ML_PS4

Jiewen Luo

3/2/2020

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.2
```

```
## -- Attaching packages ---------------------------------------------------
-------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
## Warning: package 'tibble' was built under R version 3.6.2
```

```
## Warning: package 'tidyr' was built under R version 3.6.2
```

```
## Warning: package 'readr' was built under R version 3.6.2
```

```
## Warning: package 'purrr' was built under R version 3.6.2
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
## Warning: package 'stringr' was built under R version 3.6.2
```

```
## Warning: package 'forcats' was built under R version 3.6.2
```

```
## -- Conflicts ------------------------------------------------------------
-- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 3.6.2
```

```
library(dendextend)
```

```
## Warning: package 'dendextend' was built under R version 3.6.2
```

```
##
## --------------------
## Welcome to dendextend version 1.13.3
## Type citation('dendextend') for how to cite the package.
```

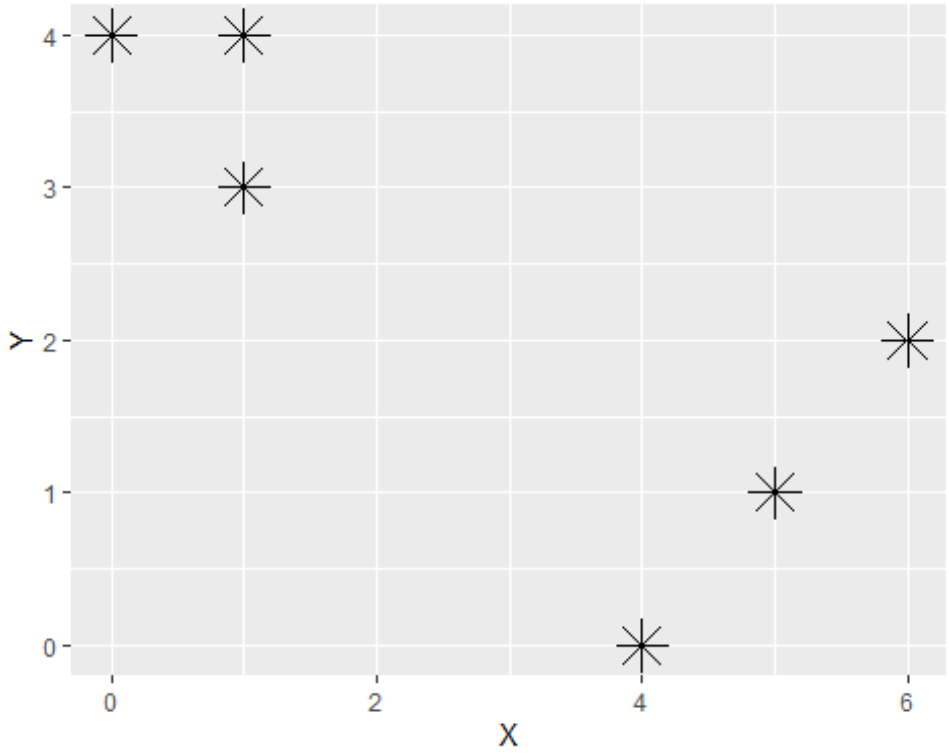
```
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at:
https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:
suppressPackageStartupMessages(library(dendextend))
## ----------------------

##
## Attaching package: 'dendextend'

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

library(ggplot2)
```

Performing K-Means By Hand

```
x<-cbind(X=c(1,1,0,5,6,4),Y=c(4,3,4,1,2,0))
x<-as_tibble(x)
```

##1. Plot the observations

