Data Mining For BI: Unsupervised Learning Methods

Radhika Sood Rupanjali Chattopadhyay Rekha Raj Achal Khullar
Ajinkya Dalvi Aman Rastogi Prathiba Swamykannu Umesh Singh

4/28/2020

Unsupervised Learning on Breast Tumor Samples

Background

Data

```
library(MASS)
data("biopsy")
```

The *biopsy* from *MASS* package was used for this project.

Reading *biopsy* dataset from **MASS** package

Libraries

The clustering functions (e.g., *prcomp*, *hclust*, and *kmeans*) used in this project are intrinsic to R-Base package. Other packages were needed for managing, and visualizing data and results.

```
library(readr)
library(dplyr)
library(ggplot2)
library(stringr)
library(gridExtra)
library(grid)
library(cluster)
library(DT)
library(ggplot2)
library(plotly)
library(gapminder)
library(purrr)
library(repurrrsive)
library(tibble)
library(dplyr)
library(tidyr)
library(reshape)
library(ggpmisc)
library(naniar)
library(fpc)
library(cluster)
library(factoextra)
library(fpc)
library(NbClust)
```

biopsy Data Documentation

```
datatable(biopsy, filter = "top", options = list(pageLenght = 5, scrollX=T))
```

Data source: 'biopsy' data documentation

Data columns

- ID: sample code number (not unique)
- V1: clump thickness
- V2: uniformity of cell size.
- V3: uniformity of cell shape.
- V4: marginal adhesion.
- V5: single epithelial cell size.
- V6: bare nuclei (16 values are missing).
- V7: bland chromatin.
- V8: normal nucleoli.
- V9: mitoses.
- class: "benign" or "malignant".

Data obtained from the University of Wisconsin Hospitals, Madison (Wolberg). It is based on assessment of bx of breast tumours for 699 patients. Each of nine attributes V1-V9 is scored on a scale of 1 to 10. The outcome *class* is also known i.e., *benign/malignant*.

Objectives

To cluster the biopsy samples based on features of histopathological slides into two distinct groups, that will correlate with clinical diagnosis (i.e., benign or malignant tumor).

Whether the membership of the cluster, i.e., the samples within the cluster, are distinctively benign or malignant.

In clustering - each cluster is distinct from each other cluster - objects within each cluster are broadly similar to each other.

Aims

To use several unsupervised machine learning algorithms such as **Principal Component Analysis**, **Hierarchical Clustering**, and **K-Means Clustering** for building the unsupervised machine learning model and compare their clustering accuracies based on known diagnosis of the tumors.

Exploratory Data Analysis

```
dim(biopsy)
```

Data Dimension

```
## [1] 699  11
```

```
str(biopsy)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ ID   : chr  "1000025" "1002945" "1015425" "1016277" ...
##  $ V1   : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ V2   : int  1 4 1 8 1 10 1 1 1 2 ...
```

```
##  $ V3   : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ V4   : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ V5   : int  2 7 2 3 2 7 2 2 2 2 ...
##  $ V6   : int  1 10 2 4 1 10 10 1 1 1 ...
##  $ V7   : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ V8   : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ V9   : int  1 1 1 1 1 1 1 1 5 1 ...
##  $ class: Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
```

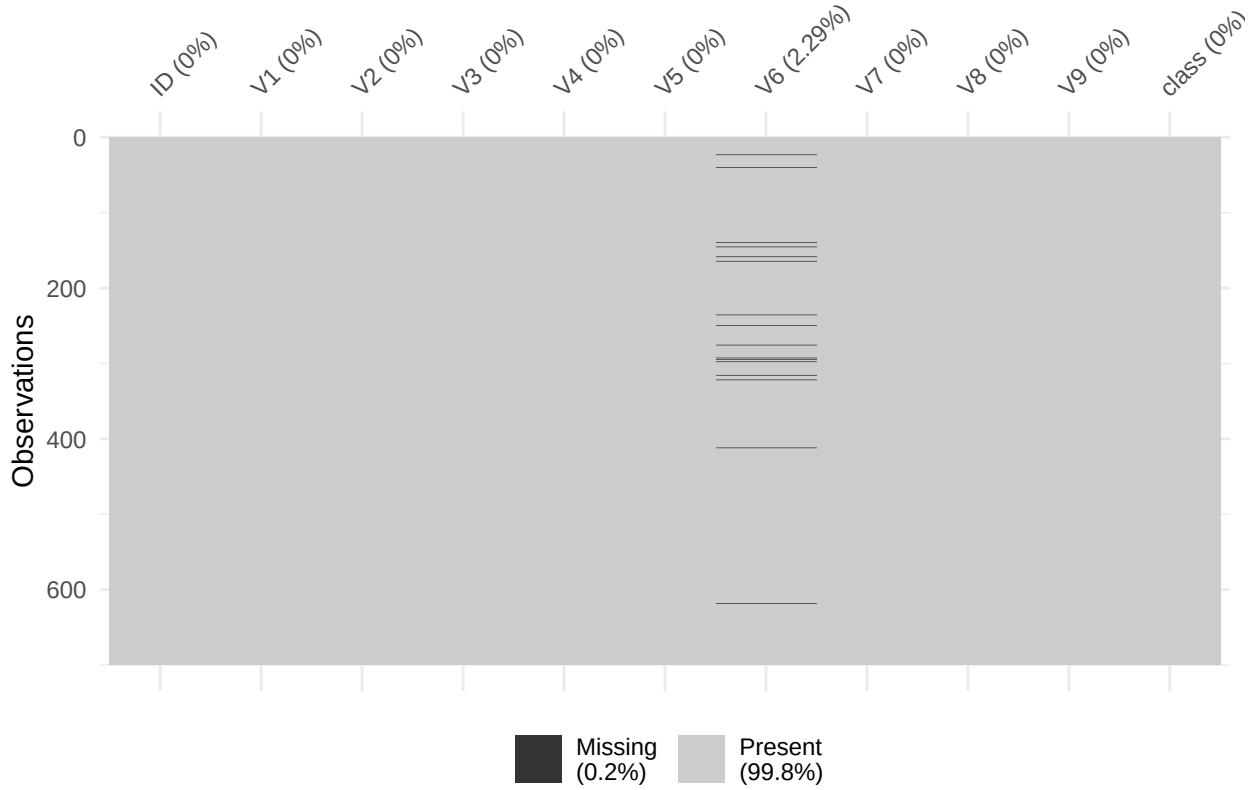
The *original data* has **699 rows and 11 columns**.

```
# check if any missing values in the dat
anyNA(biopsy)
```

