SC-HW5

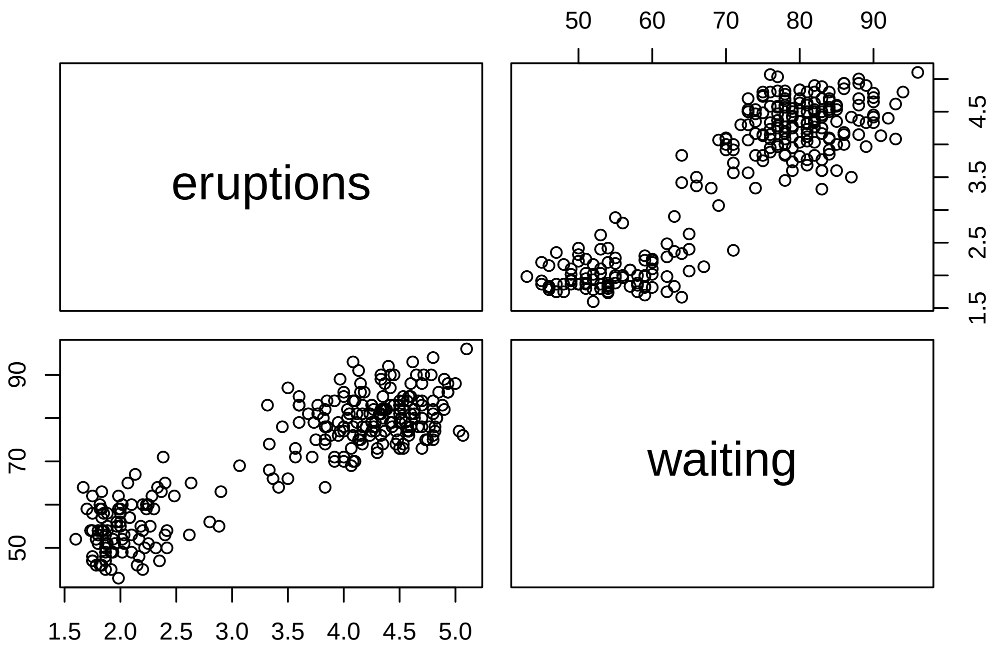
ID : 111024517 Name : 鄭家豪

Problem 1

Let X_i and Y_i be represented as eruptions and waiting observation obtained from R(**faithful**),i=1,2,..,n=272.

- Pairwise scatter plot:

```
data(faithful)
pairs(faithful)
```



- Model:

$$(X_i, Y_i) \overset{\text{i.i.d.}}{\sim} \gamma N_2(\mu_1, \Sigma_1) + (1-\gamma) N_2(\mu_2, \Sigma_2), p_i \overset{\text{i.i.d.}}{\sim} Ber(\gamma).$$

(a)

- EM algorithm:

1. Given initial:From the pairwise scatter plot,set

$$\mu_1^{(0)} = (2, 60), \mu_2^{(0)} = (4.5, 80); \Sigma_1^{(0)} = \Sigma_2^{(0)} = I_2; p^{(0)} = 0.5.$$

2. E-step:

   Let $f(x, y; \mu, \Sigma)$ be the joint pdf,$\theta = (\mu_1, \Sigma_1, \mu_2, \Sigma_2, p)$,

$$\begin{aligned} Q(\theta|\hat{\theta}) &= \sum_{i=1}^{n} \{\hat{p}_i \log f(X_i, Y_i; \mu_1, \Sigma_1) + (1 - \hat{p}_i) \log f(X_i, Y_i; \mu_2, \Sigma_2)\} \\ &+ \sum_{i=1}^{n} \{\hat{p}_i \log \gamma + (1 - \hat{p}_i) \log(1 - \gamma)\}, \text{where } \hat{p}_i = \frac{\gamma f(X_i, Y_i; \mu_1, \Sigma_1)}{\gamma f(X_i, Y_i; \mu_1, \Sigma_1) + (1 - \gamma) f(X_i, Y_i; \mu_{x2}, \Sigma_2)} \end{aligned}$$

3. M-step:

   Updata θ via

$$\theta^{(t+1)} = \arg\ \max_\theta Q(\theta|\theta^{(t)})$$

   Refer to https://arxiv.org/pdf/1901.06708.pdf , the analytic form of θ:

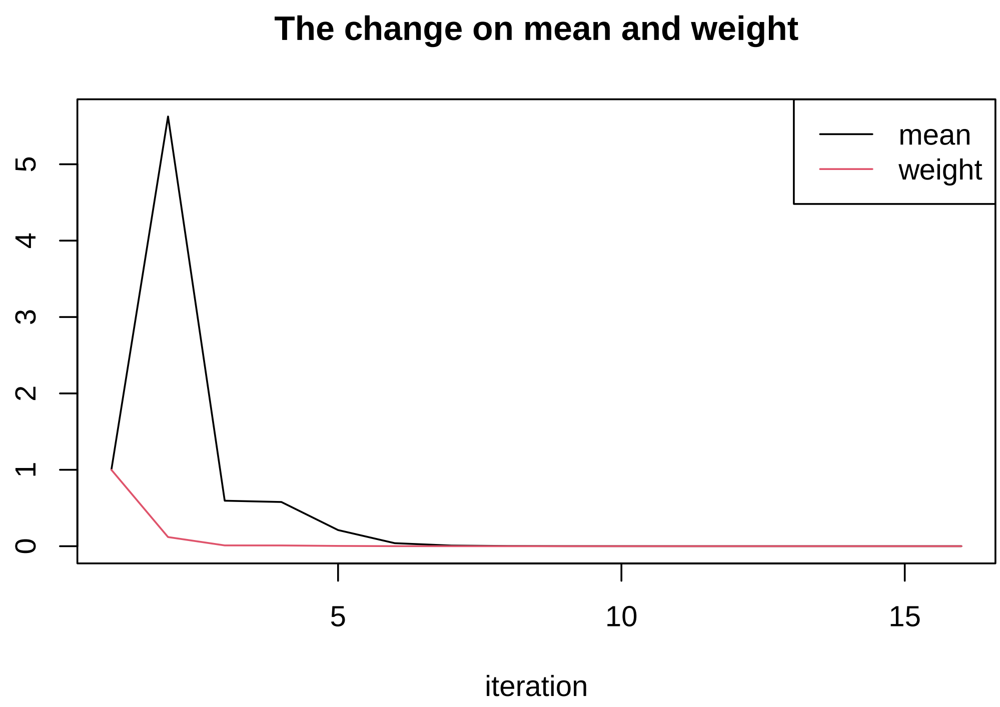
$$\begin{aligned} &\hat{\mu}_j = \frac{\hat{p_j}^T [X, Y]}{\sum_{i=1}^n \hat{p}_{ij}} \in \mathbb{R}^{2\times 1}, \hat{\Sigma}_j = \frac{[X - \hat{\mu}_{j1}, Y - \hat{\mu}_{j2}]^T \text{diag}(\hat{p}_{ij})[X - \hat{\mu}_{j1}, Y - \hat{\mu}_{j2}]}{\sum_{i=1}^n \hat{p}_{ij}} \in \mathbb{R}^{2\times 2} \\ &\hat{\gamma}_j = \frac{1}{n} \sum_{i=1}^{n} \hat{p}_{ij}, j = 1, 2. \end{aligned}$$