```
ggplot(data=x, aes(x=X, y=Y))+
 geom_point(shape=8, size=6)
```

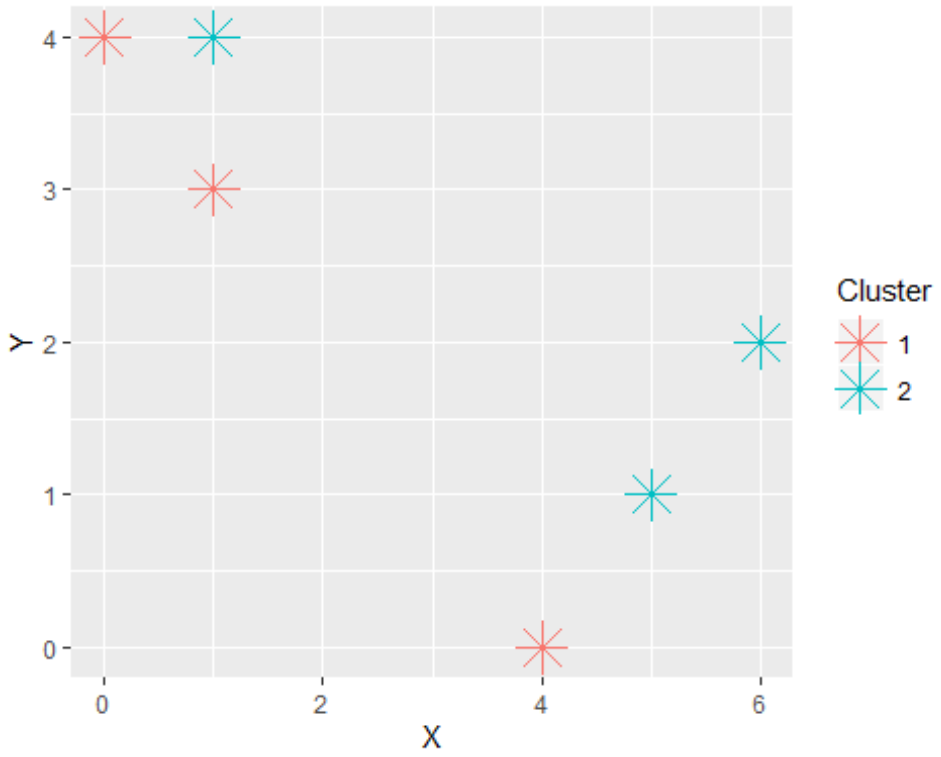



```
print(x)
```

```
## # A tibble: 6 x 2
##       X     Y
##   <dbl> <dbl>
## 1     1     4
## 2     1     3
## 3     0     4
## 4     5     1
## 5     6     2
## 6     4     0
```

##2.

```
set.seed(20)
x<-x%>%mutate(Cluster=sample(c(1,2), size = 6, replace = TRUE))
x$Cluster<-as.factor(x$Cluster)
ggplot(data=x, aes(x=X, y=Y, color=Cluster)) +
 geom_point(shape=8, size=6)
```



##3.

```
  centriod1<-x%>%filter(Cluster=="1")
  centriod2<-x%>%filter(Cluster=="2")
  x1<-sum(centriod1$X)/length(centriod1$X)
  x2<-sum(centriod2$X)/length(centriod2$X)
  y1<-sum(centriod1$Y)/length(centriod1$Y)
  y2<-sum(centriod2$Y)/length(centriod2$Y)
Centriod1<-as_tibble(cbind(mean_X=c(x1,x2), mean_Y=c(y1,y2),Cluster=c(1,2)))
Centriod1$Cluster<-as.factor(Centriod1$Cluster)
print(Centriod1)
```

```
## # A tibble: 2 x 3
##   mean_X mean_Y Cluster
##    <dbl>  <dbl> <fct>
## 1   1.67   2.33 1
## 2   4      2.33 2
```

```
print(x)
```

```
## # A tibble: 6 x 3
##       X     Y Cluster
##   <dbl> <dbl> <fct>
## 1     1     4 2
## 2     1     3 1
## 3     0     4 1
## 4     5     1 2
```

```
## 5     6     2 2
## 6     4     0 1
```

##4.