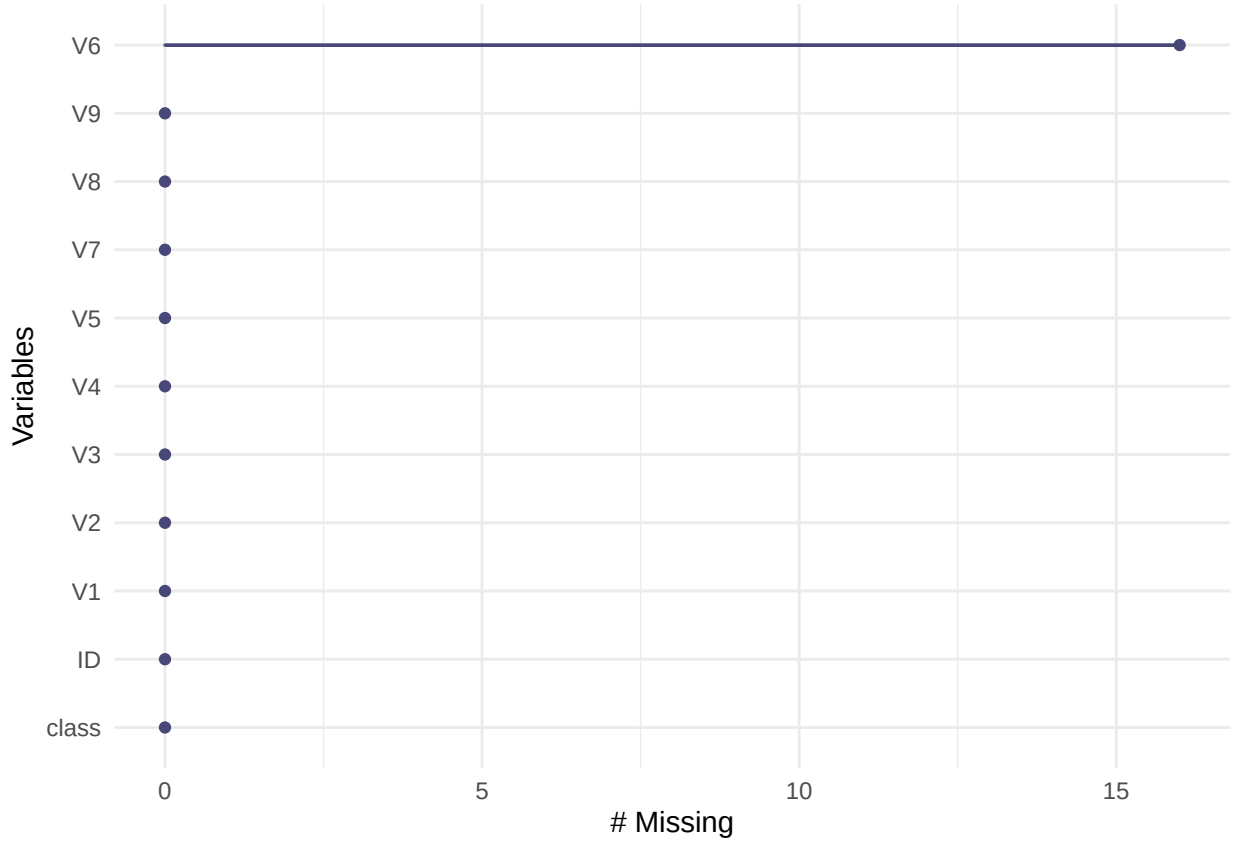
Missingness

```
## [1] TRUE
```

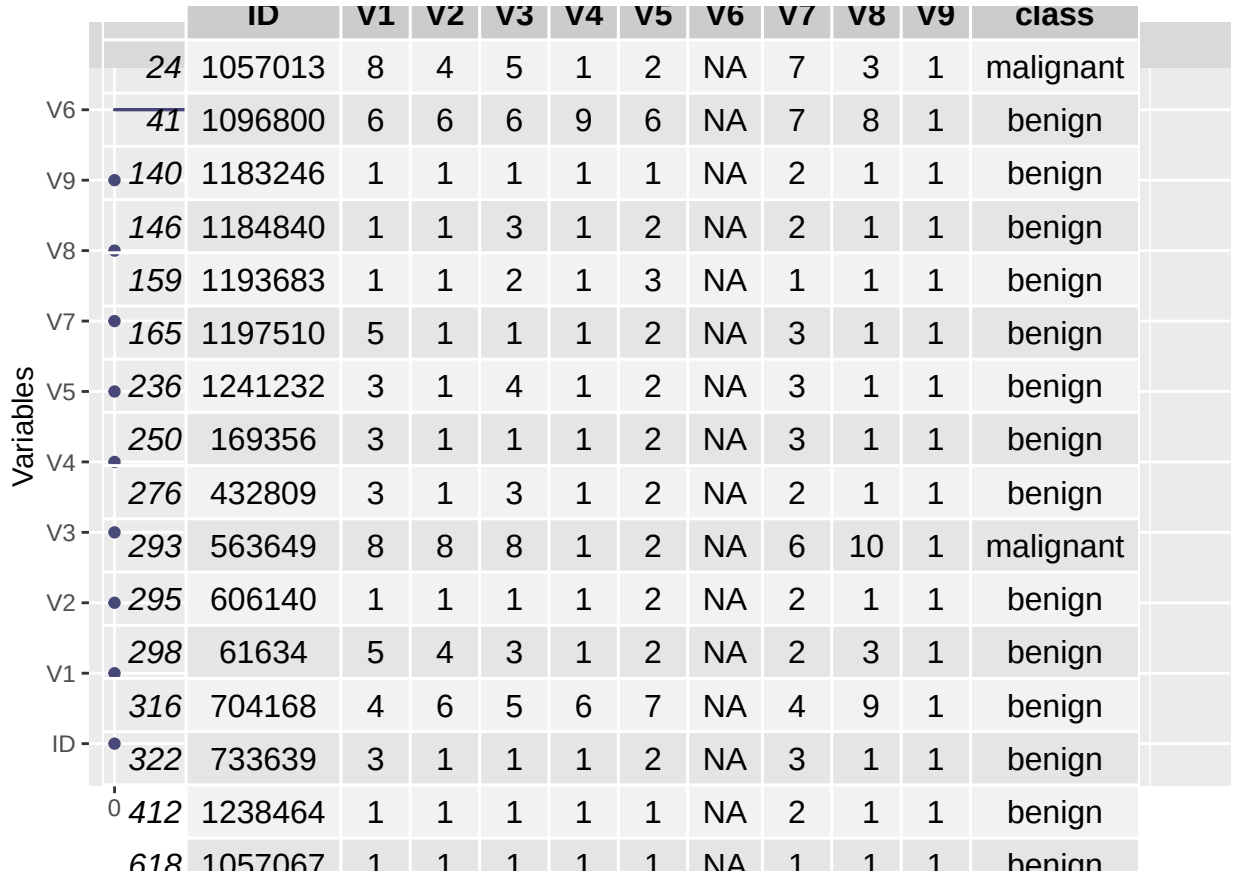
```
# Visualizing missing values for the training data
par(mfrow=c(1,2))
vis_miss(biopsy)
```



```
gg_miss_var(biopsy) + theme_minimal()
```




```
gg_miss_var(biopsy, facet = class) + theme_gray()
# list rows of data that have missing values
grid.table(biopsy[!complete.cases(biopsy),])
```



	ID	V1	V2	V3	V4	V5	V6	V7	V8	V9	class
24	1057013	8	4	5	1	2	NA	7	3	1	malignant
41	1096800	6	6	6	9	6	NA	7	8	1	benign
140	1183246	1	1	1	1	1	NA	2	1	1	benign
146	1184840	1	1	3	1	2	NA	2	1	1	benign
159	1193683	1	1	2	1	3	NA	1	1	1	benign
165	1197510	5	1	1	1	2	NA	3	1	1	benign
236	1241232	3	1	4	1	2	NA	3	1	1	benign
250	169356	3	1	1	1	2	NA	3	1	1	benign
276	432809	3	1	3	1	2	NA	2	1	1	benign
293	563649	8	8	8	1	2	NA	6	10	1	malignant
295	606140	1	1	1	1	2	NA	2	1	1	benign
298	61634	5	4	3	1	2	NA	2	3	1	benign
316	704168	4	6	5	6	7	NA	4	9	1	benign
322	733639	3	1	1	1	2	NA	3	1	1	benign
412	1238464	1	1	1	1	1	NA	2	1	1	benign
618	1057067	1	1	1	1	1	NA	1	1	1	benign

```
# create a subset of complete dataset without missing values
biopsy1 <- na.exclude(biopsy)
dim(biopsy1)
```

```
## [1] 683  11
```

A total of 16 missing values; all for *V6* (i.e., *missing not at random*). These observations were deleted before applying clustering algorithms. The *complete data* has **683 rows and 11 columns**.

```
table(biopsy$class)
```

Diagnosis (Benign/Malignant)

```
##
##    benign malignant
##       458       241
```

```
# Assigning a numeric value to pathological diagnosis based on features on the complete dataset
diagnosis <- as.numeric(biopsy1$class == "benign")
table(biopsy1$class)
```

```
##
##    benign malignant
##       444       239
```

```
table(diagnosis)
```

```
## diagnosis
```

```
##   0   1
## 239 444
```

The last-column i.e., *class* specifies the specific diagnosis of the tumors. This variable will be used to assess the *accuracy* of clustering.

```
biopsy2 <- as.matrix(biopsy1[, 2:10])
str(biopsy2)
```

Data-Matrix

```
##  int [1:683, 1:9] 5 5 3 6 4 8 1 2 2 4 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:683] "1" "2" "3" "4" ...
##   ..$ : chr [1:9] "V1" "V2" "V3" "V4" ...
```

```
head(biopsy2)
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9
## 1  5  1  1  1  2  1  3  1  1
## 2  5  4  4  5  7 10  3  2  1
## 3  3  1  1  1  2  2  3  1  1
## 4  6  8  8  1  3  4  3  7  1
## 5  4  1  1  3  2  1  3  1  1
## 6  8 10 10  8  7 10  9  7  1
```

```
row.names(biopsy2) <- biopsy1$ID
datatable(biopsy2, filter = "top", options = list(pageLenght = 5, scrollX=T))
```

```
#head(biopsy2)
```

Creating a data matrix of the attributes (numeric). Unsupervised learning methods will be applied on this matrix.

```
mdata <- melt(biopsy, id=c("ID","class"))
p <- ggplot(data = mdata, aes(x=variable, y=value)) + geom_boxplot(aes(fill=class))
p + facet_wrap( ~ variable, scales="free")
```

Boxplots of Attributes

Malignant tumors have higher values, on the scale of 1 to 10, for all the features compared to benign tumors.