4. Repeat 2 and 3 until $||\mu_1^{(t)} - \mu_1^{(t-1)}||_1 + ||\mu_2^{(t)} - \mu_2^{(t-1)}||_1 < 10^{-6}$ and $||\gamma_1^{(t)} - \gamma_1^{(t-1)}||_1 \leq 10^{-9}$.

```
fmul <- function(mu,sigma2,xvec){
  det <- det(sigma2)
  d = length(xvec)
  maha <- (xvec-mu) %*% solve(sigma2) %*% t(xvec-mu)
  ((2*pi)^d*det)^(-1/2)*exp(-maha/2)
} #joint pdf
mu1 <- mu1.all <- t(c(2,60));mu2 <- mu2.all <- t(c(4.5,80))
sigma2.1 <- sigma2.2 <-diag(c(1,1))
error <-err.all<- c(1,1)
p.all <- p <- 0.5
n= dim(faithful)[1]
count = 1
X <- as.matrix(faithful)
while(error[1]>=10^(-6) | error[2]>=10^(-9)){
  # E-step
```

```r
  r <- numeric(n) #membership
  for(i in 1:n){
    tot <- p*fmul(mu1,sigma2.1,X[i,])+(1-p)*fmul(mu2,sigma2.2,X[i,])
    r[i] <- p*fmul(mu1,sigma2.1,X[i,])/tot #rhat
  }
  # M-step
  mu1.t <- (r%*%X)/sum(r) ; mu2.t <- ((1-r)%*%X)/sum(1-r)
  sigma21.t <- t(X- rep(1,n) %*% mu1.t) %*% diag(r) %*% (X- rep(1,n) %*% mu1.t)/sum(r)
  sigma22.t <- t(X- rep(1,n) %*% mu2.t) %*% diag(1-r) %*% (X- rep(1,n) %*% mu2.t)/sum(1-r)
  p1 <- sum(r)/n
  # stop criterion
  error[1] <- sum(abs(mu1-mu1.t)) + sum(abs(mu2-mu2.t))
  error[2] <- abs(p1-p)
  # update
  mu1.all <-rbind(mu1.all,mu1.t)
  mu2.all <-rbind(mu2.all,mu2.t)
  p.all <-c(p.all,p1)
  mu1 <- mu1.t ; mu2 <-mu2.t
  sigma2.1 <- sigma21.t
  sigma2.2 <- sigma22.t
  p <- p1
  err.all <- rbind(err.all,error)
  count=count+1
}
```

```
The iteration : 16
```



From the above, we know that the EM algorithm is successfully converged for the estimated normal components to 2-dimenstional data.

Summary,the estimated normal components:

- μ:

```
mu <- rbind(mu1.all[count,],mu2.all[count,])
mu
```

```
     eruptions  waiting
[1,]  2.036388 54.47852
[2,]  4.289662 79.96812
```

- Σ_1:

```
sigma21.t
```

```
           eruptions    waiting
eruptions 0.06916767  0.4351676
waiting   0.43516763 33.6972821
```

- Σ_2:

```
sigma22.t
```

```
          eruptions    waiting
eruptions 0.1699684  0.9406093
waiting   0.9406093 36.0462113
```

- The weight:

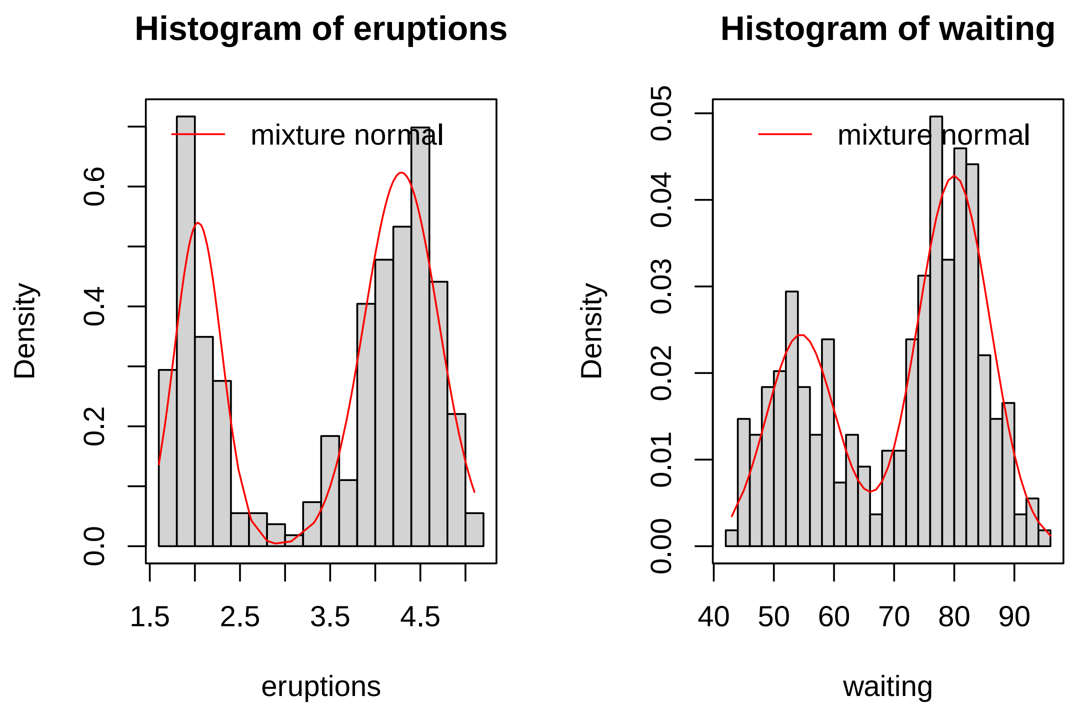
```
cat("1 cluster:",p.all[count],"\n") ; cat("2 cluster:",1-p.all[count],"\n")
```

```
1 cluster: 0.3558729

2 cluster: 0.6441271
```

