270627 0.007590759 0.008250825 0.007920792 0.009570957
## [13] 0.008250825 0.006600660 0.007590759 0.004290429 0.005610561 0.003300330
## [19] 0.004950495 0.002310231 0.002640264 0.000990099 0.000660066 0.000990099
## [25] 0.000000000 0.000000000 0.000000000 0.000330033 0.000660066 0.000330033
## [31] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
## [37] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
## [43] 0.000000000 0.000000000 0.000330033
##
## $mids
##  [1] 125 135 145 155 165 175 185 195 205 215 225 235 245 255 265 275 285 295 305
## [20] 315 325 335 345 355 365 375 385 395 405 415 425 435 445 455 465 475 485 495
## [39] 505 515 525 535 545 555 565
##
## $xname
## [1] "heart$chol"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
lines(seq(min(heart$chol),max(heart$chol),length=length(heart$chol)),
      mxfit[["prob"]][1]*dGA(seq(min(heart$chol),max(heart$chol),                                    lengt
      lty=2,lwd=3,col=2)
lines(seq(min(heart$chol),max(heart$chol),length=length(heart$chol)), mxfit[["prob"]][2]*dGA(seq(min(hea
lines(seq(min(heart$chol),max(heart$chol),length=length(heart$chol)),
      mxfit[["prob"]][1]*dGA(seq(min(heart$chol),max(heart$chol),
                                 length=length(heart$chol)),mu=mu.hat1,sigma=sigma.hat1)+
        mxfit[["prob"]][2]*dRG(seq(min(heart$chol),
                                   max(heart$chol),length=length(heart$chol)),
                               mu= mu.hat2,sigma = sigma.hat2), lty = 1,
      lwd = 3, col = 1)
```



Since we have selected K=2, we can see in the plot 2 peaks, each corresponding to a distribution.The dotted red line is relative to the first distribution,the dotted green one to the second distribution, while the black line is relative to the mixture, i.e. the overall model for all data.

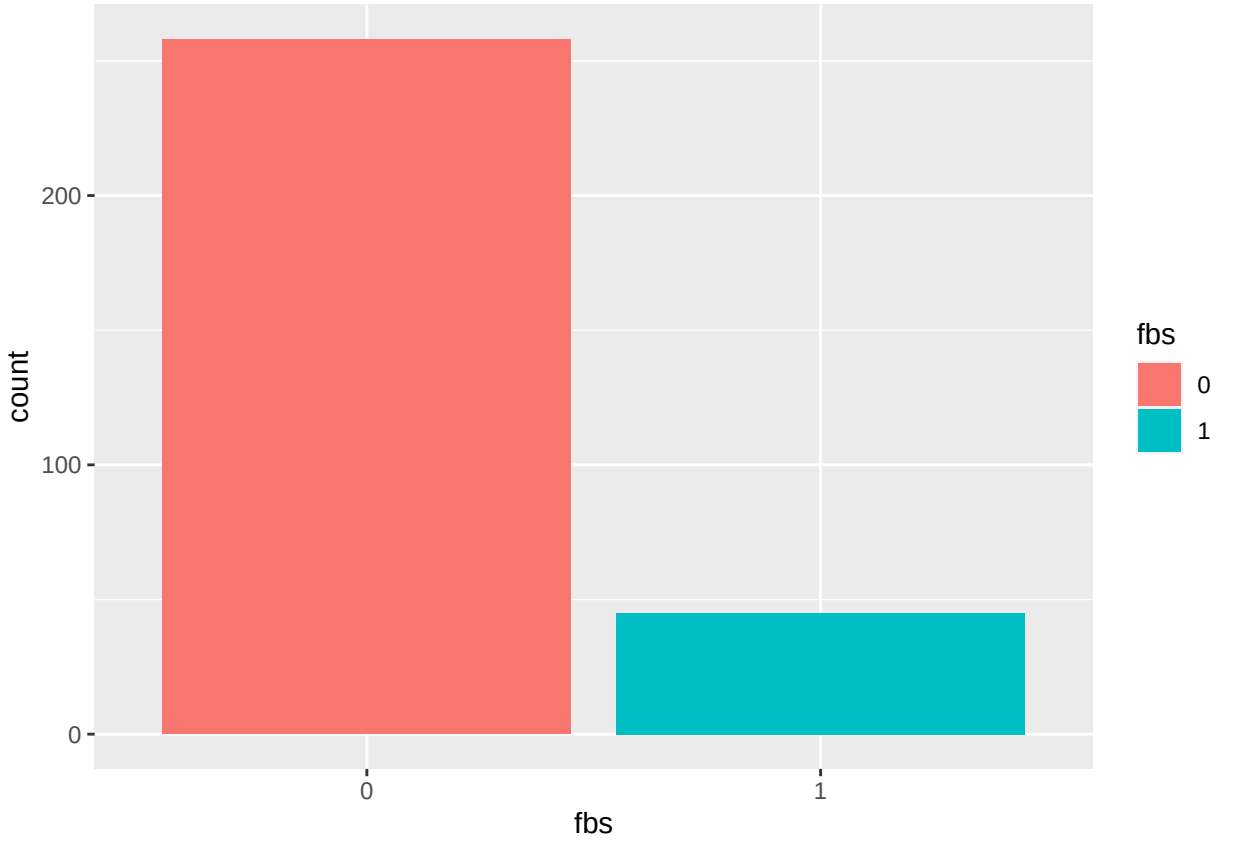
Fasting blood sugar(fbs)

Fasting blood sugar is a categorical variable that aims to show if a patient has fbs more than 120 mg/dl or not.

```
summary(heart$fbs)
```

```
##   0   1
## 258  45
```

```
ggplot(heart, aes(x = fbs, fill=fbs)) + geom_bar()
```



As the result shows 258 people have fbs less than 120 mg/dl and 45 more than 120 mg/dl.So from both plot and summary we can see that our data is not balanced.

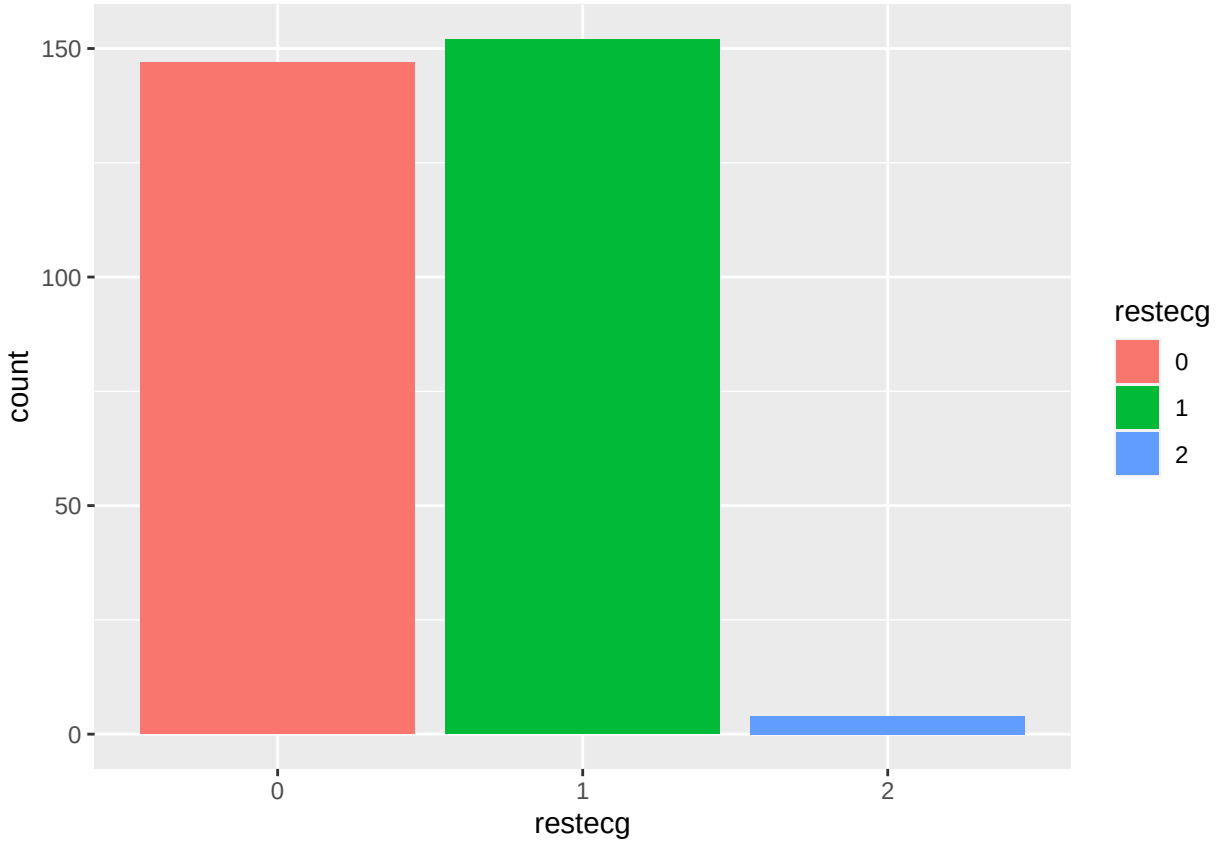
Resting electrocardiographic results (restecg)

Resting electrocardiographic results is a categorical variable that categorizes results in 3 groups 0, 1,2.

```
summary(heart$restecg)
```

```
##   0   1   2
## 147 152   4
```

```
ggplot(heart, aes(x = restecg, fill=restecg)) + geom_bar()
```



As the result shows group 0 contains 147 records, group 1 contains 152 records and group 2 contains 4 values. As you can see the data is not balanced between groups and group 1 contains maximum records while group 2 includes minimum records.

Maximum heart rate achieved(thalach)

Maximum heart rate achieved(thalach) is a numerical continuous variable in rage (0,).

```
length(heart$thalach)
```

```
## [1] 303
```

```
min(heart$thalach)
```

```
## [1] 71
```

```
max(heart$thalach)
```

```
## [1] 202
```

The length of sample is 303, and the values are in range 71-202.

```
summary(heart$thalach)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    71.0   133.5   153.0   149.6   166.0   202.0
```

```
sd(heart$thalach)
```

```
## [1] 22.90516
```

```r
var(heart$thalach)
```

```
## [1] 524.6464
```

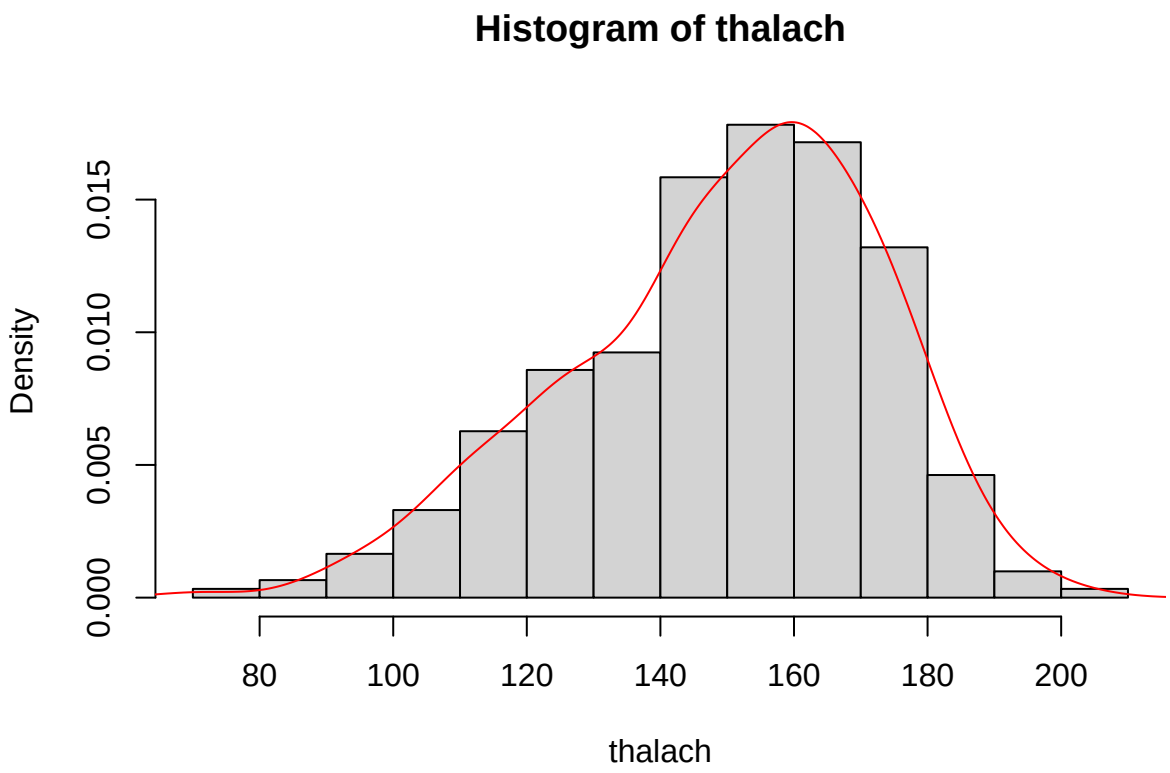
From the results we can see the Mean and Median are not equal so the distribution is asymmetrical or skewed. As we can see mean<median so the distribution should be negative and from histogram it appears to be a unimodal distribution.

```r
skewness(heart$thalach)
```

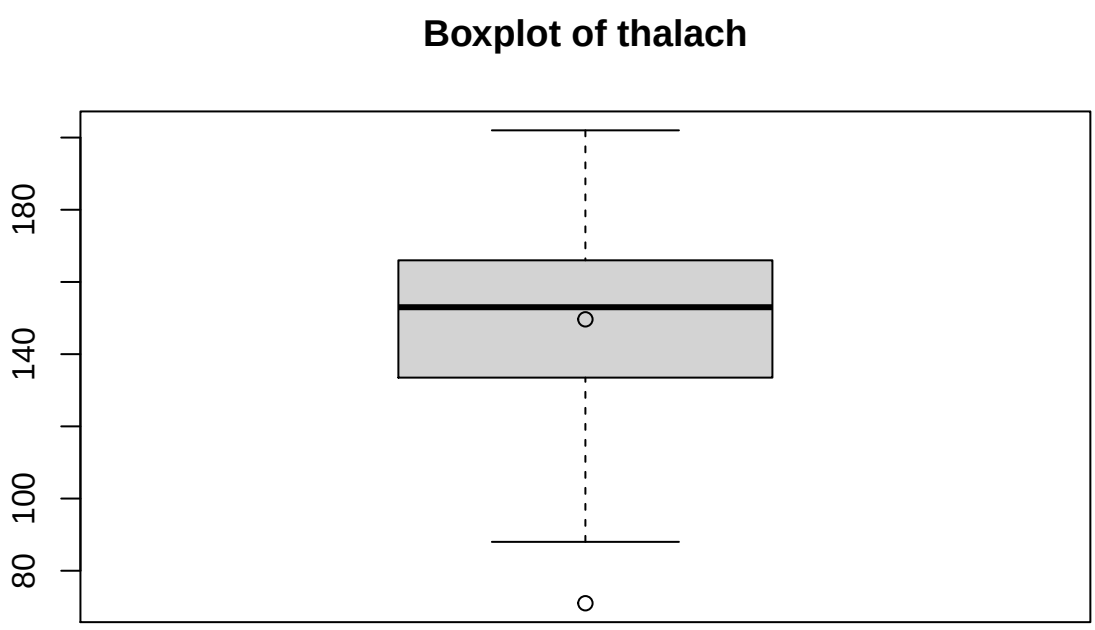
```
## [1] -0.5321005
```

The skewness is negative so distribution is skewed towards the left, and many values are lower than the mean.

```r
hist(heart$thalach, freq=FALSE, xlab= "thalach", main = "Histogram of thalach")
lines(density(heart$thalach), col="red")
```




```r
boxplot(heart$thalach, main = "Boxplot of thalach")
points(mean(heart$thalach))
```

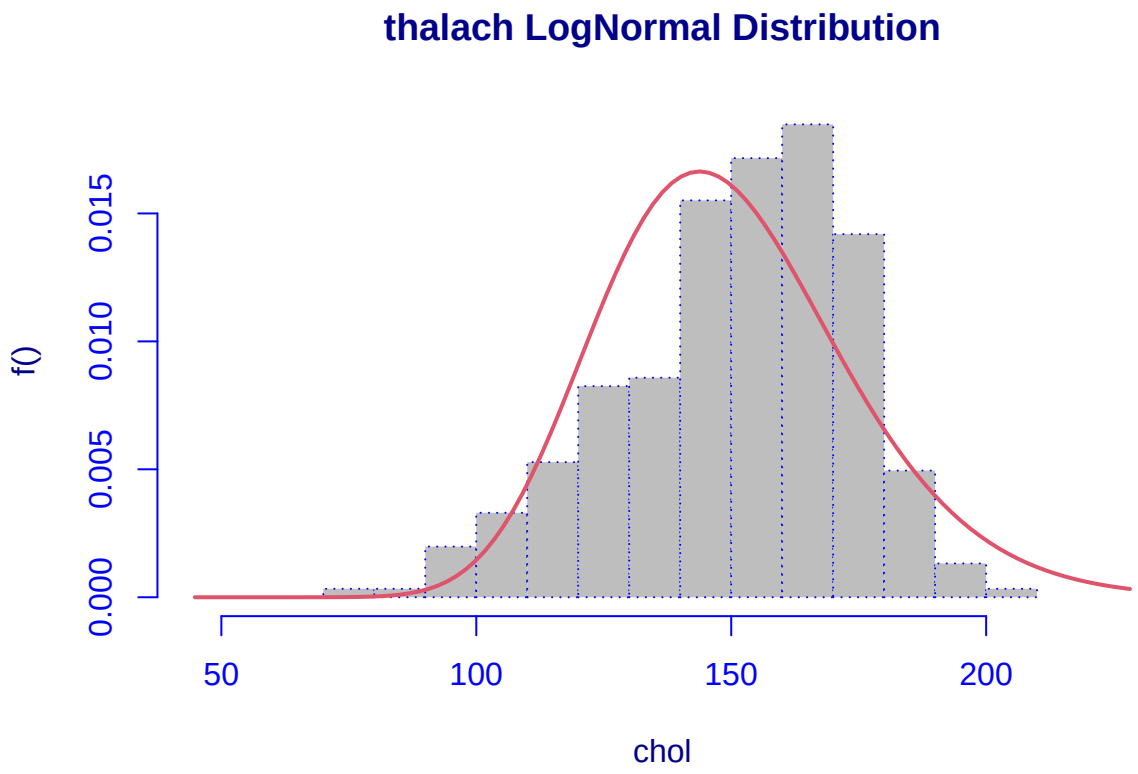


The distribution of data above the mean is more dense than below the mean line. We can see that most of the patients have a maximum heart rate more than 155.

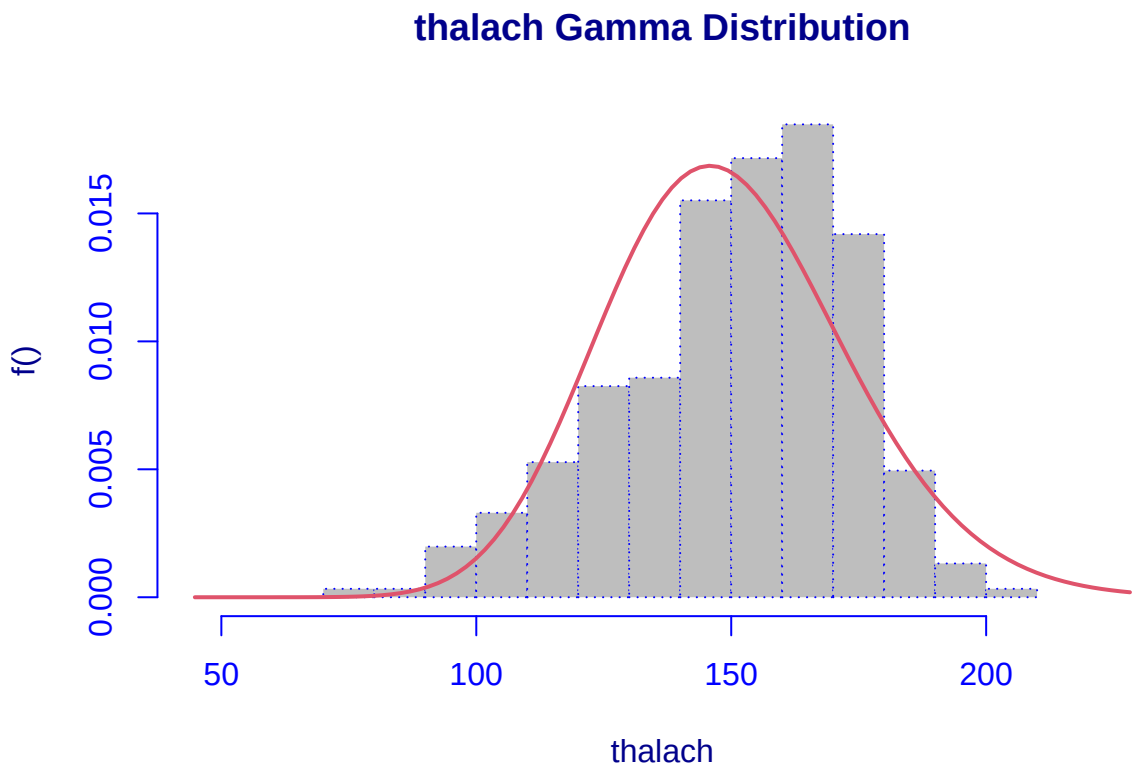
Data fitting for thalach

According to the domain of the variable, the distributions supported on the set of positive infinite real numbers are fitted on the data. Log-likelihood value, BIC and AIC are estimated in order to evaluate the fit of the distribution.

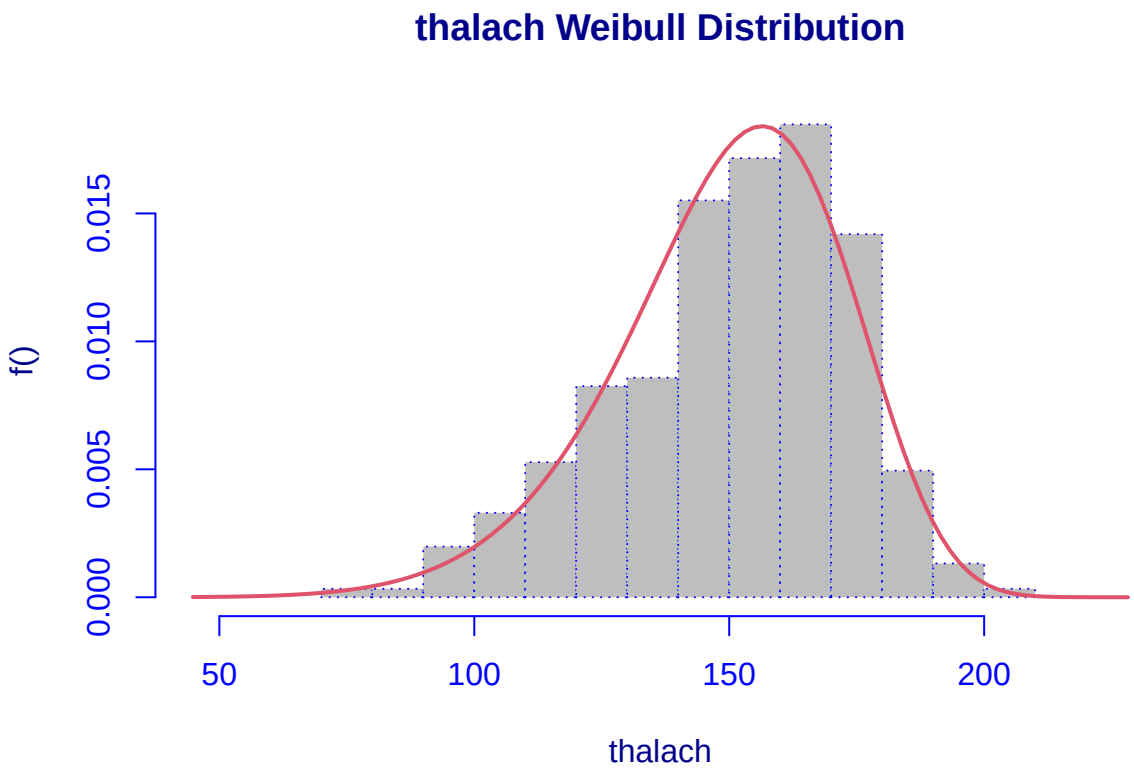
```
thalach.logno<-histDist(heart$thalach, xlab = "chol" , family=LOGNO, nbins=13,
  main = "thalach LogNormal Distribution")
```



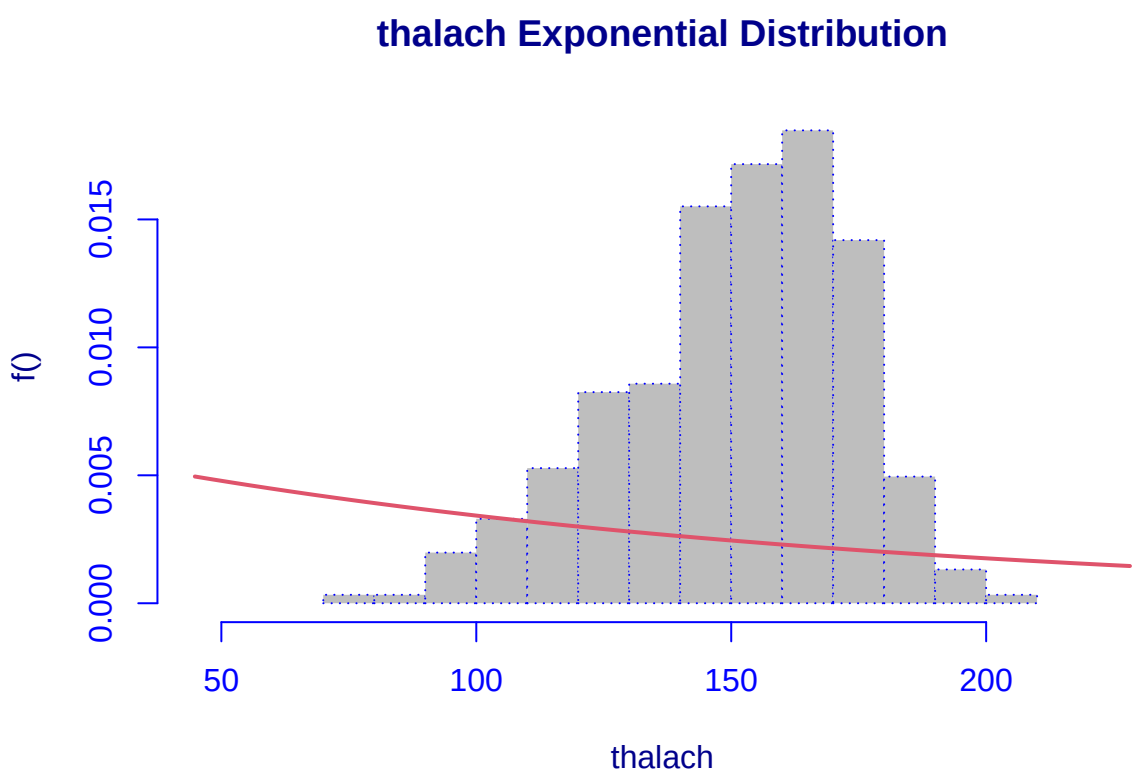

```
thalach.ga<-histDist(heart$thalach, xlab = "thalach" , family=GA, nbins=13,
          main = "thalach Gamma Distribution")
```



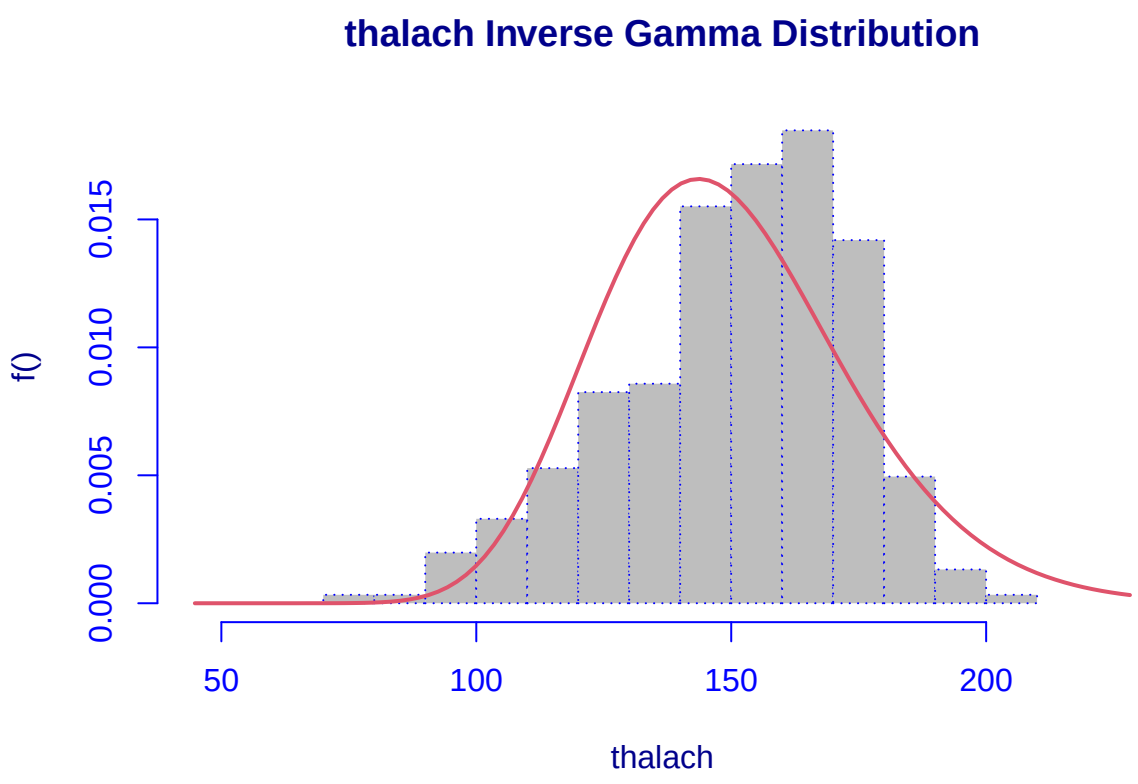

```
thalach.wei<-histDist(heart$thalach, xlab = "thalach" , family=WEI, nbins=13,
         main = "thalach Weibull Distribution")
```




```
thalach.exp<-histDist(heart$thalach, xlab = "thalach" , family=EXP, nbins=13,
         main = "thalach Exponential Distribution")
```




```
thalach.ig<-histDist(heart$thalach, xlab = "thalach" , family=IG, nbins=13,
         main = "thalach Inverse Gamma Distribution")
```




```
thalach.matrix<-matrix(c(thalach.logno$df.fit, logLik(thalach.logno), AIC(thalach.logno), thalach.logno
colnames(thalach.matrix)<-c("df","LogLik", "AIC", "BIC")
rownames(thalach.matrix)<-c("EXP", "GA", "LOGNO", "IG", "WEI")
thalach.matrix<-as.table(thalach.matrix)
thalach.matrix
```

```
##              df    LogLik       AIC       BIC
## EXP       2.000 -1396.657  2797.315  2804.742
## GA        2.000 -1389.205  2782.409  2789.837
## LOGNO     2.000 -1368.355  2740.711  2748.138
## IG        1.000 -1820.508  3643.017  3646.730
## WEI       2.000 -1397.157  2798.313  2805.741
```

As we can see the model that has maximum value of log likelihood and less value in AIC and BIC is "LogNormal distribution" so based on maximum likelihood method its safe to say that our data fits better in Exponential distribution.

Likelihood ratio test

The Likelihood-ratio test goodness of fit performed between the Inverse Gamma model (under the null hypothesis) and the LogNormal model (under the alternative hypothesis).

```
lrtest(thalach.ig, thalach.logno)
```

```
## Likelihood ratio test
##
## Model 1: gamlssML(formula = heart$thalach, family = "IG")
## Model 2: gamlssML(formula = heart$thalach, family = "LOGNO")
```

```
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   2 -1397.2
## 2   2 -1396.7  0 0.9983  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis would be rejected at nearly every significance level. Thus, we know that we should definitely use the LogNormal model as it increases the accuracy of our model by a substantial amount.

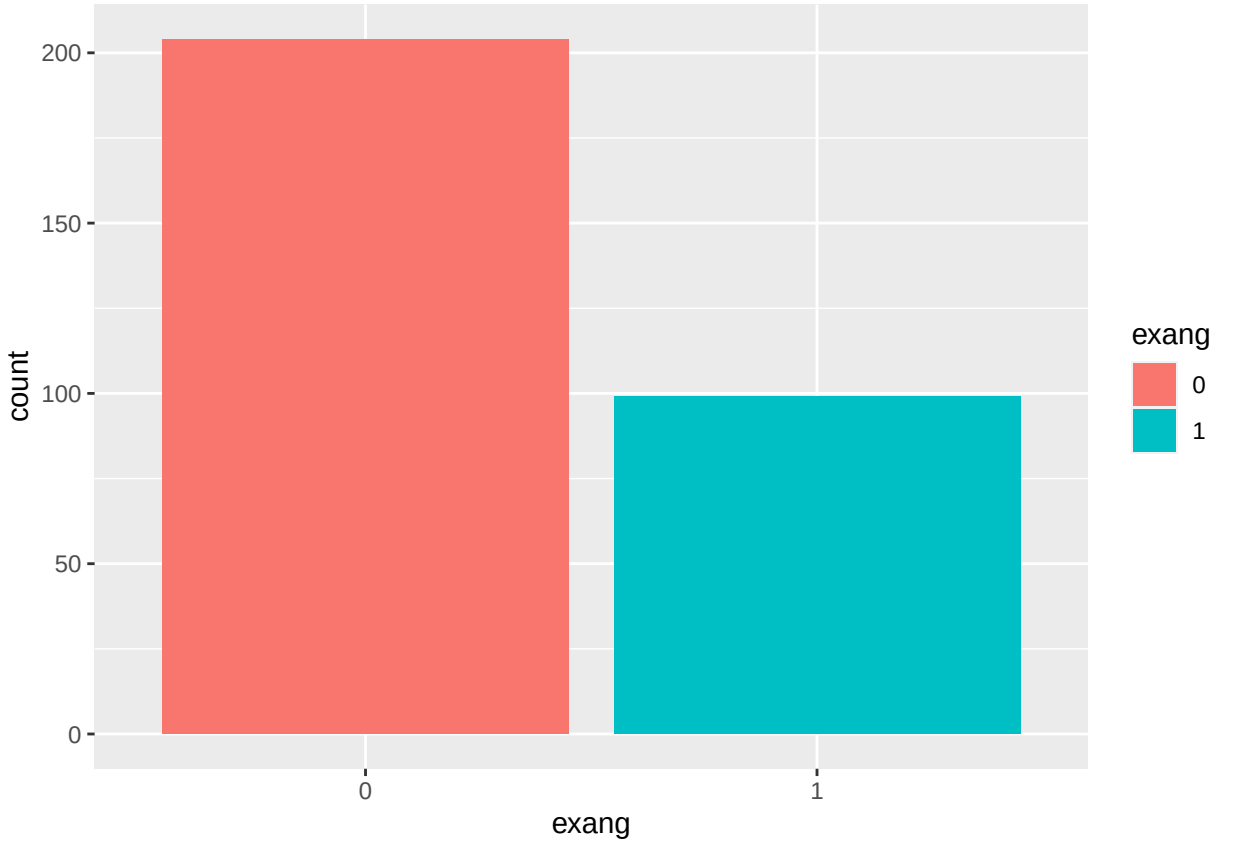
Exercise induced angina (exang)

Exercise induced angina (exang) is a categorical variable that categorizes results in 2 groups 0 for false, 1 for true.

```
summary(heart$exang)
```

```
##   0   1
## 204  99
```

```
ggplot(heart, aes(x = exang, fill=exang)) + geom_bar()
```



As the result shows group 0 contains 204 records, group 1 contains 99 records. As you can see the data is not balanced and most records are related to group 0 that give us the assumption that exercise most of the time is not a factor to induce angina .

ST depression induced by exercise relative to rest(oldpeak)

ST depression induced by exercise relative to rest(oldpeak) is a numerical continuous variable in rage [0,).

```r
length(heart$oldpeak)
```

```
## [1] 303
```

```r
min(heart$oldpeak)
```

```
## [1] 0
```

```r
max(heart$oldpeak)
```

```
## [1] 6.2
```

The length of sample is 303, and the values are in range 0-6.2.

```r
summary(heart$oldpeak)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.80    1.04    1.60    6.20
```

```r
sd(heart$oldpeak)
```

```
## [1] 1.161075
```

```r
var(heart$oldpeak)
```

```
## [1] 1.348095
```

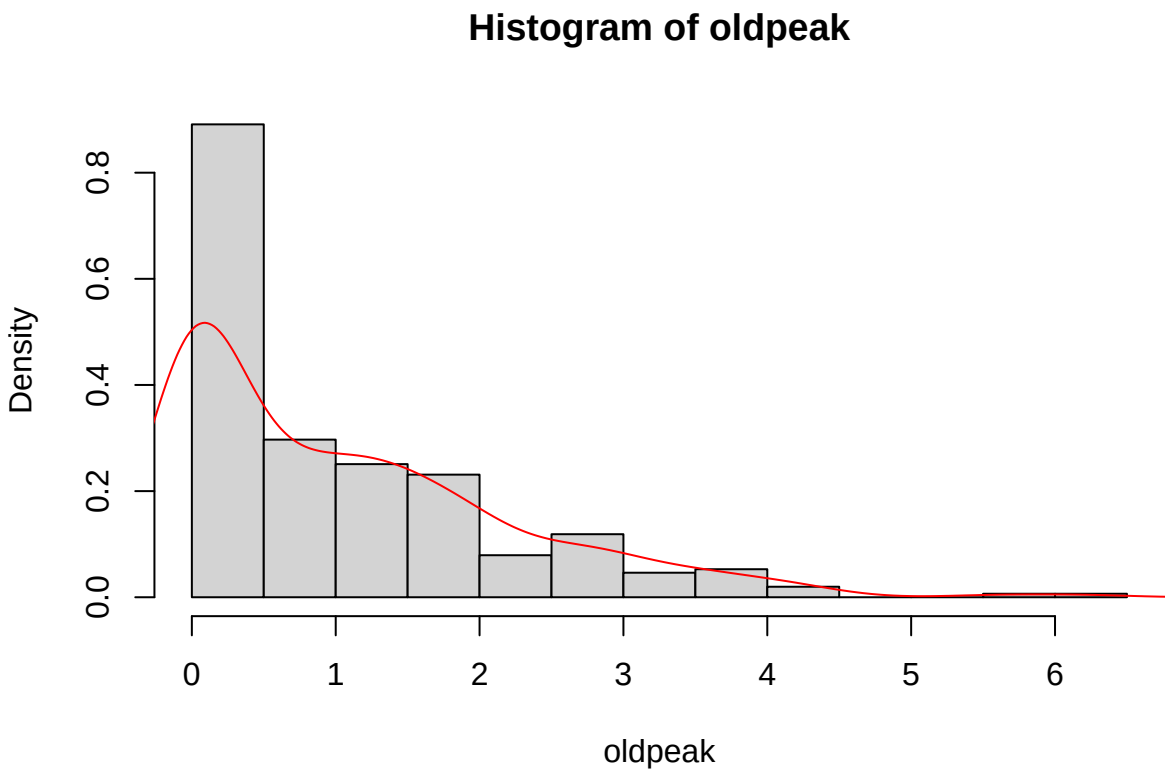
From the results we can see the Mean and Median are not equal so the distribution is asymmetrical or skewed. As we can see mean>median so the distribution should be positive and from histogram it appears to be a unimodal distribution.

```r
skewness(heart$oldpeak)
```

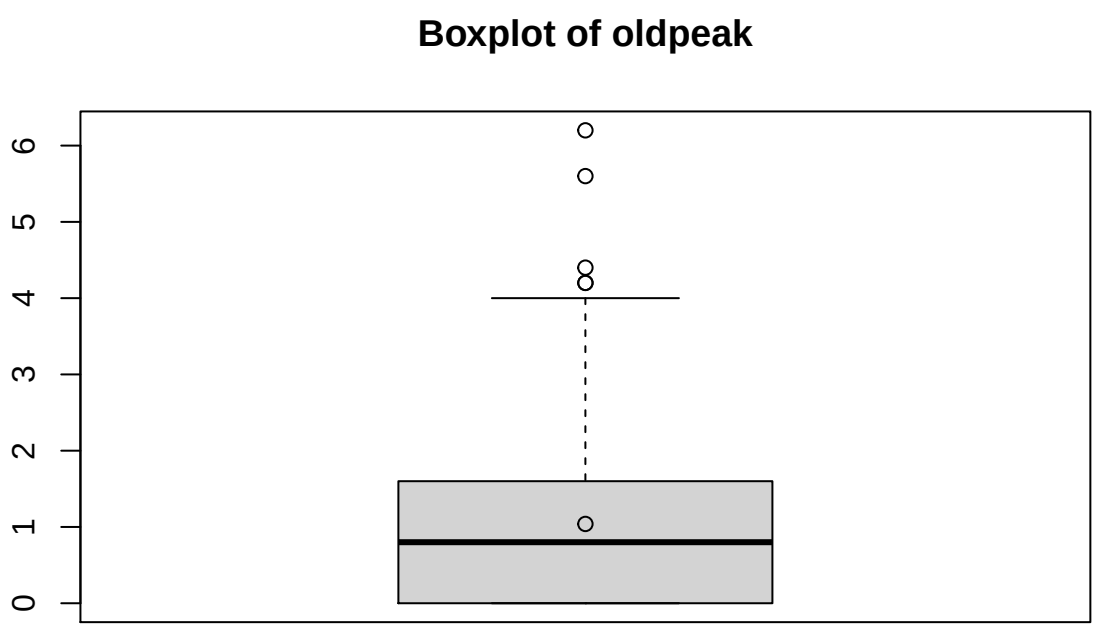
```
## [1] 1.257176
```

The skewness is positive so distribution is skewed towards the left, and many values are greater than the mean.

```r
hist(heart$oldpeak, freq=FALSE, xlab= "oldpeak", main = "Histogram of oldpeak")
lines(density(heart$oldpeak), col="red")
```




```
boxplot(heart$oldpeak, main = "Boxplot of oldpeak")
points(mean(heart$oldpeak))
```

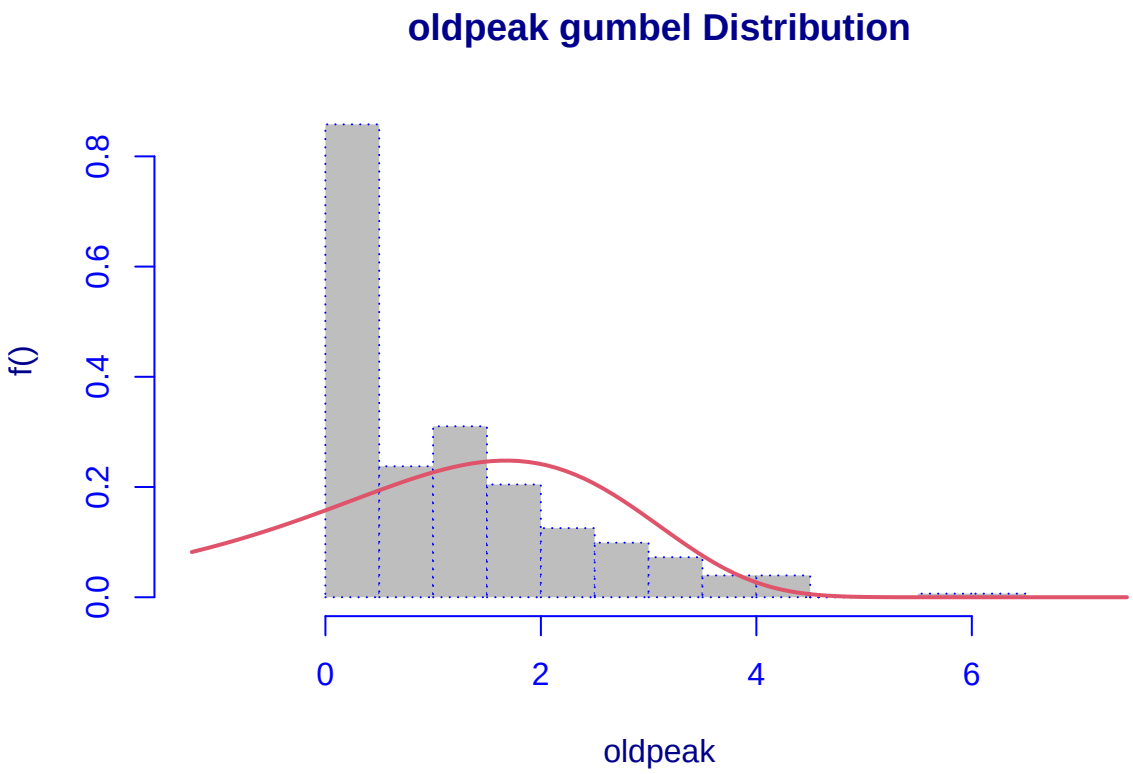


The plot shows the presence of some outliers.

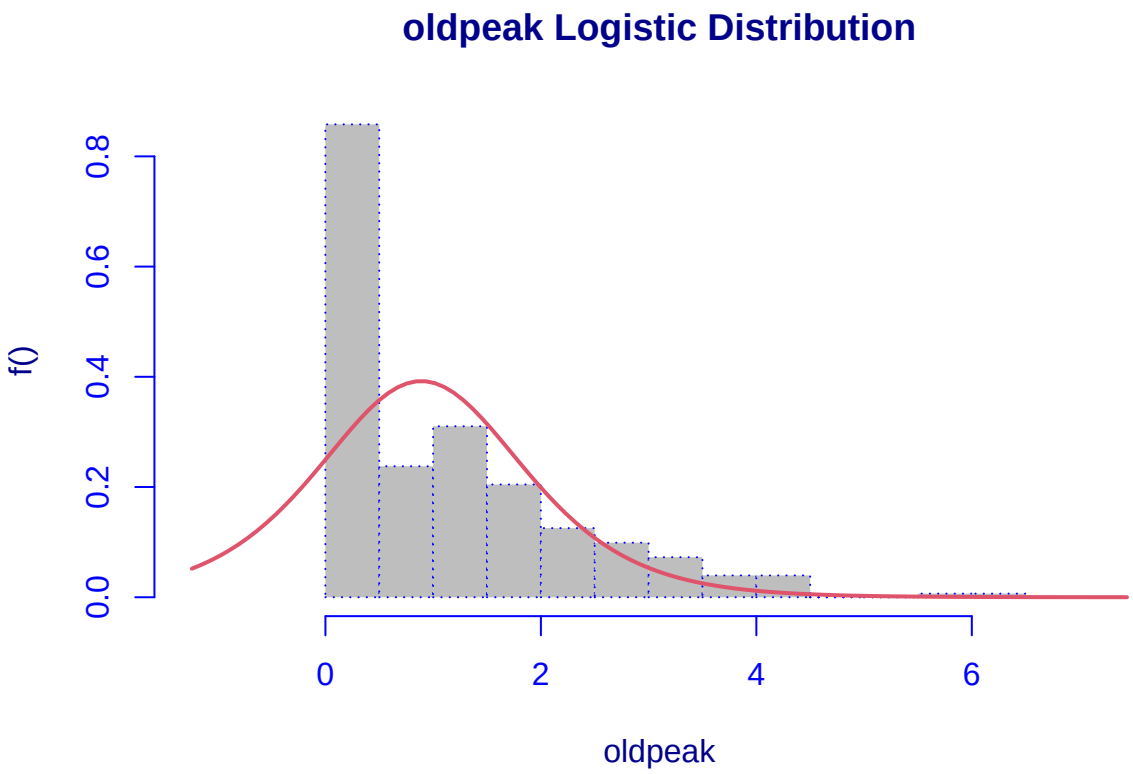
Data fitting for oldpeak

According to the domain of the variable, the distributions from last analysis are not applicable here because Lognormal, Gamma, weibull, inverse gamma work in range (0, infinite) so we need other distributions that work in range [0, infinite). The distributions supported these data are fitted on the data. Log-likelihood value, BIC and AIC are estimated in order to evaluate the fit of the distribution.

