post2-clustering

Rongyun Tang

December 10, 2018

Breast Cancer

research outlines:

1. Objective:task is to determine what kind of a person are probabilty to have bread cancer based on blood analysis data.
2. Method: k-means clustering and hierachical clustering
3. Data Source:this "breast cancer" dataset was downloaded from UCI Machine Learning website: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra#. Clinical features were observed or measured for 64 patients with breast cancer and 52 healthy controls. There are 10 predictors, all quantitative, and a binary dependent variable, indicating the presence or absence of breast cancer. The predictors are anthropometric data and parameters which can be gathered in routine blood analysis. Prediction models based on these predictors, if accurate, can potentially be used as a biomarker of breast cancer.

==========

Attribute Information:

==========

- Listing of attributes:
- Quantitative Attributes:
- Age (years)
- BMI (kg/m2)
- Glucose (mg/dL)
- Insulin (µU/mL)
- HOMA
- Leptin (ng/mL)
- Adiponectin (µg/mL)
- Resistin (ng/mL)
- MCP-1(pg/dL)
- Labels:
- 1=Healthy controls
- 2=Patients

1. Data preprocessing

```
rawdata <- read.csv('dataR2.csv')
head(rawdata)
```

```
##   Age      BMI Glucose Insulin      HOMA  Leptin Adiponectin Resistin
## 1  48 23.50000      70   2.707 0.4674087  8.8071    9.702400  7.99585
## 2  83 20.69049      92   3.115 0.7068973  8.8438    5.429285  4.06405
## 3  82 23.12467      91   4.498 1.0096511 17.9393   22.432040  9.27715
## 4  68 21.36752      77   3.226 0.6127249  9.8827    7.169560 12.76600
## 5  86 21.11111      92   3.549 0.8053864  6.6994    4.819240 10.57635
## 6  49 22.85446      92   3.226 0.7320869  6.8317   13.679750 10.31760
##     MCP.1 Classification
## 1 417.114              1
## 2 468.786              1
## 3 554.697              1
## 4 928.220              1
## 5 773.920              1
## 6 530.410              1
```

```
data<-scale(rawdata[,1:9])
head(data)
```

```
##             Age        BMI    Glucose    Insulin       HOMA     Leptin
## [1,] -0.5772891 -0.8131475 -1.2338692 -0.7255915 -0.6116289 -0.9283067
## [2,]  1.5949016 -1.3727948 -0.2571837 -0.6850661 -0.5458722 -0.9263936
## [3,]  1.5328390 -0.8879123 -0.3015785 -0.5476970 -0.4627448 -0.4522571
## [4,]  0.6639627 -1.2379325 -0.9231056 -0.6740408 -0.5717293 -0.8722371
## [5,]  1.7810894 -1.2890089 -0.2571837 -0.6419582 -0.5188300 -1.0381783
## [6,] -0.5152265 -0.9417381 -0.2571837 -0.6740408 -0.5389559 -1.0312817
##      Adiponectin   Resistin       MCP.1
## [1,] -0.06991818 -0.5431610 -0.33977652
## [2,] -0.69433755 -0.8604811 -0.19039777
## [3,]  1.79023159 -0.4397524  0.05796261
## [4,] -0.44003562 -0.1581811  1.13778142
## [5,] -0.78348187 -0.3348991  0.69171506
## [6,]  0.51128181 -0.3557818 -0.01224876
```

```
# identify count of NAs in data frame
sum(is.na(data))
```

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

```
print("Numbers of missing values : 0 ")
```

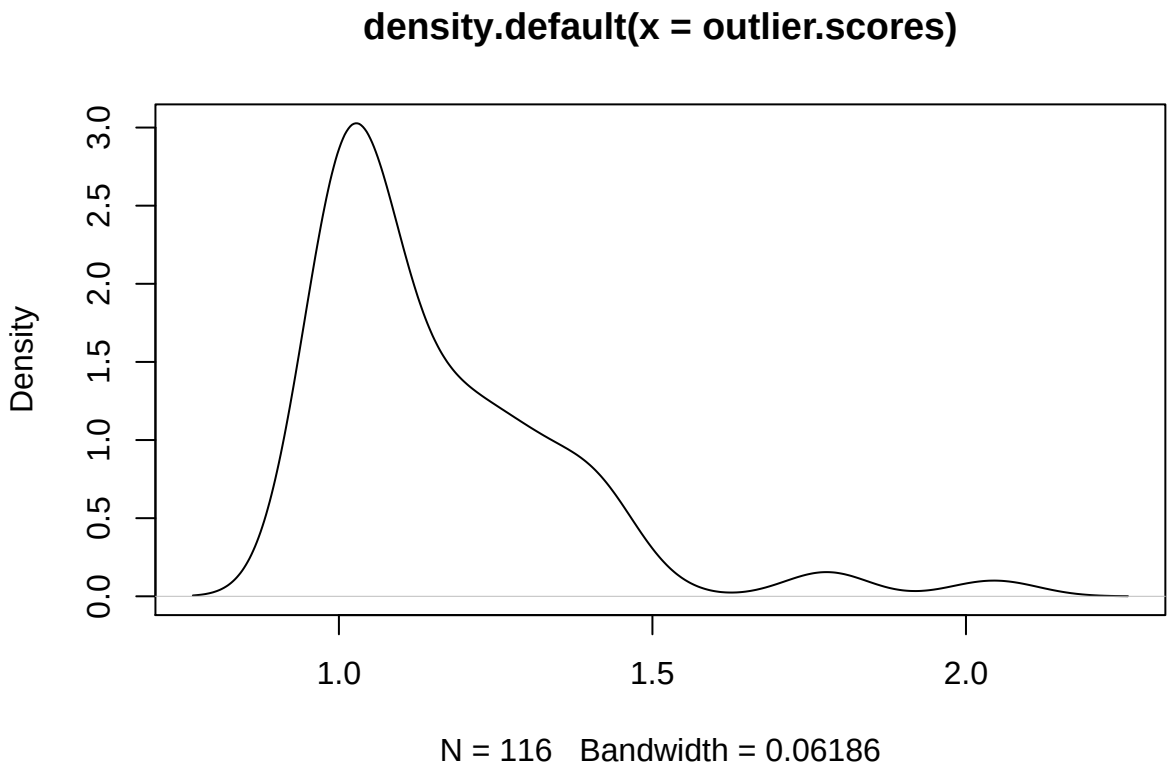
```
## [1] "Numbers of missing values : 0 "
```

```
# outliers detection
library(DMwR)
```

```
## Loading required package: lattice
```

```
## Loading required package: grid
```

```
outlier.scores <- lofactor(data, k=5)
plot(density(outlier.scores))
```