(b)

Using the EM estimate to draw the fitted line in two histogram:

```
par(mfrow=c(1,2))
x1 <- sort(X[,1])
hist(x1, probability=TRUE,20,
     main="Histogram of eruptions",xlab="eruptions")

pdf1.em <- p*dnorm(x1,mean = mu1[1],sd = sqrt(sigma2.1[1,1])) +
  (1-p)*dnorm(x1,mean = mu2[1],sd = sqrt(sigma2.2[1,1]))
lines(x1,pdf1.em,col="red")
box()
legend("topleft",legend=c("mixture normal"),
       col="red",lty=1,lwd=1,bty="n")
x2 <- sort(X[,2])
hist(x2, probability=TRUE,20,
     main="Histogram of waiting",xlab="waiting")
pdf2.em <- p*dnorm(x2,mean = mu1[2],sd = sqrt(sigma2.1[2,2])) +
  (1-p)*dnorm(x2,mean = mu2[2],sd = sqrt(sigma2.2[2,2]))
lines(sort(x2),pdf2.em,col="red",lwd=1)
box()
legend("top",legend=c("mixture normal"),
       col="red",lty=1,lwd=1, bty="n")
```

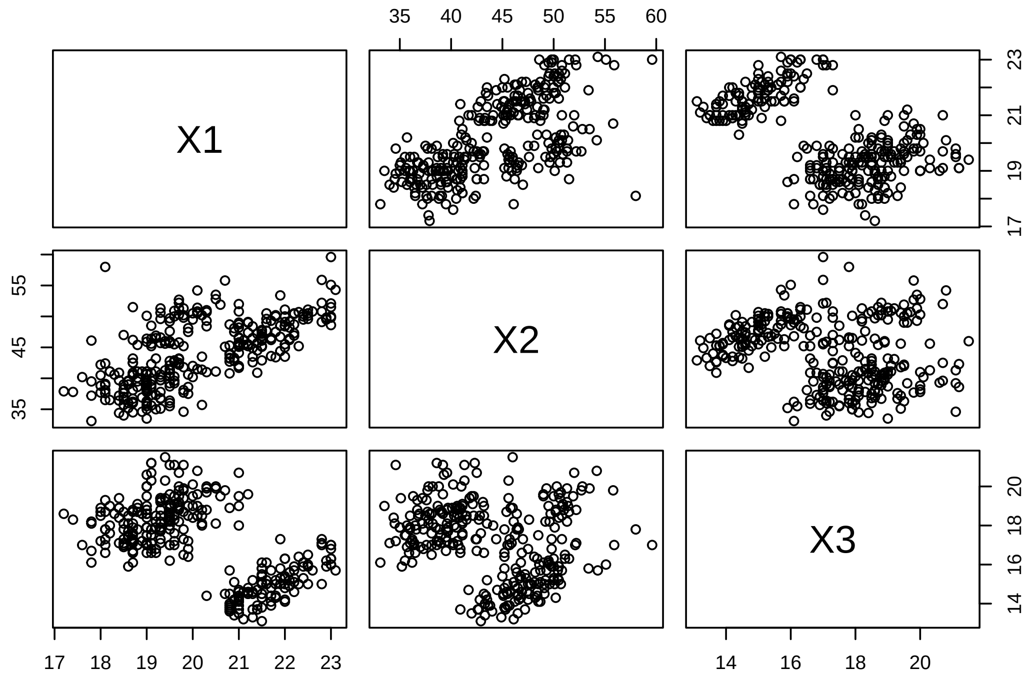


From the above,we can see that the EM estimates are quite good.

In general,the distribution pattern is well captured.

Problem 2

```
data2 <- read.csv("DataC.csv")
pairs(data2)
```



(a)

Assume that this data obeys Multivariate mixture of Gaussians.

Let $g(x = (x1, x2, x3); \mu, \Sigma)$ be the pdf of Multivariate normal,

$$f(x; \mu_1, ..., \mu_K, \Sigma_1, ..., \Sigma_K) = \sum_{k=1}^{K} \tau_k g(x; \mu_k, \Sigma_k),$$

$$\text{where the weight } p_{ki} \overset{i.i.d}{\sim} \text{Multinomial}(n, \tau_1, ..., \tau_K), \sum_{j=1}^{K} \tau_j = 1.$$

- EM algorithm

  Refer to https://arxiv.org/pdf/1901.06708.pdf:

1. Given initial:

$$\mu_k^{(0)} \in \mathbb{R}^{3x1}, \Sigma_k^{(0)} = I_3 \in \mathbb{R}^{3x3}; p_k^{(0)} = (1/k, ..., 1/k) \in \mathbb{R}^{kx1}$$

2. E-step:

$$\hat{\gamma}_{ik} = \frac{p_k f(X_i, Y_i; \mu_k, \Sigma_k)}{\sum_{j=1}^{3} p_k f(X_i, Y_i; \mu_k, \Sigma_k))}$$

