final_binf702_spr22_Sandoval2

Lorena Sandoval

5/16/2022

binf702 final

Problem 1 - Consider the Pima Indian data contained in the MASS package.Provide a table of of probability of correct classification results from a 1, 3, 5 nearest neighbor classifier. Train on Pima.tr and test on Pima.te.Be careful not to include the type column as this is effectively the class label.

#KNN 1 cluster1

```
#---------------
library(class)
library(tidyverse) #ggplot and dplyr
library(MASS) #Modern Applied Statistics with S

train = Pima.tr[,-8]
test = Pima.te[,-8]

train.type =Pima.tr[,8]
test.type =Pima.te[,8]
set.seed (1)
knn.pred1=knn(train,test,train.type ,k=1)
table(knn.pred1 ,test.type)
```

```
##          test.type
## knn.pred1  No Yes
##       No  174  56
##       Yes  49  53
```

```
##this function divides the correct predictions by total number of predictions that tell us how accurat
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
table(knn.pred1 ,test.type)
```

```
##          test.type
## knn.pred1  No Yes
##       No  174  56
##       Yes  49  53
```

```
#(174+53)/332
#[1] 0.6837349
```

```
accuracy(table(knn.pred1 ,test.type))
```

```
## [1] 68.37349
```

accuracy for 1 clusters is 68%

#KNN 3 clusters

```
knn.pred3=knn(train,test,train.type ,k=3)
table(knn.pred3 ,test.type)
```

```
##          test.type
## knn.pred3  No Yes
##       No  192  45
##       Yes  31  64
```

```
#(192+64)/332
#[1] 0.7710843
```

```
accuracy(table(knn.pred3 ,test.type))
```

```
## [1] 77.10843
```

accuracy for 3 clusters is 77%

#KNN 5 clusters

```
knn.pred5=knn(train,test,train.type ,k=5)
table(knn.pred5 ,test.type)
```

```
##          test.type
## knn.pred5  No Yes
##       No  196  43
##       Yes  27  66
```

```
# (196+66)/332
# [1] 0.7891566
```

```
accuracy(table(knn.pred5 ,test.type))
```

```
## [1] 78.91566
```

accuracy for 5 clusters is 79%

Problem 2 - Return to the Pima Indian data and provide probability of correct clasification results using a support vector machine. Once again train on Pima.tr and test on Pima.te.

#SVM Classification

```
library(e1071)
library(tidyverse) #ggplot and dplyr
library(MASS) #Modern Applied Statistics with S

train = Pima.tr#[,-8]
test = Pima.te[,-8]

set.seed (1)
svmfit =svm(data=train ,type~.,type = "C-classification", kernel ="linear")
svmPredict <- predict(svmfit, test)
sum(svmPredict == Pima.te[,8]) / nrow(Pima.te)
```

```
## [1] 0.7981928
```

probability of correct clasification is 80%

Problem 3 - Repeat your analysis of problem 2 using Random Forests.

```
library (randomForest)
set.seed (1)
rforest =randomForest(x=train[,-8], y=as.factor(train$type),ntree = 1000, importance=TRUE, proximity=TRU
table(rfpret = rforest$pred,training=Pima.tr[,8])
```

```
##       training
## rfpret  No Yes
##    No  110  34
##    Yes  22  34
```

```
pv <- predict(rforest, Pima.te[,-8], type="class", proximity=TRUE)
table(rfpredv=pv$pred,testing=Pima.te[,8])
```

```
##        testing
## rfpredv  No Yes
##     No  192  45
##     Yes  31  64
```

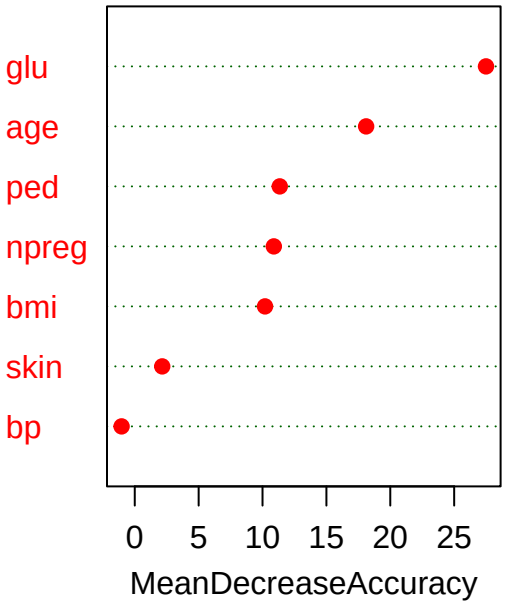
```
# Probability of correct classification on testing data
1- ((47+33)/332)
```

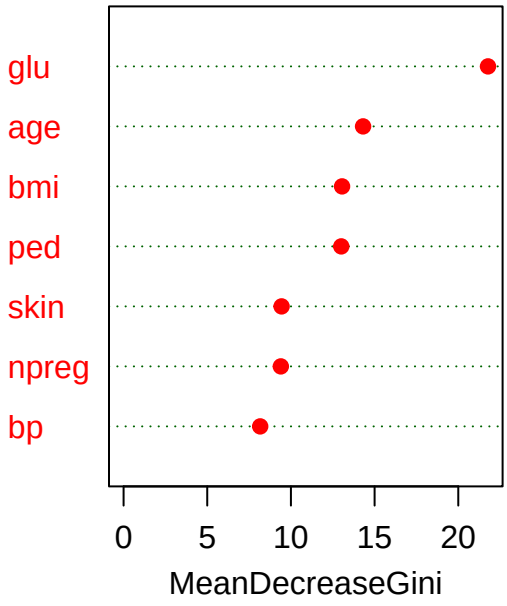
```
## [1] 0.7590361
```

probability of correct clasification is 76%

Problem 4 - Return to your model of problem 3 and perform a variable importance plot based on MeanDecreaseGini.Identify the top 3 variables and discuss their biological relevance.

```
varImpPlot(rforest,
 n.var = 7,
 pch=19,
 main=NULL,
 col="red",
 gcolor="blue",
 lcolor="darkgreen")
```