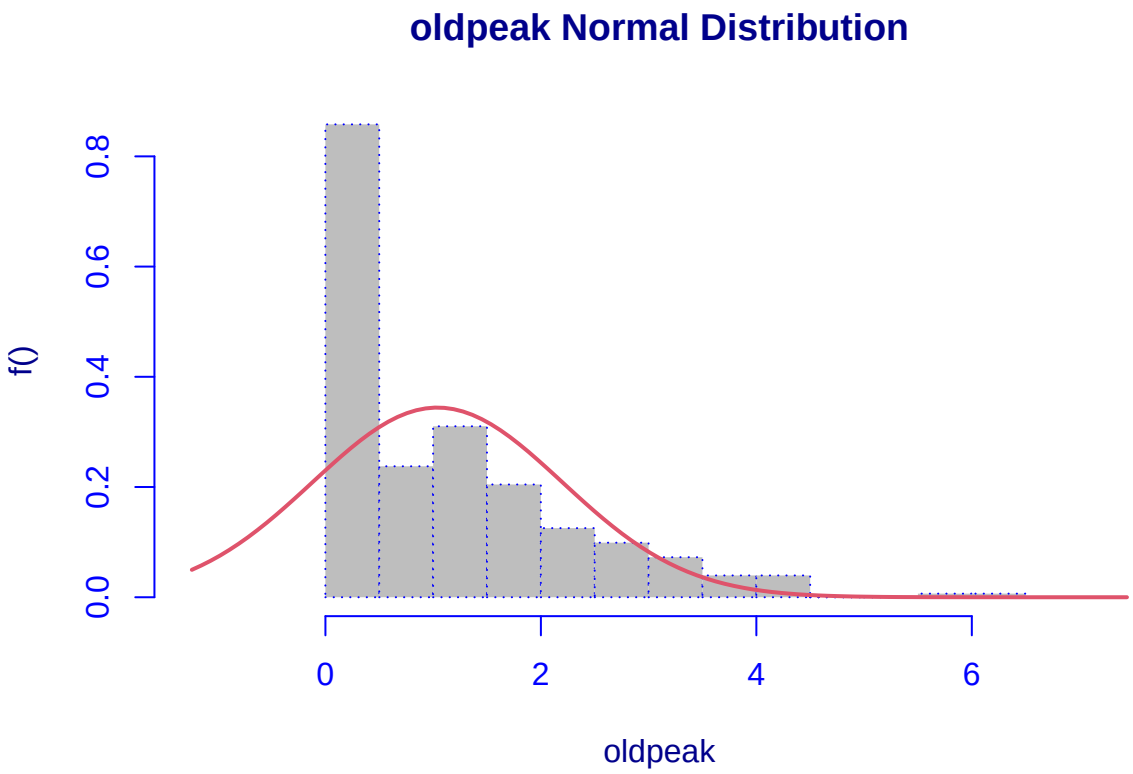
```
oldpeak.gu<-histDist(heart$oldpeak, xlab = "oldpeak" , family=GU, nbins=13,
  main = "oldpeak gumbel Distribution")
```



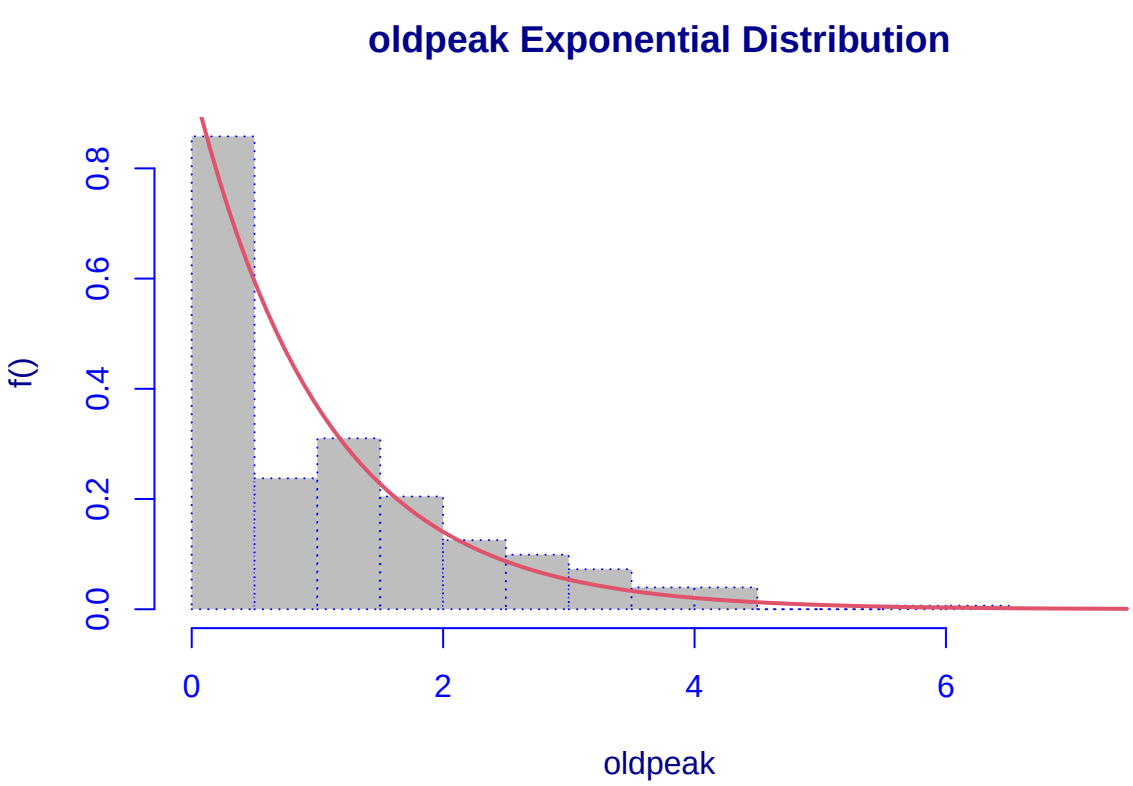

```
oldpeak.log<-histDist(heart$oldpeak, xlab = "oldpeak" , family=LO, nbins=13,
        main = "oldpeak Logistic Distribution")
```



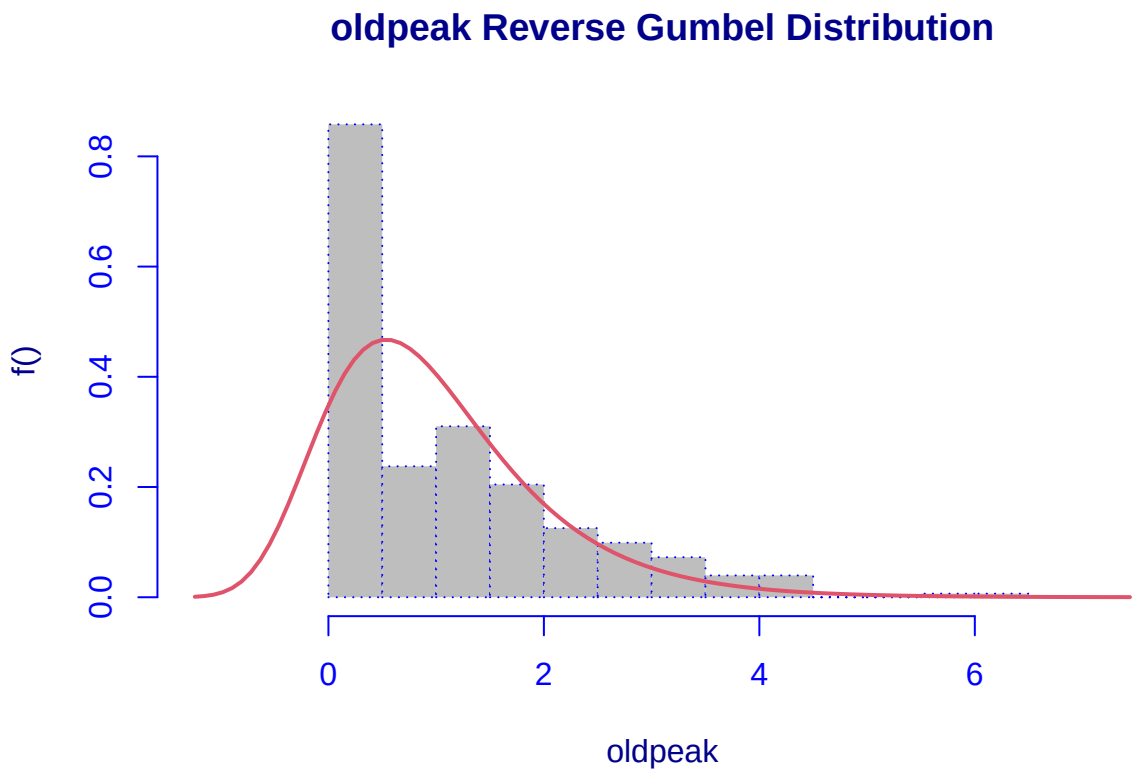

```
oldpeak.normal<-histDist(heart$oldpeak, xlab = "oldpeak" , family=NO, nbins=13,
         main = "oldpeak Normal Distribution")
```




```
oldpeak.exp<-histDist(heart$oldpeak, xlab = "oldpeak" , family=EXP, nbins=13,
         main = "oldpeak Exponential Distribution")
```




```
oldpeak.rg<-histDist(heart$oldpeak, xlab = "oldpeak" , family=RG, nbins=13,
          main = "oldpeak Reverse Gumbel Distribution")
```




```
oldpeak.matrix<-matrix(c(oldpeak.gu$df.fit, logLik(oldpeak.gu),
                         AIC(oldpeak.gu), oldpeak.gu$sbc, oldpeak.log$df.fit,  logLik(oldpeak.log), AIC
                       ncol=4, byrow=TRUE )
colnames(oldpeak.matrix)<-c("df","LogLik", "AIC", "BIC")
rownames(oldpeak.matrix)<-c("GU", "LO", "NO", "EXP", "RG")
oldpeak.matrix<-as.table(oldpeak.matrix)
oldpeak.matrix
```

```
##            df    LogLik       AIC       BIC
## GU     2.0000 -552.5162 1109.0324 1116.4599
## LO     2.0000 -469.2152  942.4304  949.8579
## NO     2.0000 -474.6895  953.3790  960.8064
## EXP    1.0000 -314.7685  631.5369  635.2507
## RG     2.0000 -424.4572  852.9143  860.3418
```

As we can see the model that has maximum value of log likelihood and less value in AIC and BIC is "Exponential distribution" so based on maximum likelihood method its safe to say that our data fits better in Exponential distribution.

Likelihood ratio test

The Likelihood-ratio test goodness of fit performed between the Normal model (under the null hypothesis) and the Exponential model (under the alternative hypothesis).

```
lrtest(oldpeak.normal, oldpeak.exp)
```

```
## Likelihood ratio test
##
```

```
## Model 1: gamlssML(formula = heart$oldpeak, family = "NO")
## Model 2: gamlssML(formula = heart$oldpeak, family = "EXP")
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   2 -474.69
## 2   1 -314.77 -1 319.84  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis would be rejected at nearly every significance level. Thus, we know that we should definitely use the Exponential model as it increases the accuracy of our model by a substantial amount.

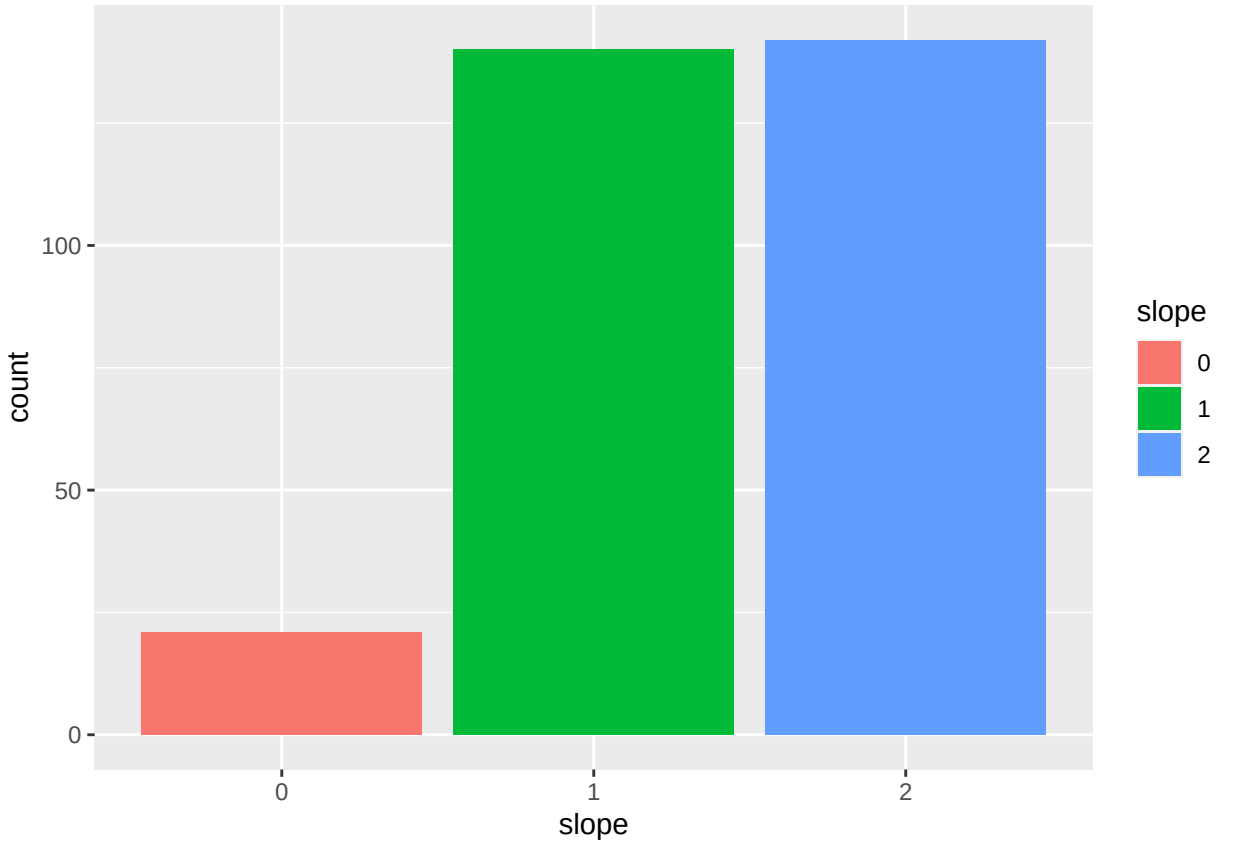
The slope of the peak exercise ST segment (slope)

The slope of the peak exercise ST segment (slope) is a categorical variable that categorizes results in 3 groups 0, 1, 2.

```
summary(heart$slope)
```

```
##   0   1   2
##  21 140 142
```

```
ggplot(heart, aes(x = slope, fill=slope)) + geom_bar()
```



As the result shows group 0 contains least value between other groups while values in group1 and 2 are distributed almost equal.

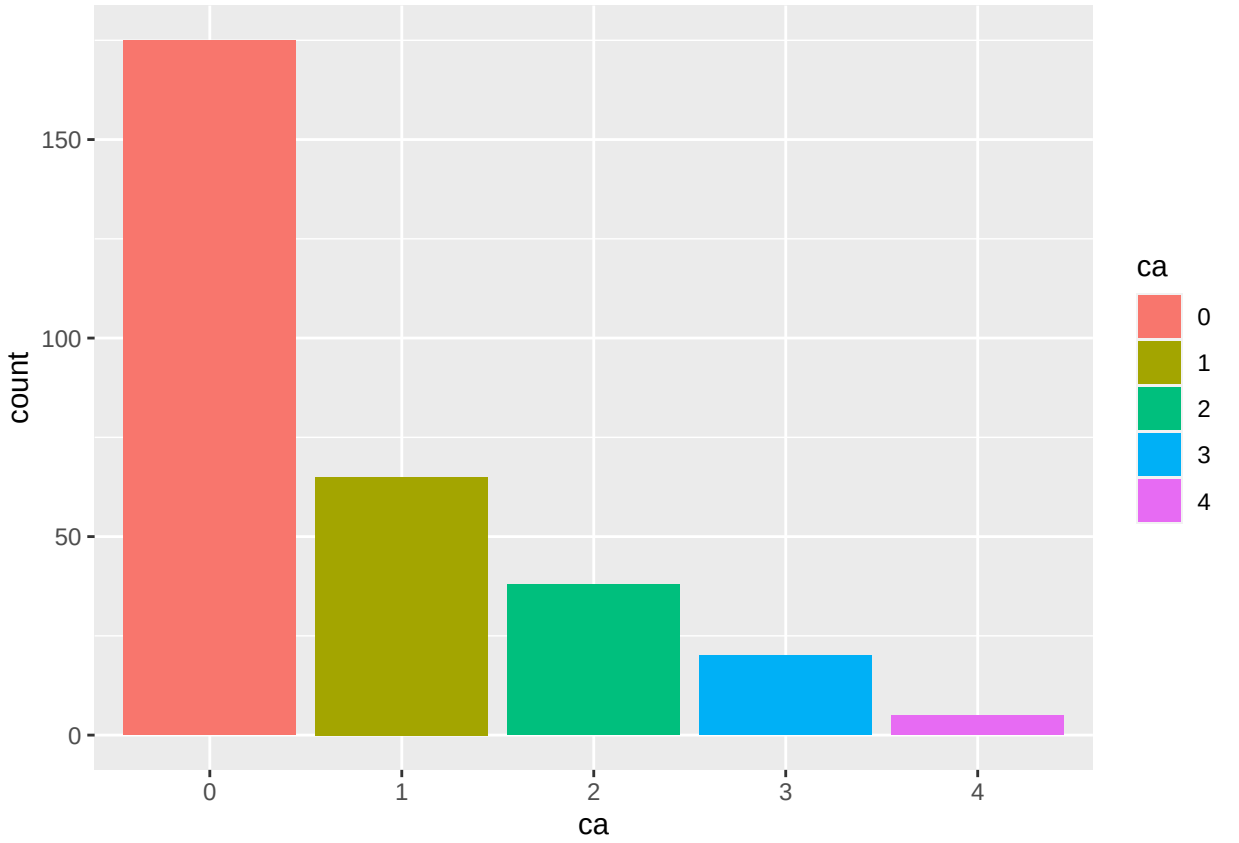
Number of major vessels colored by fluoroscopy(ca)

Number of major vessels colored by fluoroscopy(ca) is a categorical variable that categorizes results in 5 groups 0, 1, 2, 3, 4.

```
summary(heart$ca)
```

```
##   0   1   2   3   4
## 175  65  38  20   5
```

```
ggplot(heart, aes(x = ca, fill=ca)) + geom_bar()
```



As the result shows data are not balanced between groups and group 0 contains most values between other groups while group4 has the least value.

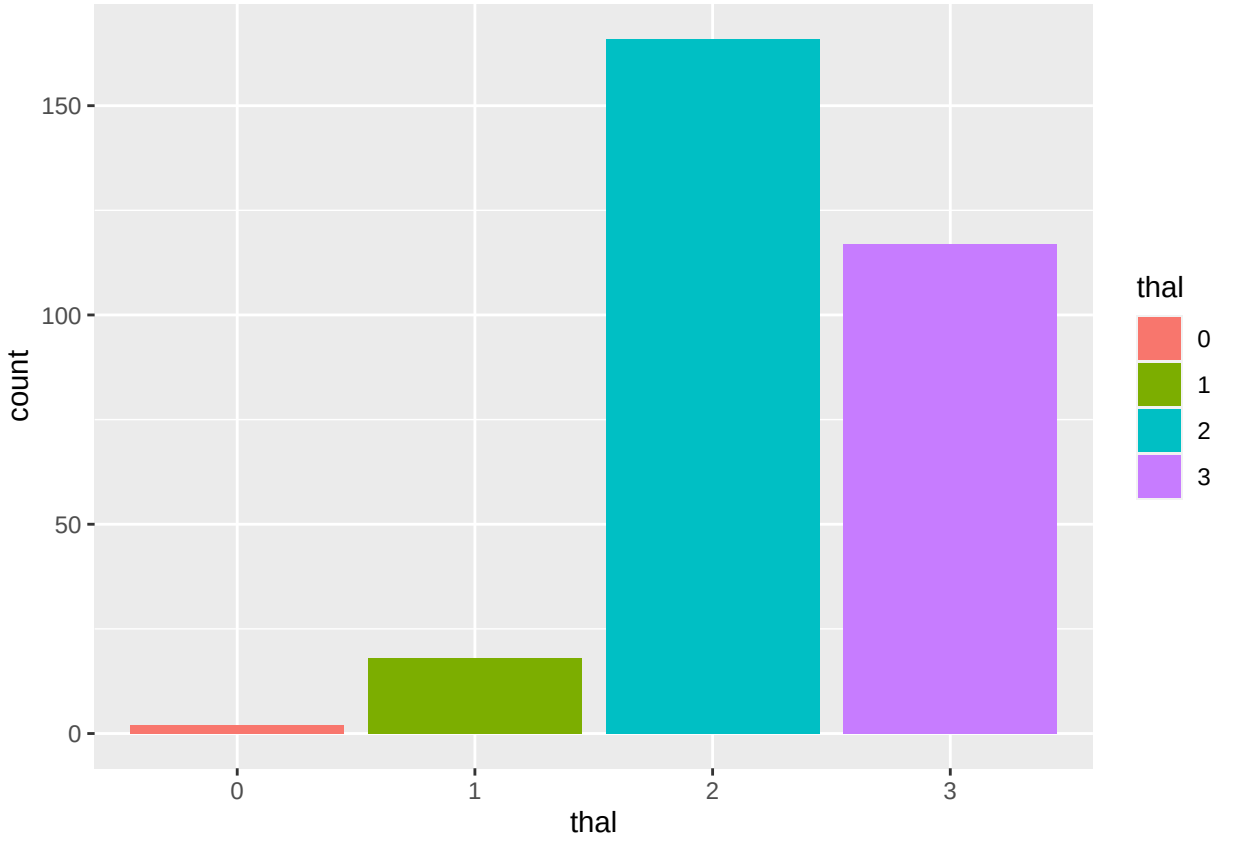
Thal

thal is a categorical variable that categorizes results in 4 groups.

```
summary(heart$thal)
```

```
##   0   1   2   3
##   2  18 166 117
```

```
ggplot(heart, aes(x = thal, fill=thal)) + geom_bar()
```



As the result shows data are not balanced between groups and group 0 contains the least values between other groups while group2 has the most value.

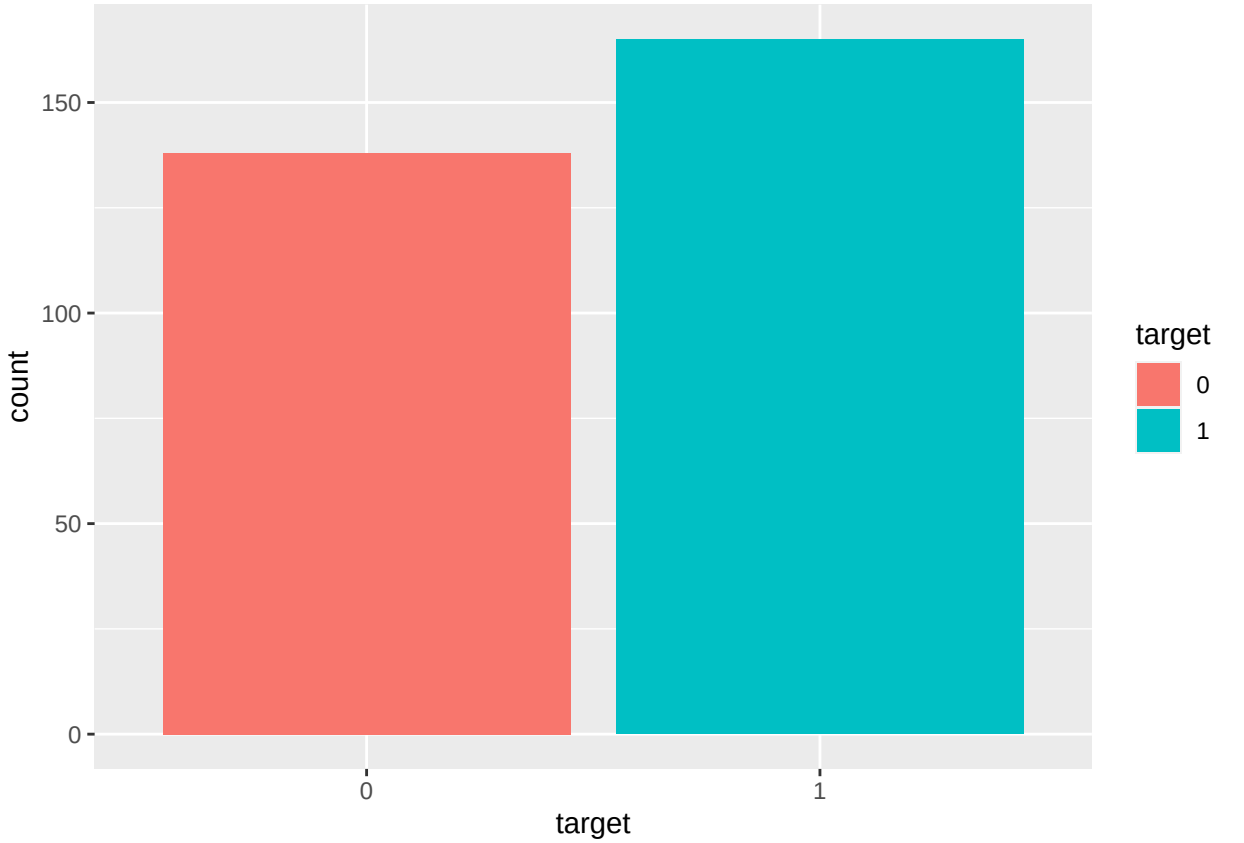
Target

Target is a categorical variable that categorizes results in 2 groups.

```
summary(heart$target)
```

```
##   0   1
## 138 165
```

```
ggplot(heart, aes(x = target, fill=target)) + geom_bar()
```



As the result shows data are not balanced between groups and group 0 contains the least values between other groups while group2 has the most value.

Principal Component Analysis

Before proceeding with the PCA, it is necessary to evaluate whether there is correlation between the numerical variables. For this aim first we need a sub dataset with all continuous values.

```
heart_sub <- heart[ -c(2,3,6,7,9,11:14) ]
```

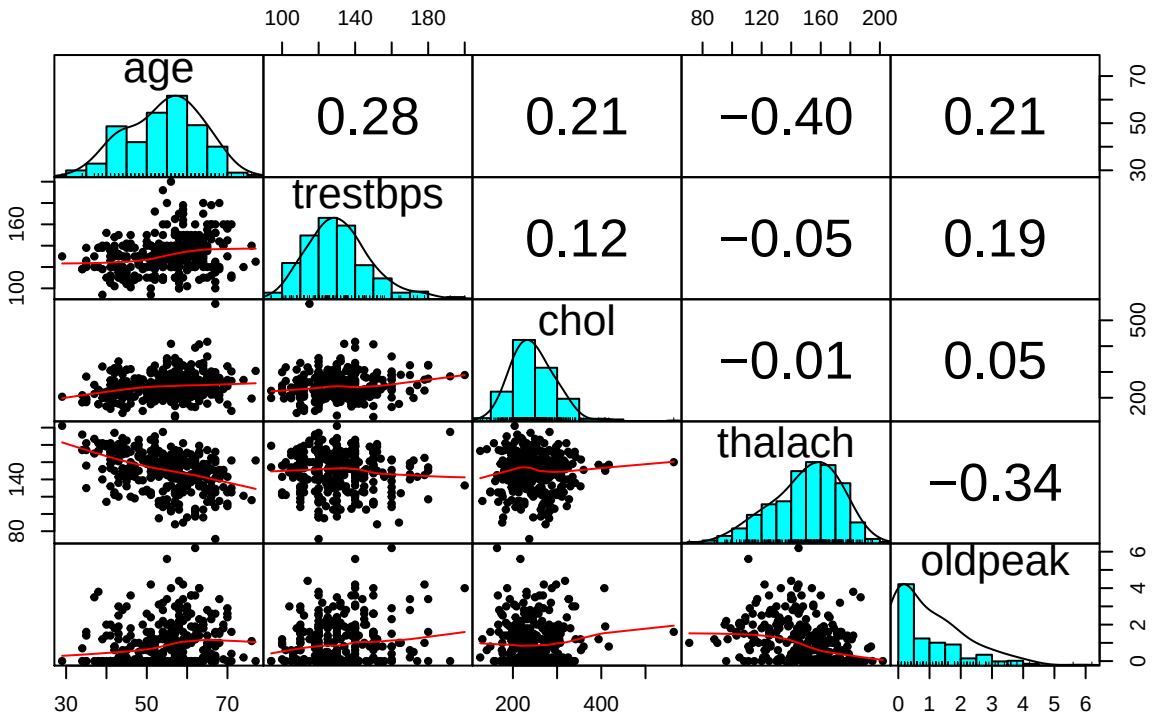
```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
## The following object is masked from 'package:gamlss':
##
##     cs
```

```
pairs.panels(heart_sub, main= "Original space-Bivariate scatter plots",
             ellipses = FALSE, gap = 0)
```



This is a preliminary analysis of the data in the original space, which aims 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 CA could be useful or not. Specifically, if we look at the plot, we can see that there some variables has no correlation to each other as their correlation is less than 0.1 and is close 0 like correlation between trestbps and thalsch (-0.05) and chol and thalach (0.01) but some variables are quite related. In particular, the variables age and thalach are the most positively correlated (0.40) and the variables thalach and oldpeak are the most negatively correlated (-0.34). According to this result, the Principal Component analysis seems justified. This means that a PCA in this dataset could be really 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 bimodal, 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(heart_sub, 2, mean)
```

```
##        age   trestbps       chol    thalach    oldpeak
##  54.366337 131.623762 246.264026 149.646865   1.039604
```

```r
apply(heart_sub, 2, var)
```

```
##         age    trestbps        chol     thalach     oldpeak
##   82.484558  307.586453 2686.426748  524.646406    1.348095
```

The variables are very different from each other. In order to work on homogeneous variables, it is better to standardize the variable in order to have zero mean and unitary variance.

```r
scaled_heart<-apply(heart_sub, 2, scale)
head(scaled_heart)
```

```
##             age    trestbps        chol     thalach    oldpeak
## [1,]  0.9506240  0.76269408 -0.25591036  0.01541728  1.0855423
## [2,] -1.9121497 -0.09258463  0.07208025  1.63077374  2.1190672
## [3,] -1.4717230 -0.09258463 -0.81542377  0.97589950  0.3103986
## [4,]  0.1798773 -0.66277043 -0.19802967  1.23784920 -0.2063639
## [5,]  0.2899839 -0.66277043  2.07861109  0.58297496 -0.3786180
## [6,]  0.2899839  0.47760118 -1.04694656 -0.07189928 -0.5508722
```

Computing PCs

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

```r
heart_cov <- cov(scaled_heart)
heart_eigen<-eigen(heart_cov)
heart_eigen$value
```

```
## [1] 1.8065754 1.0775475 0.8833766 0.7591714 0.4733290
```

As an example the eigen vectors of the first two PCs are shown:

```r
phi <- heart_eigen$vectors[,1:3]
phi <- -phi
row.names(phi) <- c("age", "trestbps", "chol", "thalach", "oldpeak")
colnames(phi) <- c("PC1", "PC2", "PC3")
phi
```

```
##                 PC1        PC2        PC3
## age       0.5673990  0.1084469 -0.1783796
## trestbps  0.3758463  0.4231668  0.7394680
## chol      0.2453800  0.6922836 -0.5317555
## thalach  -0.5056805  0.4824636  0.2983680
## oldpeak   0.4699721 -0.3116750  0.2226667
```

By examining the loading we note that first loading vector phi 1 places most of its weight on age(0.567) and much less weight on thalach(-0.505). The second loading vector phi 2 places most of its weight on chol (0.692) and much less weight on oldpeak(-0.311).

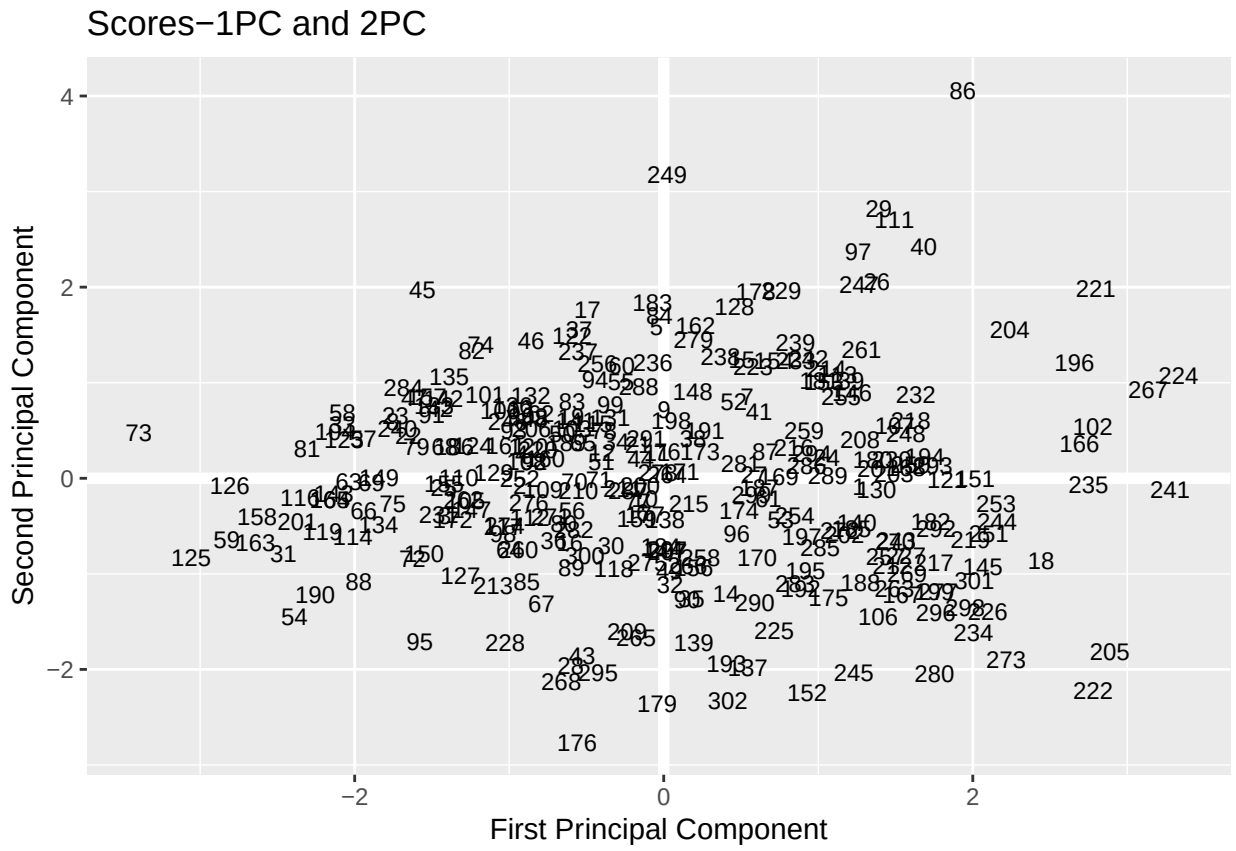
principal component scores

```r
PC1 <- scaled_heart %*% phi[,1]
PC2 <- scaled_heart %*% phi[,2]
PC <- data.frame(ID = row.names(heart), PC1, PC2)
head(PC)
```

```
##   ID         PC1         PC2
## 1  1  1.26562200 -0.08222156
```

```
## 2  2 -0.93081032 -0.07031681
## 3  3 -1.41755513 -0.38919460
## 4  4 -0.91857160  0.26348764
## 5  5 -0.04725251  1.58924385
## 6  6 -0.13539506 -0.35422683
```

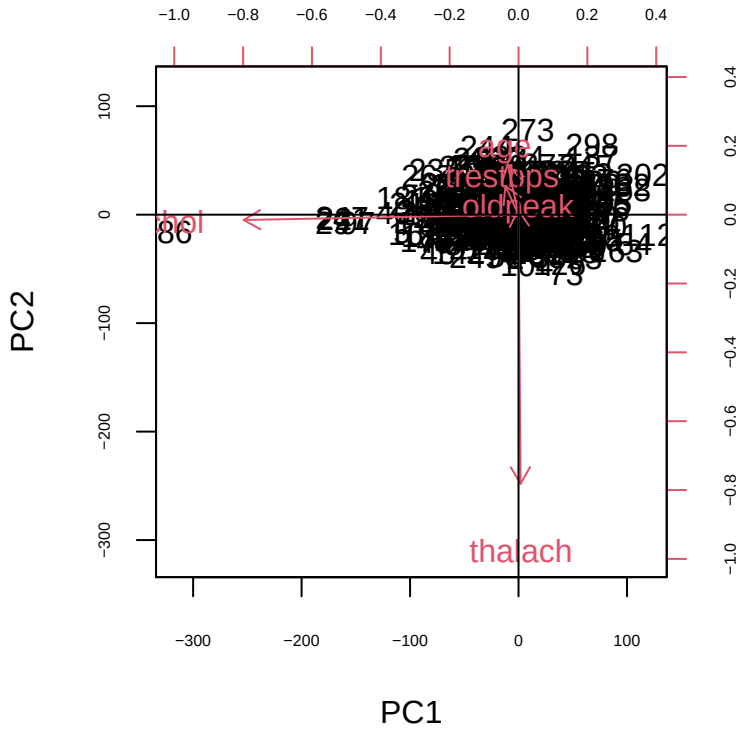
```
library(ggplot2)
library(modelr)
ggplot(PC, aes(PC1, PC2)) +
modelr::geom_ref_line(h = 0) +
modelr::geom_ref_line(v = 0) +
geom_text(aes(label = ID), size = 3) +
xlab("First Principal Component") +
ylab("Second Principal Component") +
ggtitle("Scores-1PC and 2PC")
```



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,

```
set.seed(123)
heart_pc <- prcomp(heart_sub, center = TRUE, scale. = FALSE)
biplot(heart_pc, cex.axis = 0.5, scale=0)
abline(h=0)
abline(v=0)
```



The angle between the arrows gives information on the correlation between the two variables.For example thalach and oldpeak are negatively correlated as their degree is 180.

```
cor(heart_sub)
```

```
##                 age    trestbps         chol      thalach     oldpeak
## age       1.0000000  0.27935091  0.213677957 -0.398521938  0.21001257
## trestbps  0.2793509  1.00000000  0.123174207 -0.046697728  0.19321647
## chol      0.2136780  0.12317421  1.000000000 -0.009939839  0.05395192
## thalach  -0.3985219 -0.04669773 -0.009939839  1.000000000 -0.34418695
## oldpeak   0.2100126  0.19321647  0.053951920 -0.344186948  1.00000000
```

To select the number of principal components, three heuristic methods are proposed.

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 <- heart_eigen$values/sum(heart_eigen$values))
```

```
## [1] 0.36131509 0.21550951 0.17667531 0.15183429 0.09466581
```

- The first PC explains 36.13% of the variability;
- The second PC explains 21.55% of the variability;
- The third PC explains 17.66% of the variability;
- The fourth PC explains 15.18% of the variability;
- The fifth PC explains 9.46% of the variables;

```
cumsum(PVE)
```

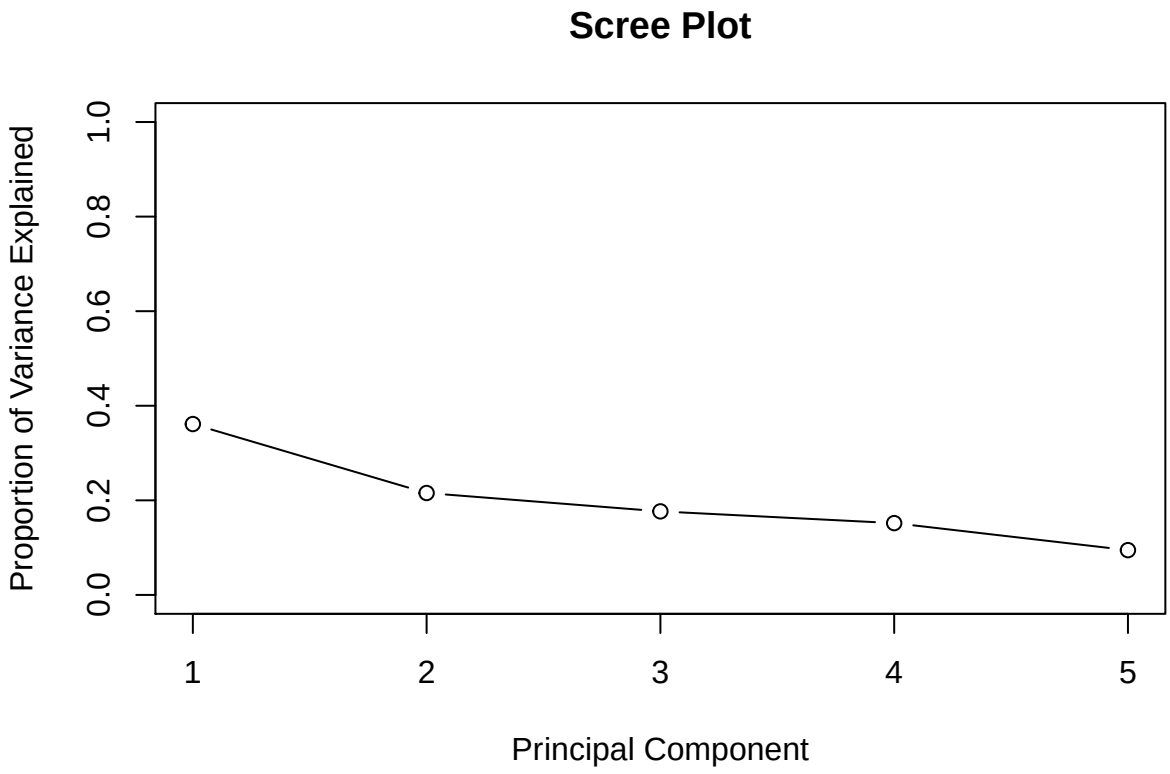
```
## [1] 0.3613151 0.5768246 0.7534999 0.9053342 1.0000000
```

According to the CPVE we have to retain as many PCs as needed to explain at least the 80% of the total variance, hence we have to retain the first 4 PCs.

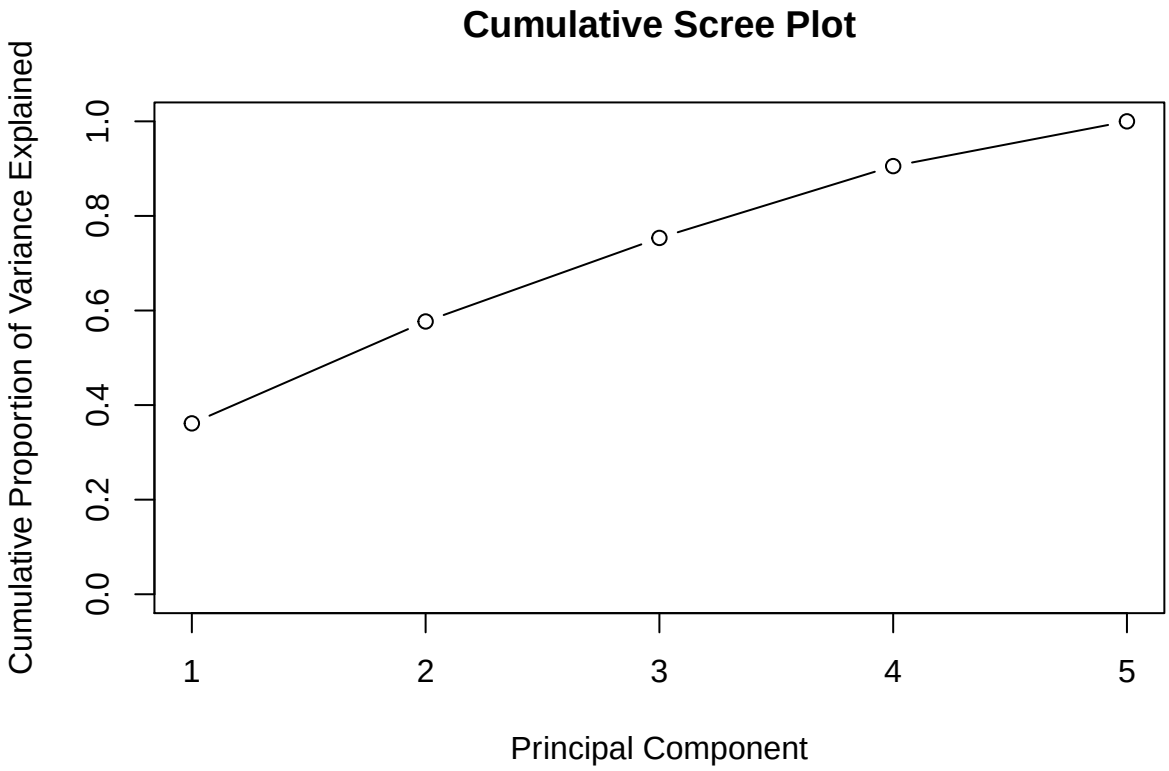
Scree plot

The scree plot suggests selecting q corresponding to the value of m where the curve becomes flat.

```
plot(PVE, xlab="Principal Component", ylab="Proportion of Variance Explained", main="Scree Plot", ylim=
```




```
plot(cumsum(PVE), xlab="Principal Component", main="Cumulative Scree Plot", ylab="Cumulative Proportion
```



In the "Proportion of variance Explained" plot, the elbow point is not so clear and it may be at q=3 or q=4. However, according to this method, it seems reasonable to retain the first three principal components.

Kaiser's rule

According to the Kaiser's rule, for standardized data, the principal components with the variance greater than 1 are taken.

```
heart_eigen$values
```

```
## [1] 1.8065754 1.0775475 0.8833766 0.7591714 0.4733290
```

The Kaiser's rule suggests keeping the first two principal components.

PCA result

Based on different strategies I obtained different results.The cumulative PVE rule suggests retaining the first four principal components, the Kaiser's rule suggests retaining the first two principal components while the Scree plot doesn't provide a clear result. I decided to choose CPVE results as it will obtain more PCs.

Cluster Analysis

The purpose of the clustering analysis is to identify patterns of similar units within the heart dataset.

Hopkins statistic

```
heart_scale <- scale(heart_sub)
```

```
library(clustertend)
hopkins(heart_scale, n = nrow(heart_scale)-1)
```

```
## $H
## [1] 0.2520289
```

The Hopkins statistics value is close to 0. The result indicates clustered data, under the assumption that the configuration without cluster is the uniform distribution.

Computing Euclidean distance

```
dist.eucl <- dist(heart_scale, method = "euclidean")
eucl <- round(as.matrix(dist.eucl)[1:5,1:5],2)
row.names(eucl) <- c("age", "trestbps", "chol", "thalach", "oldpeak")
colnames(eucl) <- c("age", "trestbps", "chol", "thalach", "oldpeak")
eucl
```

```
##           age trestbps chol thalach oldpeak
## age      0.00     3.57 2.90    2.41    3.22
## trestbps 3.57     0.00 2.16    3.22    4.07
## chol     2.90     2.16 0.00    1.94    3.53
## thalach  2.41     3.22 1.94    0.00    2.38
## oldpeak  3.22     4.07 3.53    2.38    0.00
```

Computing Manhattan distance

```
dist.man <- dist(heart_scale, method = "manhattan")
man <- round(as.matrix(dist.eucl)[1:5,1:5],2)
row.names(man) <- c("age", "trestbps", "chol", "thalach", "oldpeak")
colnames(man) <- c("age", "trestbps", "chol", "thalach", "oldpeak")
man
```

```
##           age trestbps chol thalach oldpeak
## age      0.00     3.57 2.90    2.41    3.22
## trestbps 3.57     0.00 2.16    3.22    4.07
## chol     2.90     2.16 0.00    1.94    3.53
## thalach  2.41     3.22 1.94    0.00    2.38
## oldpeak  3.22     4.07 3.53    2.38    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).