Unsupervised Learning Methods

Principal Component Analysis

PCA is particularly useful when working with "wide" data sets. In datasets with many variables, it is often difficult to plot the data in its raw format, making it difficult to determine the trends present within the dataset. PCA enables visualization of the "shape" of the data, identifying which samples are similar to one another and which are very different. This can enable identification of groups of samples that are similar and determine which variables make one group different from another. DataCamp

```
biopsy.pr <- prcomp(biopsy2, scale = T, center = T)
```

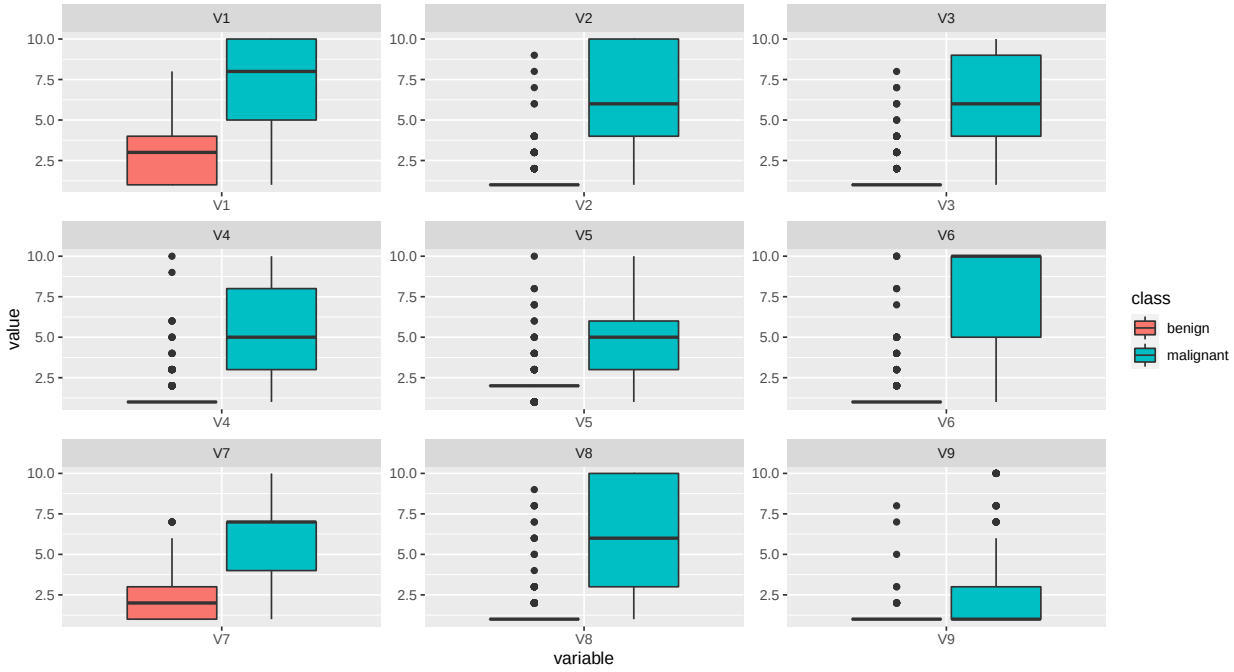


Figure 1: Box-plots of Attributes

Execute

Applying *prcomp* function (from R-Base Package) to execute principal component analysis after scaling and centering the features The prinicipal components are stored as an object *biopsy.pr*.

Summarizing Principal Components PCA is a type of linear transformation on a given data set that has values for a certain number of variables (coordinates) for a certain amount of spaces. This linear transformation fits this dataset to a new coordinate system in such a way that the most significant variance is found on the first coordinate, and each subsequent coordinate is orthogonal to the last and has a lesser variance. In this way, you transform a set of x correlated variables over y samples to a set of p uncorrelated principal components over the same samples.DataCamp

```
summary(biopsy.pr)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4289 0.88088 0.73434 0.67796 0.61667 0.54943 0.54259
## Proportion of Variance 0.6555 0.08622 0.05992 0.05107 0.04225 0.03354 0.03271
## Cumulative Proportion  0.6555 0.74172 0.80163 0.85270 0.89496 0.92850 0.96121
##                           PC8     PC9
## Standard deviation     0.51062 0.29729
## Proportion of Variance 0.02897 0.00982
## Cumulative Proportion  0.99018 1.00000
```

There are nine prinicipal comoponents of which the first component (PC1) itself explains about 65% of the variability in the data, as shown by *Proportion of Variance* = 0.65; and the first two components (PC1, PC2) explain about 74% (as shown by the *Cumulative proportion* = 0.74 under PC2) of the variability. Thus, variability explained by all nine features can be explained by values of PC1 and PC2 only (*dimensionality reduction*).

Interpretation

```
library(ggbiplot)
biplot<- ggbiplot(biopsy.pr, pc.biplot = TRUE, scale= TRUE, obs.scale = 1, groups= diagnosis, labels = c
ggplotly(biplot)
```

Biplot Biplot of PC1 and PC2

The correlation circle visualizes the correlation between the first two principal components and the 9 dataset features. All the 8 features (V1-V8) are aligned close together and parallel to PC1 axis. Only one feature, V9, is aligned orthogonal to others and parallel to PC2. Thus, PC1 alone explains most of the variability explained by all the 8 features (V1-V8) and combined with PC2, can explain all the variability in teh data and differentiate between benign and malignant tumors.

- Features with a positive correlation are grouped together.
- Uncorrelated feature (V9) is orthogonal to other features.
- Features with a negative correlation will be plotted on the opposing quadrants of this plot.

PC: Scatterplots PC1 vs PC2

```
library(tidyverse)
plot(biopsy.pr$x[, c(1, 2)], col = (diagnosis + 1),
     xlab = "PC1", ylab = "PC2")
```

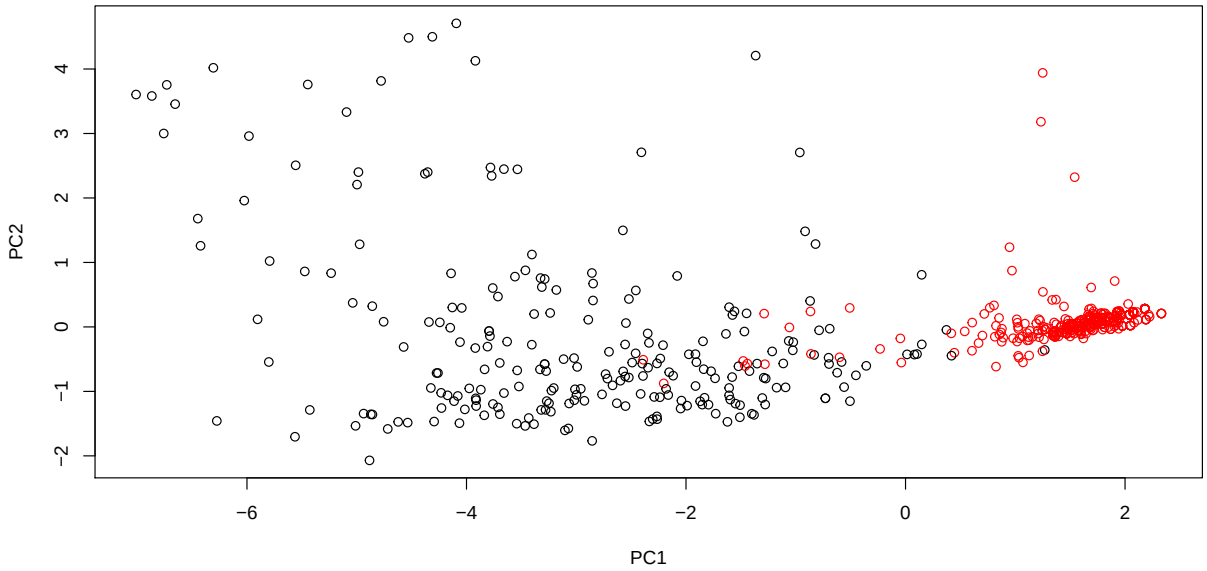


Figure 2: Scatter plot observations by components 1 and 2

PC1 vs PC3

```
# Repeat for components 1 and 3
plot(biopsy.pr$x[, c(1, 3)], col = (diagnosis + 1),
     xlab = "PC1", ylab = "PC3")
```