3. M-step:

$$\mu_k^{(t)} = \frac{\sum_{i=1}^{n} \hat{r}_{ik}^{(t)} x_i}{\sum_{i=1}^{n} \hat{r}_{ik}^{(t)}} \in \mathbb{R}^3; \hat{\Sigma}_k = \frac{\sum_{i=1}^{n} \hat{r}_{ik}^{(t)} (x_i - \mu_k^{(t)})(x_i - \mu_k^{(t)})^T}{\sum_{i=1}^{n} \hat{r}_{ik}^{(t)}} \in \mathbb{R}^{3x3},$$

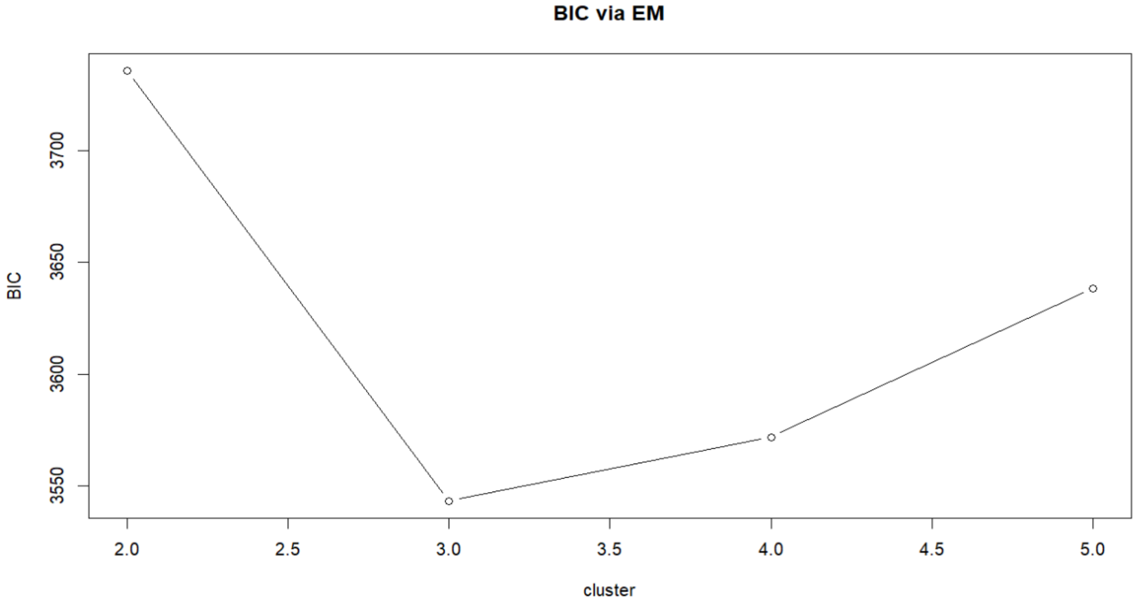
$$p_k = \frac{1}{n} \sum_{i=1}^{n} \hat{r}_{ik}, k = 1, 2, ..., K.$$

4. Repeat 2 and 3 until $\sum_{j=1}^{K} ||\mu_j^{(t+1)} - \mu_j^{(t)}|| < 10^{-6}$ and $\sum_{k=1}^{K} ||p_k^{(t+1)} - p_k^{(t)}|| < 10^{-9}$.

First,the parameters for clusters $k = \{2, 3, 4, 5\}$ are estimated by the EM algorithm ,and the appropriate k is determined via comparing its BIC.

The following is the line chart of BIC:

(The R cdoe for line chart is attached to the appendix)



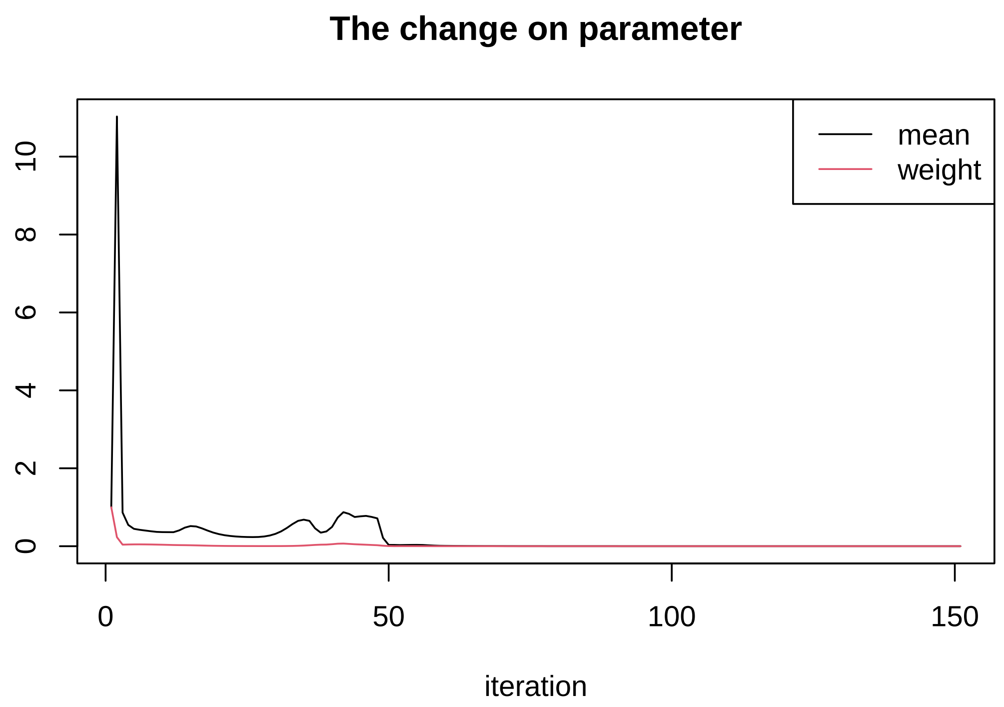
From the above,I choose k = 3.