```
  for (i in 1:6){
    distance_1<-sqrt((x[i,1]-Centriod1[1,1])**2+(x[i,2]-Centriod1[1,2])**2)
    distance_2<-sqrt((x[i,1]-Centriod1[2,1])**2+(x[i,2]-Centriod1[2,2])**2)
    x$Cluster1[i]<-ifelse(distance_1 >=distance_2, "2", "1")
  }
```

```
## Warning: Unknown or uninitialised column: 'Cluster1'.
```

```
  print(x)
```

```
## # A tibble: 6 x 4
##       X     Y Cluster Cluster1
##   <dbl> <dbl> <fct>   <chr>
## 1     1     4 2       1
## 2     1     3 1       1
## 3     0     4 1       1
## 4     5     1 2       2
## 5     6     2 2       2
## 6     4     0 1       2
```

##5.

```
x$Cluster1<-as.factor(x$Cluster1)
  centriod1<-x%>%filter(Cluster1=="1")
  centriod2<-x%>%filter(Cluster1=="2")
  x1<-sum(centriod1$X)/length(centriod1$X)
  x2<-sum(centriod2$X)/length(centriod2$X)
  y1<-sum(centriod1$Y)/length(centriod1$Y)
  y2<-sum(centriod2$Y)/length(centriod2$Y)
Centriod2<-as_tibble(cbind(mean_X=c(x1,x2), mean_Y=c(y1,y2),Cluster=c(1,2)))
Centriod2$Cluster<-as.factor(Centriod2$Cluster)
print(Centriod2)
```

```
## # A tibble: 2 x 3
##   mean_X mean_Y Cluster
##    <dbl>  <dbl> <fct>
## 1  0.667   3.67 1
## 2  5       1    2
```

```
  for (i in 1:6){
    distance_1<-sqrt((x[i,1]-Centriod2[1,1])**2+(x[i,2]-Centriod2[1,2])**2)
    distance_2<-sqrt((x[i,1]-Centriod2[2,1])**2+(x[i,2]-Centriod2[2,2])**2)

    x$Cluster2[i]<-ifelse(distance_1 >=distance_2, "2", "1")
    x[i,6]<-ifelse(distance_1 >=distance_2, "2", "1")
  }
```

```
## Warning: Unknown or uninitialised column: 'Cluster2'.
```

```
print(x)
```

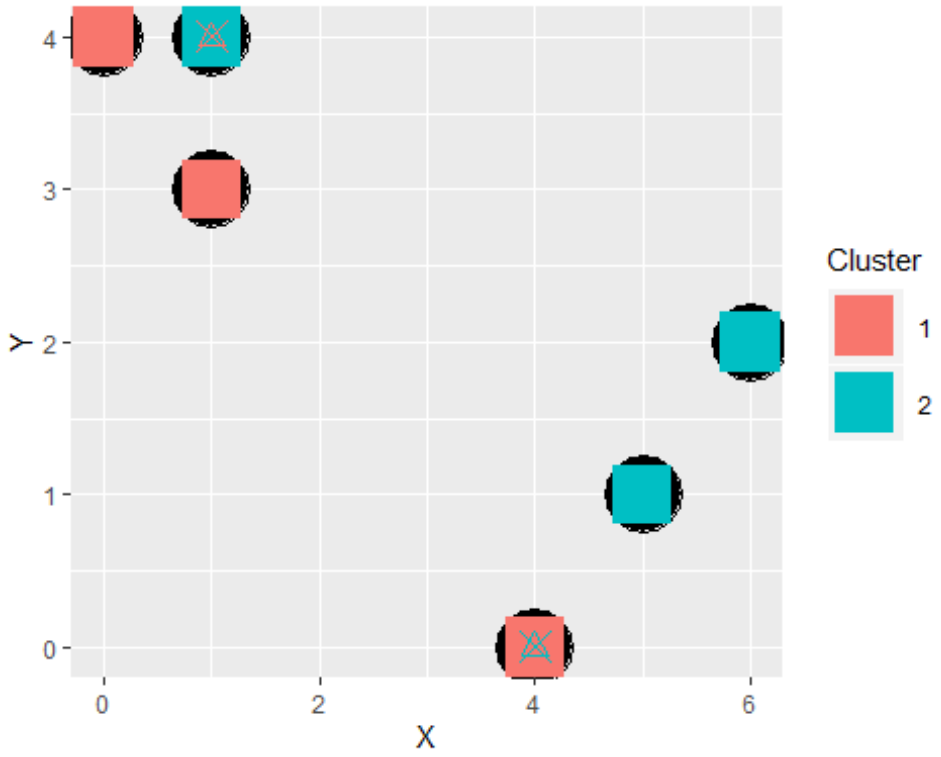
```
## # A tibble: 6 x 6
##       X     Y Cluster Cluster1 Cluster2 V6[,1]
##   <dbl> <dbl> <fct>   <fct>    <chr>    <chr>
## 1     1     4 2       1        1        1
## 2     1     3 1       1        1        1
## 3     0     4 1       1        1        1
## 4     5     1 2       2        2        2
## 5     6     2 2       2        2        2
## 6     4     0 1       2        2        2
```

##6