PC2 vs PC3

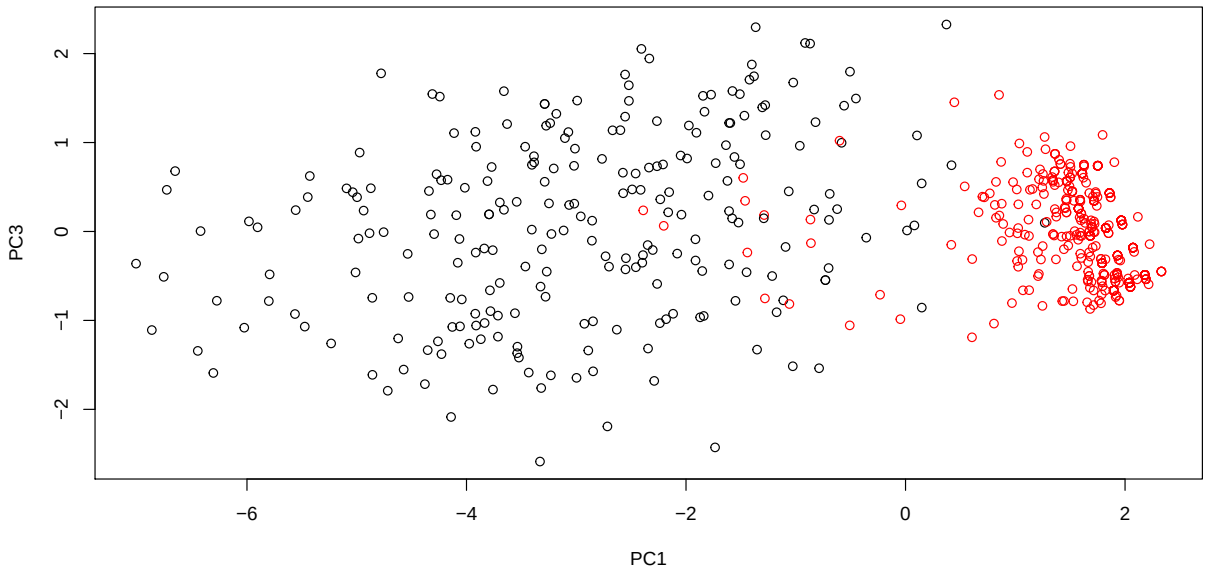


Figure 3: Scatter plot observations by components 1 and 3

```
# Do additional data exploration of your choosing below (optional)
plot(biopsy.pr$x[, c(2, 3)], col = (diagnosis + 1),
     xlab = "PC2", ylab = "PC3")
```

```
par(mfrow = c(1, 2))
# Calculate variability of each component: pr.var
pr.var <- biopsy.pr$sdev^2

# Variance explained by each principal component: pve
pve <- pr.var / sum(pr.var)

# Plot variance explained for each principal component
plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "b")

# Plot cumulative proportion of variance explained
plot(cumsum(pve), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained",
     ylim = c(0, 1), type = "b")
```

PC: Variance

The first two principal component explain most of the variability in the data

Hierarchical Clustering

Preprocessing

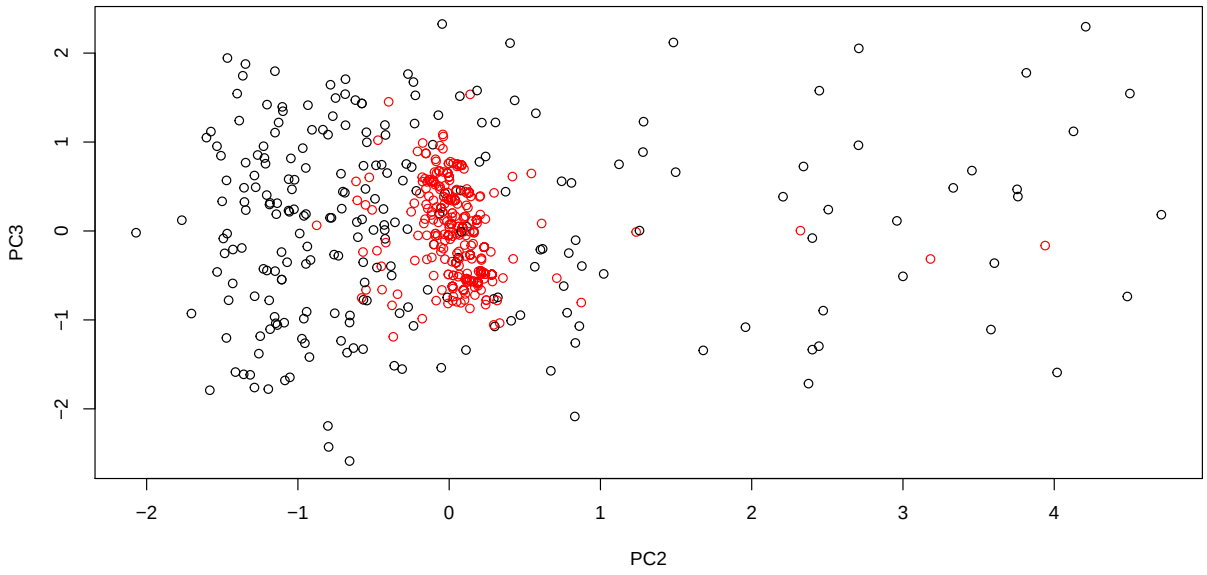


Figure 4: Scatter plot observations by components 2 and 3

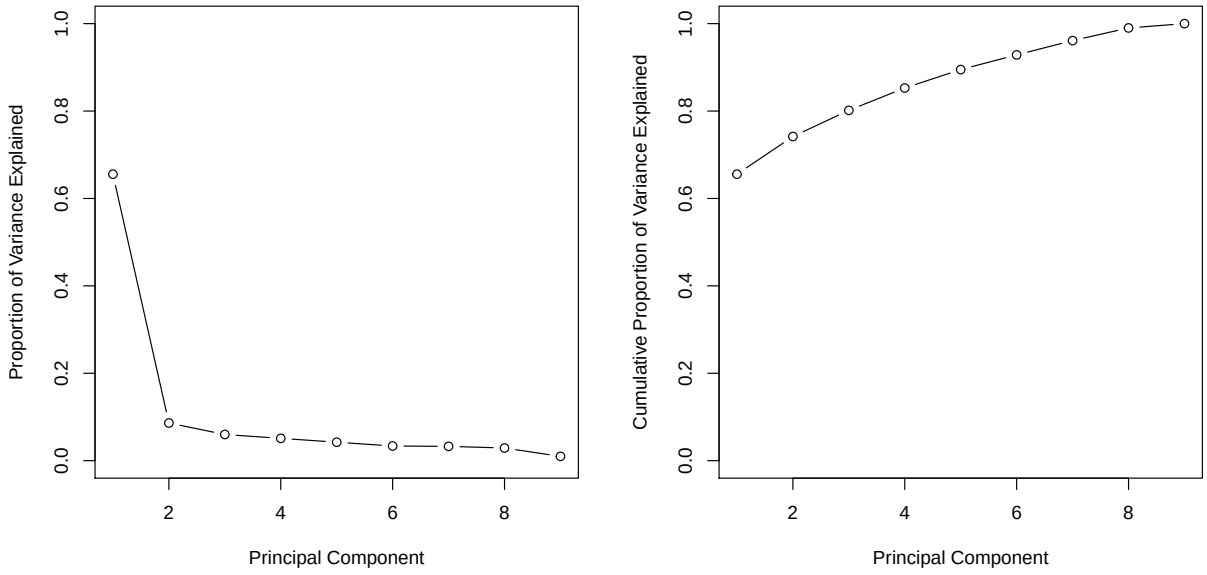


Figure 5: Variance explained by each component and cumulative variance by all components

```
# Scale the biopsy2 data: data.scaled
data.scaled <- scale(biopsy2)
head(data.scaled)
```

