Statististic Computing hw3

106070020

2021年4月28日

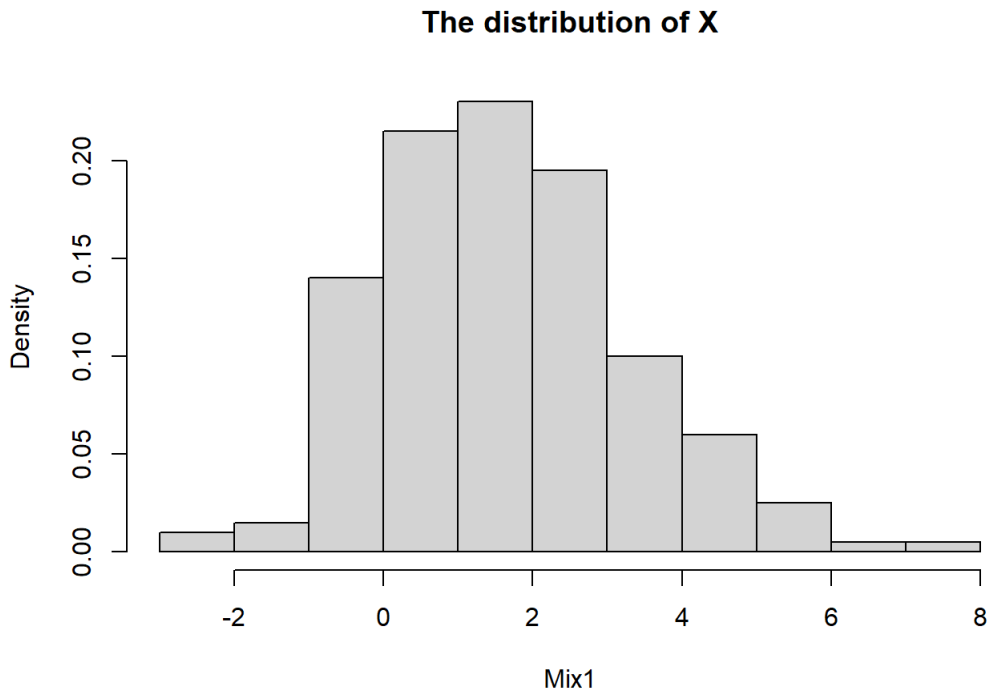
Libraries

Problem 2

```
#Q2
p1<-0.6;p2<-0.4
n1<-100*p1; mx1<-1; mx2<-2; sx1<-1; sx2<-2; corx<-0.4
n2<-100*p2; mz1<-2; mz2<-1; sz1<-2; sz2<-1; corz<-0.6

set.seed(03)
U <-runif(100,0,1)
Mix1<-matrix(nrow=100, ncol=2, byrow = T)
for(i in 1:nrow(Mix1)){
  if(U[i]<.6){
    x1<-rnorm(1,mx1,sx1)
    x2<-sx2*corx*(x1-mx1)/sx1+mx2+sx2*rnorm(1,0,sqrt(1-corx^2))
    X<-cbind(x1,x2)
    Mix1[i,]<-X
  }else{
    z1<-rnorm(1,mz1,sz1)
    z2<-sz2*corz*(z1-mz1)/sz1+mz2+sz2*rnorm(1,0,sqrt(1-corz^2))
    Z<-cbind(z1,z2)
    Mix1[i,]<-Z
  }
}
head(Mix1)
```

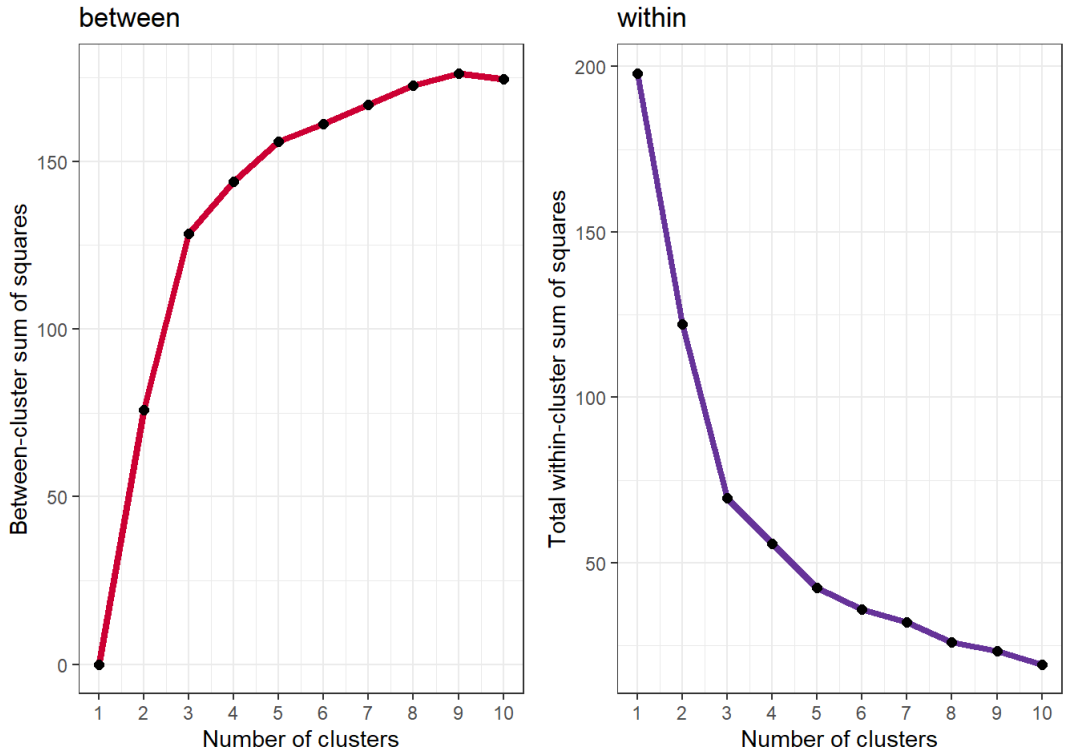
```
##             [,1]        [,2]
## [1,]  1.72683890  1.09774144
## [2,]  2.53417023 -0.22955990
## [3,] -0.41142514  0.03948676
## [4,] -0.03549128  3.66845615
## [5,]  3.83491347  0.92236031
## [6,]  3.14703635  2.07866787
```

```
hist(Mix1,prob=T,breaks = 10, main="The distribution of X")
```

