Project 2

Kehinde Fagbamigbe

2022-09-18

The package for multivariate distribution is the dmvnorm which is in the mvtnorm package in R

```
library(mvtnorm)
```

Installing the mvtnorm library

Importing the dataser

Data

```
x <- read.table("/Users/alhajidot/Documents/BGSU/Project/gaussian.txt", quote="\"", comment.char="")
head(x)
```

```
##          V1        V2         V3        V4
## 1 0.6920919 0.8901322 0.05163798 0.6420625
## 2 0.8191378 0.1139812 0.59344880 0.0961274
## 3 0.8759624 0.9787950 0.58887500 0.6170371
## 4 0.7309516 0.6469564 0.55979040 0.3477307
## 5 0.6785288 1.4131210 0.26681430 0.9525697
## 6 0.7595809 0.9434503 0.33931530 0.7332092
```

Goal

Multivariate Gaussian Mixture

$$\mathcal{N}(x_i|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{(p/2)}|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}\left(x_i-\mu_k\right)^{'}\Sigma_k^{-1}\left(x_i-\mu_k\right)\right\}$$

Preliminary

$$g(x_i, \vartheta) = \sum_{k=1}^{k} \tau_k f_k(x_i, \theta_k)$$

where

$$g(x_i, \vartheta)\ is\ a\ pdf$$
$$\tau_k\ is\ the\ mixing\ proportion\ or\ the\ weights$$
$$x_i\ is\ the\ mixing\ component$$
$$\theta_k\ is\ the\ parameter$$

- The mixture component knows the cluster by taking a weighted sum of several probalantity distritutions and it assumes each clusters in your data corresponds to one mixtire component.

- The best no of cluster is chosen based on the model with the best Bayesian Information Criterion also known as penalized Log Likelihood.
- In the Expectation Maximization (EM) Algorithm, The E-step gives us the posterior Probability, which is the probability that the ith observation is coming fro the kth component.
- The M-step, gives us the maximum likelihood estimate of the mixture model Parameters.
- The joint probabbility aka likelihood is the product of each prior probability e.g P(A) * P(B). For an independent and identically distributed (iid) variable xi, the joint probability is the product of the pdf

$$\prod_{i=1}^{n} g(x_i, \vartheta) = L(\vartheta)$$

$$where\ (\vartheta)\ is\ the\ Likelihood$$

$$Recall\ g(x_i, \vartheta) = \sum_{k=1}^{k} \tau_k f_k(x_i, \theta_k)$$

$$Therefore : L(\vartheta) = \prod_{i=1}^{n} g(x_i, \vartheta) = \pi_{i=1}^{n} \sum_{k=1}^{k} \tau_k f_k(x_i, \theta_k)$$

Taking the Logarithm of the Likelihood(called Log Likelihood)

$$LogL(\vartheta) = \pi_{i=1}^{n} g(x_i, \vartheta) = Log(\pi_{i=1}^{n} \sum_{k=1}^{k} \tau_k f_k(x_i, \theta_k))$$

$$From\ Calculus : Log \prod = \sum$$

$$LogL(\vartheta) = \sum_{k=1}^{n}$$

$$Log(\sum_{k=1}^{k} \tau_k f_k(x_i, \theta_k))$$

$$From\ Calculus :$$

$$Log(mn) = Logm + Logn$$

$$Log(m+n)\ ! = Logm + Logn$$

To solve further, we need to find the complete Log Likelihood to work around the logarithm roadblock

Assumptions of Complete Log Likelihood

- We assume we know the missing labels of the clusters
- The missing labels will denoted as Zi

$$if\ k\ is\ the\ number\ of\ clusters:$$

$$Zi \in \{1, 2, 3, ..., k\}$$

$$each\ of\ the\ observations\ x_i\ to\ x_n\ belongs\ to\ one\ component\ Z_i\ to\ Z_k$$

$$i.e\ x_i\ can\ only\ come\ from\ one\ component\ Z_i This\ makes\ us\ have\ the\ complete\ Log\ likelihood\ denoted\ as\ LogL_c$$

$$g(x_i, \vartheta) = \sum_{k=1}^{k} \tau_k f_k(x_i, \theta_k) = \prod_{k=1}^{k} (\tau_k f_k(x_i, \theta_k))^{I(Z_i=k)}$$

$$since\ \sum_{k=1}^{k} \tau_k f_k(x_i, \theta_k) = \prod_{k=1}^{k} (\tau_k f_k(x_i, \theta_k))^{I(Z_i=k)}$$

$$LogL_c(\vartheta) = \sum_{k=1}^{n} Log(\sum_{k=1}^{k} \tau_k f_k(x_i, \theta_k)) = \sum_{k=1}^{n} Log(\prod_{k=1}^{k} (\tau_k f_k(x_i, \theta_k))^{I(Z_i=k)})$$

$$Recall\ Log \prod = \sum$$

$$LogL_c(\vartheta) = \sum_{i=1}^{n} \sum_{k=1}^{k} I(Z_i = k) Log(\tau_k f_k(x_i, \theta_k))$$

Expectation Maximization Algorithm method (EM)