Scale

```
##                 V1         V2         V3          V4         V5         V6
## 1000025  0.1977598 -0.7016978 -0.7412304 -0.63889730 -0.5552016 -0.6983413
## 1002945  0.1977598  0.2770488  0.2625905  0.75747664  1.6939247  1.7715689
## 1015425 -0.5112687 -0.7016978 -0.7412304 -0.63889730 -0.5552016 -0.4239068
## 1016277  0.5522740  1.5820442  1.6010185 -0.63889730 -0.1053763  0.1249621
## 1017023 -0.1567545 -0.7016978 -0.7412304  0.05928967 -0.5552016 -0.6983413
## 1017122  1.2613024  2.2345419  2.2702324  1.80475710  1.6939247  1.7715689
##                V7         V8         V9
## 1000025 -0.181694 -0.6124785 -0.3481446
## 1002945 -0.181694 -0.2848960 -0.3481446
## 1015425 -0.181694 -0.6124785 -0.3481446
## 1016277 -0.181694  1.3530163 -0.3481446
## 1017023 -0.181694 -0.6124785 -0.3481446
## 1017122  2.267589  1.3530163 -0.3481446
```

Scaling feature values before clustering process: Feature values from each row are represented as coordinates in n-dimensional space (n is the number of features) and then the distances between these coordinates are calculated. If these coordinates are not normalized, then it may lead to false results. Ref: Hierarchical Clustering in R

Euclidean-Dist Euclidean distance is used as an input for the clustering algorithm. The proximity matrix containing the distance between each point is determined using a distance function.

```
# Calculate similarity as Euclidean distance between observations
data.dist <- dist(data.scaled, method = "euclidean")
```

Calculated (Euclidean) distance is stored as an object *data.dist*

H-clustering Model

```
biopsy.hclust <- hclust(data.dist, method = "complete")
biopsy.hclust2 <- hclust(data.dist, method = "mcquitty")
```

Creating Model: Linkage

Create a hierarchical clustering model using *hclust* function and two separate methods (i.e. "complete" and "mcquitty"); both models are stored as objects: *biopsy.hclust* and *biopsy.hclust2*

Details on hclust

```
plot(biopsy.hclust)
```

```
plot(biopsy.hclust2)
```

Dendrogram: H-Clusters

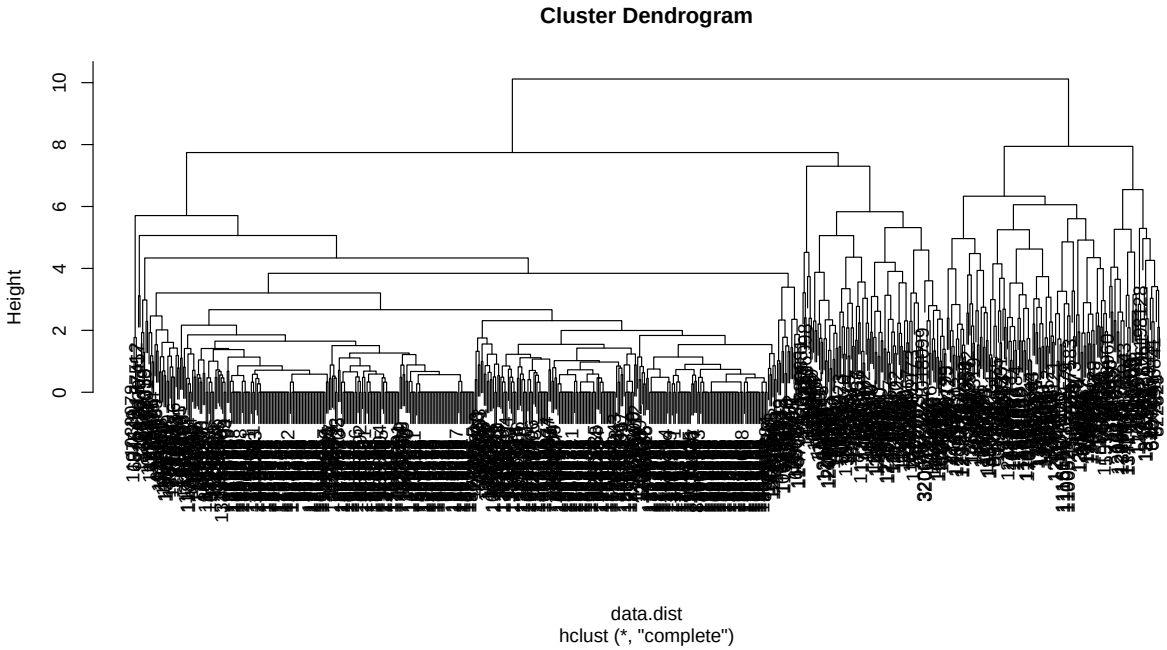


Figure 6: Results of hierarchical clustering

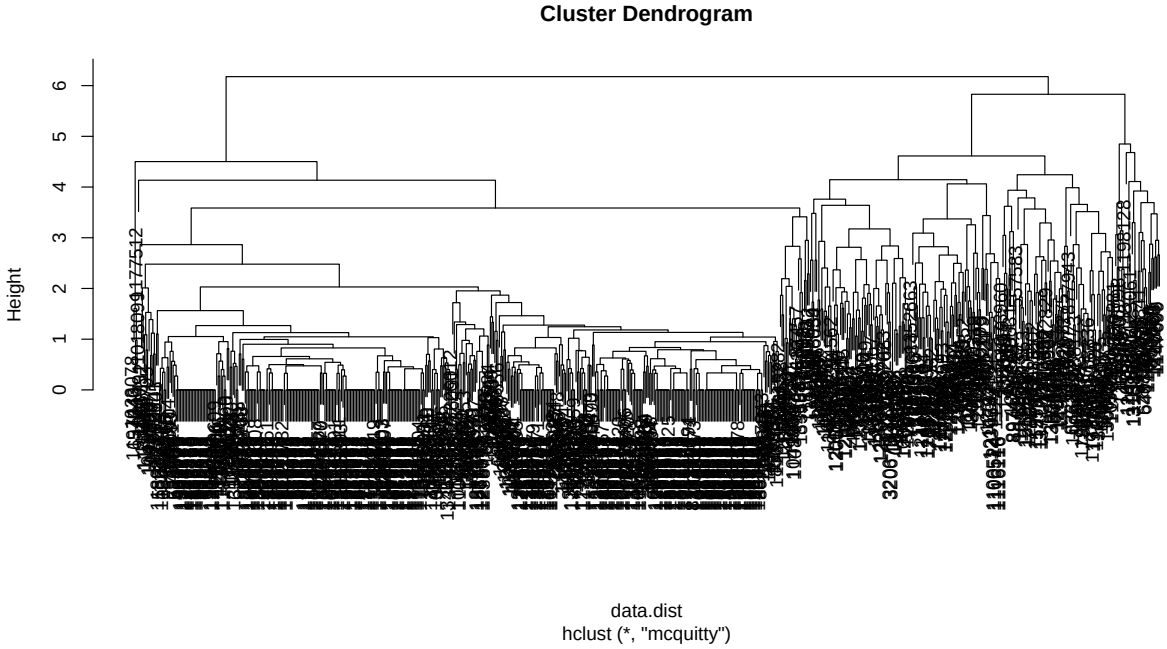


Figure 7: Results of hierarchical clustering

Cutting the height at 9 will give 2 clusters

```
# Cut by number of clusters k
plot(biopsy.hclust)
biopsy.hclust.clusters <- cutree(biopsy.hclust, k = 2)
rect.hclust(biopsy.hclust, k=2, border="red")
```

