Problem Set 4 - Runhua Li

Runhua Li

3/2/2020

1.

1.0 Set Up

```
rm(list = ls())
x <- cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
X <- data.frame(x)
```

1.1 Ploting

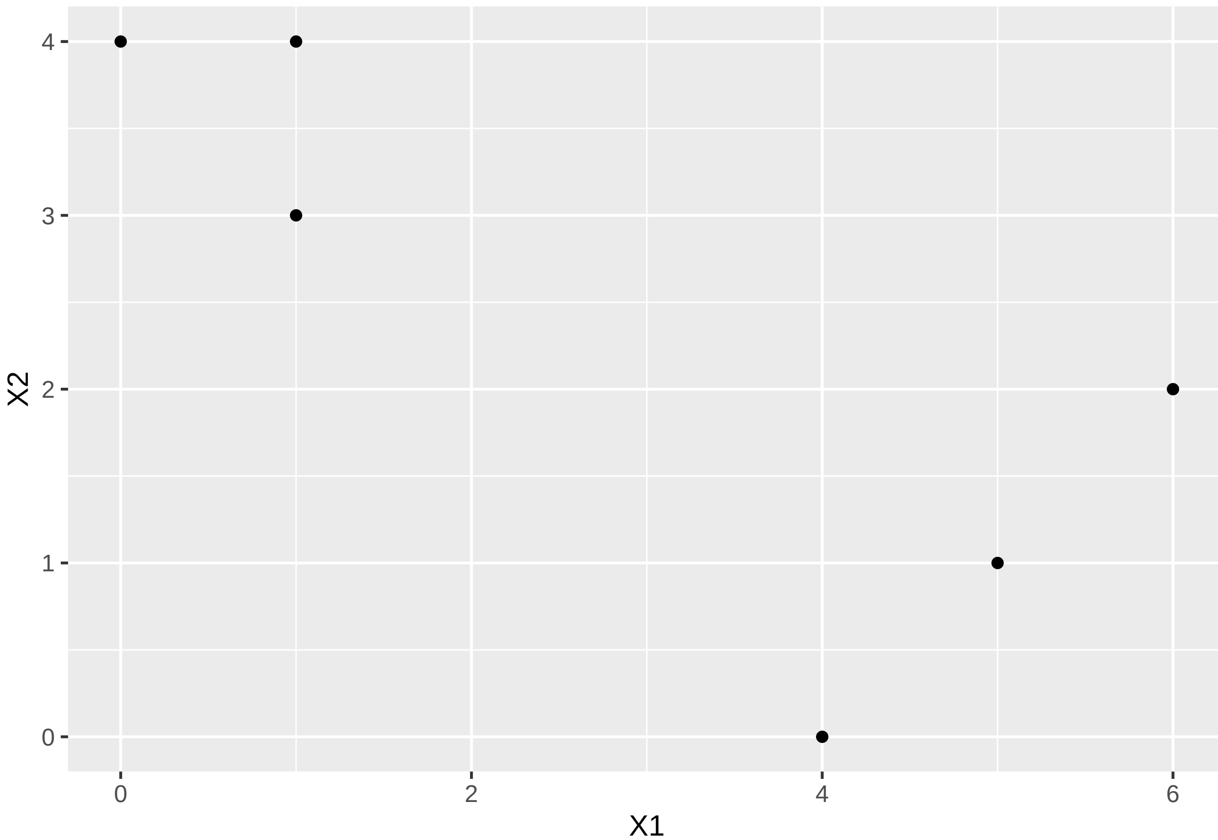
```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
ggplot(X, aes(x = X1, y = X2)) +
  geom_point()
```