```
x$Cluster2<-as.factor(x$Cluster2)
print(x)
```

```
## # A tibble: 6 x 6
##       X     Y Cluster Cluster1 Cluster2 V6[,1]
##   <dbl> <dbl> <fct>   <fct>    <fct>    <chr>
## 1     1     4 2       1        1        1
## 2     1     3 1       1        1        1
## 3     0     4 1       1        1        1
## 4     5     1 2       2        2        2
## 5     6     2 2       2        2        2
## 6     4     0 1       2        2        2
```

```
ggplot(data=x, aes(x=X, y=Y)) +
  geom_point(shape=20, size=20)+
    geom_point(aes(color=Cluster),size=10, shape=15 )+
  geom_point(aes(color=Cluster1),size=5, shape=4 )+
    geom_point(aes(color=Cluster2),size=3, shape=2 )
```



#Clustering State Legislative Professionalism ##1.

```
library(miceadds)

## Warning: package 'miceadds' was built under R version 3.6.2

## Loading required package: mice

## Warning: package 'mice' was built under R version 3.6.2

##
## Attaching package: 'mice'

## The following objects are masked from 'package:base':
##
##     cbind, rbind

## * miceadds 3.8-9 (2020-02-17 11:03:15)

setwd("C:/Users/amber/OneDrive/Documents/Uchicago_MAPSS/Winter 2020/Machine
Learning/Data")
dat<- miceadds::load.Rdata2( filename="legprof-components.v1.0.Rdata")

tail(dat)

##     fips stateabv   state  sessid t_slength slength salary_real    expend
year
## 945   56       WY Wyoming 1999/00        58      58    9.372179  98.84684
2000
```

```
## 946   56       WY Wyoming  2001/2        60      37    5.694574 138.57949
2002
## 947   56       WY Wyoming  2003/4        64      58    8.515029 127.29152
2004
## 948   56       WY Wyoming  2005/6        58      58    8.003167 141.31708
2006
## 949   56       WY Wyoming  2007/8        67      67   10.452655 133.17257
2008
## 950   56       WY Wyoming 2009/10        40      40    6.049208 147.18433
2010
##          mds1      mds2
## 945 -1.549295 0.2637456
## 946 -1.552887 0.2822173
## 947 -1.498865 0.2396363
## 948 -1.533220 0.3037646
## 949 -1.449164 0.2214548
## 950 -1.670888 0.4607199
```

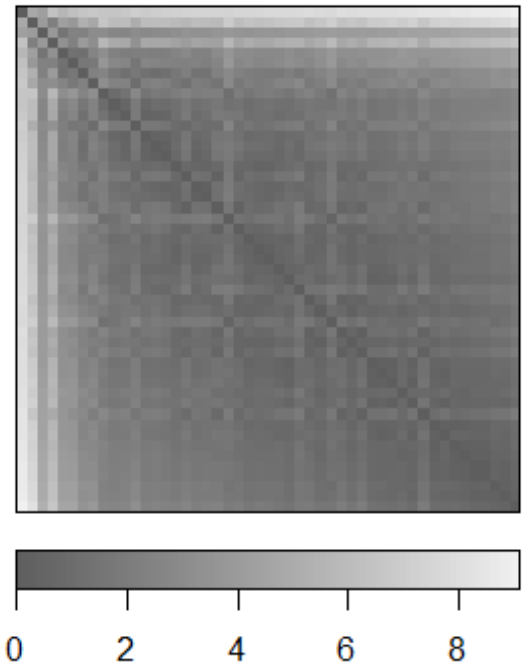
##2.