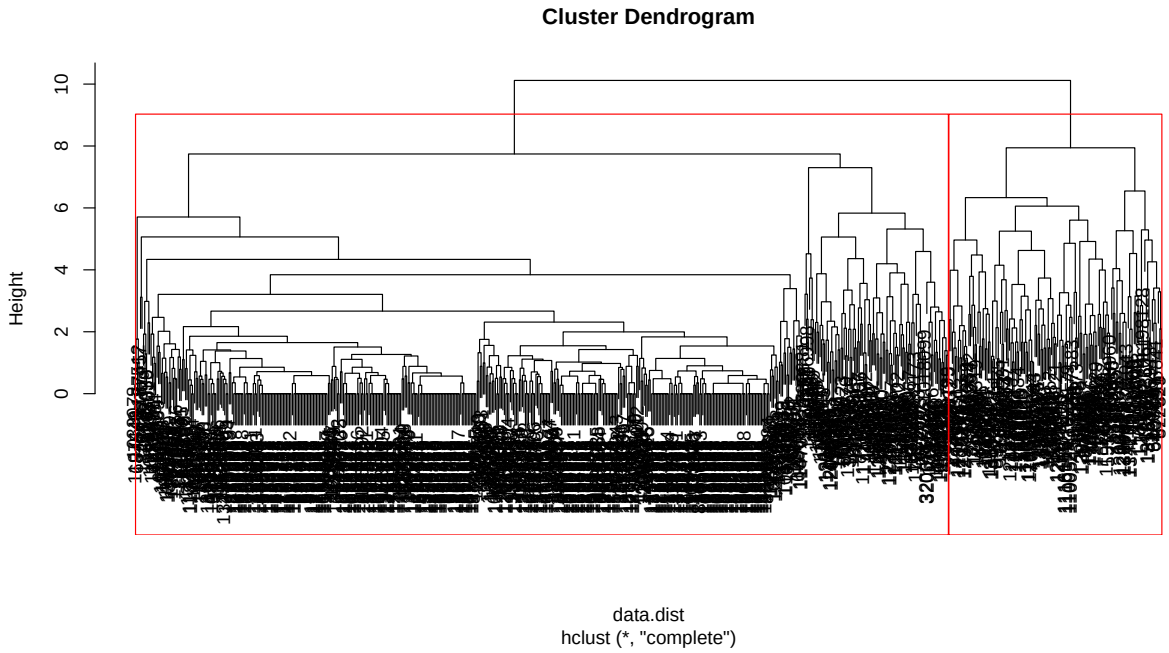


Figure 8: Results of hierarchical clustering

```
plot(biopsy.hclust2)
biopsy.hclust2.clusters <- cutree(biopsy.hclust2, k = 2)
rect.hclust(biopsy.hclust2, k=2, border="red")
```

Dendrogram: Outlining H-Clusters

Using cutree() on biopsy.hclust, assign cluster membership to each observation. Assumed two clusters and assigned the result to a vector called biopsy.hclust.clusters.

Evaluating H-Clusters H-Clusters vs Actual

```
# Clusters using 'complete' method
table(biopsy.hclust.clusters)
```

```
## biopsy.hclust.clusters
##   1   2
## 541 142
```

```
thc <- table(biopsy.hclust.clusters, biopsy1$class)
thc
```

```
##
## biopsy.hclust.clusters benign malignant
```

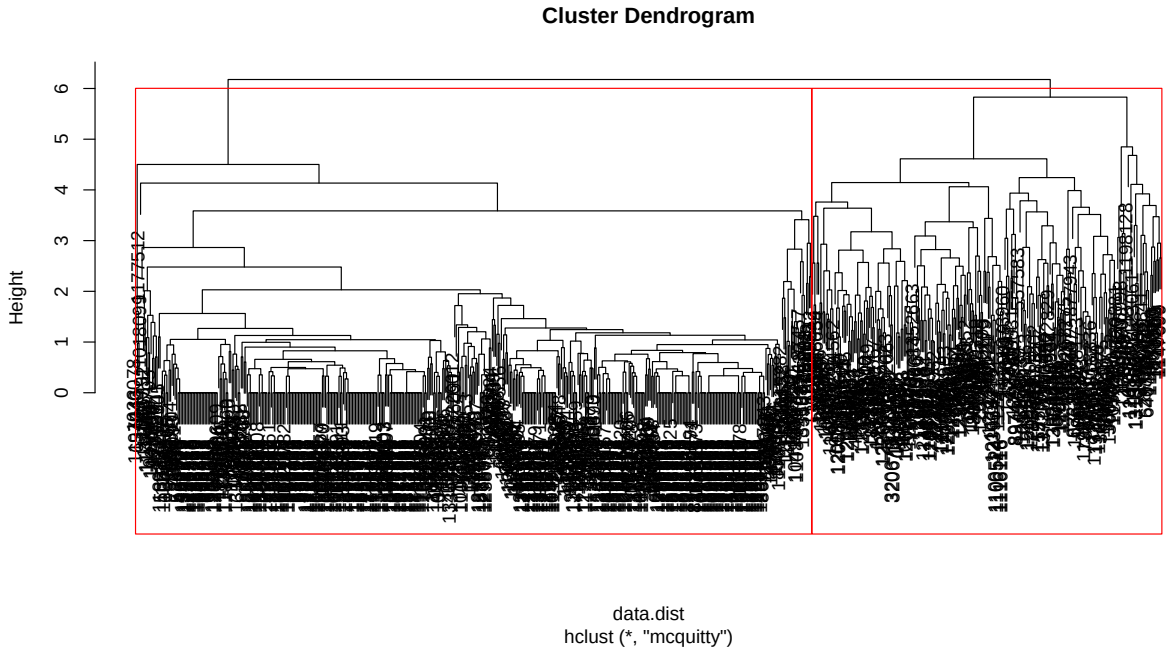


Figure 9: Results of hierarchical clustering

```
##                         1    442        99
##                         2      2       140
```

```
# Clusters using 'mcquitty' method
table(biopsy.hclust2.clusters)
```

```
## biopsy.hclust2.clusters
##   1   2
## 450 233
```

```
thc2 <- table(biopsy.hclust2.clusters, biopsy1$class)
thc2
```

```
##
## biopsy.hclust2.clusters benign malignant
##                         1    435        15
##                         2      9       224
```

Compare cluster membership to actual diagnoses based on 'complete' and 'mcquitty' method of hclust

Sample Errors By H-Clustering

```
sum(apply(table(biopsy.hclust.clusters, diagnosis), 1, min))
```

```
## [1] 101
```

```
sum(apply(table(biopsy.hclust2.clusters, diagnosis), 1, min))
```

```
## [1] 24
```

- Count out of place observations based on cluster by summing the row minimums

Based on "complete" h-clustering method, **101** tumors do not agree with the actual diagnosis Based on

"mcquitty" h-clustering method, **24** tumors do not agree with the actual diagnosis

H-Clustering Model Accuracy

complete method

```
torg<-table(biopsy1$class)
biop <- c("benign", "malignant")
actual <- factor(rep(biop, times = c(torg[1], torg[2])), levels = rev(biop))
predhc <- factor(
        c(
                rep(biop, times = c(thc[1], thc[2])),
                rep(biop, times = c(thc[3], thc[4]))),
        levels = rev(biop))
xtab.hclust <- table(predhc, actual)
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
cmhclust <- confusionMatrix(xtab.hclust)
cmhclust$overall['Accuracy']
```

```
## Accuracy
## 0.852123
```

Accuracy H-Clust *complete* method is **0.85**

mcquitty method

```
biop <- c("benign", "malignant")
actual <- factor(rep(biop, times = c(torg[1], torg[2])), levels = rev(biop))
predhc2 <- factor(
        c(
                rep(biop, times = c(thc2[1], thc2[2])),
                rep(biop, times = c(thc2[3], thc2[4]))),
        levels = rev(biop))
xtab.hclust2 <- table(predhc2, actual)
library(caret)
cmhclust2 <- confusionMatrix(xtab.hclust2)
cmhclust2$overall['Accuracy']
```

```
##  Accuracy
## 0.9648609
```