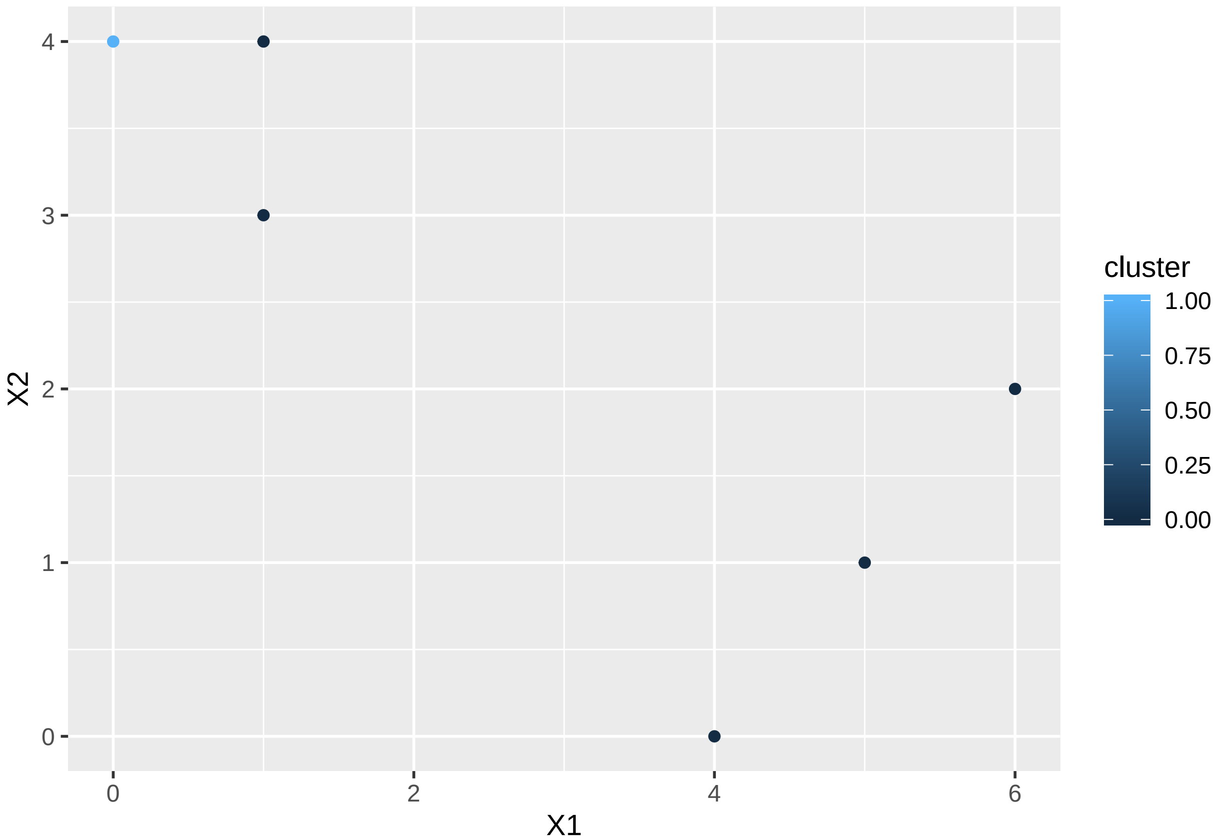


1.2 Assigning Cluster Label

```
set.seed(3751)
X <- X %>%
  mutate(cluster = sample(c(0, 1), 6, replace = T))

(X$cluster == 1)
```

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

```
ggplot(X, aes(x = X1, y = X2, color = cluster)) +
  geom_point()
```



Observation no. 3 is clustered to group 1, while the others are clustered to group 0.

1.3 Computing Centroids

```
C0 <- subset(X, cluster == 0)
C1 <- subset(X, cluster == 1)

X <- X %>%
  mutate(c01 = mean(C0$X1),
         c02 = mean(C0$X2),
         c11 = mean(C1$X1),
         c12 = mean(C1$X2))
```

Centroids are computed and muted into the data frame.