```
dat<-as.data.frame(dat)
da<-dat%>%filter(sessid=="2009/10")%>%select(state, t_slength, slength,
salary_real,expend)%>%drop_na()
state<-da%>%select(state)
rownames(da) <- da[,1]
da<-da%>%select(-state)%>%scale()


dim(da)

## [1] 49  4

head(da)

##             t_slength    slength salary_real     expend
## Alabama    -0.3716599 -0.4594723  -1.0920009 -0.2399910
## Alaska     -0.2294089 -0.1452309   0.4011333  0.8591198
## Arizona     1.6453067  0.7951955  -0.1335656 -0.1299408
## Arkansas   -0.8036462 -0.7881756  -0.4923902 -0.2612061
## California  2.8807257  1.7767099   3.2069914  5.4785453
## Colorado    0.6827338  0.9008887   0.1113595 -0.3485530
```

##3.

```
library(seriation)

## Warning: package 'seriation' was built under R version 3.6.2

## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
```

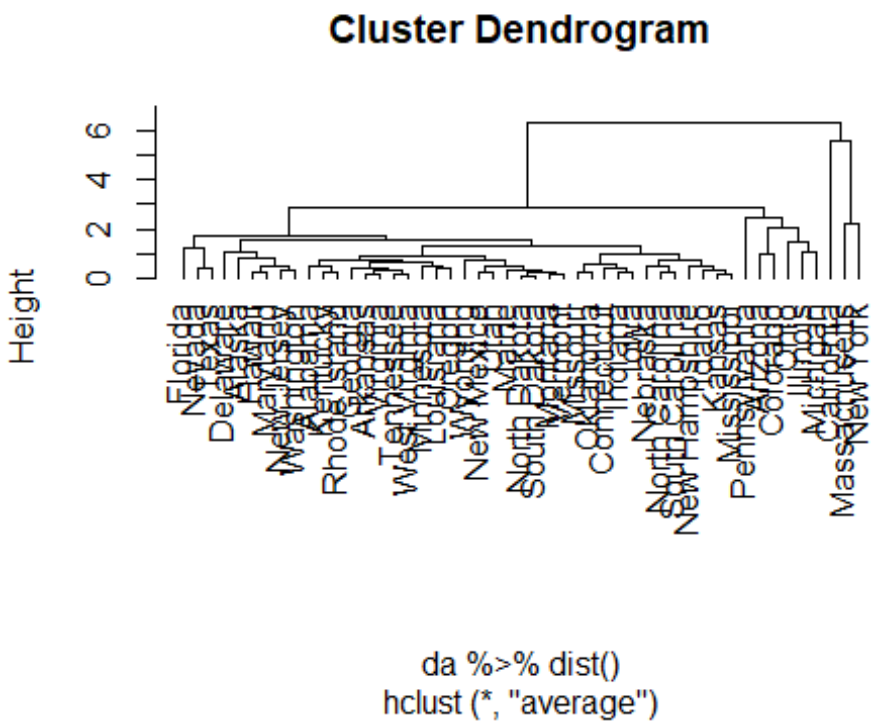
```
library(factoextra)

## Warning: package 'factoextra' was built under R version 3.6.2

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

dissplot(da%>%dist(), labels = NULL, method = "Spectral")
```



In the dissimilarity plot, compact clusters are visible as dark squares (low dissimilarity) on the diagonal of the plot. We can see a negative diagonositc dark line in the graph, which indicate high clusterability of feature space.We can also see from that graph that there are likely two clusters. One is smaller, as shown on the samll black square on the upper-left corner. The other one is much bigger, as shown on the big black square on the bottom-right corner.

##. 4

```
hc_average <- hclust(da%>%dist(),
                     method = "average"); plot(hc_average, hang = -1)
```



The amount of branches decrease as we increase the height. We can see from the figure that the best choices for total number of clusters are either 2 or 3. There are fair amount of similarly when we dividing the samples into 2 or 3 clustering.If we divide the samples into 2 clusters, the there is a small group with two members and all the others are belong to the other group.

##5.