Next,print the EM result for k=3:

```
fmul <- function(mu,sigma2,xvec){
  det <- det(sigma2)
  d = length(xvec)
  maha <- (xvec-mu) %*% solve(sigma2) %*% t(xvec-mu)
  ((2*pi)^d*det)^(-1/2)*exp(-maha/2)
}
X <- as.matrix(data2)
mu1 <- mu1.all<-t(c(18,35,18))
mu2 <-mu2.all<-t(c(21,40,15))
mu3 <-mu3.all<-t(c(20,50,18))

sigma2.1 <- sigma2.2<-sigma2.3<-diag(c(1,1,1))
error <-err.all<- c(1,1)
p.all <- p <- t(rep(1/3,3))
n= dim(data2)[1]
count = 1
while(error[1]>=10^(-6) | error[2]>=10^(-9)){
  # E-step
  r1<-r2<-r3 <- numeric(n) #membership
```

```r
  for(i in 1:n){
    tot <- p[1]*fmul(mu1,sigma2.1,X[i,])+p[2]*fmul(mu2,sigma2.2,X[i,])+
      p[3]*fmul(mu3,sigma2.3,X[i,])
    r1[i] <- p[1]*fmul(mu1,sigma2.1,X[i,])/tot
    r2[i] <- p[2]*fmul(mu2,sigma2.2,X[i,])/tot
    r3[i] <- p[3]*fmul(mu3,sigma2.3,X[i,])/tot
  }
  # M-step
  mu1.t <- (r1%*%X)/sum(r1) ; mu2.t <- (r2%*%X)/sum(r2) ;  mu3.t <- (r3%*%X)/sum(r3)
  sigma21.t <- t(X- rep(1,n) %*% mu1.t) %*% diag(r1) %*% (X- rep(1,n) %*% mu1.t)/sum(r1)
  sigma22.t <- t(X- rep(1,n) %*% mu2.t) %*% diag(r2) %*% (X- rep(1,n) %*% mu2.t)/sum(r2)
  sigma23.t <- t(X- rep(1,n) %*% mu3.t) %*% diag(r3) %*% (X- rep(1,n) %*% mu3.t)/sum(r3)
  p1 <- sum(r1)/n;p2 <- sum(r2)/n;p3 <- sum(r3)/n
  # stop criterion
  error[1] <- sum(abs(mu1-mu1.t)) + sum(abs(mu2-mu2.t)) + sum(abs(mu3-mu3.t))
  error[2] <- abs(p1-p[1]) + abs(p2-p[2])+abs(p3-p[3])
  # update
  mu1.all <-rbind(mu1.all,mu1.t)
  mu2.all <-rbind(mu2.all,mu2.t)
  mu3.all <-rbind(mu3.all,mu3.t)
  p.all <-rbind(p.all,cbind(p1,p2,p3))
  mu1 <- mu1.t ; mu2 <-mu2.t ; mu3 <- mu3.t
  sigma2.1 <- sigma21.t
  sigma2.2 <- sigma22.t
  sigma2.3 <- sigma23.t
  p <- cbind(p1,p2,p3)
  err.all <- rbind(err.all,error)
  count=count+1
}
```

```
The iteration : 151
```



From the above, we know that the EM algorithm for k =3 is successfully converged.

Summary,the estimated components:

* μ_1, μ_2 amd μ_3:

```
mumu <- data.frame(rbind(mu1.all[count,],mu2.all[count,],mu3.all[count,]))
colnames(mumu) <- names(data2)
rownames(mumu) <- c("$\\mu_1$","$\\mu_2$","$\\mu_3$")
knitr::kable(mumu,row.names = TRUE)
```

	X1	X2	X3
μ_1	18.98691	39.09246	18.32533
μ_2	21.73974	47.52369	15.01281
μ_3	19.70473	49.14284	18.52037

- Σ_1:

```
sigma2.1
```

```
          X1        X2         X3
X1 0.4238396 0.438845 0.2461799
X2 0.4388450 7.462674 1.0446056
X3 0.2461799 1.044606 1.4499908
```

- Σ_2:

```
sigma2.2
```

```
          X1        X2         X3
X1 0.4557461 1.443737 0.4853381
X2 1.4437374 9.925157 2.0680982
X3 0.4853381 2.068098 0.9738259
```

- Σ_3:

```
sigma2.3
```

```
          X1         X2         X3
X1 0.4896974 0.5909721 0.4143226
X2 0.5909721 9.6123852 2.1025512
X3 0.4143226 2.1025512 1.1883874
```

- The Weight p:

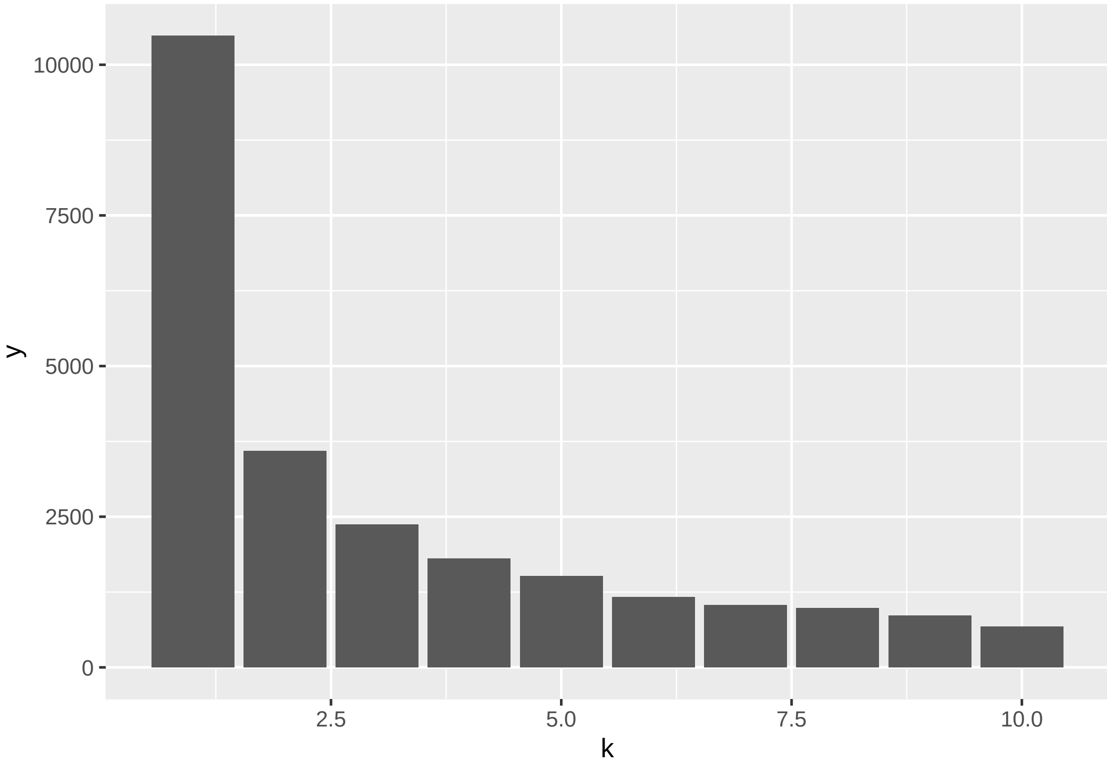
```
cat("1 cluster:",p.all[count,1],"\n");cat("2 cluster:",p.all[count,2],"\n");cat("3 cluster:",p.all
```

```
1 cluster: 0.4490266

2 cluster: 0.3598301

3 cluster: 0.1911433
```