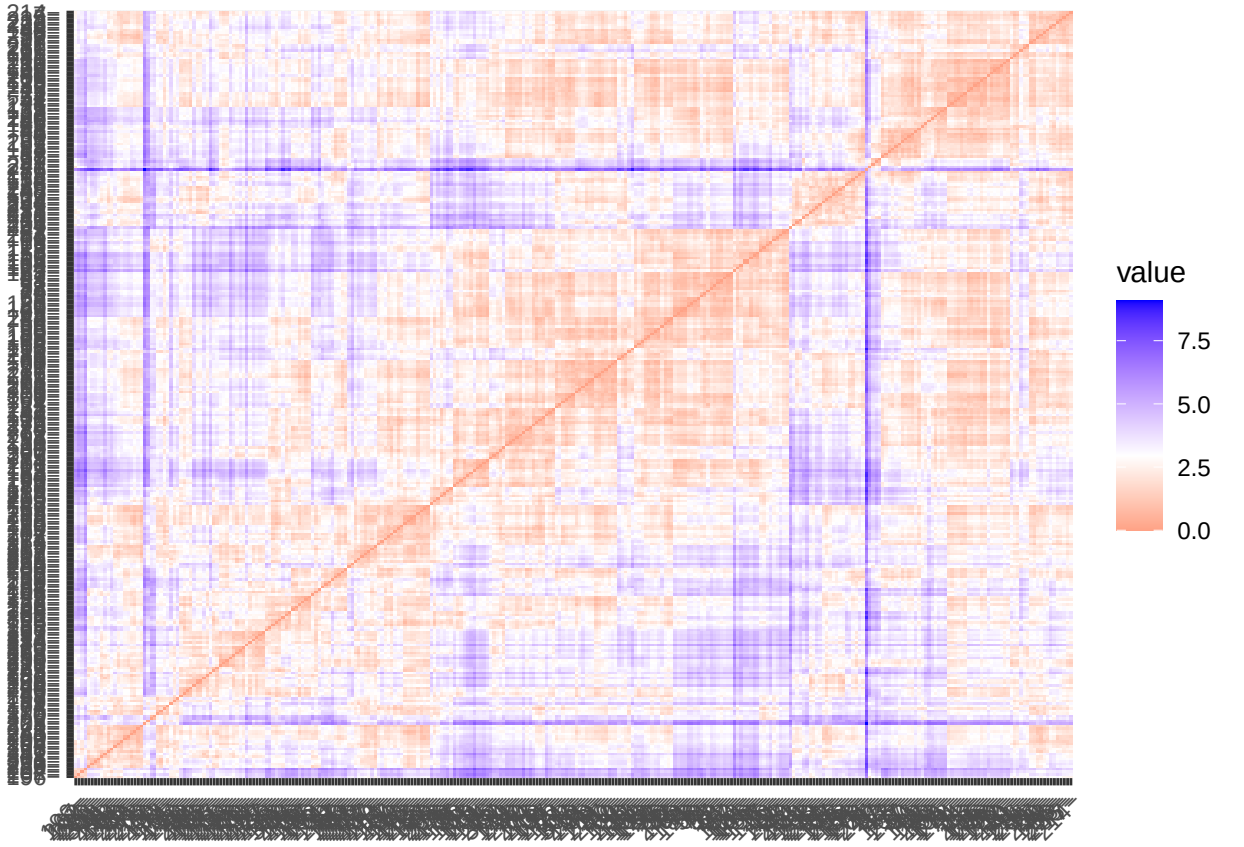
Visualizing distance matrices

Euclidean:

```
library(factoextra)
```

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

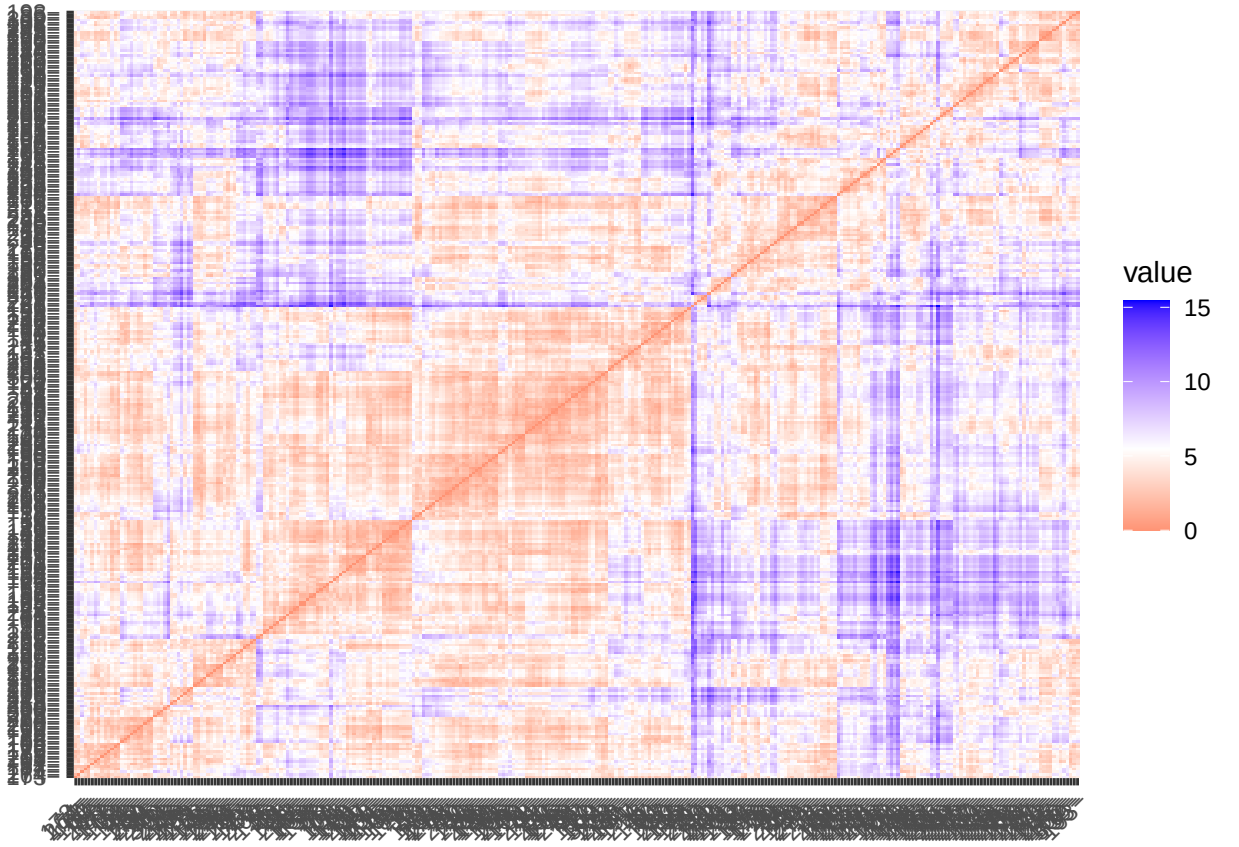
```
fviz_dist(dist.eucl)
```



The color level is proportional to the value of the dissimilarity between observations. Red indicates high similarity (i.e.: low dissimilarity) while blue indicates low similarity. According to the Visual method, that uses the Euclidean distance as distance between units, the data should not contain a noticeable clustering structure. While not sure of the presence of a clustering structure, the analysis continues.

Manhattan:

```
library(factoextra)
fviz_dist(dist.man)
```



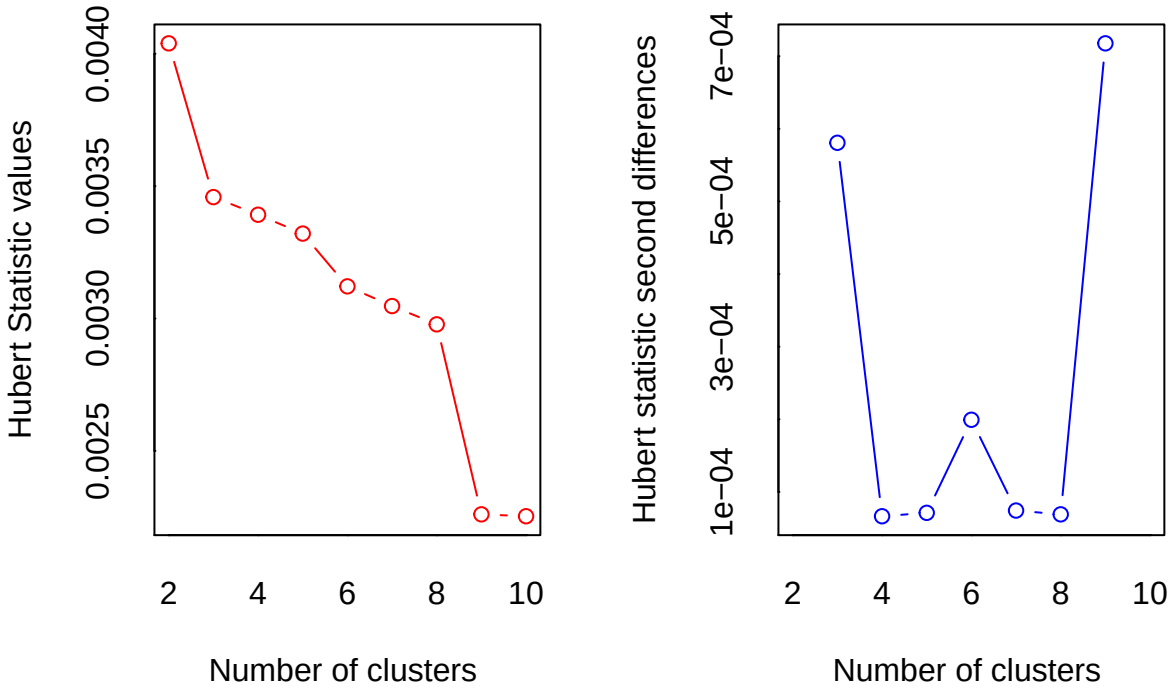
Optimal number of clusters

Using both Euclidean and Manhattan distance, according to different clustering methods, the optimal number of clusters will be computed

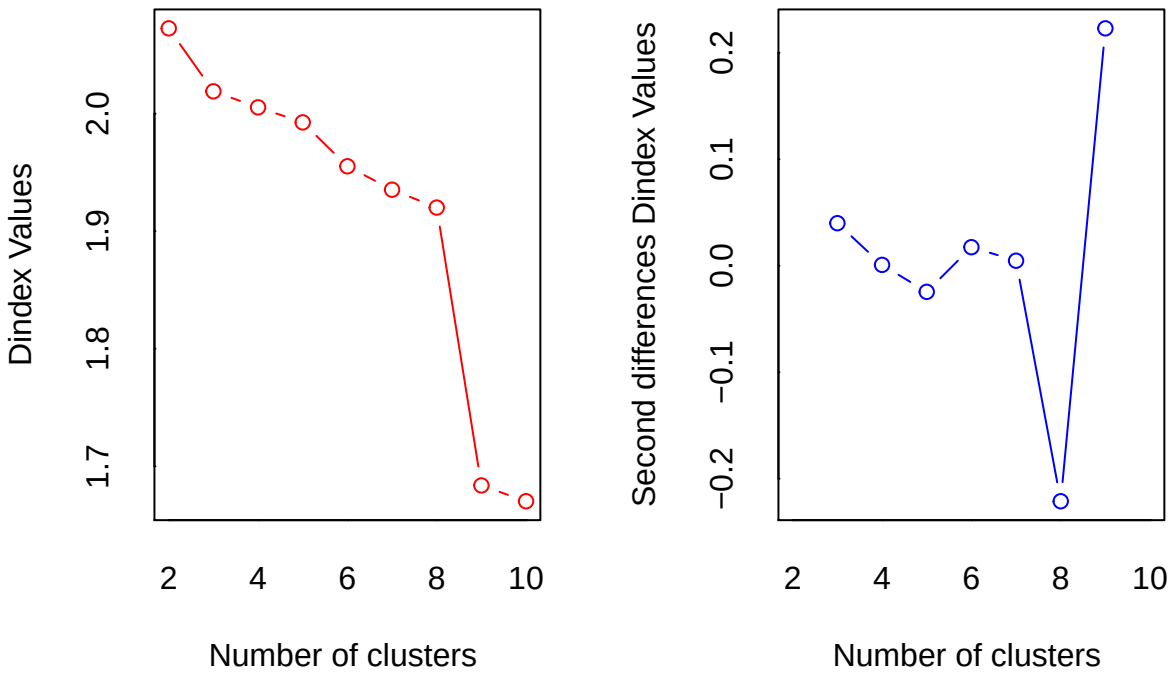
Hierarchical method

Average linkage method and Euclidean distance

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




```
## *** : 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.
##
```

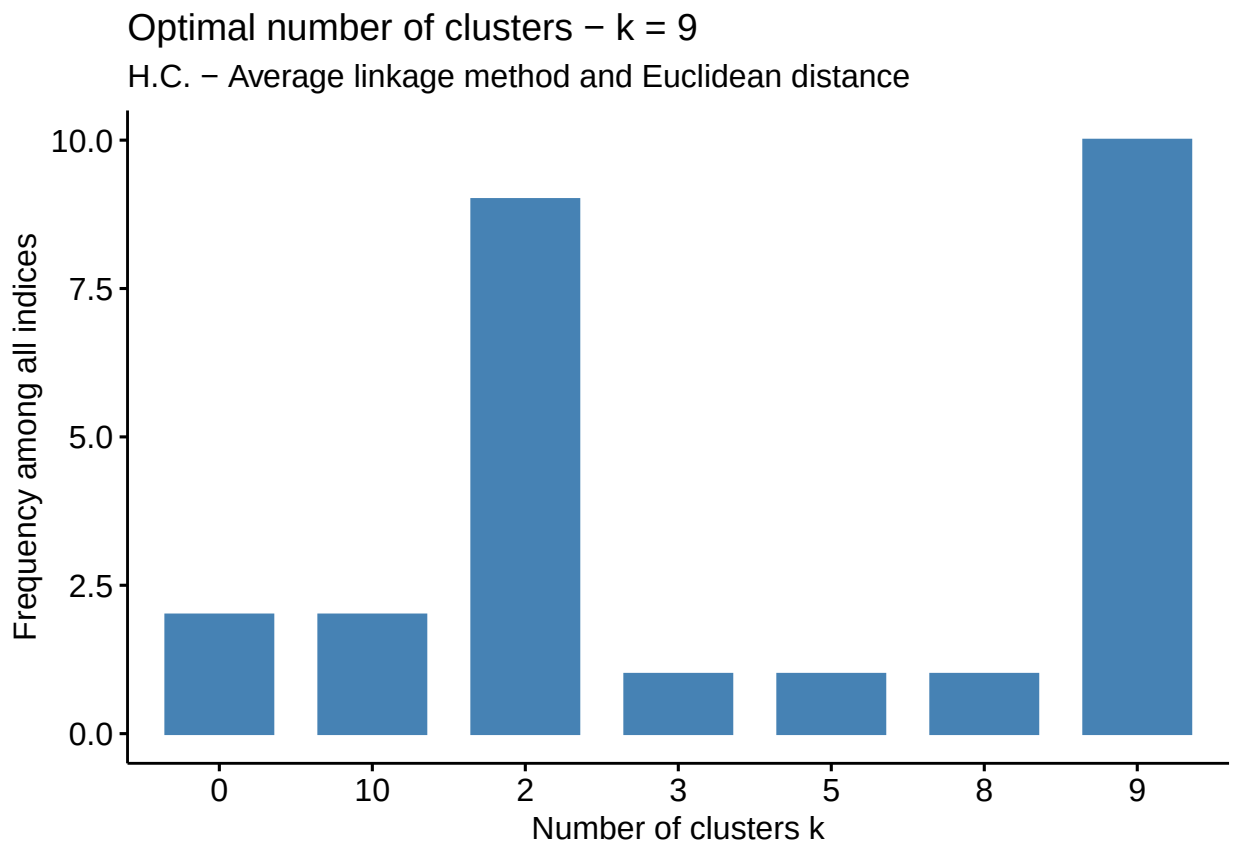



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##       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

```
library(factoextra)
fviz_nbclust(nb)+labs(subtitle = "H.C. - Average linkage method and 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
## * 1 proposed  3 as the best number of clusters
## * 1 proposed  5 as the best number of clusters
## * 1 proposed  8 as the best number of clusters
## * 10 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  9 .
```



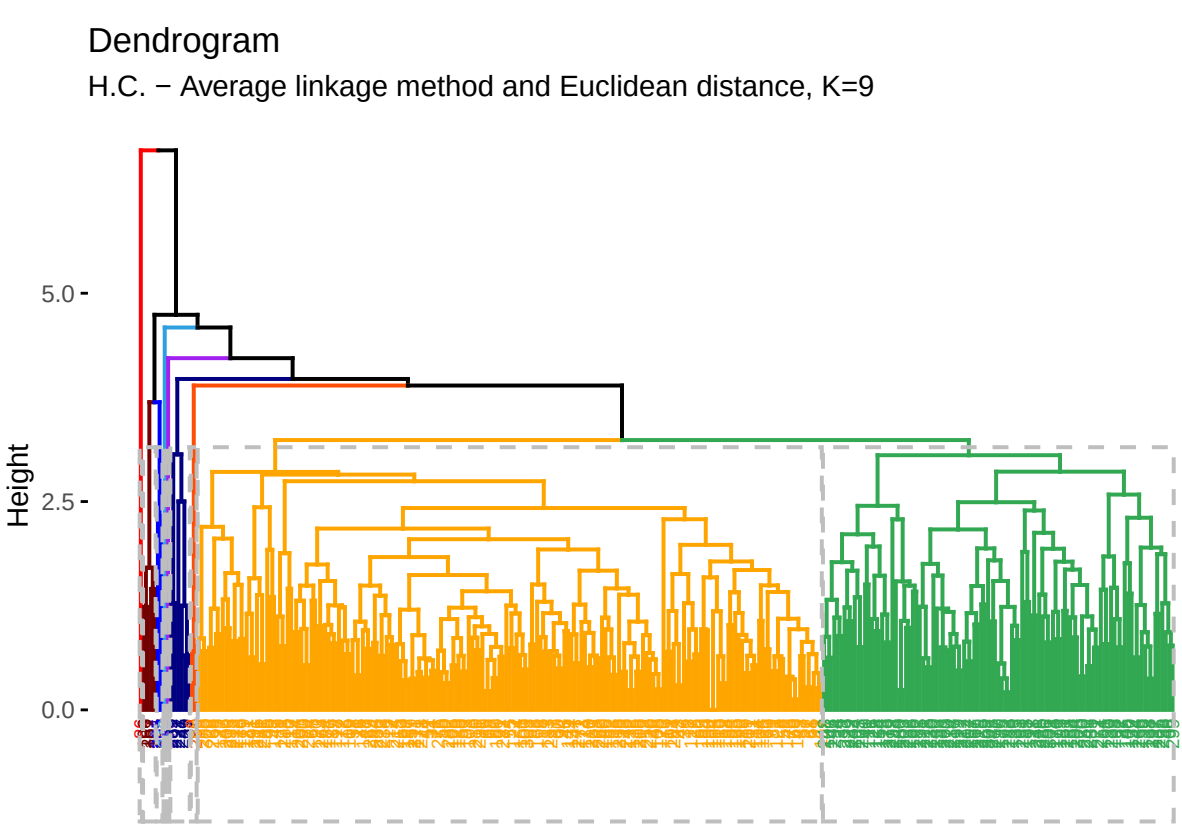

```r
hc <- hclust(dist.eucl, method = "average")
grp <- cutree(hc, k=9)
table(grp)
```

```
## grp
##   1   2   3   4   5   6   7   8   9
## 103   2 183   6   1   4   2   1   1
```

```
grp
```

```
##   [1] 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 3 3 3 1 3 1 3 3 4 3 3 1 3 3 1 3 3
##  [38] 3 1 1 3 3 3 3 3 3 3 3 1 3 3 1 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3
##  [75] 3 3 3 3 3 3 3 3 3 3 3 5 1 3 3 3 3 3 3 3 3 1 4 3 3 3 3 6 3 3 3 1 1 3 3 3 1
## [112] 3 1 3 3 3 3 3 3 3 1 3 3 3 3 3 3 1 3 1 3 3 3 3 3 3 1 1 1 1 3 3 3 1 1 1 3 3
## [149] 3 3 1 1 1 1 3 1 3 3 3 3 3 3 3 3 3 1 1 1 3 1 3 3 3 1 1 3 3 1 3 1 3 1 3 3 1
## [186] 3 3 1 3 3 3 1 1 1 1 6 1 1 1 3 3 1 1 1 7 3 3 1 3 3 3 1 3 3 3 3 1 1 1 3 4 7
## [223] 3 6 1 1 1 3 1 1 3 1 1 1 1 3 3 3 1 3 1 1 1 1 1 3 4 1 8 1 1 3 1 1 1 3 1 1 3
## [260] 2 1 3 1 3 1 1 6 3 1 1 3 1 9 3 3 3 1 3 1 1 3 3 1 3 1 3 3 3 4 1 3 4 1 1 3 1
## [297] 1 1 1 3 1 1 3
```

```
fviz_dend(hc, k = 9, cex = 0.5, k_colors = c(
  "red", "#720000", "blue", "#2E9FDF", "purple", "navy", "#FC4E07", "orange", "#32a852"), color_labels_
```

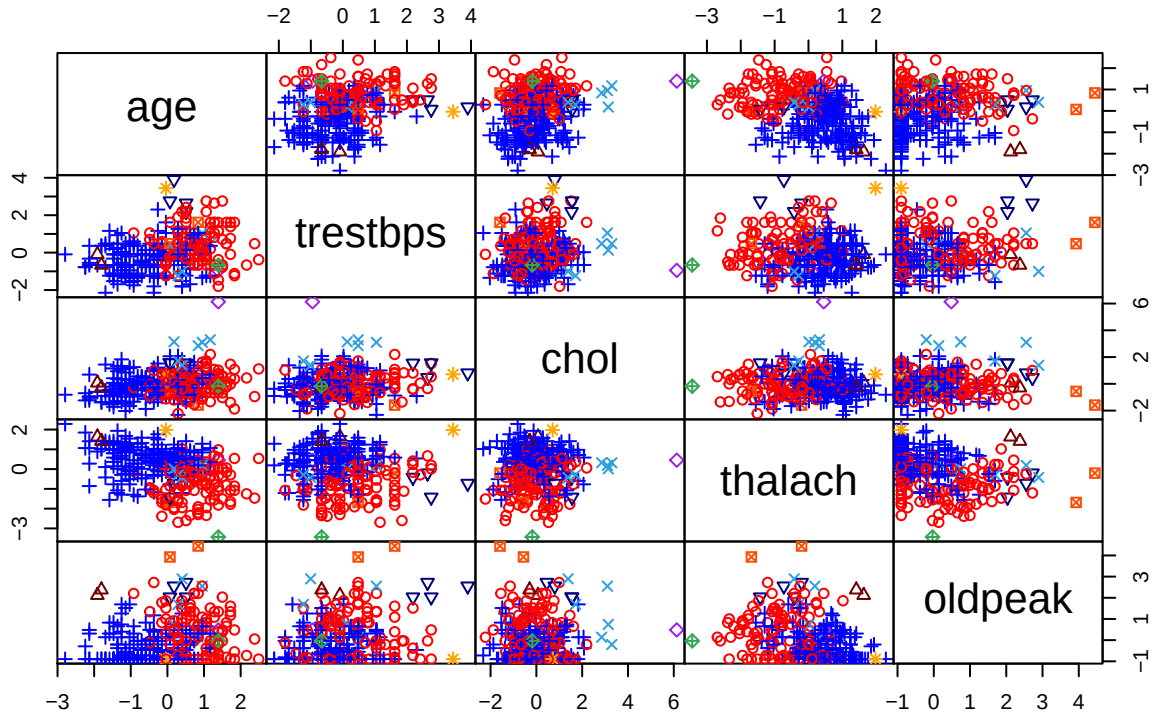



```
cor(dist.eucl, cophenetic(hc))
```

```
## [1] 0.7016001
```

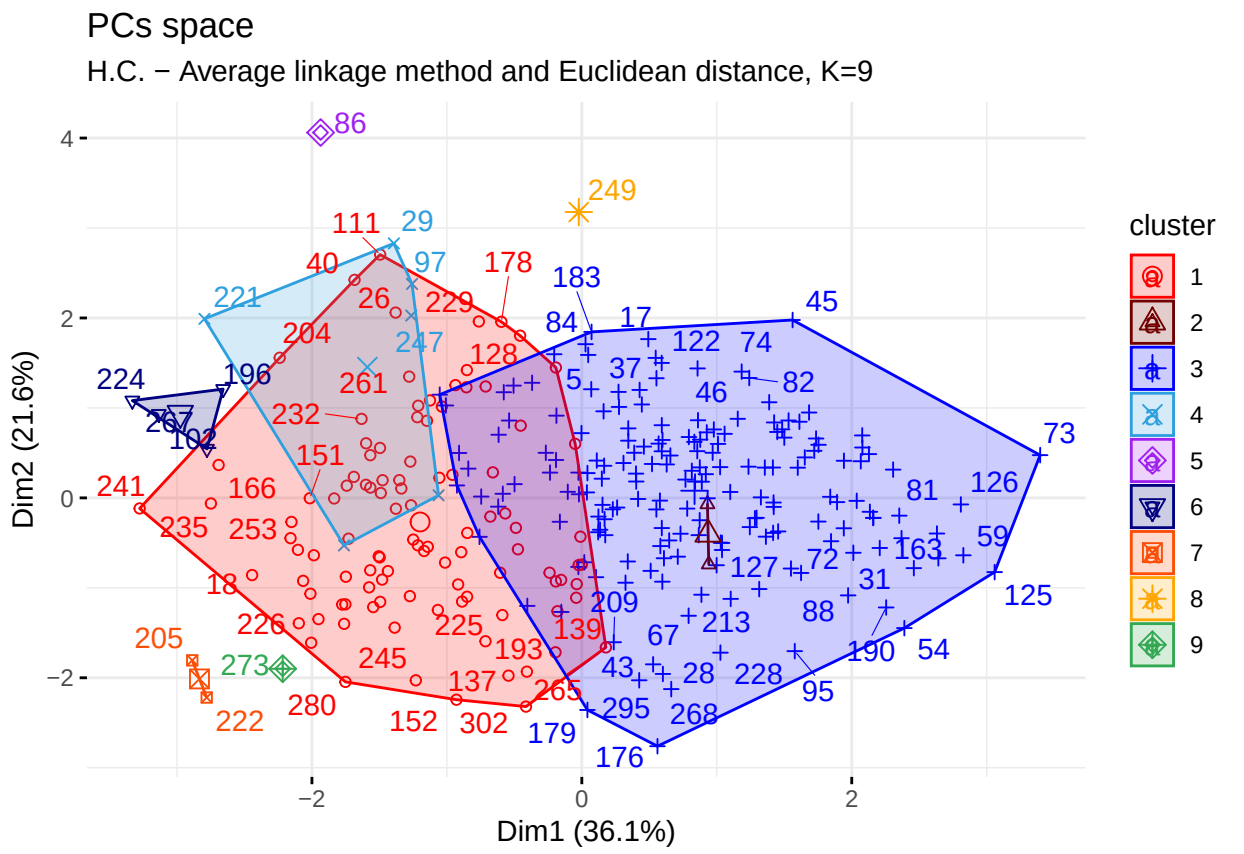
According to the result of "NbClust", the best number of clusters, applying hierarchical clustering method and using average linkage method and Euclidean distance is 9.

```
pairs(heart_scale, gap=0, pch=grp, cex.main= 0.7, main="Original space\nH.C.-Average linkage method and
```




```
fviz_cluster(list(data = heart_scale, cluster = grp), palette = c("red", "#720000", "blue", "#2E9FDF",
             show.clust.aver = FALSE, ggtheme = theme_minimal())+
  labs(subtitle = "H.C. - Average linkage method and Euclidean distance, K=9", cex.sub= 0.5)
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Internal validation measures: silhouette width and Dunn index

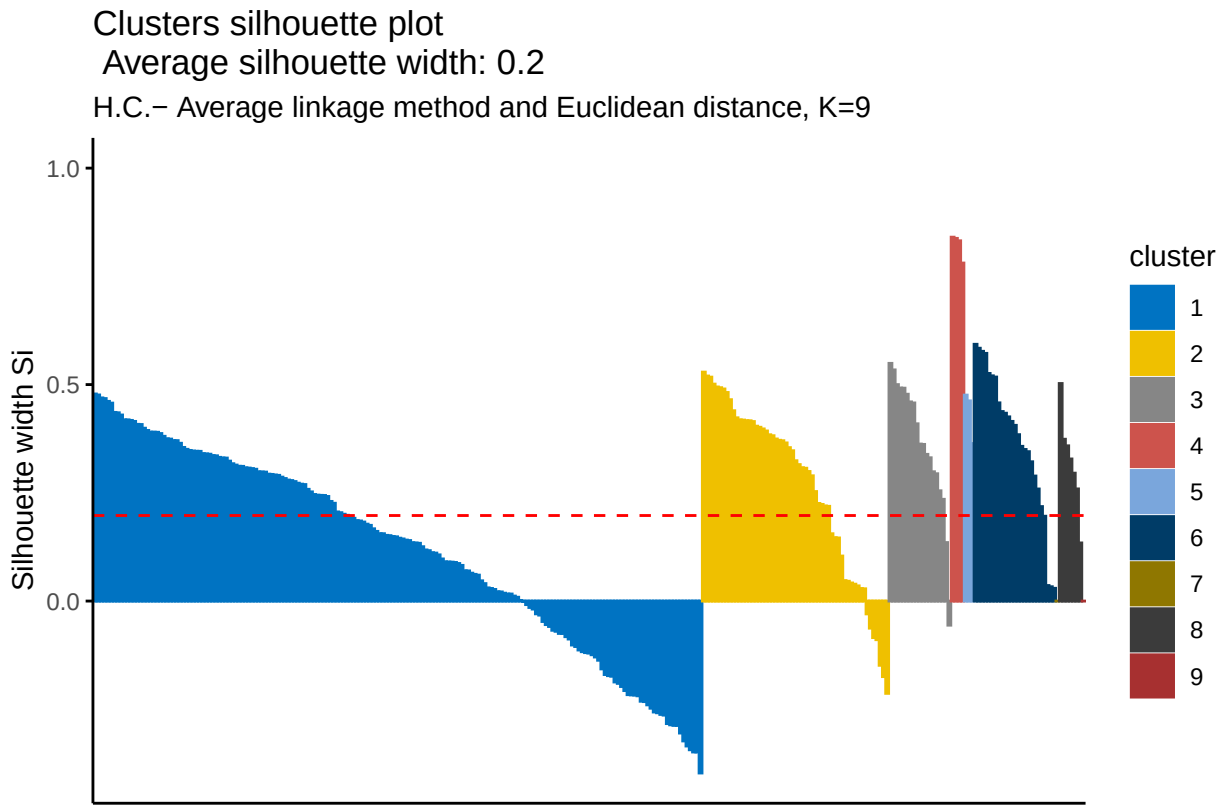
```
hclust<- eclust(heart_sub,k=9 ,"hclust",hc_method  = "average",nboot = 50, hc_metric = "euclidean")
silinfo <- hclust$silinfo
silinfo$avg.width
```

Silhouette width

```
## [1] 0.1975733
```

```
fviz_silhouette(hclust, palette = "jco", ggtheme = theme_classic())+
  labs(subtitle = "H.C.- Average linkage method and Euclidean distance, K=9", cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  186          0.12
## 2       2   57          0.26
## 3       3   19          0.36
## 4       4    4          0.82
## 5       5    3          0.43
## 6       6   25          0.37
## 7       7    1          0.00
## 8       8    7          0.32
## 9       9    1          0.00
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.1173228 0.2628790 0.3646535 0.8227958 0.4340081 0.3713990 0.0000000
## [8] 0.3217085 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
## 175       1        3 -0.0009350035
## 136       1        2 -0.0082096896
## 182       1        6 -0.0148114264
## 168       1        8 -0.0185417439
## 18        1        6 -0.0295075498
## 277       1        6 -0.0324271348
## 288       1        8 -0.0472231431
## 96        1        6 -0.0543870764
## 290       1        3 -0.0587313094
## 9         1        8 -0.0674766192
## 64        1        3 -0.0699145521
## 202       1        6 -0.0747718547
## 132       1        2 -0.0748892520
## 106       1        3 -0.0822816159
## 185       1        8 -0.0877951698
## 259       1        8 -0.1015596031
## 44        1        6 -0.1051059073
```

```
## 34        1        2 -0.1130718197
## 278       1        6 -0.1169629231
## 299       1        6 -0.1190982726
## 6         1        3 -0.1202364640
## 24        1        8 -0.1240561128
## 246       1        2 -0.1292523427
## 212       1        6 -0.1367529267
## 301       1        3 -0.1567444006
## 258       1        3 -0.1697274860
## 208       1        8 -0.1723540754
## 195       1        3 -0.1737083459
## 297       1        3 -0.1869320380
## 87        1        2 -0.1901750407
## 63        1        5 -0.1974305669
## 139       1        3 -0.2067841624
## 210       1        3 -0.2162164968
## 228       1        3 -0.2168086463
## 66        1        5 -0.2171916277
## 265       1        3 -0.2188955171
## 36        1        3 -0.2313863254
## 156       1        3 -0.2322818034
## 151       1        8 -0.2397806198
## 59        1        5 -0.2475524351
## 154       1        2 -0.2563128471
## 296       1        3 -0.2580955089
## 145       1        3 -0.2619211967
## 107       1        8 -0.2634694554
## 219       1        6 -0.2830520036
## 12        1        6 -0.2859326687
## 248       1        8 -0.2872116794
## 146       1        8 -0.2874152426
## 257       1        6 -0.3048733478
## 194       1        2 -0.3231469830
## 39        1        8 -0.3348239588
## 173       1        2 -0.3434975500
## 164       1        5 -0.3485998611
## 165       1        5 -0.3485998611
## 10        1        5 -0.3971655082
## 121       2        6 -0.0296491571
## 128       2        9 -0.0625814990
## 254       2        6 -0.0843424582
## 251       2        6 -0.0892183313
## 74        2        1 -0.1483877715
## 229       2        9 -0.1745050428
## 40        2        4 -0.2131465497
## 190       3        5 -0.0556481505
```

The value of average silhouette width indicates that in average the units are well enough clustered. As in particular, in each cluster the units are on average the same silhouette value with respect to the silhouette width. 63 units are not well clustered.

```
library(fpc)
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
```

```
stats$dunn
```

Dunn index

```
## [1] 0.08031805
```

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

External validation measures: confusion matrix, correct Rand index, Meila's IV index #### Confusion matrix According to the Confusion matrix, the number of clusters is more than nominal values. The clusters found are 9 while the nominal variable can take 2 possible values.

```
table(heart$sex, hclust$cluster)
```

```
##
##       1   2   3   4   5   6   7   8   9
##   0  52  24   4   4   1   6   1   4   0
##   1 134  33  15   0   2  19   0   3   1
```

A large number of "female" sex (n = 134) has been classified in cluster 1 while cluster 5 and 8 have 0 number of values. The same happened for "male" sex (n=52) classified in cluster 1 while cluster 9 has 0 values.

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

Correct Rand Index

```
## [1] 0.03730664
```

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.

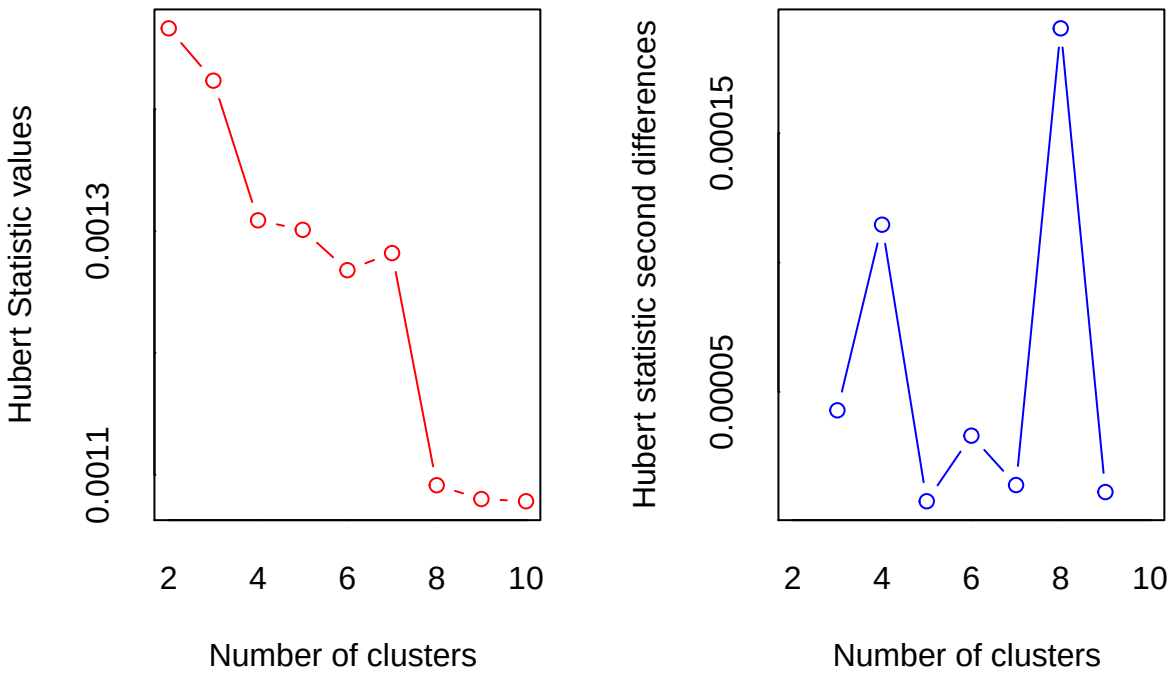
```
stats$vi
```

Meila's VI Index

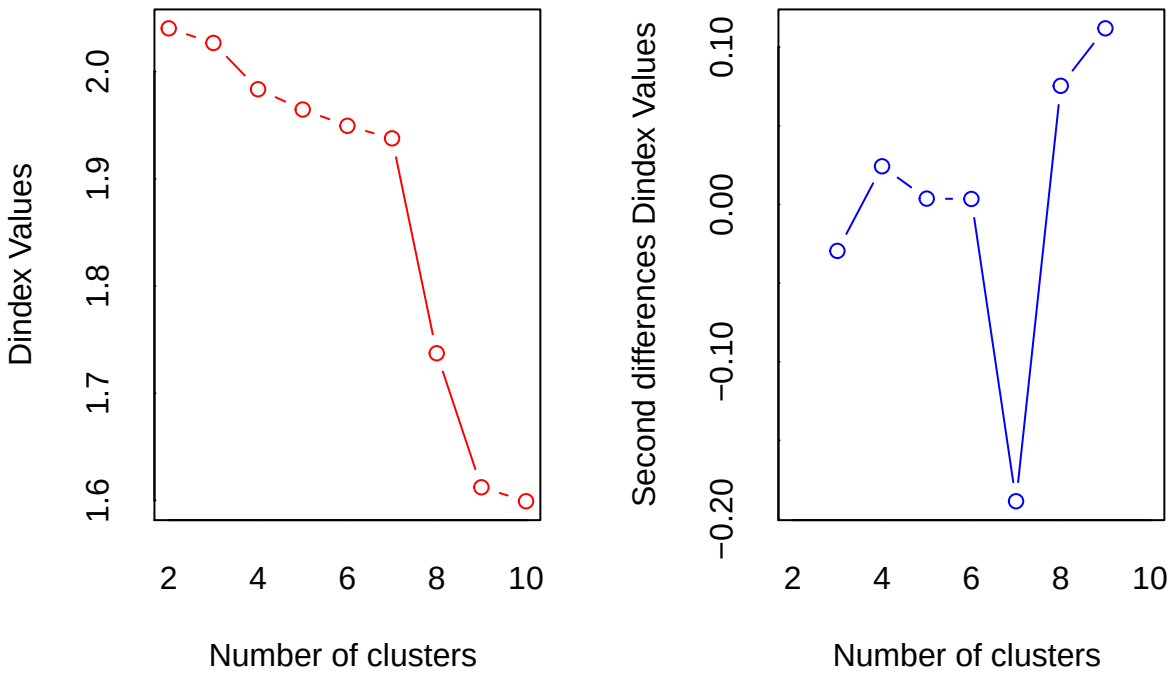
```
## [1] 1.779665
```

Average linkage method and Manhattan distance

```
nb <- NbClust(heart_scale, distance = "manhattan", min.nc = 2, max.nc = 10,
              method = "average")
```




```
## *** : 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.
##
```

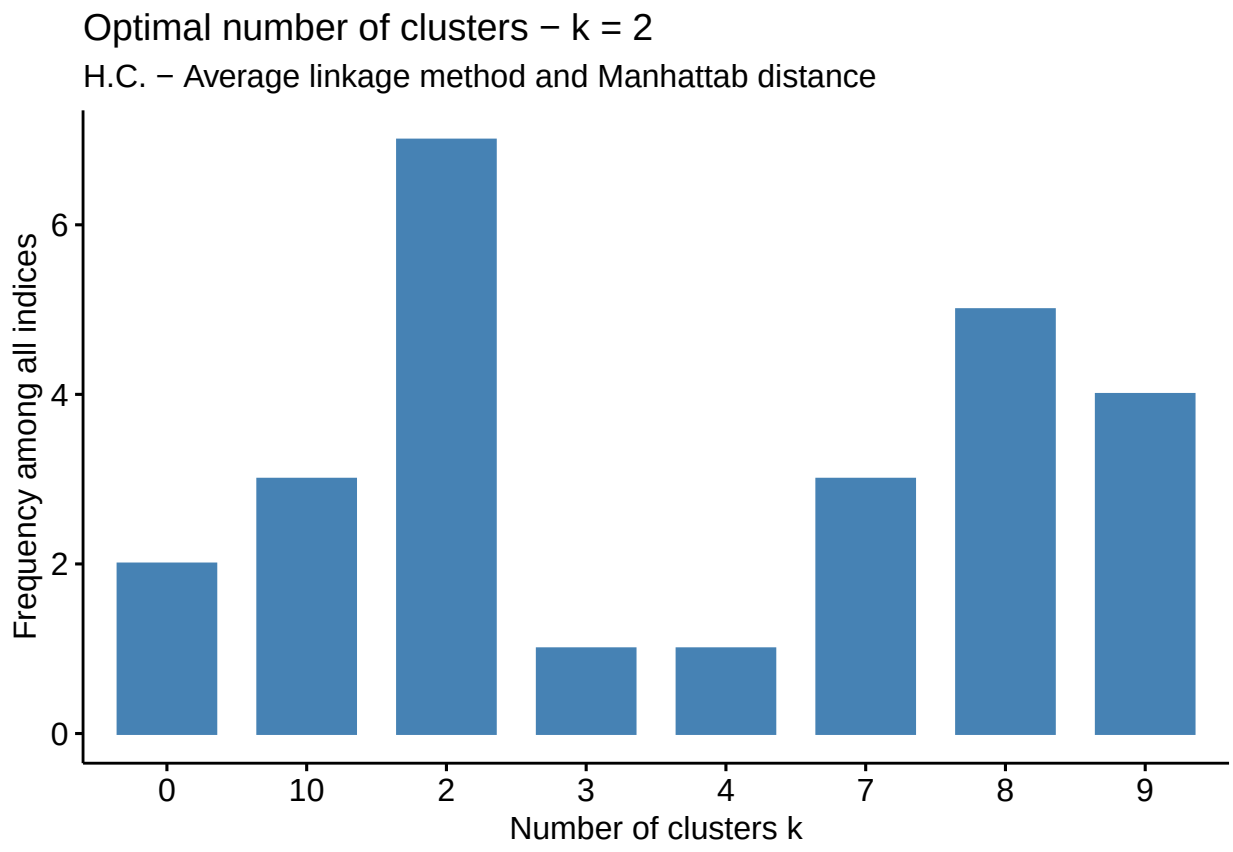



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 7 proposed 2 as the best number of clusters
## * 1 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 3 proposed 7 as the best number of clusters
## * 5 proposed 8 as the best number of clusters
## * 4 proposed 9 as the best number of clusters
## * 3 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 and 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
## * 7 proposed  2 as the best number of clusters
## * 1 proposed  3 as the best number of clusters
## * 1 proposed  4 as the best number of clusters
## * 3 proposed  7 as the best number of clusters
## * 5 proposed  8 as the best number of clusters
## * 4 proposed  9 as the best number of clusters
## * 3 proposed  10 as the best number of clusters
##
## Conclusion
## =========================
## * According to the majority rule, the best number of clusters is  2 .
```

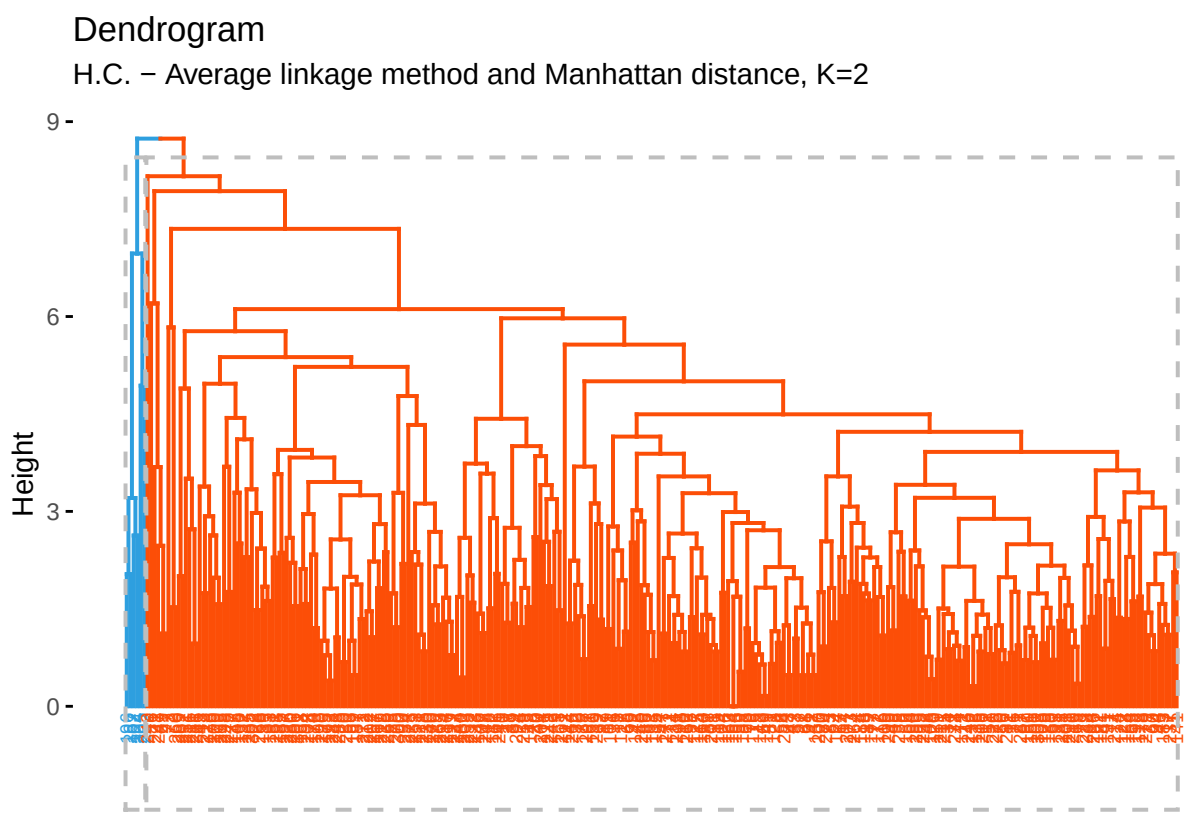



```r
dist.man <- dist(heart_scale, method = "manhattan")
hc <- hclust(dist.man, method = "average")
grp <- cutree(hc, k=2)
```

```
table(grp)
```

```
## grp
##   1   2
## 297   6
```

```
fviz_dend(hc, k = 2, cex = 0.5, k_colors = c("#2E9FDF", "#FC4E07"), color_labels_by_k = TRUE, rect = TRU
subtitle = "H.C. - Average linkage method and Manhattan distance, K=2", cex.subtitle= 0.5)
```