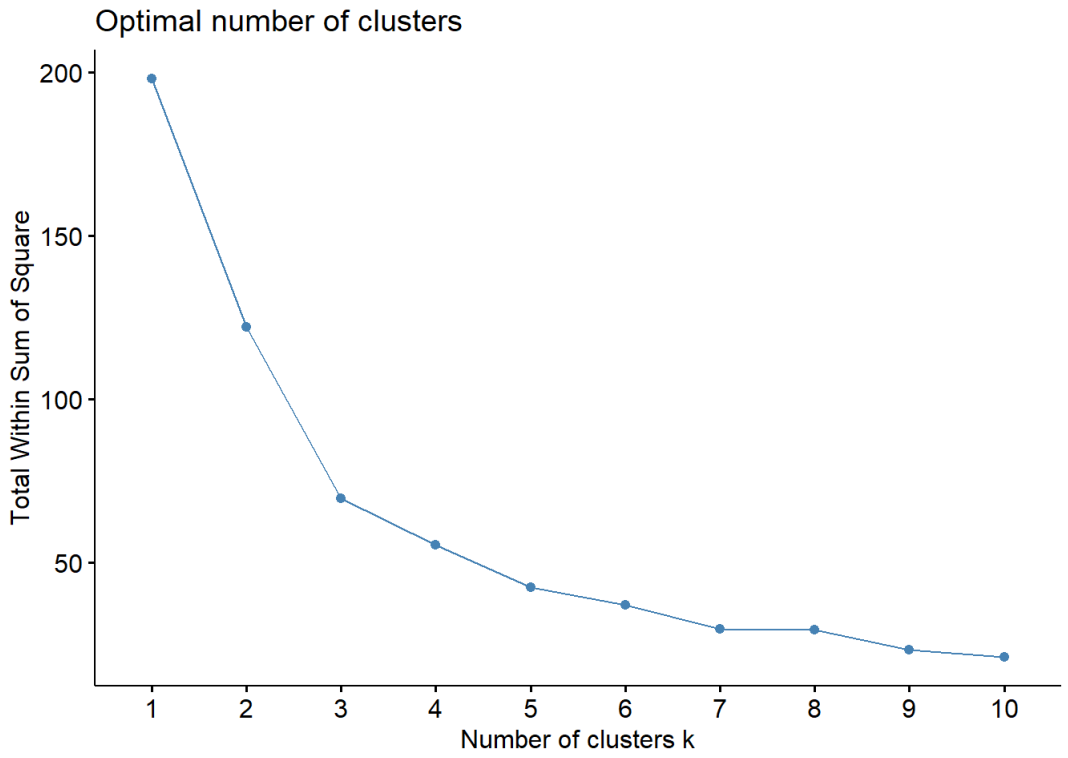


(a)

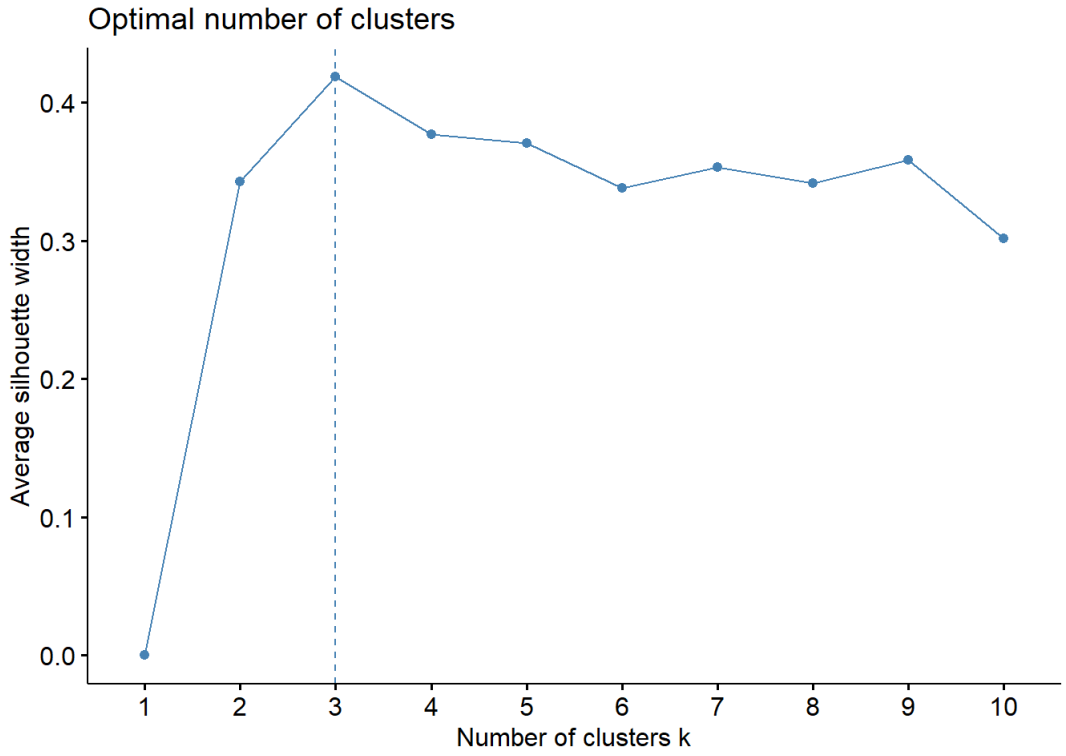
```
#(a)
bet<-numeric()
wit<-numeric()
# mixnorm<-scale(Mix)
mixnorm1<-as.data.frame(scale(Mix1))
set.seed(12)
for(i in 1:10){

  # For each k, calculate betweenss and tot.withinss
  bet[i] <- kmeans(mixnorm1, centers=i)$betweenss
  wit[i] <- kmeans(mixnorm1, centers=i)$tot.withinss

}
betw <- qplot(1:10, bet, geom=c("point", "line"),
              xlab="Number of clusters", ylab="Between-cluster sum of squares") +
  geom_line(color="#CC0033", size=1.5)+
  geom_point(size=2)+
  scale_x_continuous(breaks=seq(0, 10, 1)) +
  theme_bw()+
  labs(title = 'between')

with1 <- qplot(1:10, wit, geom=c("point", "line"),
               xlab="Number of clusters", ylab="Total within-cluster sum of squares") +
  geom_line(color="#663399", size=1.5)+
  geom_point(size=2)+
  scale_x_continuous(breaks=seq(0, 10, 1)) +
  theme_bw()+
  labs(title = 'within')

grid.arrange(betw, with1, ncol=2)
```



