CS539 hw1

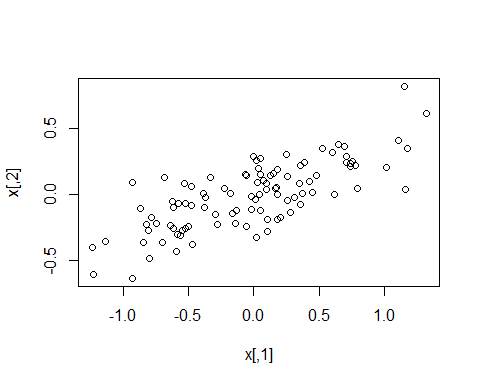
Enbo Tian

2022/1/24

# problem 1

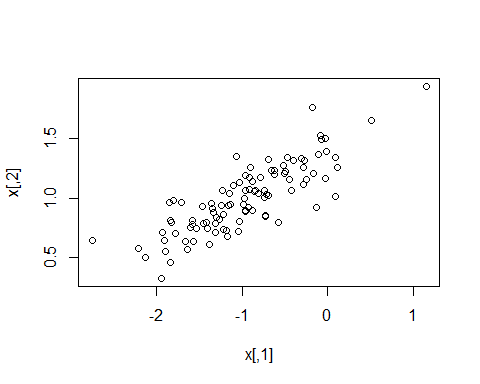
## a)

X <- matrix(runif(2 \* 2), 2, 2)  
COV <- crossprod(X)  
mu <- rep(0, 2)  
library(MASS)  
x <- mvrnorm(100, mu, COV)  
plot(x)



# b)

mu <- c(-1,1)  
x <- mvrnorm(100, mu, COV)  
plot(x)

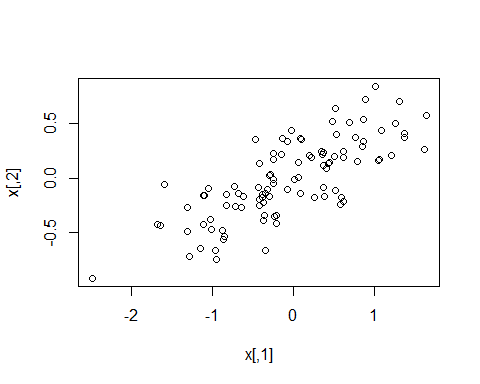


mu <- c(0,0)

The interval of x1 is moving left by about 1, and the interval of x2 is moving up by about 1.

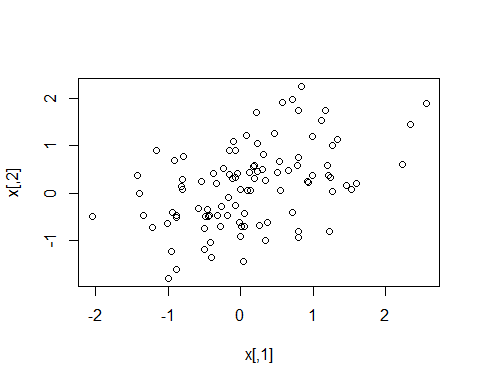
## c)

COV <- 2\*COV  
x <- mvrnorm(100, mu, COV)  
plot(x)



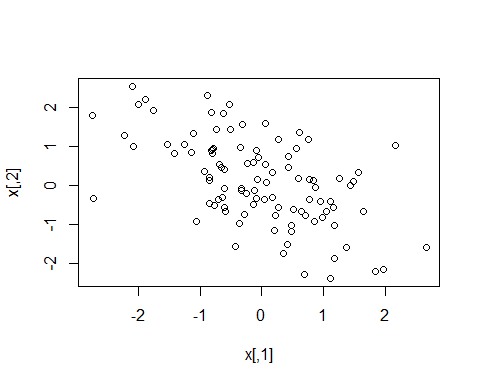
## d)

COV <- matrix(c(1,0.5,0.5,1), nrow = 2, ncol = 2)  
x <- mvrnorm(100, mu, COV)  
plot(x)



## e)

COV <- matrix(c(1,-0.5,-0.5,1), nrow = 2, ncol = 2)  
x <- mvrnorm(100, mu, COV)  
plot(x)



## f)

X <- matrix(runif(2 \* 2), 2, 2)  
COV <- crossprod(X)  
mu <- rep(0, 2)  
x <- mvrnorm(100, mu, COV)  
mean(x)

