Assignment 7

STAT 32950

Ki Hyun

Due: 09:00 (CT) 2023-05-09

```
library(stats)
library(dplyr)
library(ggplot2)
library(mclust)
library(mgcv)
library(rlang)
```

Problem 1.

```
Ch1_2 = 0.76; Ch1_3 = 2.97; Ch1_4 = 4.88; Ch1_5 = 3.86
Ch2_3 = 0.80; Ch2_4 = 4.17; Ch2_5 = 1.96
Ch3_4 = 0.21; Ch3_5 = 1.51
Ch4_5 = 0.51
q1_data <- tibble("Ch1" = c(0, Ch1_2, Ch1_3, Ch1_4, Ch1_5),
                  "Ch2" = c(Ch1_2, 0, Ch2_3, Ch2_4, Ch2_5),
                  "Ch3" = c(Ch1_3, Ch2_3, 0, Ch3_4, Ch3_5),
                  "Ch4" = c(Ch1_4, Ch2_4, Ch3_4, 0, Ch4_5),
                  "Ch5" = c(Ch1_5, Ch2_5, Ch3_5, Ch4_5, 0))
q1_dmat <- as.dist(q1_data)
```

(a)

```
Msingle = hclust(q1_dmat, method = "single")
plot(Msingle)
```

```
Mcomplete = hclust(q1_dmat, method = "complete")
plot(Mcomplete)
```

```
Maverage = hclust(q1_dmat, method = "average")
plot(Maverage)
```

The `complete` and the `average` methods result in a similar dendrogram. However, the vertical scales are different. The `single` method has a different cluster dendrogram and the smallest vertical scale.

(b)

- Single Method: (Ch1, Ch2), Ch5, (Ch3, Ch4)
- Complete Method: (Ch3, Ch4), (Ch1, Ch2), Ch5
- Average Method: (Ch3, Ch4), (Ch1, Ch2), Ch5

Problem 2.

```
X1 = c(5, 1, -1, 3)
X2 = c(-4, -2, 1, 1)
A = c(X1[1], X2[1])
B = c(X1[2], X2[2])
C = c(X1[3], X2[3])
D = c(X1[4], X2[4])
```

```
get_dist <- function(obj1, obj2){
  if(length(obj1) != length(obj2)){
    print("Dimensions do not match")
    return(NULL)
  }
  return(sqrt(sum((obj1 - obj2)^2)))
}

get_center <- function(objects){
  n <- length(objects)
  centroid <- 0
  for(obj in objects){
    centroid = centroid + obj
  }
  centroid/n
}

# since k = 2 there are two groups
groups = list(list(A, B), list(C, D)) # initial groups

counter <- 0
deletion <- c()
centroid = get_center(groups[[1]])
this_group = groups[[1]]
other_group = groups[[2]]
for(i in 1:length(this_group)){
  member = this_group[[i]]
  current_d <- get_dist(member, centroid)
  temp_cent = get_center(append(other_group, member))
  other_d <- get_dist(member, temp_cent)
  if(other_d < current_d){
    other_group = append(other_group, member)
    deletion <- c(deletion, i)
    counter = counter + 1
  }
}
if(length(deletion) > 0){
  groups[[1]] = this_group[-deletion]
}
groups[[2]] = other_group

deletion <- c()
centroid = get_center(groups[[2]])
```

```
this_group = groups[[2]]
other_group = groups[[1]]
for(i in 1:length(this_group)){
  member = this_group[[i]]
  current_d <- get_dist(member, centroid)
  temp_cent = get_center(append(this_group, member))
  other_d <- get_dist(member, temp_cent)
  if(other_d < current_d){
    other_group = append(other_group, member)
    deletion <- c(deletion, i)
    counter = counter + 1
  }
}
if(length(deletion) > 0){
  groups[[2]] = this_group[-deletion]
}
groups[[1]] = other_group
```

The final clusters are (B, C, D), A

The corresponding centroids are (x1 = 1, x2 = 0) and (x1 = 5, x2 = -4)

The squared sitances to cluster centroids are:

A - Group 1: 32, Group 2: 0 B - Group 1: 4, Group 2: 20 C - Group 1: 5, Group 2: 61 D - Group 1: 5, Group 2: 29

Problem 3.

```
ladyrun = read.table("ladyrun23.dat")
colnames(ladyrun)=c("Country","100m","200m","400m","800m","1500m","3000m",
                    "Marathon")
```

(a)

```
dist_m <- tibble("Null" = rep(0, 54))
for(i in 1:nrow(ladyrun)){
  country <- ladyrun$Country[i]
  temp <- c()
  for(j in 1:nrow(ladyrun)){
    temp <- c(temp, get_dist(ladyrun[i, -1], ladyrun[j, -1]))
  }
  dist_m[country] = temp
}
dist_m <- dist_m[-1]
```

```
max_countries <- which(dist_m == max(as.dist(dist_m))) %/% nrow(ladyrun)
max_countries[2] = max_countries[2] + 1
print(paste0(ladyrun$Country[max_countries[1]],
             " and ",
             ladyrun$Country[max_countries[2]]))
```

```
## [1] "JPN and PNG"
```