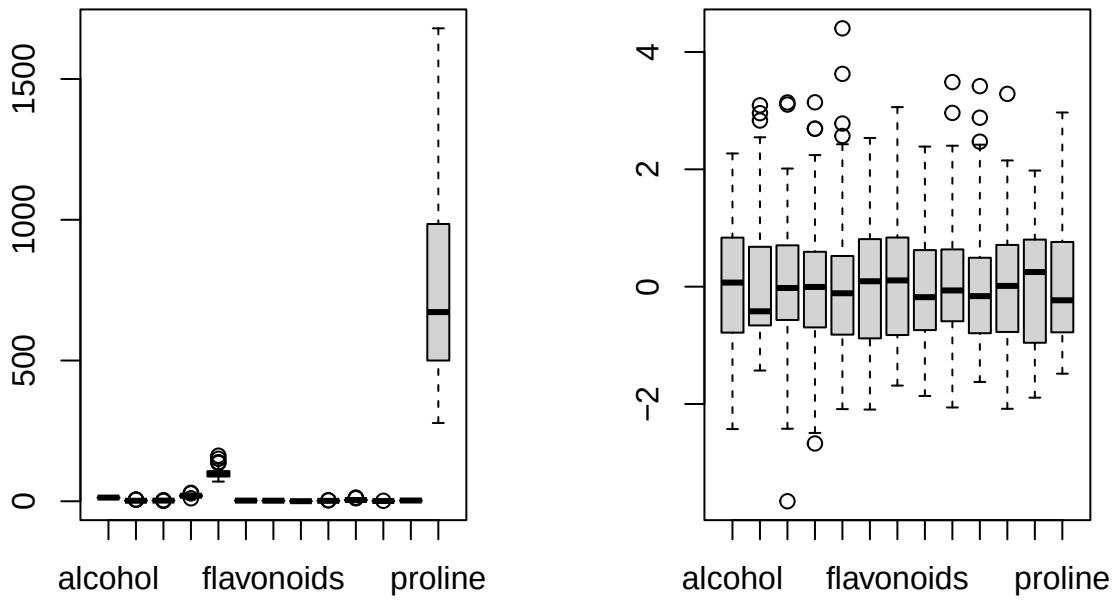


The three most important variables are glucose in tolerance test, age, and diabetes pedigree function.

The glucose tolerance test is a clinical test for diabetes or prediabetes .It is important in predicting diabetes in the dataset. Age and pedigree are also important because family history and increased age can increase the likelihood of developing diabetes.

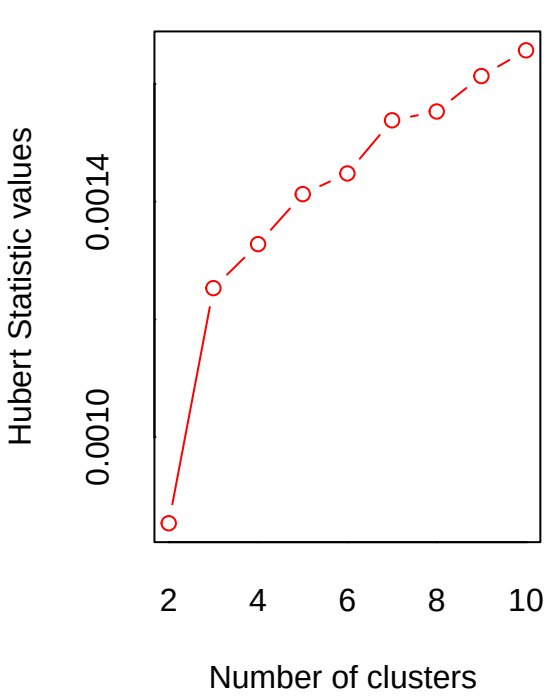
Problem 5 - Consider the wines dataset contained in the kohonen package.Provide side by side boxplots of the original wines data along with the wines data that has been subjected to a column wise standardizing transformation.

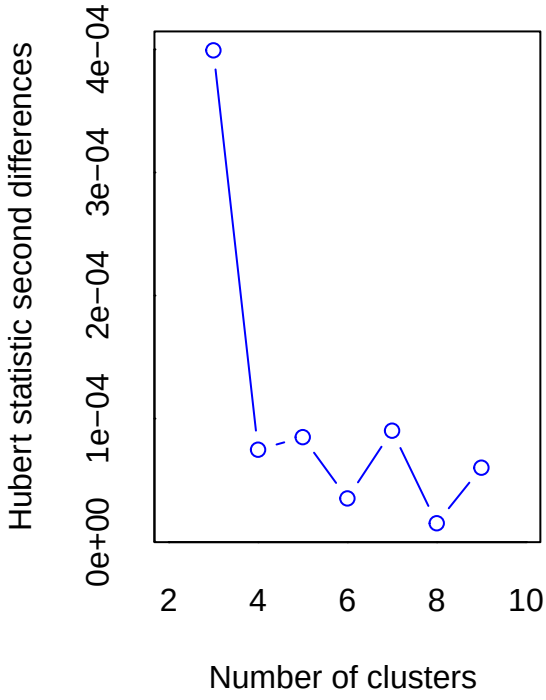
```
# install.packages("kohonen")
library(kohonen)
 data(wines)
dat =wines
dat3 = scale(as.matrix(dat))
par(mfrow=c(1,2))    # set the plotting area into a 1*2 array

boxplot(dat)
boxplot(dat3)
```