1.4 Reassigning Observations

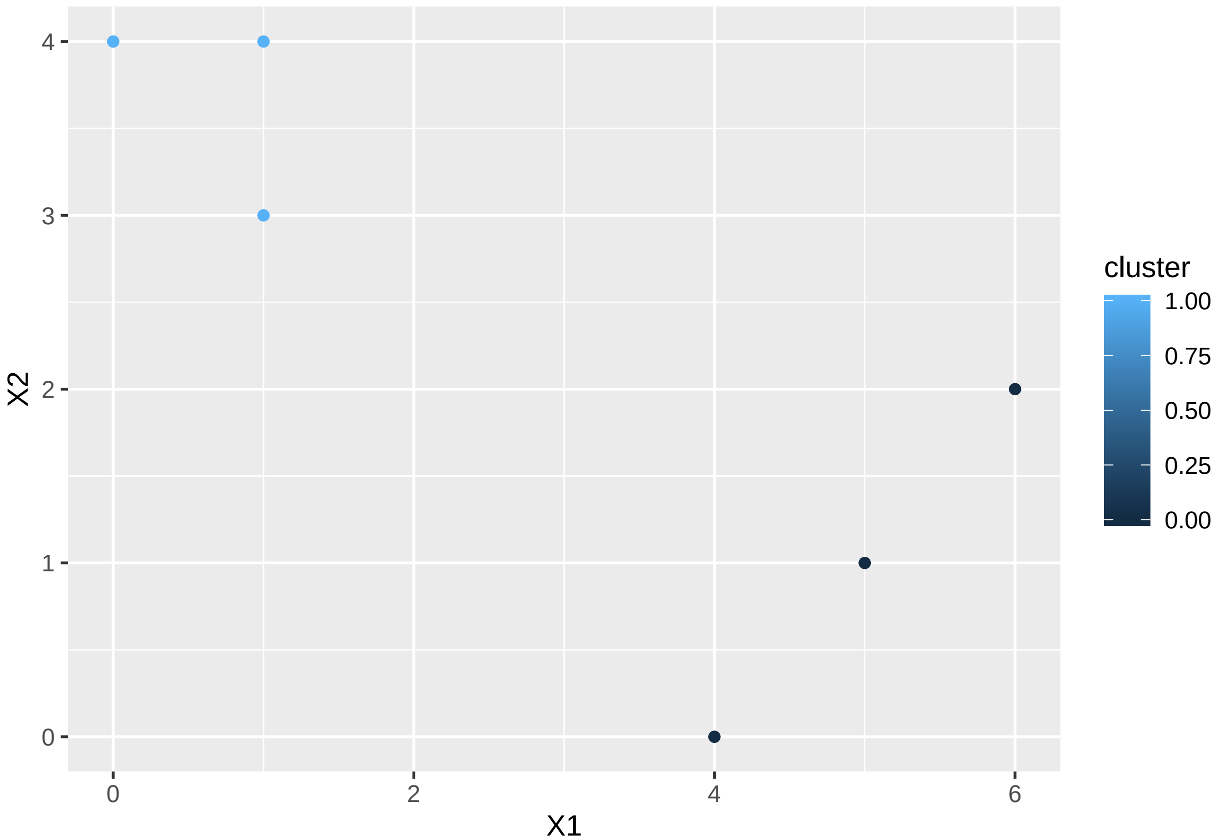
```
X <- X %>%
  mutate(cluster = 1 * ((X1 - c01)^2 +  (X2 - c02)^2
                        > (X1 - c11)^2 + (X2 - c12)^2))

(X$cluster == 1)
```

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

Now the first 3 observations are clustered to group 1, and the others group 0.

1.5 Repeating

```
check <- 9
repeat{
  C0 <- subset(X, cluster == 0)
  C1 <- subset(X, cluster == 1)
  X <- X %>%
    mutate(c01 = mean(C0$X1),
           c02 = mean(C0$X2),
           c11 = mean(C1$X1),
           c12 = mean(C1$X2))
  X <- X %>%
    mutate(cluster = 1 * ((X1 - c01)^2 +  (X2 - c02)^2
                          > (X1 - c11)^2 + (X2 - c12)^2))
  check = c(X$cluster, check)

  if(check[1] == check[7] &
     check[2] == check[8] &
     check[3] == check[9] &
     check[4] == check[10] &
     check[5] == check[11] &
     check[6] == check[12]){
    break
  }
}

ggplot(X, aes(x = X1, y = X2, color = cluster)) +
  geom_point()
```