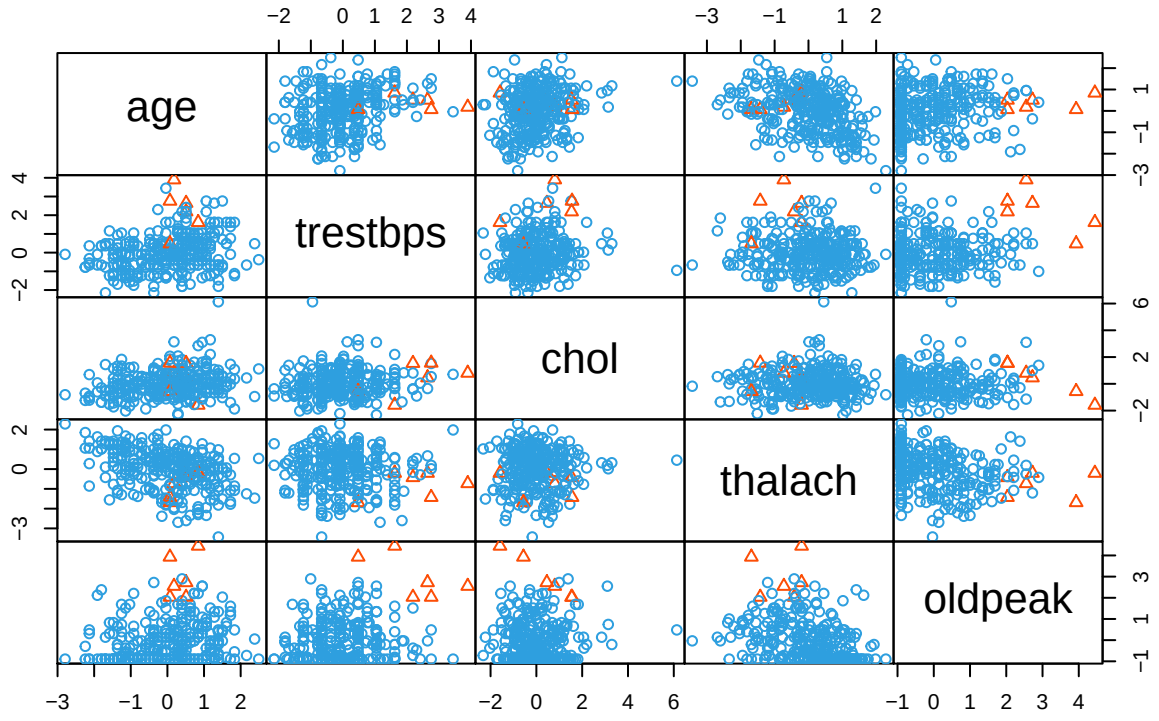



```
cor(dist.man, cophenetic(hc))
```

```
## [1] 0.6442014
```

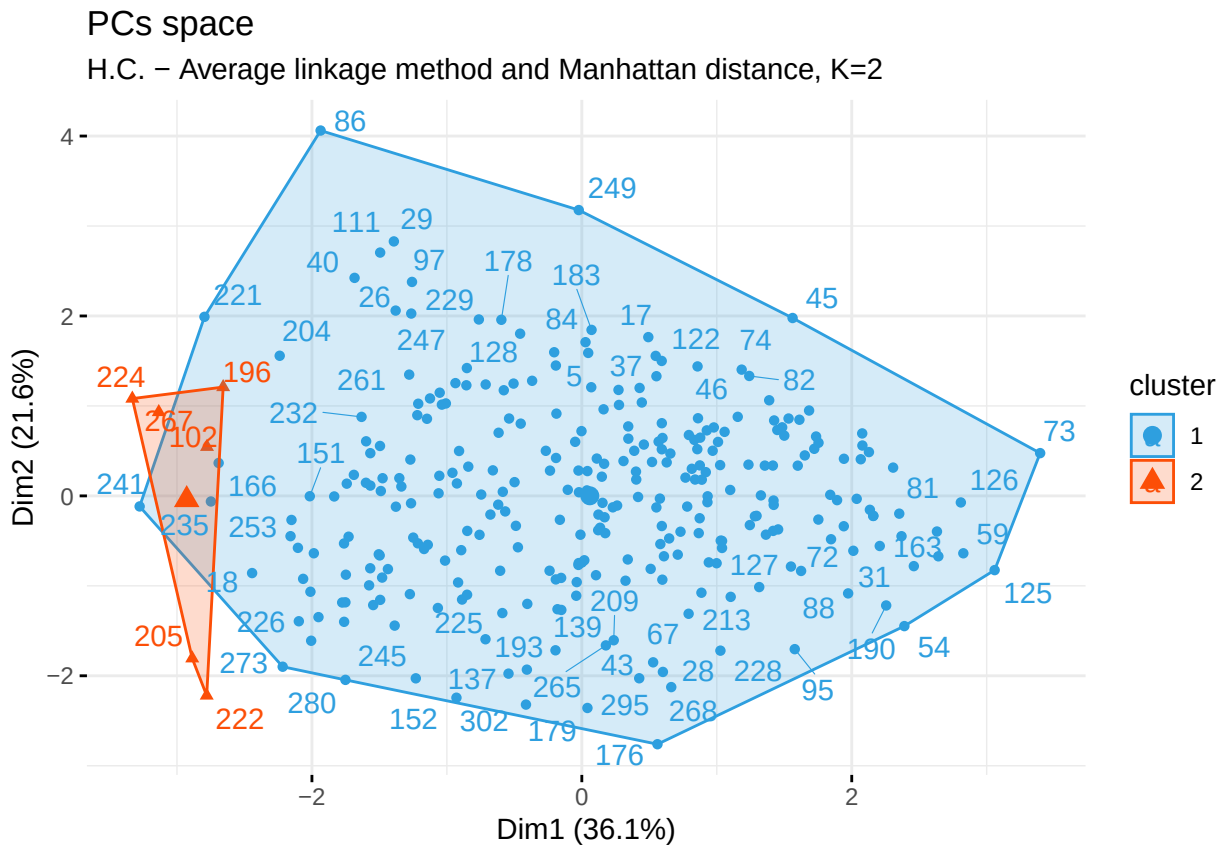
According to the result of "NbClust", the best number of clusters, applying hierarchical clustering method and using average linkage method and Manhattan distance is 2.

```
pairs(heart_scale, gap=0, pch=grp, cex.main= 0.7, main="Original space\nH.C.-Average linkage method and
```




```
fviz_cluster(list(data = heart_scale, cluster = grp), palette = c("#2E9FDF", "#FC4E07"), ellipse.type =
subtitle = "H.C. - Average linkage method and Manhattan distance, K=2", cex.sub= 0.5)
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analysed.

Internal validation measures: silhouette width and Dunn index

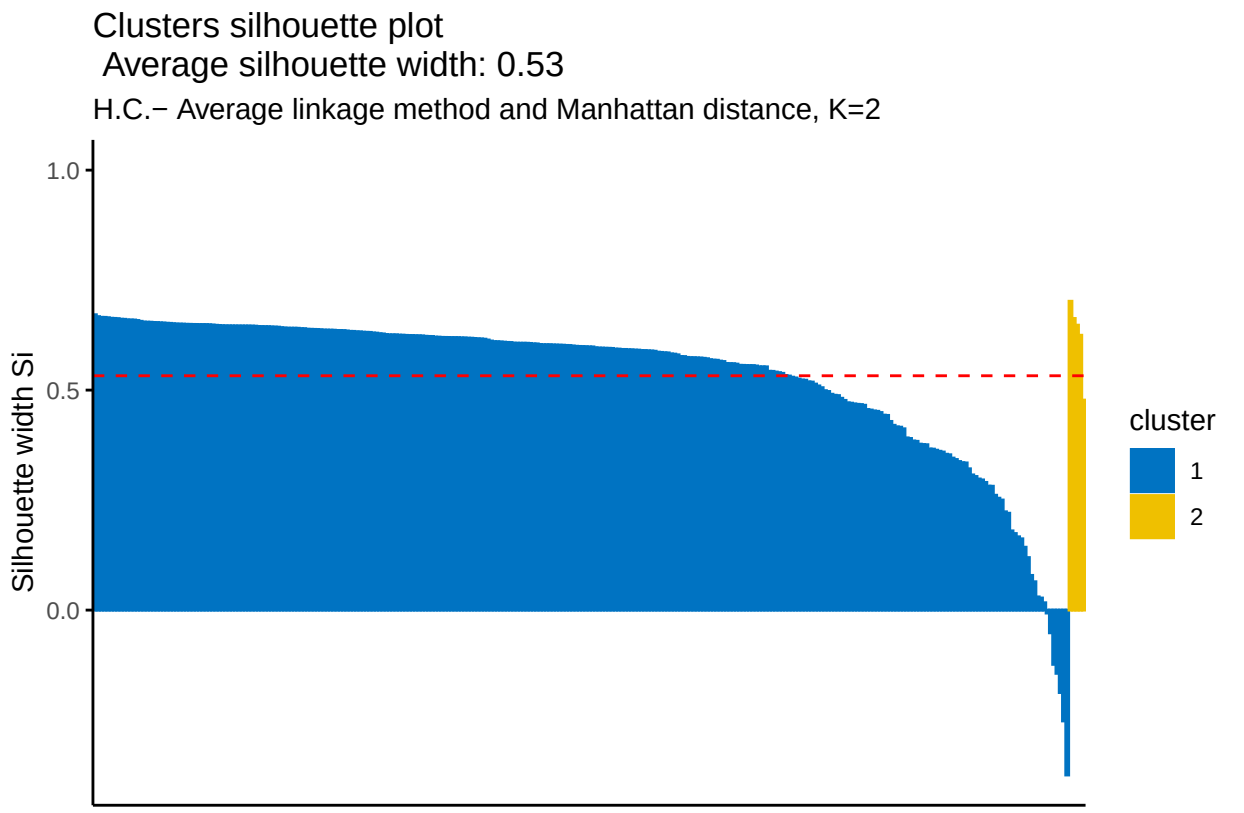
Silhouette width

```
hclust<- eclust(heart_sub,k=2 ,"hclust",hc_method  = "average",nboot = 50, hc_metric = "manhattan")
silinfo <- hclust$silinfo
silinfo$avg.width
```

```
## [1] 0.5325982
```

```
fviz_silhouette(hclust, palette = "jco", ggtheme = theme_classic())+
  labs(subtitle = "H.C.- Average linkage method and Manhattan distance, K=2", cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  298          0.53
## 2       2    5          0.62
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.5310825 0.6229340
```

```
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
## 216       1        2 -0.006733703
## 17        1        2 -0.051647410
## 162       1        2 -0.123382636
## 178       1        2 -0.143761836
## 181       1        2 -0.187398330
## 5         1        2 -0.251947295
## 40        1        2 -0.374833461
```

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) also the units are on average the same silhouette value with respect to silhouette width. According to this index, six units (216, 17, 162, 181, 5, 40)that belong to cluster 1 are not well clustered: they should belong to cluster 2.

Dunn index

```
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.1825866
```

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

External validation measures: confusion matrix, correct Rand index, Meila's

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

```
table(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  91   5
##   1 207   0
```

For "Female" sex data has been classified almost equally in each cluster. For
"male" sex (n=143) classified in cluster 2 while cluster 1 has 64 values, safe to say data are not well balanced in each cluster.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.03865368
```

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.

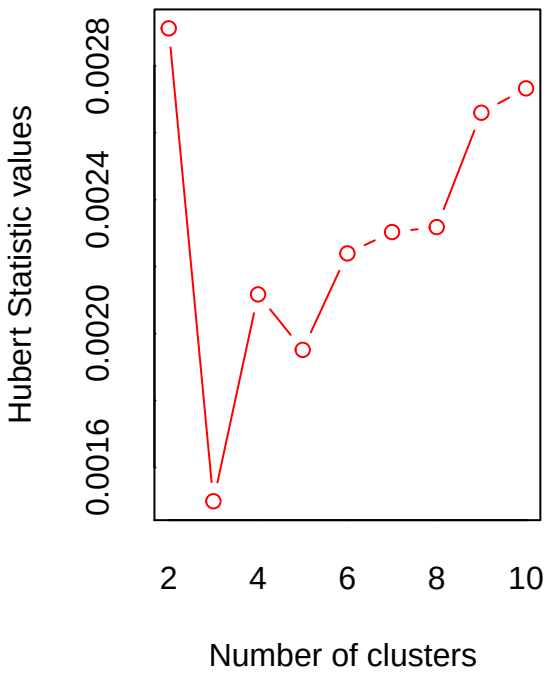
Meila's VI Index

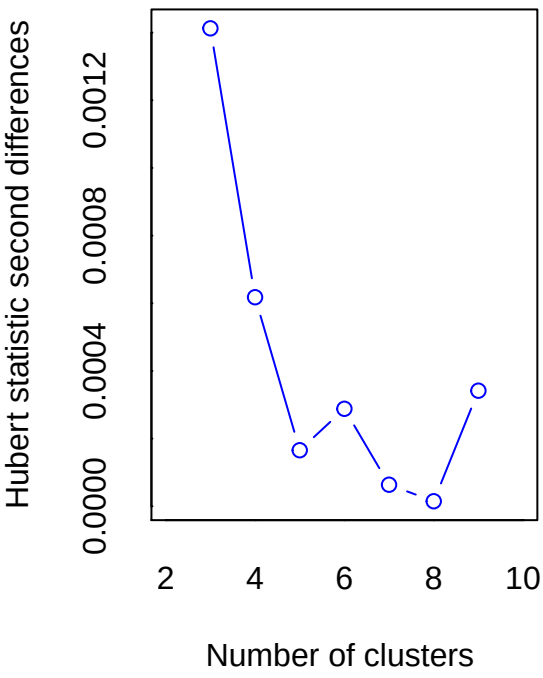
```
stats$vi
```

```
## [1] 0.6700161
```

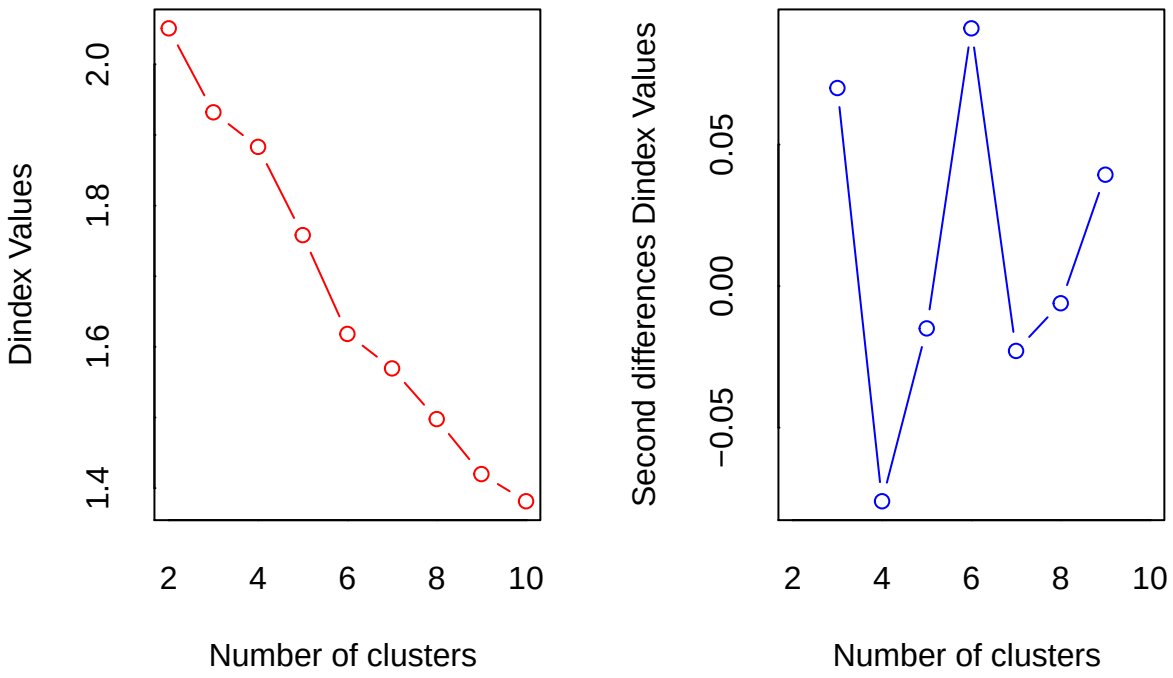
Complete linkage method and Euclidean distance

```
nb <- NbClust(heart_scale, distance = "euclidean", 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:
## * 10 proposed 2 as the best number of clusters
## * 2 proposed 3 as the best number of clusters
## * 8 proposed 6 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 1 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 and 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
```

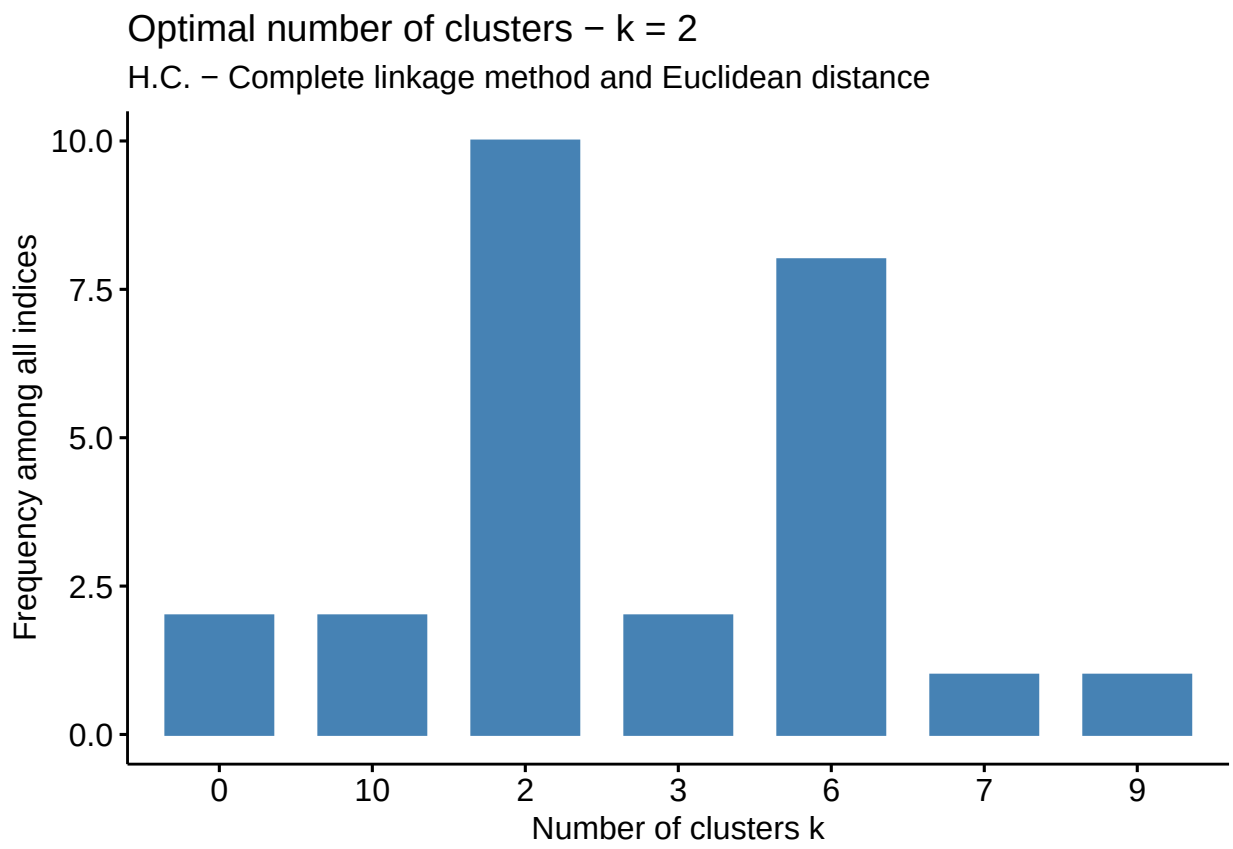
```
## will be used
```

```
## 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
## * 10 proposed  2 as the best number of clusters
## * 2 proposed  3 as the best number of clusters
## * 8 proposed  6 as the best number of clusters
## * 1 proposed  7 as the best number of clusters
## * 1 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 .
```

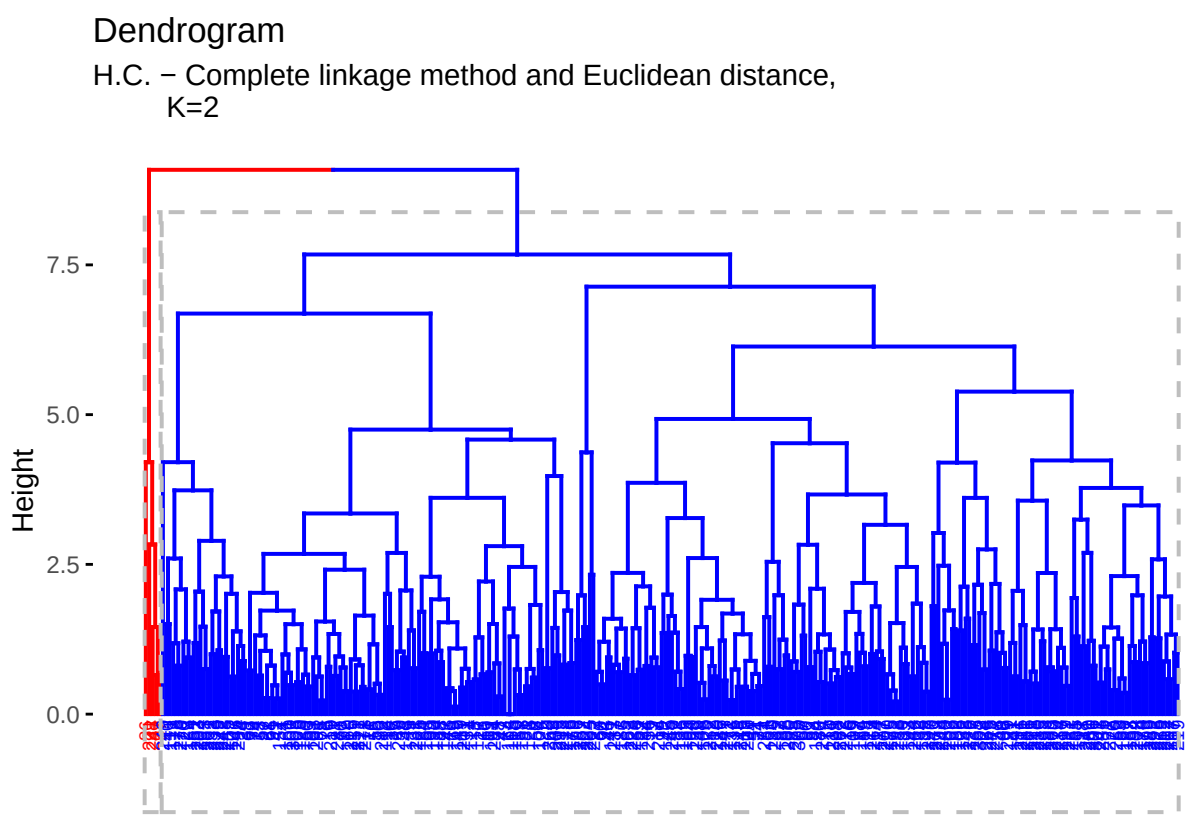



```r
hc <- hclust(dist.eucl, method = "complete")
grp <- cutree(hc, k=2)
```

```
table(grp)
```

```
## grp
##   1   2
## 298   5
```

```
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. - Complete linkage method and Euclidean distance,
         K=2",
         cex.subtitle= 0.5)
```

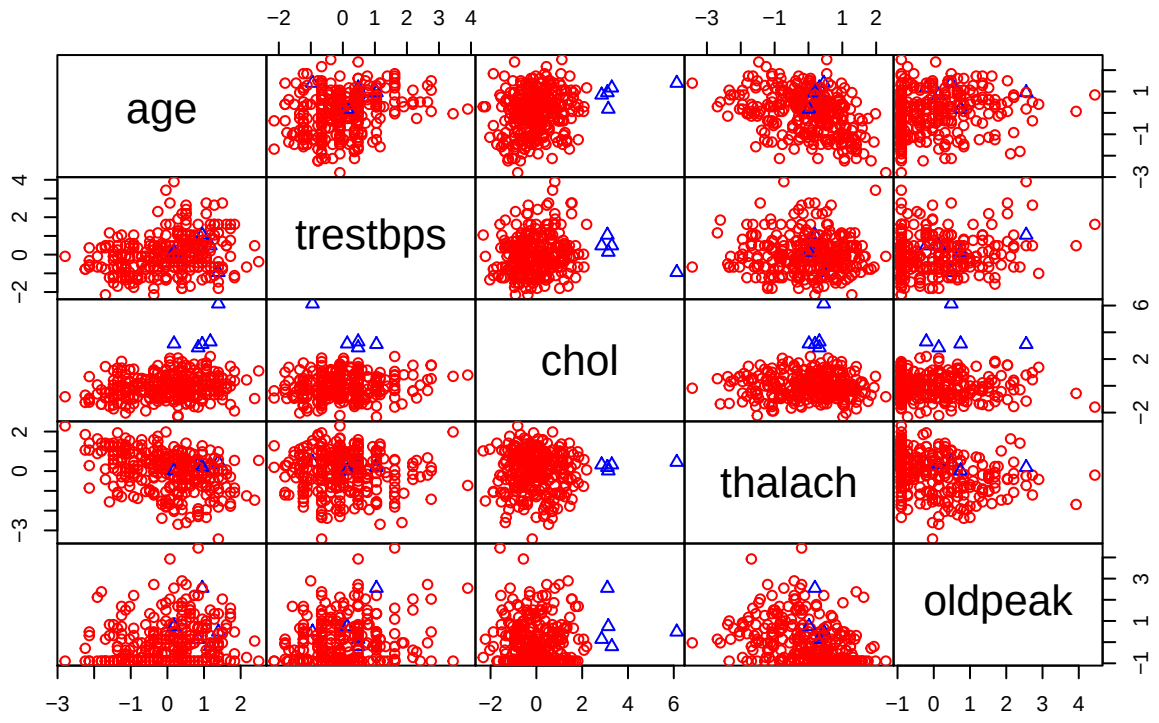



```
cor(dist.eucl, cophenetic(hc))
```

```
## [1] 0.4405635
```

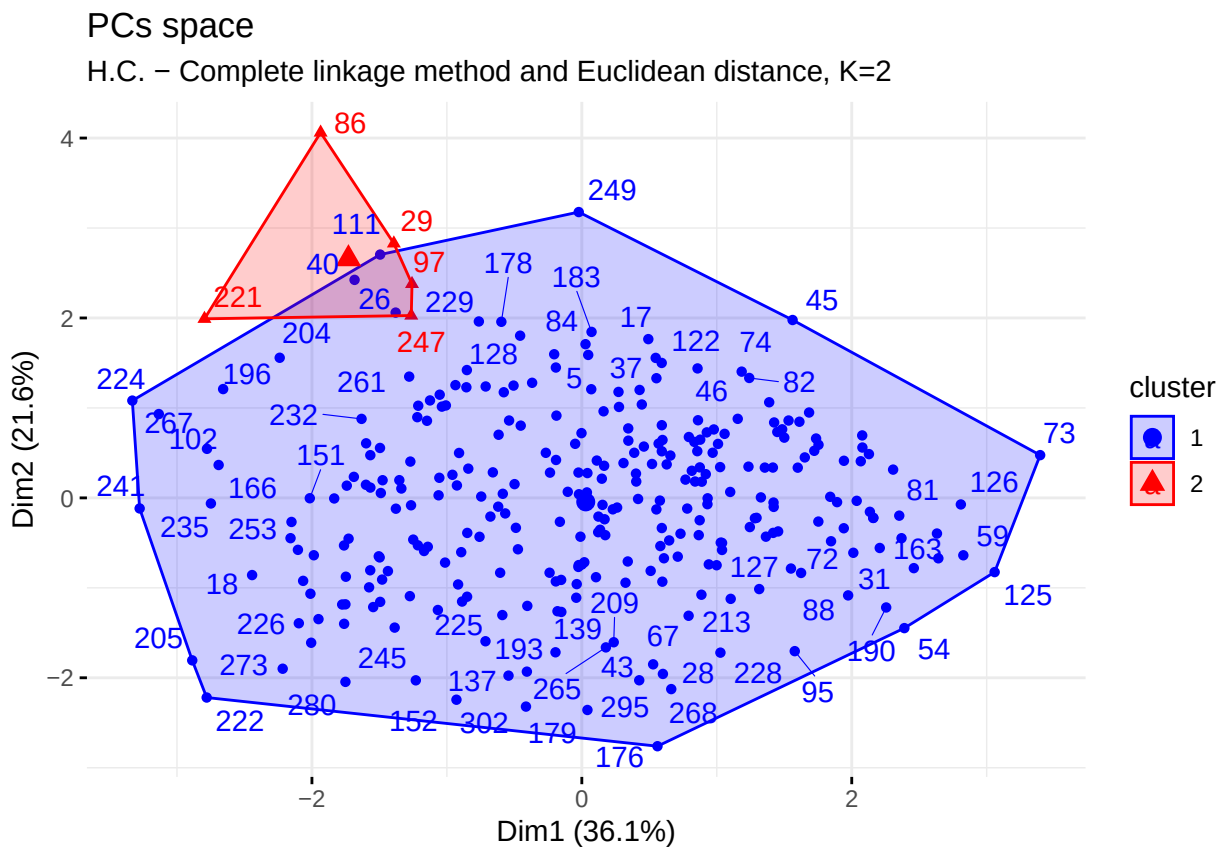
According to the result of "NbClust", the best number of clusters, applying hierarchical clustering method and using complete linkage method and Euclidean distance is 2.

```
pairs(heart_scale, gap=0, pch=grp, cex.main= 0.7,
main="Original space\nH.C.-Complete linkage method and Euclidian distance,
K=2",
col=c("red", "blue")[grp])
```




```
fviz_cluster(list(data = heart_scale, cluster = grp),
             palette = c("blue", "red"), ellipse.type = "convex",
             main="PCs space", repel = TRUE, show.clust.aver = FALSE,
             ggtheme = theme_minimal())+
  labs(subtitle = "H.C. - Complete linkage method and Euclidean distance, K=2",
       cex.sub= 0.5)
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analysed.

Internal validation measures: silhouette width and Dunn index

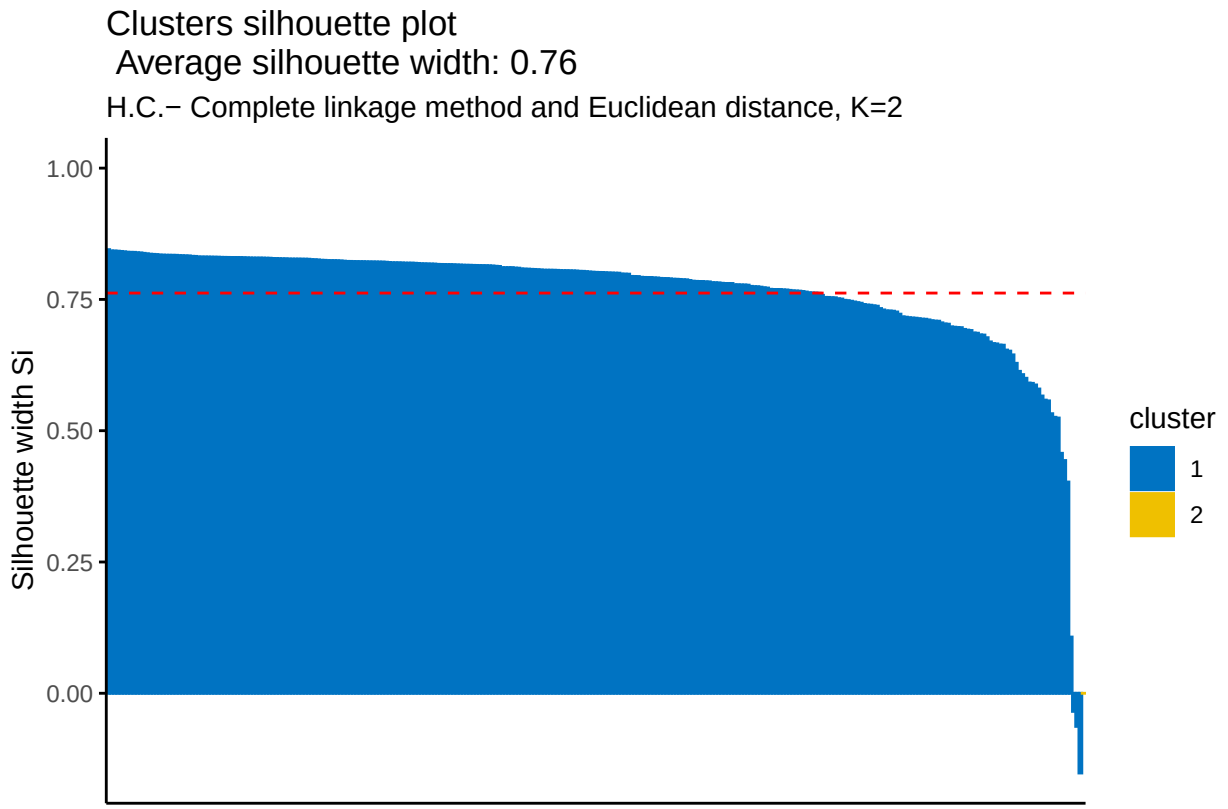
Silhouette width

```
hclust<- eclust(heart_sub,k=2 ,"hclust",hc_method  = "complete",nboot = 50, hc_metric = "euclidean")
silinfo <- hclust$silinfo
silinfo$avg.width
```

```
## [1] 0.76208
```

```
fviz_silhouette(hclust, palette = "jco", ggtheme = theme_classic())+
labs(subtitle = "H.C.- Complete linkage method and Euclidean distance, K=2",
     cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  302          0.76
## 2       2    1          0.00
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.7646034 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
## 221       1        2 -0.03449182
## 247       1        2 -0.06307939
## 29        1        2 -0.15182483
```

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) also the units are on average the same silhouette value with respect to silhouette width. According to this index, 3 units (221, 247, 29)that belong to cluster 1 are not well clustered: they should belong to cluster 2.

Dunn index

```
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.4246056
```

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

External validation measures: confusion matrix, correct Rand index, Meila's

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

```
table(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  95   1
##   1 207   0
```

A large number of "female" sex (n = 95) has been classified in cluster 1 while cluster 2 have 1 number of values. The same happened for "male" sex (n=207) classified in cluster 1 while cluster 2 has 0 values.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.00761681
```

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.

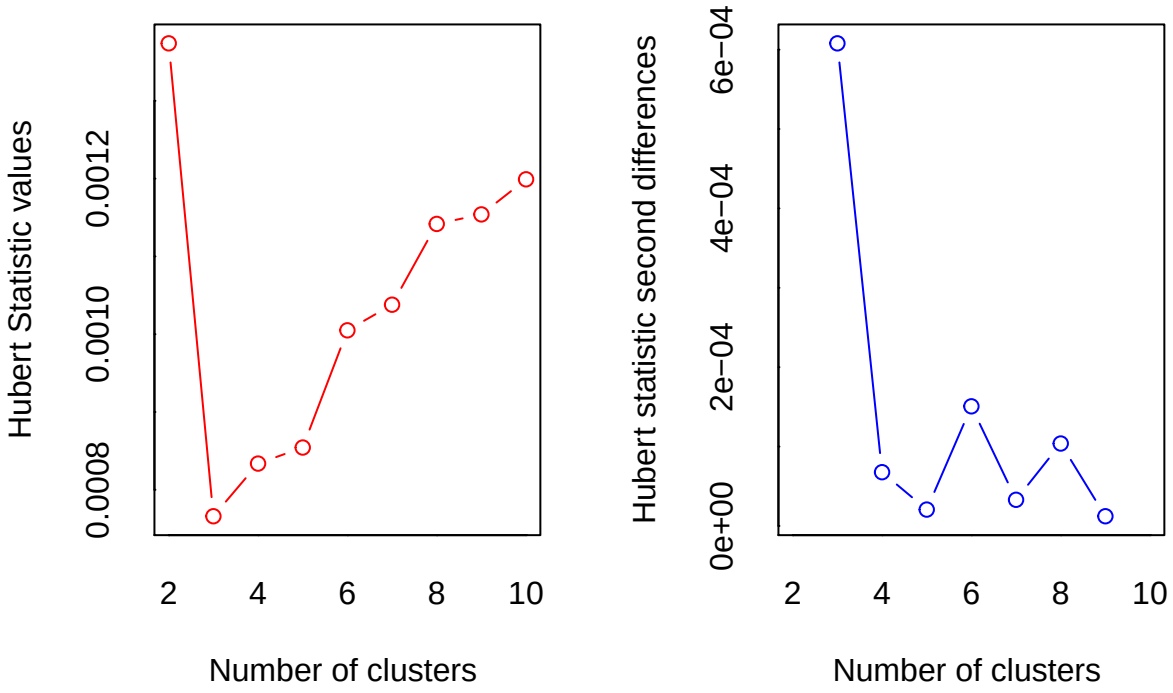
Meila's VI Index

```
stats$vi
```

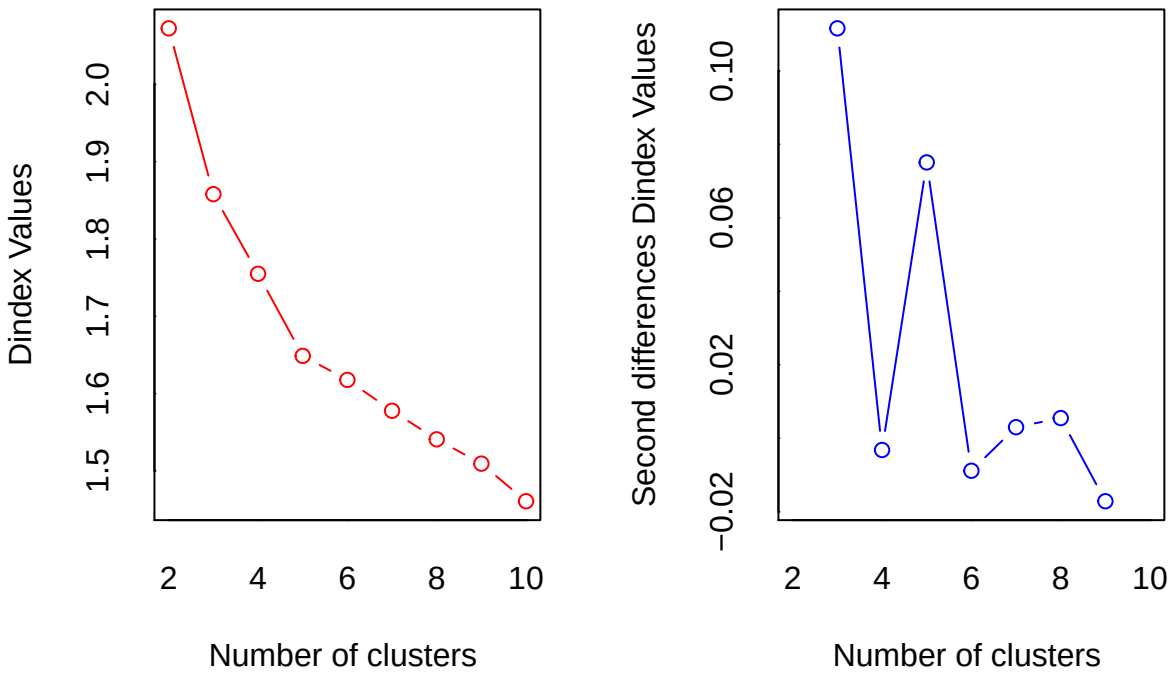
```
## [1] 0.639
```

Complete linkage method and Manhattan distance

```
nb <- NbClust(heart_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:
## * 11 proposed 2 as the best number of clusters
## * 8 proposed 3 as the best number of clusters
## * 4 proposed 5 as the best number of clusters
## * 1 proposed 7 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 and Manhattan distance", cex.sub= 0.5)
```

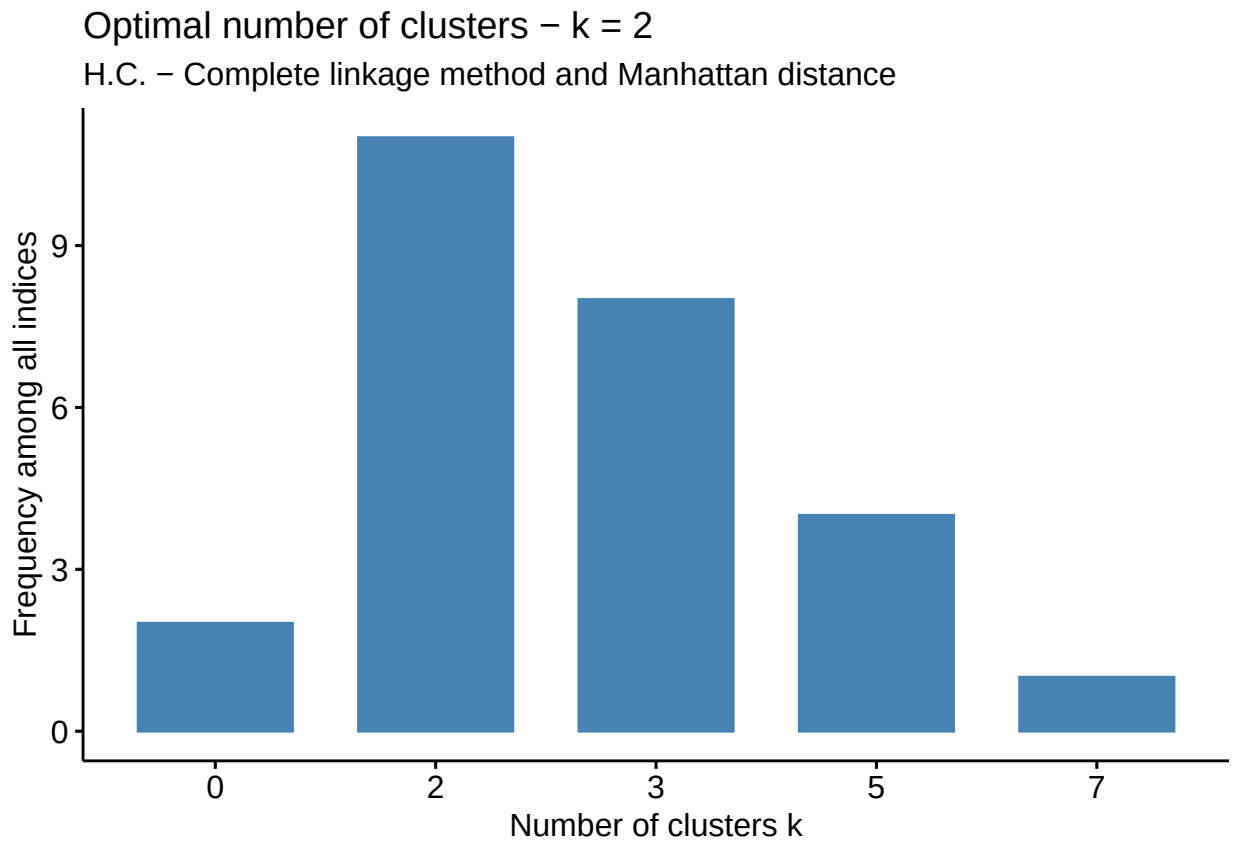
```
## Warning in if (class(best_nc) == "numeric") print(best_nc) else if
## (class(best_nc) == : the condition has length > 1 and only the first element
## will be used
```

```
## 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
```

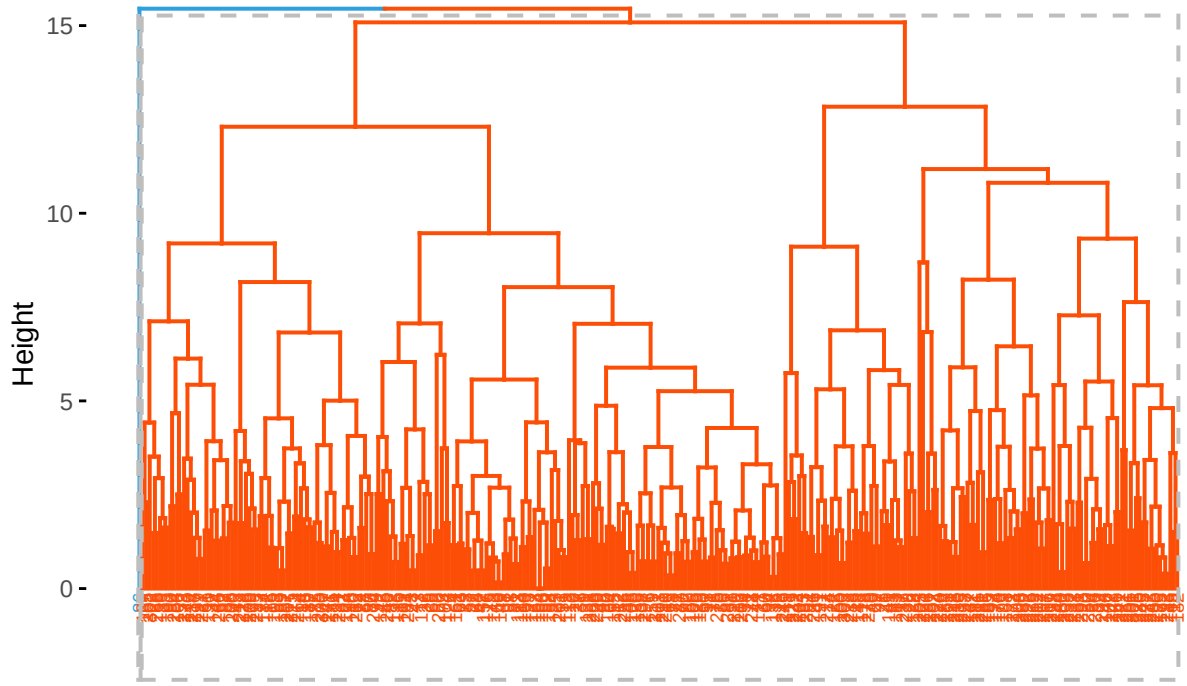
```
## Among all indices:
## ==================
## * 2 proposed  0 as the best number of clusters
## * 11 proposed  2 as the best number of clusters
## * 8 proposed  3 as the best number of clusters
## * 4 proposed  5 as the best number of clusters
## * 1 proposed  7 as the best number of clusters
##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




```r
dist.man <- dist(heart_scale, method = "manhattan")
hc <- hclust(dist.man, method = "complete")
grp <- cutree(hc, k=2)
table(grp)
```

```
## grp
##   1   2
## 302   1
```

```
fviz_dend(hc, k = 2, cex = 0.5, k_colors = c("#2E9FDF", "#FC4E07"),
          r_labels_by_k = TRUE, rect = TRUE)+labs(title = "Dendrogram",
          subtitle = "H.C. - Complete linkage method and Manhattan distance, K=2",cex.subtitle= 0.5)
```

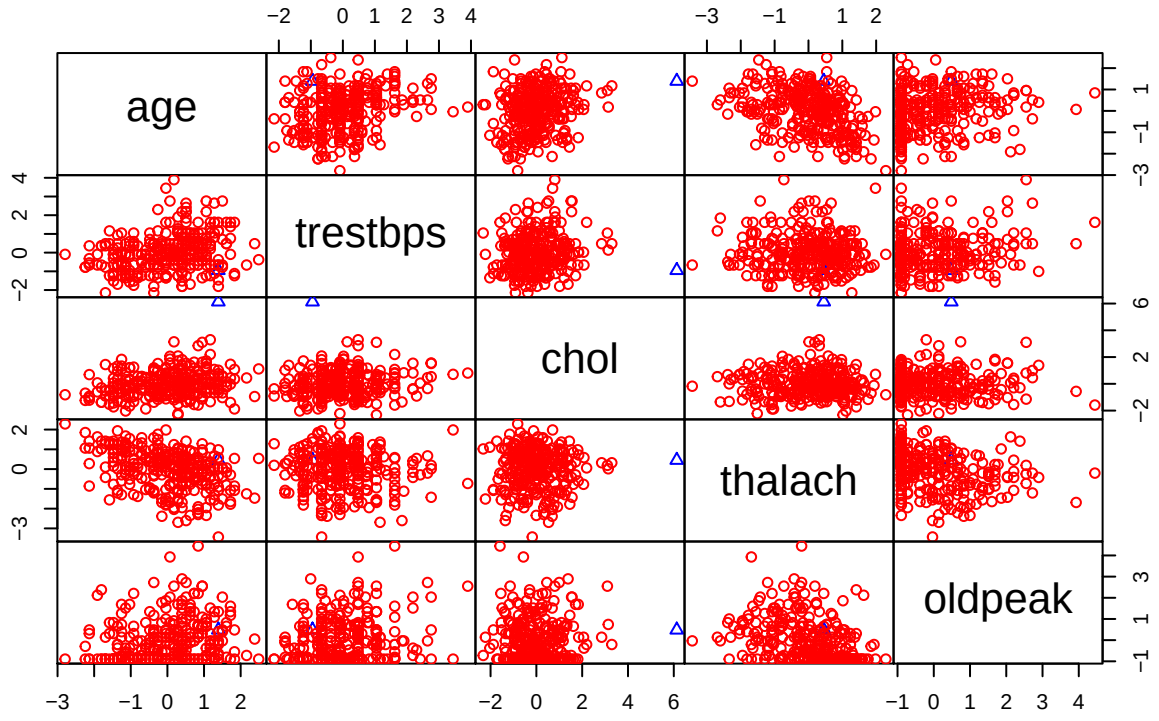



```
cor(dist.man, cophenetic(hc))
```

```
## [1] 0.4771181
```

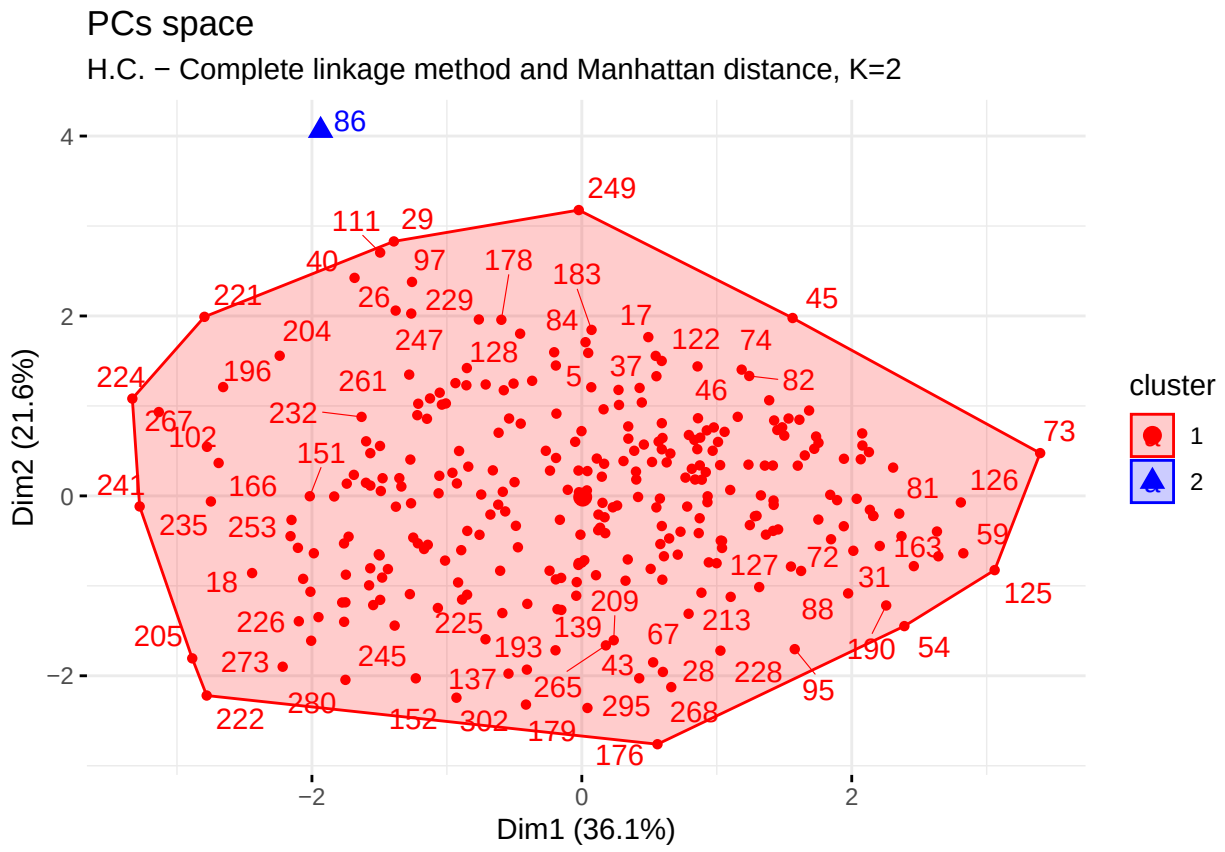
According to the result of "NbClust", the best number of clusters, applying hierarchical clustering method and using complete linkage method and Manhattan distance is 2.