(b)

Using Elbow method for selecting the appropriate k,

```
wss <- c()
wss[1] <- sum(scale(data2,scale = FALSE)^2)
for(i in 2:10){
  km = kmeans(data2,centers = i)
  wss[i] = sum(km$withinss)
}
library(ggplot2)
wss.data <- data.frame("k"=1:10,"y"=wss)
ggplot(wss.data,aes(x=k,y=y)) + geom_col()
```



Since there is little change after k=3, I choose k=3.

Next,estimate the paramter of 3-mixture model via MLE:

$$\hat{\mu}_k = \frac{\sum_{i=1}^n p_{i,k} x_i}{n_k} \in \mathbb{R}^{3x1}; \hat{\Sigma_k} = \frac{\sum_{i=1}^n p_{i,k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{n_k} \in \mathbb{R}^{3x3},$$

$$\text{where } p_{i,k} = I(x_i \text{ is grouped as k}), n_k = \sum_{i=1}^n p_{i,k} \ .$$

```
kmean <- kmeans(data2,3)
n1 <- as.numeric(kmean$cluster ==1)
n2 <- as.numeric(kmean$cluster ==2)
n3 <- as.numeric(kmean$cluster ==3)
mu1.hat <- n1 %*% X / sum(n1)
mu2.hat <- n2 %*% X / sum(n2)
mu3.hat <- n3 %*% X / sum(n3)
sigma21.hat <- t(X- rep(1,n) %*% mu1.hat) %*% diag(n1) %*% (X- rep(1,n) %*% mu1.hat)/sum(n1)
sigma22.hat <- t(X- rep(1,n) %*% mu2.hat) %*% diag(n2) %*% (X- rep(1,n) %*% mu2.hat)/sum(n2)
sigma23.hat <- t(X- rep(1,n) %*% mu3.hat) %*% diag(n3) %*% (X- rep(1,n) %*% mu3.hat)/sum(n3)
```

- μ_1, μ_2 amd μ_3:

```
mumu.hat <- data.frame(rbind(mu1.hat,mu2.hat,mu3.hat))
colnames(mumu.hat) <- names(data2)
rownames(mumu.hat) <- c("$\\mu_1$","$\\mu_2$","$\\mu_3$")
knitr::kable(mumu.hat,row.names = TRUE)
```

	X1	X2	X3
μ_1	20.69510	45.49804	15.68529
μ_2	18.95410	38.56639	18.28607
μ_3	21.19868	50.89868	17.37368

- Σ_1:

```
sigma21.hat
```

```
           X1         X2        X3
X1  1.1787015  0.2995982 -1.441151
X2  0.2995982  3.2268589 -0.105323
X3 -1.4411505 -0.1053230  3.468705
```

- Σ_2:

```
sigma22.hat
```

```
          X1        X2        X3
X1 0.4310078 0.3144410 0.2432948
X2 0.3144410 5.1268214 0.8393678
X3 0.2432948 0.8393678 1.4289042
```

- Σ_3:

```
sigma23.hat
```

```
            X1          X2        X3
X1  1.83697195 -0.04802805 -1.709377
X2 -0.04802805  4.29934037  0.877597
X3 -1.70937673  0.87759695  3.252729
```

- The Weight p:

```
cat("1 cluster:",kmean$size[1]/300,"\n");cat("2 cluster:",kmean$size[2]/300,"\n");cat("3 cluster:'
```

```
1 cluster: 0.34

2 cluster: 0.4066667

3 cluster: 0.2533333
```

The above results are obtained using MLE, which has many similarities with the results obtained by EM. The estimated performance of the EM algorithm is quite good.