2

2.0 Loading Data

```
rm(list = ls())
load("/Users/RunhuaLi/R/legprof-components.v1.0.RData")
```

2.1

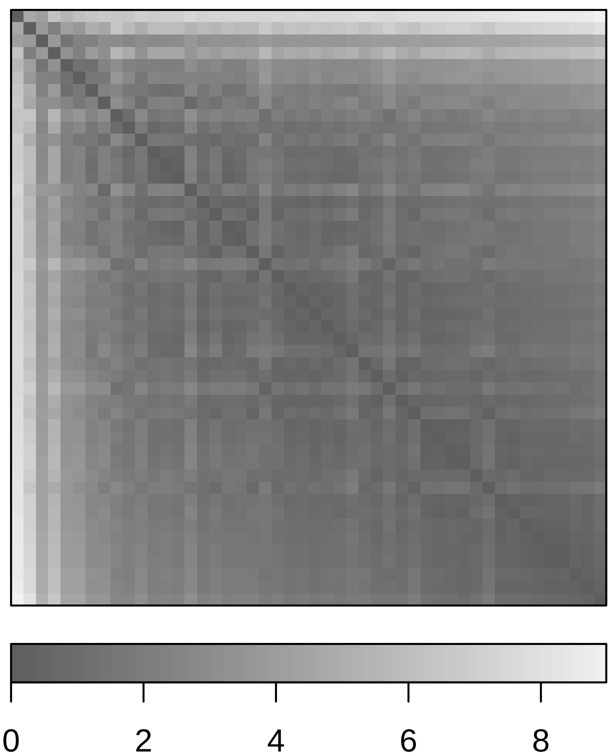
```
legprof <- na.omit(x) #deleting NAs

legprof <- legprof[(legprof$year == 2009 |
                      legprof$year == 2010), ] #keeping time-relavent obs

states <- legprof$stateabv #storing state names

legprof <- legprof[,c(2, 5:8)] #keeping relevant variables

legprof_sd <- scale(legprof[, -1]) #standardizing


legprof_dis <- legprof %>%
  select(`t_slength`, `slength`, `salary_real`, `expend`) %>%
  scale() %>%
  dist()
```

2.3

```r
library(seriation)
```

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

```r
dissplot(legprof_dis)
```



The observations seem clusterable into several groups. For example, 7 groups or 4 groups.

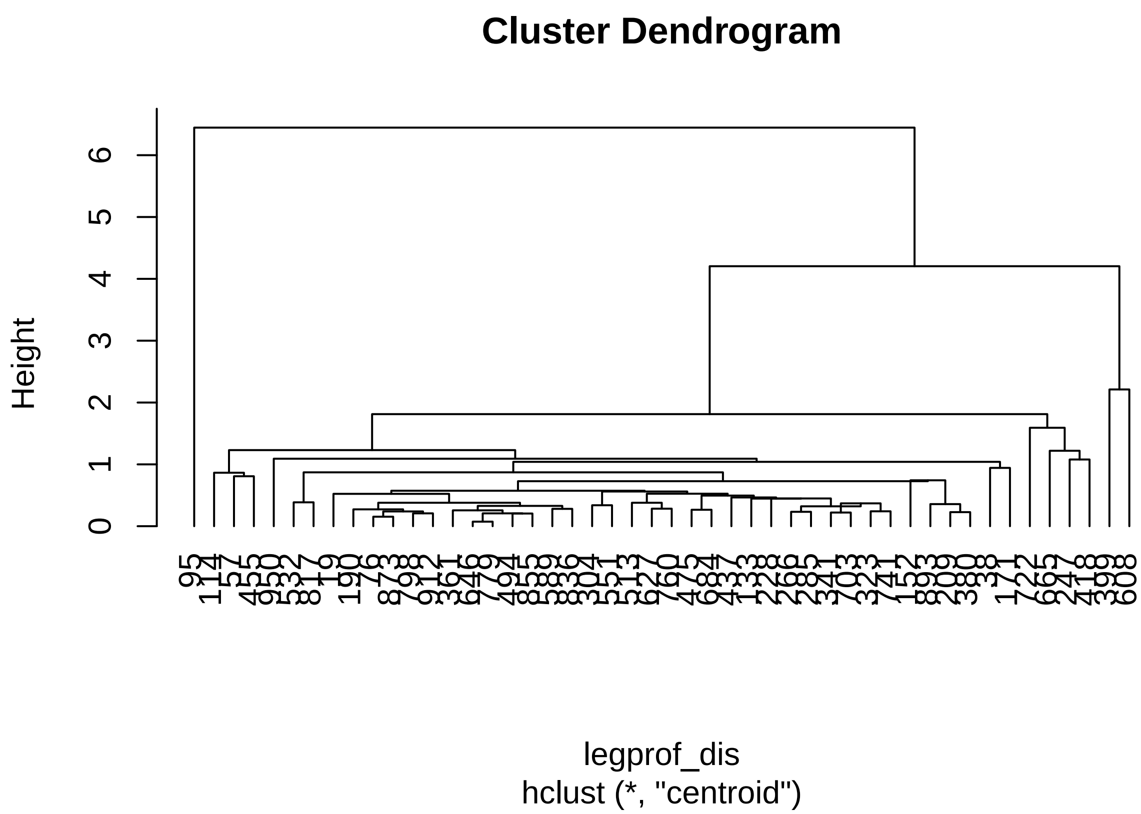
2.4 HAC

```r
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.13.4
## 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
```

```r
HAC <- hclust(legprof_dis,
              method = "centroid"); plot(HAC, hang = -1)
```




```r
cuts <- cutree(HAC,
               k = c(3, 5, 7))

cuts
```

```
##     3 5 7
## 19  1 1 1
## 38  1 1 1
## 57  1 1 2
## 76  1 1 1
## 95  2 2 3
## 114 1 1 2
## 133 1 1 1
## 152 1 1 1
## 171 1 1 1
## 190 1 1 1
## 209 1 1 1
## 228 1 1 1
## 247 1 3 4
## 266 1 1 1
## 285 1 1 1
```

```
## 304 1 1 1
## 323 1 1 1
## 341 1 1 1
## 361 1 1 1
## 380 1 1 1
## 399 3 4 5
## 418 1 3 4
## 437 1 1 1
## 455 1 1 2
## 475 1 1 1
## 494 1 1 1
## 513 1 1 1
## 532 1 1 1
## 551 1 1 1
## 589 1 1 1
## 608 3 5 6
## 627 1 1 1
## 646 1 1 1
## 665 1 3 4
## 684 1 1 1
## 703 1 1 1
## 722 1 3 7
## 741 1 1 1
## 760 1 1 1
## 779 1 1 1
## 798 1 1 1
## 817 1 1 1
## 836 1 1 1
## 855 1 1 1
## 873 1 1 1
## 893 1 1 1
## 912 1 1 1
## 950 1 1 1
```