- This consist of two steps The Expectation
  - Expectation Step (E-Step)
  - Maximization (M-Step)
- E-step: The expectaton of the complete Log likelihood given x. this is also called the Q-function.$$

$$Q = (\vartheta|\vartheta^{B-1}, x_i, x_2, \dots, x_n) = \sum_{i=1}^{n} \sum_{k=1}^{k} E[I(Z_i = k)|x_i] Log(\tau_k f_k(x_i, \theta_k))$$

$$Using\ Bayes\ theorem$$

$$P(A|B) = \frac{(A \cap B)}{P(B)}$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$Therefore: E[I(Z_i = k)|x_i] = \frac{P(x_i|Z_i = k)P(Z = k)}{\sum_{k=1}^{k} P(Z_i = k')P(x_i|Z_i = k')}$$

$$E[I(Z_i = k)|x_i] = \frac{\tau_k^{(b_i-1)} f_k(x_i, \theta_k^{(b_i-1)}}{\sum_{i=1}^{k} \tau_k^{(b_i-1)} f_k(x_i, \theta_k^{(b_i-1)}}) = \pi_{ik}^{(b)}$$

$$\pi_{ik}^{(b)} => This\ is\ the\ Posterior\ Probability$$

$$Substituting\ the\ expectation\ calculated\ in\ the\ Q-function$$

$$Q = (\vartheta|\vartheta^{B-1}, x_i, x_2, ..., x_n) = \sum_{i=1}^{n} \sum_{k=1}^{k} E[I(Z_i = k)|x_i] Log(\tau_k f_k(x_i, \theta_k))$$

$$Recall : Log(mn) = Logm + Logn$$

$$Q = (\vartheta|\vartheta^{B-1}, x_i, x_2, ..., x_n) = \sum_{i=1}^{n} \sum_{k=1}^{k} E[I(Z_i = k)|x_i] Log(\tau_k f_k(x_i, \theta_k)) = \sum_{i=1}^{n} \sum_{k=1}^{k} E[I(Z_i = k)|x_i](Log(\tau_k) + Log f_k(x_i, \theta_k))$$

$$Q = (\vartheta|\vartheta^{B-1}, x_i, x_2, ..., x_n) = \sum_{i=1}^{n} \sum_{k=1}^{k} \pi_{ik}^{(b)} (Log(\tau_k) + Log f_k(x_i, \theta_k))$$

$$Differentiating\ to\ get\ the\ parameters\ of\ the\ distribution$$

$$\frac{\partial Q}{\partial \tau_k} = \frac{\partial}{\partial \tau_k} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{k} \pi_{ik}^{(b)} Log(\tau_k) \right\}$$

$$Recall\ we\ have\ a\ constraint : \sum_{k=1}^{k} \tau_k = 1$$

$$Using\ Lagrange\ multiplier(\lambda)$$

$$(\lambda) : \frac{\partial Q}{\partial \tau_k} = \frac{\partial}{\partial \tau_k} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{k} \pi_{ik}^{(b)} Log(\tau_k) - \lambda(\sum_{k=1}^{k} \tau_k - 1) \right\}$$

$$\frac{\partial Q}{\partial \tau_k} = \frac{\partial}{\partial \tau_k} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{k} \pi_{ik}^{(b)} Log(\tau_k) - \lambda(\sum_{k=1}^{k} \tau_k - 1) \right\} = \frac{\partial}{\partial \tau_k} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{k} \pi_{ik}^{(b)} Log(\tau_k) - \lambda \sum_{k=1}^{k} \tau_k + \lambda \right\}$$

$$\frac{\partial Q}{\partial \tau_k} = \frac{\partial}{\partial \tau_k} \left\{ \sum_{i=1}^{n} \pi_{ik}^{(b)} \frac{1}{\tau_k} - \lambda \right\}$$

$$Equating \frac{\partial Q}{\partial \tau_k} = 0$$

$$\frac{\partial Q}{\partial \tau_k} = \frac{\partial}{\partial \tau_k} \left\{ \sum_{i=1}^{n} \pi_{ik}^{(b)} \frac{1}{\tau_k} \right\} - \lambda = 0$$

$$\left\{ \sum_{i=1}^{n} \pi_{ik}^{(b)} \frac{1}{\tau_k} \right\} - \lambda = 0 \left\{ \frac{1}{\tau_k} \sum_{i=1}^{n} \pi_{ik}^{(b)} \right\} = \lambda$$

$$\tau_k = \left\{ \frac{1}{\lambda} \sum_{i=1}^{n} \pi_{ik}^{(b)} \right\}$$

$$\sum_{k=1}^{k} \tau_k = 1$$

$$Therefore\ \sum_{k=1}^{k} \tau_k = \left\{ \sum_{k=1}^{k} \frac{1}{\lambda} \sum_{i=1}^{n} \pi_{ik}^{(b)} = 1 \right\}$$

$$\frac{1}{\lambda} \sum_{i=1}^{n} \sum_{k=1}^{k} \pi_{ik}^{(b)} = 1$$

$$\sum_{k=1}^{k} \pi_{ik}^{(b)} = 1$$

$$\frac{1}{\lambda} \sum_{i=1}^{n} \sum_{k=1}^{k} \pi_{ik}^{(b)} = 1$$

$$\frac{1}{\lambda} \sum_{i=1}^{n} 1 = 1$$

$$\lambda = \sum_{i=1}^{n} 1 = 1$$

$$\lambda = n$$

$$Recall, \tau_k = \left\{ \frac{1}{\lambda} \sum_{i=1}^{n} \pi_{ik}^{(b)} \right\}$$

$$Therefore, \tau_k = \left\{ \frac{1}{n} \sum_{i=1}^{n} \pi_{ik}^{(b)} \right\}$$

Multivariate Normal Distribution

$$X \sim \mathcal{N}(\mu, \Sigma) = \frac{1}{(2\pi)^{(p/2)} |\Sigma_k|^{1/2}} \exp \left\{ -\frac{1}{2} \left(x_i - \mu_k\right)^T \Sigma_k^{-1} \left(x_i - \mu_k\right) \right\}$$

