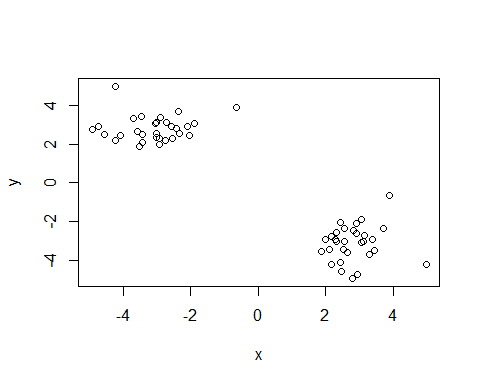
Class08

Zhuohang Wu

February 5, 2019

# Generate some example data for clustering  
tmp <- c(rnorm(30,-3), rnorm(30,3))  
x <- cbind(x=tmp, y=rev(tmp))  
plot(x)



y <- kmeans(x,centers = 5,nstart = 30)  
y

## K-means clustering with 5 clusters of sizes 18, 12, 8, 12, 10  
##   
## Cluster means:  
## x y  
## 1 2.813335 -2.517331  
## 2 -3.102231 2.414084  
## 3 -4.247947 3.065750  
## 4 2.804426 -3.995330  
## 5 -2.204557 3.079812  
##   
## Clustering vector:  
## [1] 5 2 5 3 3 2 2 5 5 2 5 2 5 2 5 3 3 2 3 2 5 2 3 2 3 2 5 3 5 2 1 1 4 1 4  
## [36] 4 1 4 4 1 1 4 1 4 4 1 1 1 1 1 1 1 1 1 4 4 4 1 4 1  
##   
## Within cluster sum of squares by cluster:  
## [1] 10.637142 2.770192 7.263624 10.823312 5.464910  
## (between\_SS / total\_SS = 96.7 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss"   
## [5] "tot.withinss" "betweenss" "size" "iter"   
## [9] "ifault"

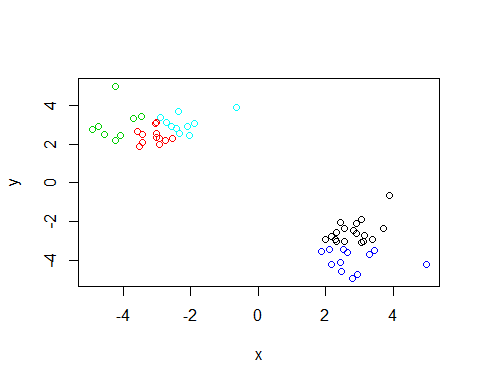
y$betweenss

## [1] 1088.208

?kmeans

## starting httpd help server ... done

plot(x,col=y$cluster)

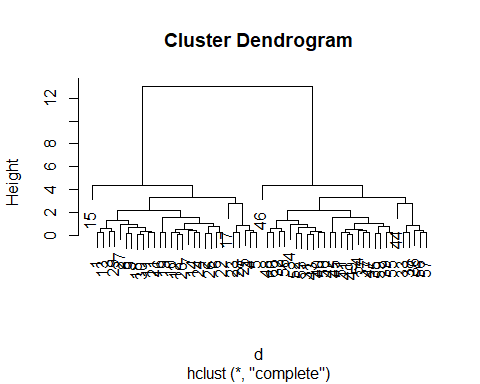


## Hierarchical clastering

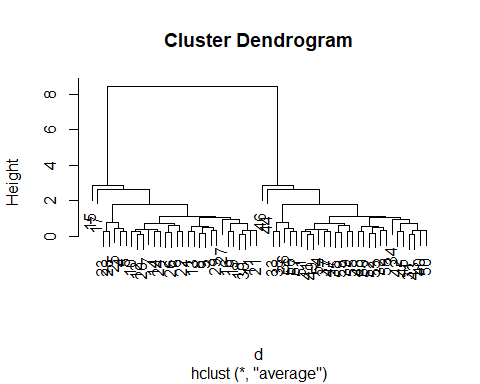
# First we need to calculate point (dis)similarity  
# as the Euclidean distance between observations  
dist\_matrix <- dist(x)  
# The hclust() function returns a hierarchical  
# clustering model  
hc <- hclust(d = dist\_matrix)  
# the print method is not so useful here  
hc

##   
## Call:  
## hclust(d = dist\_matrix)  
##   
## Cluster method : complete   
## Distance : euclidean   
## Number of objects: 60

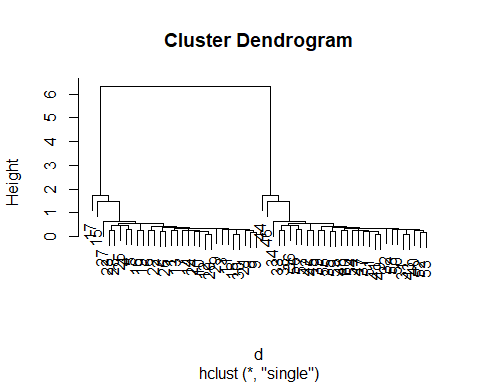
d <- dist\_matrix  
# Using different hierarchical clustering methods  
hc.complete <- hclust(d, method="complete")  
plot(hc.complete)