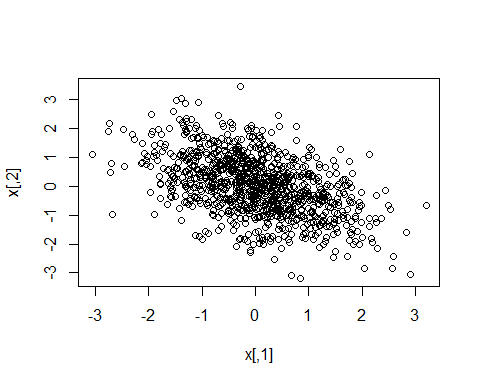
## [1] 0.003945048

cov(x)

## [,1] [,2]  
## [1,] 1.910007 1.331396  
## [2,] 1.331396 0.970151

## g)

COV <- matrix(c(1,-0.5,-0.5,1), nrow = 2, ncol = 2)  
x <- mvrnorm(1000, mu, COV)  
plot(x)



## h)

mean(x)

## [1] 0.01909849

cov(x)

## [,1] [,2]  
## [1,] 0.9773492 -0.484074  
## [2,] -0.4840740 1.036510

## i)

x <- mvrnorm(10, mu, COV)  
mean(x[,1])

## [1] 0.2690736

x <- mvrnorm(100, mu, COV)  
mean(x[,1])

## [1] 0.01383696

x <- mvrnorm(1000, mu, COV)  
mean(x[,1])

## [1] 0.01326269

Mean is tend to 0, as the more samples we have

## j)

COV # the initial covariance we use to get the sample

## [,1] [,2]  
## [1,] 1.0 -0.5  
## [2,] -0.5 1.0

x <- mvrnorm(10, mu, COV)  
cov(x)

## [,1] [,2]  
## [1,] 1.196858 -1.169756  
## [2,] -1.169756 2.085406

x <- mvrnorm(100, mu, COV)  
cov(x)

## [,1] [,2]  
## [1,] 1.008264 -0.5121600  
## [2,] -0.512160 0.9278948

x <- mvrnorm(1000, mu, COV)  
cov(x)

## [,1] [,2]  
## [1,] 0.9841772 -0.4980792  
## [2,] -0.4980792 0.9746574

covariance is getting closer to the initial covariance, When we have more sample

# problem 2

## a)

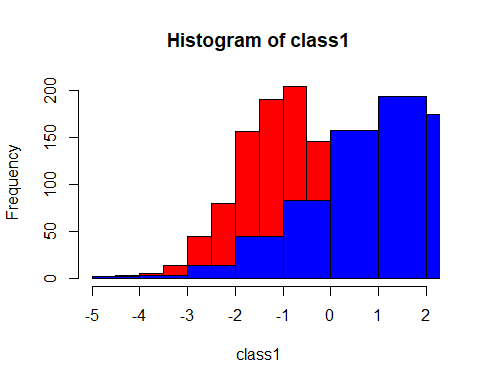
class1 <- rnorm(1000,-1,1)

## b)

class2 <- rnorm(1000,2,2)

## c)

hist(class1, col='red')  
hist(class2, col='blue', add=TRUE)



## d)

library(stats4)  
library(methods)  
options(warn = -1)  
LL1 <- function(mu,sigma){  
 -sum(log(dnorm(class1,mu,sigma)))  
}  
m1<-mle(LL1,start = list(mu=0,sigma=1))  
m1

##   
## Call:  
## mle(minuslogl = LL1, start = list(mu = 0, sigma = 1))  
##   
## Coefficients:  
## mu sigma   
## -0.9980152 1.0034457

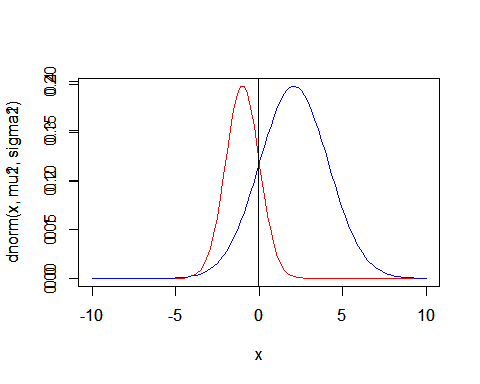
LL2 <- function(mu,sigma){  
 R = dnorm(class2,mu,sigma)  
 -sum(log(R))  
}  
m2<-mle(LL2,start = list(mu=0,sigma=1))  
m2