Japan and Papua New Guinea have the maximum distance

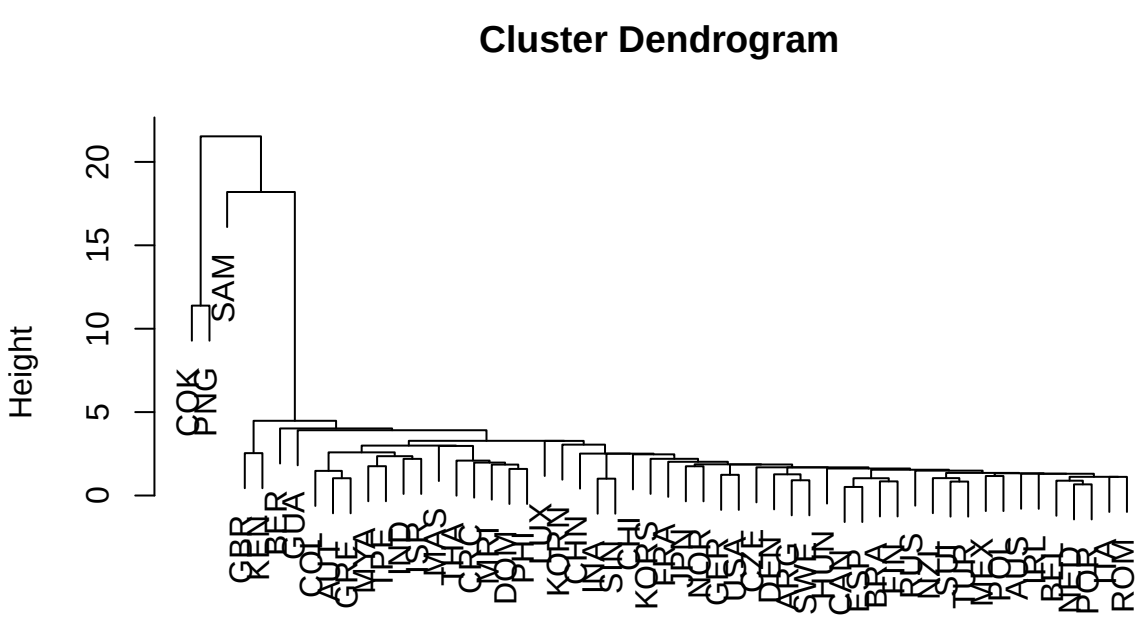
```
min_countries <- which(dist_m == min(as.dist(dist_m))) %/% nrow(ladyrun)
min_countries[2] = min_countries[2] + 1
print(paste0(ladyrun$Country[min_countries[1]],
             " and ",
             ladyrun$Country[min_countries[2]]))
```

```
## [1] "BRA and ESP"
```

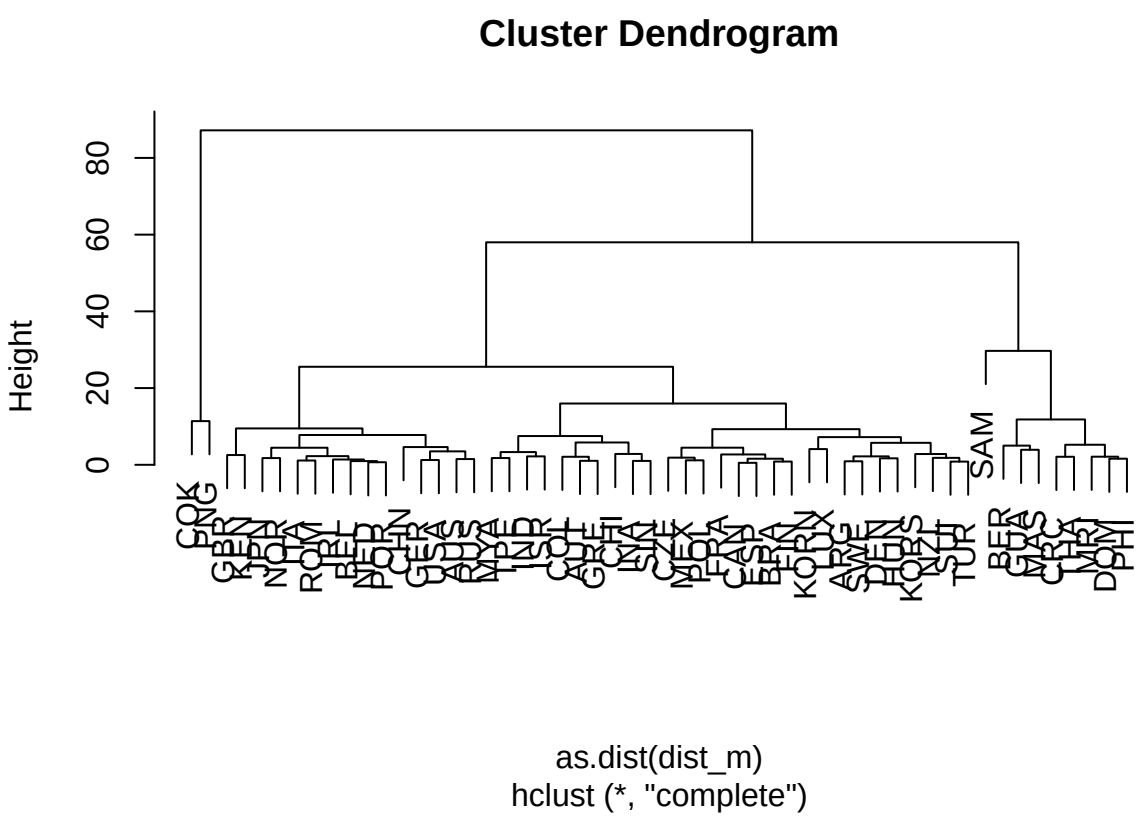
Brazil and Spain have the minimum distance.

(b)

```
Msingle_2 = hclust(as.dist(dist_m), method = "single")
plot(Msingle_2)
```




```
Mcomplete_2 = hclust(as.dist(dist_m), method = "complete")
plot(Mcomplete_2)
```



The structure and the vertical scale of the two dendrograms are different. However, the two are similar in the fact that ("COK", "PNG") and "SAM" are clustered different to other countries.

When $k = 8$ or 7, the three smallest clusters are "SAM", ("COK", "PNG"), and ("GBR", "KEN").