Log Likelihood

$$LogL_c = \sum_{i=1}^{n} \sum_{i=1}^{k} I(Z_i = k) Log(\tau_k f_k(x_i, \theta_k))$$

$$E- \ Step$$

Posterior Probability

$$\pi_{ik}^{b} = \frac{\tau_k^{(b_i-1)} \phi(\mu_k^{b-1}, \sigma_k^2)}{\sum_{i=1}^{k} \tau_k^{(b_i-1)} \phi(\mu_k^{b-1}, \sigma_k^2)}$$

$$cov_{x,y} = \frac{\sum_{i=1}^{n} (x_i - \overline{x}) \cdot (y_i - \overline{y})}{n-1}$$

The E- step returns the posterior Probability, which is the probability that the ith observation is coming fro the kth component

- The dataframe x is converted to a matrix and the dimensions are noted. n = number of rows or observations, p = number of columns

```
E.step <- function(x, tau, Mu, covariance){  #tau is mixture proportion, Mu is mean S2 is Covariance

 x <- as.matrix(x)
 n <- dim(x)[1]
 K <- length(tau)
 p <- dim(x)[2]
 Pi <- matrix(NA, n, K)

 for (i in 1:n){

  for (k in 1:K){
   Pi[i,k] <- tau[k] * mvtnorm::dmvnorm(t(x[i,]), mean = (Mu[,k]), sigma = covariance[, , k])  #dnorm m

   }
  Pi[i,] <- Pi[i,] / sum(Pi[i,])
  # cat("E_STEP", Pi[i,])
 }
return(Pi)

}
```

$$M - Step$$

Weights

$$\tau_k^{(b)} = \frac{1}{n} \sum_{i=1}^{n} \pi_{ik}^{b}$$

Mean

$$\mu_k^{b-1} = \frac{\sum_{i=1}^{n} \pi_{ik}^{b} \cdot x_i}{\sum_{i=1}^{n} \pi_{ik}^{(b)}}$$

Covariance

$$\sum_{P*P} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,p} \\ a_{1,2} & a_{2,2} & \cdots & a_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,p} & a_{2,p} & \cdots & a_{p,p} \end{pmatrix}$$

$$\Sigma_k^{(b)} = \frac{\sum_{i=1}^{n} \pi_{ik}^{b} \cdot (x_i - \mu_k^b) \cdot (x_i - \mu_k^b)^T}{\sum_{i=1}^{n} \pi_{ik}^{(b)}}$$

```
M.step <- function(x, Pi){
 x <- as.matrix(x)
 K <- dim(Pi)[2]
 n <- dim(x)[1]
 p <- dim(x)[2]
 Mu = matrix(0, nrow = p, ncol = K)
 covariance = array(0, dim = c(p,p,K))

 Sum.Pi <- apply(Pi, 2, sum)
 tau <- Sum.Pi / n


 for (k in 1:K){

  for (i in 1:n){
   Mu[,k] <- Mu[,k] + Pi[i,k] %*% t(x[i,])
  }
  Mu[,k] <- Mu[,k] / Sum.Pi[k]

  for (i in 1:n){
   covariance[, , k] <- covariance[, , k] + (Pi[i,k] *  ( as.matrix(x[i,] - Mu[,k])) %*%  t(as.matrix(x
  }
  covariance[, , k] <- covariance[, , k] / Sum.Pi[k]

 }

 return(list(tau = tau, Mu = Mu, covariance = covariance))

}
```

$$Complete - Logarithm\ Likelihood$$

```
logL <- function(x, tau, Mu, covariance){

 x <- as.matrix(x)
 n <- dim(x)[1]
 K <- length(tau)

 ll <- 0
```

```
 for (i in 1:n){

  ll2 <- 0

  for (k in 1:K){
   ll2 <- ll2 + tau[k] * mvtnorm::dmvnorm(t(x[i,]), mean = (Mu[,k]), sigma = covariance[, , k])
  }

  ll <- ll + log(ll2)
 }

 return(ll)

}
```

Expectation − Maximization Algorithm

```
EM <- function(x, tau, Mu, covariance, eps){

 x <- as.matrix(x)
# cat("I am here 0", "\n")
 K <- length(tau)
 n <- dim(x)[1]
 p <- dim(x)[2]
# cat("I am here 1", "\n")
 b <- 0

 # cat("Iteration",b , "tau =", tau, "\n")
 # cat("Iteration",b , "Mu =", Mu, "\n")
 # cat("Iteration",b , "Cov =", covariance, "\n")
 ll.old <- -Inf
 ll <- logL(x, tau, Mu, covariance)
 # cat("Iteration",b , "logL =", ll, "\n")
 # cat("I am here 2", "\n")
 repeat{

  b <- b + 1

  if ((ll - ll.old) / abs(ll) < eps) break
# cat("I am here 3", "\n")
  ll.old <- ll
  # cat("ll.old", ll.old)
  Pi <- E.step(x, tau, Mu, covariance)
  # cat("I am here 4", "\n")
  # cat("Pi", Pi)
  M <- M.step(x,Pi)
  # cat("I am here 5", "\n")

  tau <- M$tau
  Mu <- M$Mu
  covariance <- M$covariance
```

```
  ll <- logL(x, tau, Mu, covariance)
  cat("Iteration", b, "logL =", ll, "\n")
 }

 id <- apply(Pi, 1, which.max)

 M <- 3 * K - 1
 BIC <- -2 * ll + M * log(n)
 AIC <- -2 * ll + M * 2

 return(list(tau = tau, Mu = Mu, covariance = covariance, logL = ll, BIC = BIC, Pi = Pi , id = id, AIC =

}
```