##   
## Call:  
## mle(minuslogl = LL2, start = list(mu = 0, sigma = 1))  
##   
## Coefficients:  
## mu sigma   
## 2.082832 2.043154

options(warn = getOption("warn"))

## e)

mu1 <- m1@coef[1]  
sigma1<- m1@coef[2]  
mu2<-m2@coef[1]  
sigma2<-m2@coef[2]  
  
x <- seq(-10, 10, length=100)  
plot(x,dnorm(x,mu1,sigma1), type = "l",col = "red")  
par(new=TRUE)  
plot(x,dnorm(x,mu2,sigma2), type = "l",col="blue")  
i = -2  
while(round(dnorm(i,mu1,sigma1),5)!=round(dnorm(i,mu2,sigma2),5)){  
 i=i+0.00001  
}  
par(new=TRUE)  
abline(v=-0.014)

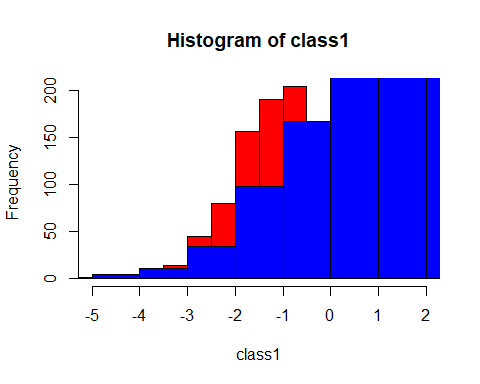


## f)

both of the decision boundary of pdf and histogram are about 0

## g)

class2 <- rnorm(2000,2,2)  
#c  
hist(class1, col='red')  
hist(class2, col='blue', add=TRUE)



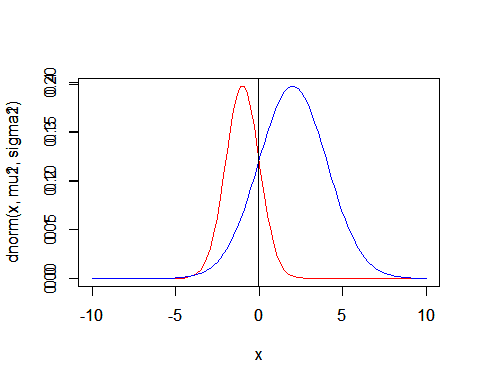
#d  
options(warn = -1)  
LL1 <- function(mu,sigma){  
 -sum(log(dnorm(class1,mu,sigma)))  
}  
m1<-mle(LL1,start = list(mu=0,sigma=1))  
m1

##   
## Call:  
## mle(minuslogl = LL1, start = list(mu = 0, sigma = 1))  
##   
## Coefficients:  
## mu sigma   
## -0.9980152 1.0034457

LL2 <- function(mu,sigma){  
 R = dnorm(class2,mu,sigma)  
 -sum(log(R))  
}  
m2<-mle(LL2,start = list(mu=0,sigma=1))  
m2

##   
## Call:  
## mle(minuslogl = LL2, start = list(mu = 0, sigma = 1))  
##   
## Coefficients:  
## mu sigma   
## 2.014727 2.036637

options(warn = getOption("warn"))  
#e  
mu1 <- m1@coef[1]  
sigma1<- m1@coef[2]  
mu2<-m2@coef[1]  
sigma2<-m2@coef[2]  
  
x <- seq(-10, 10, length=100)  
plot(x,dnorm(x,mu1,sigma1), type = "l",col = "red")  
par(new=TRUE)  
plot(x,dnorm(x,mu2,sigma2), type = "l",col="blue")  
i = -2  
while(round(dnorm(i,mu1,sigma1),5)!=round(dnorm(i,mu2,sigma2),5)){  
 i=i+0.0001  
}  
par(new=TRUE)  
abline(v=-0.016)



#f

Since there are more samples in class2, the decision boundary of histogram comes to -1, but the decision boundary of pdf does not change.