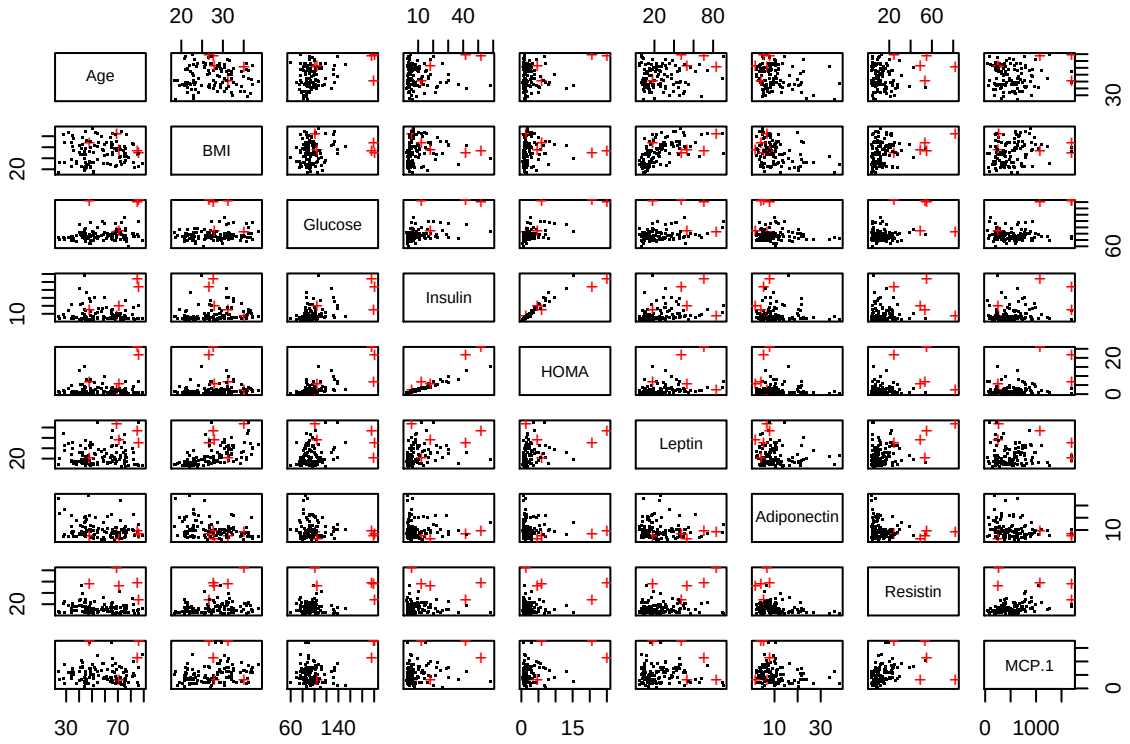

```
outliers <- order(outlier.scores, decreasing=T)[1:5]
print("Top 5 outliers are: ")
```

```
## [1] "Top 5 outliers are: "
```

```
print(outliers) # who are outliers
```

```
## [1] 89 38 79 88 99
```

```
n <- nrow(rawdata[,1:9])

# In case that outliers might be key factors to determine breast cancer, we didn't delete outliers here
pch <- rep(".", n) # show outliers
pch[outliers] <- "+"
col <- rep("black", n)
col[outliers] <- "red"
pairs(rawdata[,1:9], pch=pch, col=col)
```



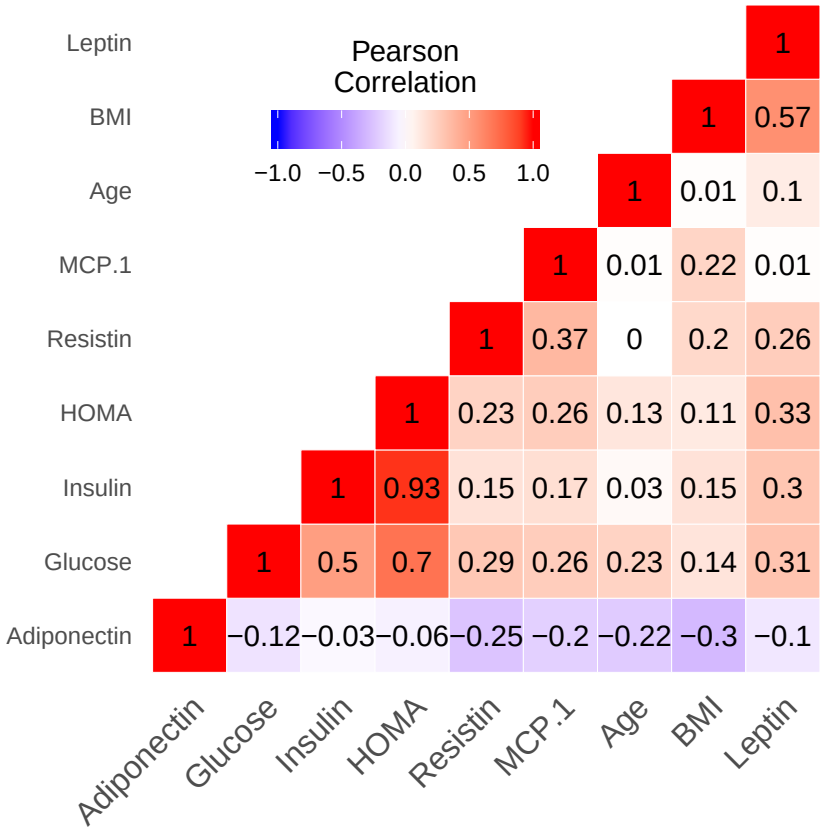
2. Data exploration

```
library(reshape2)
library(ggplot2)
# Compute the correlation matrix
cormat <- round(cor(data),2)
head(cormat)
```

```
##           Age  BMI Glucose Insulin HOMA Leptin Adiponectin Resistin MCP.1
## Age      1.00 0.01    0.23    0.03 0.13   0.10       -0.22     0.00  0.01
## BMI      0.01 1.00    0.14    0.15 0.11   0.57       -0.30     0.20  0.22
## Glucose 0.23 0.14    1.00    0.50 0.70   0.31       -0.12     0.29  0.26
## Insulin 0.03 0.15    0.50    1.00 0.93   0.30       -0.03     0.15  0.17
## HOMA    0.13 0.11    0.70    0.93 1.00   0.33       -0.06     0.23  0.26
## Leptin  0.10 0.57    0.31    0.30 0.33   1.00       -0.10     0.26  0.01
```

```
# Difine functions to reorder the correlation matrix
 get_upper_tri <- function(cormat){
    cormat[lower.tri(cormat)]<- NA
    return(cormat)
  }
reorder_cormat <- function(cormat){
dd <- as.dist((1-cormat)/2) # Use correlation between variables as distance
```

```r
hc <- hclust(dd)
cormat <-cormat[hc$order, hc$order]
}
# Reorder the correlation matrix
cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()
# Print the heatmap
ggheatmap +
geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
```



DATA ANALYSIS:

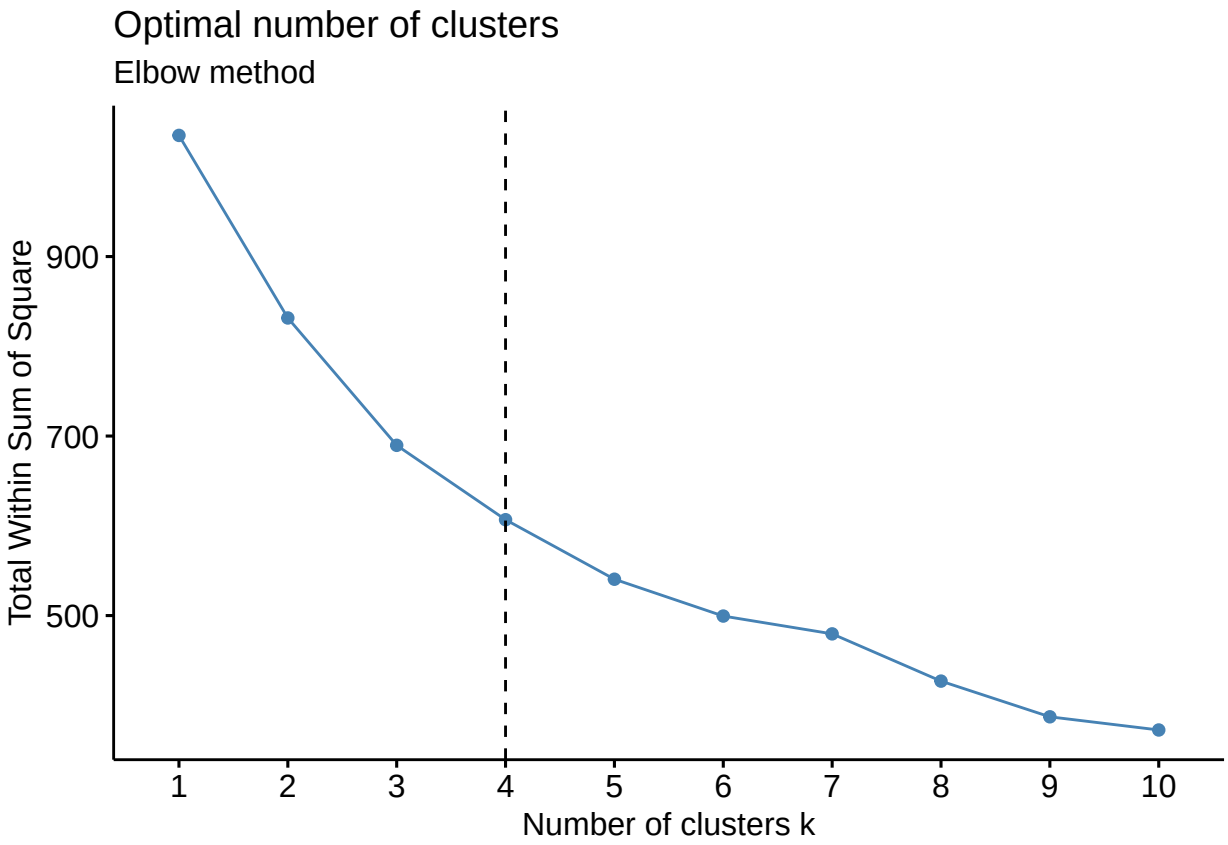
- most of variables have low correlation coefficents(-0.3~ 0.4)
- only HOMA have relatively high correlation coefficents with insulin(0.97) and glusose(0.7).
- adiponectin have low negtive correlationships with all of other variables.

3. K-means Clustering

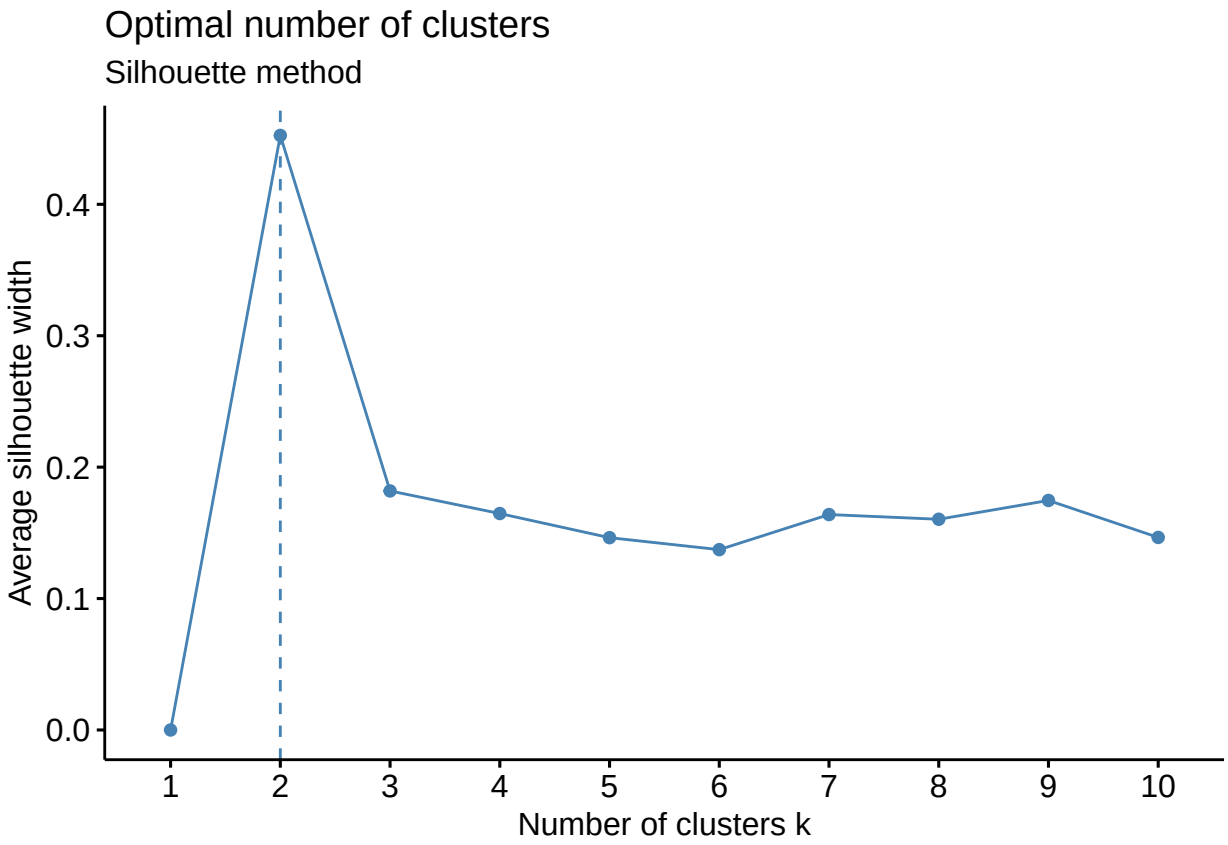
```
# determining number of clusters

#pkgs <- c("factoextra",  "NbClust")
#install.packages(pkgs)
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

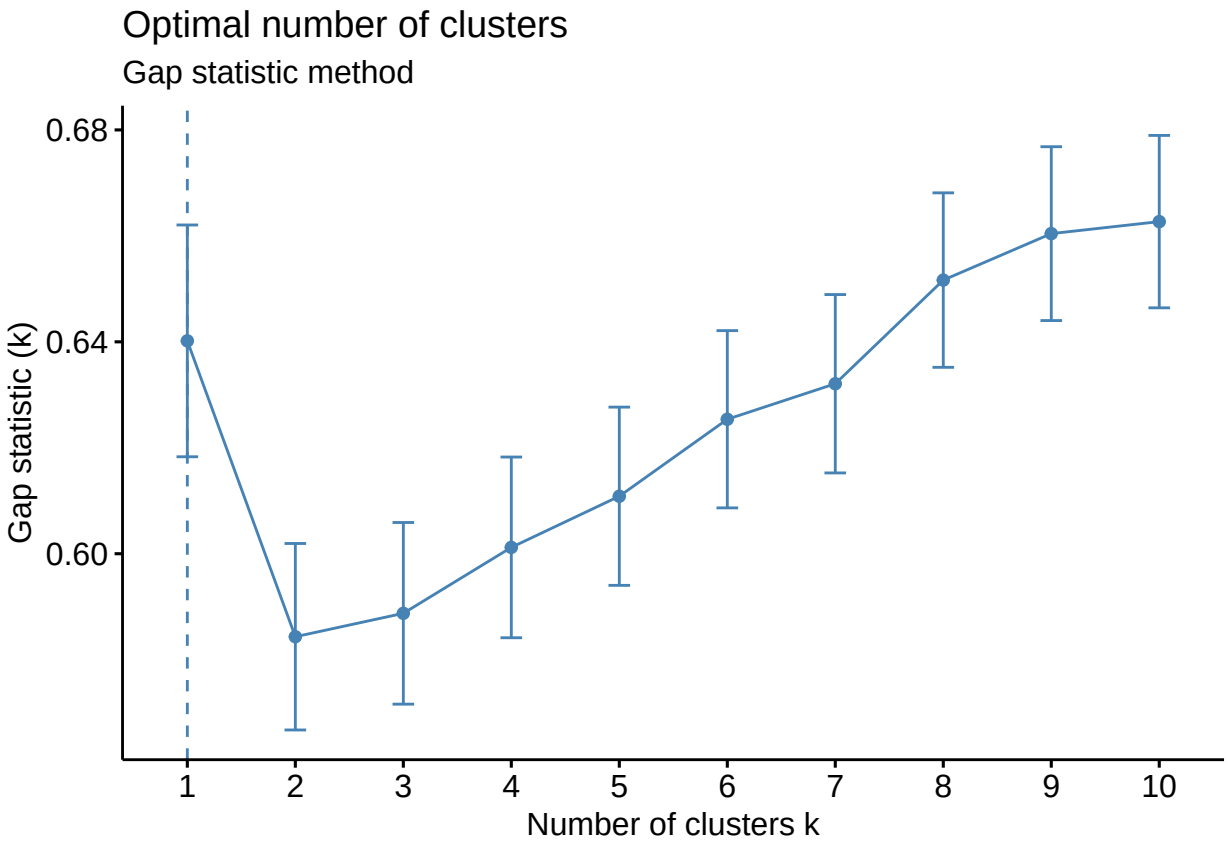
```
library(NbClust)