Parameters

```
#Tau
t_test = c(0.3,0.2,0.5)
```

```
# Mean
m1 = c(3,1,4,7)
m2 = c(1,1,6,2)
m3 = c(3,5,7,6)

mu_test = matrix(c(m1,m2,m3), nrow = 4, ncol = 3 )
mu_test
```

```
##      [,1] [,2] [,3]
## [1,]    3    1    3
## [2,]    1    1    5
## [3,]    4    6    7
## [4,]    7    2    6
```

```
# Covariance
e1 = c(2,1,2,5)
e2 = c(9,5,8,3)
e3 = c(3,7,3,7)

cov_e1 <- matrix(cov(as.matrix(e1)), nrow = 1, ncol = 4)
cov_e1 = as.vector(cov_e1)
cv_1 = diag(cov_e1, nrow =  4, ncol = 4)
cv_1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    3    0    0    0
## [2,]    0    3    0    0
## [3,]    0    0    3    0
## [4,]    0    0    0    3
```

```
cov_e2 <- matrix(cov(as.matrix(e2)), nrow = 1, ncol = 4)
cov_e2 = as.vector(cov_e2)
cv_2 = diag(cov_e2, nrow =  4, ncol = 4)
cv_2
```

```
##          [,1]     [,2]     [,3]     [,4]
```

```
## [1,] 7.583333 0.000000 0.000000 0.000000
## [2,] 0.000000 7.583333 0.000000 0.000000
## [3,] 0.000000 0.000000 7.583333 0.000000
## [4,] 0.000000 0.000000 0.000000 7.583333
```

```r
cov_e3 <- matrix(cov(as.matrix(e3)), nrow = 1, ncol = 4)
cov_e3 = as.vector(cov_e3)
cv_3 = diag(cov_e3, nrow =  4, ncol = 4)
cv_3
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 5.333333 0.000000 0.000000 0.000000
## [2,] 0.000000 5.333333 0.000000 0.000000
## [3,] 0.000000 0.000000 5.333333 0.000000
## [4,] 0.000000 0.000000 0.000000 5.333333
```

```r
cov_test = array(c(cv_1, cv_2, cv_3), dim = c(4,4,3))
cov_test
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    3    0    0    0
## [2,]    0    3    0    0
## [3,]    0    0    3    0
## [4,]    0    0    0    3
##
## , , 2
##
##          [,1]     [,2]     [,3]     [,4]
## [1,] 7.583333 0.000000 0.000000 0.000000
## [2,] 0.000000 7.583333 0.000000 0.000000
## [3,] 0.000000 0.000000 7.583333 0.000000
## [4,] 0.000000 0.000000 0.000000 7.583333
##
## , , 3
##
##          [,1]     [,2]     [,3]     [,4]
## [1,] 5.333333 0.000000 0.000000 0.000000
## [2,] 0.000000 5.333333 0.000000 0.000000
## [3,] 0.000000 0.000000 5.333333 0.000000
## [4,] 0.000000 0.000000 0.000000 5.333333
```

```r
EM
```

```
## function(x, tau, Mu, covariance, eps){
##
##   x <- as.matrix(x)
## # cat("I am here 0", "\n")
##   K <- length(tau)
##   n <- dim(x)[1]
##   p <- dim(x)[2]
## # cat("I am here 1", "\n")
##   b <- 0
##
##   # cat("Iteration",b , "tau =", tau, "\n")
```

```
##  # cat("Iteration",b , "Mu =", Mu, "\n")
##  # cat("Iteration",b , "Cov =", covariance, "\n")
##  ll.old <- -Inf
##  ll <- logL(x, tau, Mu, covariance)
##  # cat("Iteration",b , "logL =", ll, "\n")
##  # cat("I am here 2", "\n")
##  repeat{
##
##   b <- b + 1
##
##   if ((ll - ll.old) / abs(ll) < eps) break
## # cat("I am here 3", "\n")
##   ll.old <- ll
##   # cat("ll.old", ll.old)
##   Pi <- E.step(x, tau, Mu, covariance)
##   # cat("I am here 4", "\n")
##   # cat("Pi", Pi)
##   M <- M.step(x,Pi)
##   # cat("I am here 5", "\n")
##
##   tau <- M$tau
##   Mu <- M$Mu
##   covariance <- M$covariance
##
##   ll <- logL(x, tau, Mu, covariance)
##   cat("Iteration", b, "logL =", ll, "\n")
##  }
##
##  id <- apply(Pi, 1, which.max)
##
##  M <- 3 * K - 1
##  BIC <- -2 * ll + M * log(n)
##  AIC <- -2 * ll + M * 2
##
##  return(list(tau = tau, Mu = Mu, covariance = covariance, logL = ll, BIC = BIC, Pi = Pi , id = id, A
##
## }
```

```
e_step = E.step(x,tau = t_test, Mu = mu_test, covariance = cov_test)
head(e_step)
```

```
##             [,1]      [,2]        [,3]
## [1,] 0.004034141 0.9906020 0.005363904
## [2,] 0.001789985 0.9956391 0.002570897
## [3,] 0.005642620 0.9864853 0.007872039
## [4,] 0.002938487 0.9924795 0.004581965
## [5,] 0.007858597 0.9809369 0.011204473
## [6,] 0.005820876 0.9870476 0.007131493
```

```
M.step(x,e_step)
```

```