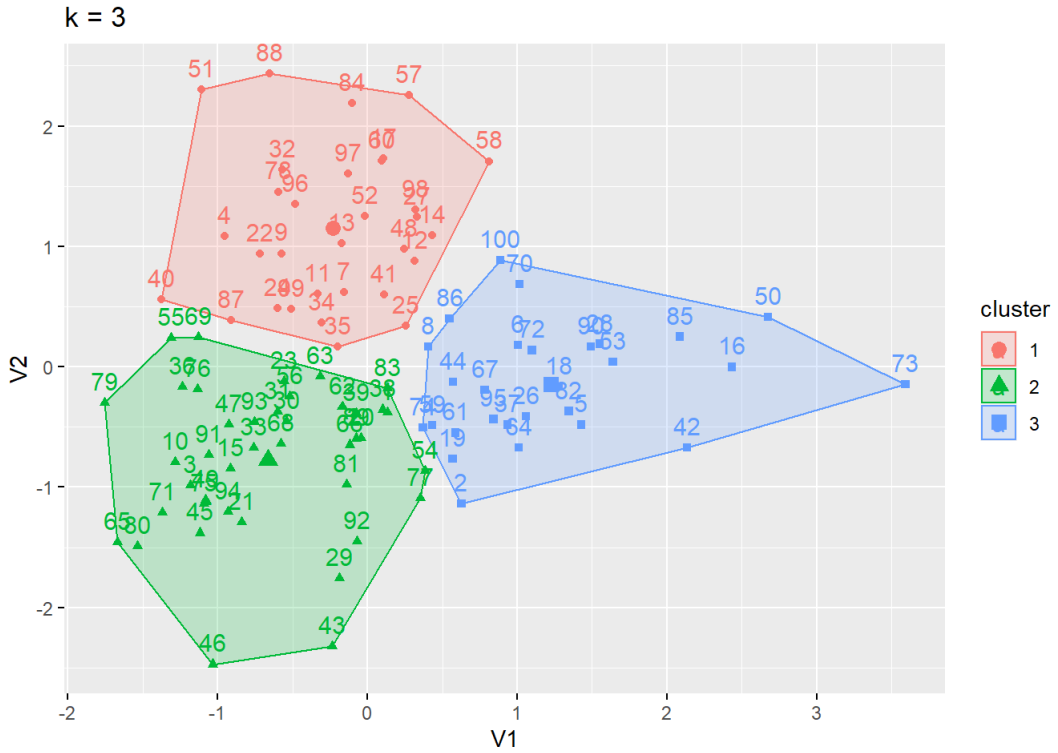

```
fviz_nbclust(mixnorm1, kmeans, method = "wss")
```




```
fviz_nbclust(mixnorm1, kmeans, method = "silhouette")
```

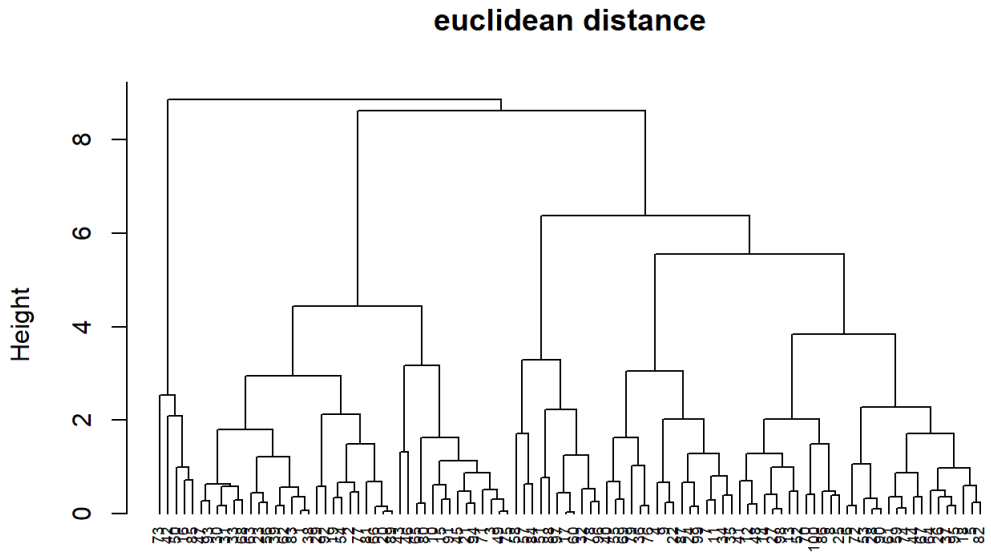



```
#choose k=3
mix_km3 <- kmeans(mixnorm1, centers=3)
pic3<-fviz_cluster(mix_km3, data = mixnorm1)+
  ggtitle("k = 3")
pic3
```