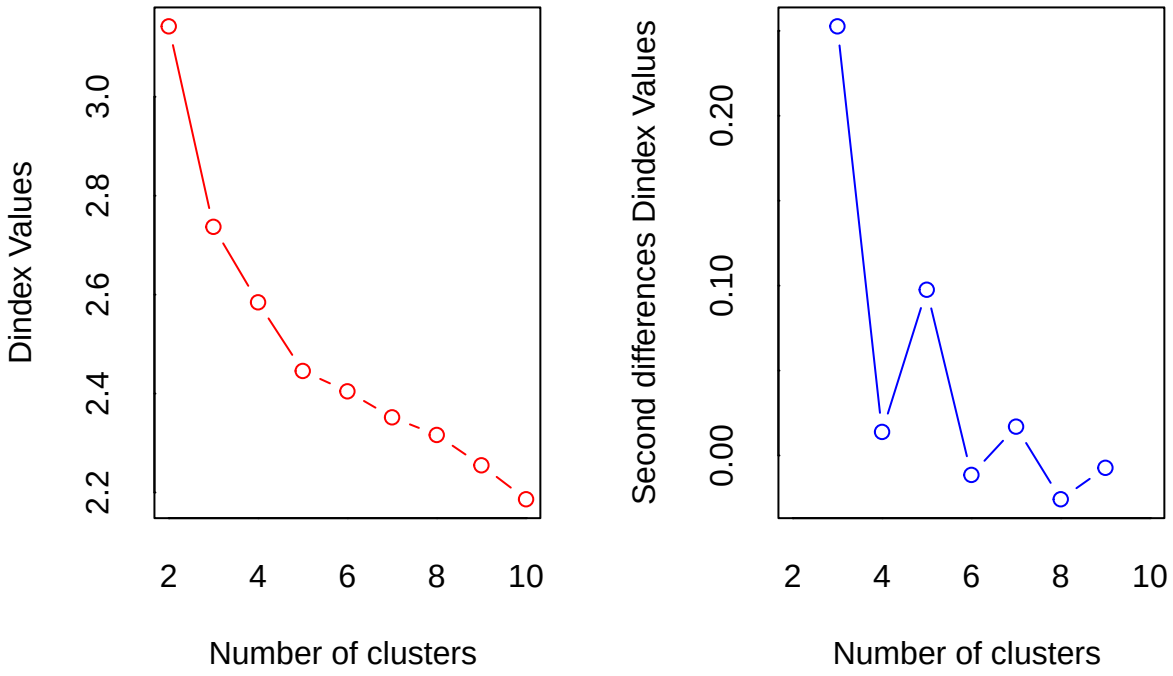
Problem 6 - In this problem we will use a new package to calculate the desired number of clusters in the wine data and then we will use this number along with hclust to create a cluster index which we will compare to the true labels indicating the region the wine grapes were grown in. [Hint - Please install the NbClust package and execute the following command to determine the number of clusters.] no_of_Clusters = NbClust(wines.sc, distance = "euclidean", min.nc = 2, max.nc = 10, method = "complete", index ="all") Compare the obtained class labels to the true ones contained in vintages.

```
# install.packages("NbClust")
library(NbClust)
wines.sc = scale(wines, center = T, scale = T)
no_of_Clusters = NbClust(data = wines.sc, diss=NULL,
                         distance = "euclidean", min.nc = 2, max.nc = 10, method = "complete", index ="a
```