## h)

library(MASS)  
fitdistr(class1, densfun="normal")

## mean sd   
## -0.99793273 1.00348621   
## ( 0.03173302) ( 0.02243863)

class2 <- rnorm(1000,2,2)  
fitdistr(class2, densfun="normal")

## mean sd   
## 1.97822236 1.94124094   
## (0.06138743) (0.04340747)

the error rate are on the second line.

library(MASS)  
fitdistr(class1, densfun="normal")

## mean sd   
## -0.99793273 1.00348621   
## ( 0.03173302) ( 0.02243863)

class2 <- rnorm(2000,2,2)  
fitdistr(class2, densfun="normal")

## mean sd   
## 1.95616385 2.03949020   
## (0.04560439) (0.03224717)

## i)

options(warn = -1)  
df <- data.frame(class1,class2)  
library(pROC)

## Type 'citation("pROC")' for a citation.

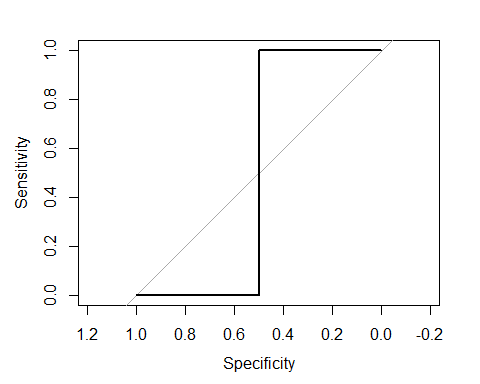
##   
## 载入程辑包：'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

roc(df$class1,df$class2,plot=TRUE)

## Setting levels: control = -4.6552738797062, case = -4.31843317823687

## Setting direction: controls < cases



##   
## Call:  
## roc.default(response = df$class1, predictor = df$class2, plot = TRUE)  
##   
## Data: df$class2 in 2 controls (df$class1 -4.6552738797062) < 2 cases (df$class1 -4.31843317823687).  
## Area under the curve: 0.5

options(warn = getOption("warn"))

# problem 3

## a)

library("Rlab")

## Rlab 2.15.1 attached.

##   
## 载入程辑包：'Rlab'

## The following object is masked from 'package:MASS':  
##   
## michelson

## The following objects are masked from 'package:stats':  
##   
## dexp, dgamma, dweibull, pexp, pgamma, pweibull, qexp, qgamma,  
## qweibull, rexp, rgamma, rweibull

## The following object is masked from 'package:datasets':  
##   
## precip

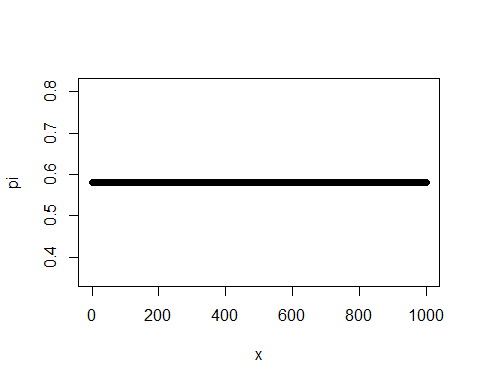
coin <- rbern(1000, 0.6)

##b)

options(warn = -1)  
LL1 <- function(p){  
 -sum(log(dbern(coin,p)))  
}  
m1<-mle(LL1,start = list(p=0.01))  
m1@coef

## p   
## 0.5809974

LL <- function(n,p){  
 -sum(log(dbern(rbern(n, 0.6),p)))  
}  
  
for(n in 1:1000){  
 ll <- function(p){  
 LL(n,p)  
 }  
 m1<-mle(LL1,start = list(p=0.01))  
 pi[n]<-m1@coef  
}  
  
x<-seq(0, 1000, length=1000)  
plot(x,pi)



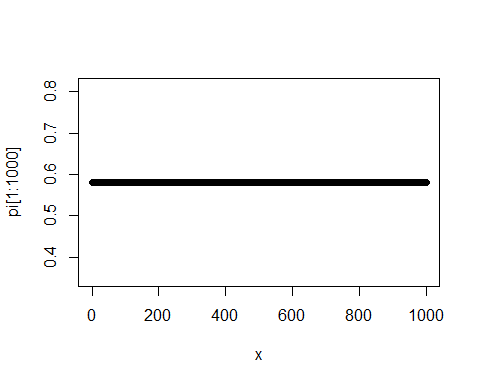
options(warn = getOption("warn"))

## c)

coin2 <- rbern(1000, 0.6)  
LL2 <- function(p){  
 -sum(log(dbern(coin2,p)))  
}  
m2<-mle(LL2,start = list(p=0.01))  
m2@coef

## p   
## 0.5999919

LL <- function(n,p){  
 -sum(log(dbern(rbern(n, 0.6),p)))  
}  
  
for(n in 1:100){  
 ll <- function(p){  
 LL(n,p)  
 }  
 m1<-mle(LL1,start = list(p=0.01))  
 pi[n]<-m1@coef  
}  
x<-seq(0, 1000, length=1000)  
plot(x,pi[1:1000])