```
pairs(heart_scale, gap=0,  pch=grp, cex.main= 0.7, main="Original space\nH.C.-Complete linkage method a
"blue")[grp])
```




```
fviz_cluster(list(data = heart_scale, cluster = grp),  palette = c("red", "blue"), ellipse.type = "conv
             show.clust.aver = FALSE, ggtheme = theme_minimal())+labs(
               subtitle = "H.C. - Complete linkage method and Manhattan distance, K=2", cex.sub= 0.5)
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analysed

Internal validation measures: silhouette width and Dunn index

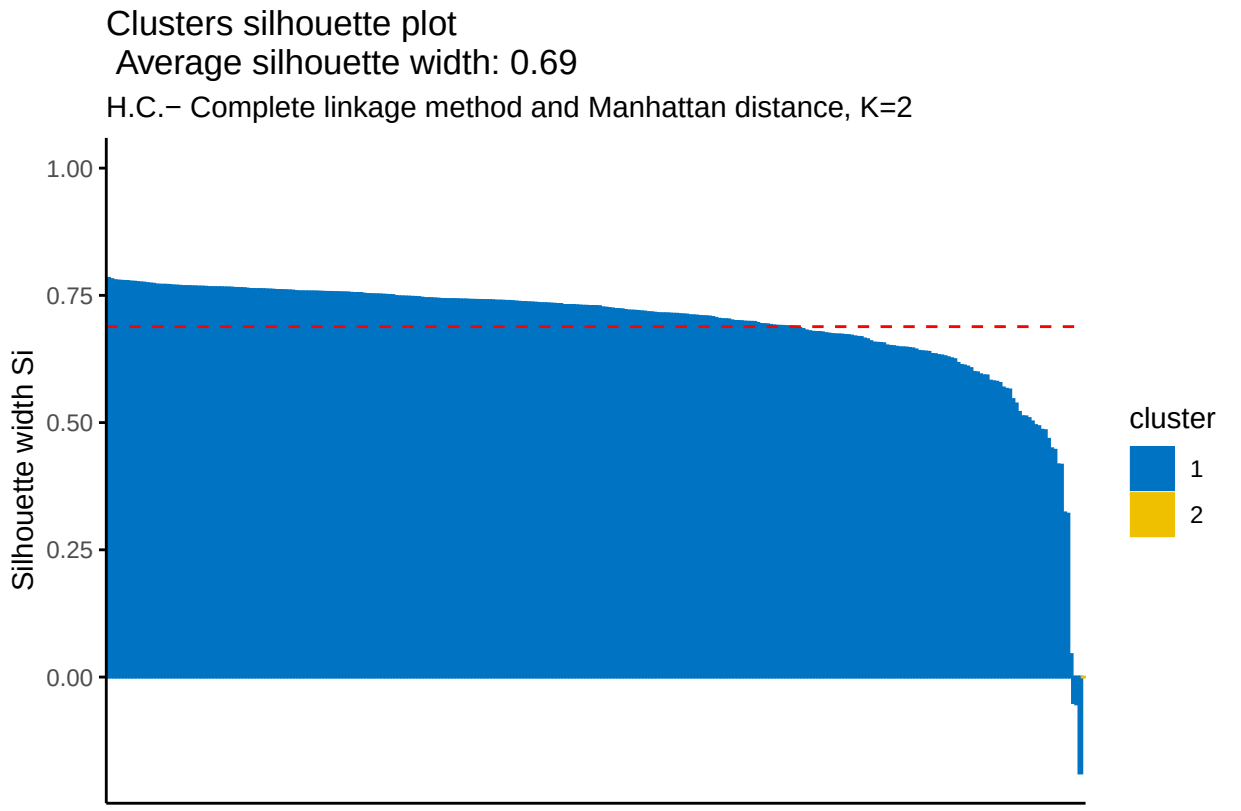
Silhouette width

```
hclust<- eclust(heart_sub,k=2 ,"hclust",hc_method  = "complete",nboot = 50, hc_metric = "manhattan")
silinfo <- hclust$silinfo
silinfo$avg.width
```

```
## [1] 0.6885344
```

```
fviz_silhouette(hclust, palette = "jco", ggtheme = theme_classic())+
labs(subtitle = "H.C.- Complete linkage method and Manhattan distance, K=2", cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  302          0.69
## 2       2    1          0.00
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.6908143 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
## 247       1        2 -0.05007579
## 221       1        2 -0.05279753
## 29        1        2 -0.18855131
```

The value of complete silhouette width indicates that on average the units are well enough clustered. As in particular, in each cluster the units are on average the same silhouette value with respect to the silhouette width. According to this index, 3 units (221, 247, 29)that belong to cluster 1 are not well clustered: they should belong to cluster 2.

Dunn index

```
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.4246056
```

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

External validation measures: confusion matrix, correct Rand index,

Meila's IV index ### 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(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  95   1
##   1 207   0
```

A large number of "female" sex (n = 95) has been classified in cluster 1 while cluster 2 and 1 number of values. The same happened for "male" sex (n=207) classified in cluster 1 while cluster 2 has 0 values.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.00761681
```

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

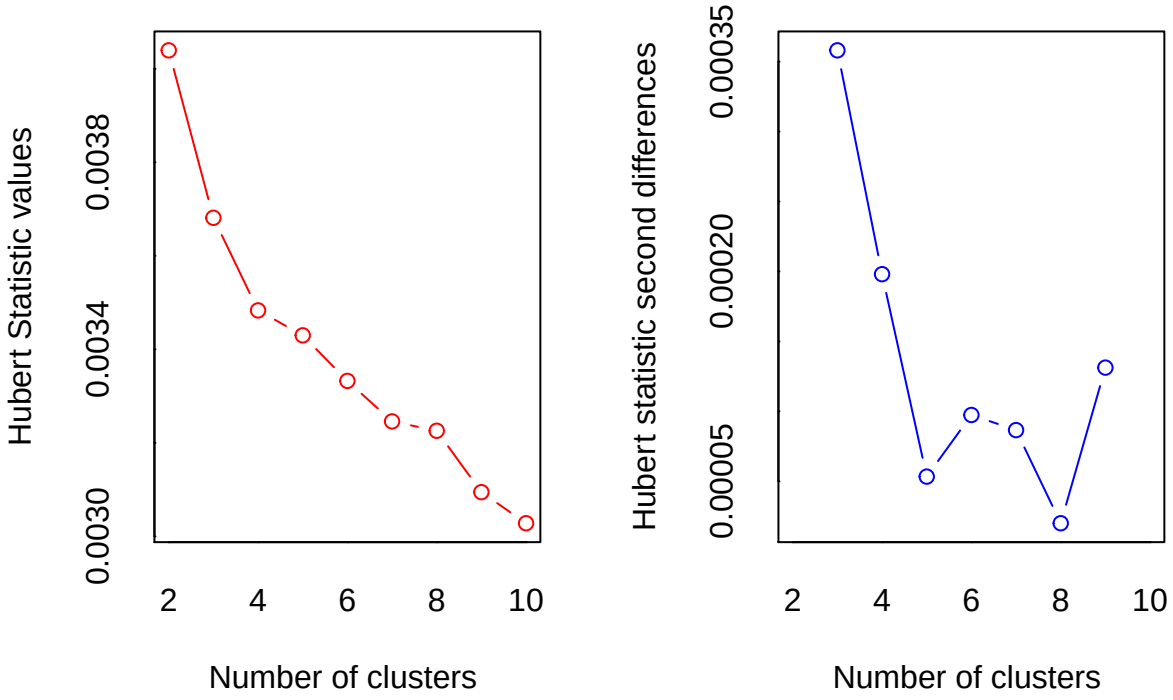
```
stats$vi
```

```
## [1] 0.639
```

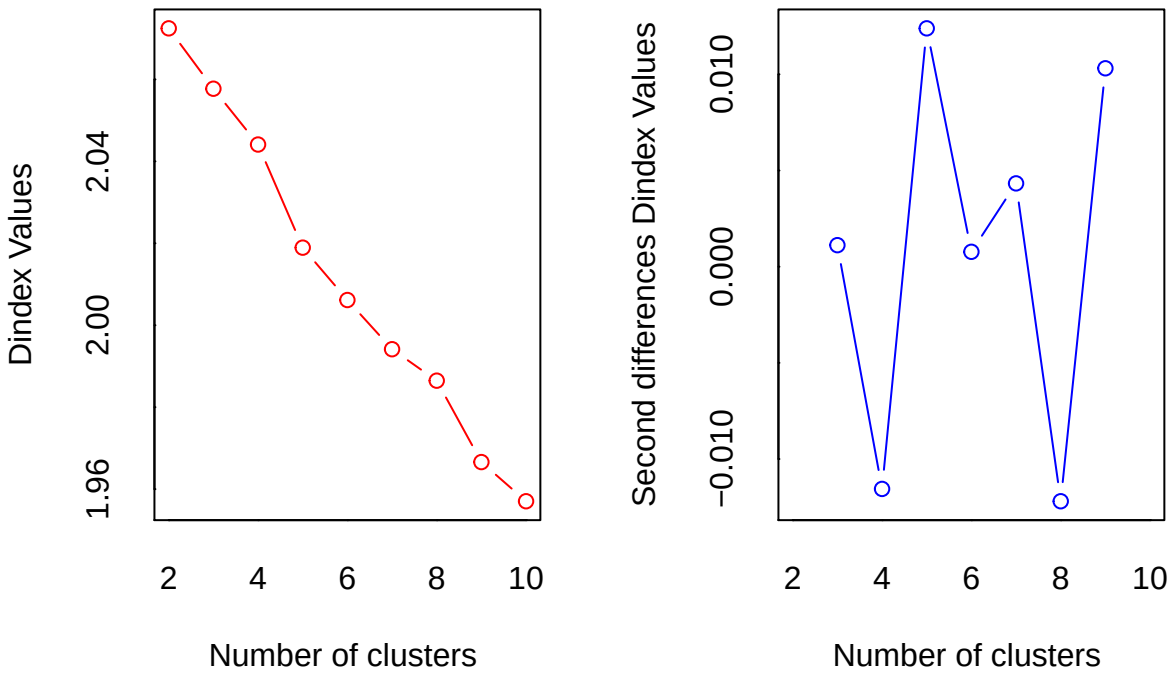
Centroid linkage method and Euclidean distance

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

```
## 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.
##
```

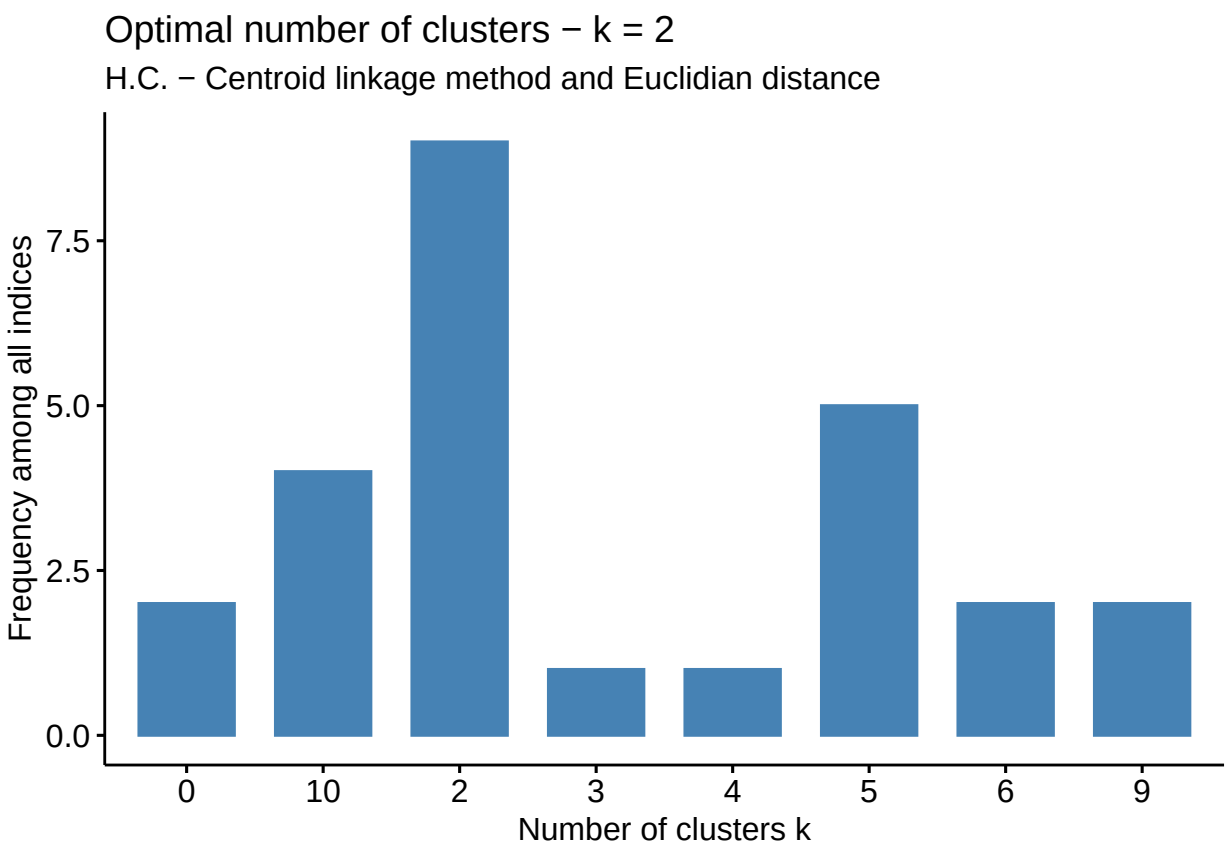



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##       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

```
library(factoextra)
fviz_nbclust(nb)+labs(subtitle = "H.C. - Centroid linkage method and 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
## * 9 proposed  2 as the best number of clusters
## * 1 proposed  3 as the best number of clusters
## * 1 proposed  4 as the best number of clusters
## * 5 proposed  5 as the best number of clusters
## * 2 proposed  6 as the best number of clusters
## * 2 proposed  9 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  2 .
```

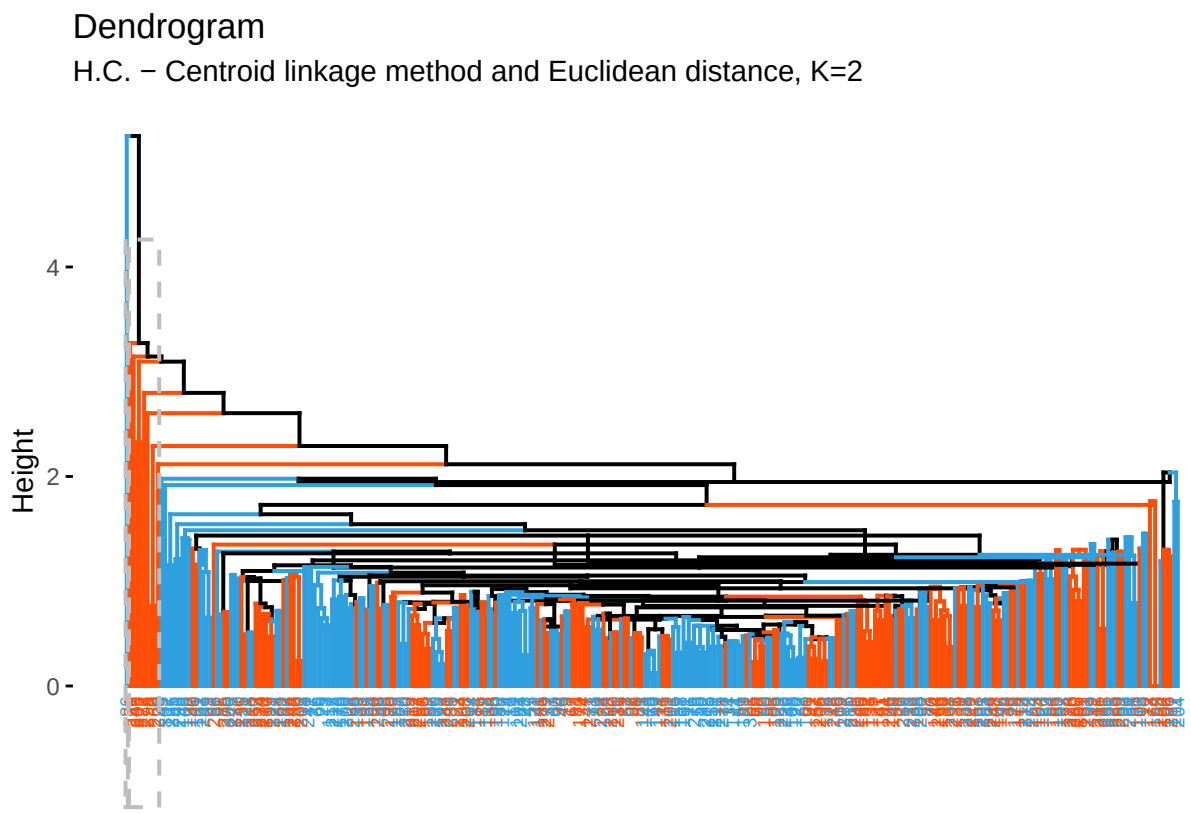


```
dist.eucl <- dist(heart_scale, method = "euclidian")
hc <- hclust(dist.eucl, method = "centroid")
grp <- cutree(hc, k=2)
table(grp)
```

```
## grp
##   1   2
## 302   1
```

```
fviz_dend(hc, k = 2, cex = 0.5, k_colors = c("#2E9FDF", "#FC4E07"), color_labels_by_k = TRUE, rect = TRU
```

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

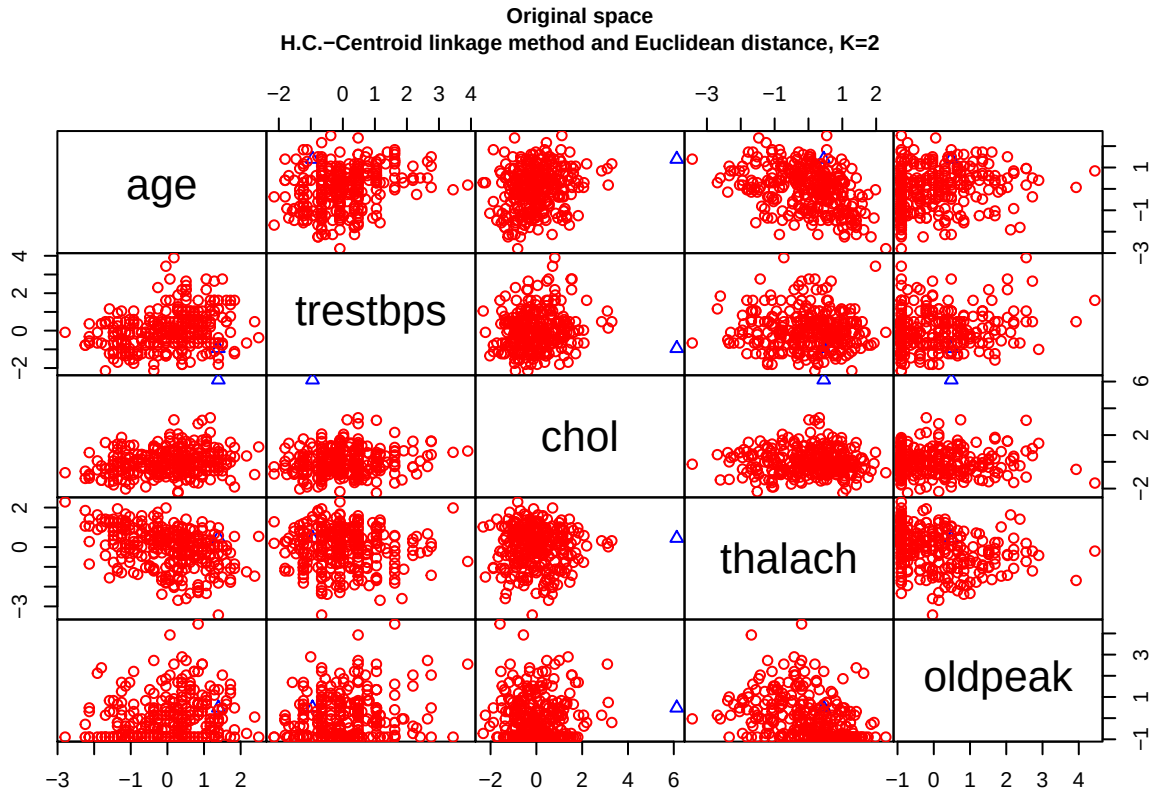



```
cor(dist.eucl, cophenetic(hc))
```

```
## [1] 0.6099643
```

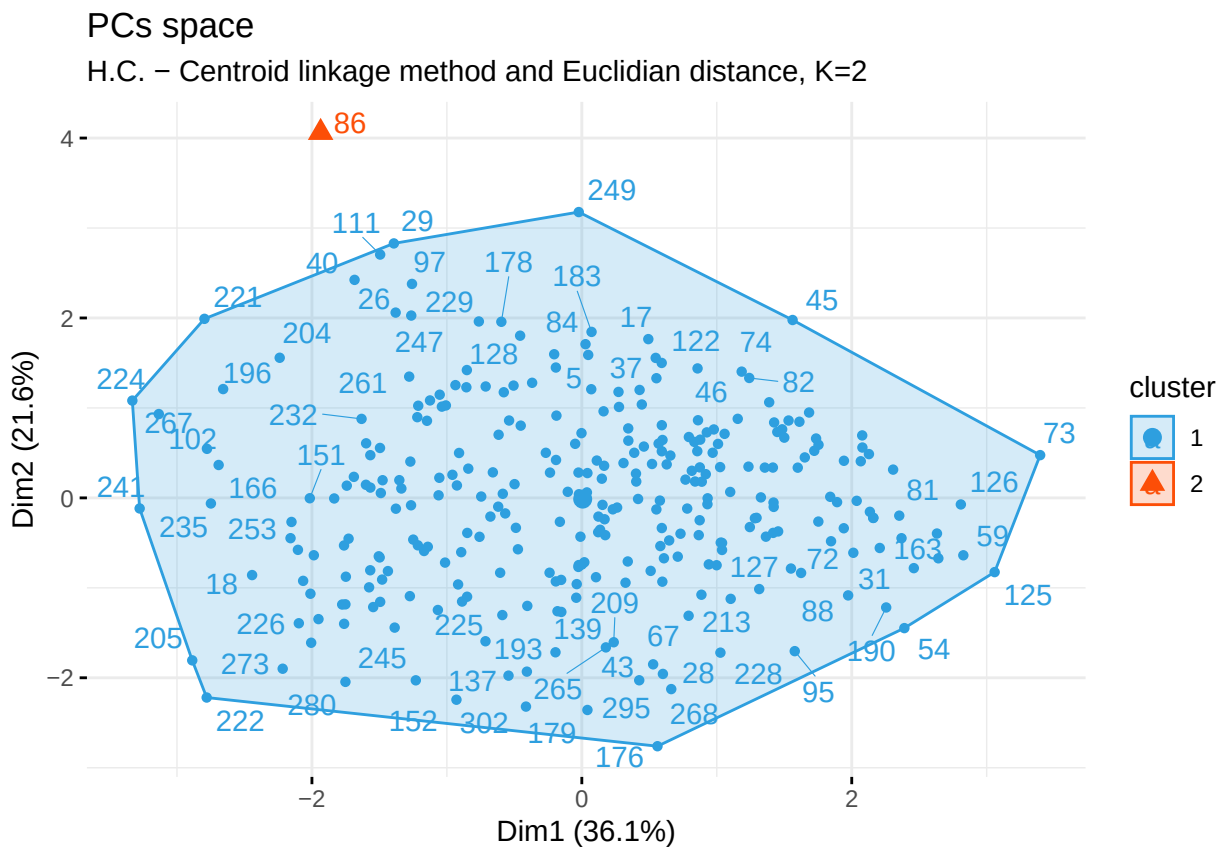
According to the result of "NbClust", the best number of clusters, applying hierarchical clustering method and using the Centroid linkage method and Euclidean distance is 2.

```
pairs(heart_scale, gap=0, pch=grp, cex.main= 0.7, main="Original space\nH.C.-Centroid linkage method an
```




```
fviz_cluster(list(data = heart_scale, cluster = grp), palette = c("#2E9FDF", "#FC4E07"), ellipse.type =
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analysed.

Internal validation measures: silhouette width and Dunn index

Silhouette width

```
hclust<- eclust(heart_sub,k=2 ,"hclust",hc_method  = "centroid", nboot = 50, hc_metric = "euclidean")
```

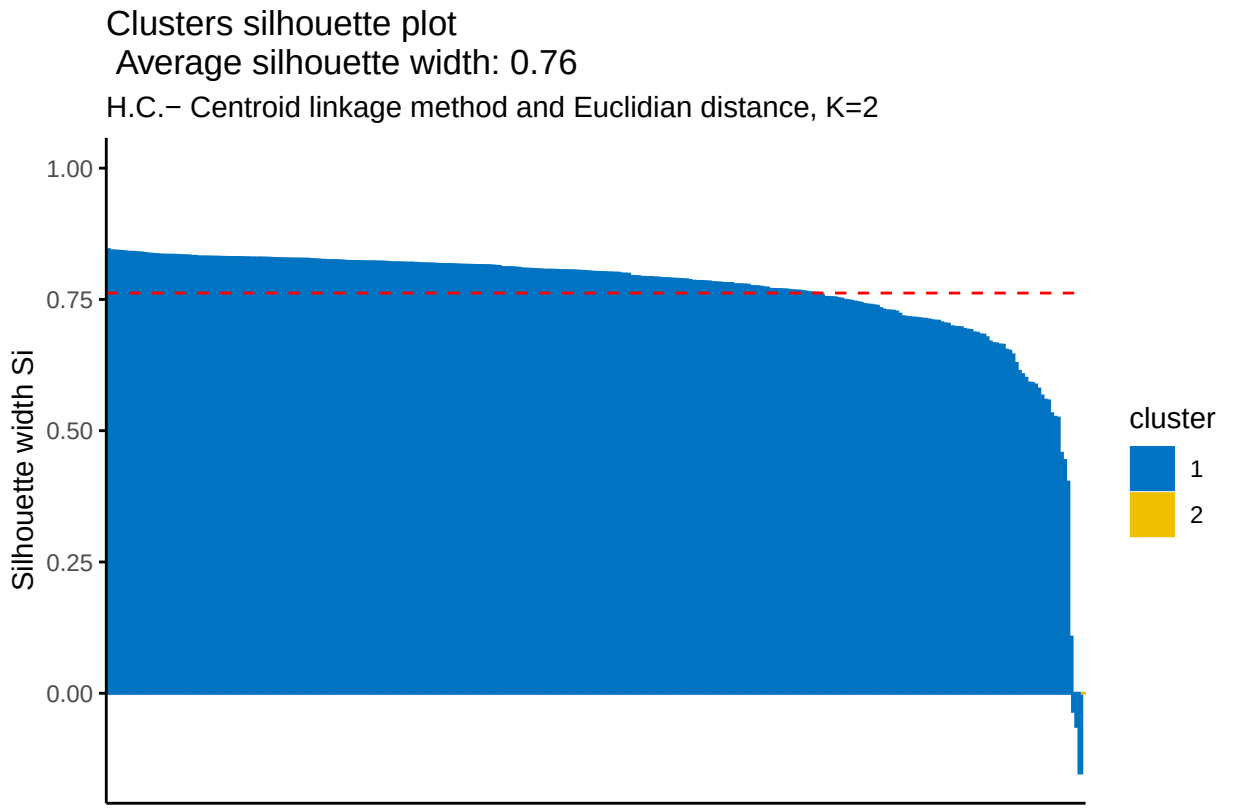
```
## Warning in get_col(col, k): Length of color vector was shorter than the number
## of clusters - color vector was recycled
```

```
silinfo <- hclust$silinfo
silinfo$avg.width
```

```
## [1] 0.76208
```

```
fviz_silhouette(hclust, palette = "jco", ggtheme = theme_classic())+labs(
subtitle = "H.C.- Centroid linkage method and Euclidian distance, K=2",
cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  302          0.76
## 2       2    1          0.00
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.7646034 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
## 221       1        2 -0.03449182
## 247       1        2 -0.06307939
## 29        1        2 -0.15182483
```

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) also the units are on average the same silhouette value with respect to silhouette width. According to this index, 3 units (221, 247, 29)that belong to cluster 1 are not well clustered: they should belong to cluster 2.

Dunn index

```
library(fpc)
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.4246056
```

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

External validation measures: confusion matrix, correct Rand index,

Meila's IV index ### Confusion matrix According to the Confusion matrix, the number of clusters is equal to the nominal values.

```
table(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  95   1
##   1 207   0
```

A large number of "female" sex (n = 95) has been classified in cluster 1 while cluster 2 have 1 number of values. The same happened for "male" sex (n=207) classified in cluster 1 while cluster 2 has 0 values.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.00761681
```

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

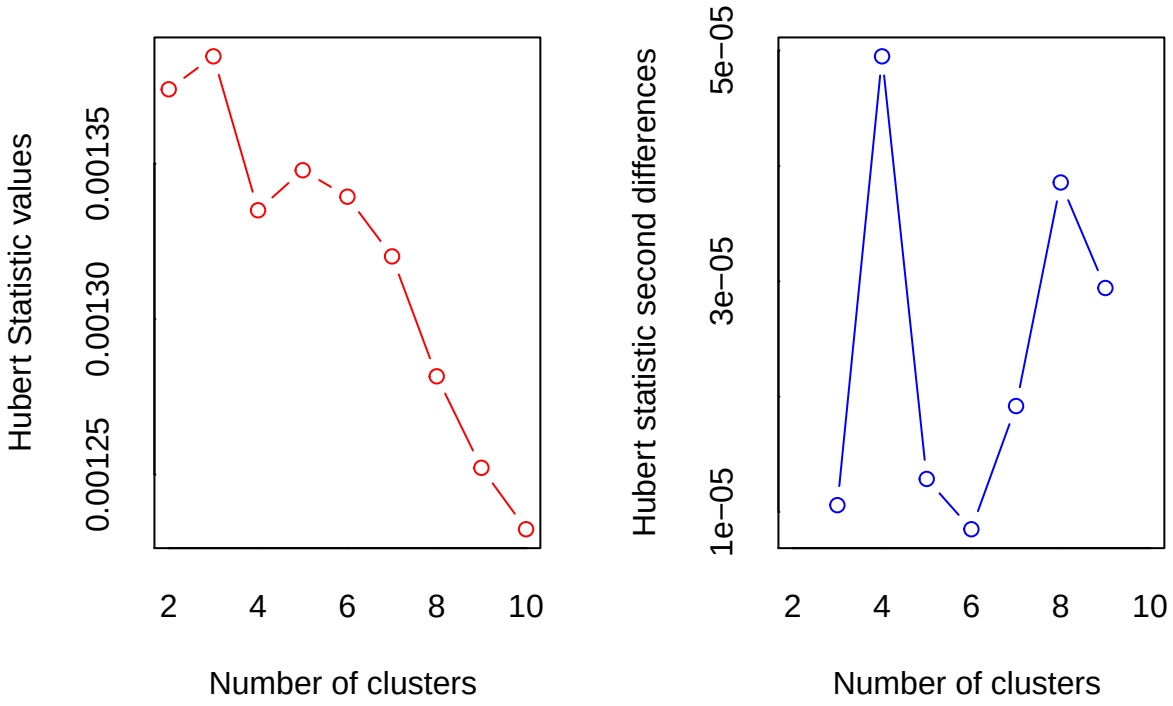
```
stats$vi
```

```
## [1] 0.639
```

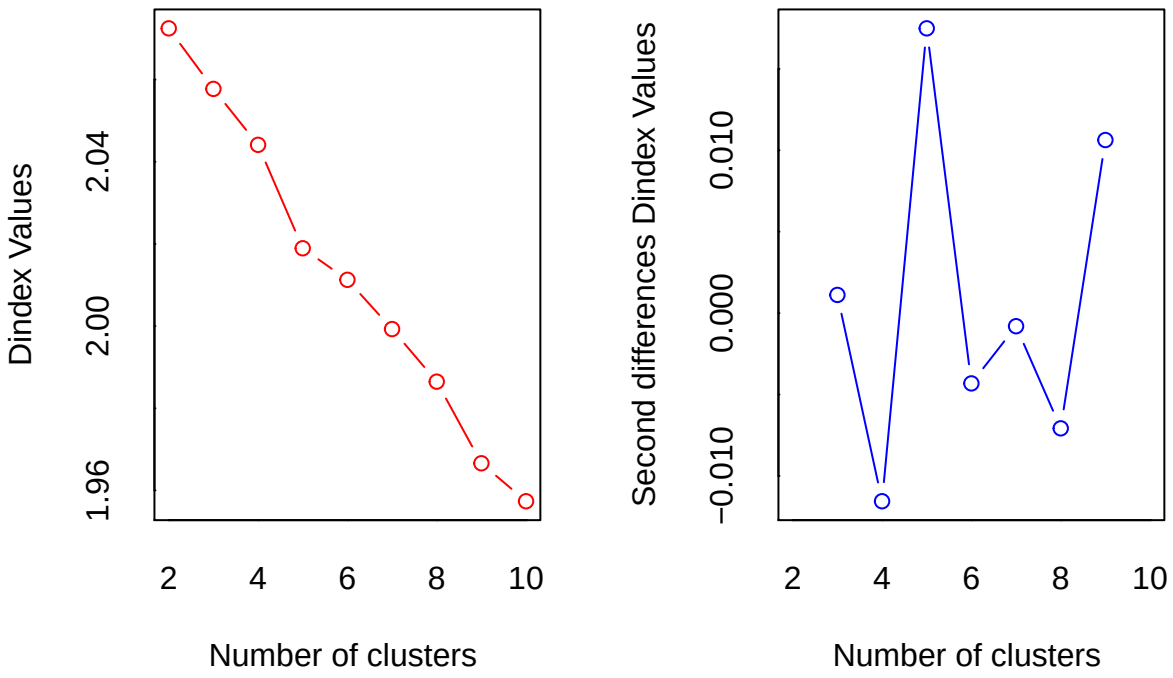
Centroid linkage method and Manhattan distance

```
nb <- NbClust(heart_scale, distance = "manhattan", min.nc = 2, max.nc = 10,
              method = "centroid")
```

```
## 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.
##
```

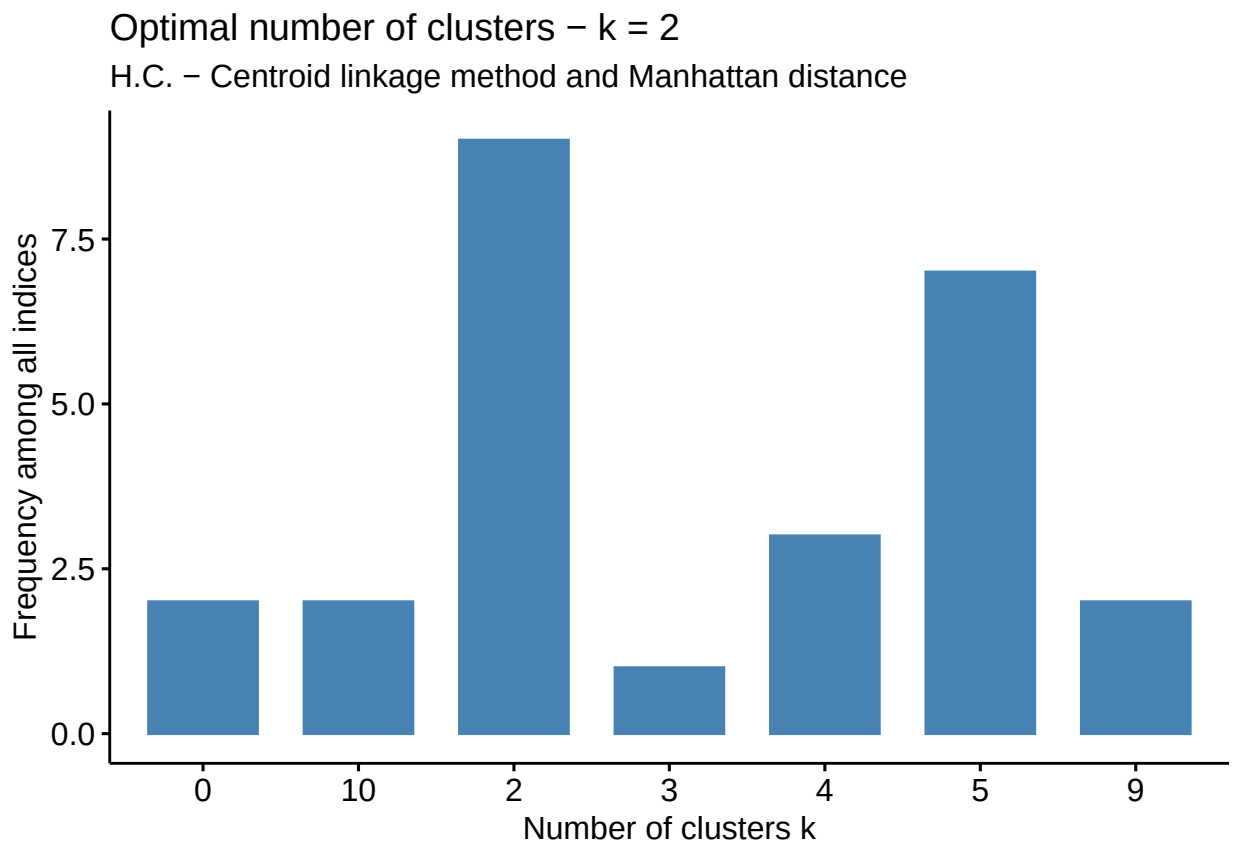



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##       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




```r
dist.man <- dist(heart_scale, method = "manhattan")
hc <- hclust(dist.man, method = "centroid")
grp <- cutree(hc, k=2)
table(grp)
```

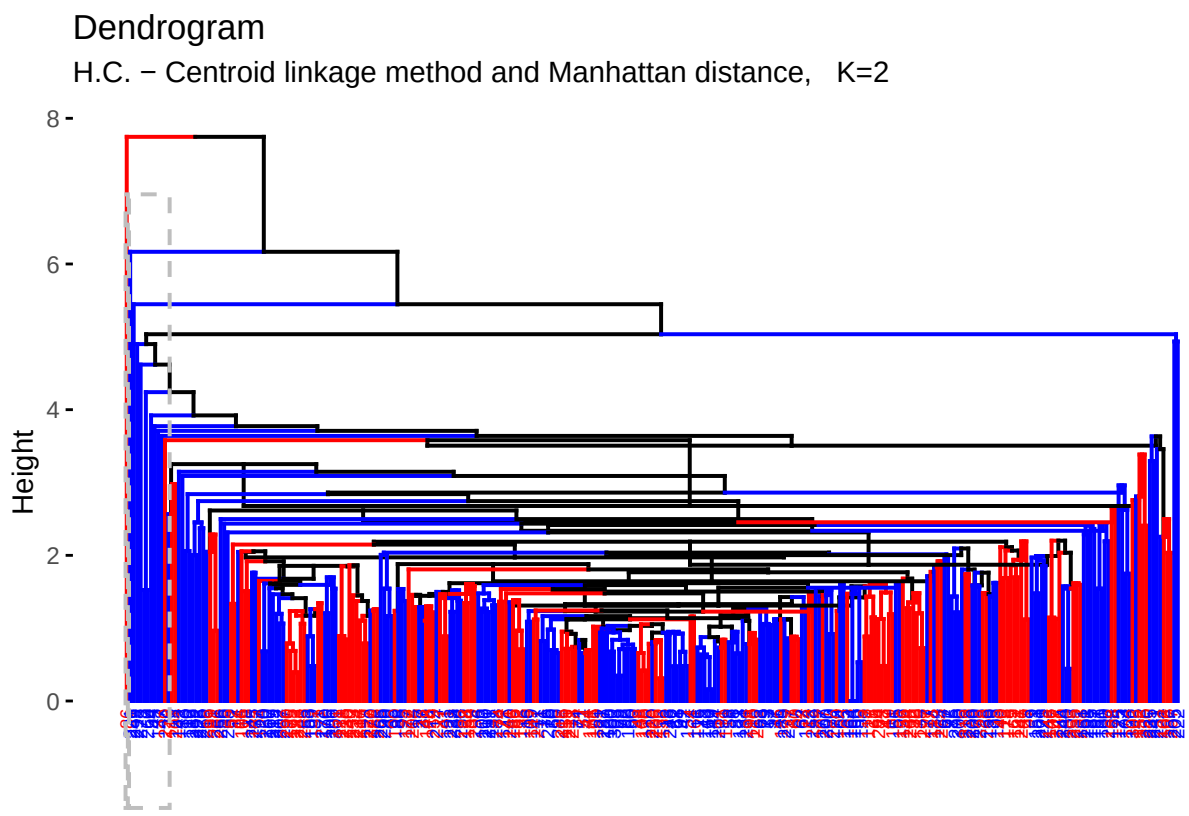
```
## grp
##   1   2
## 302   1
```

```
grp
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [297] 1 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 and 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
```