(c)

```
kmeans(ladyrun[-1], 9)
```

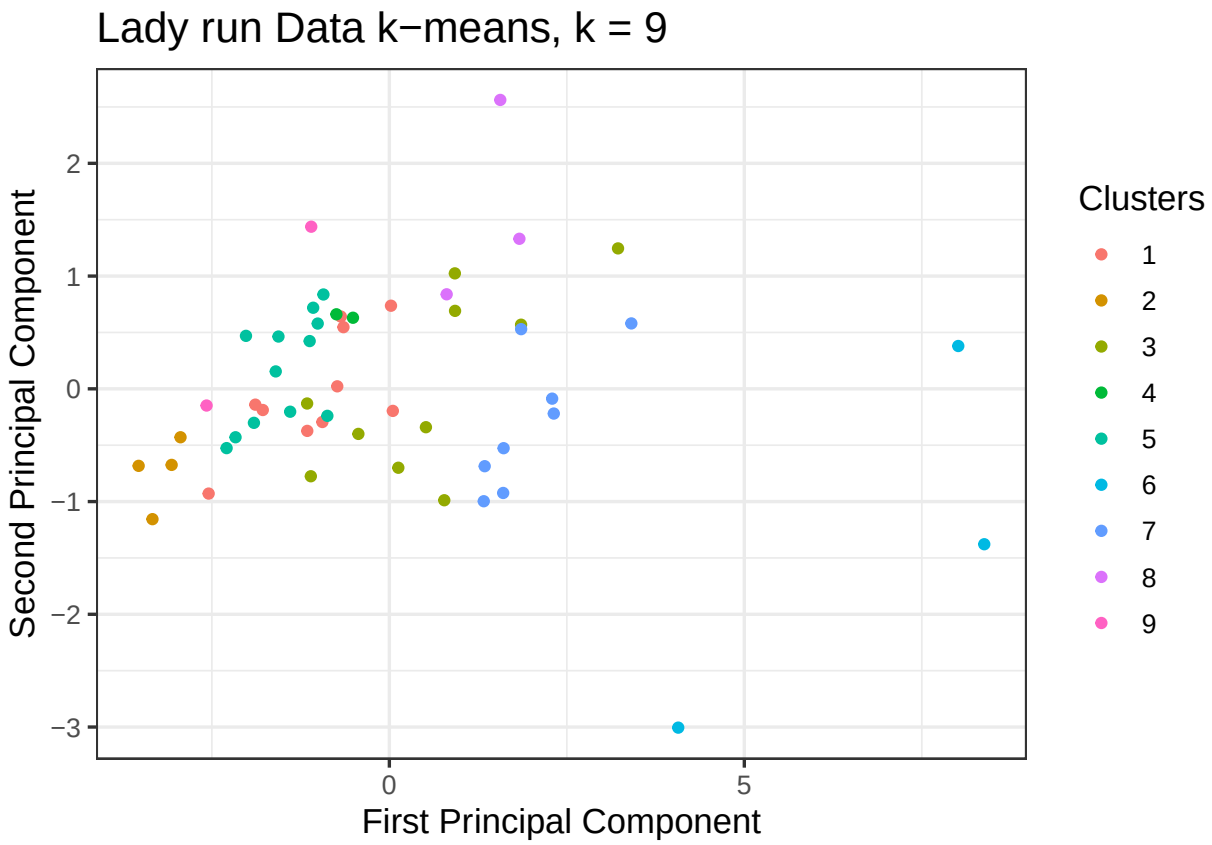
```
## K-means clustering with 9 clusters of sizes 7, 5, 3, 6, 8, 15, 2, 6, 2
##
## Cluster means:
##       100m     200m     400m     800m    1500m     3000m Marathon
## 1 11.57857 23.65571 53.64571 2.101429 4.422857  9.658571 168.3071
## 2 11.34000 23.10400 52.23000 2.048000 4.260000  9.444000 158.9600
## 3 12.06333 24.82667 57.71667 2.270000 4.953333 11.476667 208.3500
## 4 11.37167 23.13833 52.09000 2.006667 4.061667  8.693333 146.7767
## 5 11.37000 23.28625 52.34250 2.030000 4.233750  9.226250 153.0725
## 6 11.03933 22.45867 49.84667 1.962667 3.994000  8.500000 141.9440
## 7 11.16000 22.73500 50.30500 1.955000 3.965000  8.380000 134.6600
## 8 11.02667 22.28500 49.43000 1.953333 4.046667  8.661667 147.0517
## 9 11.78000 24.53000 56.15000 2.020000 4.300000  9.085000 147.2700
##
## Clustering vector:
```

```
##  [1] 5 6 5 6 1 8 8 5 6 5 3 1 8 4 1 8 8 6 7 5 1 4 5 2 6 2 6 6 7 4 9 9 1 1 6 2 6 4
## [39] 6 3 1 6 6 6 6 3 5 8 5 4 2 2 4 6
##
## Within cluster sum of squares by cluster:
## [1]  83.65709  24.55632 494.69053  21.42585  61.83926 103.62221   3.22680
## [8]  12.74872   8.38785
##  (between_SS / total_SS =  94.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

I would choose $k = 9$.

```
M0_q3 <- kmeans(ladyrun[-1], 9)
PC <- princomp(ladyrun[-1], cor = T)