## $tau
## [1] 0.003792102 0.990966614 0.005241285
##
## $Mu
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.7519923 0.7263730 0.7387400
## [2,] 0.7729384 0.6095877 0.7778393
## [3,] 0.5363171 0.5725524 0.5641754
## [4,] 0.5663530 0.4046128 0.5266918
##
## $covariance
## , , 1
##
##              [,1]        [,2]        [,3]         [,4]
## [1,]  0.027026228 -0.01129484  0.01044372 -0.003548204
## [2,] -0.011294844  0.12558039 -0.02027371  0.074279190
## [3,]  0.010443719 -0.02027371  0.06252995 -0.035975256
## [4,] -0.003548204  0.07427919 -0.03597526  0.077698737
##
## , , 2
##
##              [,1]         [,2]         [,3]        [,4]
## [1,]  0.030641674 -0.004234911  0.006552543  0.00633473
## [2,] -0.004234911  0.156627784 -0.016070655  0.09252424
## [3,]  0.006552543 -0.016070655  0.056824155 -0.02952570
## [4,]  0.006334730  0.092524239 -0.029525696  0.09167038
##
## , , 3
##
##               [,1]         [,2]         [,3]          [,4]
## [1,]  0.0264948453 -0.007384687  0.008418063  0.0005217973
## [2,] -0.0073846873  0.123860415 -0.017523184  0.0741857267
## [3,]  0.0084180626 -0.017523184  0.063870400 -0.0358561288
## [4,]  0.0005217973  0.074185727 -0.035856129  0.0810740332
```

```r
w = logL(x,tau = t_test, Mu = mu_test, covariance = cov_test)
w
```

```
## [1] -2294.995
```

```r
t_test
```

```
## [1] 0.3 0.2 0.5
```

```r
 mu_test
```

```
##      [,1] [,2] [,3]
## [1,]    3    1    3
## [2,]    1    1    5
## [3,]    4    6    7
## [4,]    7    2    6
```

```r
cov_test
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    3    0    0    0
## [2,]    0    3    0    0
## [3,]    0    0    3    0
## [4,]    0    0    0    3
```

```
##
## , , 2
##
##          [,1]     [,2]     [,3]     [,4]
## [1,] 7.583333 0.000000 0.000000 0.000000
## [2,] 0.000000 7.583333 0.000000 0.000000
## [3,] 0.000000 0.000000 7.583333 0.000000
## [4,] 0.000000 0.000000 0.000000 7.583333
##
## , , 3
##
##          [,1]     [,2]     [,3]     [,4]
## [1,] 5.333333 0.000000 0.000000 0.000000
## [2,] 0.000000 5.333333 0.000000 0.000000
## [3,] 0.000000 0.000000 5.333333 0.000000
## [4,] 0.000000 0.000000 0.000000 5.333333
```

```r
A2 <- EM(x,tau = t_test, Mu = mu_test, covariance = cov_test, eps = 1e-4)
```

```
## Iteration 1 logL = 57.67506
## Iteration 2 logL = 57.87712
## Iteration 3 logL = 58.29356
## Iteration 4 logL = 59.20043
## Iteration 5 logL = 61.28155
## Iteration 6 logL = 65.98723
## Iteration 7 logL = 74.72532
## Iteration 8 logL = 85.1141
## Iteration 9 logL = 93.35875
## Iteration 10 logL = 98.21477
## Iteration 11 logL = 101.3029
## Iteration 12 logL = 104.9472
## Iteration 13 logL = 110.9578
## Iteration 14 logL = 116.3597
## Iteration 15 logL = 118.7097
## Iteration 16 logL = 120.4017
## Iteration 17 logL = 122.0915
## Iteration 18 logL = 123.5305
## Iteration 19 logL = 124.4628
## Iteration 20 logL = 124.982
## Iteration 21 logL = 125.2709
## Iteration 22 logL = 125.4475
## Iteration 23 logL = 125.5712
## Iteration 24 logL = 125.6699
## Iteration 25 logL = 125.7576
## Iteration 26 logL = 125.8414
## Iteration 27 logL = 125.9263
## Iteration 28 logL = 126.0168
## Iteration 29 logL = 126.1195
## Iteration 30 logL = 126.2442
## Iteration 31 logL = 126.4027
## Iteration 32 logL = 126.5986
## Iteration 33 logL = 126.8076
## Iteration 34 logL = 126.9814
## Iteration 35 logL = 127.0931
## Iteration 36 logL = 127.1553
```

```
## Iteration 37 logL = 127.1907
## Iteration 38 logL = 127.213
## Iteration 39 logL = 127.2282
## Iteration 40 logL = 127.2391
```

```
A2$logL
```

```
## [1] 127.2391
```

```
A2$BIC
```

```
## [1] -212.0917
```

```
A2$Pi
```

```