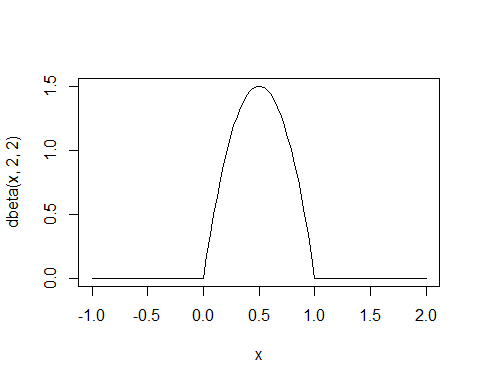


options(warn = getOption("warn"))

both from b) and c) are approximate and get close to 0.6,

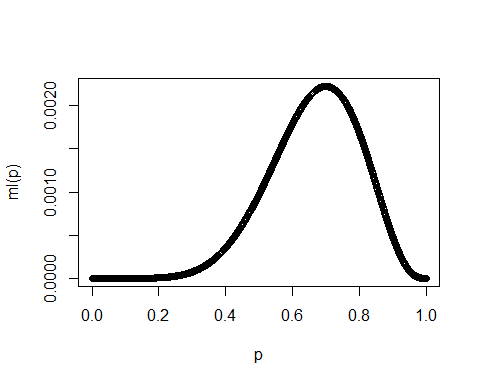
## d)

x <- seq(-1, 2, length=100)  
plot(x,dbeta(x, 2, 2), type = "l")



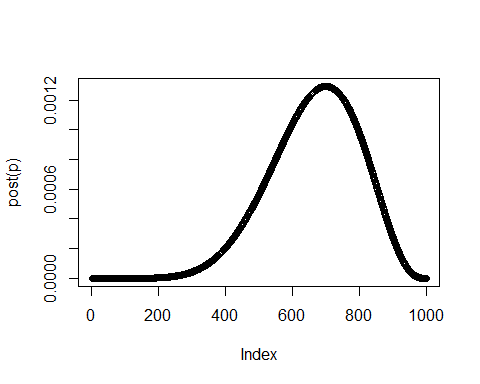
## e)

p <- seq(0, 1, length=1000)  
ml <- function(p){  
 mult = 1   
 for(i in 1:10){  
 mult <- mult\*p^coin[i]\*(1-p)^(1-coin[i])  
 }  
 mult  
}  
plot(p,ml(p))



## f)

post <- function(p){  
 ml(p)\*pi  
}  
plot(post(p))

 Since the posterior is proportion to prior and likelihood, the curve is not change too much.

## g)

max(post(p))

## [1] 0.001291883

MAP is 6.915e-04

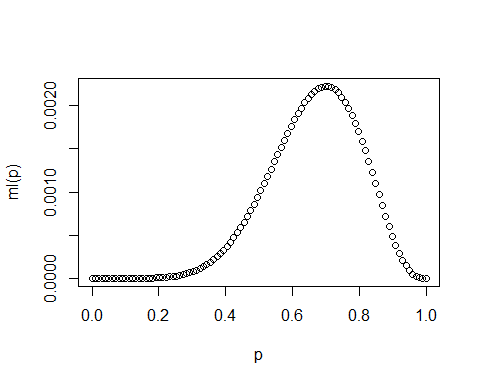
## h)

var(post(p))

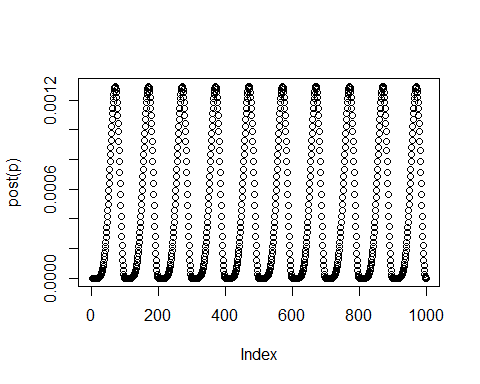
## [1] 2.211729e-07

## i)

p <- seq(0, 1, length=100)  
ml <- function(p){  
 mult = 1   
 for(i in 1:10){  
 mult <- mult\*p^coin[i]\*(1-p)^(1-coin[i])  
 }  
 mult  
}  
plot(p,ml(p))



post <- function(p){  
 ml(p)\*pi  
}  
plot(post(p))



max(post(p))