# 1. Elbow method
fviz_nbclust(data, kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```




```
# 2. Silhouette method
fviz_nbclust(data, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
```




```
# 3. Gap statistic
set.seed(123)
fviz_nbclust(data, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")
```




```
# We considered silhouette method as optimal method, and potentially hoped to seperate normal people an
fit <- kmeans(data, 2)
fit          # print all available components
```

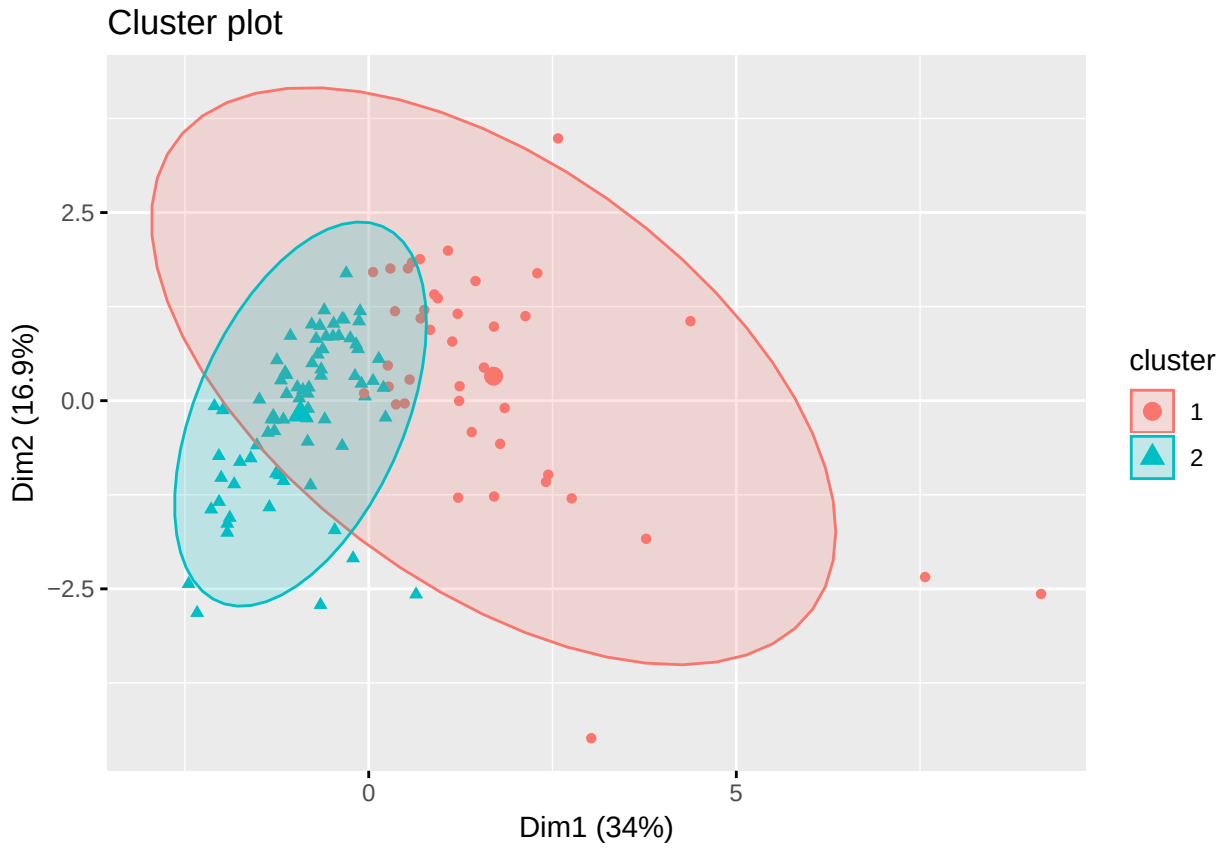
```
## K-means clustering with 2 clusters of sizes 41, 75
##
## Cluster means:
##           Age        BMI    Glucose    Insulin       HOMA     Leptin
## 1  0.05847403  0.8485899  0.6285467  0.7039841  0.6582597  0.8429241
## 2 -0.03196580 -0.4638958 -0.3436055 -0.3848446 -0.3598487 -0.4607985
##   Adiponectin   Resistin      MCP.1
## 1  -0.2114374  0.5041985  0.3884149
## 2   0.1155858 -0.2756285 -0.2123335
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 1 1 1 1 2 1 1 1 1 2
##  [36] 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [71] 2 1 2 2 2 2 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 2 1 2 2 2 1 2 1
## [106] 1 2 2 1 1 1 2 2 1 2 1
##
## Within cluster sum of squares by cluster:
## [1] 476.0327 355.5497
##  (between_SS / total_SS =  19.7 %)
##
## Available components:
##
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
kcenter<-fit$center  # centers of each variable
kcluster<-fit$cluster # cluster ID for each observation
#fit$cluster<-factor("healthy","patient")
fviz_cluster(fit, data = data, geom = "point",
             stand = FALSE, frame.type = "norm")
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```