tibble(x = PC$scores[,1],
       y = PC$scores[,2],
       groups_lab = as.character(M0_q3$cluster)) %>%
  ggplot(mapping = aes(x = x, y = y, color = groups_lab)) +
  geom_point() +
  scale_color_discrete() +
  labs(x = "First Principal Component",
       y = "Second Principal Component",
       color = "Clusters",
       title = "Lady run Data k-means, k = 9") +
  theme_bw(base_size = 13)
```


Lady run Data k−means, k = 9
Second Principal Component
First Principal Component
2
1
0
−1
−2
−3
0
5
Clusters
1
2
3
4
5
6
7
8
9

Problem 4.

(a)

$$
\begin{aligned}
f_1(0.1) = 0.2,\ f_2(0.1) = 1.8 \\
f_1(0.2) = 0.4,\ f_2(0.2) = 1.6 \\
f_1(0.3) = 0.6,\ f_2(0.1) = 1.4 \\
f_1(0.4) = 0.8,\ f_2(0.4) = 1.2 \\
f_1(0.7) = 1.4,\ f_2(0.7) = 0.6
\end{aligned}
$$

The likelihood function becomes:

$$
\begin{aligned}
&\prod_{i=1}^{5}(p_1 f_1(x_i) + p_2 f_2(x_i)) \\
=&(0.2p_1 + 1.8p_2) \cdot (0.4p_1 + 1.6p_2) \cdot (0.6p_1 + 1.4p_2) \cdot (0.8p_1 + 1.2p_2) \cdot (1.4p_1 + 0.6p_2)
\end{aligned}
$$

(b)

$$
\begin{aligned}
f_1(0.1) = 0.2,\ f_2(0.1) = 1.8 \\
f_1(0.2) = 0.4,\ f_2(0.2) = 1.6 \\
f_1(0.3) = 0.6,\ f_2(0.1) = 1.4 \\
f_1(0.4) = 0.8,\ f_2(0.4) = 1.2 \\
f_1(0.9) = 1.8,\ f_2(0.9) = 0.2
\end{aligned}
$$

The likelihood function becomes:

$$
\begin{aligned}
&\prod_{i=1}^{5}(p_1 f_1(x_i) + p_2 f_2(x_i)) \\
=&(0.2p_1 + 1.8p_2) \cdot (0.4p_1 + 1.6p_2) \cdot (0.6p_1 + 1.4p_2) \cdot (0.8p_1 + 1.2p_2) \cdot (1.8p_1 + 0.2p_2)
\end{aligned}
$$

(c)

$$
\begin{aligned}
f_1(0.1) = 0.2,\ f_2(0.1) = 1.8 \\
f_1(0.2) = 0.4,\ f_2(0.2) = 1.6 \\
f_1(0.3) = 0.6,\ f_2(0.1) = 1.4 \\
f_1(0.6) = 1.2,\ f_2(0.6) = 0.8 \\
f_1(0.9) = 1.8,\ f_2(0.9) = 0.2
\end{aligned}
$$

The likelihood function becomes:

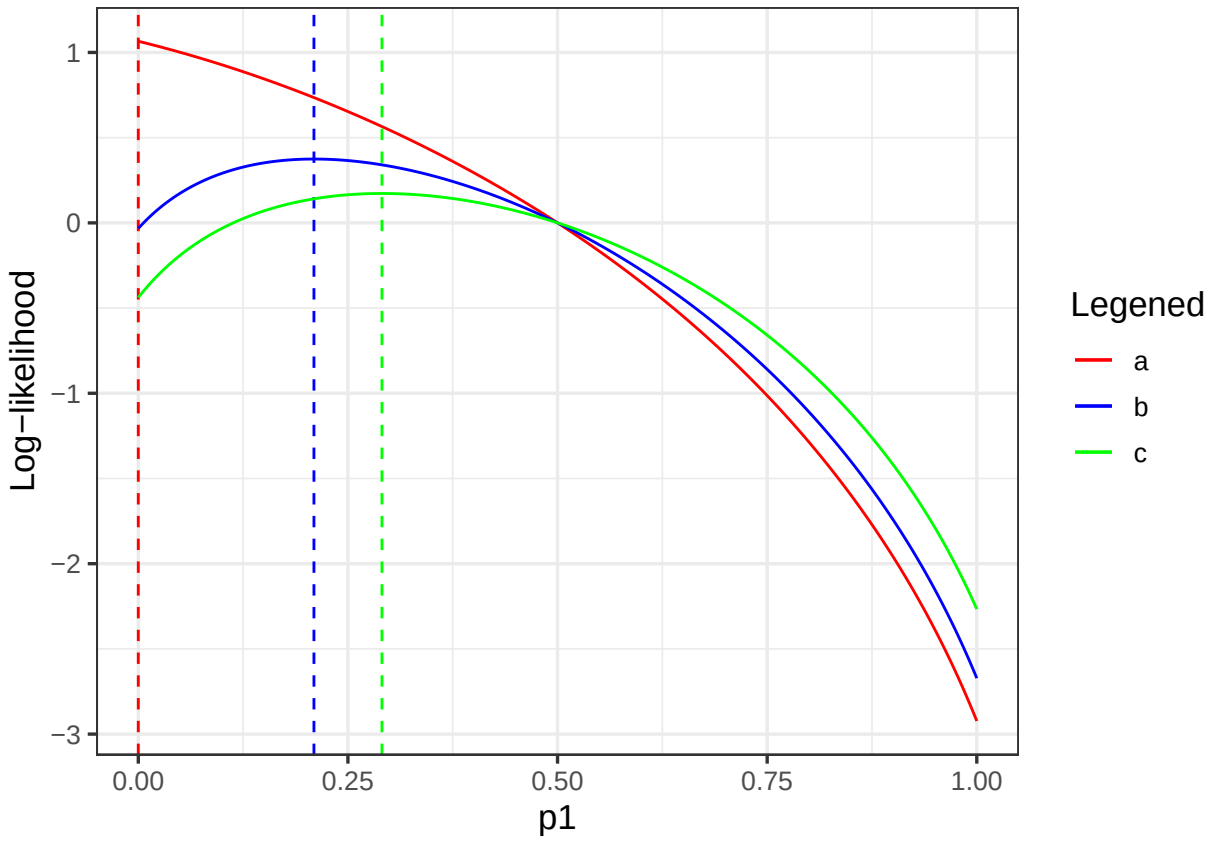
$$
\begin{aligned}
&\prod_{i=1}^{5}(p_1 f_1(x_i) + p_2 f_2(x_i)) \\
=&(0.2p_1 + 1.8p_2) \cdot (0.4p_1 + 1.6p_2) \cdot (0.6p_1 + 1.4p_2) \cdot (1.2p_1 + 0.8p_2) \cdot (1.8p_1 + 0.2p_2)
\end{aligned}
$$

(d)

```
q3_log_likelihood <- function(p, xs){
  ret = 0
  for(constant in xs){
    ret = ret + log(2*constant * p + (2 - 2*constant) * (1 - p))
  }
  ret
}
```