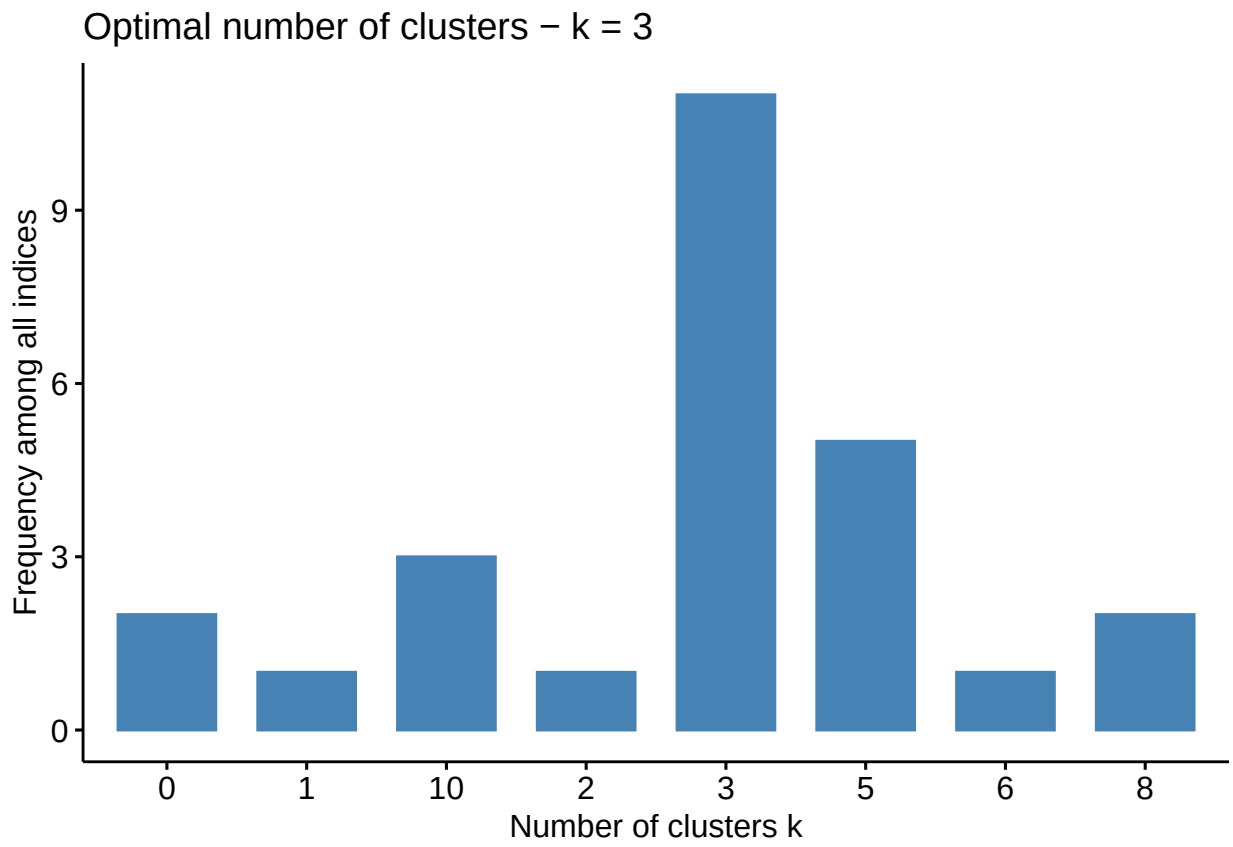

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 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:
## * 1 proposed 2 as the best number of clusters
## * 11 proposed 3 as the best number of clusters
## * 5 proposed 5 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 2 proposed 8 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  3
##
##
## *******************************************************************
```

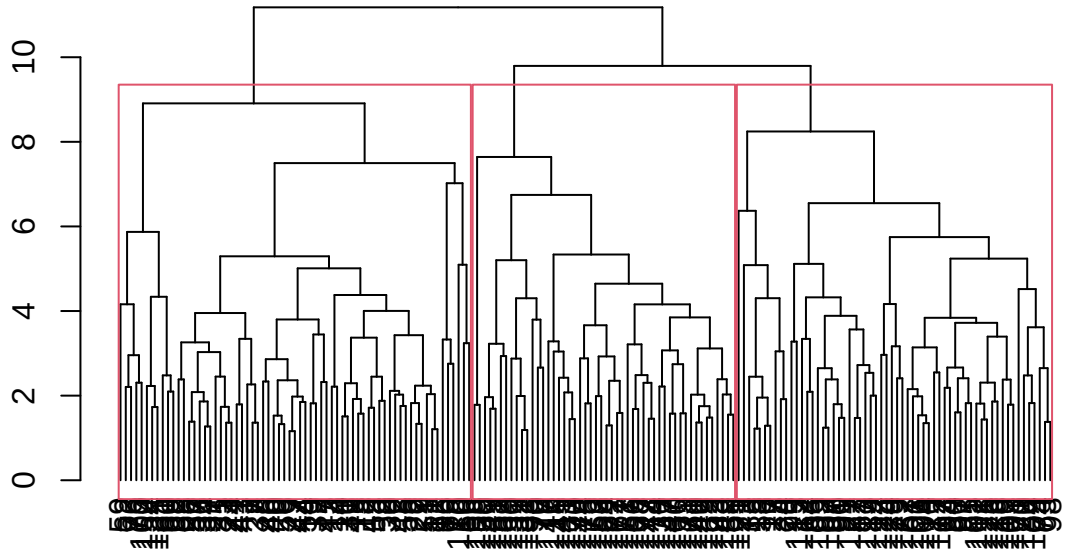
```
#Best number of clusters is 3
labels = vintages
library(factoextra)
fviz_nbclust(no_of_Clusters)
```

```
## Among all indices:
## ===================
## * 2 proposed  0 as the best number of clusters
## * 1 proposed  1 as the best number of clusters
## * 1 proposed  2 as the best number of clusters
## * 11 proposed  3 as the best number of clusters
## * 5 proposed  5 as the best number of clusters
## * 1 proposed  6 as the best number of clusters
## * 2 proposed  8 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  3 .
```

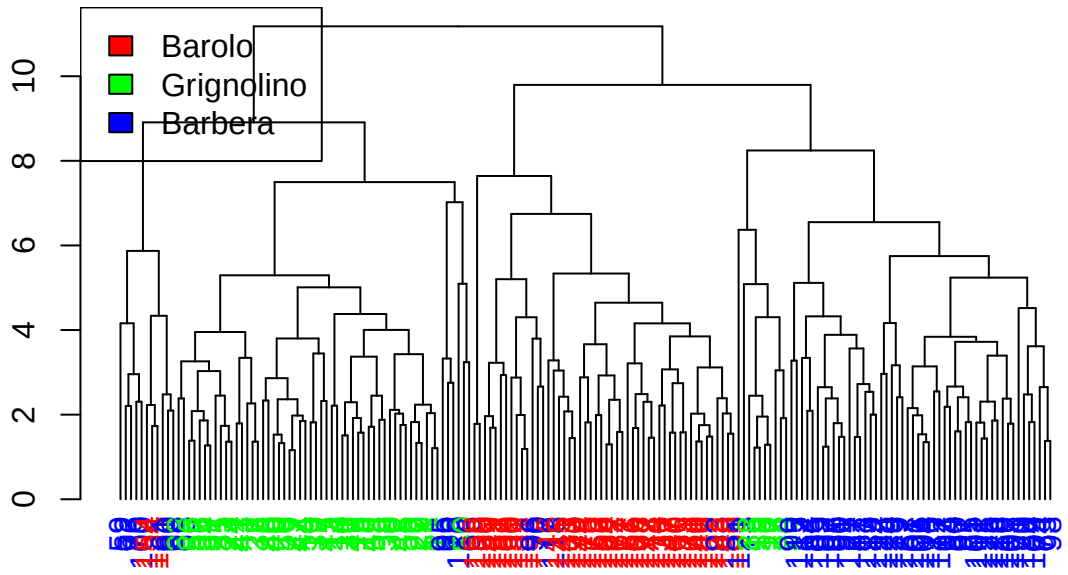


Optimal number of clusters is 3

```
h= hclust(dist(wines.sc))
plot(as.dendrogram(h))
rect.hclust(h, k = 3)
```




```
library(dendextend)
dend <- as.dendrogram(h)
COLS = c("red","green","blue")
names(COLS) = unique(vintages)
dend <- color_labels(dend, col = COLS[vintages[labels(dend)]])
plot(dend)
legend("topleft", legend = c("Barolo", "Grignolino", "Barbera"), fill = COLS)
```