##               [,1]         [,2]         [,3]
##  [1,] 9.995905e-01 4.084825e-04 1.022681e-06
##  [2,] 3.601605e-09 1.000000e+00 8.889233e-15
##  [3,] 7.091440e-01 2.056891e-01 8.516692e-02
##  [4,] 4.228164e-05 4.381380e-01 5.618197e-01
##  [5,] 9.999530e-01 4.697777e-05 5.142498e-12
##  [6,] 7.859908e-01 1.320463e-03 2.126888e-01
##  [7,] 8.922859e-03 2.488073e-01 7.422698e-01
##  [8,] 9.510230e-01 4.897702e-02 1.711783e-08
##  [9,] 8.940437e-02 2.005333e-03 9.085903e-01
## [10,] 1.693192e-12 1.000000e+00 4.459058e-24
## [11,] 4.529813e-02 9.547019e-01 1.120783e-17
## [12,] 2.792791e-12 1.000000e+00 1.808308e-18
## [13,] 3.689013e-09 5.100868e-01 4.899132e-01
## [14,] 2.703789e-03 9.972962e-01 8.614883e-39
## [15,] 4.516456e-06 8.228146e-01 1.771809e-01
## [16,] 3.541894e-13 2.000547e-01 7.999453e-01
## [17,] 3.507829e-14 1.000000e+00 2.299652e-11
## [18,] 2.140700e-04 9.997859e-01 5.037984e-10
## [19,] 6.259590e-01 3.739896e-01 5.146029e-05
## [20,] 1.638664e-04 9.998361e-01 2.412570e-46
## [21,] 1.052925e-17 9.998772e-01 1.227513e-04
## [22,] 8.027106e-01 1.972894e-01 7.351494e-19
## [23,] 3.270800e-11 3.569256e-01 6.430743e-01
## [24,] 9.092052e-01 1.038632e-03 8.975615e-02
## [25,] 6.158494e-37 1.000000e+00 9.776589e-14
## [26,] 3.603269e-19 1.000000e+00 2.711749e-09
## [27,] 8.825327e-16 9.999103e-01 8.970152e-05
## [28,] 1.722991e-19 1.000000e+00 5.013326e-09
## [29,] 8.462776e-17 9.999994e-01 6.097251e-07
## [30,] 1.609667e-07 9.999998e-01 1.441403e-15
## [31,] 9.870330e-01 1.316270e-05 1.295385e-02
## [32,] 2.144637e-10 1.000000e+00 1.567786e-20
## [33,] 2.727743e-11 9.487846e-01 5.121536e-02
## [34,] 7.674714e-01 3.084175e-03 2.294445e-01
## [35,] 1.024862e-08 9.853091e-01 1.469091e-02
## [36,] 1.052385e-09 1.000000e+00 6.212381e-60
## [37,] 1.499020e-09 4.438483e-01 5.561517e-01
## [38,] 4.029196e-21 1.000000e+00 2.684457e-11
## [39,] 9.375499e-01 7.578531e-04 6.169228e-02
## [40,] 5.768946e-02 9.423105e-01 2.404153e-17
```

```
## [41,] 1.902031e-06 9.999981e-01 3.247399e-34
## [42,] 9.472674e-01 5.273260e-02 3.776133e-14
## [43,] 3.375116e-06 9.969532e-01 3.043461e-03
## [44,] 6.376908e-16 1.000000e+00 1.603291e-20
## [45,] 2.760926e-06 2.627320e-01 7.372653e-01
## [46,] 4.063097e-06 4.402688e-01 5.597271e-01
## [47,] 4.884485e-35 1.000000e+00 6.782514e-10
## [48,] 9.555281e-01 3.575098e-02 8.720875e-03
## [49,] 1.331468e-01 8.668532e-01 5.594239e-27
## [50,] 1.373563e-13 9.998707e-01 1.292968e-04
## [51,] 1.329910e-01 8.392206e-01 2.778836e-02
## [52,] 9.999955e-01 4.466384e-06 3.580639e-09
## [53,] 9.613484e-01 1.783939e-02 2.081220e-02
## [54,] 1.216694e-07 1.811705e-01 8.188294e-01
## [55,] 1.863899e-10 9.449403e-01 5.505970e-02
## [56,] 6.421426e-05 9.999357e-01 6.482078e-08
## [57,] 9.995950e-01 4.049523e-04 4.405394e-11
## [58,] 3.391042e-02 1.021113e-03 9.650685e-01
## [59,] 1.758583e-01 7.490152e-04 8.233926e-01
## [60,] 9.901403e-01 3.060320e-05 9.829109e-03
## [61,] 4.005866e-17 9.999999e-01 1.189999e-07
## [62,] 6.075013e-07 9.999994e-01 2.687877e-31
## [63,] 9.183368e-03 2.103630e-01 7.804537e-01
## [64,] 1.990342e-06 9.999980e-01 1.467029e-10
## [65,] 2.052367e-10 1.000000e+00 1.120500e-24
## [66,] 1.184121e-10 1.000000e+00 1.173399e-14
## [67,] 5.081999e-06 1.919806e-01 8.080144e-01
## [68,] 8.342666e-01 5.606287e-05 1.656774e-01
## [69,] 3.266398e-01 1.352903e-03 6.720073e-01
## [70,] 6.761225e-07 3.798810e-01 6.201183e-01
## [71,] 4.684220e-10 1.000000e+00 6.411458e-24
## [72,] 1.600140e-11 1.000000e+00 9.562967e-10
## [73,] 9.740292e-01 2.597082e-02 1.571338e-10
## [74,] 8.027270e-06 2.718023e-01 7.281897e-01
## [75,] 1.981088e-10 1.165132e-01 8.834868e-01
## [76,] 7.861558e-01 1.329259e-02 2.005516e-01
## [77,] 4.603625e-01 2.353433e-03 5.372841e-01
## [78,] 1.048608e-04 8.742197e-01 1.256754e-01
## [79,] 1.177921e-03 2.130041e-01 7.858180e-01
## [80,] 1.554656e-03 9.984453e-01 3.887560e-34
## [81,] 2.152598e-08 1.000000e+00 1.622619e-22
## [82,] 1.796786e-09 8.545606e-01 1.454394e-01
## [83,] 