Appendix(2.(b) BIC plot)

```
## k=2
X <- as.matrix(data2)
fmul <- function(mu,sigma2,xvec){
  det <- det(sigma2)
  d = length(xvec)
  mu <- as.matrix(mu)
  maha <- (xvec-mu) %*% solve(sigma2) %*% t(xvec-mu)
  ((2*pi)^d*det)^(-1/2)*exp(-maha/2)
}
mu <- data.frame(kmeans(data2,2)$center)
sigma2 <- replicate(2,diag(c(1,1,1)))
error <-err.all<- c(1,1)
p.all <- p <- t(rep(1/2,2))
n= dim(data2)[1]
count = 1
while(error[1]>=10^(-6) | error[2]>=10^(-9)){
  # E-step
  r <- data.frame(0,0) #membership
  for(i in 1:n){
    tot <- p[1]*fmul(mu[1,],sigma2[,,1],X[i,])+p[2]*fmul(mu[2,],sigma2[,,2],X[i,])
    r[i,1] <- p[1]*fmul(mu[1,],sigma2[,,1],X[i,])/tot
    r[i,2] <- p[2]*fmul(mu[2,],sigma2[,,2],X[i,])/tot
  }
  # M-step
  mu1.t <- (r[,1]%*%X)/sum(r[,1]) ; mu2.t <- (r[,2]%*%X)/sum(r[,2])
  sigma21.t <- t(X- rep(1,n) %*% mu1.t) %*% diag(r[,1]) %*% (X- rep(1,n) %*% mu1.t)/sum(r[,1])
  sigma22.t <- t(X- rep(1,n) %*% mu2.t) %*% diag(r[,2]) %*% (X- rep(1,n) %*% mu2.t)/sum(r[,2])
  p1 <- sum(r[,1])/n;p2 <- sum(r[,2])/n
  # stop criterion
  error[1] <- sum(abs(mu[1,]-mu1.t)+abs(mu[2,]-mu2.t))
  error[2] <- abs(p1-p[1]) + abs(p2-p[2])
  # update
  mu[1,] <- mu1.t ; mu[2,] <-mu2.t
  sigma2[,,1] <- sigma21.t
  sigma2[,,2] <- sigma22.t
  p <- cbind(p1,p2)
  err.all <- rbind(err.all,error)
  count=count+1
```

```
}
s2 <- 0
for (i in 1:n){
  L = p[1]*fmul(mu[1,],sigma2[,,1],X[i,]) + p[2]*fmul(mu[2,],sigma2[,,2],X[i,])
  s2 = s2+ log(L)
}
bic2 <- -2*s2 + log(n)*(10*2-1)

## k=3
mu <- data.frame(kmeans(data2,3)$center)
sigma2 <- replicate(3,diag(c(1,1,1)))
error <-err.all<- c(1,1)
p.all <- p <- t(rep(1/3,3))
n= dim(data2)[1]
count = 1
while(error[1]>=10^(-6) | error[2]>=10^(-9)){
  # E-step
  r <- data.frame(0,0,0) #membership
  for(i in 1:n){
    tot <- p[1]*fmul(mu[1,],sigma2[,,1],X[i,])+p[2]*fmul(mu[2,],sigma2[,,2],X[i,])+
      p[3]*fmul(mu[3,],sigma2[,,3],X[i,])
    r[i,1] <- p[1]*fmul(mu[1,],sigma2[,,1],X[i,])/tot
    r[i,2] <- p[2]*fmul(mu[2,],sigma2[,,2],X[i,])/tot
    r[i,3] <- p[3]*fmul(mu[3,],sigma2[,,3],X[i,])/tot
  }
  # M-step
  mu1.t <- (r[,1]%*%X)/sum(r[,1]) ; mu2.t <- (r[,2]%*%X)/sum(r[,2]);mu3.t <- (r[,3]%*%X)/sum(r[,3]
  sigma21.t <- t(X- rep(1,n) %*% mu1.t) %*% diag(r[,1]) %*% (X- rep(1,n) %*% mu1.t)/sum(r[,1])
  sigma22.t <- t(X- rep(1,n) %*% mu2.t) %*% diag(r[,2]) %*% (X- rep(1,n) %*% mu2.t)/sum(r[,2])
  sigma23.t <- t(X- rep(1,n) %*% mu3.t) %*% diag(r[,3]) %*% (X- rep(1,n) %*% mu3.t)/sum(r[,3])
  p1 <- sum(r[,1])/n;p2 <- sum(r[,2])/n;p3 <- sum(r[,3])/n
  # stop criterion
  error[1] <- sum(abs(mu[1,]-mu1.t)+abs(mu[2,]-mu2.t)+abs(mu[3,]-mu3.t))
  error[2] <- abs(p1-p[1]) + abs(p2-p[2]) + abs(p3-p[3])
  # update
  mu[1,] <- mu1.t ; mu[2,] <-mu2.t ; mu[3,] <-mu3.t
  sigma2[,,1] <- sigma21.t
  sigma2[,,2] <- sigma22.t
  sigma2[,,3] <- sigma23.t
  p <- cbind(p1,p2,p3)
```

```
  err.all <- rbind(err.all,error)
  count=count+1
}
s3 <- 0
for (i in 1:n){
  L = p[1]*fmul(mu[1,],sigma2[,,1],X[i,]) + p[2]*fmul(mu[2,],sigma2[,,2],X[i,])+
    p[3]*fmul(mu[3,],sigma2[,,3],X[i,])
  s3 = s3+ log(L)
}
bic3 <- -2*s3 + log(n)*(10*3-1)

## k=4
mu <- data.frame(kmeans(data2,4)$center)
sigma2 <- replicate(4,diag(c(1,1,1)))
error <-err.all<- c(1,1)
p.all <- p <- t(rep(1/4,4))
n= dim(data2)[1]
count = 1
while(error[1]>=10^(-6) | error[2]>=10^(-9)){
  # E-step
  r <- data.frame(0,0,0,0) #membership
  for(i in 1:n){
    tot <- p[1]*fmul(mu[1,],sigma2[,,1],X[i,])+p[2]*fmul(mu[2,],sigma2[,,2],X[i,])+
      p[3]*fmul(mu[3,],sigma2[,,3],X[i,]) + p[4]*fmul(mu[4,],sigma2[,,4],X[i,])
    r[i,1] <- p[1]*fmul(mu[1,],sigma2[,,1],X[i,])/tot
    r[i,2] <- p[2]*fmul(mu[2,],sigma2[,,2],X[i,])/tot
    r[i,3] <- p[3]*fmul(mu[3,],sigma2[,,3],X[i,])/tot
    r[i,4] <- p[4]*fmul(mu[4,],sigma2[,,4],X[i,])/tot
  }
