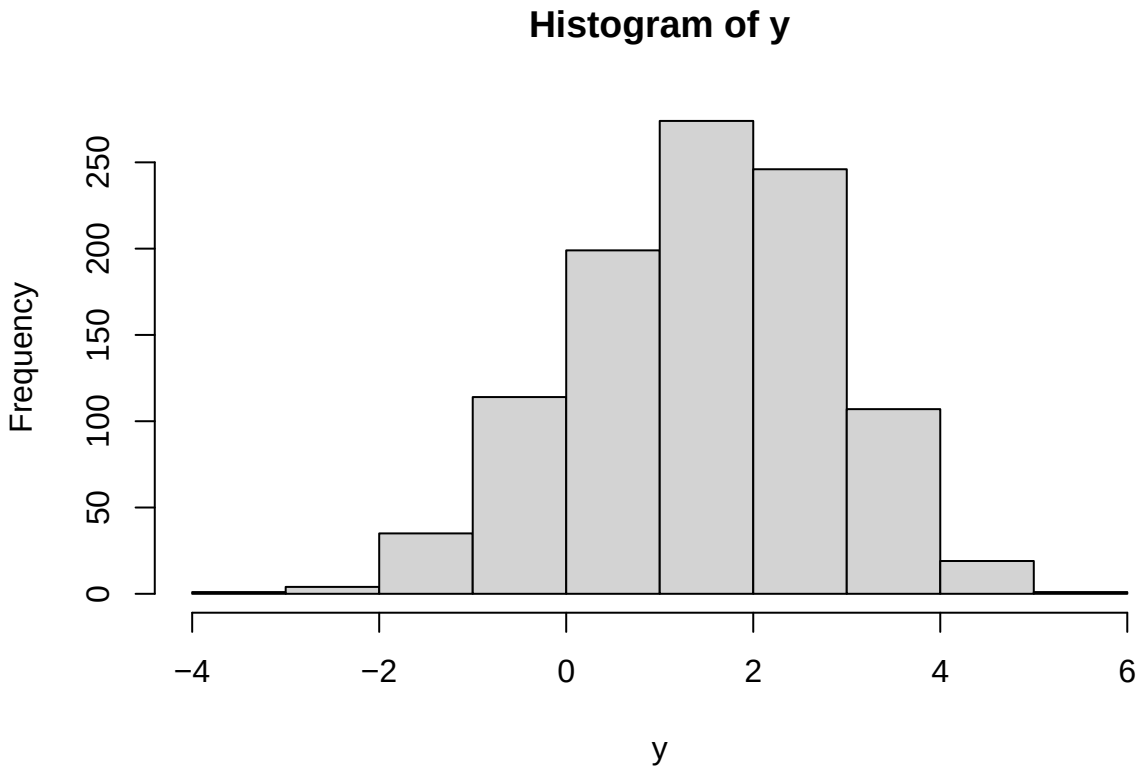
lecture_7

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
##########################################
### GRID SEARCH FOR SIMPLE MIXTURE MODEL ###
##########################################