```
p <- seq(0, 1, by = 10^(-4))
y1 = q3_log_likelihood(p, c(0.1, 0.2, 0.3, 0.4, 0.7))
y2 = q3_log_likelihood(p, c(0.1, 0.2, 0.3, 0.4, 0.9))
y3 = q3_log_likelihood(p, c(0.1, 0.2, 0.3, 0.6, 0.9))

ggplot() +
  geom_line(mapping = aes(x, y, col = "a"),
            data = tibble(x = p,
                          y = y1)) +
  geom_vline(xintercept = p[y1 == max(y1)],
             color = "red", linetype = "dashed") +
  geom_line(mapping = aes(x, y, col = "b"),
            data = tibble(x = p,
                          y = y2)) +
  geom_vline(xintercept = p[y2 == max(y2)],
             color = "blue", linetype = "dashed") +
  geom_line(mapping = aes(x, y, col = "c"),
            data = tibble(x = p,
                          y = y3)) +
  geom_vline(xintercept = p[y3 == max(y3)],
             color = "green", linetype = "dashed") +
  labs(x = "p1", y = "Log-likelihood", color = "Legened") +
  scale_color_manual(values = c("a" = "red", "b" = "blue", "c" = "green")) +
  theme_bw(base_size = 13)
```



(e)

For (a), $(\hat{p}_1, \hat{p}_2)$ can be estimated as $(0, 1)$.

For (b), $(\hat{p}_1, \hat{p}_2)$ can be estimated as $(0.21, 0.79)$.

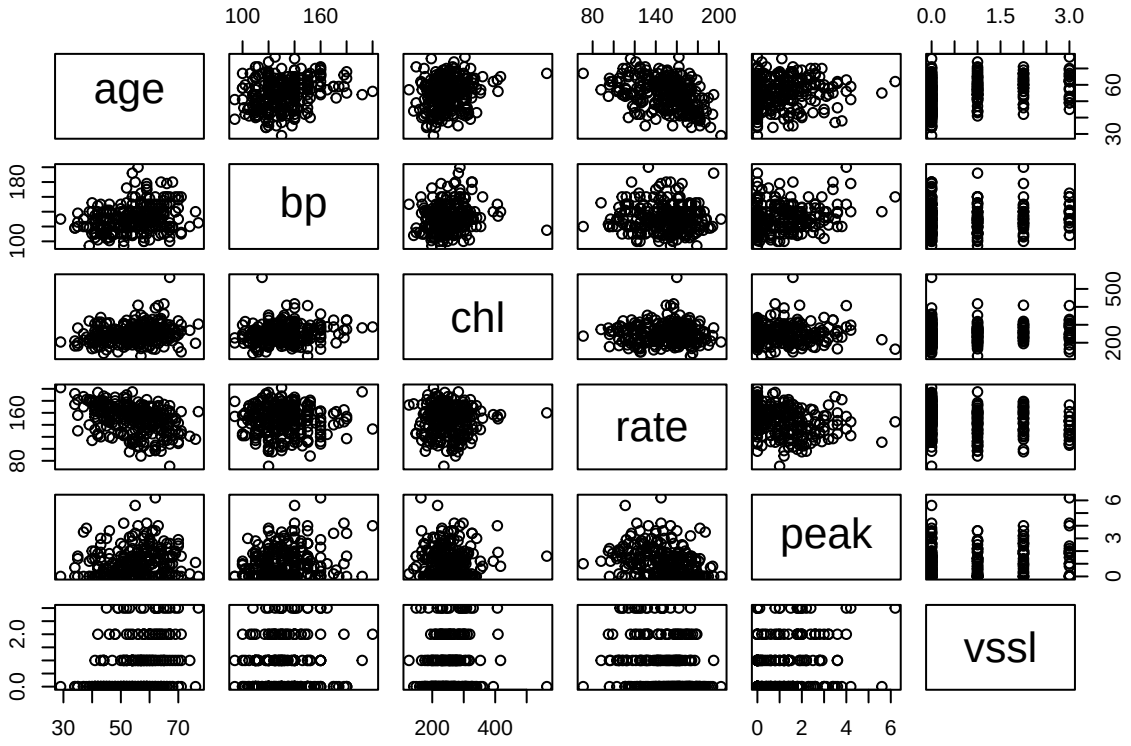
For (c), $(\hat{p}_1, \hat{p}_2)$ can be estimated as $(0.29, 0.71)$.

The estimates seems reasonable except for the case of (a)

Problem 5.

```
heart=read.table("heart.dat")
colnames(heart)=c("age","sex","chest","bp","chl","sugar","ecg","rate","angina",
                  "peak","slope","vssl","thal","ill")
```

```
heart_real = heart[c(1, 4, 5, 8, 10, 12)]
pairs(heart_real)
```



Now using BIC to choose the number of clusters:

```
mheart = Mclust(heart_real)
summary(mheart)
```

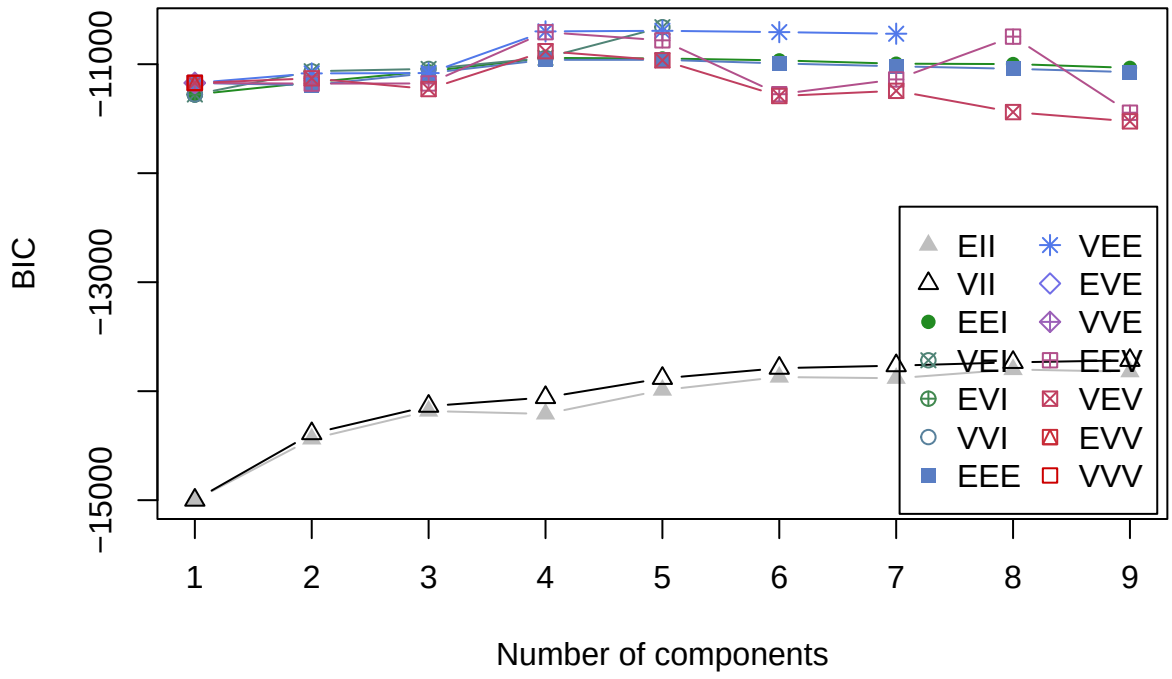
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEI (diagonal, equal shape) model with 5 components:
##
##  log-likelihood   n df       BIC       ICL
##       -5207.073 270 44 -10660.48 -10698.41
##
## Clustering table:
```

```
##  1  2  3  4  5
## 52 42 54 58 64
```