```
set.seed(666)
kmeans <- kmeans(da%>%dist(),
                 centers = 2,
                 nstart = 15)
str(kmeans)
```

```
## List of 9
##  $ cluster     : Named int [1:49] 2 2 2 2 1 2 2 2 2 2 ...
##   ..- attr(*, "names")= chr [1:49] "Alabama" "Alaska" "Arizona" "Arkansas"
...
##  $ centers     : num [1:2, 1:49] 6.26 1.32 4.99 1.62 4.39 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:49] "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ totss       : num 4855
##  $ withinss    : num [1:2] 470 1150
##  $ tot.withinss: num 1620
##  $ betweenss   : num 3235
##  $ size        : int [1:2] 4 45
```

```
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"

# Assess a little more descriptively
t2 <- as.table(kmeans$cluster)
t2 <- data.frame(t2)
dim(t2)

## [1] 49  2

t2<-cbind(state, t2)
rownames(t2) <- state[,1]
colnames(t2)[colnames(t2)=="Freq"] <- "kmean_cluster"
t2$Var1 <- NULL
table(t2$kmean_cluster)

##
##  1  2
##  4 45
```

From the summary table we can see that 4 states are assigned to cluster 1 and 46 states are assigned to cluster 2

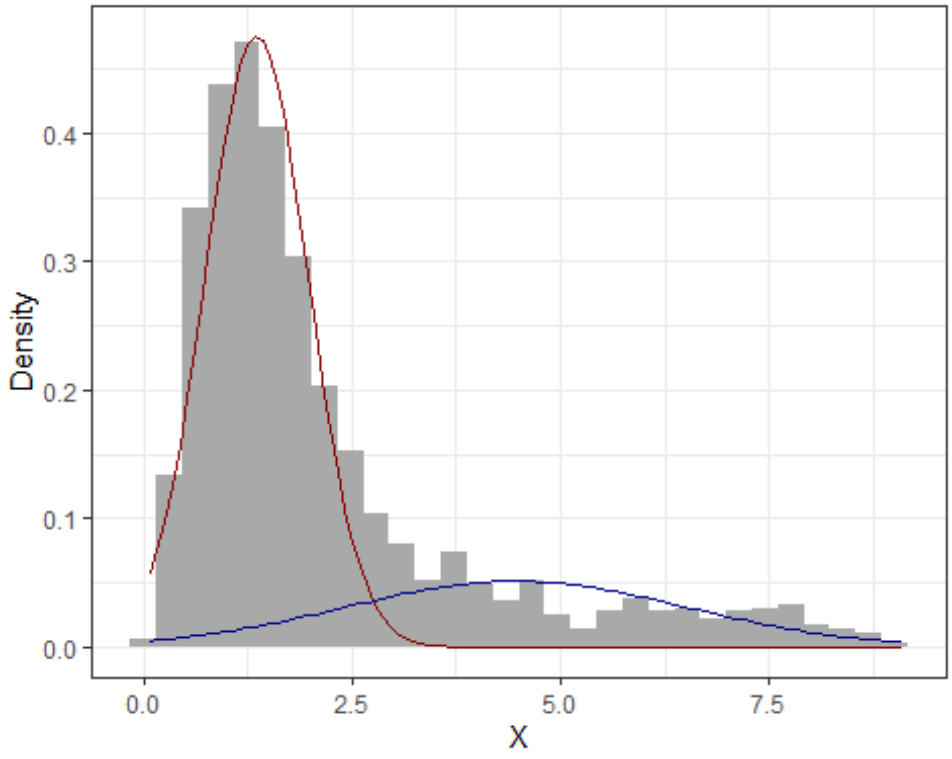
##6.