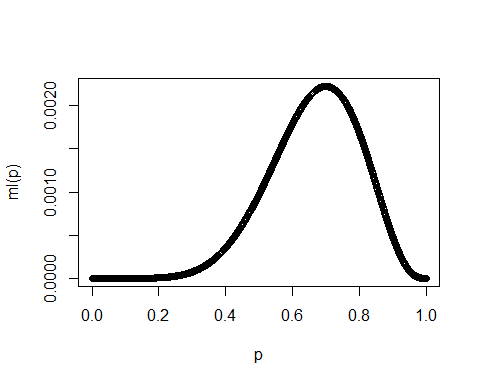
## [1] 0.001291605

var(post(p))

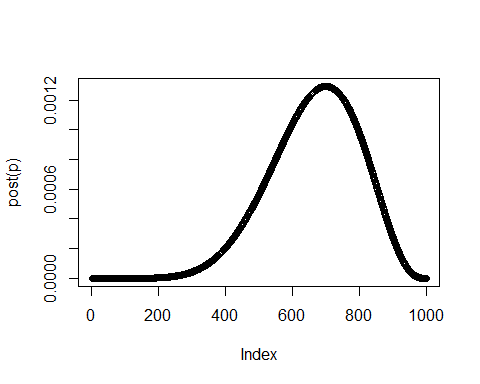
## [1] 2.209082e-07

## j)

p <- seq(0, 1, length=1000)  
ml <- function(p){  
 mult = 1   
 for(i in 1:10){  
 mult <- mult\*p^coin[i]\*(1-p)^(1-coin[i])  
 }  
 mult  
}  
plot(p,ml(p))



post <- function(p){  
 ml(p)\*pi  
}  
plot(post(p))



max(post(p))

## [1] 0.001291883

var(post(p))

## [1] 2.211729e-07

# problem 4

## a)

library(MASS)  
pi1 <- 0.1  
mu1 <- c(3, 2)  
COV1 <- matrix(c(1,0,0,1), nrow = 2, ncol = 2)  
  
pi2 <- 0.6  
mu2 <- c(-5, -3)  
COV2 <- matrix(c(2,-1,-1,3), nrow = 2, ncol = 2)  
  
pi3<-0.3  
COV3 <- matrix(c(6,3,3,3), nrow = 2, ncol = 2)  
mu3 <- c(4, 2)  
  
p <- pi1\*mvrnorm(1000, mu1, COV1)+pi2\*mvrnorm(1000, mu2, COV2)+pi3\*mvrnorm(1000, mu3, COV3)

## b)

mean(p[1])

## [1] -3.523997

mean(p[2])

## [1] -3.288717

cov(p)

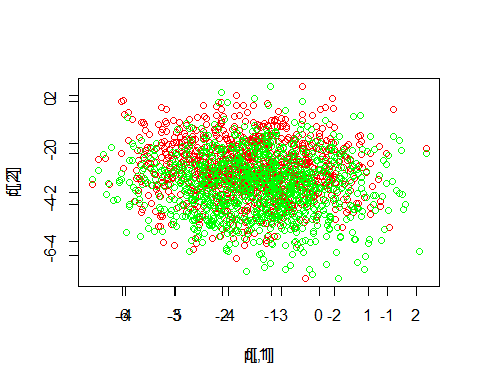
## [,1] [,2]  
## [1,] 1.298926 -0.109365  
## [2,] -0.109365 1.385695

## c)

mu <- c(mean(p[1]),mean(p[2]))  
COV <- cov(p)  
f <- mvrnorm(1000, mu, COV)

## d)

plot(p,col="red")  
par(new=TRUE)  
plot(f,col="green")

 the mixture model have the same concentration area with multivariate normal distribution. The difference is that the mixture model have a more range.

## e)

K3<-kmeans(p, centers = 3, nstart = 25)  
str(K3)

## List of 9  
## $ cluster : int [1:1000] 2 2 2 1 2 2 1 3 2 1 ...  
## $ centers : num [1:3, 1:2] -2.239 -1.715 -0.166 -0.147 -2.185 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:3] "1" "2" "3"  
## .. ..$ : NULL  
## $ totss : num 2682  
## $ withinss : num [1:3] 466 422 324  
## $ tot.withinss: num 1212  
## $ betweenss : num 1470  
## $ size : int [1:3] 368 348 284  
## $ iter : int 3  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"