As shown in the dendrogram, it seems the observations can be easily clustered into 2, 3 or 4 groups. In either case, the majority of observations are in one group.

2.5 K-means

```
library(skimr)
set.seed(3751)

kmeans <- kmeans(legprof_sd,
                 centers = 2,
                 nstart = 10)

kmeans$cluster
```

```
##  19  38  57  76  95 114 133 152 171 190 209 228 247 266 285 304 323 341
##   2   2   2   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2
## 361 380 399 418 437 455 475 494 513 532 551 589 608 627 646 665 684 703
##   2   2   1   1   2   2   2   2   2   2   2   2   1   2   2   1   2   2
## 722 741 760 779 798 817 836 855 873 893 912 950
##   1   2   2   2   2   2   2   2   2   2   2   2
```

```
kmeans$centers
```

```
##    t_slength    slength salary_real     expend
## 1  2.0079549  2.0643454     2.04323  1.4647791
## 2 -0.2868507 -0.2949065    -0.29189 -0.2092542
```

```
kmeans$size
```

```
## [1]  6 42
```

Several observations are clustered into group 1, while the majority are in group 2. Group 1 includes observations with longer session, higher salary and higher expenditure.

2.6

```
library(mixtools)
```

```
## 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-0518
```

```
library(plotGMM)
set.seed(3751)

GMM <- mvnormalmixEM(legprof_sd, k = 2)
```

```
## number of iterations= 18
```

```
GMM$lambda
```

```
## [1] 0.1825163 0.8174837
```

```
GMM$mu
```

```
## [[1]]
## [1] 0.5011774 0.1960824 0.3665275 1.2027275
##
## [[2]]
## [1] -0.11189588 -0.04377854 -0.08183313 -0.26852817
```

```
GMM$sigma
```

```
## [[1]]
##           [,1]      [,2]      [,3]      [,4]
## [1,] 1.3733894 0.9328596 1.0735800 1.0118430
## [2,] 0.9328596 0.6658422 0.8586936 0.7775733
## [3,] 1.0735800 0.8586936 1.7952327 1.9922008
## [4,] 1.0118430 0.7775733 1.9922008 2.9025545
##
## [[2]]
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.8225497 0.9205741 0.5429434 0.2336199
## [2,] 0.9205741 1.0386206 0.6083028 0.2437542
## [3,] 0.5429434 0.6083028 0.7602761 0.2303708
## [4,] 0.2336199 0.2437542 0.2303708 0.1546659
```

Two 4-variable joint normal distributions are mixed together with weights of roughly 1:4. The first distribuiton has higher mean and higher covariance.