```
clusters =cutree(h, k=3)
class = factor(vintages, levels = c("Barolo", "Grignolino", "Barbera"))
table(class, clusters)
```

```
##             clusters
## class          1  2  3
##   Barolo      50  8  0
##   Grignolino 14 52  5
##   Barbera      3  0 45
```

```
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(table(class,clusters))
```

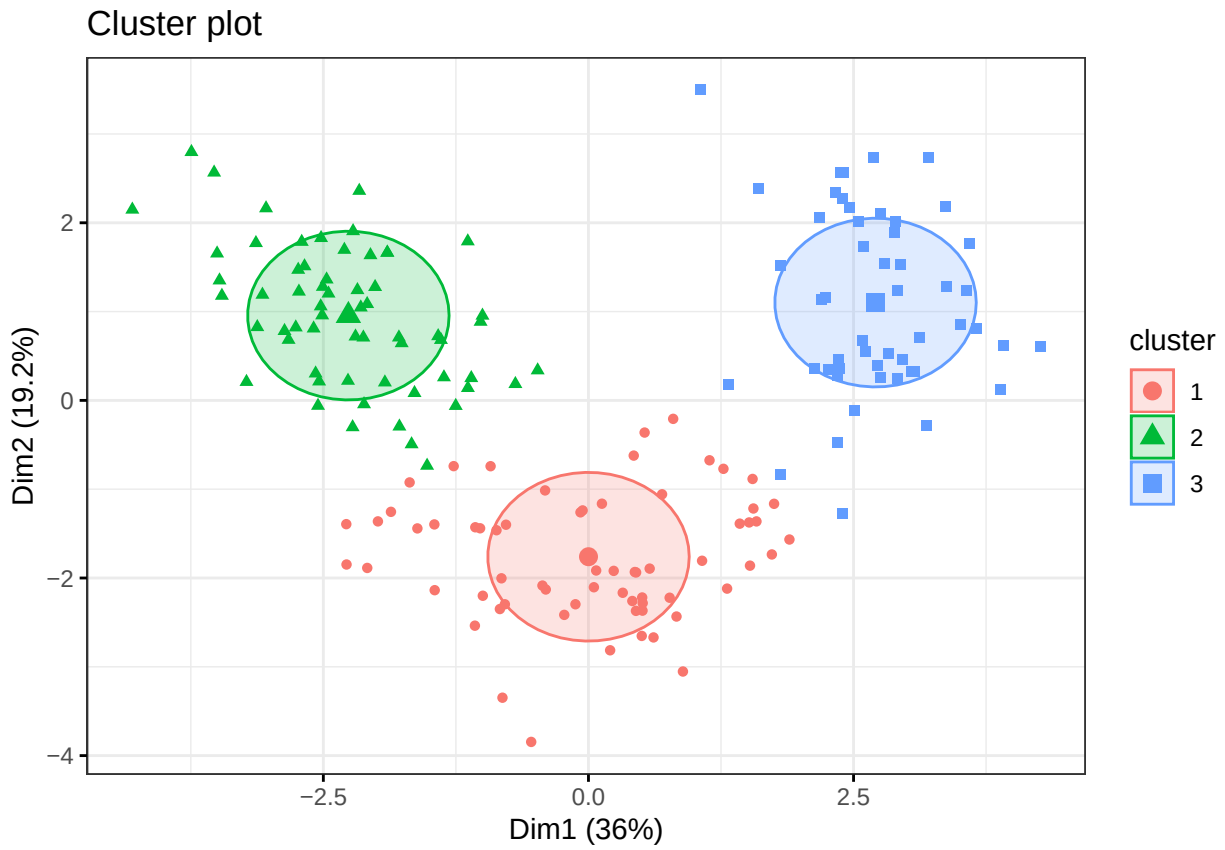
```
## [1] 83.05085
```

accuracy of hierarchical clustering is 83%

Problem 7 - Repeat your analysis or problem 6 using kmeans. First run set.seed(0)

```
set.seed(0)
wines.sc = scale(wines, center = T, scale = T)
#rownames(wines.sc) <- vintages

# kmeans
km <- kmeans(wines.sc, 3, nstart = 20)
```

```
# plot the clusters
fviz_cluster(km, data = wines.sc, geom ="point",
             ellipse.type = "euclid",
             ggtheme = theme_bw())
```



km$size# gives no. of records in each cluster

```
km$size# gives no. of records in each cluster
```

```
## [1] 65 61 51
```

```
class = factor(vintages, levels = c("Barolo", "Grignolino", "Barbera"))
cluster =factor(km$cluster,levels =c(3,1,2))
table(class, cluster)
```