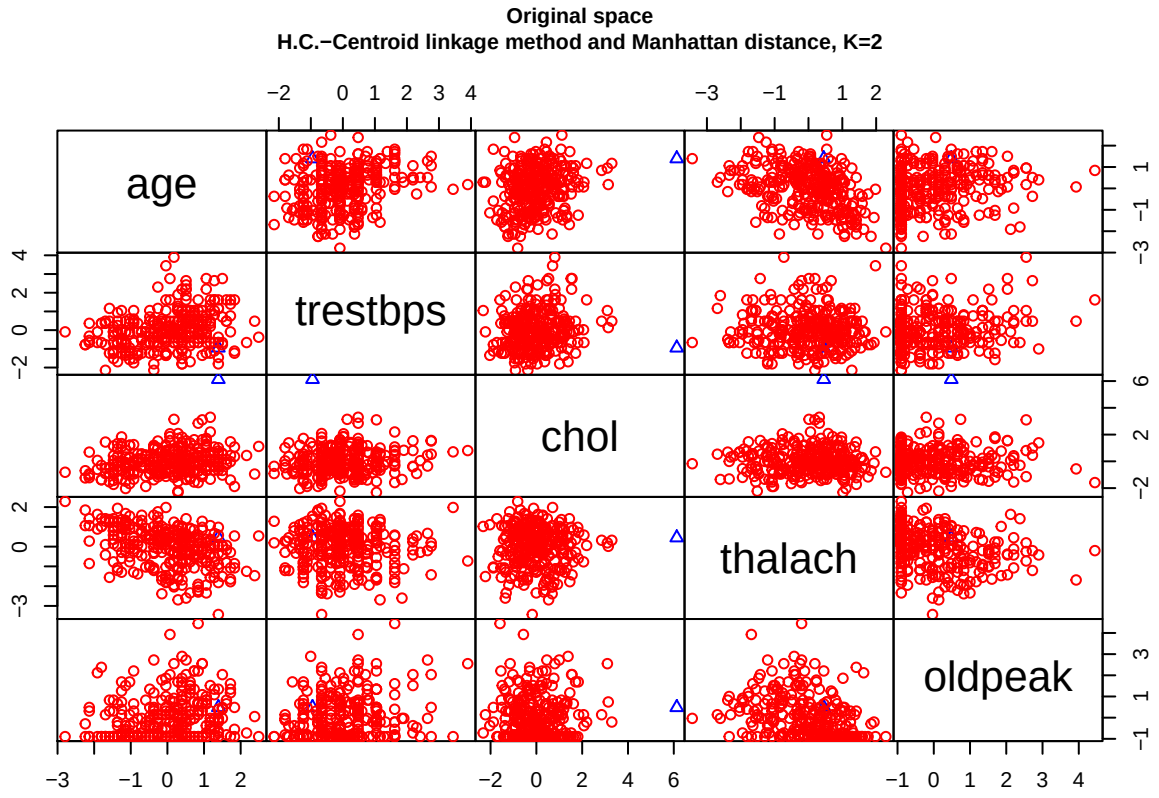



```
cor(dist.eucl, cophenetic(hc))
```

```
## [1] 0.6263482
```

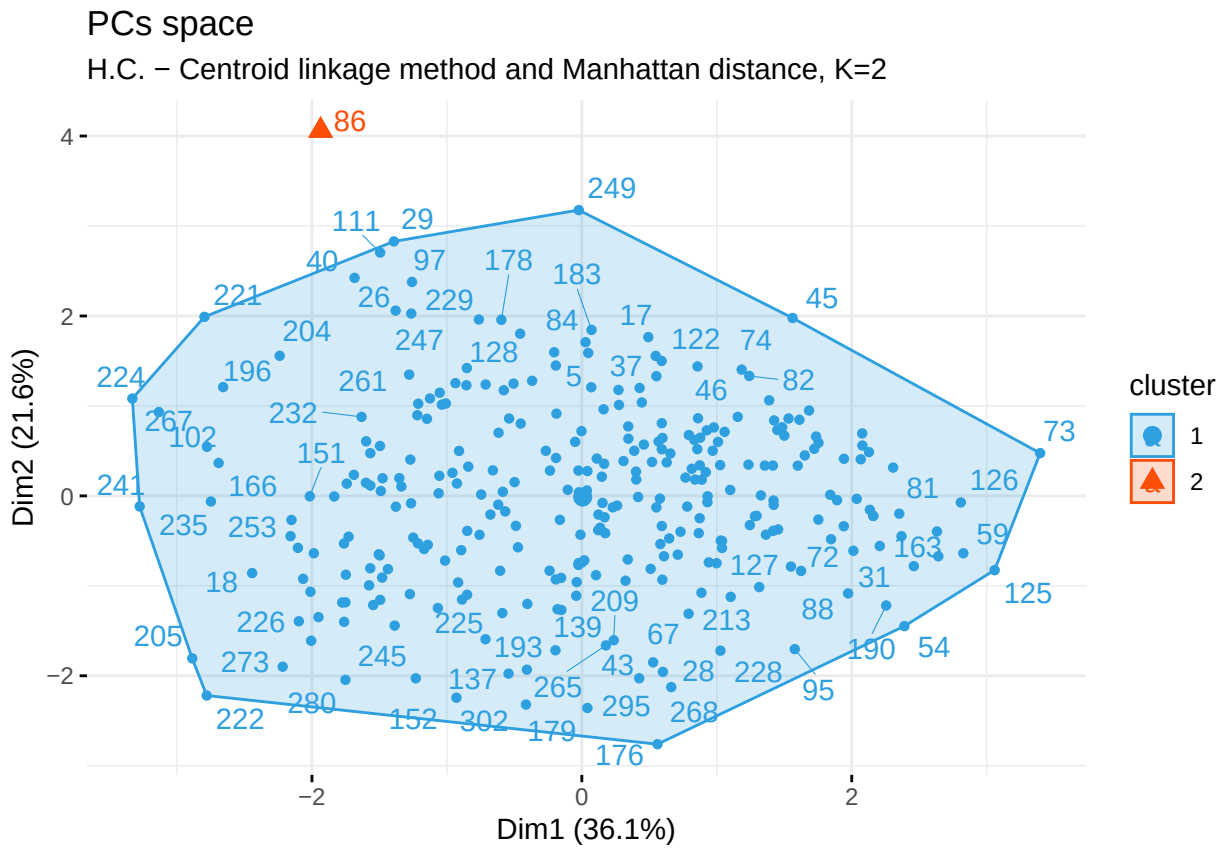
According to the result of "NbClust", the best number of clusters, applying hierarchical clustering method and using complete linkage method and Manhattan distance is 2.

```
pairs(heart_scale, gap=0, pch=grp, cex.main= 0.7, main="Original space\nH.C.-Centroid linkage method an
```




```
fviz_cluster(list(data = heart_scale, cluster = grp), palette = c("#2E9FDF", "#FC4E07"), ellipse.type =
labs(subtitle = "H.C. - Centroid linkage method and Manhattan distance, K=2", cex.sub= 0.5)
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analysed

Internal validation measures: silhouette width and Dunn index

Silhouette width

```
hclust<- eclust(heart_sub,k=2 ,"hclust",hc_method  = "centroid",nboot = 50, hc_metric = "manhattan")
```

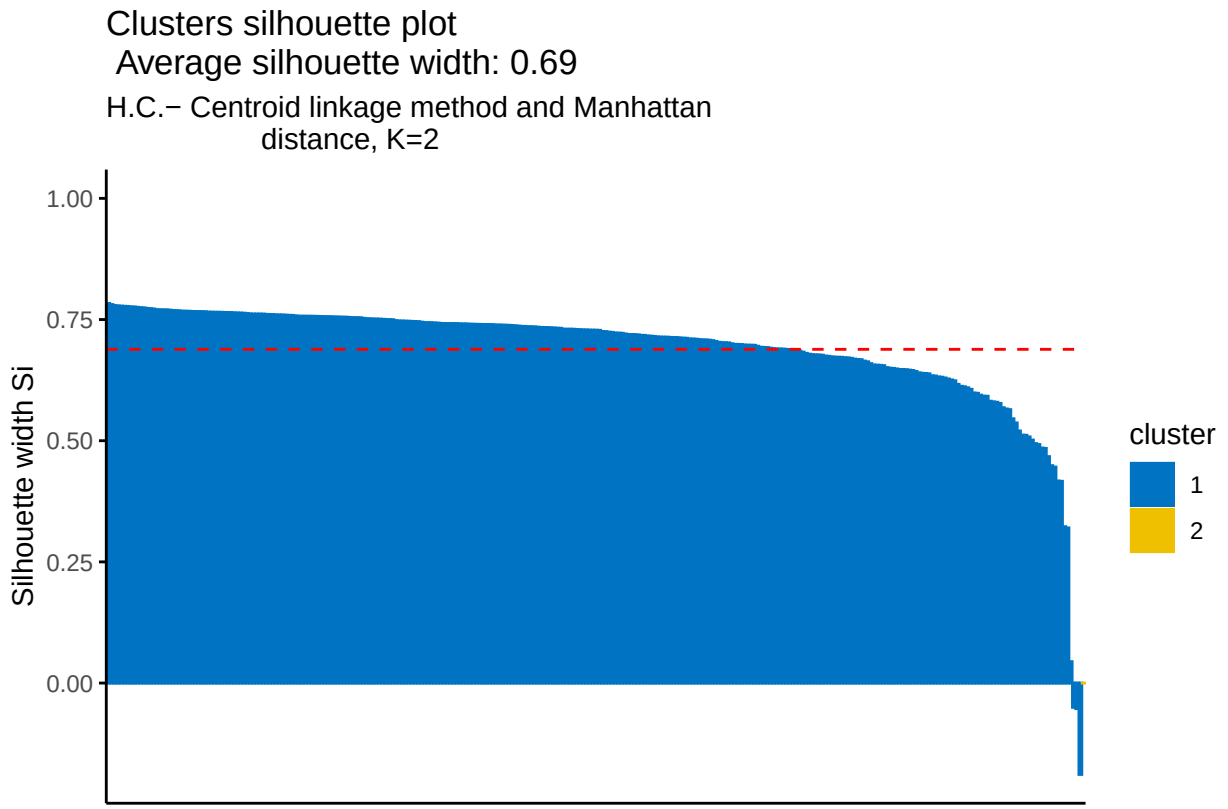
```
## Warning in get_col(col, k): Length of color vector was shorter than the number
## of clusters - color vector was recycled
```

```
silinfo <- hclust$silinfo
silinfo$avg.width
```

```
## [1] 0.6885344
```

```
fviz_silhouette(hclust, palette = "jco", ggtheme = theme_classic())+labs(
                  subtitle = "H.C.- Centroid linkage method and Manhattan
                   distance, K=2", cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  302          0.69
## 2       2    1          0.00
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.6908143 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
## 247       1        2 -0.05007579
## 221       1        2 -0.05279753
## 29        1        2 -0.18855131
```

The value of complete silhouette width indicates that on average the units are well enough clustered. As in particular, in each cluster the units are on average the same silhouette value with respect to the silhouette width. According to this index, 3 units (221, 247, 29)that belong to cluster 1 are not well clustered: they should belong to cluster 2.

Dunn index

```
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.4246056
```

External validation measures: confusion matrix, correct Rand index,

Meila's IV index ### 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(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  95   1
##   1 207   0
```

A large number of "female" sex (n = 95) has been classified in cluster 1 while cluster 2 and 1 number of values. The same happened for "male" sex (n=207) classified in cluster 1 while cluster 2 has 0 values.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.00761681
```

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.

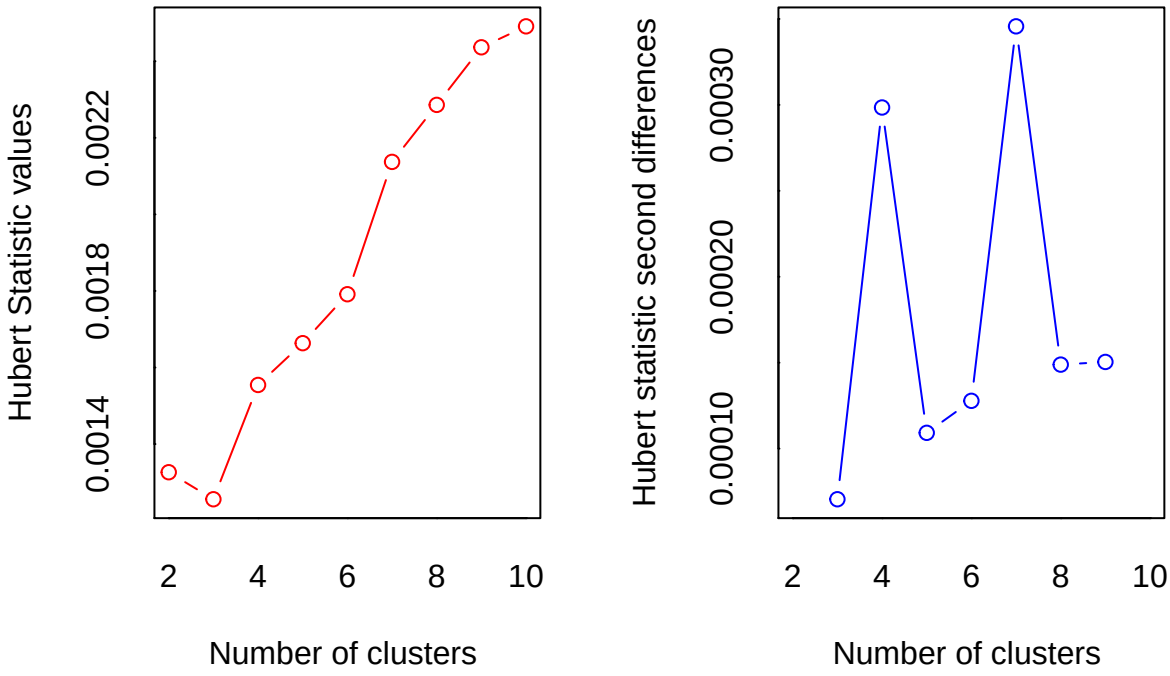
Meila's VI Index

```
stats$vi
```

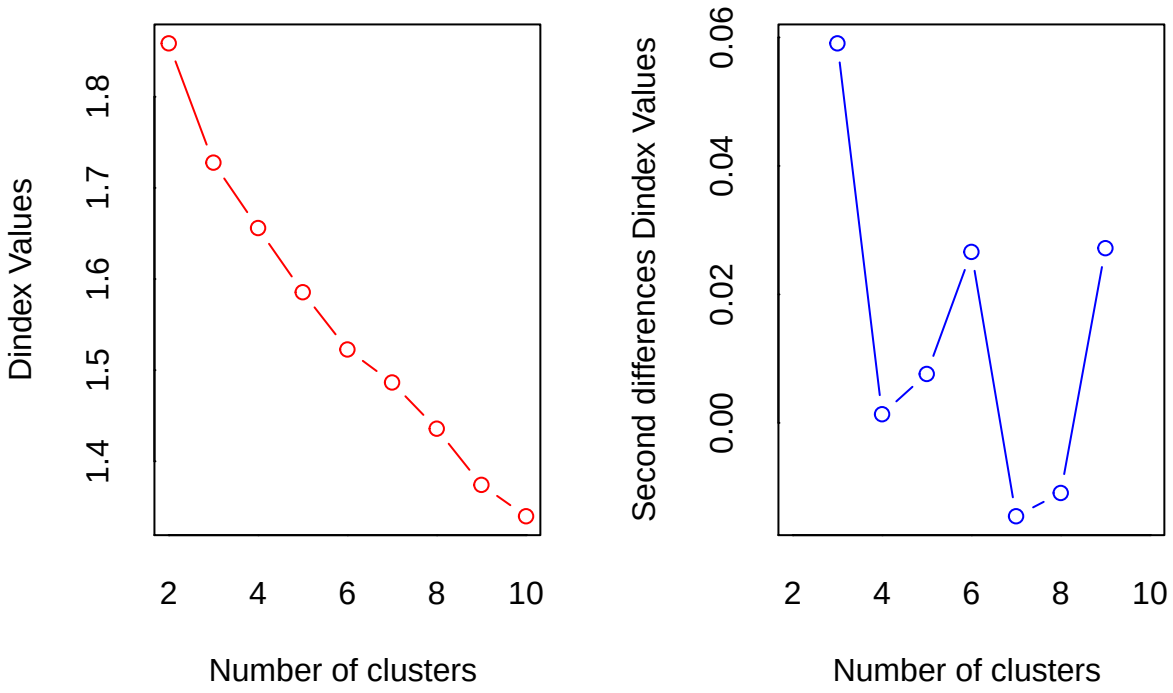
```
## [1] 0.639
```

Ward's (minimum deviance) method

```
library(NbClust)
nb <- NbClust(heart_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.
##
```

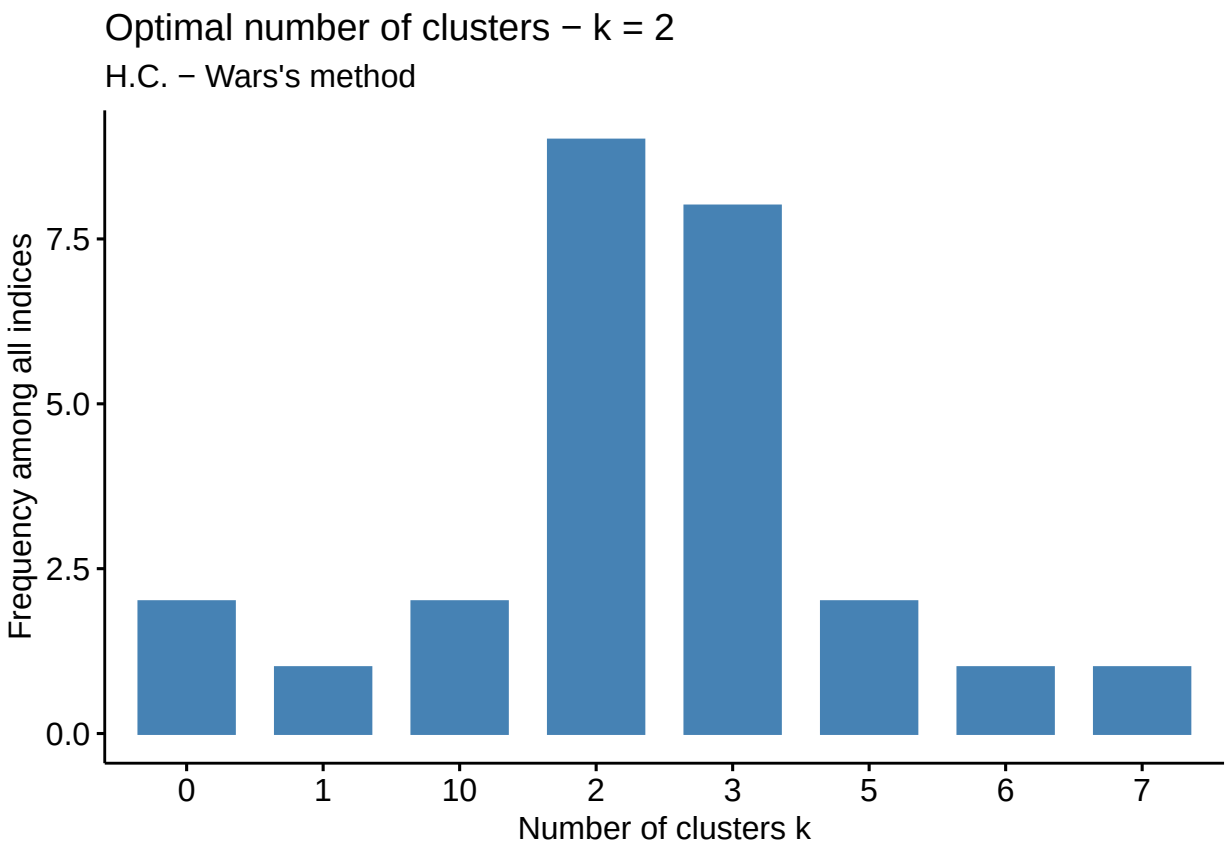



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##       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

```
library(factoextra)
fviz_nbclust(nb)+labs(subtitle = "H.C. - Wars'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
## * 1 proposed  1 as the best number of clusters
## * 9 proposed  2 as the best number of clusters
## * 8 proposed  3 as the best number of clusters
## * 2 proposed  5 as the best number of clusters
## * 1 proposed  6 as the best number of clusters
## * 1 proposed  7 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 .
```



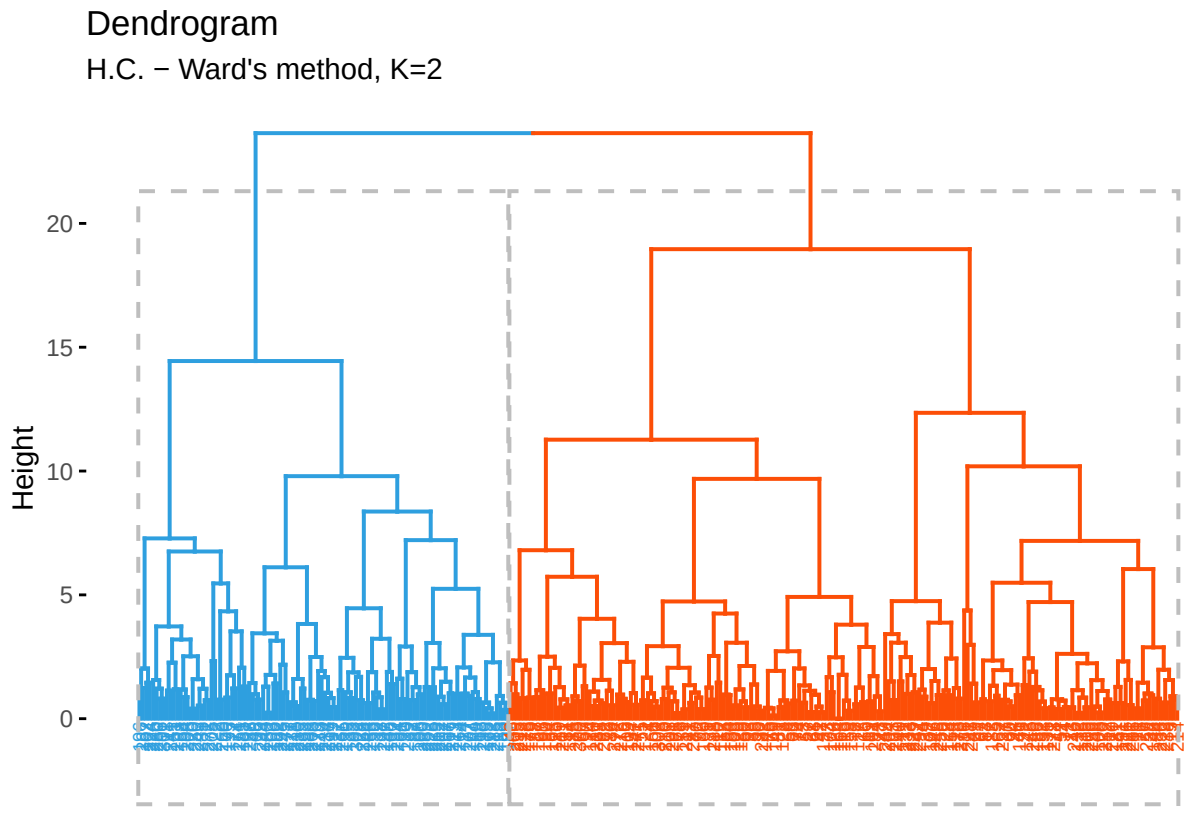

```r
hc <- hclust(dist.eucl, method = "ward.D2")
grp <- cutree(hc, k=2)
table(grp)
```

```
## grp
##   1   2
## 108 195
```

```
grp
```

```
##   [1] 1 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2 1 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2
## [112] 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 1 1 1 2 2 2 1 1 2 2 2
## [149] 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 1 1 1 2 1 2 2 1 2 1 2 1 1
## [186] 2 1 1 2 2 2 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1 1 2 2 2 1 2 2 1 2 1 1 1 2 2 1
## [223] 2 1 1 1 1 2 2 1 2 2 2 1 1 2 2 2 1 2 1 2 1 1 1 2 2 2 2 1 1 2 1 1 2 2 1 1 2
## [260] 1 2 2 1 1 1 1 1 1 1 1 2 1 1 2 2 2 1 2 2 1 2 2 1 2 1 2 2 2 1 1 2 1 1 2 2 1
## [297] 1 1 1 2 1 1 2
```

```
fviz_dend(hc, k = 2, cex = 0.5, k_colors = c("#2E9FDF", "#FC4E07"),  color_labels_by_k = TRUE, rect = TR
subtitle = "H.C. - Ward's method, K=2", cex.subtitle= 0.5)
```

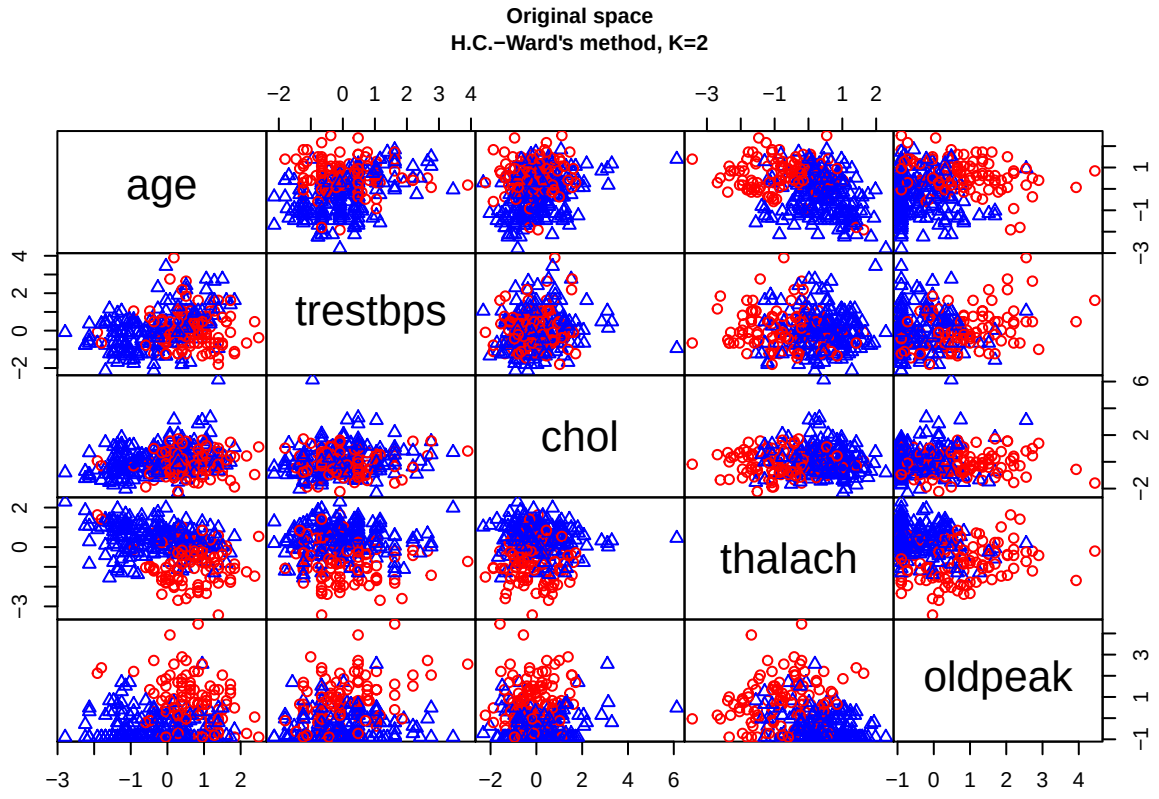



```
cor(dist.eucl, cophenetic(hc))
```

```
## [1] 0.4229219
```

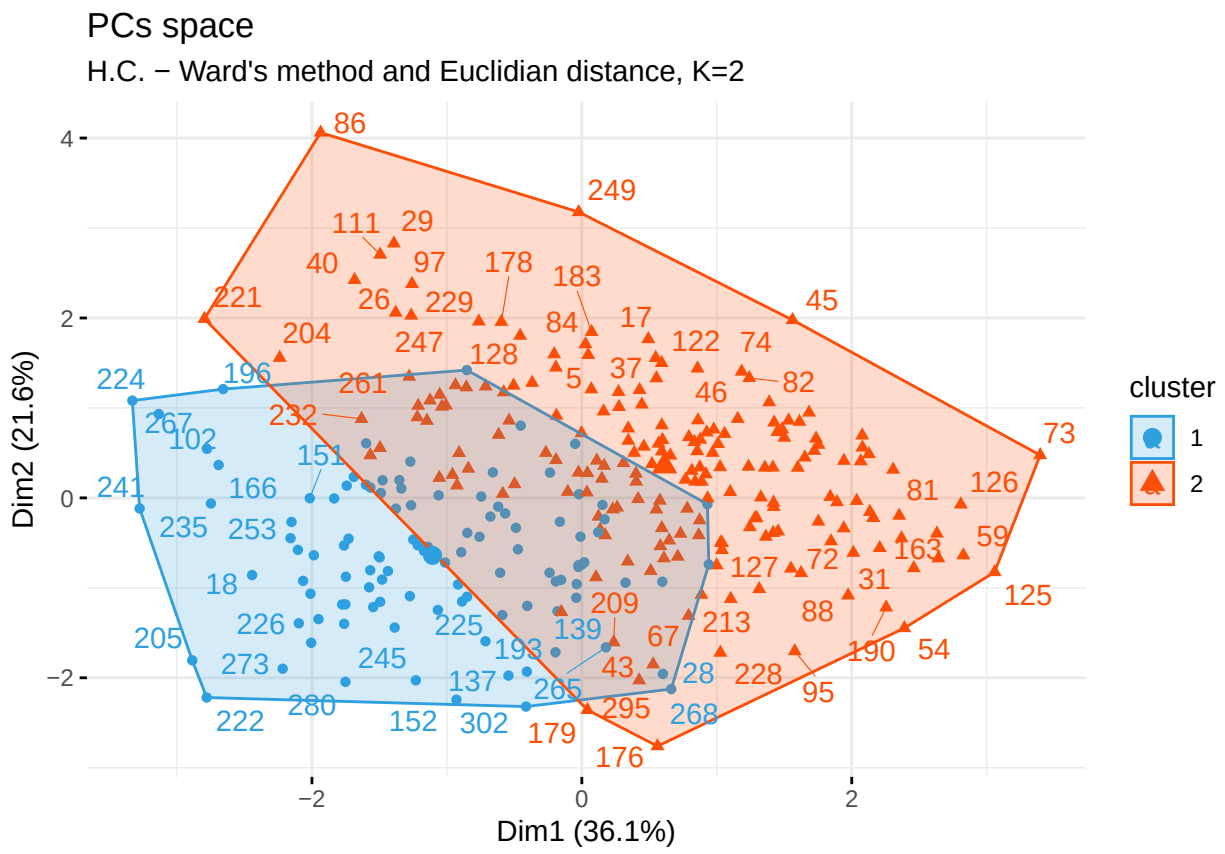
According to the function "NbClust", the best number of clusters, applying hierarchical clustering method and using Ward's method and Euclidean distance, is 2: cluster 1 with 108 units, cluster 2 with 195 units.

```
pairs(heart_scale, gap=0, pch=grp, cex.main= 0.7, main="Original space\nH.C.-Ward's method, K=2", col=c
```




```
fviz_cluster(list(data = heart_scale, cluster = grp), palette = c("#2E9FDF", "#FC4E07"), ellipse.type =
  labs(subtitle = "H.C. - Ward's method and Euclidian distance, K=2", cex.sub= 0.5)
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



To evaluate the goodness of clustering algorithm results, internal and external validation measures will be analysed

Internal validation measures: silhouette width and Dunn index

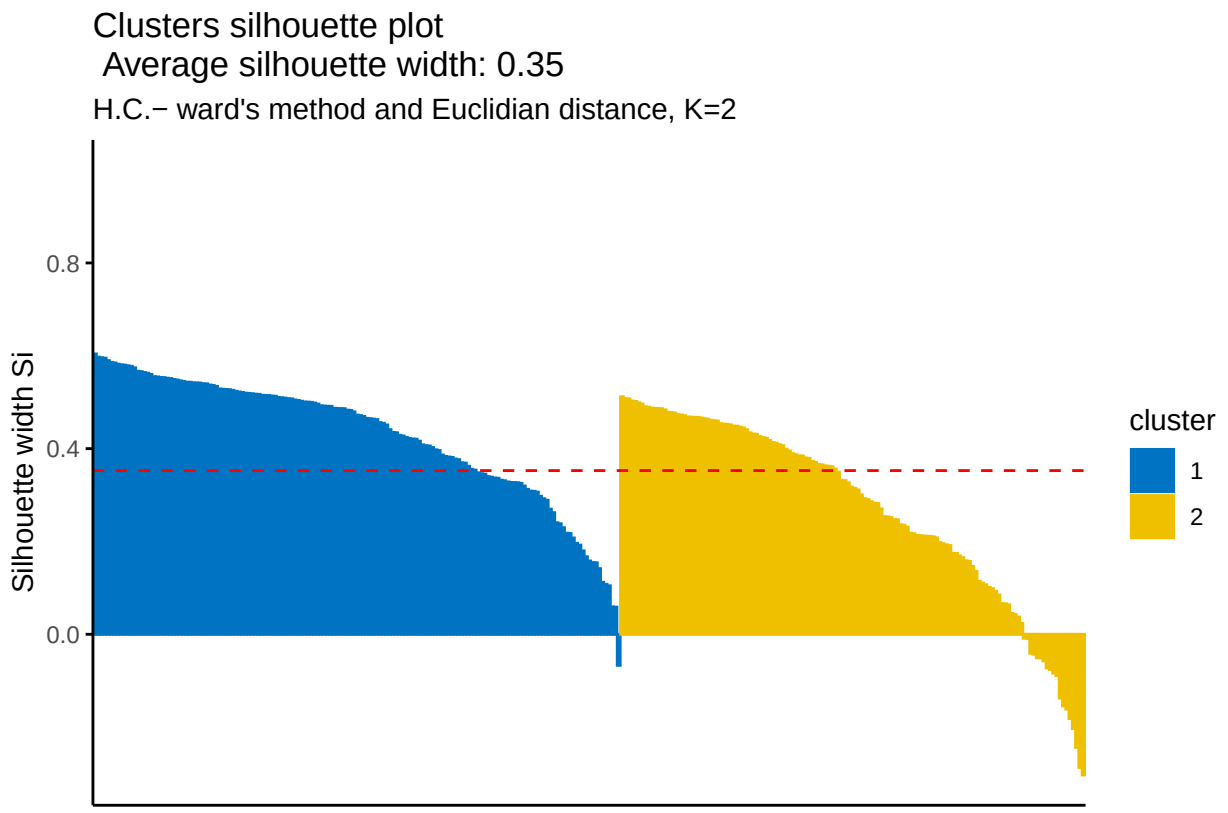
Silhouette width

```
hclust<- eclust(heart_sub,k=2 ,"hclust",hc_method  = "ward.D2",nboot = 50)
silinfo <- hclust$silinfo
silinfo$avg.width
```

```
## [1] 0.3526915
```

```
fviz_silhouette(hclust, palette = "jco", ggtheme = theme_classic())+labs(
subtitle = "H.C.- ward's method and Euclidian distance, K=2", cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  161          0.43
## 2       2  142          0.27
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.4270232 0.2684140
```

```
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
## 250       1        2 -0.066885556
## 157       2        1 -0.008557270
## 55        2        1 -0.008827214
## 234       2        1 -0.040769677
## 242       2        1 -0.042984024
## 90        2        1 -0.050553490
## 76        2        1 -0.050740606
## 215       2        1 -0.057300439
## 81        2        1 -0.072542991
## 2         2        1 -0.075882126
## 200       2        1 -0.083605413
## 273       2        1 -0.088944493
## 100       2        1 -0.137401746
## 42        2        1 -0.154300627
## 77        2        1 -0.161294500
## 109       2        1 -0.181702419
## 225       2        1 -0.203123906
## 161       2        1 -0.243760798
```

```
## 4         2        1 -0.286929176
## 303       2        1 -0.303156145
```

The value of complete silhouette width indicates that on average the units are well enough clustered. As in particular, in each cluster the units are on average the same silhouette value with respect to the silhouette width. According to this index, 20 units that belong to cluster 1 are not well clustered: they should belong to cluster 2.

Dunn index

```
library(fpc)
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.04426994
```

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

External validation measures: confusion matrix, correct Rand index,

Meila's IV index ### 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(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  42  54
##   1 119  88
```

A in "female" group of sex data has classified almosy equal in two clusters. For "male" sex (n=117) classified in cluster 1 while cluster 2 has 88 values.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.01681322
```

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

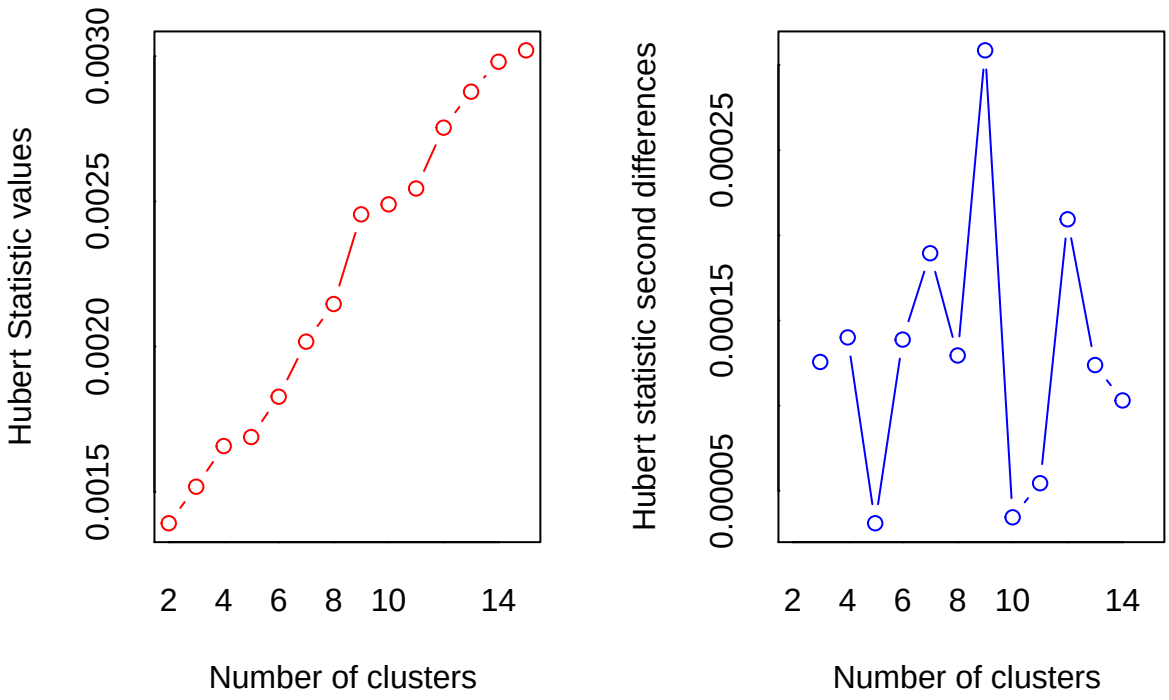
```
stats$vi
```

```
## [1] 1.29923
```

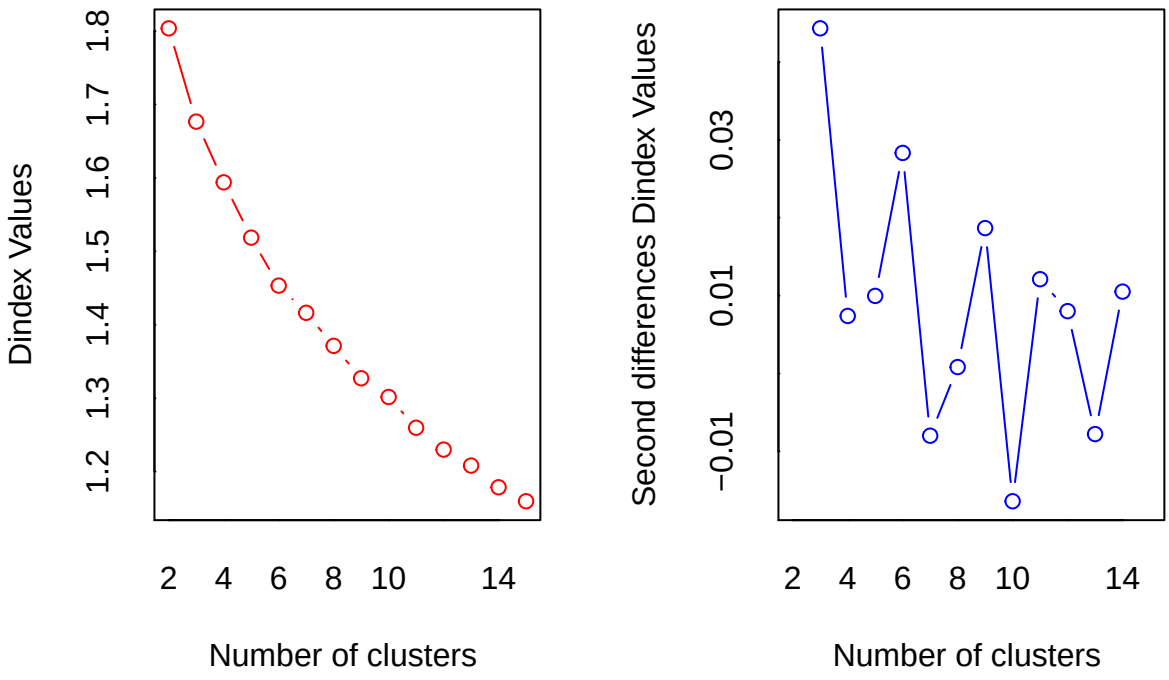
Partitional method

K-means

```
library(NbClust)
library(ggplot2)
nb <- NbClust(heart_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.
##
```

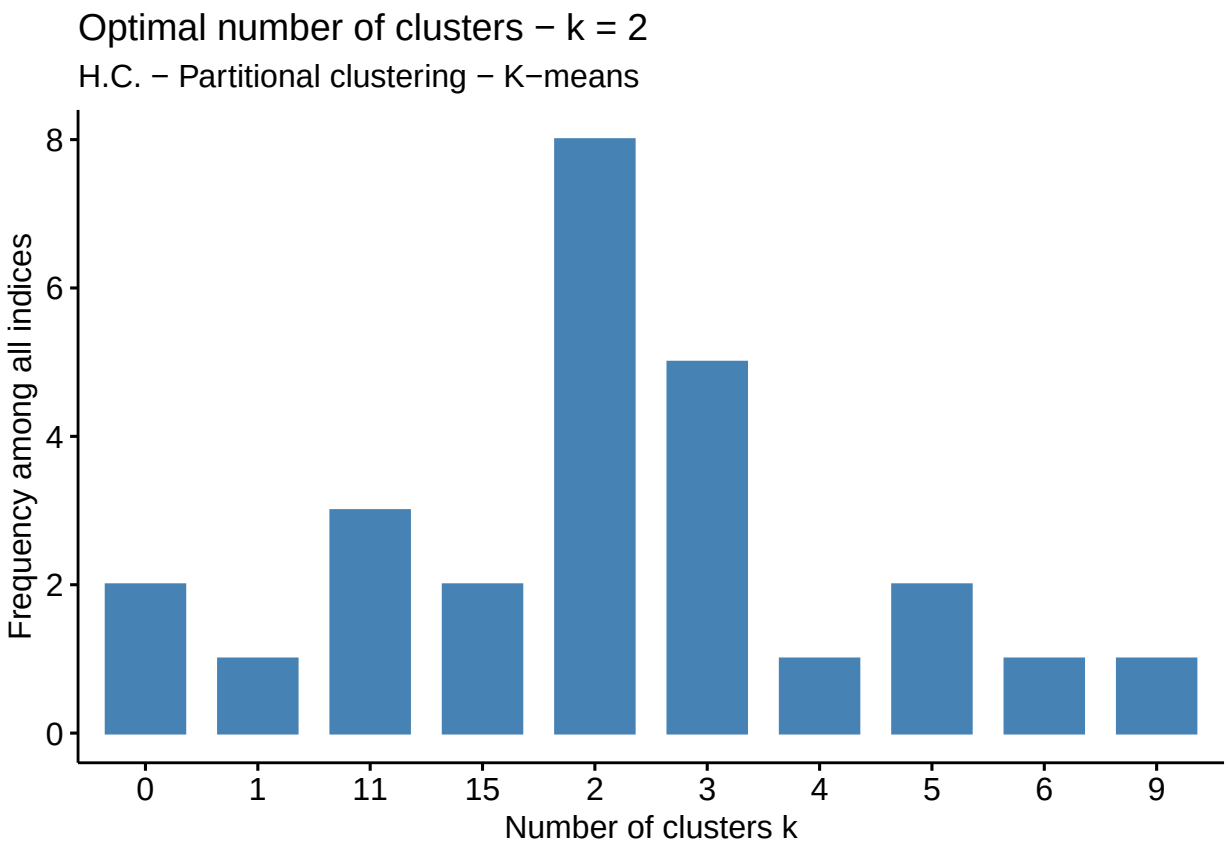



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 8 proposed 2 as the best number of clusters
## * 5 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 2 proposed 5 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 1 proposed 9 as the best number of clusters
## * 3 proposed 11 as the best number of clusters
## * 2 proposed 15 as the best number of clusters
##
##                   ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

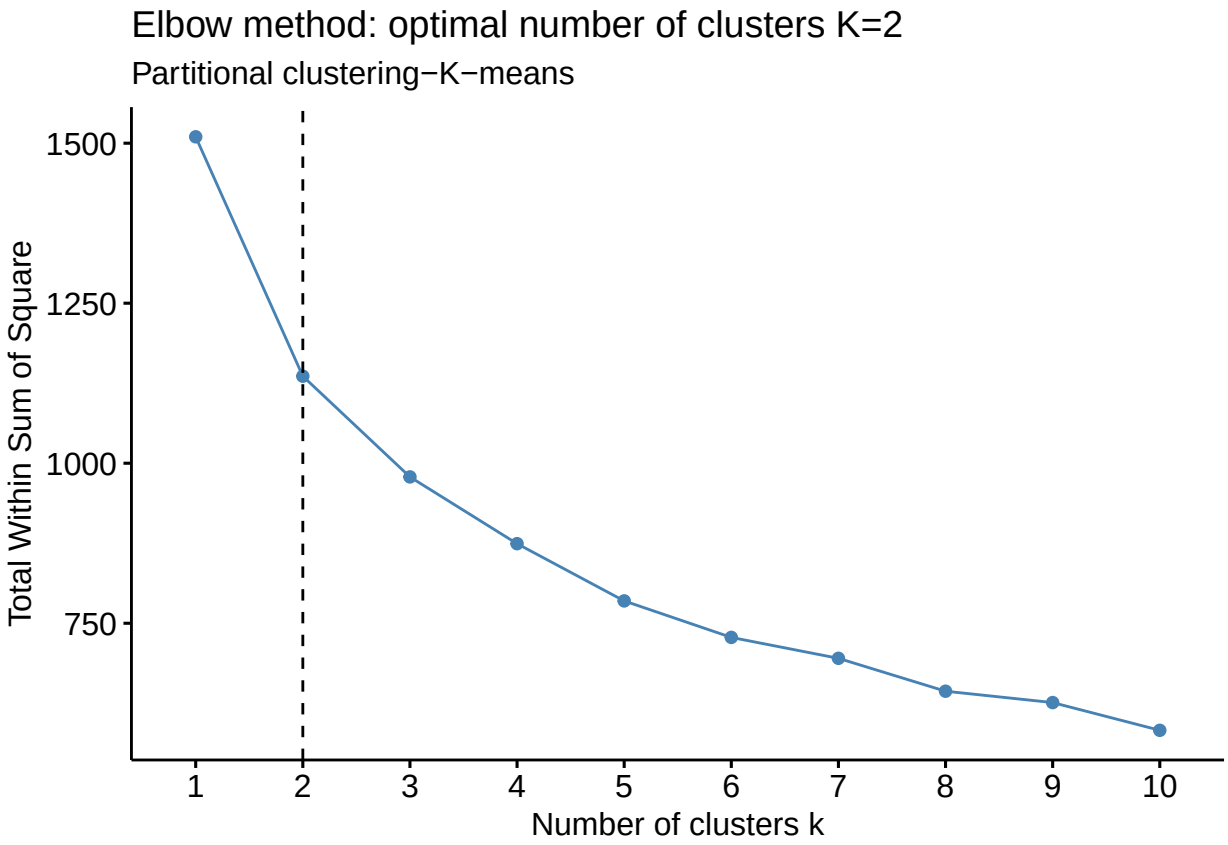
```
fviz_nbclust(nb)+labs(subtitle = "H.C. - Partitional clustering - K-means")
```

```
## 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
## * 1 proposed  1 as the best number of clusters
## * 8 proposed  2 as the best number of clusters
## * 5 proposed  3 as the best number of clusters
## * 1 proposed  4 as the best number of clusters
## * 2 proposed  5 as the best number of clusters
## * 1 proposed  6 as the best number of clusters
## * 1 proposed  9 as the best number of clusters
## * 3 proposed  11 as the best number of clusters
## * 2 proposed  15 as the best number of clusters
##
## Conclusion
## =========================
## * According to the majority rule, the best number of clusters is  2 .
```



```
fviz_nbclust(heart_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(heart_scale, 2, nstart = 25))
```

```
## K-means clustering with 2 clusters of sizes 172, 131
##
## Cluster means:
##          age   trestbps       chol    thalach    oldpeak
## 1 -0.5415776 -0.3355766 -0.2416645  0.5202796 -0.4537289
## 2  0.7110790  0.4406044  0.3173000 -0.6831152  0.5957356
##
## Clustering vector:
##   [1] 2 1 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1 2 1 2 1 1 1 2 1 2 2 1 2 1 1 1 1 1 2 1 1
##  [38] 1 2 2 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 2 2 1 1 1 2
## [112] 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 2 1 2 2 1 1 1 1 2 2 1 1
## [149] 1 1 2 2 2 2 1 2 1 1 1 1 1 2 1 1 1 2 2 2 2 2 1 1 2 2 2 1 1 2 1 2 2 2 1 1 2
## [186] 1 2 2 1 1 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 1 1 2 1 1 1 2 1 2 2 2 2 2 2 1 2 2
## [223] 2 2 2 2 2 1 2 2 1 2 2 2 2 2 1 1 2 2 1 2 2 2 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2
## [260] 1 2 1 2 1 1 2 2 1 2 2 1 2 2 1 1 1 2 1 2 2 2 1 2 1 2 2 1 1 2 2 1 2 2 2 1 2
## [297] 1 2 2 1 2 2 1
##
## Within cluster sum of squares by cluster:
```

```
## [1] 512.4914 623.1961
##  (between_SS / total_SS =  24.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
aggregate(heart_sub, by=list(cluster=km.res$cluster), mean)
```

```
##   cluster      age trestbps     chol thalach   oldpeak
## 1       1 49.44767 125.7384 233.7384 161.564 0.5127907
## 2       2 60.82443 139.3511 262.7099 134.000 1.7312977
```