9.999997e-01 2.557306e-07 9.060081e-11
## [84,] 9.912740e-01 8.724423e-03 1.570627e-06
## [85,] 9.999187e-01 5.509306e-06 7.574900e-05
## [86,] 1.601232e-19 1.000000e+00 7.900461e-14
## [87,] 3.432585e-01 6.567415e-01 7.215415e-16
## [88,] 2.766732e-07 9.999997e-01 2.381034e-14
## [89,] 3.907992e-02 1.905562e-01 7.703639e-01
## [90,] 8.006438e-16 9.783620e-01 2.163799e-02
## [91,] 1.400724e-17 1.000000e+00 1.536302e-13
## [92,] 3.712026e-02 6.249804e-02 9.003817e-01
## [93,] 7.482762e-08 2.121931e-01 7.878068e-01
## [94,] 9.993922e-01 5.815402e-04 2.628541e-05
```

```
##  [95,] 5.153887e-16 9.999999e-01 9.184090e-08
##  [96,] 5.084643e-21 1.000000e+00 2.201921e-14
##  [97,] 1.198369e-08 1.000000e+00 4.462894e-38
##  [98,] 2.535017e-04 9.997465e-01 1.444111e-43
##  [99,] 2.790770e-08 8.868511e-01 1.131488e-01
## [100,] 4.852629e-15 1.000000e+00 1.603838e-14
## [101,] 2.961883e-12 6.637472e-01 3.362528e-01
## [102,] 9.873898e-01 1.259736e-02 1.288120e-05
## [103,] 4.050439e-08 1.000000e+00 8.224600e-35
## [104,] 2.956782e-13 9.811442e-01 1.885578e-02
## [105,] 1.163203e-17 1.000000e+00 2.214531e-09
## [106,] 5.776475e-07 9.999994e-01 8.924341e-10
## [107,] 6.834864e-10 3.223574e-01 6.776426e-01
## [108,] 5.005785e-14 1.000000e+00 5.136753e-16
## [109,] 9.632792e-01 3.671869e-02 2.136248e-06
## [110,] 4.829057e-01 2.392746e-05 5.170704e-01
## [111,] 1.128609e-04 9.741541e-01 2.573299e-02
## [112,] 2.893856e-24 9.999983e-01 1.736510e-06
## [113,] 1.904311e-13 9.958802e-01 4.119821e-03
## [114,] 2.079837e-06 2.248519e-01 7.751460e-01
## [115,] 1.774955e-09 9.953886e-01 4.611429e-03
## [116,] 9.902939e-01 1.805348e-04 9.525543e-03
## [117,] 9.999633e-01 3.667556e-05 3.221785e-08
## [118,] 9.998189e-01 1.718372e-04 9.264880e-06
## [119,] 7.388266e-04 1.334476e-01 8.658136e-01
## [120,] 3.994071e-08 1.000000e+00 4.372845e-29
## [121,] 3.778110e-08 1.000000e+00 4.011258e-19
## [122,] 9.997289e-01 2.699260e-04 1.180348e-06
## [123,] 9.229592e-03 9.907295e-01 4.086300e-05
## [124,] 4.250383e-01 1.803721e-02 5.569245e-01
## [125,] 6.200258e-07 8.411959e-01 1.588035e-01
## [126,] 9.996009e-01 3.991393e-04 6.522016e-13
## [127,] 7.151974e-02 2.002533e-02 9.084549e-01
## [128,] 1.402747e-02 9.859725e-01 2.790048e-40
## [129,] 1.920184e-16 1.000000e+00 5.744161e-17
## [130,] 2.887285e-10 9.999999e-01 1.243553e-07
## [131,] 1.363941e-09 1.000000e+00 5.792355e-15
## [132,] 1.076899e-06 2.187390e-01 7.812599e-01
## [133,] 9.598998e-01 9.332504e-05 4.000686e-02
## [134,] 1.010144e-05 5.175306e-01 4.824593e-01
## [135,] 7.909886e-09 9.426774e-01 5.732259e-02
## [136,] 9.996707e-01 3.292880e-04 9.808405e-18
## [137,] 1.050972e-01 8.941988e-01 7.040040e-04
## [138,] 7.651916e-07 9.965789e-01 3.420333e-03
## [139,] 8.894010e-01 3.291183e-02 7.768719e-02
## [140,] 9.385732e-22 9.997759e-01 2.241251e-04
## [141,] 2.955337e-13 3.801737e-01 6.198263e-01
## [142,] 6.758768e-01 6.639003e-04 3.234593e-01
## [143,] 6.586601e-06 9.999934e-01 5.456008e-47
## [144,] 5.143177e-03 8.789395e-01 1.159173e-01
## [145,] 9.819992e-03 9.851309e-01 5.049109e-03
## [146,] 3.363658e-07 9.999997e-01 6.932764e-29
## [147,] 6.793584e-01 3.168792e-01 3.762393e-03
## [148,] 8.720077e-13 6.877639e-01 3.122361e-01
```

```