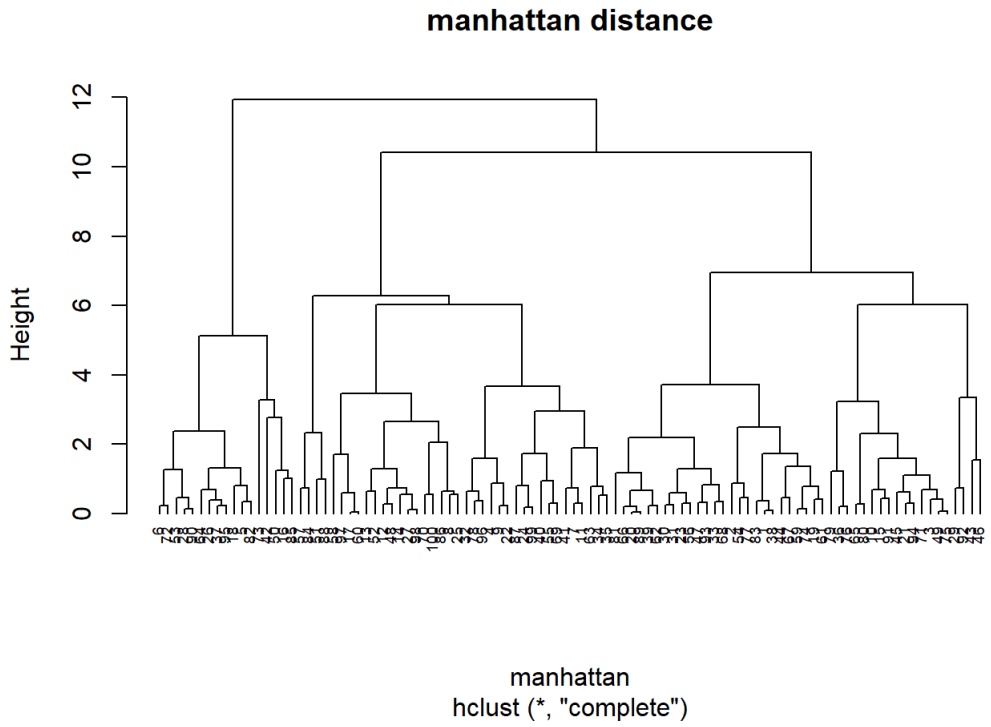


(b)

```
#(b)
Eu.dist <- dist(x = Mix1, method = "euclidean")
Ma.dist <- dist(x = Mix1, method = "manhattan")
h.Eu.cluster <- hclust(Eu.dist)
plot(h.Eu.cluster, xlab="euclidean", main='euclidean distance', hang = -1, cex = 0.6)
```



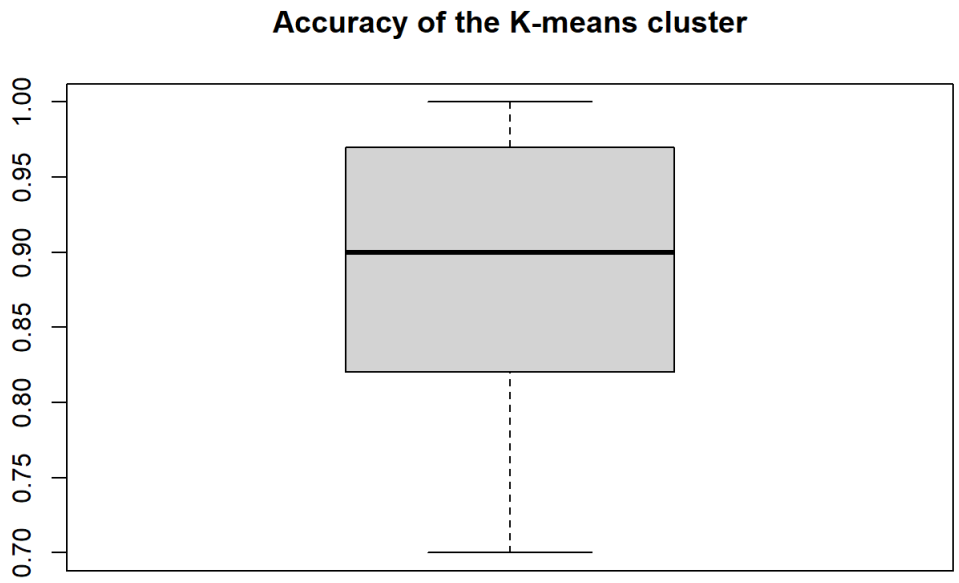

```
h.Ma.cluster <- hclust(Ma.dist)
plot(h.Ma.cluster, xlab="manhattan", main='manhattan distance', hang = -1, cex = 0.6)
```