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

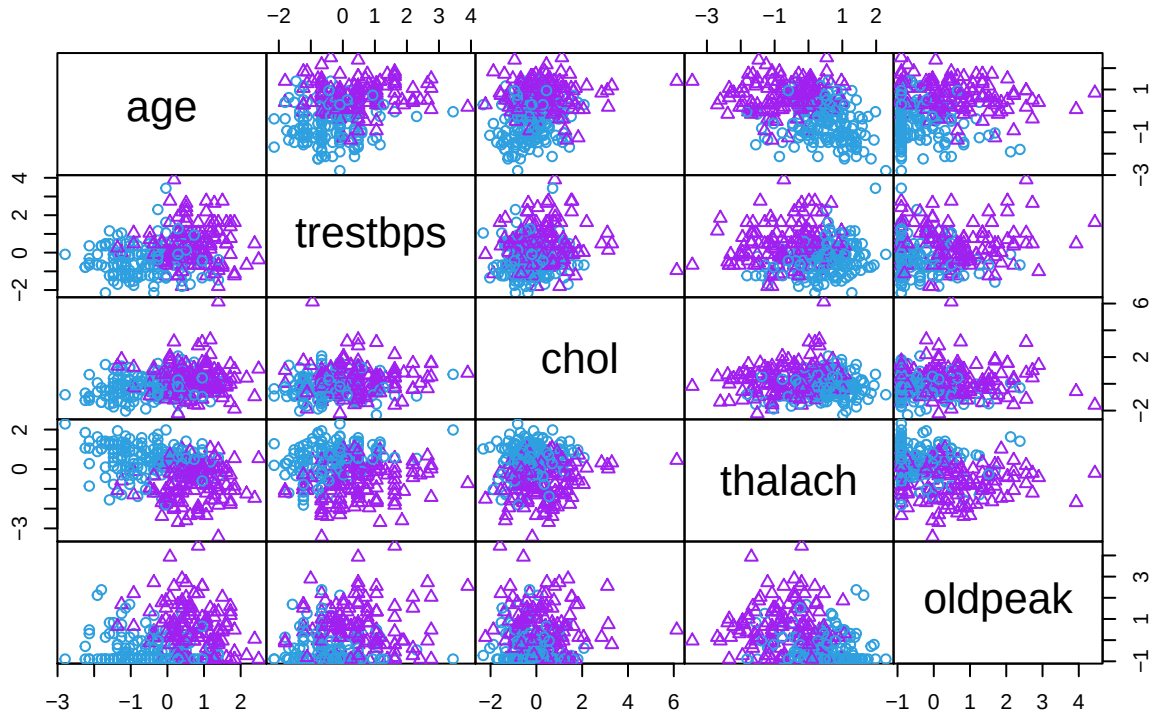
```
dd <- cbind(heart_sub, cluster = km.res$cluster)
head(dd)
```

```
##   age trestbps chol thalach oldpeak cluster
## 1  63      145  233     150     2.3       2
## 2  37      130  250     187     3.5       1
## 3  41      130  204     172     1.4       1
## 4  56      120  236     178     0.8       1
## 5  57      120  354     163     0.6       1
## 6  57      140  192     148     0.4       1
```

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

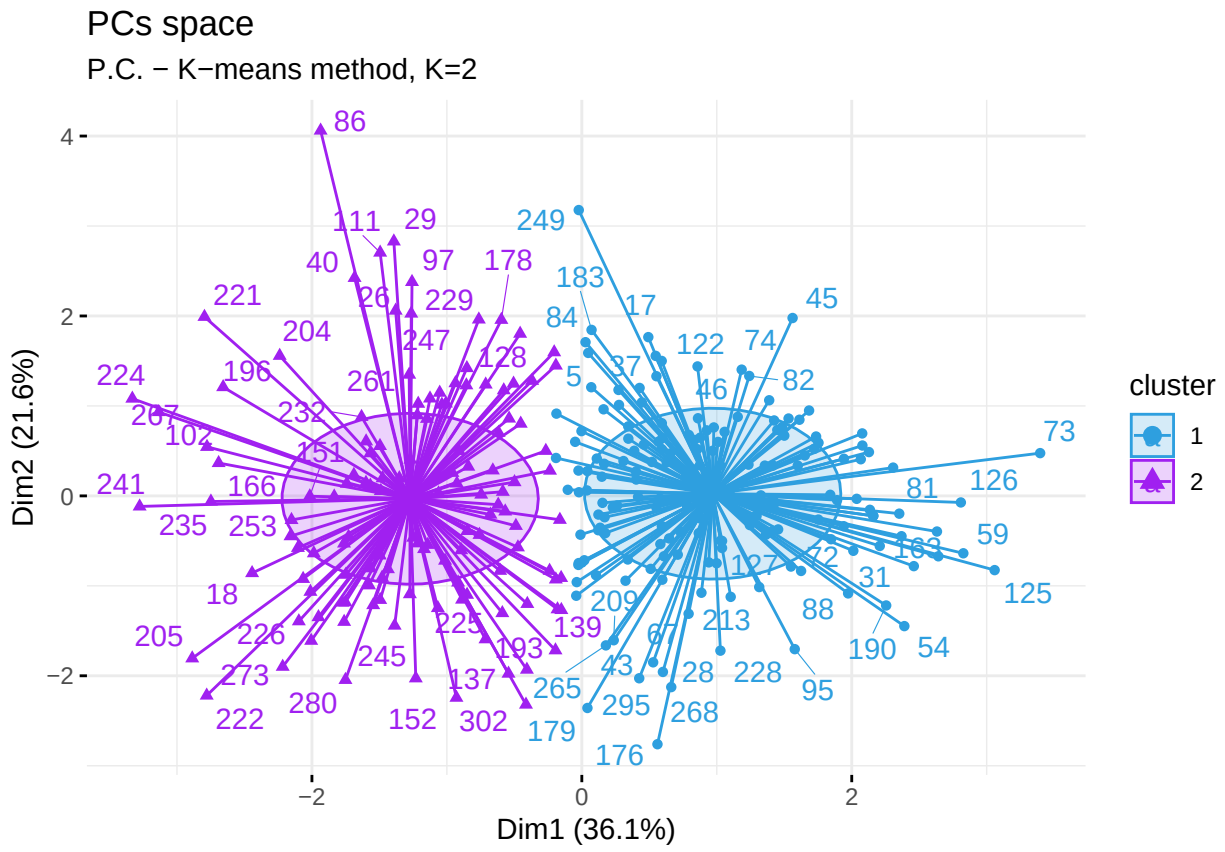
```
## cl
##   1   2
## 172 131
```

```
pairs(heart_scale, gap=0, pch=cl, main="Original space\nP.C.-K-means method, K=2", cex.main= 1, col=c("
```




```
fviz_cluster(km.res, data = heart_scale, palette = c("#2E9FDF", "purple"),
             ellipse.type = "euclid", star.plot = TRUE, repel = TRUE,
             main= "PCs space", ggtheme = theme_minimal())+
  labs(subtitle = "P.C. - K-means method, K=2")
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



According to the function "NbClust", the best number of clusters, applying partitional clustering method and using K-means method is 2: cluster 1 with 172 units, cluster 2 with 131 units. The ariability between different clusters: 24.8% of the total variability is explained by the separation between clusters. In the PCs space, there is not separation between clusters.

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

Internal validation measures: silhouette width and Dunn index

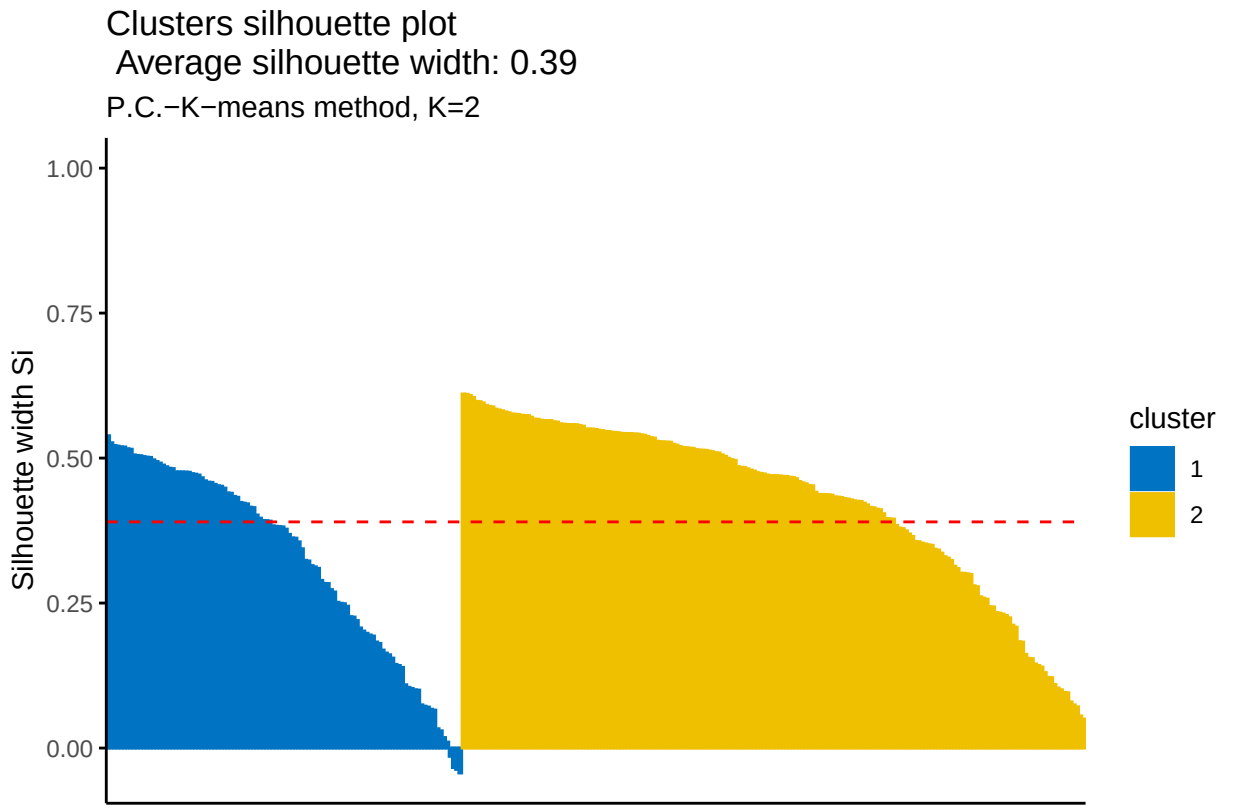
Silhouette width

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

```
## [1] 0.3899835
```

```
fviz_silhouette(hclust, palette = "jco", ggtheme = theme_classic())+
  labs(subtitle = "P.C.-K-means method, K=2")
```

```
##   cluster size ave.sil.width
## 1       1  110          0.32
## 2       2  193          0.43
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.3227643 0.4282950
```

```
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
## 8         1        2 -0.01436074
## 202       1        2 -0.03369483
## 208       1        2 -0.03692880
## 74        1        2 -0.04306885
```

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) also the units are on average the same silhouette value with respect to silhouette width. According to this index, 4 units (8, 202, 208, 74)that belong to cluster 1 are not well clustered: they should belong to cluster 2.

Dunn index

```
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.05015525
```

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

External validation measures: confusion matrix, correct Rand index,

Meila's IV index ### Confusion matrix

```
table(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  46  50
##   1  64 143
```

According to the Confusion matrix, there is not a perfect agreement between the nominal variable Sex and the cluster solution. A large number of "male" sex (n = 143) has been classified in cluster 2 but Some of them (n = 64) have been classified in cluster 1. For "female" sex data have been classified almost equally between clusters.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.04917225
```

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.

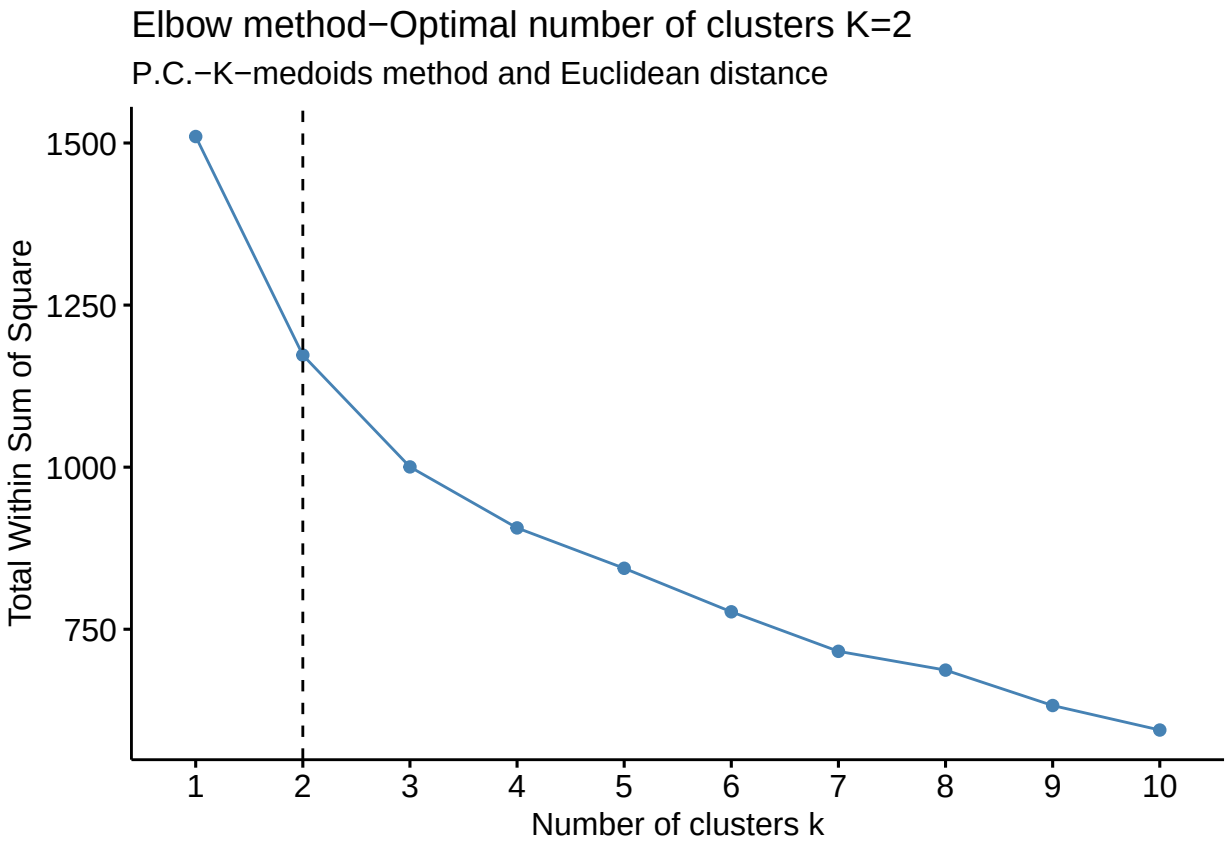
Meila's VI Index

```
stats$vi
```

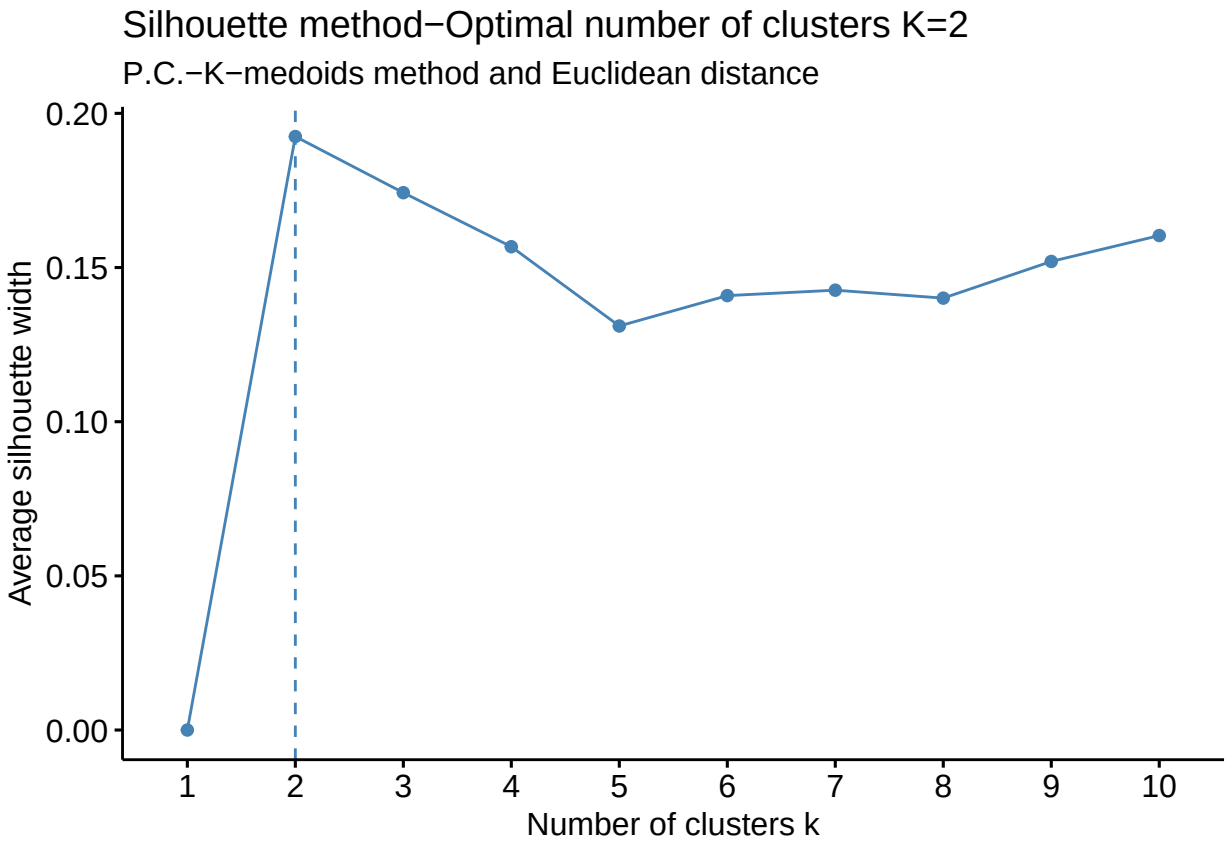
```
## [1] 1.252985
```

Partitioning around medoids (PAM) method and Euclidean distance

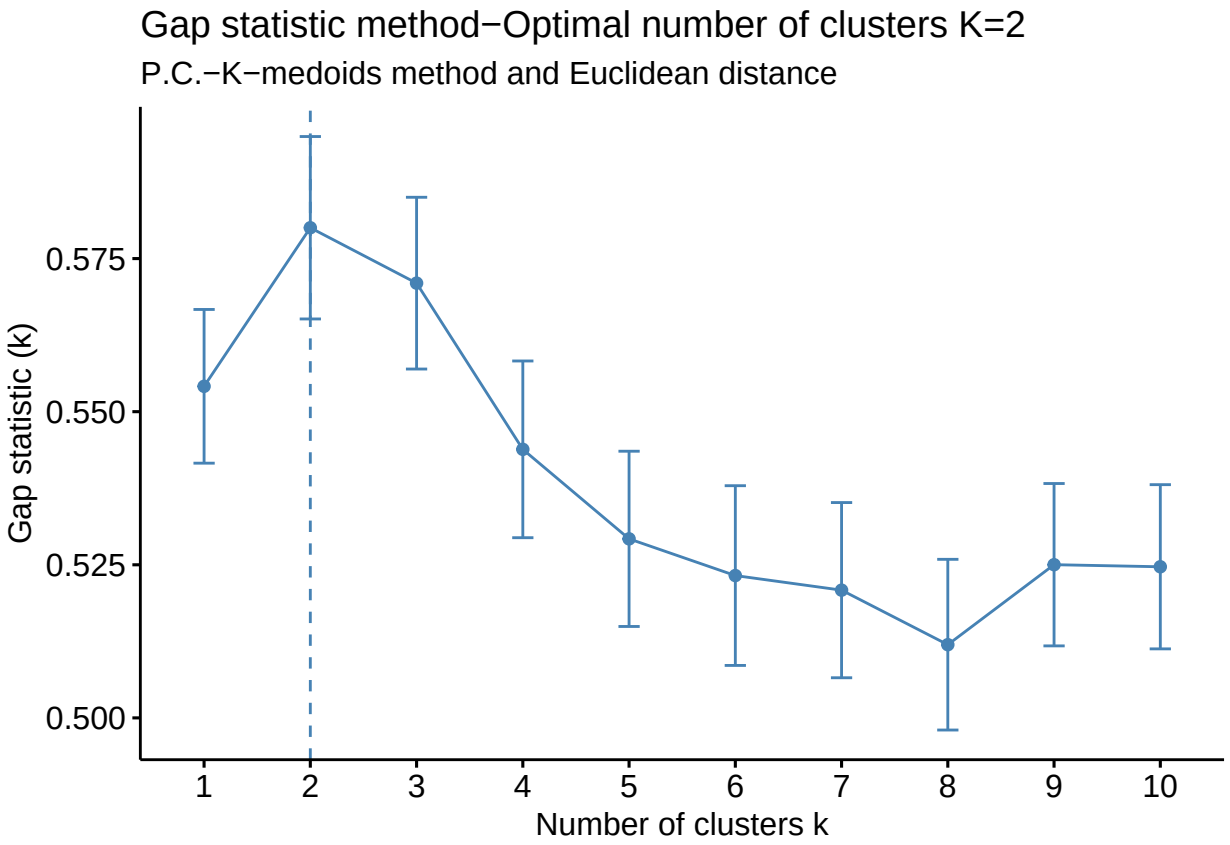
```
fviz_nbclust(heart_scale, cluster::pam, method = "wss") +
  geom_vline(xintercept = 2, linetype = 2)+
  labs(title = "Elbow method-Optimal number of clusters K=2",
       subtitle="P.C.-K-medoids method and Euclidean distance", cex.sub= 0.5)
```




```
fviz_nbclust(heart_scale, cluster::pam, method = "silhouette")+
labs(title = "Silhouette method-Optimal number of clusters K=2",
     subtitle="P.C.-K-medoids method and Euclidean distance", cex.sub= 0.5)
```




```
fviz_nbclust(heart_scale, cluster::pam, method = "gap_stat", nboot = 500)+
labs(title = "Gap statistic method-Optimal number of clusters K=2",
     subtitle="P.C.-K-medoids method and Euclidean distance", cex.sub= 0.5)
```



In order to find the optimal number of clusters, three indices were used. The elbow method does not seem to give a very clear result; according to Total within sum of square, the suggested number of clusters is assumed to be 2. Silhouette method suggests 2 clusters. Gap statistics 2 cluster, i.e. no presence of clusters in the data. It was decided to proceed by identifying 2 clusters.

```
library(cluster)
set.seed(123)
(pam.res <- pam(heart_scale, 2, metric = "euclidean"))
```

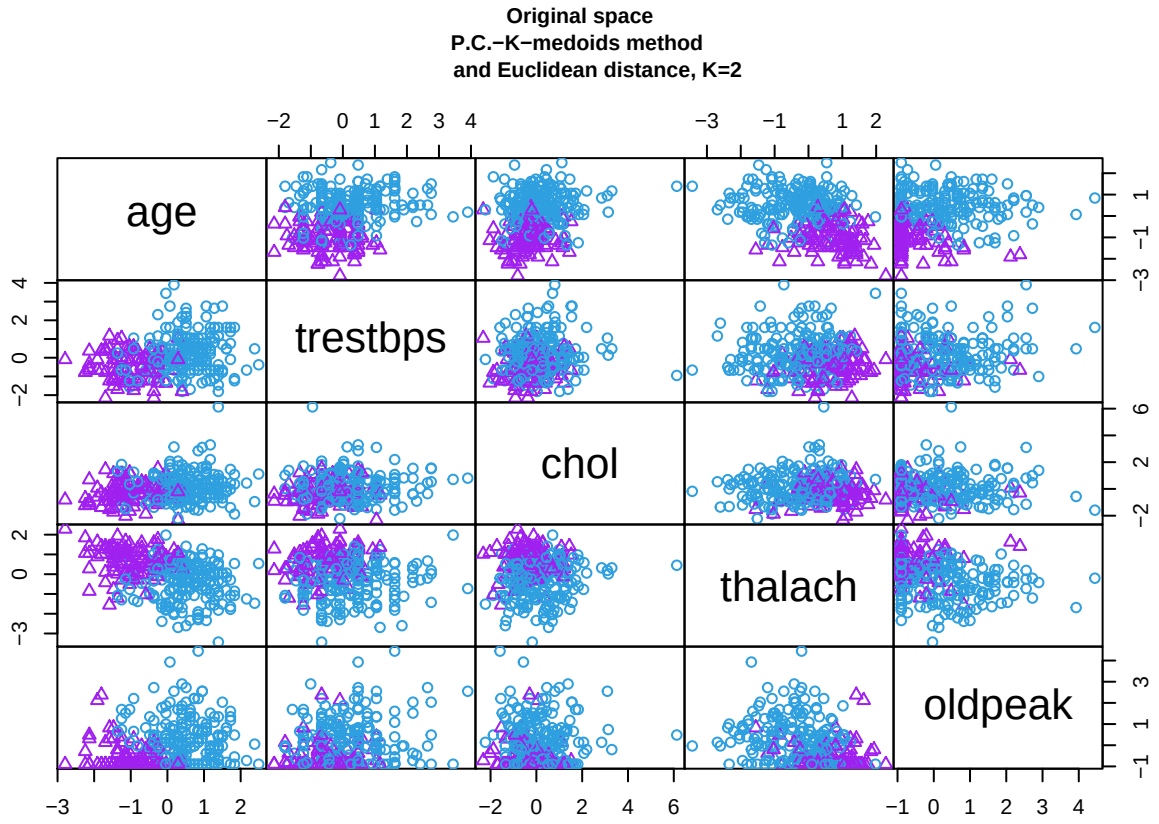
```
## Medoids:
##       ID       age    trestbps       chol    thalach    oldpeak
## [1,] 187  0.620304 -0.09258463  0.1299609 -0.2465324  0.3103986
## [2,] 149 -1.141403 -0.66277043 -0.3909653  0.8449247 -0.8953805
## Clustering vector:
##   [1] 1 2 2 2 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1 1 2 2 1 2 1 1 1 1 1 2 1 2 1 1 2 2
##  [38] 1 1 1 1 2 1 1 2 2 2 2 1 2 1 1 1 2 1 2 2 2 2 1 1 2 2 2 1 2 2 2 2 1 1 2 2 2
##  [75] 2 1 1 1 2 1 2 2 1 1 2 1 1 2 1 1 2 2 2 1 2 1 1 2 1 2 2 1 1 2 2 1 1 2 2 2 1
## [112] 2 1 2 1 2 2 1 2 2 1 1 2 2 2 2 2 1 2 1 1 2 2 2 2 2 1 1 1 1 1 2 2 1 1 1 2 1
## [149] 2 2 1 1 1 1 2 1 2 2 1 2 2 1 2 2 2 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1
## [186] 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1
## [223] 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1
## [260] 2 1 2 1 1 1 1 1 2 1 1 2 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
## [297] 1 1 1 1 1 1 2
## Objective function:
##    build     swap
## 1.900356 1.852010
##
```

```
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```
pam.res$clusinfo
```

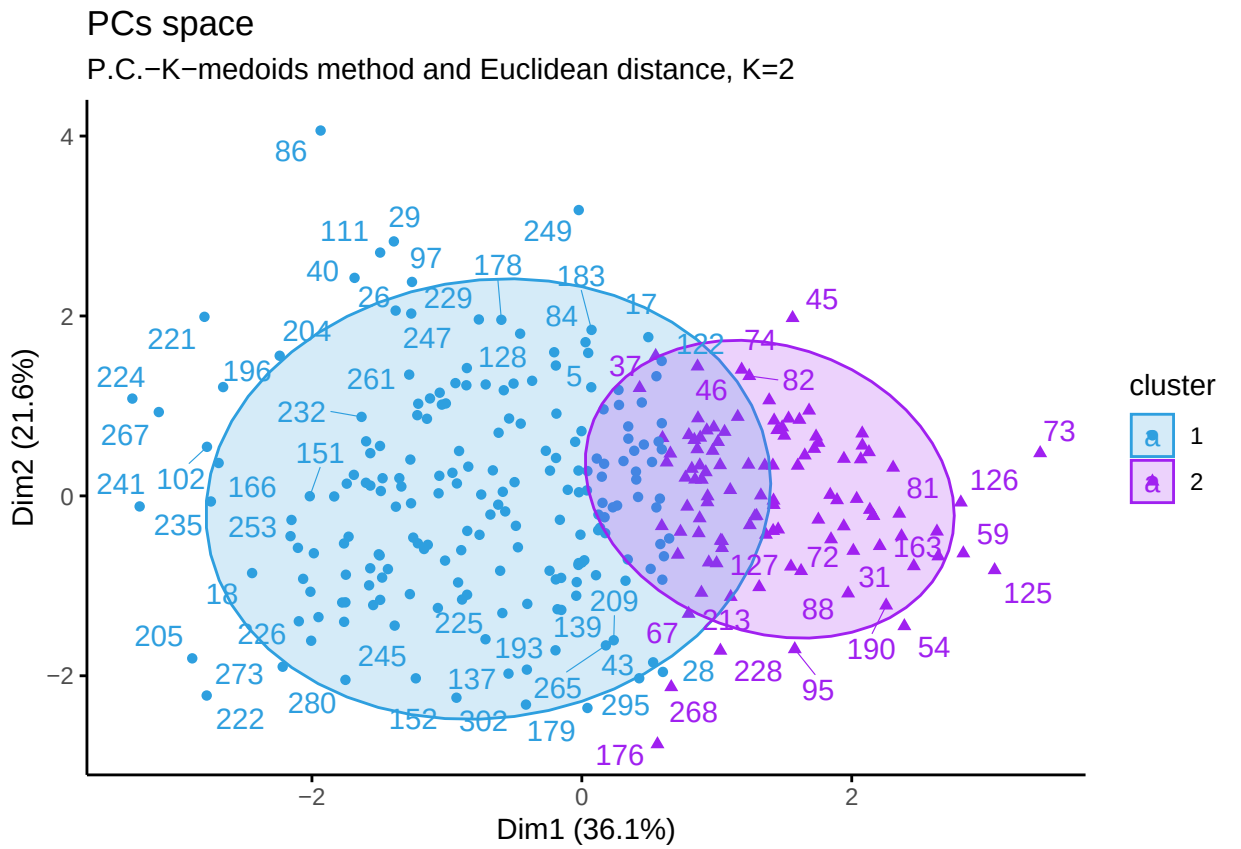
```
##      size max_diss  av_diss diameter separation
## [1,]  198 6.151972 2.047876 9.086823  0.3970971
## [2,]  105 3.388109 1.482662 4.852517  0.3970971
```

```
cc <- pam.res$cluster
pairs(heart_scale, gap=0, pch=cc, main="Original space\nP.C.-K-medoids method
      and Euclidean distance, K=2", cex.main= 0.7,
      col=c("#2E9FDF","purple")[cc])
```




```
fviz_cluster(pam.res, palette = c("#2E9FDF", "purple"), ellipse.type = "t",
repel = TRUE, main= "PCs space", ggtheme = theme_classic())+labs(
subtitle = "P.C.-K-medoids method and Euclidean distance, K=2", cex.sub= 0.5)
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Applying partitioning clustering method and using PAM algorithm and Euclidean distance, two clusters are composed in this way: cluster 1 with 176 units, cluster 2 with 16 units.

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

Internal validation measures: silhouette width and Dunn index

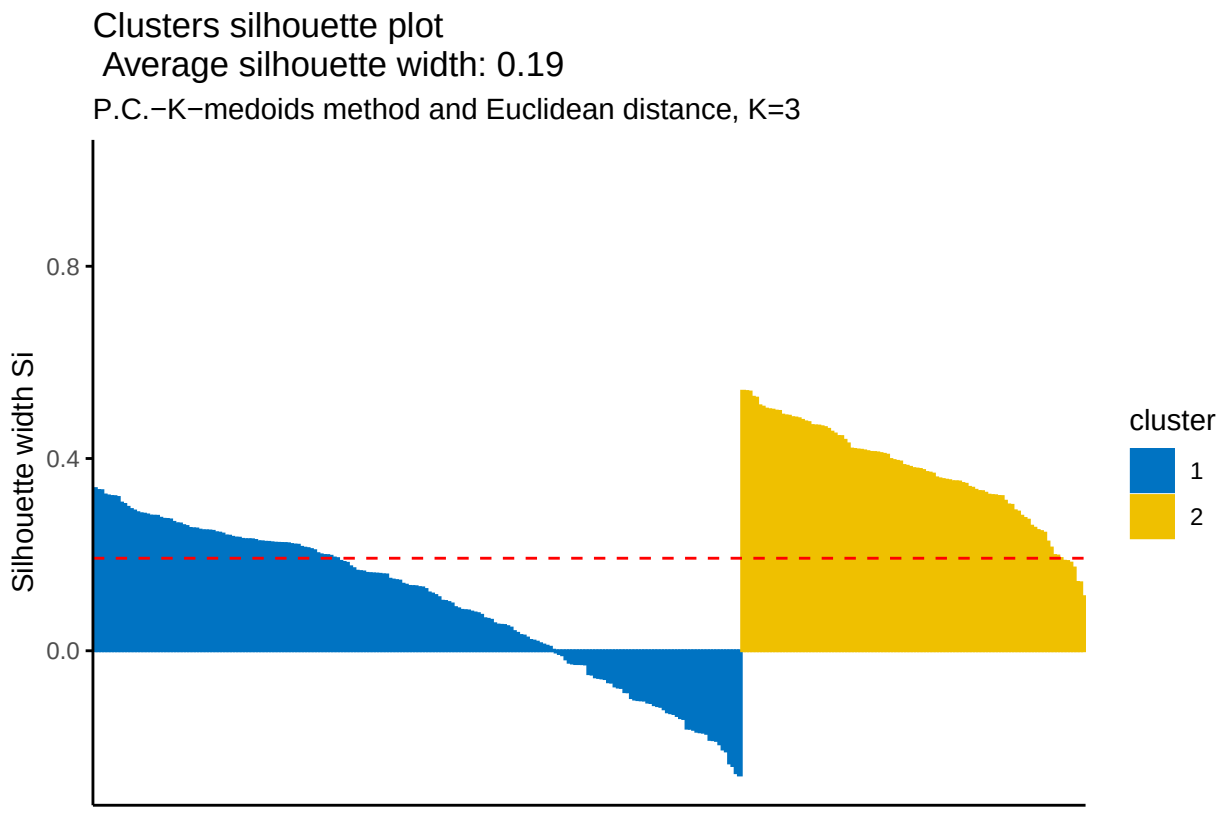
Silhouette width

```
hclust<- eclust(heart_sub,k=2 ,"pam", graph = FALSE, hc_metric = "euclidean")
silinfo <- hclust$silinfo
silinfo$avg.width
```

```
## [1] 0.3500083
```

```
fviz_silhouette(pam.res, palette = "jco",
ggtheme = theme_classic())+labs(
subtitle = "P.C.-K-medoids method and Euclidean distance, K=3", cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  198          0.10
## 2       2  105          0.37
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.4406961 0.2656743
```

```
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
## 252       2        1 -0.01477539
## 200       2        1 -0.01942204
## 146       2        1 -0.02535151
## 81        2        1 -0.03403557
## 2         2        1 -0.03758498
## 273       2        1 -0.05250158
## 19        2        1 -0.05508691
## 185       2        1 -0.05786452
## 24        2        1 -0.06510157
## 259       2        1 -0.07467265
## 100       2        1 -0.08172070
## 299       2        1 -0.08811995
## 77        2        1 -0.09815228
## 42        2        1 -0.11081251
## 101       2        1 -0.11484255
## 109       2        1 -0.11706453
## 120       2        1 -0.11801070
## 231       2        1 -0.12543356
```

```
## 225       2        1 -0.14009810
## 147       2        1 -0.14224913
## 20        2        1 -0.17156592
## 80        2        1 -0.17310467
## 207       2        1 -0.17398031
## 161       2        1 -0.19855278
```

The value of average silhouette width indicates that in average the units are not well enough clustered. In particular, in cluster 1 (blue) the units are in average a higher silhouette value with respect to the silhouette width while in cluster 2 (yellow) the unit have in average a lower value then the total average silhouette. According to this index, 24 units are not well clustered: units that belong to cluster 2, should belong to cluster 1.

Dunn index

```
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.03190583
```

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

External validation measures: confusion matrix, correct Rand index, Meila's

IV index ### Confusion matrix

```
table(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  38  58
##   1 108  99
```

According to the Confusion matrix, there is not a perfect agreement between the nominal variable Sex and the cluster solution. A large number of "female" sex (n = 58) has been classified in cluster 2 but Some of them (n = 38) have been classified in cluster 1. For "male" sex a large number of data (n=108) classified in cluster 1 and (n=99) in cluster 2.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.006139011
```

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.

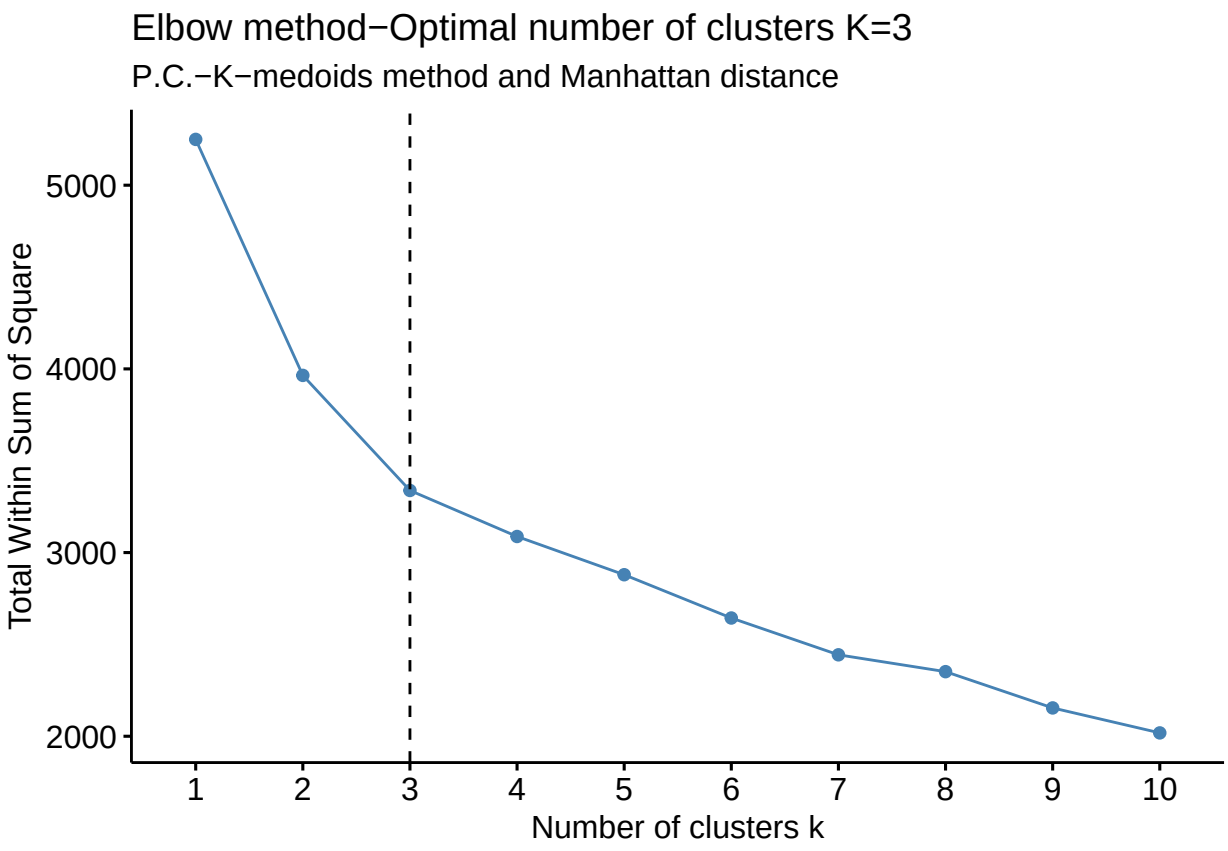
Meila's VI Index

```
stats$vi
```

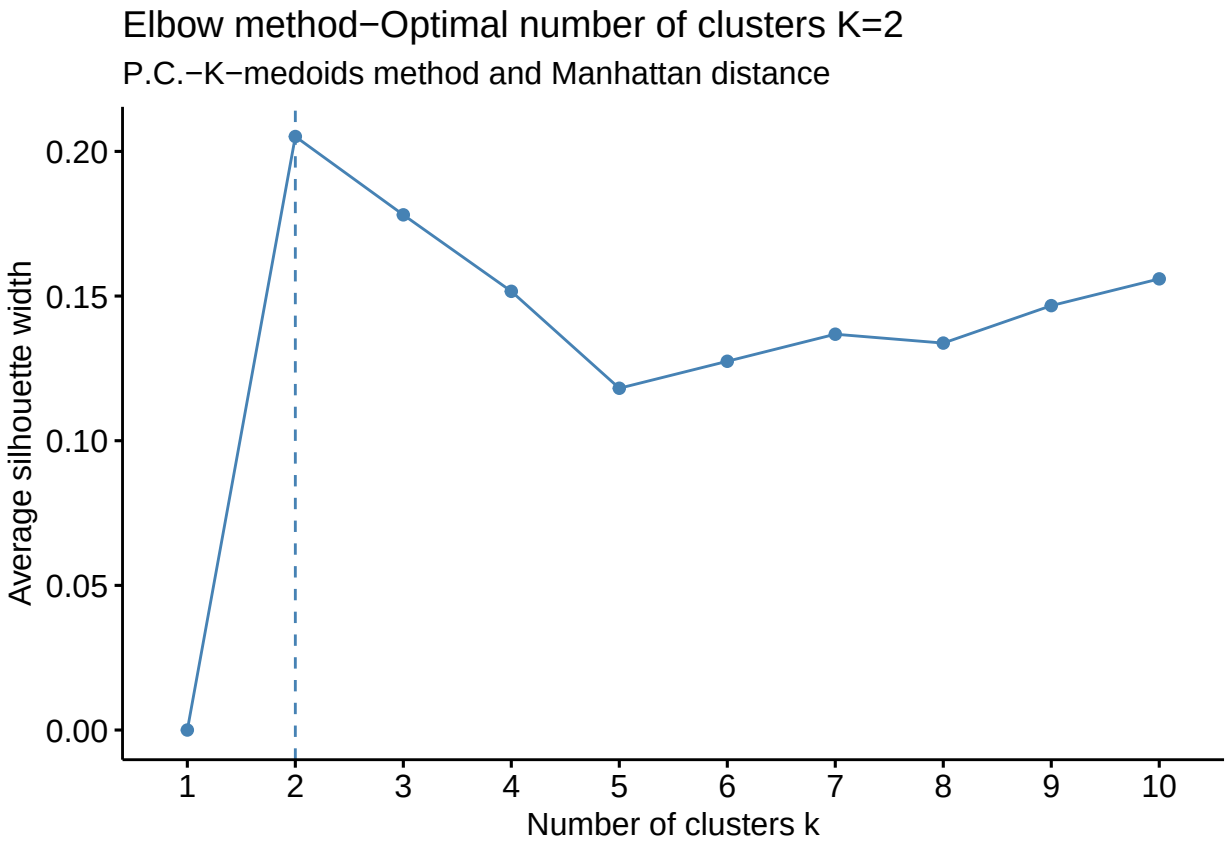
```
## [1] 1.30312
```

Partitioning around medoids (PAM) method and Manhattan distance

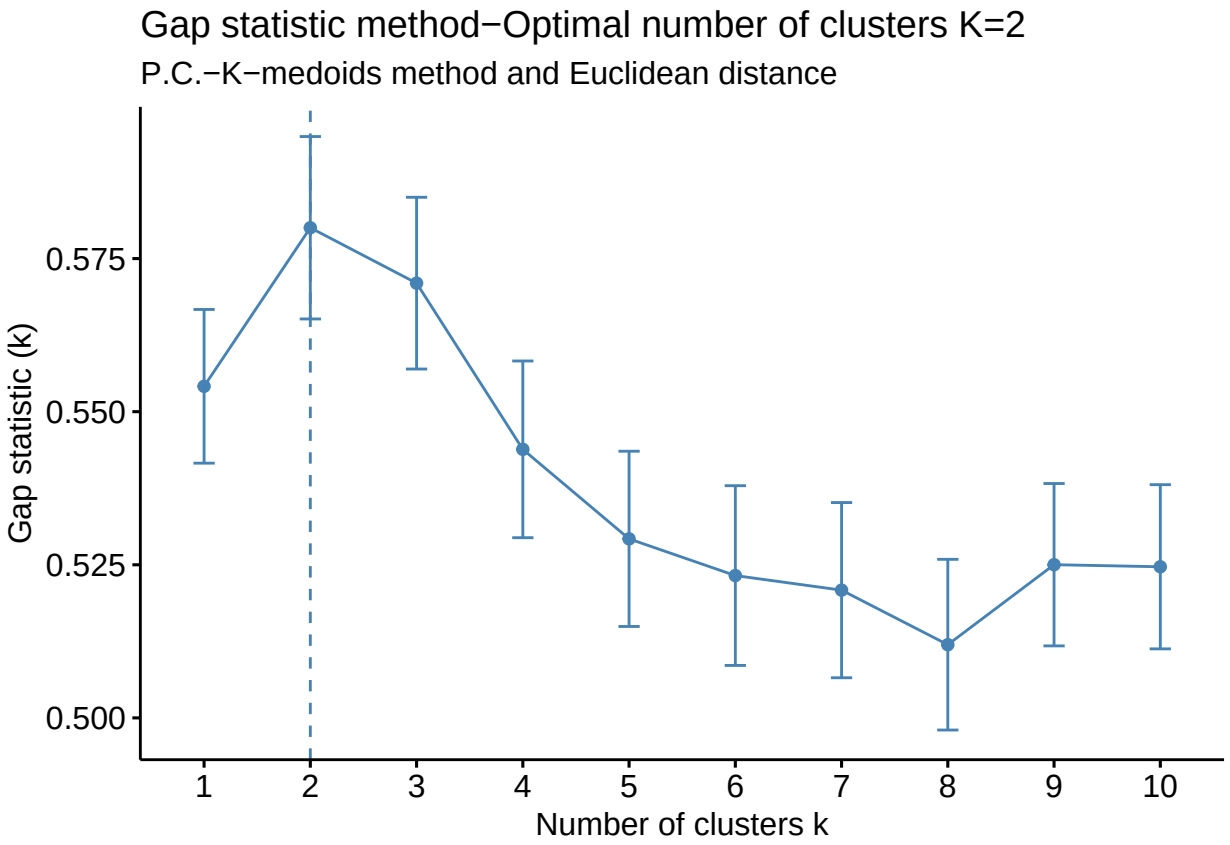
```
set.seed(123)
fviz_nbclust(heart_scale, cluster::pam, method = "wss",
             diss = dist(heart_scale, method = "manhattan"))+
  geom_vline(xintercept = 3, linetype = 2)+ labs(title = "Elbow method-Optimal number of clusters K=3",
subtitle="P.C.-K-medoids method and Manhattan distance", cex.sub= 0.5)
```




```
fviz_nbclust(heart_scale, cluster::pam, method = "silhouette", diss = dist(heart_scale, method = "manha
```




```
set.seed(123)
fviz_nbclust(heart_scale, cluster::pam, method = "gap_stat", nboot = 500, diss=dist(heart_scale, method
```