```
library(mixtools)

## Warning: package 'mixtools' was built under R version 3.6.2

## mixtools package, version 1.2.0, Released 2020-02-05
## This package is based upon work supported by the National Science
Foundation under Grant No. SES-0518772.

library(plotGMM)

## Warning: package 'plotGMM' was built under R version 3.6.2

set.seed(735)
gmm1 <- normalmixEM(da%>%dist(), k = 2)

## number of iterations= 27

str(gmm1)

## List of 9
##  $ x         : num [1:1176] 1.886 2.564 0.809 8.171 2.103 ...
##  $ lambda    : num [1:2] 0.736 0.264
##  $ mu        : num [1:2] 1.35 4.47
##  $ sigma     : num [1:2] 0.618 2.045
##  $ loglik    : num -1922
##  $ posterior : num [1:1176, 1:2] 9.34e-01 6.76e-01 9.69e-01 1.60e-25
8.96e-01 ...
```

```
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "comp.1" "comp.2"
##  $ all.loglik: num [1:28] -2721 -2324 -2149 -1991 -1936 ...
##  $ restarts  : num 0
##  $ ft        : chr "normalmixEM"
##  - attr(*, "class")= chr "mixEM"
```

```
ggplot(data.frame(x = gmm1$x)) +
  geom_histogram(aes(x, ..density..), fill = "darkgray") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[1], gmm1$sigma[1], lam = gmm1$lambda[1]),
                colour = "darkred") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[2], gmm1$sigma[2], lam = gmm1$lambda[2]),
                colour = "darkblue") +
  xlab("X") +
  ylab("Density") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



As we can see from the above histogram, the sample is divided into 2 clusters, one group has high density and the other group has very low density.

##7

##8

```r
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 3.6.2
```

```
## Package 'mclust' version 5.4.5
## Type 'citation("mclust")' for citing this R package in publications.
```

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

```
## The following object is masked from 'package:mixtools':
##
##     dmvnorm
```

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

```r
library(clValid)
```

```
## Warning: package 'clValid' was built under R version 3.6.2
```

```
## Loading required package: cluster
```

```r
set.seed(209)
dim(da)
```

```
## [1] 49  4
```

```r
cl_validation <- clValid(da, nClust = 2:5, validation = "internal", clMethods
= c("hierarchical", "kmeans", "model"))
summary(cl_validation )
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2 3 4 5
##
## Validation Measures:
##                                  2       3       4       5
##
## hierarchical Connectivity   6.0869  6.9536 16.1885 18.6774
##              Dunn           0.3637  0.4371  0.2562  0.2836
##              Silhouette     0.6994  0.6711  0.4932  0.4440
## kmeans       Connectivity   8.4460 10.8960 16.1885 28.7437
##              Dunn           0.1735  0.2581  0.2562  0.1090
##              Silhouette     0.6458  0.6131  0.4932  0.3042
## model        Connectivity  10.7393 28.6119 39.0687 67.8401
##              Dunn           0.1522  0.0633  0.0225  0.0258
##              Silhouette     0.6314  0.2588  0.1861  0.0085
```

```
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 6.0869 hierarchical 2
## Dunn         0.4371 hierarchical 3
## Silhouette   0.6994 hierarchical 2
```

##9 The fits tell us that HAC has the highest Dunn and Silhouette width but also the lowest connectivity; gmm has the highest connectivity but it also has the lowest Dunn and Silhouette width. kmeans has middle Connectivity Dunn, and Silhouette. The result from the validation process show that HAC has better performance compared with the other two algorithms. The optimal value of k is 2 in terms of Connectivity and Silhouette width, but 3 in terms of Dunn. A “sub-optimal” method could be selected if it does not vary much from the optimal choice. Sometimes a “sub-optimal” is preferable in terms of interpretation and other characteristics. For example, the result from k-means method’s mechanism is very easy to interpret and understand, therefore, one might choose this method even if its performance is not an statistically optimal choice.