```
##             cluster
## class         3  1  2
##   Barolo      0  0 58
##   Grignolino  3 65  3
##   Barbera    48  0  0
```

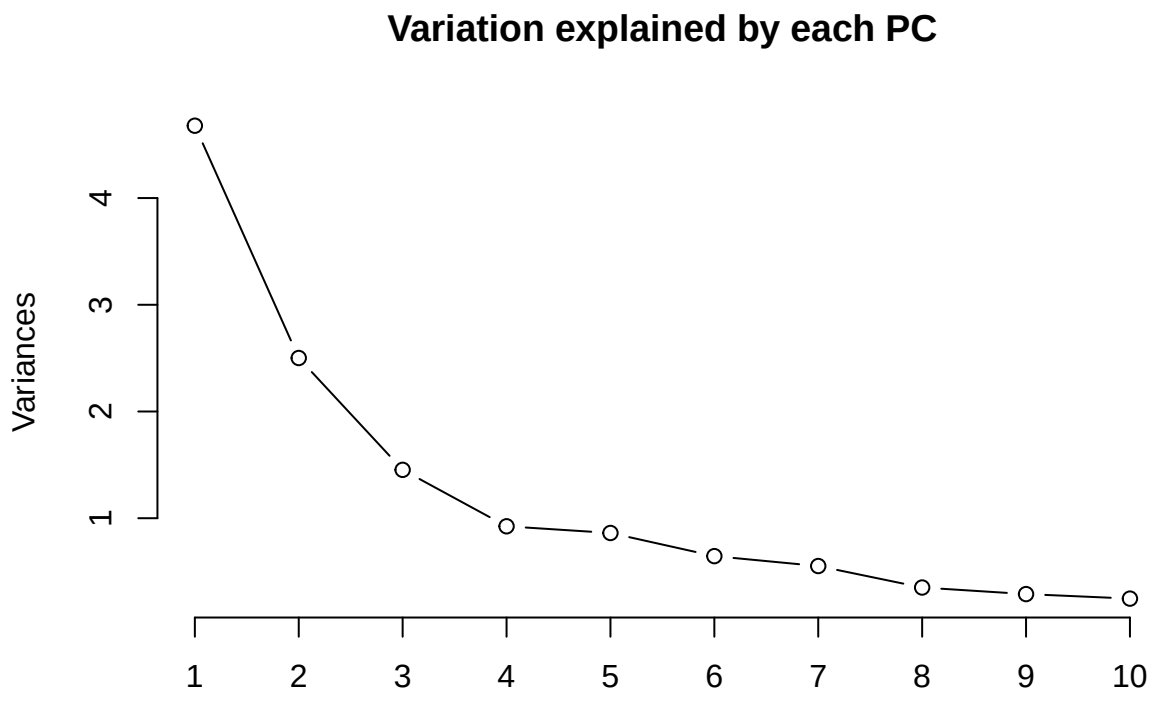
```
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(table(class, cluster))
```

```
## [1] 36.72316
```

accuracy of kmeans clustering is 96%

Problem 8 - Perform principle components analysis on the standardized wine data using prcomp.[Hint - Remember the data has already been centered and scaled]. How many principle components are needed to capture roughly 89% of the variance?

```
pc = prcomp(wines.sc, scale = F, center = F)
plot(pc,type="l", main = "Variation explained by each PC")
```




```
#Calc how much variation in the original data each principal component accounts for
pca.var = pc$sdev^2
pca.var.per = round(pca.var/sum(pca.var)*100,1)

sum(pca.var.per[1:7])
```

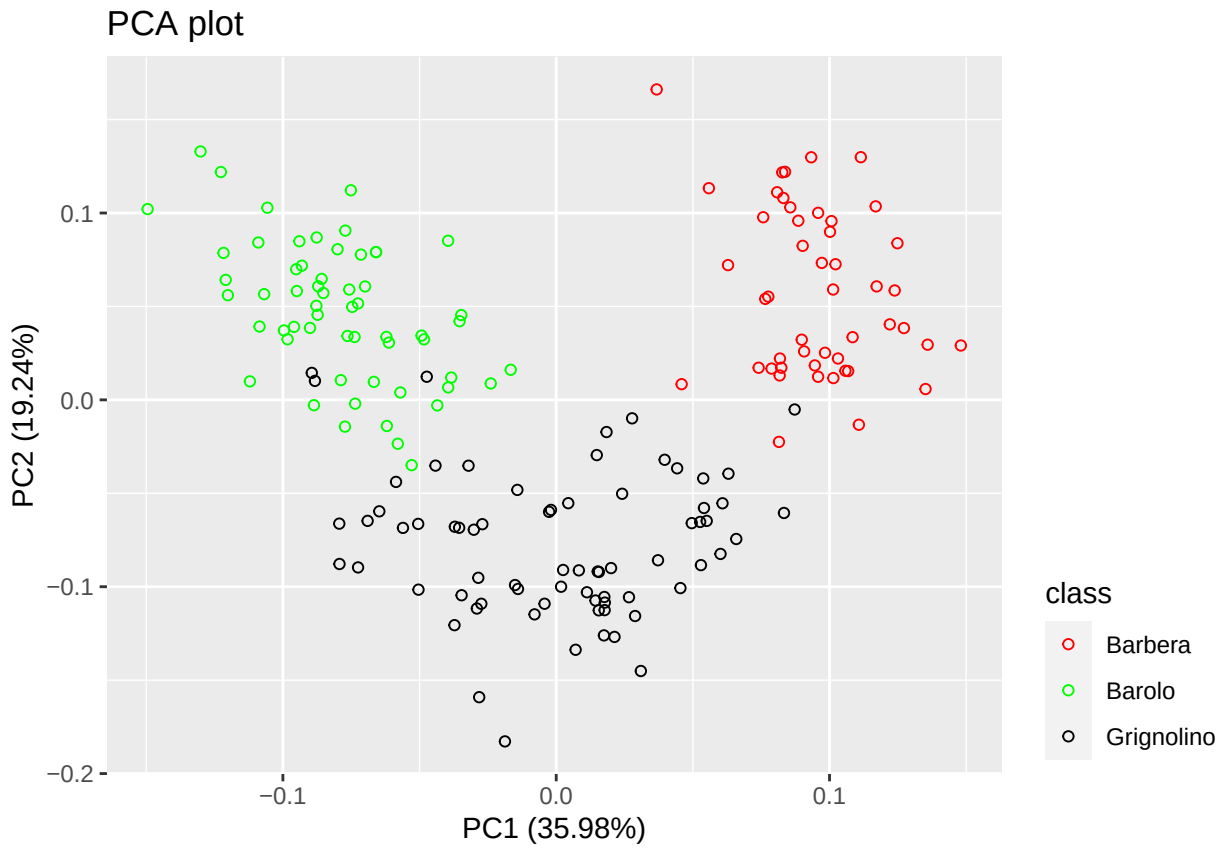
```
## [1] 89.3
```

The first 7 PCs hold 89% of the variance.

Problem 9 - Make a plot of the scaled wines observations in the first two principle components. Plot "o" as the symbol and use red colors for the Barbera, green for the Barolo, and black for the Grigolino. Add in a legend in the bottom left portion of the plot.