```
## Warning: argument frame.type is deprecated; please use ellipse.type
## instead.
```



4. Validation for K-means Clustering

```
# exract real classes from raw data
real.class=rawdata[,10] #
real.class
```

```
##   [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
```

```
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [71] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [106] 2 2 2 2 2 2 2 2 2 2
```

```
# relable k-means clusters
relable<-fit$cluster
relable[relable==2] <-0 # exchange lable 1 and lable 2 of k-means clusters
relable[relable==1] <-2
relable[relable==0] <-1
relable
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 2 2 2 2 1 2 2 2 2 1
##  [36] 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 1 2 1 1 1 1 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1 1 2 1 1 1 2 1 2
## [106] 2 1 1 2 2 2 1 1 2 1 2
```

```
# extract dfference between real classes lables and labled k-means clusters
difference=real.class-relable
difference
```

```
##   [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0 -1  0  0 -1
##  [24]  0  0 -1 -1 -1 -1  0 -1 -1 -1 -1  0  0  0 -1  0  0  0  0  0  0 -1  0
##  [47]  0  0  0  0 -1  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [70]  1  1  0  1  1  1  1  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0
##  [93]  0  0  0  1  1  1  0  1  1  1  0  1  0  0  1  1  0  0  0  1  1  0  1
## [116]  0
```

```
# accuracy

accuracy.kmeans=length(difference[difference=='0'])/length(difference)
accuracy.kmeans
```

```
## [1] 0.5603448
```

```
print("k-means accuracy is: 56%")
```

```
## [1] "k-means accuracy is: 56%"
```

5. Hierarchical Clustering

```
library('dendextend')
```

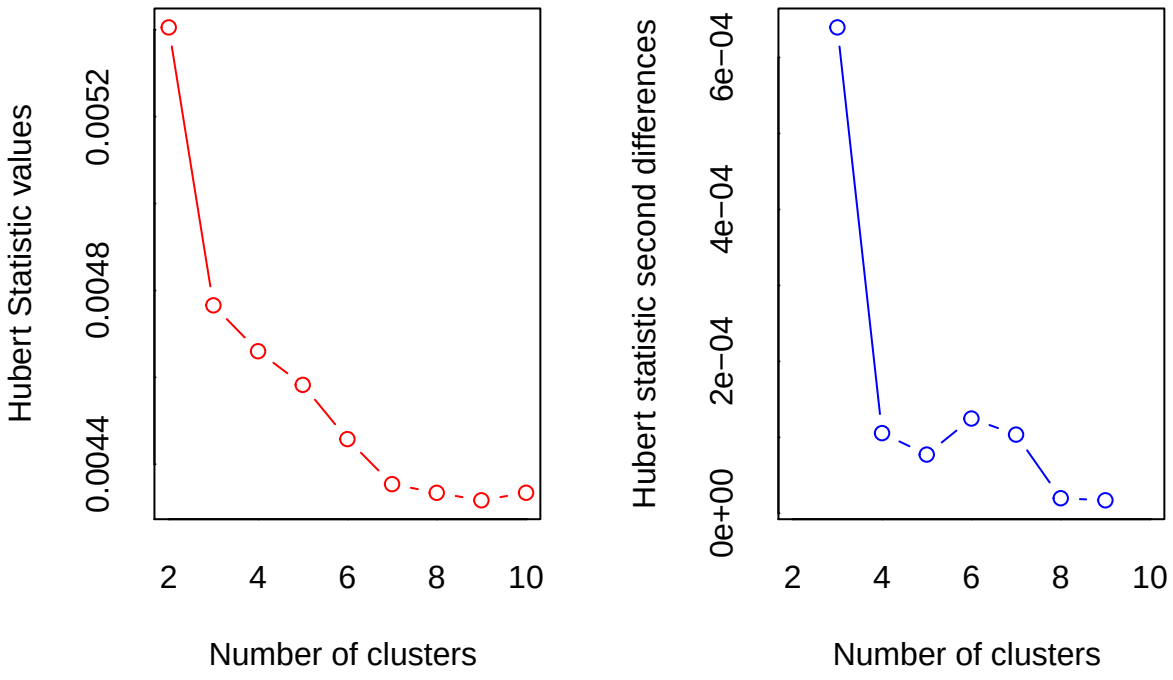
```
##
## ---------------------
## Welcome to dendextend version 1.8.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
```