Accuracy H-Clust *mcquitty* method is **0.96**

K-Means Clustering

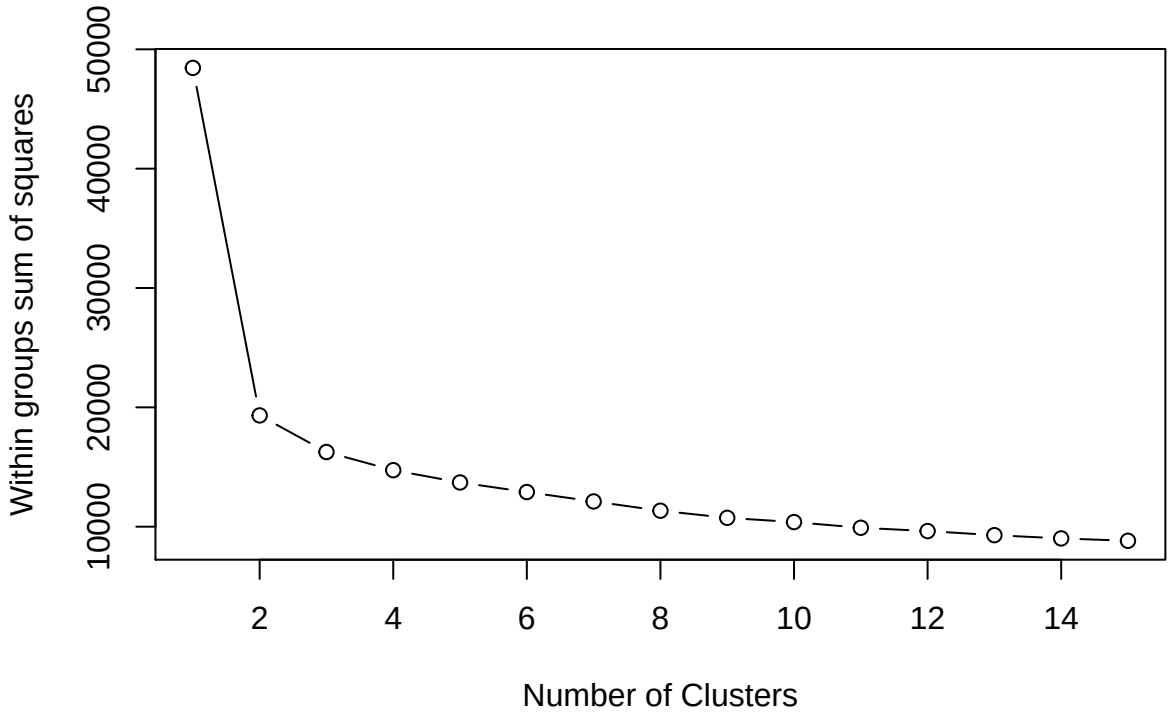
The data are clustered by the k-means method, which aims to partition the points into k groups such that the sum of squares from points to the assigned cluster centres is minimized.

Find K

```
# Initialize total within sum of squares error: wss
wss <- 0
# Look over 1 to 15 possible clusters
for (i in 1:15) {
  # Fit the model: km.out
  km.out <- kmeans(biopsy2, centers = i, nstart = 20, iter.max = 50)
  # Save the within cluster sum of squares
  wss[i] <- km.out$tot.withinss
}
```

WSS kmeans documentation

```
# Produce a scree plot
plot(1:15, wss, type = "b",
     xlab = "Number of Clusters",
     ylab = "Within groups sum of squares")
```



Scree Plot

Build KMeans Model

Fitting a k-means model to the data using 2 centers and run the k-means algorithm 20 times. The result will be stored in biopsy.km

```
set.seed(4)
# Select number of clusters
k <- 2
biopsy.km <- kmeans(scale(biopsy2), centers = 2, nstart = 20, iter.max = 10, algorithm = c("Hartigan-Wo
```

KM Cluster Model created using *kmeans* function while applying scaling on features, and stored as an object *biopsy.km*

Clusters The cluster membership of the biopsy.km model object is contained in its cluster component and is accessed with the $ operator.

```
clusplot(biopsy2, biopsy.km$cluster, main='2D representation of the Cluster solution',
         color=TRUE, shade=TRUE, labels=2, lines=0)
```

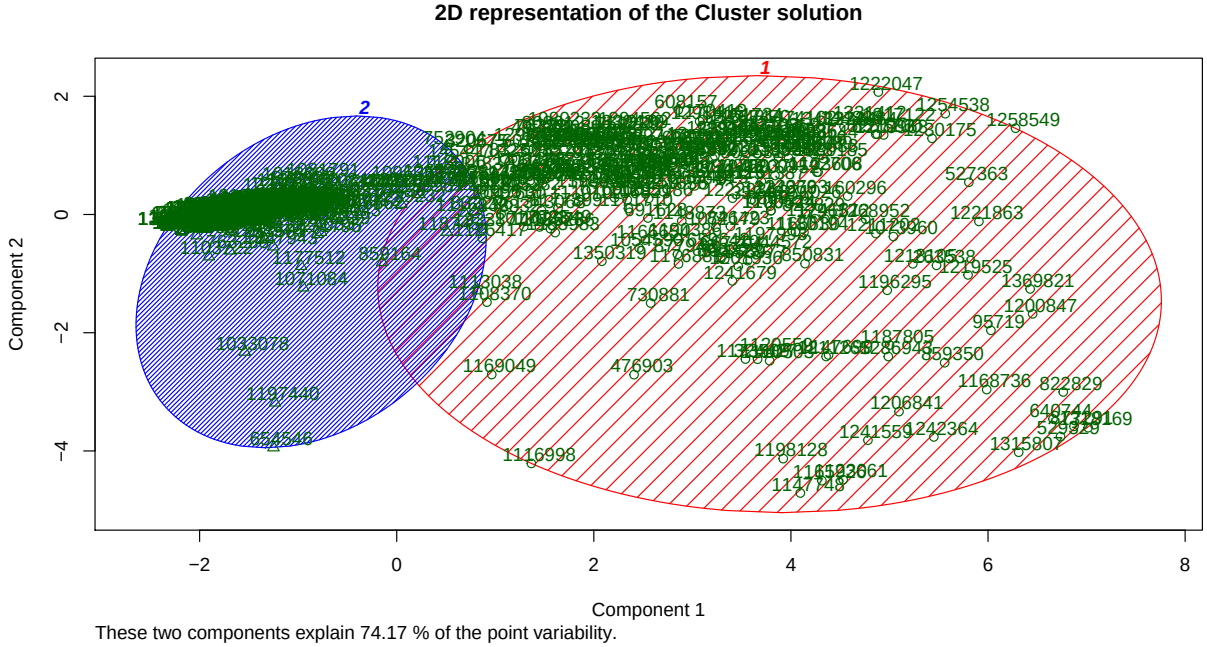


Figure 10: Results of K-Means clustering

Evaluating KM Clusters

```
table(biopsy.km$cluster)
```

KM Clusters vs Actual

```
##
##   1   2
## 230 453
```

```
tkmc <- table(biopsy.km$cluster, biopsy1$class)
tkmc
```

```
##
##     benign malignant
##   1     10       220
##   2    434        19
```

Compare cluster membership to actual diagnoses based on K-Means clustering Based on K-Means Clustering, two clusters of 453 and 230 samples are created. In the former group of 453, the actual number of benign samples are 434 and malignant samples are 19. Of the 230 samples in the second cluster, there are 10 benign samples and 220 malignant samples

```
sum(apply(table(biopsy.km$cluster, diagnosis), 1, min))
```

Errors in KM-Clusters

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

Number of Counts out of place observations based on cluster by summing the row minimums Based on the K-Means clustering, **29** tumors do not agree with the actual diagnosis

KMeans Model Accuracy

```
torg <-table(biopsy1$class)
biop <- c("benign", "malignant")
actual <- factor(rep(biop, times = c(torg[1], torg[2])), levels = rev(biop))
predkm <- factor(
        c(
                rep(biop, times = c(tkmc[2], tkmc[1])),
                rep(biop, times = c(tkmc[4], tkmc[3]))),
        levels = rev(biop))
xtab.kmeans <- table(predkm, actual)
library(caret)
cmkmeans<-confusionMatrix(xtab.kmeans)
cmkmeans$overall['Accuracy']
```

```
##  Accuracy
## 0.9575403
```

Accuracy of K-Means clustering method is 0.957

H-Clustering Using Principal Components

```
summary(biopsy.pr)
```

Recall PCA Summary

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4289 0.88088 0.73434 0.67796 0.61667 0.54943 0.54259
## Proportion of Variance 0.6555 0.08622 0.05992 0.05107 0.04225 0.03354 0.03271
## Cumulative Proportion  0.6555 0.74172 0.80163 0.85270 0.89496 0.92850 0.96121
##                           PC8     PC9
## Standard deviation     0.51062 0.29729
## Proportion of Variance 0.02897 0.00982
## Cumulative Proportion  0.99018 1.00000
```