```
library(ggfortify)
library(ggplot2)
```

```
wines.sc = scale(wines, center = T, scale = T)
wines.sc = as.data.frame(wines.sc)
class = vintages


dat = wines.sc
dat$class = class
pc <- prcomp(wines.sc, scale = F)

autoplot(pc, data = dat, colour ='class', shape = 'class')+
        ggtitle("PCA plot")+
        scale_colour_manual(values=c('red','green','black'))+
        scale_shape_manual(values=c(1,1,1))+
        theme(legend.position = 'right')+
        theme(legend.justification = "bottom")
```

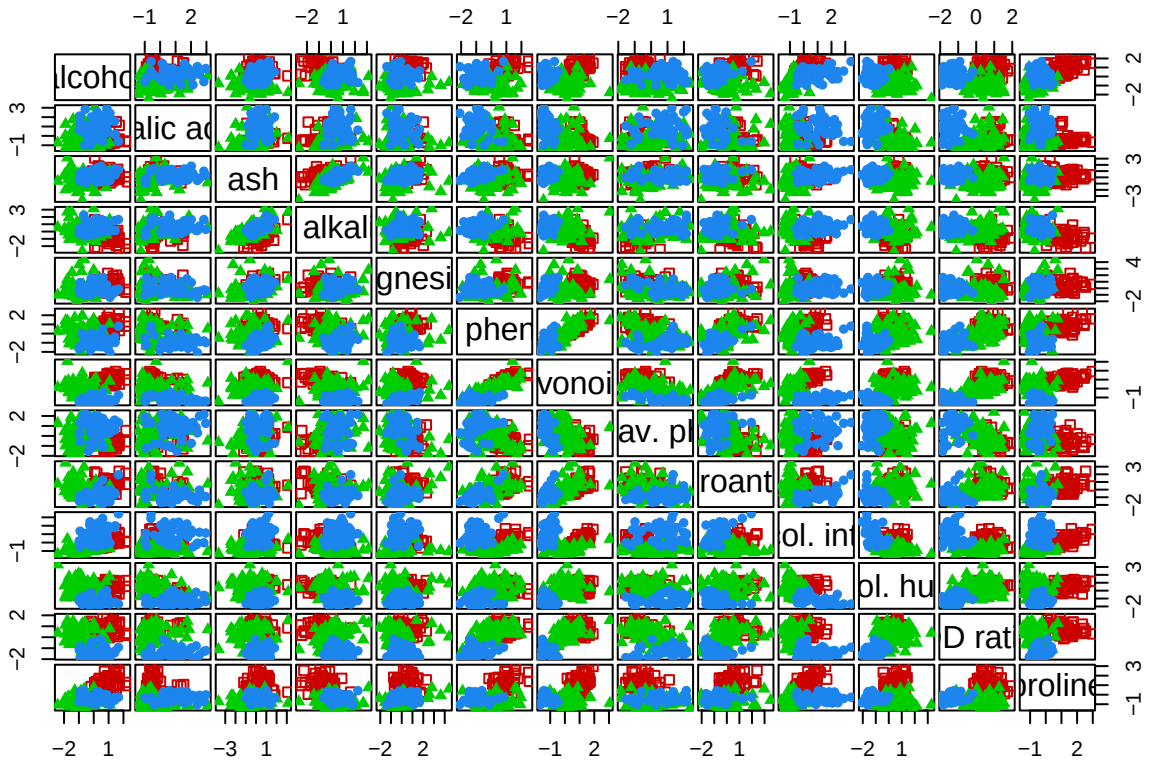


Problem 10 - In this problem we will perform an analysis of the sacled wines data using Gaussian mixture-based clustering. [Hint - We will be using the mclust package and patterning our analysis after the tutorial contained here https://cran.r-project.org/web/packages/mclust/vignettes/mclust.html] Calculate the BIC values using the scaled wines data, call these wines.sc.BIC. plot these BIC values and using the summary method on the BIC object provide an interpretation as to which model is the best. Call Mclust to obtain a full model-based clustering solution on the scaled wines data, call it wines.sc.modl. You will be passing wines.sc to Mclust and setting x = wines.sc.BIC. Use the summary method to examine this object but do not set parameters = TRUE. Finally let's compare the clustering obtained using Gaussian-based mixtures agains the ground truth. Ground truth will be the rows in the table as obtained from the vintages and the columns will be the mixture terms.

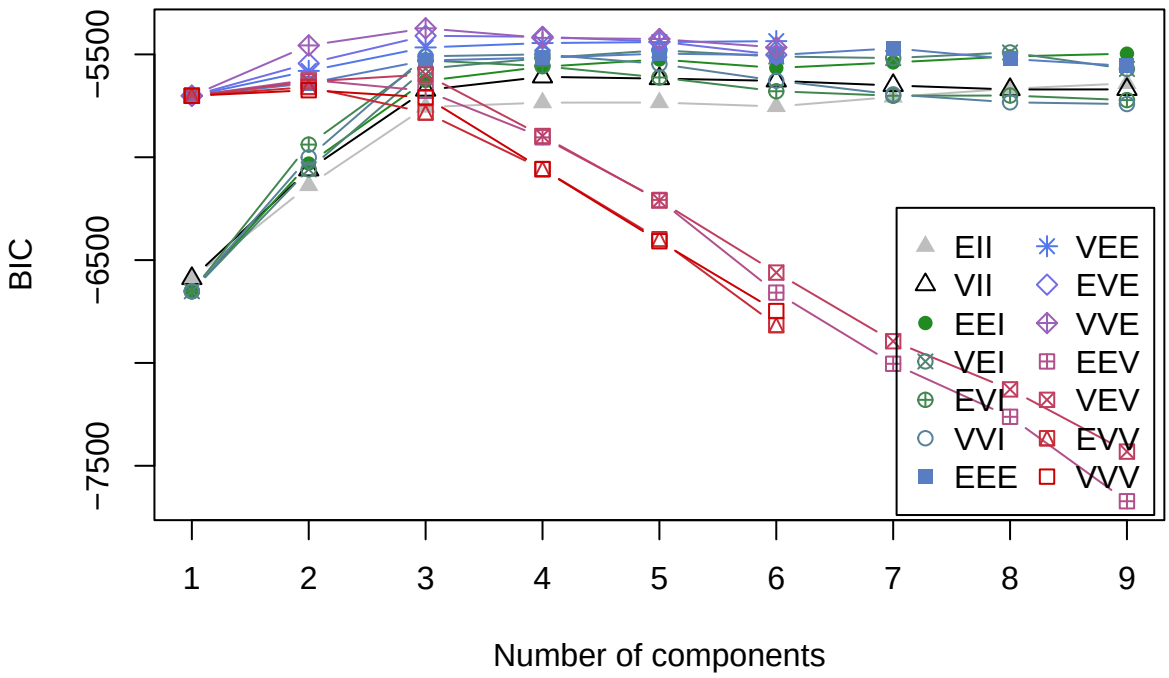
```
library(kohonen)
library(mclust)
data(wines)
wines.sc = scale(wines, center = T, scale = T)
class <- vintages
table(class)
```