```
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ----------------------
```

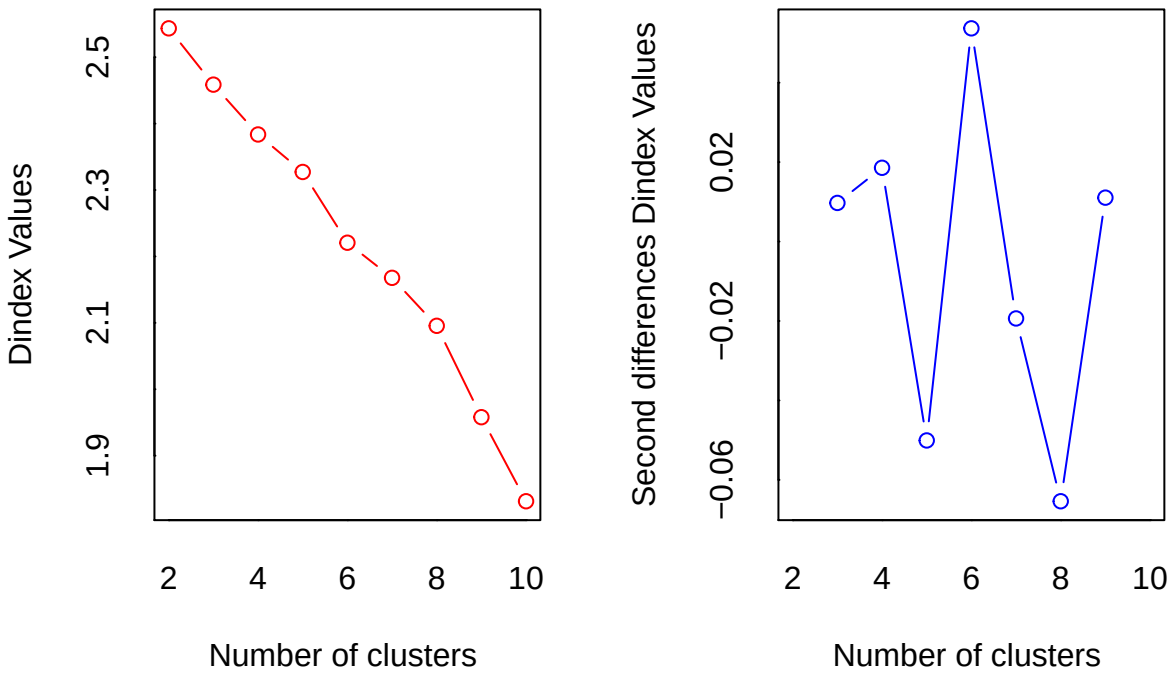
```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library("cluster")

# Best Cluster Number
nb <- NbClust(data, distance = "euclidean", min.nc = 2,
        max.nc = 10, method = "complete", index ="all")
```




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

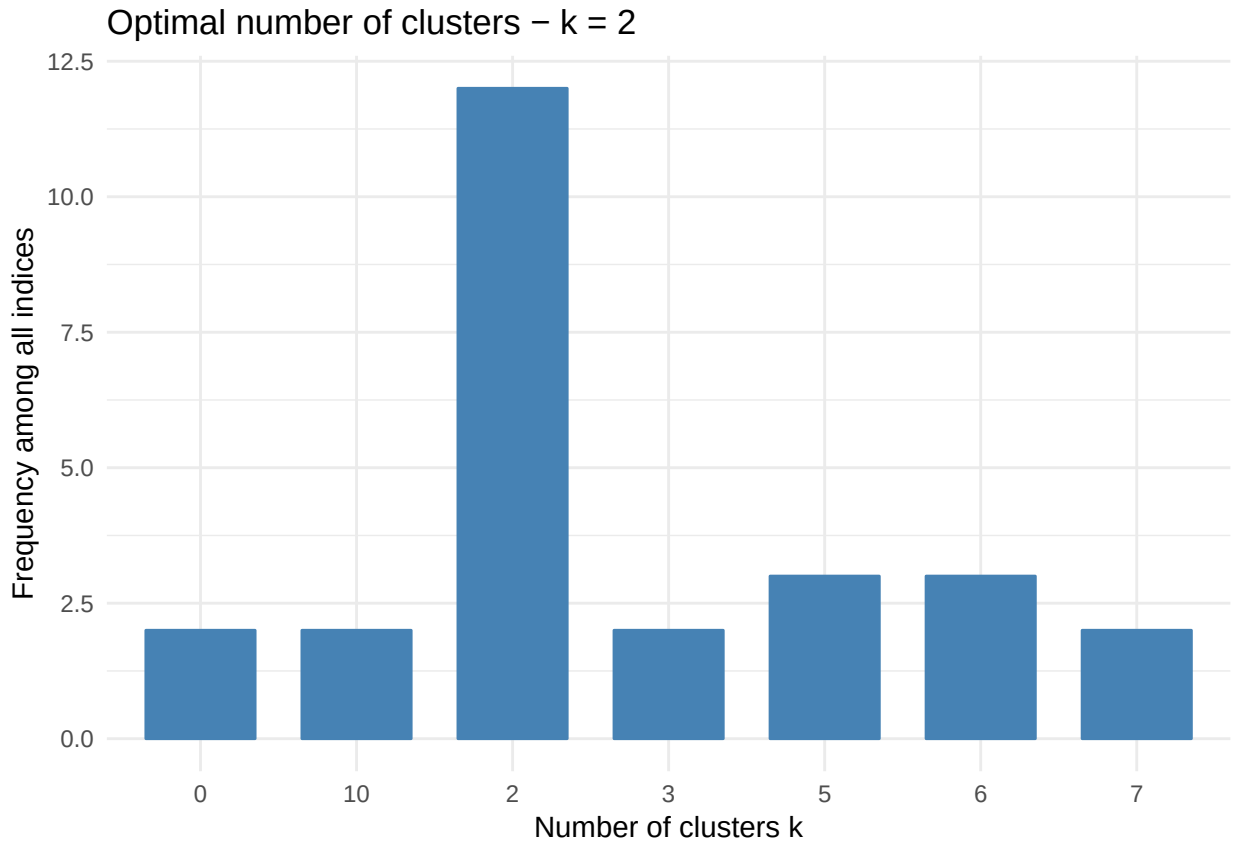



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##       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

```
fviz_nbclust(nb) + theme_minimal()
```

```
## Among all indices:
## ===================
```

```
## * 2 proposed  0 as the best number of clusters
## * 12 proposed  2 as the best number of clusters
## * 2 proposed  3 as the best number of clusters
## * 3 proposed  5 as the best number of clusters
## * 3 proposed  6 as the best number of clusters
## * 2 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 .
```

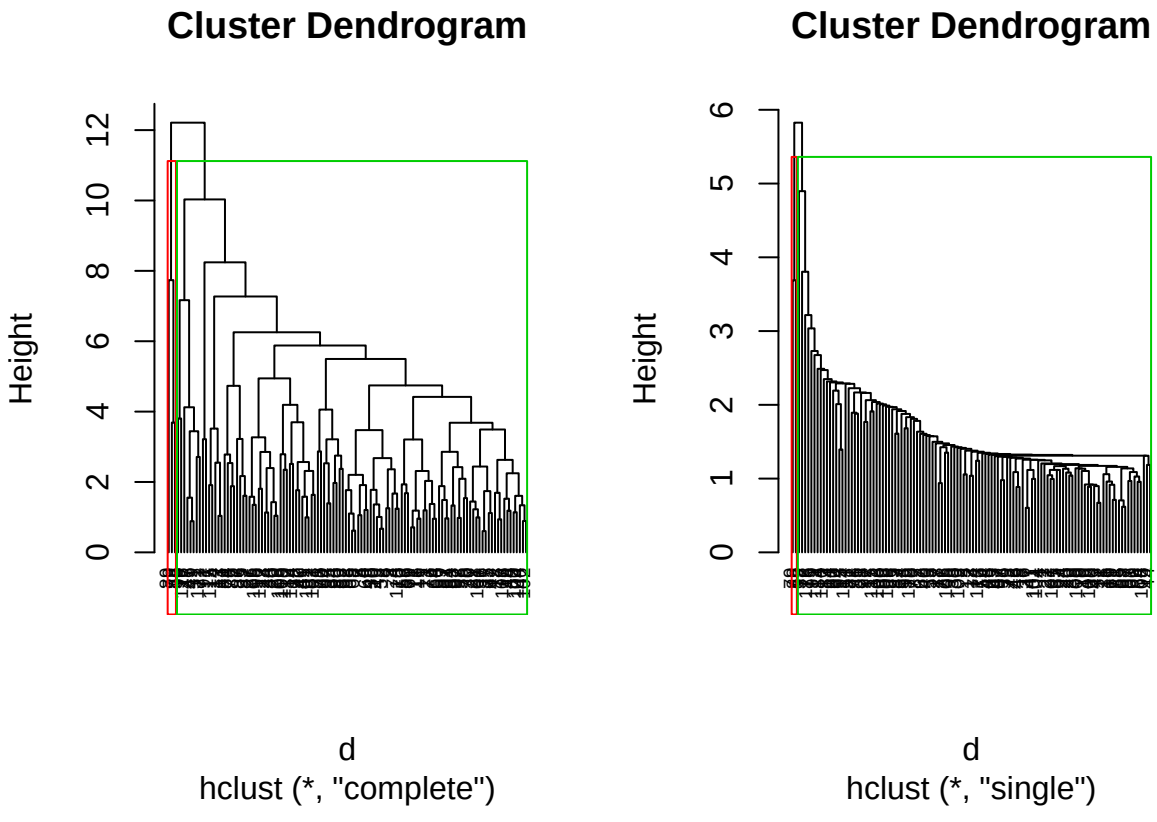