```
biopsy.pr.hclust <- hclust(dist(biopsy.pr$x[, 1:9]), method = "complete")
biopsy.pr.hclust2 <- hclust(dist(biopsy.pr$x[, 1:9]), method = "mcquitty")
```

PC: HC Model

Create a hierarchical clustering model and stored as object *biopsy.pr.hclust*

```
biopsy.pr.hclust.clusters <- cutree(biopsy.pr.hclust, k = 2)
biopsy.pr.hclust2.clusters <- cutree(biopsy.pr.hclust2, k = 2)
```

Cut Model

Cut model into 2 clusters and stored as an object *biopsy.pr.hclust.clusters*

Evaluation

Matrix **complete** method

```
# Compare to actual diagnoses
table(biopsy.pr.hclust.clusters)
```

```
## biopsy.pr.hclust.clusters
##   1   2
## 544 139
```

```
tpc <- table(biopsy.pr.hclust.clusters, biopsy1$class)
tpc
```

```
##
## biopsy.pr.hclust.clusters benign malignant
##                         1    442       102
##                         2      2       137
```

```
sum(apply(tpc, 1, min))
```

```
## [1] 104
```

104 observations were not clustered accurately using the hierarchical clustering of principal components

mcquitty method

```
# Compare to actual diagnoses

table(biopsy.pr.hclust2.clusters)
```

```
## biopsy.pr.hclust2.clusters
##   1   2
## 450 233
```

```
tpc2 <- table(biopsy.pr.hclust2.clusters, biopsy1$class)
tpc2
```

```
##
## biopsy.pr.hclust2.clusters benign malignant
##                          1    435        15
##                          2      9       224
```

```
sum(apply(tpc2, 1, min))
```

```
## [1] 24
```

24 observations were not clustered accurately using the hierarchical clustering of principal components

Accuracy complete method

```
torg<-table(biopsy1$class)
biop <- c("benign", "malignant")
actual <- factor(rep(biop, times = c(torg[1], torg[2])), levels = rev(biop))
predpc <- factor(
        c(
```

```
                rep(biop, times = c(tpc[1], tpc[2])),
                rep(biop, times = c(tpc[3], tpc[4]))),
        levels = rev(biop))
xtab.pc <- table(predpc, actual)
library(caret)
cmpc<-confusionMatrix(xtab.pc)
cmpc$overall['Accuracy']
```

```
##  Accuracy
## 0.8477306
```

mcquitty method

```
torg<-table(biopsy1$class)
biop <- c("benign", "malignant")
actual <- factor(rep(biop, times = c(torg[1], torg[2])), levels = rev(biop))
predpc2 <- factor(
        c(
                rep(biop, times = c(tpc2[1], tpc2[2])),
                rep(biop, times = c(tpc2[3], tpc2[4]))),
        levels = rev(biop))
xtab.pc2 <- table(predpc2, actual)
library(caret)
cmpc2<-confusionMatrix(xtab.pc2)
cmpc2$overall['Accuracy']
```

```
##  Accuracy
## 0.9648609
```

Comparision Between Methods

External Cluster Validation

Clustering results are evaluated based on some externally known result, such as externally provided class labels i.e., benign/malignant for this dataset.

```
cluster_models <- as.data.frame(list(
        'K Means' = round(cmkmeans$overall, 3),
        'H Clust complete' = round(cmhclust$overall, 3),
        'H Clust mcquitty' = round(cmhclust2$overall, 3),
        'Pr.Comp HClust.comp' = round(cmpc$overall, 3),
        'Pr.Comp HClust.mcquitty' = round(cmpc2$overall, 3)
        ))
# datatable(t(cluster_models))
cluster_models
```

```
##                K.Means H.Clust.complete H.Clust.mcquitty Pr.Comp.HClust.comp
## Accuracy         0.958            0.852            0.965                0.848
## Kappa            0.906            0.641            0.922                0.630
## AccuracyLower    0.940            0.823            0.948                0.819
## AccuracyUpper    0.971            0.878            0.977                0.874
## AccuracyNull     0.650            0.650            0.650                0.650
## AccuracyPValue   0.000            0.000            0.000                0.000
## McnemarPValue    0.137            0.000            0.307                0.000
##                Pr.Comp.HClust.mcquitty
## Accuracy                         0.965
```

```
## Kappa                          0.922
## AccuracyLower                  0.948
## AccuracyUpper                  0.977
## AccuracyNull                   0.650
## AccuracyPValue                 0.000
## McnemarPValue                  0.307
```

- Hieracrhical Clustering model based on *mcquitty* method was the most accurate followed by K-Means clustering for clustering benign and malignant breast tumor samples.

Internal cluster validation

The clustering result is evaluated based on the data clustered itself (internal information) without reference to external information. Internal validation measures reflect often the **compactness**, the **connectedness** and **separation** of the cluster partitions. Measures include: *Dunn Index* (= *min.separation/max.dia*), *Average Silhouette Width*, *Separation Index*

```
cshc1 <- cluster.stats(data.dist, biopsy.hclust.clusters)
cshc2<-cluster.stats(data.dist, biopsy.hclust2.clusters)
cskm <- cluster.stats(data.dist, biopsy.km$cluster)

DI<-as.data.frame(list(cshc1$dunn, cshc2$dunn, cskm$dunn))
IntVal <- data.frame("Dunn-Index" = c(cshc1$dunn, cshc2$dunn, cskm$dunn),
                "Silhouette-Width" = c(cshc1$avg.silwidth, cshc2$avg.silwidth, cskm$avg.silwidth),
                "Separation-Index" = c(cshc1$sindex, cshc2$sindex, cskm$sindex ))
row.names(IntVal) <- c("Hierarchical Clustering 'complete'", "Hierarchical Clustering 'mcquitty'", "K-M
# datatable(IntVal, style="default",  height = "auto", width = "auto")
IntVal
```

```
##                                  Dunn.Index Silhouette.Width Separation.Index
## Hierarchical Clustering 'complete'  0.1611185        0.5307819         1.739934
## Hierarchical Clustering 'mcquitty'  0.1506628        0.5704588         1.915596
## K-Means                             0.1171724        0.5732451         1.693467
```

Conclusion

- The different methods produce different cluster memberships.
- The algorithms make different assumptions about how the data is generated.
- We can choose to use one model over another based on the quality of the models' assumptions.
- **In this case, external validation reveals that hierarchical clustering based on *mcquitty* method provides the most accurate clustering of breast tumor samples. Inaccuracies can be addressed by including additional methods of assessment including clinical judgement, biomarker assays to confirm or rule out malignant tumors or vice versa.**

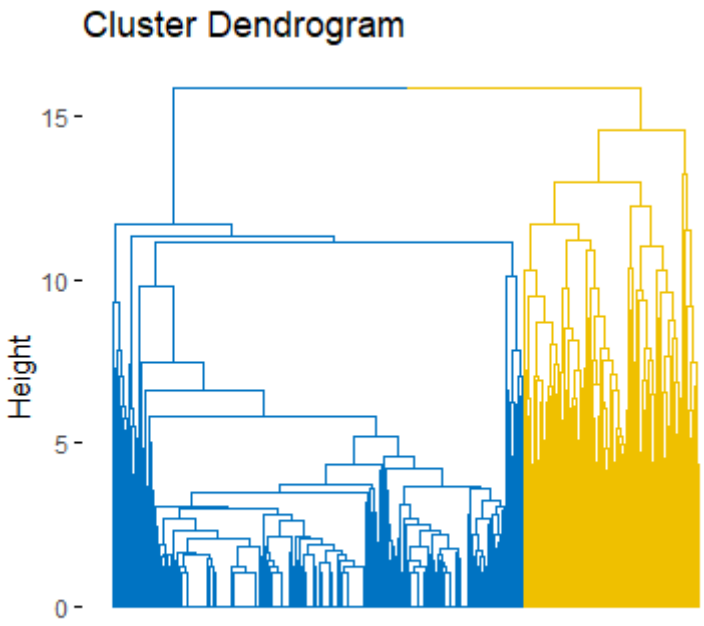


Figure 11: champion