The chosen number of clusters is $k = 5$

Now looking at BIC plot for model selection:

```
plot(mheart, what = c("BIC"))
```



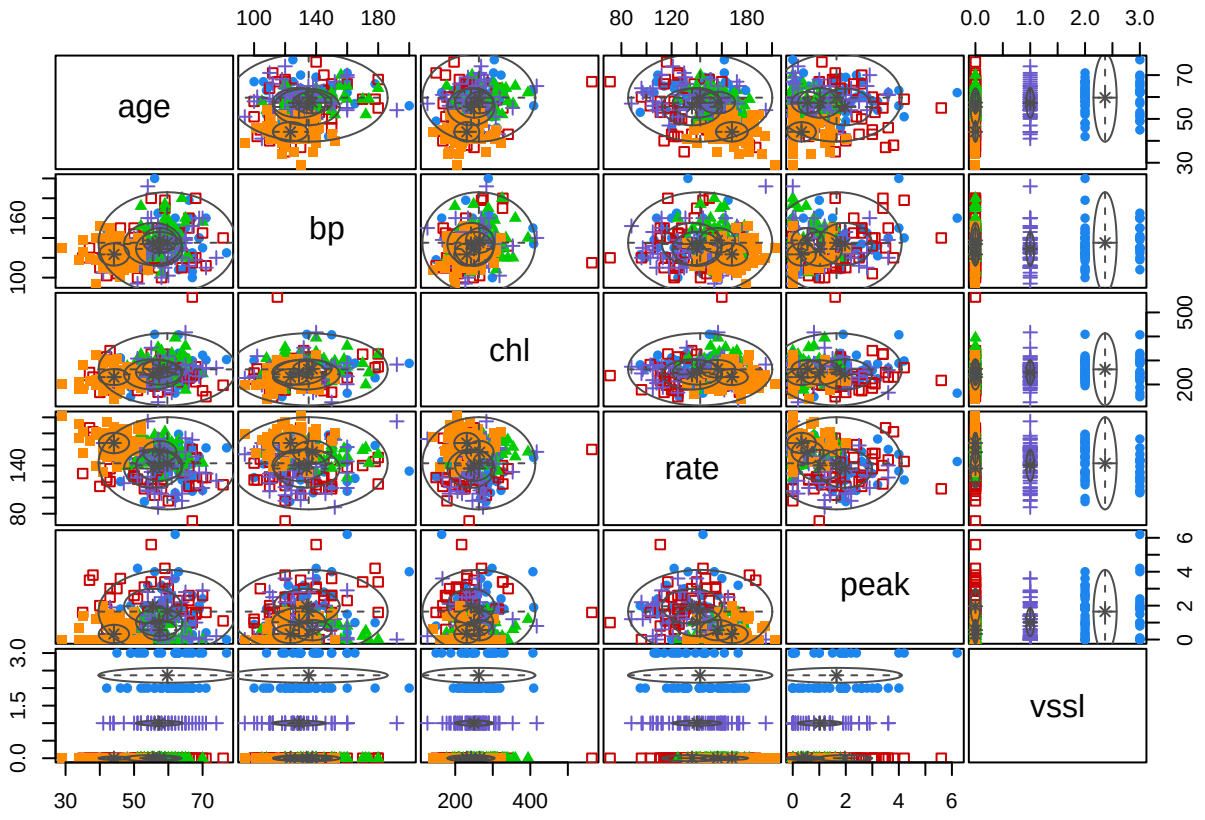
Now looking at the best candidates of model assumptions by BIC:

```
summary(mclustBIC(heart_real))
```

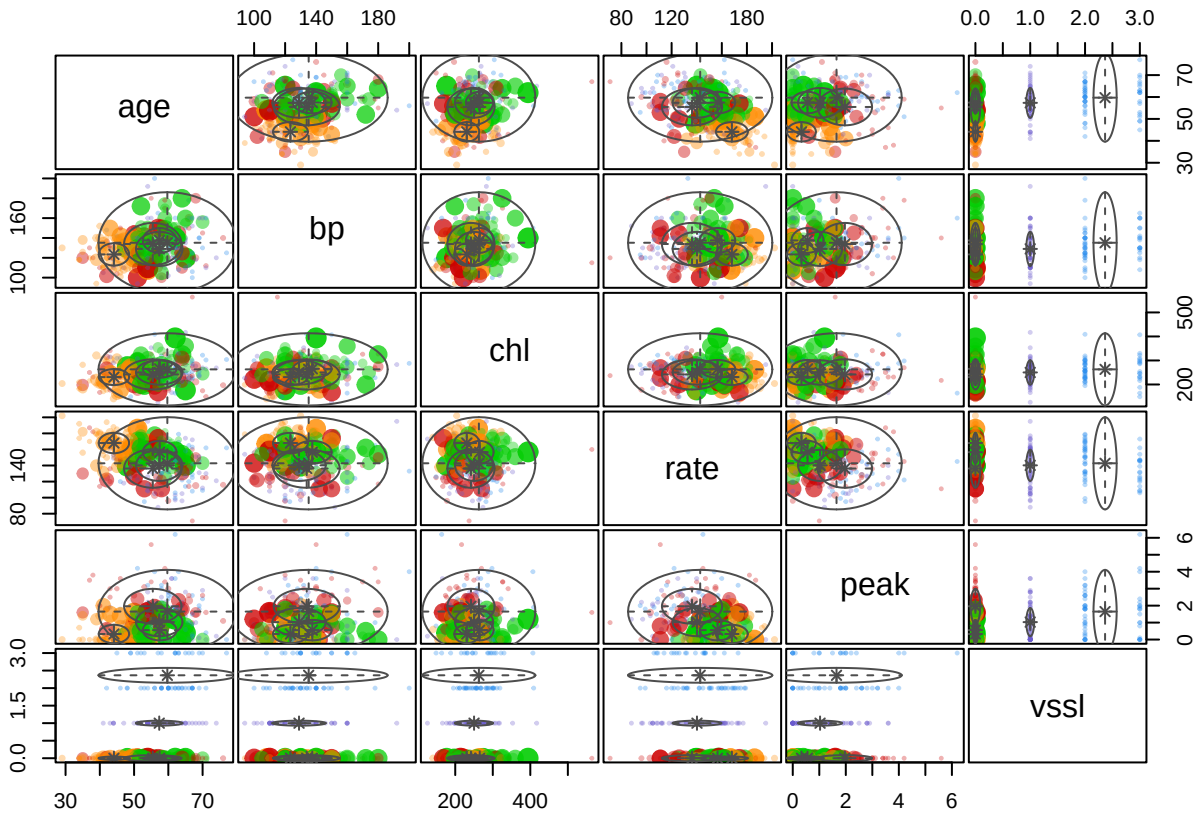
```
## Best BIC values:
##              VEI,5        VEE,5        VEE,4
## BIC      -10660.48 -10693.73993 -10699.02309
## BIC diff      0.00    -33.26405    -38.54721
```