Problem 3

(a)

```r
#(a)
set.seed(0404)
Sample<-NULL
for(i in 1:200){
  rand=NULL
  U<-NULL
  U <-runif(100,0,1)
  rand<-rep(0, 100)
  for(i in 1:length(rand)){
    if(U[i]<.6){
      rand[i]<-rnorm(1,0,1)
    }else{
      rand[i]<-rnorm(1,3,1)
    }
  }
  Sample<-rbind(Sample,rand)
}
p1<-0.6
accuracy<-rep(0,200)
for(i in 1:200){
  m=NULL
  temp=NULL
  m<-scale(Sample[i,])
  temp<-table(kmeans(m, centers=2)$cluster)[1]
  accuracy[i]<-(100-abs(temp-(100*p1)))/100
}
par(mfrow=c(1,1))
boxplot(accuracy, main='Accuracy of the K-means cluster')
```



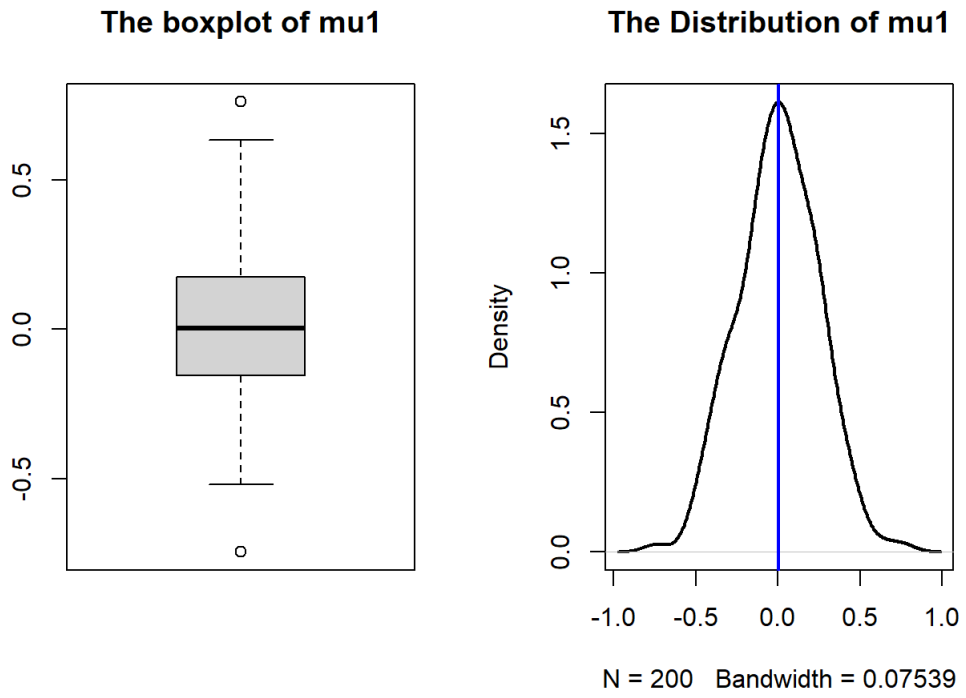
(b)

```
e_step <- function(x, mu.vector, sd.vector, alpha.vector) {
  comp1.prod <- dnorm(x, mu.vector[1], sd.vector[1]) * alpha.vector[1]
  comp2.prod <- dnorm(x, mu.vector[2], sd.vector[2]) * alpha.vector[2]
  sum.of.comps <- comp1.prod + comp2.prod
  comp1.post <- comp1.prod / sum.of.comps
  comp2.post <- comp2.prod / sum.of.comps

  sum.of.comps.ln <- log(sum.of.comps, base = exp(1))
  sum.of.comps.ln.sum <- sum(sum.of.comps.ln)

  list("loglik" = sum.of.comps.ln.sum,
       "posterior.df" = cbind(comp1.post, comp2.post))
}
m_step <- function(x, posterior.df) {
  comp1.n <- sum(posterior.df[, 1])
  comp2.n <- sum(posterior.df[, 2])

  comp1.mu <- 1/comp1.n * sum(posterior.df[, 1] * x)
  comp2.mu <- 1/comp2.n * sum(posterior.df[, 2] * x)

  comp1.var <- sum(posterior.df[, 1] * (x - comp1.mu)^2) * 1/comp1.n
  comp2.var <- sum(posterior.df[, 2] * (x - comp2.mu)^2) * 1/comp2.n

  comp1.alpha <- comp1.n / length(x)
  comp2.alpha <- comp2.n / length(x)

  list("mu" = c(comp1.mu, comp2.mu),
       "var" = c(comp1.var, comp2.var),
       "alpha" = c(comp1.alpha, comp2.alpha))
}
```