```r
# Dissimilarity matrix
d <- dist(data, method = "euclidean") # distance matrix

# Complete Linkage
hc.cp <- hclust(d, method = "complete" )

# Single Linkage
hc.sg <- hclust(d, method = "single" )

# Average Linkage
hc.av <- hclust(d, method = "average" )

# Centroid Linkage
hc.ct <- hclust(d, method = "centroid" )
```

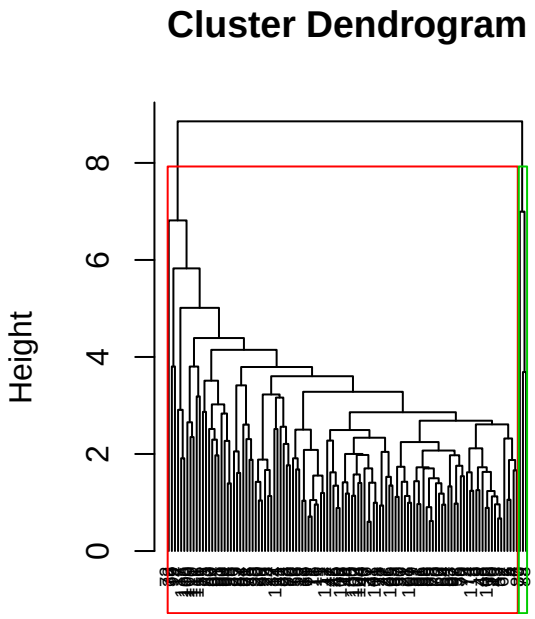
```r
# Ward.D2 Linkage
hc.wd <- hclust(d,method = "ward")
```

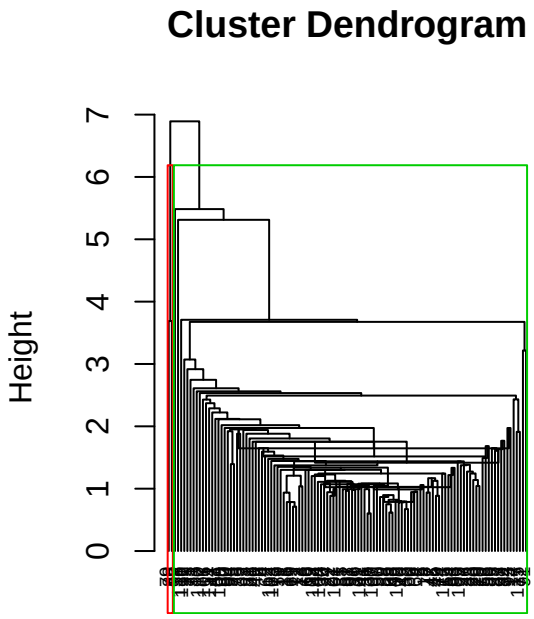
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```r
# Ward.D2 Linkage
hc.wd2 <- hclust(d,method = "ward.D2")

# Mcquitty Linkage
hc.mq <- hclust(d, method = "mcquitty" )

# Plot the obtained dendrogram
plot(hc.cp, cex = 0.6, hang = -1)
rect.hclust(hc.cp, k = 2, border = 2:4)
plot(hc.sg, cex = 0.6, hang = -1)
rect.hclust(hc.sg, k = 2, border = 2:4)
```