In order to find the optimal number of clusters, three indices were used. The elbow method does not seem suggest k=3. Silhouette method suggests 2 clusters. Gap statistics 2 cluster.It was decided to proceed by identifying 2 clusters.

```
library(cluster)
set.seed(123)
(pam.res <- pam(heart_scale, 2, metric="manhattan"))
```

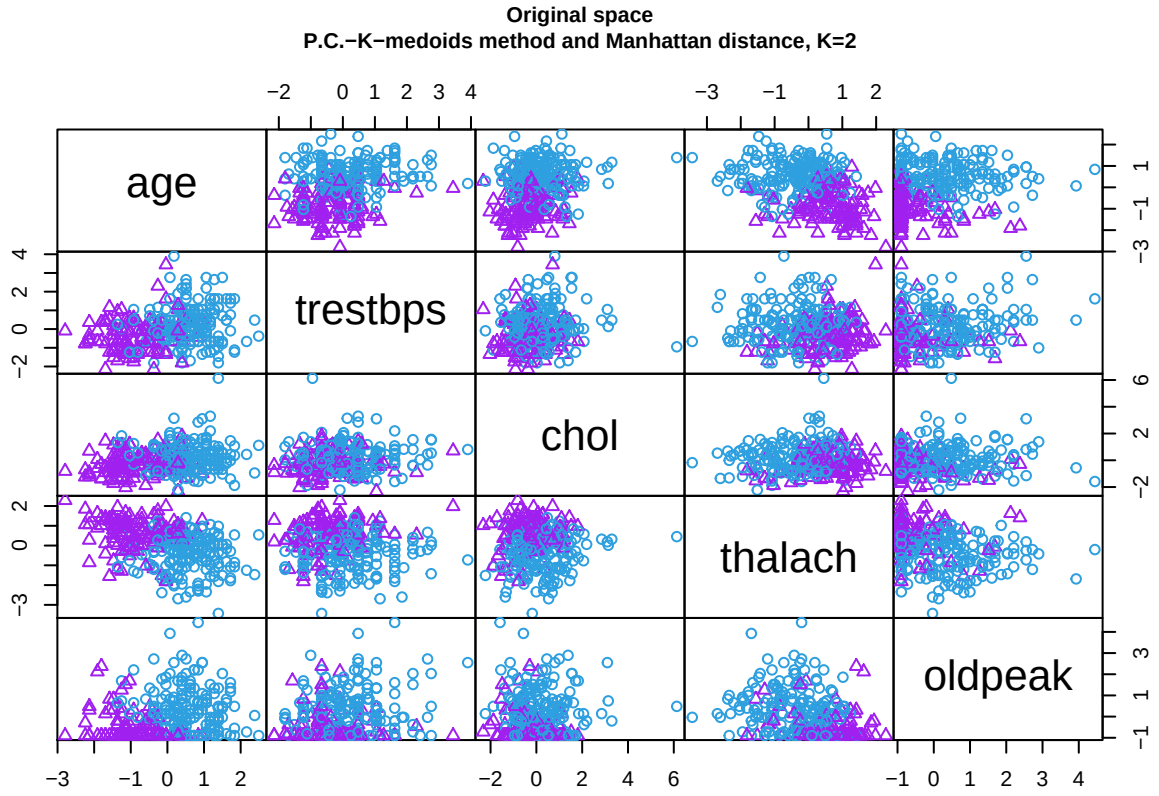
```
## Medoids:
##       ID       age    trestbps       chol    thalach    oldpeak
## [1,] 187  0.620304 -0.09258463  0.1299609 -0.2465324  0.3103986
## [2,] 149 -1.141403 -0.66277043 -0.3909653  0.8449247 -0.8953805
## Clustering vector:
##   [1] 1 2 2 2 1 1 1 2 2 1 1 1 2 1 1 2 2 1 2 1 1 2 2 1 2 1 1 2 1 1 2 1 2 1 1 2 2
##  [38] 1 1 1 1 2 2 1 2 2 2 2 2 2 1 1 1 2 1 1 2 2 2 1 1 2 2 2 2 2 1 2 2 2 1 2 2 2
##  [75] 2 1 1 2 2 1 2 2 1 1 2 1 1 2 1 1 2 2 2 1 2 1 1 2 1 2 2 1 2 2 2 1 1 2 2 2 1
## [112] 2 1 2 1 2 2 1 2 2 1 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 1 1 1 1 2 2 2 1 1 1 2 1
## [149] 2 2 1 1 1 1 2 1 2 2 1 2 2 1 2 2 2 1 1 1 1 1 1 2 1 1 1 2 1 1 2 1 1 1 1 1 1
## [186] 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1
## [223] 1 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 2 1 1 1
## [260] 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1
## [297] 1 1 1 1 1 1 2
## Objective function:
##    build     swap
## 3.534240 3.329987
##
## Available components:
```

```
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```
pam.res$clusinfo
```

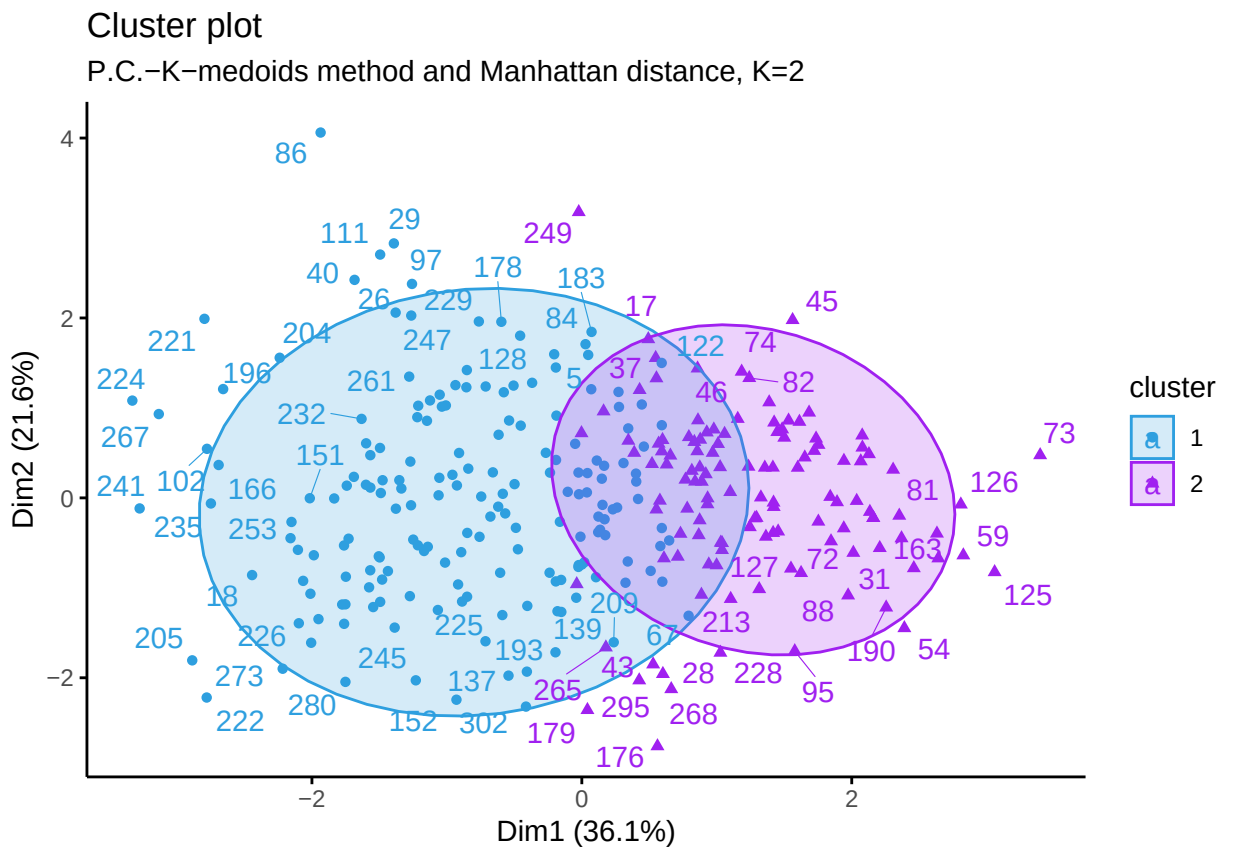
```
##      size max_diss  av_diss diameter separation
## [1,]  181 8.497111 3.677681 15.45052  0.8046137
## [2,]  122 7.441253 2.814145 13.71393  0.8046137
```

```
cm <- pam.res$cluster
pairs(heart_scale, gap=0, main="Original space\nP.C.-K-medoids method and Manhattan distance, K=2",cex.m
```




```
fviz_cluster(pam.res, palette = c("#2E9FDF", "purple"), ellipse.type = "t",
repel = TRUE, ggtheme = theme_classic())+labs(
subtitle = "P.C.-K-medoids method and Manhattan distance, K=2", cex.sub= 0.5)
```

```
## Warning: ggrepel: 232 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Applying partitioning clustering method and using PAM algorithm and Manhattan distance, two clusters are composed in this way: cluster 1 with 181 units, cluster 2 with 122 units.

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

Internal validation measures: silhouette width and Dunn index

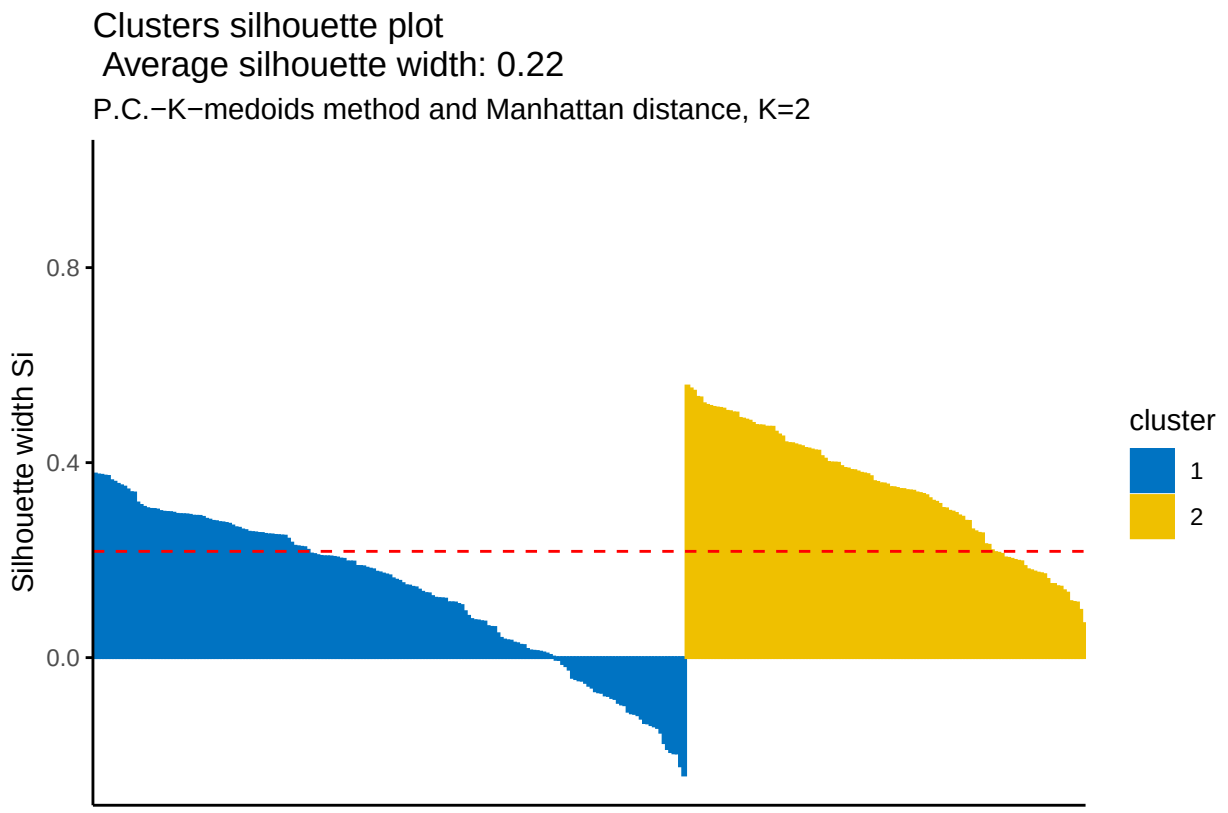
Silhouette width

```
hclust<- eclust(heart_sub,k=2 ,"pam", graph = FALSE, hc_metric = "manhattan")
silinfo <- hclust$silinfo
silinfo$avg.width
```

```
## [1] 0.3500083
```

```
fviz_silhouette(pam.res, palette = "jco",
ggtheme = theme_classic())+labs(
subtitle = "P.C.-K-medoids method and Manhattan distance, K=2", cex.sub= 0.5)
```

```
##   cluster size ave.sil.width
## 1       1  181          0.13
## 2       2  122          0.35
```




```
silinfo$clus.avg.widths
```

```
## [1] 0.4406961 0.2656743
```

```
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
## 252       2        1 -0.01477539
## 200       2        1 -0.01942204
## 146       2        1 -0.02535151
## 81        2        1 -0.03403557
## 2         2        1 -0.03758498
## 273       2        1 -0.05250158
## 19        2        1 -0.05508691
## 185       2        1 -0.05786452
## 24        2        1 -0.06510157
## 259       2        1 -0.07467265
## 100       2        1 -0.08172070
## 299       2        1 -0.08811995
## 77        2        1 -0.09815228
## 42        2        1 -0.11081251
## 101       2        1 -0.11484255
## 109       2        1 -0.11706453
## 120       2        1 -0.11801070
## 231       2        1 -0.12543356
```

```
## 225       2        1 -0.14009810
## 147       2        1 -0.14224913
## 20        2        1 -0.17156592
## 80        2        1 -0.17310467
## 207       2        1 -0.17398031
## 161       2        1 -0.19855278
```

The value of average silhouette width indicates that in average the units are not well enough clustered. In particular, in cluster 1 (blue) the units are in average a higher silhouette value with respect to the silhouette width while in cluster 2 (yellow) the unit have in average a lower value then the total average silhouette. According to this index, 24 units are not well clustered: units that belong to cluster 2, should belong to cluster 1.

Dunn index

```
stats <- cluster.stats(dist(heart_scale), hclust$cluster)
stats$dunn
```

```
## [1] 0.03190583
```

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

External validation measures: confusion matrix, correct Rand index, Meila's

IV index ### Confusion matrix

```
table(heart$sex, hclust$cluster)
```

```
##
##       1   2
##   0  38  58
##   1 108  99
```

According to the Confusion matrix, there is not a perfect agreement between the nominal variable Sex and the cluster solution. A large number of "female" sex (n = 58) has been classified in cluster 2 but Some of them (n = 38) have been classified in cluster 1. For "male" sex a large number of data (n=108)classified in cluster 1 and (n=99) in cluster 2.

Correct Rand Index

```
sex <- as.numeric(heart$sex)
stats<- cluster.stats(d = dist(heart_scale), sex, hclust$cluster)
stats$corrected.rand
```

```
## [1] 0.006139011
```

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] 1.30312
```

Soft clustering approach

Model based-clustering

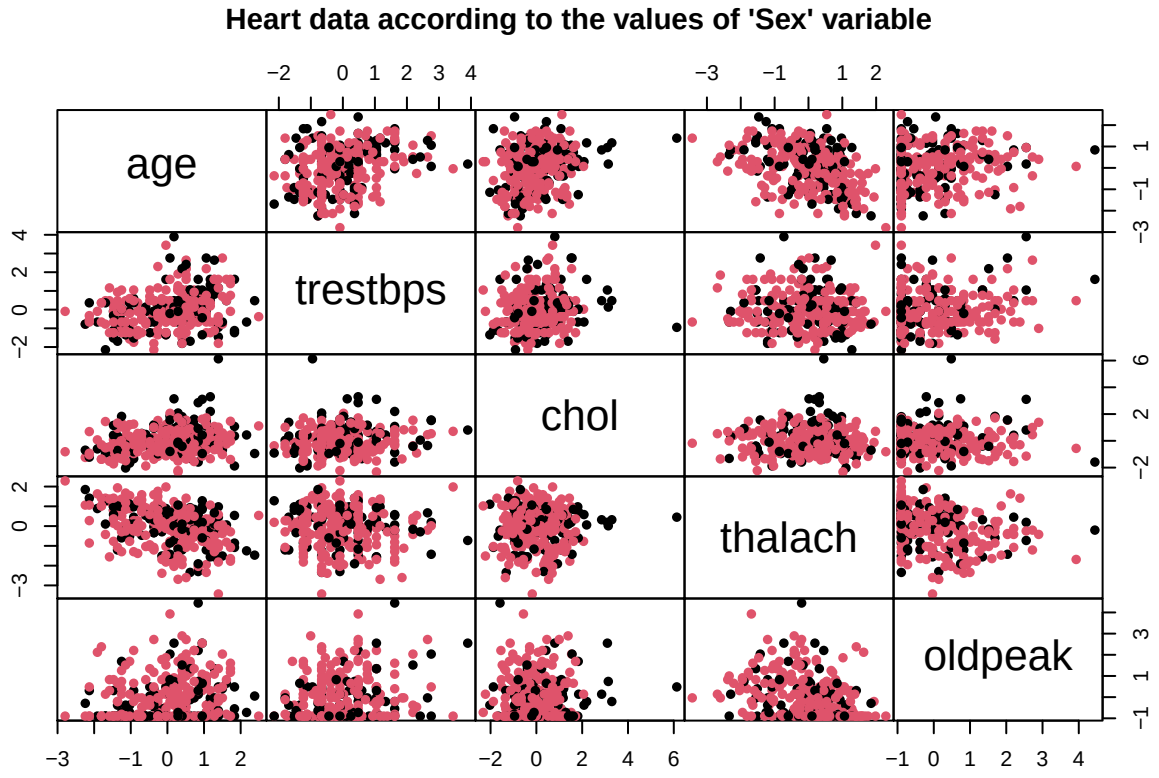
```
summary(heart$sex)
```

```
##   0   1
##  96 207
```

```
head(heart)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  63   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2  37   1  2      130  250   0       1     187     0     3.5     0  0    2
## 3  41   0  1      130  204   0       0     172     0     1.4     2  0    2
## 4  56   1  1      120  236   0       1     178     0     0.8     2  0    2
## 5  57   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6  57   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

```
X <- data.matrix(heart_sub)
sX <- scale(X)
pairs(sX, gap=0, pch = 16, col = as.numeric(heart$sex), cex.main = 0.9 ,main="Heart data according to t
```



To evaluate if there is a relation between the categorical variable Sex 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 sex: "0", "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)
```

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

```
##
## Attaching package: 'mclust'
```

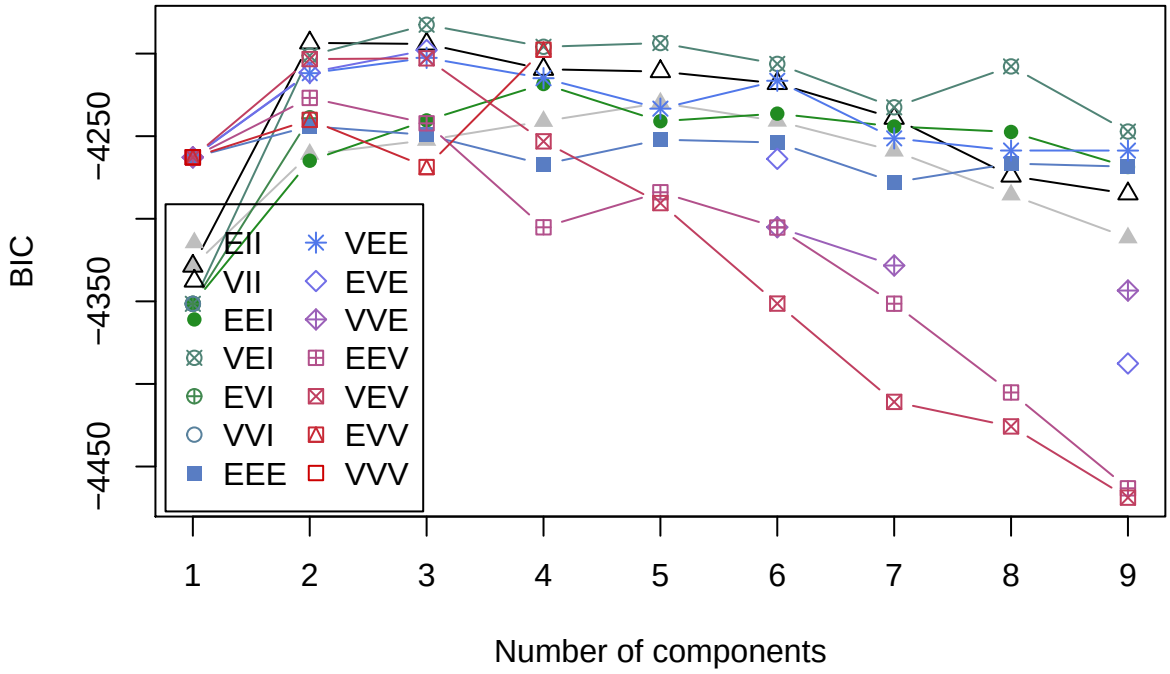
```
## The following object is masked from 'package:psych':
##
##     sim
```

```
## The following object is masked from 'package:gamlss.data':
##
##     acidity
```

```
mod <- Mclust(heart_scale)
summary(mod$BIC)
```

```
## Best BIC values:
##             VEI,3       VII,2       VEI,5
## BIC      -4182.54 -4193.51256 -4193.62766
## BIC diff     0.00   -10.97296   -11.08805
```

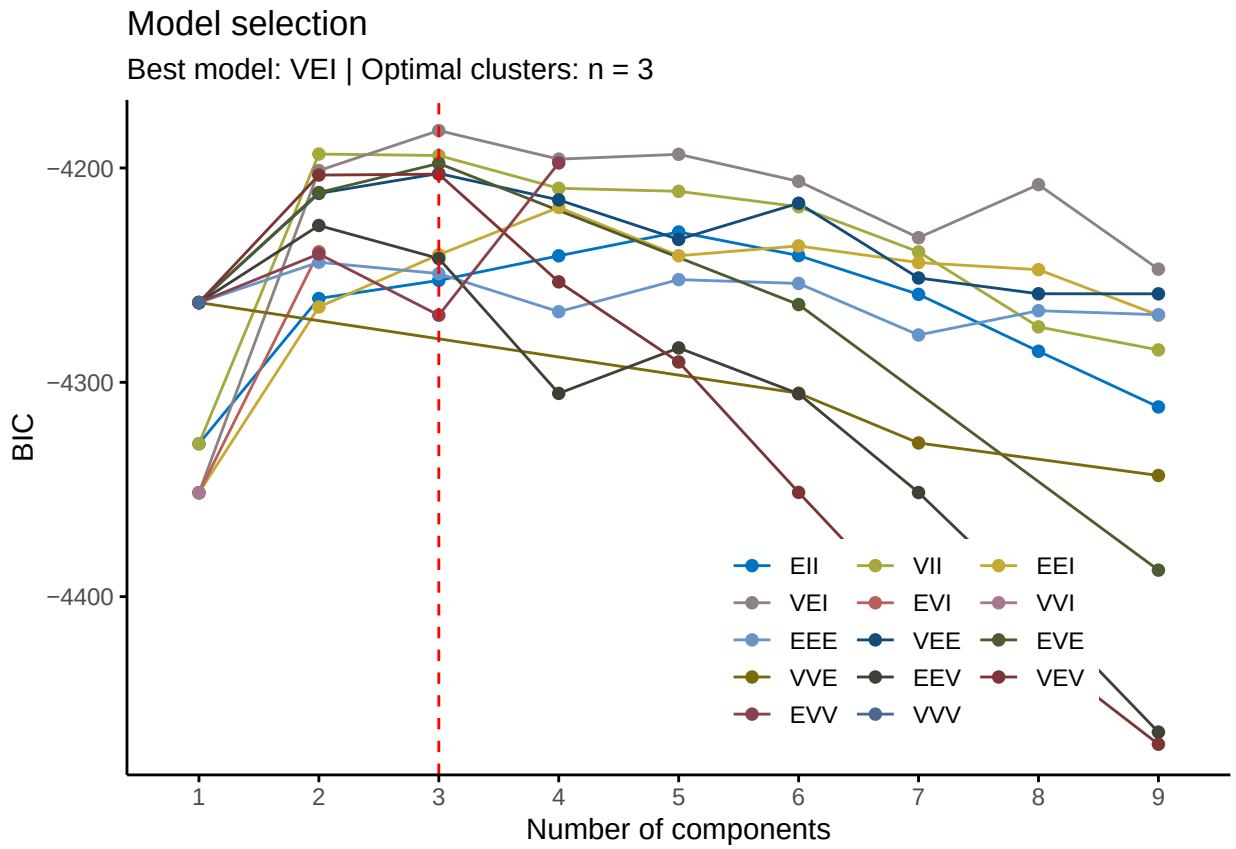
```
plot(mod, what = "BIC", ylim = range(mod$BIC, na.rm = TRUE),
     legendArgs = list(x = "bottomleft"))
```




```
summary(mod)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEI (diagonal, equal shape) model with 3 components:
##
##  log-likelihood   n df      BIC       ICL
##       -2022.705 303 24 -4182.54 -4278.622
##
## Clustering table:
##   1   2   3
##  85 148  70
```

```
library(factoextra)
fviz_mclust(mod, "BIC", palette = "jco")
```




```
head(round(mod$z, 6), 15)
```

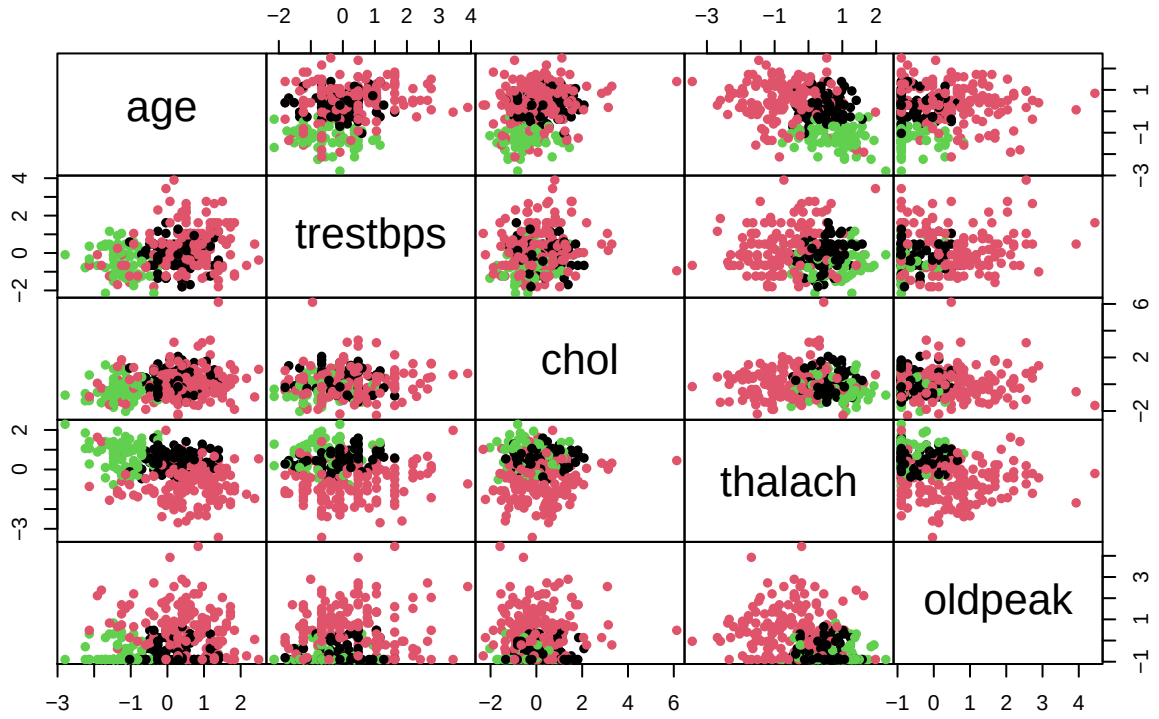
```
##           [,1]     [,2]     [,3]
##  [1,] 0.027447 0.972552 0.000001
##  [2,] 0.000007 0.998244 0.001749
##  [3,] 0.006821 0.024933 0.968246
##  [4,] 0.882783 0.049243 0.067974
##  [5,] 0.669521 0.329064 0.001415
##  [6,] 0.787366 0.206206 0.006428
##  [7,] 0.673329 0.325368 0.001303
##  [8,] 0.028000 0.003351 0.968649
##  [9,] 0.157864 0.834665 0.007471
## [10,] 0.191156 0.805490 0.003354
## [11,] 0.832574 0.145737 0.021688
## [12,] 0.607350 0.239563 0.153087
## [13,] 0.488393 0.026879 0.484728
## [14,] 0.053242 0.946750 0.000008
## [15,] 0.767790 0.231455 0.000755
```

```
head(mod$classification, 15)
```

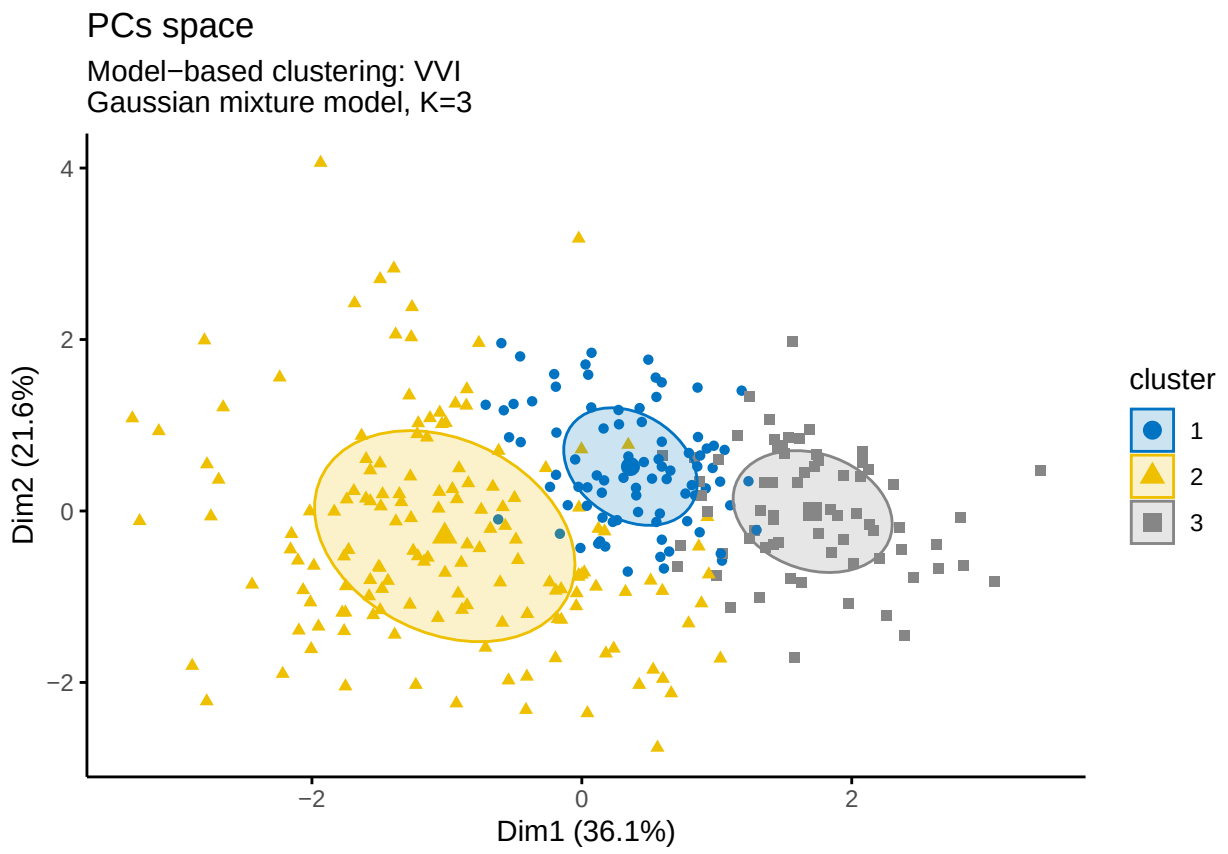
```
##  [1] 2 2 3 1 1 1 1 3 2 2 1 1 1 2 1
```

According to the penalized selection criterion called "BIC" (Bayesian Information Criterion), the three best Gaussian mixture models are: VEI with 3 clusters, VII with 2 clusters and VEI with 5 clusters.The number of clusters that maximizes the BIC of this model is 3: cluster 1 with 85 units, cluster 2 with 148 units and cluster 3 with 70 units.

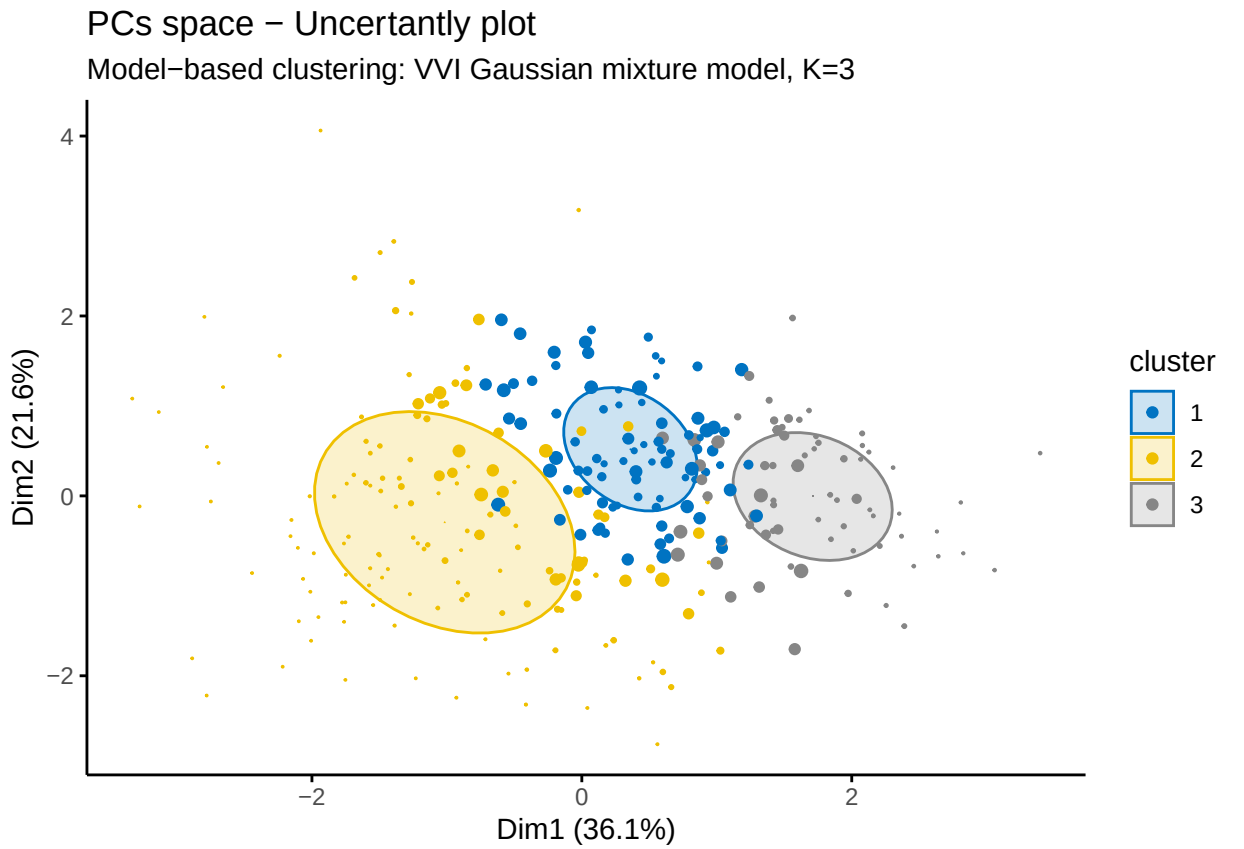
```
library(factoextra)
pairs(sX, gap=0, pch = 16, col = mod$classification, cex.main = 1,main="Original space\nModel-based clus
```




```
fviz_mclust(mod, "classification", geom = "point", pointsize = 1.5, palette = "jco", main = "PCs space"
Gaussian mixture model, K=3")
```




```
fviz_mclust(mod, "uncertainty", palette = "jco",
main = "PCs space - Uncertantly plot")+
labs(subtitle= "Model-based clustering: VVI Gaussian mixture model, K=3")
```



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 and correct Rand index

Confusion matrix

```
table(heart$sex, mod$classification)
```

```
##
##       1   2   3
##   0  31  45  20
##   1  54 103  50
```

According to the Confusion matrix, there is a good agreement between the nominal variable Sex and the cluster solution. A large number of "female" sex (n = 45) has been classified in cluster 2.A large number of "male" sex (n=109) also have been classified in the same cluster (cluster 2).

Correct Rand index

```
adjustedRandIndex(heart$sex, mod$classification)
```

```
## [1] 0.00127975
```

According the Correct Rand Index, there is a good agreement between the Sex 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: internal measures (Connectivity, Silhouette coefficient, Dunn index). The methods considered are: hierarchical method (Euclidean- Manhattan), k-means method, pam algorithm (Euclidean-Manhattan) and model-based clustering

```
library(clValid)
clmethods <- c ("hierarchical", "kmeans", "pam")
V_eucl<-clValid(heart_scale, nClust=2:6, clMethods= clmethods, metric="euclidean", validation="internal"
```

```
## Warning in clValid(heart_scale, nClust = 2:6, clMethods = clmethods, metric =
## "euclidean", : rownames for data not specified, using 1:nrow(data)
```

```
summary(V_eucl)
```

```
##
## Clustering Methods:
##  hierarchical kmeans pam
##
## Cluster sizes:
##  2 3 4 5 6
##
## Validation Measures:
##                                   2        3        4        5        6
##
## hierarchical Connectivity    2.9290  13.8306  16.7595  19.7996  30.3246
##              Dunn            0.4246   0.2168   0.2168   0.2164   0.2048
##              Silhouette      0.5508   0.3682   0.2901   0.1924   0.1630
## kmeans       Connectivity   72.4024 115.7575 147.9873 162.7841 173.9298
##              Dunn            0.0614   0.0638   0.0378   0.0605   0.0732
##              Silhouette      0.2418   0.2161   0.1857   0.2060   0.1933
## pam          Connectivity   82.6706 123.6369 199.9099 208.0766 218.1790
##              Dunn            0.0437   0.0402   0.0430   0.0439   0.0464
##              Silhouette      0.1925   0.1743   0.1567   0.1310   0.1409
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 2.9290 hierarchical 2
## Dunn         0.4246 hierarchical 2
## Silhouette   0.5508 hierarchical 2
```

According to the result, the best clustering algorithm using the Euclidean distance is the hierarchical clustering method with 2 clusters.

```
library(clValid)
clmethods <- c ("hierarchical", "kmeans", "pam")
V_eucl<-clValid(heart_scale, nClust=2:6, clMethods= clmethods, metric="manhattan", validation="internal"
```

```
## Warning in clValid(heart_scale, nClust = 2:6, clMethods = clmethods, metric =
## "manhattan", : rownames for data not specified, using 1:nrow(data)
```

```
summary(V_eucl)
```

```
##
## Clustering Methods:
##  hierarchical kmeans pam
```

```
##
## Cluster sizes:
##  2 3 4 5 6
##
## Validation Measures:
##                                  2        3        4        5        6
##
## hierarchical Connectivity   11.0266  13.9556  21.5806  28.5675  28.7341
##              Dunn            0.1766   0.1766   0.1738   0.1713   0.1713
##              Silhouette      0.3575   0.2412   0.2185   0.1381   0.1259
## kmeans       Connectivity   81.1587 144.6786 165.1794 176.0262 215.8944
##              Dunn            0.0631   0.0478   0.0363   0.0628   0.0625
##              Silhouette      0.2538   0.1919   0.1798   0.2034   0.1848
## pam          Connectivity  102.8933 175.3897 220.4484 217.0905 243.3405
##              Dunn            0.0521   0.0438   0.0463   0.0267   0.0278
##              Silhouette      0.2180   0.1567   0.1447   0.1552   0.1473
##
## Optimal Scores:
##
##              Score   Method       Clusters
## Connectivity 11.0266 hierarchical 2
## Dunn          0.1766 hierarchical 2
## Silhouette    0.3575 hierarchical 2
```

According to the result, the best clustering algorithm using the Manhattan distance is the hierarchical clustering method with 2 clusters.

```
library(clValid)
clmethods <- c ("hierarchical", "kmeans", "pam")
V_eucl<-clValid(heart_scale, nClust=2:6, clMethods= clmethods, metric="manhattan", validation="stability
```

```
## Warning in clValid(heart_scale, nClust = 2:6, clMethods = clmethods, metric =
## "manhattan", : rownames for data not specified, using 1:nrow(data)
```

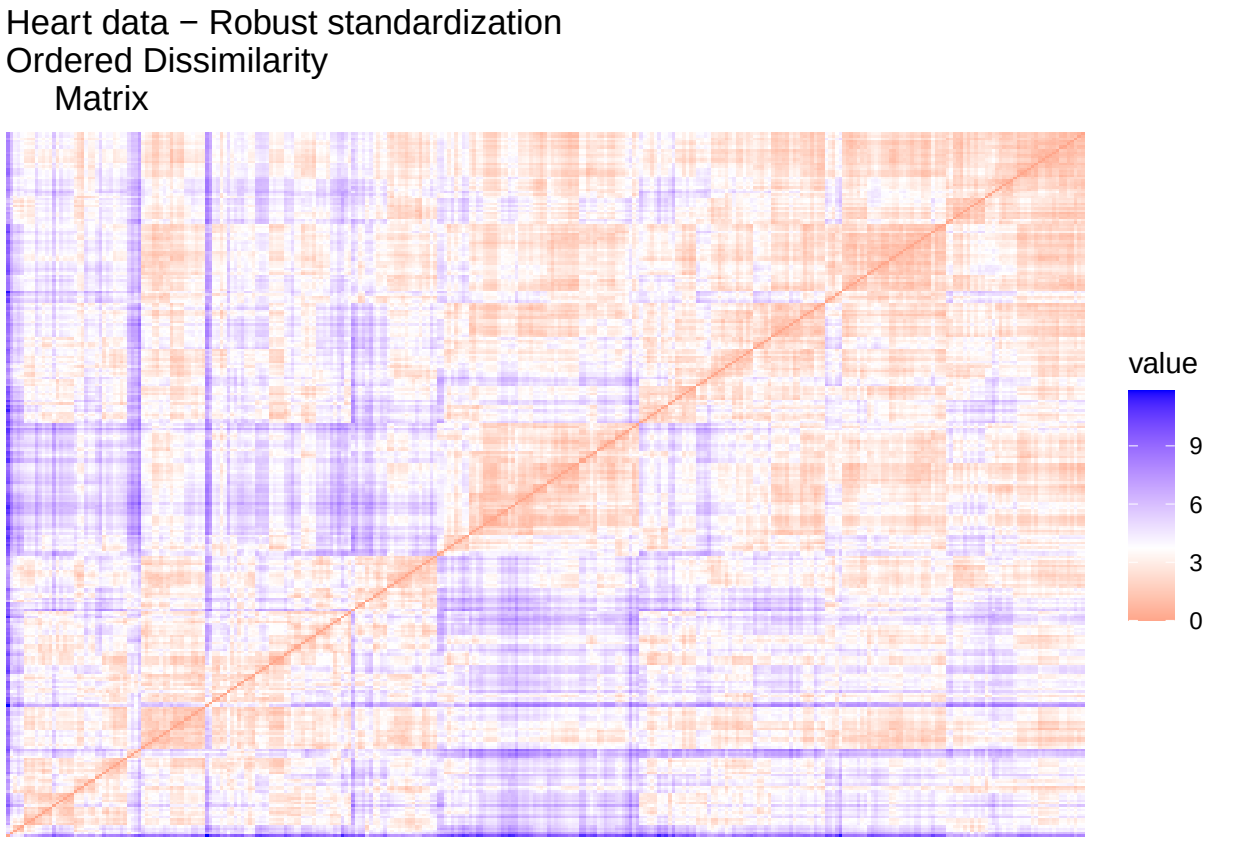
```
summary(V_eucl)
```

```
##
## Clustering Methods:
##  hierarchical kmeans pam
##
## Cluster sizes:
##  2 3 4 5 6
##
## Validation Measures:
##                         2      3      4      5      6
##
## hierarchical APN  0.0061 0.0148 0.0268 0.0932 0.2639
##              AD   5.4542 5.4031 5.3310 5.2804 5.2386
##              ADM  0.1367 0.1299 0.1868 0.3157 0.6338
##              FOM  1.0010 1.0019 1.0019 0.9985 0.9981
## kmeans       APN  0.2000 0.2649 0.3665 0.3966 0.4124
##              AD   4.9085 4.6522 4.6256 4.4135 4.3267
##              ADM  0.5290 0.6674 0.9697 0.9416 0.9883
##              FOM  0.9599 0.9569 0.9542 0.9541 0.9468
## pam          APN  0.2114 0.3668 0.4250 0.5063 0.4470
##              AD   4.9460 4.7885 4.6181 4.5579 4.3432
```

```
##              ADM  0.5233 0.8302 0.8534 1.0767 0.9408
##              FOM  0.9666 0.9615 0.9430 0.9539 0.9427
##
## Optimal Scores:
##
##     Score  Method       Clusters
## APN 0.0061 hierarchical 2
## AD  4.3267 kmeans       6
## ADM 0.1299 hierarchical 3
## FOM 0.9427 pam          6
```

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

```
library(factoextra)
scale_rob <- scale(heart_sub, center = apply(heart_sub, 2, median), scale = apply(heart_sub, 2, meanabs
rownames(scale_rob) <- rownames(heart)
fviz_dist(dist(scale_rob), show_labels = FALSE)+
labs(title = "Heart data - Robust standardization\nOrdered Dissimilarity
     Matrix", gradient = list(low = "red", mid = "white", high =
"blue"))
```




```
hopkins(scale_rob, n = nrow(scale_rob)-1)
```

```
## $H
## [1] 0.2523782
```

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.