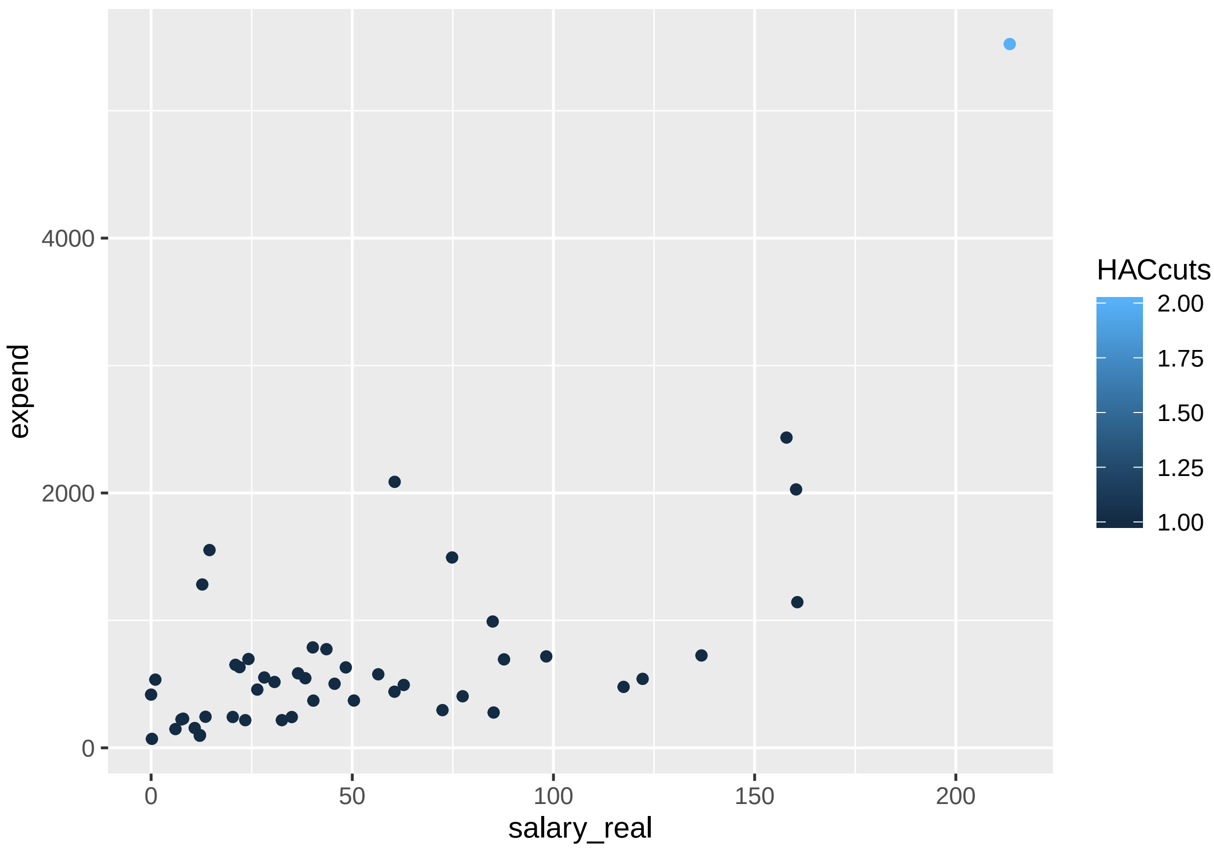
```
GMMclust <- 1 + 1 * (GMM$posterior[, 1] > 0.5)

GMMclust
```

```
##  [1] 1 2 2 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1
## [36] 1 2 1 1 1 1 2 1 1 1 1 1 1
```