  # M-step
  mu1.t <- (r[,1]%*%X)/sum(r[,1]) ; mu2.t <- (r[,2]%*%X)/sum(r[,2]);mu3.t <- (r[,3]%*%X)/sum(r[,3]
  mu4.t <- (r[,4]%*%X)/sum(r[,4])
  sigma21.t <- t(X- rep(1,n) %*% mu1.t) %*% diag(r[,1]) %*% (X- rep(1,n) %*% mu1.t)/sum(r[,1])
  sigma22.t <- t(X- rep(1,n) %*% mu2.t) %*% diag(r[,2]) %*% (X- rep(1,n) %*% mu2.t)/sum(r[,2])
  sigma23.t <- t(X- rep(1,n) %*% mu3.t) %*% diag(r[,3]) %*% (X- rep(1,n) %*% mu3.t)/sum(r[,3])
  sigma24.t <- t(X- rep(1,n) %*% mu4.t) %*% diag(r[,4]) %*% (X- rep(1,n) %*% mu4.t)/sum(r[,4])
  p1 <- sum(r[,1])/n;p2 <- sum(r[,2])/n;p3 <- sum(r[,3])/n;p4 <- sum(r[,4])/n
  # stop criterion
  error[1] <- sum(abs(mu[1,]-mu1.t)+abs(mu[2,]-mu2.t)+abs(mu[3,]-mu3.t)+abs(mu[4,]-mu4.t))
  error[2] <- abs(p1-p[1]) + abs(p2-p[2]) + abs(p3-p[3])
```

```r
  # update
  mu[1,] <- mu1.t ; mu[2,] <-mu2.t ; mu[3,] <-mu3.t ; mu[4,] <-mu4.t
  sigma2[,,1] <- sigma21.t
  sigma2[,,2] <- sigma22.t
  sigma2[,,3] <- sigma23.t
  sigma2[,,4] <- sigma24.t
  p <- cbind(p1,p2,p3,p4)
  err.all <- rbind(err.all,error)
  count=count+1
}
s4 <- 0
for (i in 1:n){
  L = p[1]*fmul(mu[1,],sigma2[,,1],X[i,]) + p[2]*fmul(mu[2,],sigma2[,,2],X[i,])+
    p[3]*fmul(mu[3,],sigma2[,,3],X[i,]) + p[4]*fmul(mu[4,],sigma2[,,4],X[i,])
  s4 = s4+ log(L)
}
bic4 <- -2*s4 + log(n)*(10*4-1)
## k=5
mu <- data.frame(kmeans(data2,5)$center)
sigma2 <- replicate(5,diag(c(1,1,1)))
error <-err.all<- c(1,1)
p.all <- p <- t(rep(1/5,5))
n= dim(data2)[1]
count = 1
while(error[1]>=10^(-6) | error[2]>=10^(-9)){
  # E-step
  r <- data.frame(0,0,0,0,0) #membership
  for(i in 1:n){
    tot <- p[1]*fmul(mu[1,],sigma2[,,1],X[i,])+p[2]*fmul(mu[2,],sigma2[,,2],X[i,])+
      p[3]*fmul(mu[3,],sigma2[,,3],X[i,]) + p[4]*fmul(mu[4,],sigma2[,,4],X[i,]) +
      p[5]*fmul(mu[5,],sigma2[,,5],X[i,])
    r[i,1] <- p[1]*fmul(mu[1,],sigma2[,,1],X[i,])/tot
    r[i,2] <- p[2]*fmul(mu[2,],sigma2[,,2],X[i,])/tot
    r[i,3] <- p[3]*fmul(mu[3,],sigma2[,,3],X[i,])/tot
    r[i,4] <- p[4]*fmul(mu[4,],sigma2[,,4],X[i,])/tot
    r[i,5] <- p[5]*fmul(mu[5,],sigma2[,,5],X[i,])/tot
  }
  # M-step
  mu1.t <- (r[,1]%*%X)/sum(r[,1]) ; mu2.t <- (r[,2]%*%X)/sum(r[,2]);mu3.t <- (r[,3]%*%X)/sum(r[,3]
  mu4.t <- (r[,4]%*%X)/sum(r[,4]) ; mu5.t <- (r[,5]%*%X)/sum(r[,5])
```

```
  sigma21.t <- t(X- rep(1,n) %*% mu1.t) %*% diag(r[,1]) %*% (X- rep(1,n) %*% mu1.t)/sum(r[,1])
  sigma22.t <- t(X- rep(1,n) %*% mu2.t) %*% diag(r[,2]) %*% (X- rep(1,n) %*% mu2.t)/sum(r[,2])
  sigma23.t <- t(X- rep(1,n) %*% mu3.t) %*% diag(r[,3]) %*% (X- rep(1,n) %*% mu3.t)/sum(r[,3])
  sigma24.t <- t(X- rep(1,n) %*% mu4.t) %*% diag(r[,4]) %*% (X- rep(1,n) %*% mu4.t)/sum(r[,4])
  sigma25.t <- t(X- rep(1,n) %*% mu5.t) %*% diag(r[,4]) %*% (X- rep(1,n) %*% mu5.t)/sum(r[,5])
  p1 <- sum(r[,1])/n;p2 <- sum(r[,2])/n;p3 <- sum(r[,3])/n;p4 <- sum(r[,4])/n
  p5 <- sum(r[,5])/n
  # stop criterion
  error[1] <- sum(abs(mu[1,]-mu1.t)+abs(mu[2,]-mu2.t)+abs(mu[3,]-mu3.t)+abs(mu[4,]-mu4.t)+
                  abs(mu[5,]-mu5.t))
  error[2] <- abs(p1-p[1]) + abs(p2-p[2]) + abs(p3-p[3]) + abs(p4-p[4])
  # update
  mu[1,] <- mu1.t ; mu[2,] <-mu2.t ; mu[3,] <-mu3.t ; mu[4,] <-mu4.t
  mu[5,] <- mu5.t
  sigma2[,,1] <- sigma21.t
  sigma2[,,2] <- sigma22.t
  sigma2[,,3] <- sigma23.t
  sigma2[,,4] <- sigma24.t
  sigma2[,,5] <- sigma25.t
  p <- cbind(p1,p2,p3,p4,p5)
  err.all <- rbind(err.all,error)
  count=count+1
}
s5 <- 0
for (i in 1:n){
  L = p[1]*fmul(mu[1,],sigma2[,,1],X[i,]) + p[2]*fmul(mu[2,],sigma2[,,2],X[i,])+
    p[3]*fmul(mu[3,],sigma2[,,3],X[i,]) + p[4]*fmul(mu[4,],sigma2[,,4],X[i,]) +
    p[5]*fmul(mu[5,],sigma2[,,5],X[i,])
  s5 = s5+ log(L)
}
bic5 <- -2*s5 + log(n)*(10*5-1)
bic5
all.bic <- c(bic2,bic3,bic4,bic5)
plot(x=2:5,y=all.bic,type="b",xlab="cluster",ylab="BIC",
     main = "BIC via EM")
```