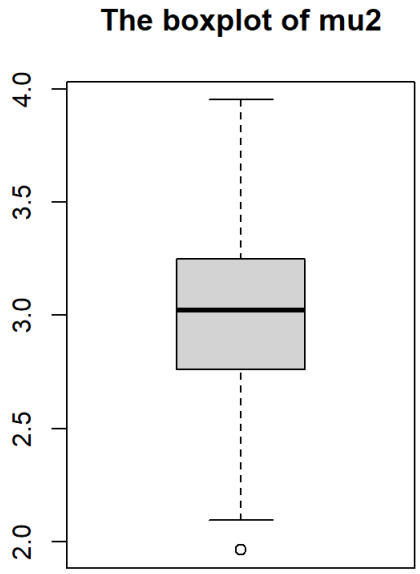
```
m1<-0; m2<-3; s1<-1; s2<-1; corx<-.6
set.seed(0404)
para<-data.frame(mu=c(m1,m2),
                 std=c(s1,s2),
                 alpha=c(.6,.4))
GMM_mu1<-NULL;GMM_mu2<-NULL;GMM_s1<-NULL;GMM_s2<-NULL;GMM_pi1<-NULL;GMM_pi2=NULL;GMMsample<-NULL
for(i in 1:200){
  # x1<-NULL
  # x2<-NULL
  U<-NULL
  U <-runif(100,0,1)
  samp<-NULL
  for(i in 1:100){
    if(U[i]<.6){
      x1<-rnorm(1,m1,s1)
      samp[i]<-x1
    }else{
      x2<-s2*corx*(x1-m1)/s1+m2+s2*rnorm(1,0,sqrt(1-corx^2))
      samp[i]<-x2
    }
  }
  GMMsample<-rbind(GMMsample,samp)
}


for(i in 1:200){
  mm=NULL;gm=NULL;gmmsamp=NULL;c1=NULL;c2=NULL;c11=NULL;c22=NULL;sum=NULL
  c=NULL;al=NULL;info=NULL
  # m.step=NULL;e.step=NULL
  mm<-GMMsample[i,]
  gm<-kmeans(mm,2)
  gmmsamp <- data.frame(x = mm, cluster = gm$cluster)
  c1<-filter(gmmsamp, gmmsamp$cluster=='1')
  c2<-filter(gmmsamp, gmmsamp$cluster=='2')
  c11<-c1 %>%
    summarize(mu = mean(x), variance = var(x), std = sd(x))
  c22<- c2%>%
    summarize(mu = mean(x), variance = var(x), std = sd(x))
  sum<-as.data.frame(rbind(c11,c22))
  c<-data.frame(cluster=c(1,2))
  al<-data.frame(size=c(nrow(c1),nrow(c2)),
                 alpha=c(nrow(c1)/(nrow(c1)+nrow(c2)),nrow(c2)/(nrow(c1)+nrow(c2))))
  info<-cbind(c,sum,al)
  for (i in 1:50) {
    if (i == 1) {
      # Initialization
      e.step <- e_step(mm, info[["mu"]], info[["std"]],
                       info[["alpha"]])
      m.step <- m_step(mm, e.step[["posterior.df"]])
      cur.loglik <- e.step[["loglik"]]
      loglik.vector <- e.step[["loglik"]]
    } else {
      # Repeat E and M steps till convergence
      e.step <- e_step(mm, m.step[["mu"]], sqrt(m.step[["var"]]),
                       m.step[["alpha"]])
      m.step <- m_step(mm, e.step[["posterior.df"]])
      loglik.vector <- c(loglik.vector, e.step[["loglik"]])

      loglik.diff <- abs((cur.loglik - e.step[["loglik"]]))
      if(loglik.diff < 1e-6) {
        break
      } else {
        cur.loglik <- e.step[["loglik"]]
      }
    }
  }
  if(m.step$mu[1]<1){
    GMM_mu1<-append(GMM_mu1,m.step$mu[1])
    GMM_s1<-append(GMM_s1,m.step$var[1])
    GMM_pi1<-append(GMM_pi1,m.step$alpha[1])
  }else{
    GMM_mu2<-append(GMM_mu2,m.step$mu[1])
```

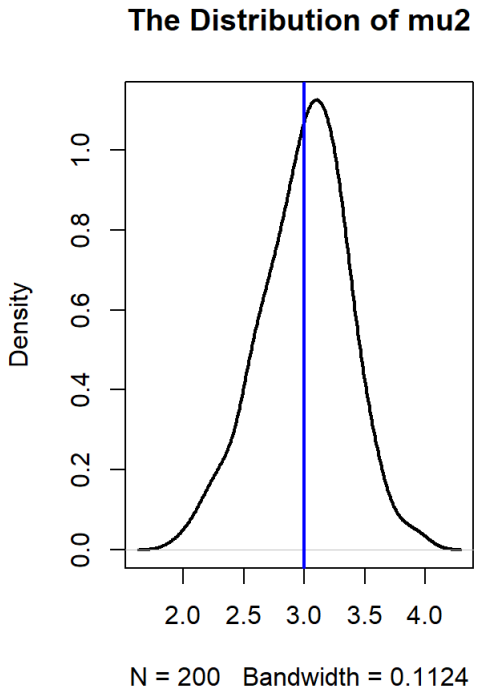
```
    GMM_s2<-append(GMM_s2,m.step$var[1])
    GMM_pi2<-append(GMM_pi2,m.step$alpha[1])
  }
  if(m.step$mu[2]<1){
    GMM_mu1<-append(GMM_mu1,m.step$mu[2])
    GMM_s1<-append(GMM_s1,m.step$var[2])
    GMM_pi1<-append(GMM_pi1,m.step$alpha[2])
  }else{
    GMM_mu2<-append(GMM_mu2,m.step$mu[2])
    GMM_s2<-append(GMM_s2,m.step$var[2])
    GMM_pi2<-append(GMM_pi2,m.step$alpha[2])
  }

}
```

```
#plots
{par(mfrow=c(1,2))
  boxplot(GMM_mu1, main='The boxplot of mu1')
  plot(density(GMM_mu1), main='The Distribution of mu1', lwd=2)
  abline(v=mean(GMM_mu1),lwd=2,col='blue')}
```