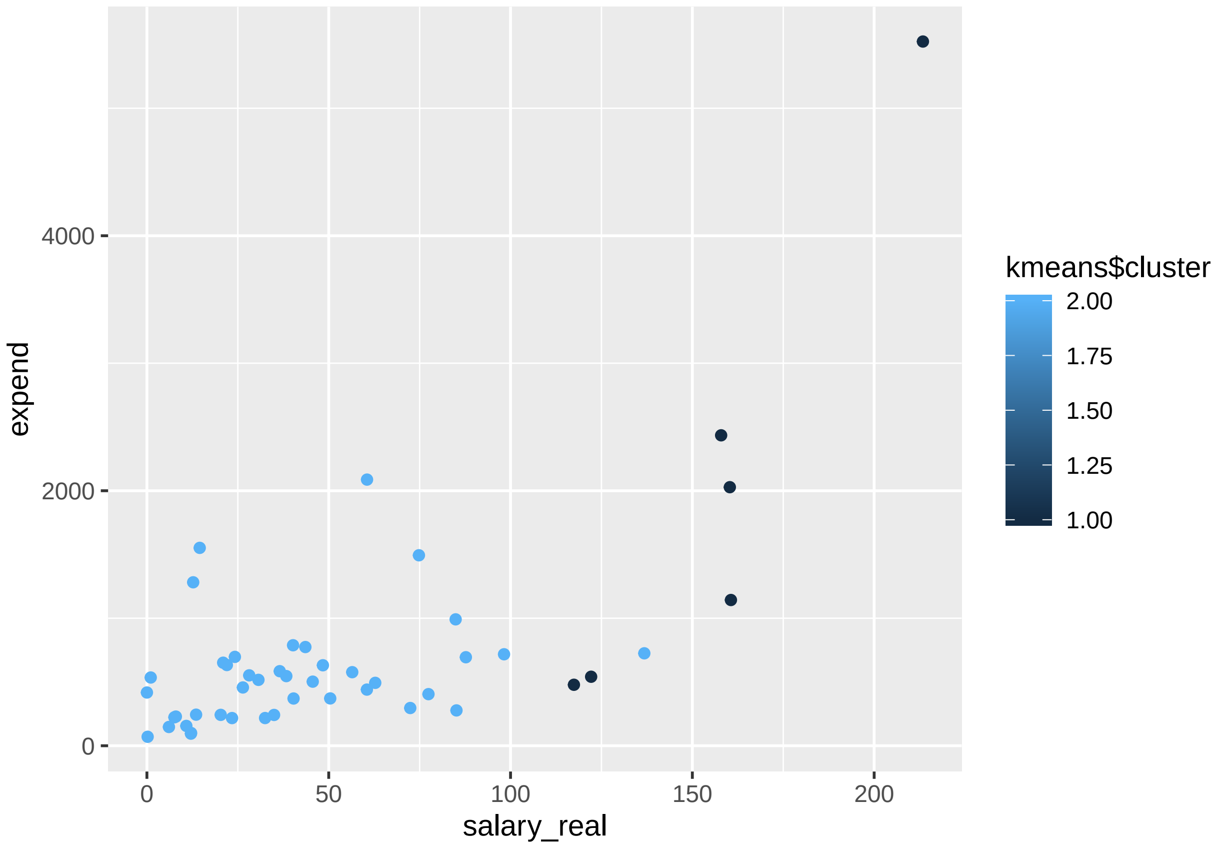
Most of the observations are clustered into group 2, while 8 observations are clustered into group 2.