Now fixing the k to be 5,

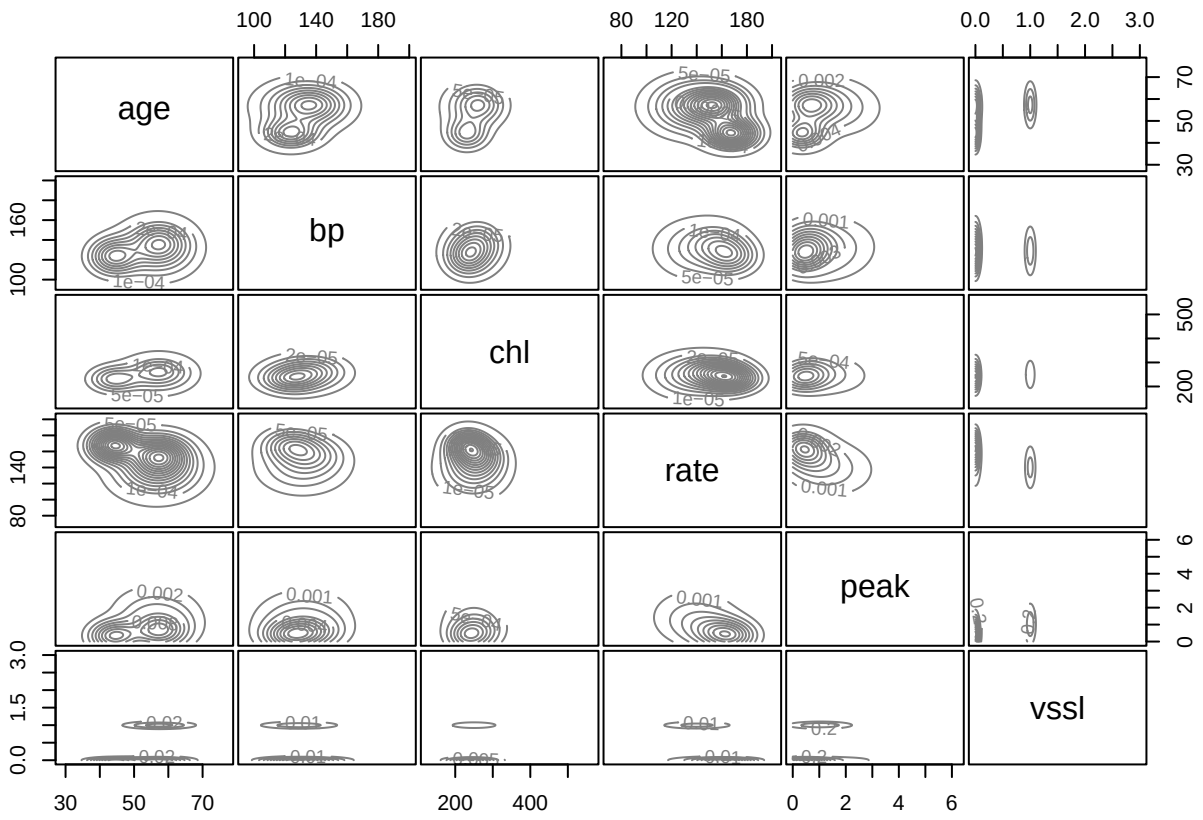
```
heart5 = Mclust(heart_real, G = 5)
plot(heart5, what = c("classification"))
```




```
plot(heart5, what = c("uncertainty"))
```




```
plot(heart5, what = c("density"))
```



Looking at the model assumptions for $k = 5$ mixtures:

```
summary(mclustBIC(heart_real, G = 5))
```

```
## Best BIC values:
##              VEI,5        VEE,5       EEV,5
## BIC      -10660.48 -10693.73993 -10781.0647
## BIC diff      0.00    -33.26405   -120.5888
```

Finally looking at the mixture proportions:

```
heart5$parameters$pro
```

```
## [1] 0.1925926 0.1666624 0.1932443 0.2148148 0.2326859
```

Problem 6.