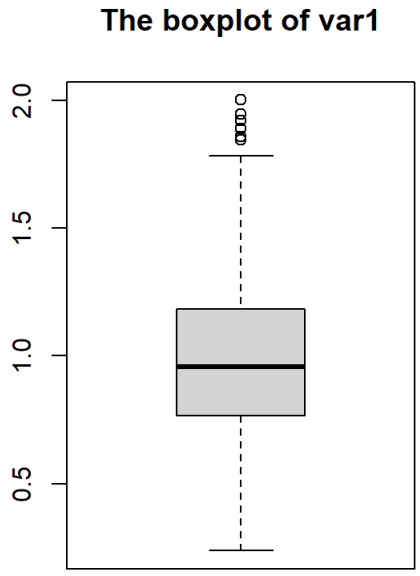


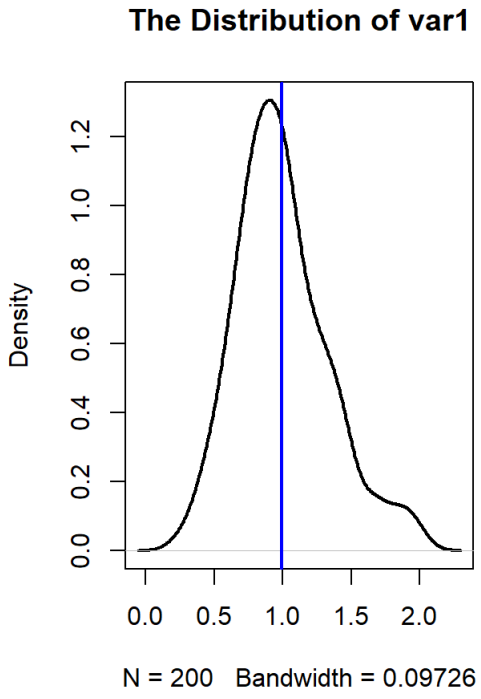

```
{par(mfrow=c(1,2))
  boxplot(GMM_mu2, main='The boxplot of mu2')
  plot(density(GMM_mu2), main='The Distribution of mu2', lwd=2)
abline(v=mean(GMM_mu2),lwd=2,col='blue')}
```



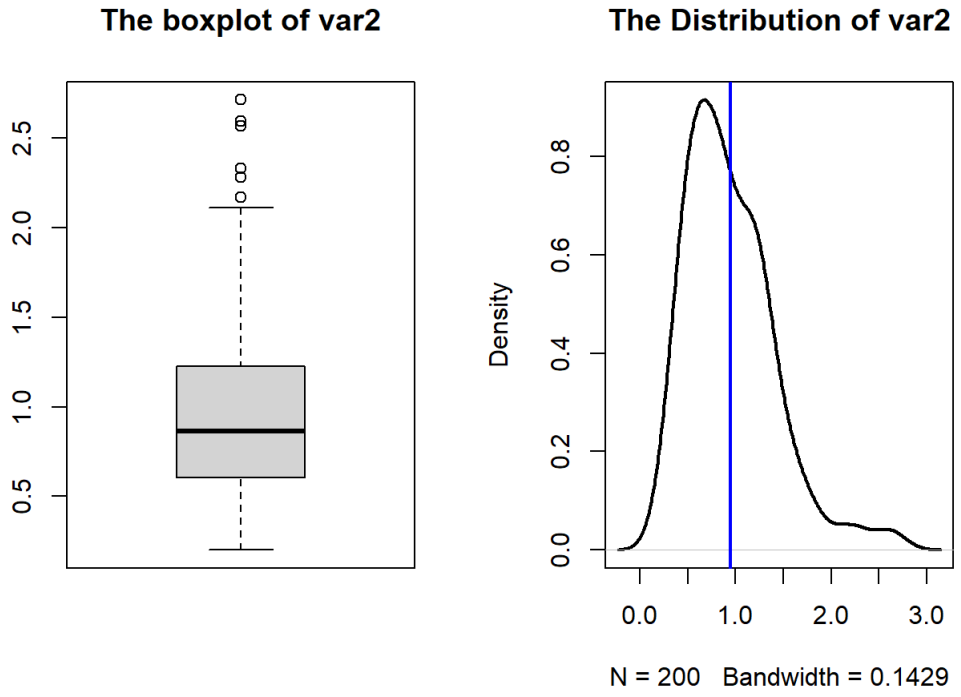



```
{par(mfrow=c(1,2))
  boxplot(GMM_s1, main='The boxplot of var1')
  plot(density(GMM_s1), main='The Distribution of var1', lwd=2)
abline(v=mean(GMM_s1),lwd=2,col='blue')}
```