2.7

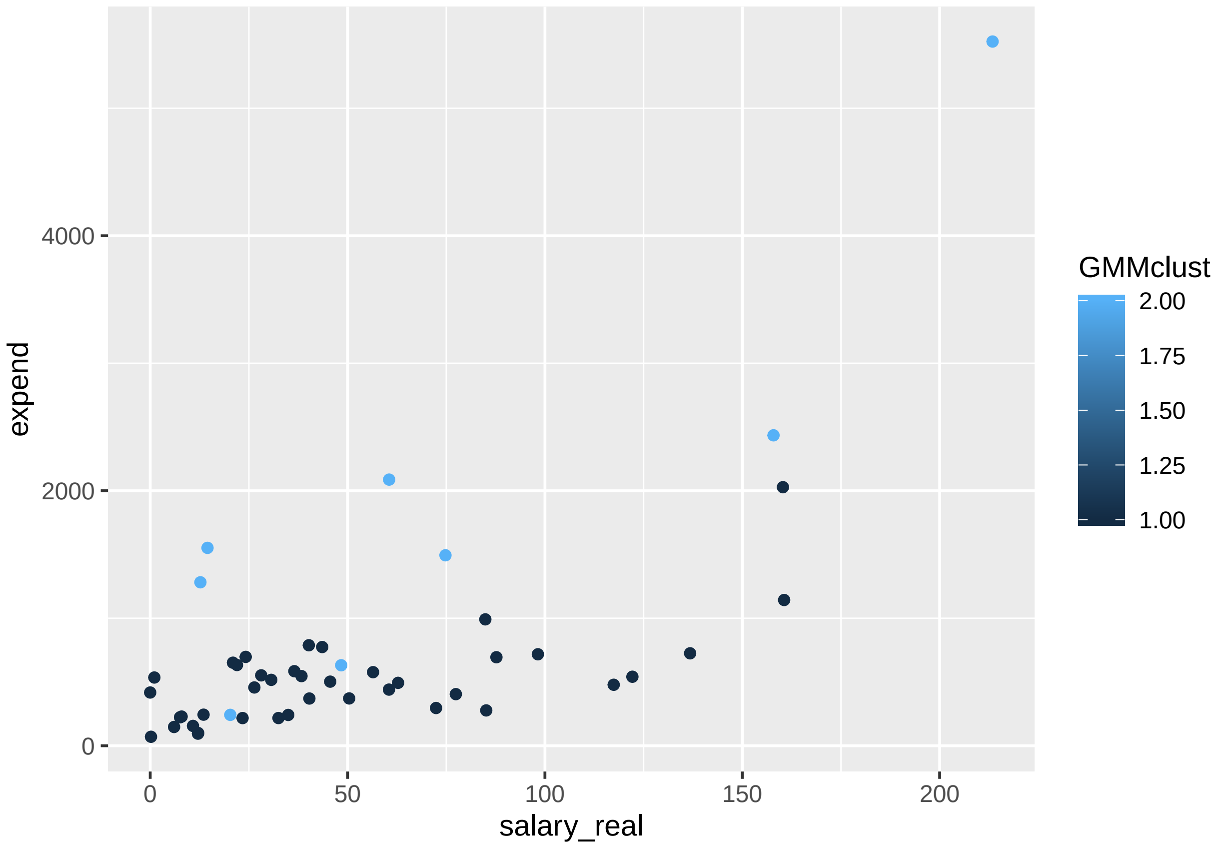
```
HACcuts <- cutree(HAC,
                  k = 2)

ggplot(legprof, aes(x = salary_real, y = expend, color = HACcuts)) +
  geom_point()
```




```
ggplot(legprof, aes(x = salary_real, y = expend, color = kmeans$cluster)) +
  geom_point()
```




```
ggplot(legprof, aes(x = salary_real, y = expend, color = GMMclust)) +
  geom_point()
```



The three clustering techniques generate pretty different results (fixing number of clusters k = 2).

HAC seperate the single observation with very high salary and expenditure from the rest of observations. K-means, in addition to the high salary-expenditure observation, clustered several other high salary observations into the group with higher salary. GMM, in contrast, seperate several high expenditure observations and 2 observations with no so high expenditure with the rest of observations.

Telling from the perspective of salary and expenditure, it seems HAC is generating the "safest" cluster prediction. K-means' prediction makes good sense as it seperates high salary observations pretty well from the others. Cluster prediction by GMM seems a bit messy. All these results can be different if we look at the predictions from the perspective of other features, e.g., session length.

2.8

```r
library(clValid)
```

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

```r
dunn(clusters = HACcuts, Data = legprof_sd)
```

```
## [1] 0.5158771
```

```r
dunn(clusters = kmeans$cluster, Data = legprof_sd)
```

```
## [1] 0.1725627
```

```r
dunn(clusters = GMMclust, Data = legprof_sd)
```

```
## [1] 0.07897372
```

```r
HACcuts <- cutree(HAC,
                  k = 4)

kmeans <- kmeans(legprof_sd,
                 centers = 4,
                 nstart = 10)

dunn(clusters = HACcuts, Data = legprof_sd)
```

```
## [1] 0.4267514
```

```r
dunn(clusters = kmeans$cluster, Data = legprof_sd)
```

```
## [1] 0.08157442
```

```r
HACcuts <- cutree(HAC,
                  k = 7)

kmeans <- kmeans(legprof_sd,
                 centers = 7,
                 nstart = 10)

dunn(clusters = HACcuts, Data = legprof_sd)
```

```
## [1] 0.283641
```

```r
dunn(clusters = kmeans$cluster, Data = legprof_sd)
```

```
## [1] 0.1780972
```

2.9

By comparing Dunn indices of these techniques, it seems GMM 2 clusters and K-means with 4 clusters perform the best in terms of minimizing the ratio of the biggest inter-cluster distance and the smallest intra-cluster distance.

One concern of using the "optimal" technique is the interpretability of the clustering result. For example, one may rationalize clustering observations into k groups which seperates legislators into different expereince or resource level. Sometimes the result might not be very interpretable, which makes a close rival clustering technique a better choice.