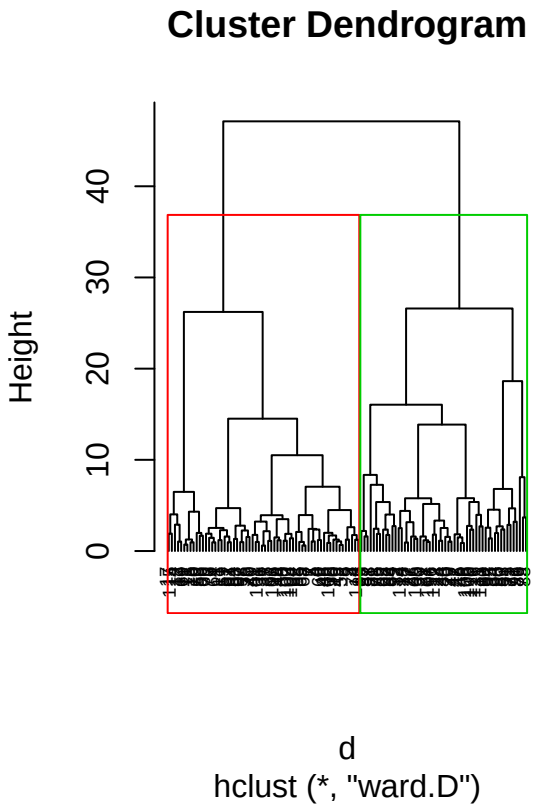


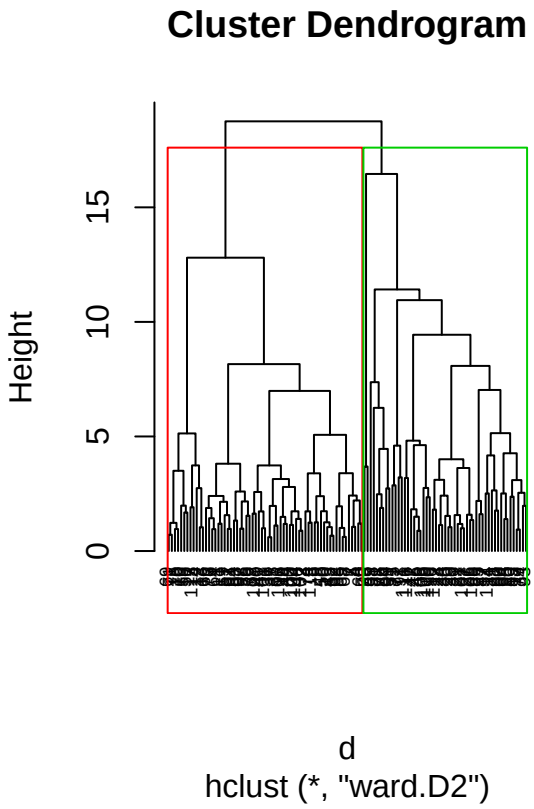

```r
plot(hc.av, cex = 0.6, hang = -1)
rect.hclust(hc.av, k = 2, border = 2:4)
plot(hc.ct, cex = 0.6, hang = -1)
rect.hclust(hc.ct, k = 2, border = 2:4)
```






```
plot(hc.wd, cex = 0.6, hang = -1)
rect.hclust(hc.wd, k = 2, border = 2:4)
plot(hc.wd2, cex = 0.6, hang = -1)
rect.hclust(hc.wd2, k = 2, border = 2:4)
```






```
print("ward linkage and ward.D2 linkage produce best clustering results")
```

```
## [1] "ward linkage and ward.D2 linkage produce best clustering results"
```

6. Validation for Hierarchical Clustering

```
# Cut into 2 groups
hc.cut <- cutree(hc.wd, k = 2)
hc.cut
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
##  [36] 1 1 2 1 1 1 2 1 1 2 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 1 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1 1 1 1 2
## [106] 2 2 1 2 2 2 2 1 2 1 2
```

```
# extract dfference between real classes lables and labled groups
difference.hc=real.class-hc.cut
difference.hc
```

```
##   [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0 -1 -1 -1 -1
##  [24] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  0  0 -1  0  0  0 -1  0  0 -1  0
##  [47] -1  0 -1  0 -1  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
##  [70]  1  1  0  1  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##  [93]  0  0  0  0  1  1  0  1  1  1  1  1  0  0  0  1  0  0  0  0  1  0  1
## [116]  0
```

```
# accuracy

accuracy.hc.ward=length(difference.hc[difference.hc=='0'])/length(difference.hc)
accuracy.hc.ward
```

```
## [1] 0.5344828
```

```
print("hierarchical method accuracy is: 52.6%")
```

```
## [1] "hierarchical method accuracy is: 52.6%"
```

```
sub1=subset(data,hc.cut==1)
summary(sub1)
```

```
##       Age                BMI             Glucose
##  Min.   :-2.06679   Min.   :-1.8350   Min.   :-1.677817
##  1st Qu.:-0.76348   1st Qu.:-1.2328   1st Qu.:-0.656737
##  Median : 0.22952   Median :-0.8863   Median :-0.257184
##  Mean   : 0.06436   Mean   :-0.6040   Mean   :-0.301579
##  3rd Qu.: 0.91221   3rd Qu.:-0.0791   3rd Qu.: 0.009185
##  Max.   : 1.96728   Max.   : 1.5947   Max.   : 0.897081
##     Insulin             HOMA              Leptin         Adiponectin
##  Min.   :-0.7529   Min.   :-0.6116   Min.   :-1.1627   Min.   :-1.1671
##  1st Qu.:-0.6480   1st Qu.:-0.5335   1st Qu.:-0.9259   1st Qu.:-0.6931
##  Median :-0.5341   Median :-0.4481   Median :-0.6620   Median :-0.2534
##  Mean   :-0.4282   Mean   :-0.3801   Mean   :-0.5871   Mean   : 0.1603
##  3rd Qu.:-0.3895   3rd Qu.:-0.3487   3rd Qu.:-0.3532   3rd Qu.: 0.4956
##  Max.   : 1.1608   Max.   : 0.6121   Max.   : 0.9074   Max.   : 4.0710
##     Resistin            MCP.1
##  Min.   :-0.92941   Min.   :-1.36172
##  1st Qu.:-0.64753   1st Qu.:-0.80349
##  Median :-0.38419   Median :-0.44425
##  Mean   :-0.28582   Mean   :-0.32753
##  3rd Qu.: 0.01258   3rd Qu.: 0.05633
##  Max.   : 1.07390   Max.   : 2.08560
```

```
sub2=subset(data,hc.cut==2)
summary(sub2)
```

```
##       Age                BMI             Glucose
##  Min.   :-1.81854   Min.   :-1.0701   Min.   :-1.23387
##  1st Qu.:-0.81002   1st Qu.: 0.2029   1st Qu.:-0.45696
##  Median :-0.20491   Median : 0.6862   Median :-0.03521
##  Mean   :-0.07389   Mean   : 0.6934   Mean   : 0.34626
##  3rd Qu.: 0.72603   3rd Qu.: 1.2280   3rd Qu.: 0.49753
##  Max.   : 1.78109   Max.   : 2.1905   Max.   : 4.58185
##     Insulin             HOMA              Leptin
```

```
##  Min.   :-0.74218   Min.   :-0.58514   Min.   :-0.8578
##  1st Qu.:-0.39801   1st Qu.:-0.35841   1st Qu.:-0.1190
##  Median : 0.02711   Median :-0.01759   Median : 0.6104
##  Mean   : 0.49163   Mean   : 0.43637   Mean   : 0.6741
##  3rd Qu.: 0.81023   3rd Qu.: 0.69516   3rd Qu.: 1.2316
##  Max.   : 4.81218   Max.   : 6.13814   Max.   : 3.3188
##   Adiponectin          Resistin           MCP.1
##  Min.   :-1.24572   Min.   :-0.85032   Min.   :-1.4131
##  1st Qu.:-0.58564   1st Qu.:-0.50770   1st Qu.:-0.6055
##  Median :-0.28827   Median :-0.07239   Median : 0.2750
##  Mean   :-0.18408   Mean   : 0.32817   Mean   : 0.3760
##  3rd Qu.: 0.05631   3rd Qu.: 0.67717   3rd Qu.: 1.0030
##  Max.   : 1.80601   Max.   : 5.43749   Max.   : 3.3644
```

7. Conclusions

In this project, we used k-means and hierachical clustering methods and blood analysis variables to predict normal people and breast cancer patients. Then, known lables (normal or patients) are used to calculate clustering accuracy and evaluate prediction abilities. Research results showed that:

- Both two methods have moderate ability (~ 50%) to predict potential breast cancer patients.
- K-means method had a higher predictiong accuracy(56%) than hierarchical clustering method(52.6%)
- Result of K-mean method showed that breast cancer patients have much higer age, BMI, Glucose,insulin,HOMA,leptin and resistin than healthy controled people.
- Adiponectin and MCP.1 show very few difference between patients and healthy ones.