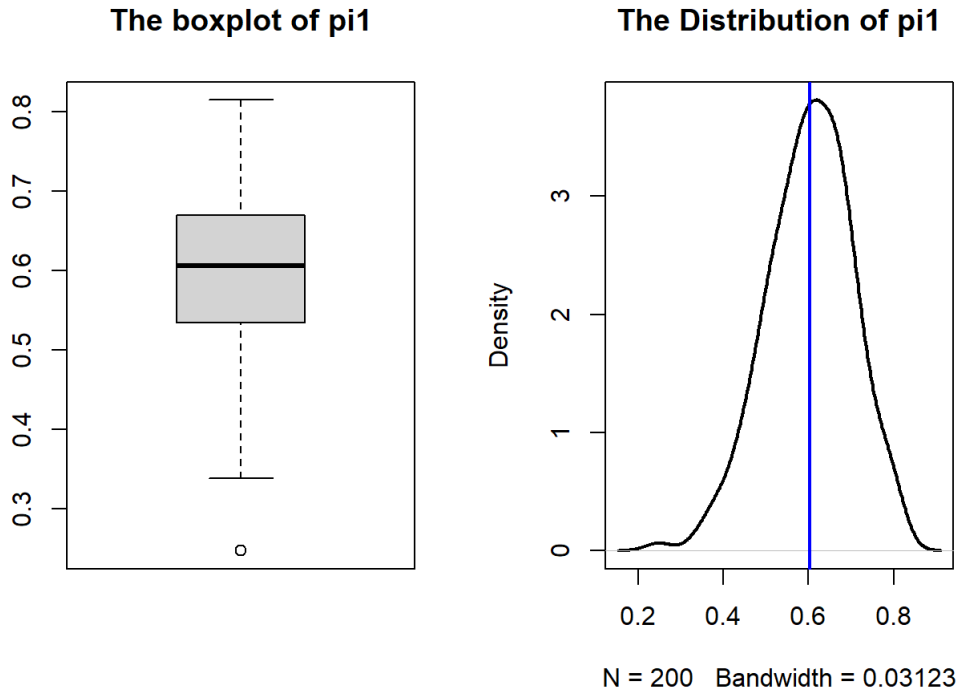




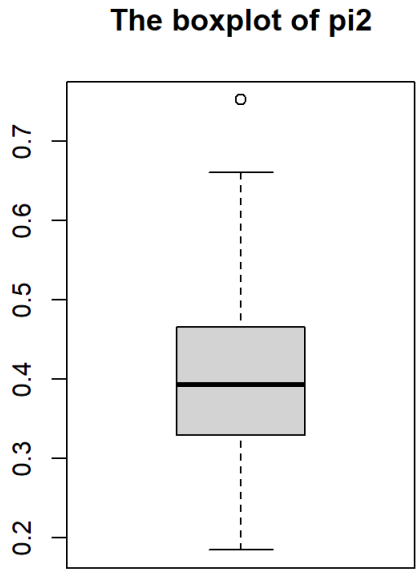

```
{par(mfrow=c(1,2))
  boxplot(GMM_s2, main='The boxplot of var2')
  plot(density(GMM_s2), main='The Distribution of var2', lwd=2)
abline(v=mean(GMM_s2),lwd=2,col='blue')}
```

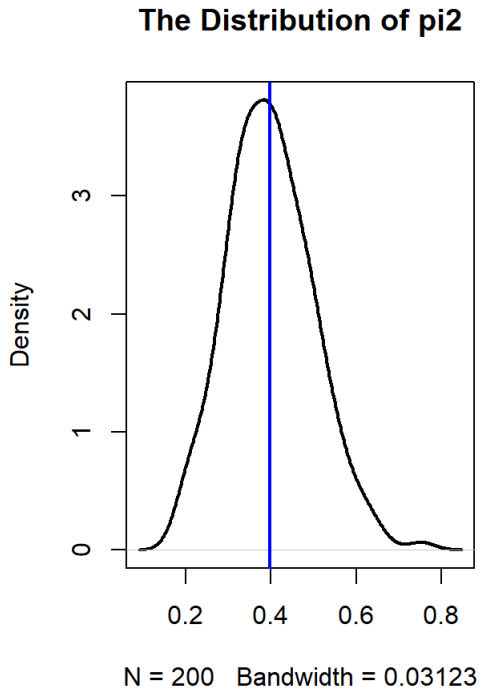



```
{par(mfrow=c(1,2))
  boxplot(GMM_pi1, main='The boxplot of pi1')
  plot(density(GMM_pi1), main='The Distribution of pi1', lwd=2)
abline(v=mean(GMM_pi1),lwd=2,col='blue')}
```




```
{par(mfrow=c(1,2))
  boxplot(GMM_pi2, main='The boxplot of pi2')
  plot(density(GMM_pi2), main='The Distribution of pi2', lwd=2)
abline(v=mean(GMM_pi2),lwd=2,col='blue')}
```






```
#95%CIs
xmu1<-mean(GMM_mu1) #sample mean
s11<-sd(GMM_mu1)/sqrt(length(GMM_mu1)) #standard_error
m1CI_95<-c(xmu1-1.96*s11,xmu1+1.96*s11)

xmu2<-mean(GMM_mu2) #sample mean
s22<-sd(GMM_mu2)/sqrt(length(GMM_mu2)) #standard_error
m2CI_95<-c(xmu2-1.96*s22,xmu2+1.96*s22)

xvar1<-mean(GMM_s1) #sample mean
s33<-sd(GMM_s1)/sqrt(length(GMM_s1)) #standard_error
var1CI_95<-c(xvar1-1.96*s33,xvar1+1.96*s33)

xvar2<-mean(GMM_s2) #sample mean
s44<-sd(GMM_s2)/sqrt(length(GMM_s2)) #standard_error
var2CI_95<-c(xvar2-1.96*s44,xvar2+1.96*s44)

xpi1<-mean(GMM_pi1) #sample mean
s55<-sd(GMM_pi1)/sqrt(length(GMM_pi1)) #standard_error
pi1CI_95<-c(xpi1-1.96*s55,xpi1+1.96*s55)

xpi2<-mean(GMM_pi2) #sample mean
s66<-sd(GMM_pi2)/sqrt(length(GMM_pi2)) #standard_error
pi2CI_95<-c(xpi2-1.96*s66,xpi2+1.96*s66)

est<-data.frame(original_paraneter=c(0,3,1,1,0.6,0.4),
                parameter=c(xmu1,xmu2,xvar1,xvar2,xpi1,xpi2),
                lower_95CI=c(m1CI_95[1],m2CI_95[1],var1CI_95[1],var2CI_95[1],pi1CI_95[1],pi2CI_95[1]),
                upper_95CI=c(m1CI_95[2],m2CI_95[2],var1CI_95[2],var2CI_95[2],pi1CI_95[2],pi2CI_95[2]))
est
```

```
##   original_paraneter   parameter  lower_95CI upper_95CI
## 1                0.0 0.003460573 -0.03041167 0.03733281
## 2                3.0 2.998622479  2.94867000 3.04857496
## 3                1.0 0.994504965  0.94769119 1.04131874
## 4                1.0 0.947407623  0.88271482 1.01210042
## 5                0.6 0.602091915  0.58821577 0.61596806
## 6                0.4 0.397908085  0.38403194 0.41178423
```