## [149,] 9.973249e-01 2.335734e-03 3.393486e-04
## [150,] 7.403955e-01 4.752011e-02 2.120844e-01
## [151,] 3.399115e-10 4.764563e-01 5.235437e-01
## [152,] 4.664489e-13 9.534945e-01 4.650555e-02
## [153,] 9.896433e-01 4.843765e-04 9.872368e-03
## [154,] 5.411330e-07 2.244363e-01 7.755631e-01
## [155,] 5.327327e-32 1.000000e+00 5.769043e-21
## [156,] 1.443693e-05 9.999856e-01 5.389355e-10
## [157,] 9.722732e-06 9.999819e-01 8.370854e-06
## [158,] 6.810935e-10 1.000000e+00 1.057579e-23
## [159,] 3.569205e-03 9.964308e-01 2.054300e-35
## [160,] 2.493695e-09 9.999909e-01 9.134559e-06
## [161,] 6.010006e-05 9.878742e-01 1.206572e-02
## [162,] 2.089967e-02 9.791003e-01 2.532134e-62
## [163,] 7.767061e-09 4.718333e-01 5.281667e-01
## [164,] 8.558934e-03 9.914183e-01 2.276595e-05
## [165,] 6.590317e-11 9.999996e-01 4.451514e-07
## [166,] 5.989188e-01 4.685220e-02 3.542290e-01
## [167,] 2.396267e-05 1.285058e-01 8.714702e-01
## [168,] 8.903117e-11 9.238380e-01 7.616198e-02
## [169,] 9.154196e-01 8.457942e-02 9.704114e-07
## [170,] 9.999774e-01 2.260905e-05 5.897390e-15
## [171,] 2.151713e-05 9.999785e-01 8.098431e-15
## [172,] 5.232390e-02 5.429770e-03 9.422463e-01
## [173,] 6.038942e-01 3.644907e-05 3.960694e-01
## [174,] 3.896229e-25 1.000000e+00 6.980488e-13
## [175,] 3.040014e-05 9.999696e-01 1.805503e-15
## [176,] 3.399556e-01 5.866551e-02 6.013789e-01
## [177,] 1.621697e-21 9.999996e-01 4.196202e-07
## [178,] 1.391606e-06 2.179000e-01 7.820986e-01
## [179,] 4.306897e-13 9.999337e-01 6.629632e-05
## [180,] 5.226031e-01 4.738699e-01 3.526987e-03
## [181,] 7.100319e-01 1.547373e-03 2.884207e-01
## [182,] 6.467650e-01 2.477974e-05 3.532102e-01
## [183,] 9.853484e-01 1.464624e-02 5.348261e-06
## [184,] 5.722956e-01 4.276988e-01 5.606982e-06
## [185,] 7.798590e-01 2.241625e-05 2.201185e-01
## [186,] 9.523718e-01 4.762821e-02 2.839056e-12
## [187,] 9.680218e-13 3.841866e-01 6.158134e-01
## [188,] 3.304399e-02 1.518063e-02 9.517754e-01
## [189,] 3.952331e-01 4.312274e-06 6.047626e-01
## [190,] 3.287164e-08 8.282449e-01 1.717551e-01
## [191,] 1.439367e-12 1.000000e+00 3.640963e-09
## [192,] 5.067753e-44 1.000000e+00 6.164414e-20
## [193,] 5.572544e-11 1.000000e+00 5.682245e-20
## [194,] 2.787076e-04 9.997213e-01 3.743346e-29
## [195,] 4.968963e-01 1.612555e-03 5.014911e-01
## [196,] 6.270324e-03 9.937297e-01 3.340329e-17
## [197,] 2.686435e-15 9.964360e-01 3.563971e-03
## [198,] 4.619433e-02 9.538057e-01 9.149034e-43
## [199,] 1.233995e-09 9.990709e-01 9.290747e-04
## [200,] 1.447338e-06 9.999986e-01 2.153884e-11
```

```r
A2$id
```

```
##   [1] 1 2 1 3 1 1 3 1 3 2 2 2 2 2 2 3 2 2 1 2 2 1 3 1 2 2 2 2 2 1 2 2 1 2 2 3
##  [38] 2 1 2 2 1 2 2 3 3 2 1 2 2 2 1 1 3 2 2 1 3 3 1 2 2 3 2 2 2 3 1 3 3 2 2 1 3
##  [75] 3 1 3 2 3 2 2 2 1 1 1 2 2 2 3 2 2 3 3 1 2 2 2 2 2 2 2 1 2 2 2 2 3 2 1 3 2
## [112] 2 2 3 2 1 1 1 3 2 2 1 2 3 2 1 3 2 2 2 2 3 1 2 2 1 2 2 1 2 3 1 2 2 2 2 1 2
## [149] 1 1 3 2 1 3 2 2 2 2 2 2 2 2 3 2 2 1 3 2 1 1 2 3 1 2 2 3 2 3 2 1 1 1 1 1 1
## [186] 1 3 3 3 2 2 2 2 2 3 2 2 2 2 2
```

```r
A2$tau
```

```
## [1] 0.2424995 0.5762891 0.1812114
```

```r
A2$Mu
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.7531344 0.7288974 0.6834262
## [2,] 0.9475086 0.4019351 0.8260401
## [3,] 0.3425331 0.6684643 0.5743474
## [4,] 0.7679217 0.2246528 0.4976532
```

```r
A2$covariance
```

```
## , , 1
## 
##              [,1]         [,2]         [,3]         [,4]
## [1,]  0.017988857 -0.013888498  0.014139303 -0.009848237
## [2,] -0.013888498  0.057504415 -0.005139298  0.018974836
## [3,]  0.014139303 -0.005139298  0.035020113 -0.004910857
## [4,] -0.009848237  0.018974836 -0.004910857  0.017213106
## 
## , , 2
## 
##              [,1]        [,2]        [,3]       [,4]
## [1,]  0.041920142 -0.00203399 0.008128636 0.01477478
## [2,] -0.002033990  0.13579775 0.024065460 0.05385697
## [3,]  0.008128636  0.02406546 0.039930792 0.01186650
## [4,]  0.014774784  0.05385697 0.011866502 0.04546178
## 
## , , 3
## 
##               [,1]         [,2]          [,3]         [,4]
## [1,]  0.0087054938 0.0004543623 -0.0009548149 -0.006665163
## [2,]  0.0004543623 0.0185073200  0.0083318628  0.010881166
## [3,] -0.0009548149 0.0083318628  0.0400180956 -0.028015353
## [4,] -0.0066651628 0.0108811660 -0.0280153535  0.050357898
```

Maths in R-Markdown https://rpruim.github.io/s341/S19/from-class/MathinRmd.html