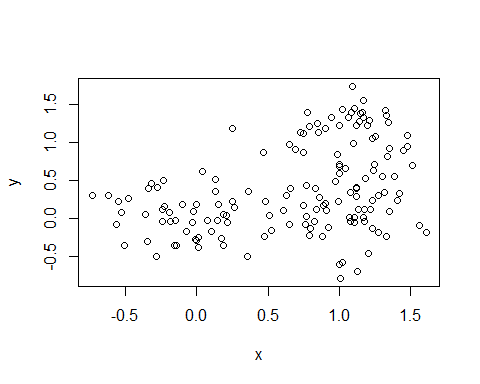
hc.average <- hclust(d, method="average")  
plot(hc.average)



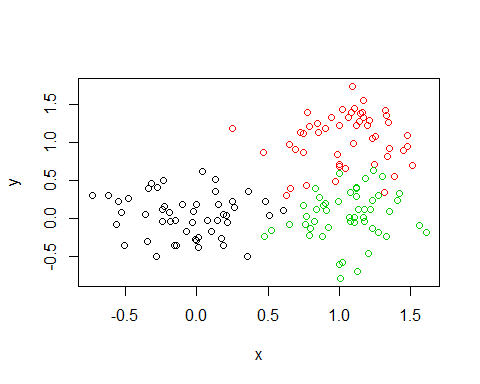
hc.single <- hclust(d, method="single")  
plot(hc.single)



# Step 1. Generate some example data for clustering  
x <- rbind(  
 matrix(rnorm(100, mean=0, sd = 0.3), ncol = 2), # c1  
 matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2), # c2  
 matrix(c(rnorm(50, mean = 1, sd = 0.3), # c3  
 rnorm(50, mean = 0, sd = 0.3)), ncol = 2))  
colnames(x) <- c("x", "y")  
# Step 2. Plot the data without clustering  
plot(x)



# Step 3. Generate colors for known clusters  
# (just so we can compare to hclust results)  
col <- as.factor( rep(c("c1","c2","c3"), each=50) )  
plot(x, col=col)

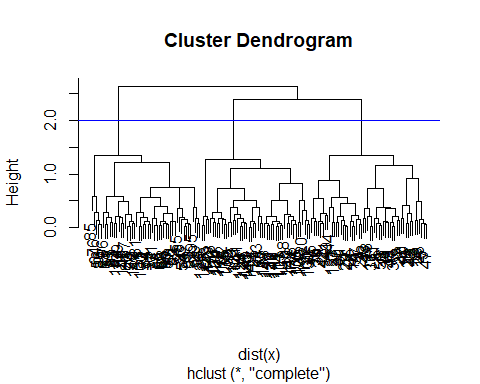


View(x)

hc <- hclust(dist(x))  
plot(hc)  
cutree(hc, h=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 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 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 3 3  
## [71] 2 2 3 3 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 3 3 3 3 3  
## [106] 3 3 3 2 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 1 3 3 3  
## [141] 3 3 3 2 2 3 3 3 3 3

abline(h=2,col="blue")



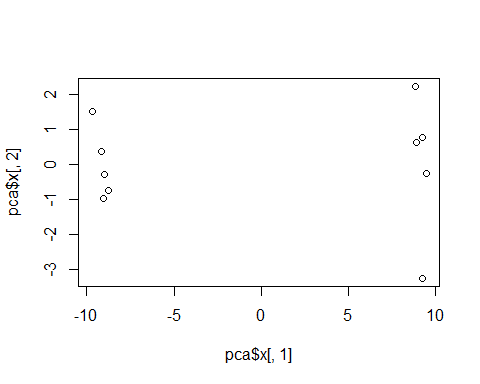
mydata <- read.csv("https://tinyurl.com/expression-CSV",  
 row.names=1)  
head(mydata)

## wt1 wt2 wt3 wt4 wt5 ko1 ko2 ko3 ko4 ko5  
## gene1 439 458 408 429 420 90 88 86 90 93  
## gene2 219 200 204 210 187 427 423 434 433 426  
## gene3 1006 989 1030 1017 973 252 237 238 226 210  
## gene4 783 792 829 856 760 849 856 835 885 894  
## gene5 181 249 204 244 225 277 305 272 270 279  
## gene6 460 502 491 491 493 612 594 577 618 638

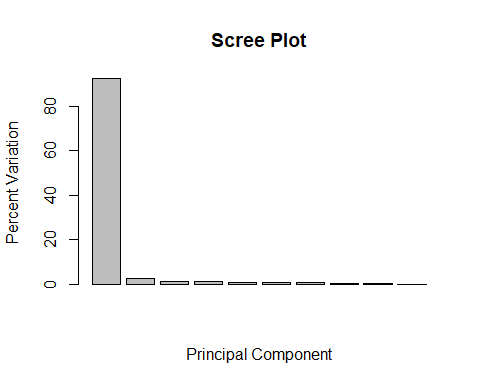
## lets do PCA  
pca <- prcomp(t(mydata), scale=TRUE)   
attributes(pca)

## $names  
## [1] "sdev" "rotation" "center" "scale" "x"   
##   
## $class  
## [1] "prcomp"

plot(pca$x[,1],pca$x[,2])



## Variance captured per PC  
pca.var <- pca$sdev^2   
pca.var.per <- round(pca.var/sum(pca.var)\*100, 1)  
barplot(pca.var.per, main="Scree Plot",  
 xlab="Principal Component", ylab="Percent Variation")



colvec <- colnames(mydata)  
colvec[grep("wt", colvec)] <- "red"  
colvec[grep("ko", colvec)] <- "blue"  
plot(pca$x[,1], pca$x[,2], col=colvec, pch=16,  
 xlab=paste0("PC1 (", pca.var.per[1], "%)"),  
 ylab=paste0("PC2 (", pca.var.per[2], "%)"))