(a)

Using averages of each column to impute initial estimates:

$$\tilde{\mathbf{X}} = \begin{bmatrix} 3 & 6 & 0 \\ 4 & 4 & 3 \\ 4 & 8 & 3 \\ 5 & 6 & 2 \end{bmatrix}$$

(b)

First, the mean can be estimated from the above as:

$$\tilde{\mu} = \begin{bmatrix} 4 \\ 6 \\ 2 \end{bmatrix}$$

The maximum likelihood covariance matrix can be estimated as:

```
X_t <- matrix(c(3, 6, 0, 4, 4, 3, 4, 8, 3, 5, 6, 2), ncol = 3, byrow = T)
mu = colMeans(X_t)
Sigma = cov(X_t) * ((nrow(X_t) - 1)/nrow(X_t))
Sigma
```

```
##      [,1] [,2] [,3]
## [1,]  0.5    0  0.5
## [2,]  0.0    2  0.0
## [3,]  0.5    0  1.5
```

$$\tilde{\Sigma} \approx \begin{bmatrix} 0.5 & 0.0 & 0.5 \\ 0.0 & 2.0 & 0.0 \\ 0.5 & 0.0 & 1.5 \end{bmatrix}$$

(c)

i)

Using the estimated $\tilde{\mu}$ and $\tilde{\Sigma}$, we may update $\tilde{x}_{13}$ with the expected value of the $X_1 \mid X_2 = x_2$ conditional distribution as:

$$\begin{aligned} &\tilde{\mu}_1 + \begin{bmatrix} 0.0 & 0.5 \end{bmatrix} \times \begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 1.5 \end{bmatrix}^{-1} \times \left(\begin{pmatrix} 8 \\ 3 \end{pmatrix} - \begin{pmatrix} 6 \\ 2 \end{pmatrix} \right) \\ =&4 + \begin{bmatrix} 0.0 & 0.5 \end{bmatrix} \times \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & \frac{2}{3} \end{bmatrix} \times \begin{pmatrix} 2 \\ 1 \end{pmatrix} \\ =&4 + \begin{bmatrix} 0.0 & \frac{1}{3} \end{bmatrix} \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ \approx&4.33 \end{aligned}$$

Therefore, the updated $\tilde{\mathbf{X}}'$ is:

$$\tilde{\mathbf{X}}' \approx \begin{bmatrix} 3 & 6 & 0 \\ 4 & 4 & 3 \\ 4.33 & 8 & 3 \\ 5 & 6 & 2 \end{bmatrix}$$

```
X_t_p <- matrix(c(3, 6, 0, 4, 4, 3, 13/3, 8, 3, 5, 6, 2), ncol = 3, byrow = T)
mu_p = colMeans(X_t_p)
Sigma_p = cov(X_t_p) * ((nrow(X_t_p) - 1)/nrow(X_t_p))
Sigma_p
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.5208333 0.1666667 0.5833333
## [2,] 0.1666667 2.0000000 0.0000000
## [3,] 0.5833333 0.0000000 1.5000000
```

Now the updated estimates of $\tilde{\mu}$ and $\tilde{\Sigma}$ are:

$$\tilde{\mu}' = \begin{bmatrix} 4.083 \\ 6 \\ 2 \end{bmatrix}$$

$$\tilde{\Sigma}' \approx \begin{bmatrix} 0.521 & 0.167 & 0.583 \\ 0.167 & 2.0 & 0.0 \\ 0.583 & 0.0 & 1.5 \end{bmatrix}$$

ii)

```
update_23 <- mu[2:3] +
  c(Sigma_p[1, 2], Sigma_p[1, 3])%*%solve(Sigma_p[2:3, 2:3]) * (5 - mu_p[1])
update_23
```

```
##          [,1]     [,2]
## [1,] 6.076389 2.356481
```

Therefore, the updated $\tilde{\mathbf{X}}''$ is:

$$\tilde{\mathbf{X}}'' \approx \begin{bmatrix} 3 & 6 & 0 \\ 4 & 4 & 3 \\ 4.33 & 8 & 3 \\ 5 & 6.08 & 2.36 \end{bmatrix}$$

```
X_t_pp <- matrix(c(3, 6, 0, 4, 4, 3, 13/3, 8, 3, 5, update_23[1], update_23[2]),
                 ncol = 3, byrow = T)
mu_pp = colMeans(X_t_pp)
Sigma_pp = cov(X_t_pp) * ((nrow(X_t_pp) - 1)/nrow(X_t_pp))
Sigma_pp
```

```
##           [,1]        [,2]        [,3]
## [1,] 0.5208333 0.184172454 0.665027006
## [2,] 0.1841725 2.001094112 0.005105855
## [3,] 0.6650270 0.005105855 1.523827321
```

Now the updated estimates, after the first iteration, of $\tilde{\mu}$ and $\tilde{\Sigma}$ are:

$$\tilde{\mu}'' = \begin{bmatrix} 4.083 \\ 6.02 \\ 2.09 \end{bmatrix}$$

$$\tilde{\Sigma}'' \approx \begin{bmatrix} 0.521 & 0.184 & 0.665 \\ 0.184 & 2.00 & 0.00511 \\ 0.665 & 0.00511 & 1.52 \end{bmatrix}$$

Problem 7.

(a)

Given that $a \in \mathbf{R}^p$ and $b \in \mathbf{R}^p$, $(a-b) \in \mathbf{R}^p$. Therefore, we may let:

$$a - b = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}$$

where $x_i \in \mathbf{R}$. Moreover,

$$(a-b)^T(a-b) = \sum_{i=1}^{p} x_i^2 \geq 0$$

In other words $d(a,b) \geq 0$ for all a and b.

Now, given that $d(a,b) - d(c,d) \geq 0$,

$$\begin{aligned} & d(a,b) - d(c,d) \geq 0 \\ \Leftrightarrow & (d(a,b) + d(c,d))(d(a,b) - d(c,d)) \geq 0 \\ \Leftrightarrow & d^2(a,b) - d^2(c,d) \geq 0 \\ \Leftrightarrow & D(a,b) - D(c,d) \geq 0 \end{aligned}$$

Therefore, the two dissimilarity measures are global-order equivalent.

(b)

Using a simple example if we let

$$a = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$
$$b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$c = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Now,

$$d_1(a,b) = \sqrt{(1-0)^2 + (-1-0)^2} = \sqrt{2}$$

$$d_1(a,c) = \sqrt{(1-1)^2 + (-1-1)^2} = 2$$

We are able to see that $d_1(a,c) > d_1(a,b)$.

However, if we look at the city-block metric:

$$d_2(a,b) = \frac{|1-0|}{|1|+|0|} + \frac{|-1-0|}{|-1|+|0|} = 2$$

$$d_2(a,c) = \frac{|1-1|}{|1|+|1|} + \frac{|-1-1|}{|-1|+|1|} = 1$$

Therefore, in this case, $d_1(a,b) > d_1(a,c)$ and the above serves as a counter example.