```
## class
##    Barbera     Barolo Grignolino
##         48         58         71
```

```
pairs = clPairs(wines.sc, class)
```




```
wines.sc.BIC <- mclustBIC(wines.sc)
plot(wines.sc.BIC)
```



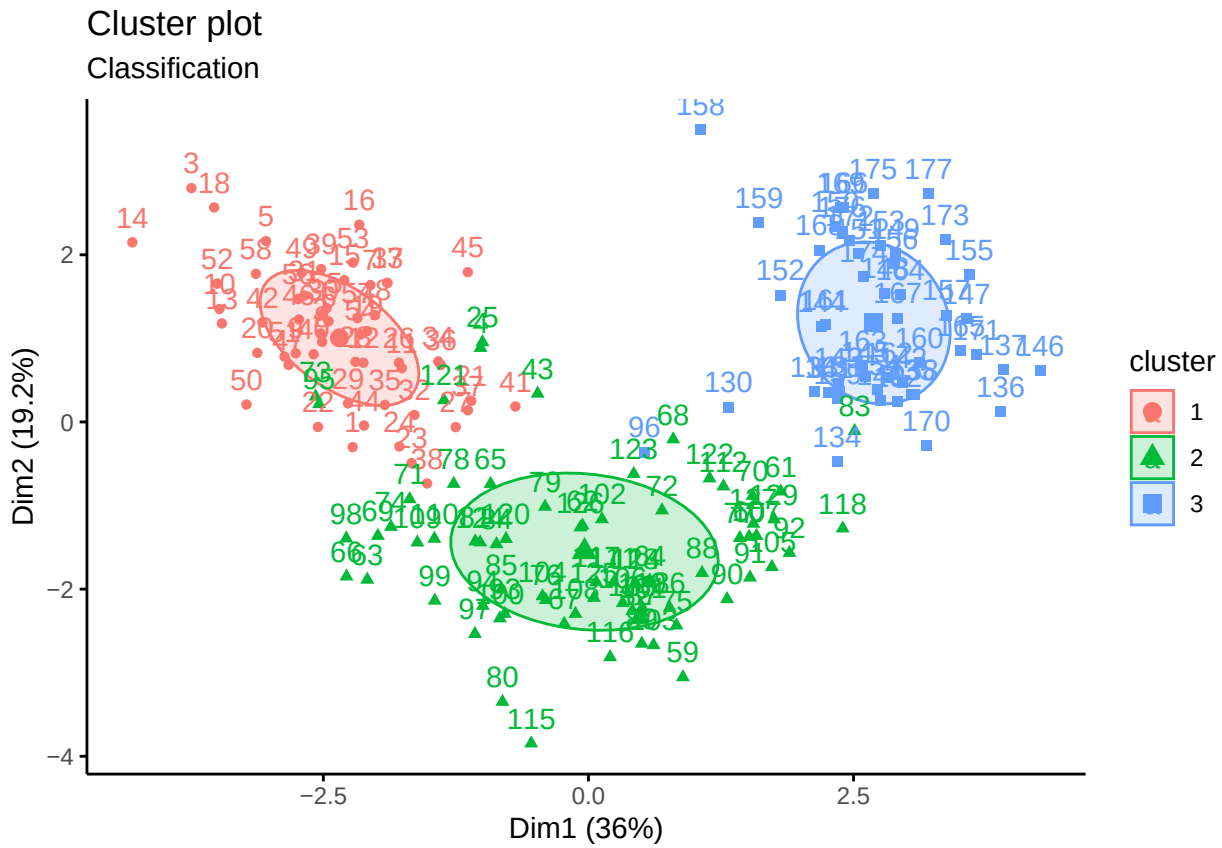

```
summary(wines.sc.BIC)
```

```
## Best BIC values:
##             VVE,3       EVE,3      EVE,4
## BIC      -5373.39 -5409.67362 -5414.3350
## BIC diff     0.00   -36.28374   -40.9451
```

```
mod1 <- Mclust(wines.sc, x = wines.sc.BIC)
summary(mod1, parameters = F)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 3 components:
##
##  log-likelihood   n  df      BIC       ICL
##       -2277.779 177 158 -5373.39 -5374.153
##
## Clustering table:
##  1  2  3
## 55 73 49
```

```
library(factoextra)
fviz_mclust(mod1, 'classification')
```




```
class = factor(vintages, levels = c("Barolo", "Grignolino", "Barbera"))
table(class, mod1$classification)
```

```
##
## class          1  2  3
##   Barolo      55  3  0
##   Grignolino   0 70  1
##   Barbera      0  0 48
```

```
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(table(class, mod1$classification))
```

```
## [1] 97.74011
```

The clustering was highly accurate at 97%