## input data:  (mixture of normals data with true mus 0 and 2 and alpha=0.75)
data <- read.table("data/normnorm1.txt")
y <- data[,2]
n <- length(y)
hist(y)
```

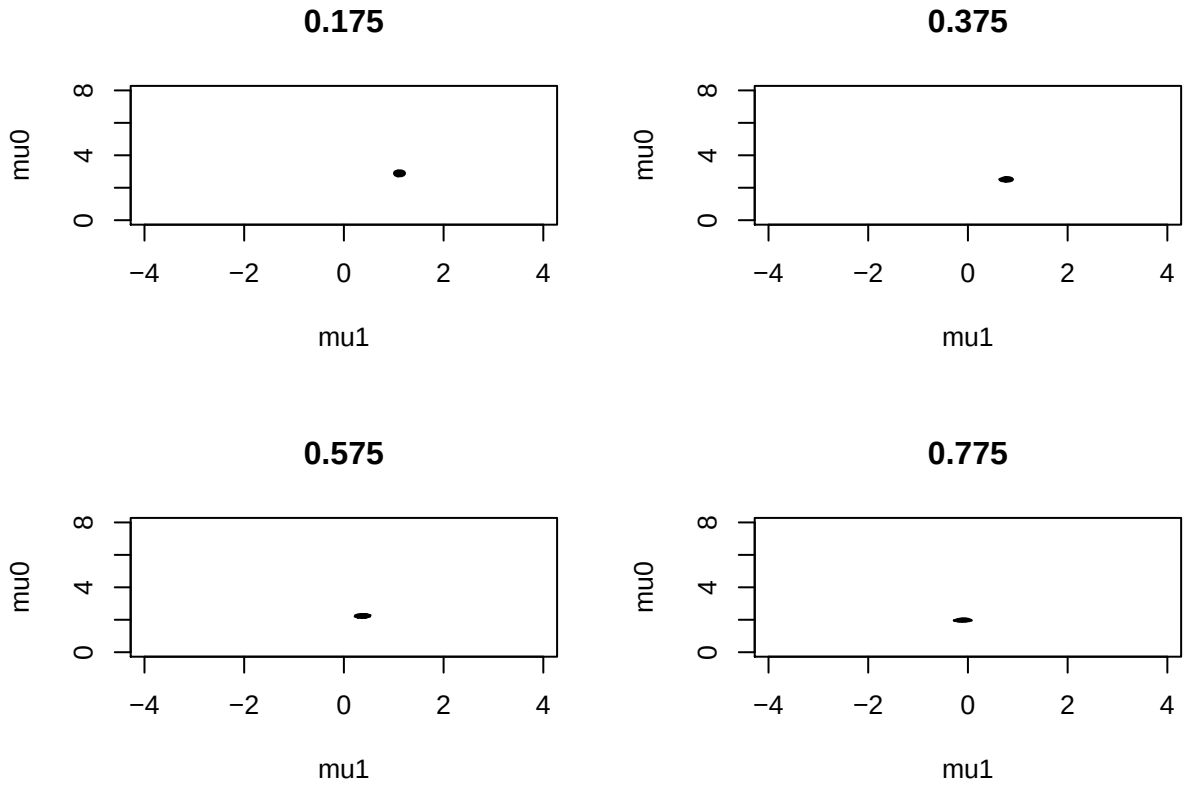



```
## function for calculating log-likelihood:

loglik.mix <- function(alpha,mu0,mu1,x){
   phi0 <- dnorm(x,mu0,1)
   phi1 <- dnorm(x,mu1,1)
```

```
   loglik <- sum(log(alpha*phi1 + (1-alpha)*phi0))
   loglik
}

## calculating likelihood over range of alpha, mu1, mu0 (and finding maximum)

alpharange <- ppoints(20)   # alpha between 0 and 1
mu0range <- ppoints(100)*8-4  # mu0 between -4 and 4
mu1range <- ppoints(100)*8  # mu1 between 0 and 8
z <- array(NA,dim=c(20,100,100))
bestz <- -Inf  # initializing optimal values
for (i in 1:20){
  for (j in 1:100){
    for (k in 1:100){
      curalpha <- alpharange[i]
      curmu0 <- mu0range[j]
      curmu1 <- mu1range[k]
      if(curmu0 < curmu1){
        z[i,j,k] <- loglik.mix(curalpha,curmu0,curmu1,y)
        if (z[i,j,k] > bestz){
          bestalpha <- alpharange[i]
          bestmu0 <- curmu0
          bestmu1 <- curmu1
          bestz <- z[i,j,k]
        }
      }
      if(curmu0 >= curmu1){  # for unique solution, constrain mu1 > mu0
        z[i,j,k] <- -Inf
      }
    }
  }
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
```

```
z <- exp(z-max(z))

## plotting slices of likelihood as 2-d contour:

par(mfrow=c(2,2))
contour(mu0range,mu1range,z[4,,],xlab="mu1",ylab="mu0",main=as.character(alpharange[4]),drawlabels=F)
contour(mu0range,mu1range,z[8,,],xlab="mu1",ylab="mu0",main=as.character(alpharange[8]),drawlabels=F)
contour(mu0range,mu1range,z[12,,],xlab="mu1",ylab="mu0",main=as.character(alpharange[12]),drawlabels=F)
contour(mu0range,mu1range,z[16,,],xlab="mu1",ylab="mu0",main=as.character(alpharange[16]),drawlabels=F)
```




```
bestalpha
```

```
## [1] 0.675
```

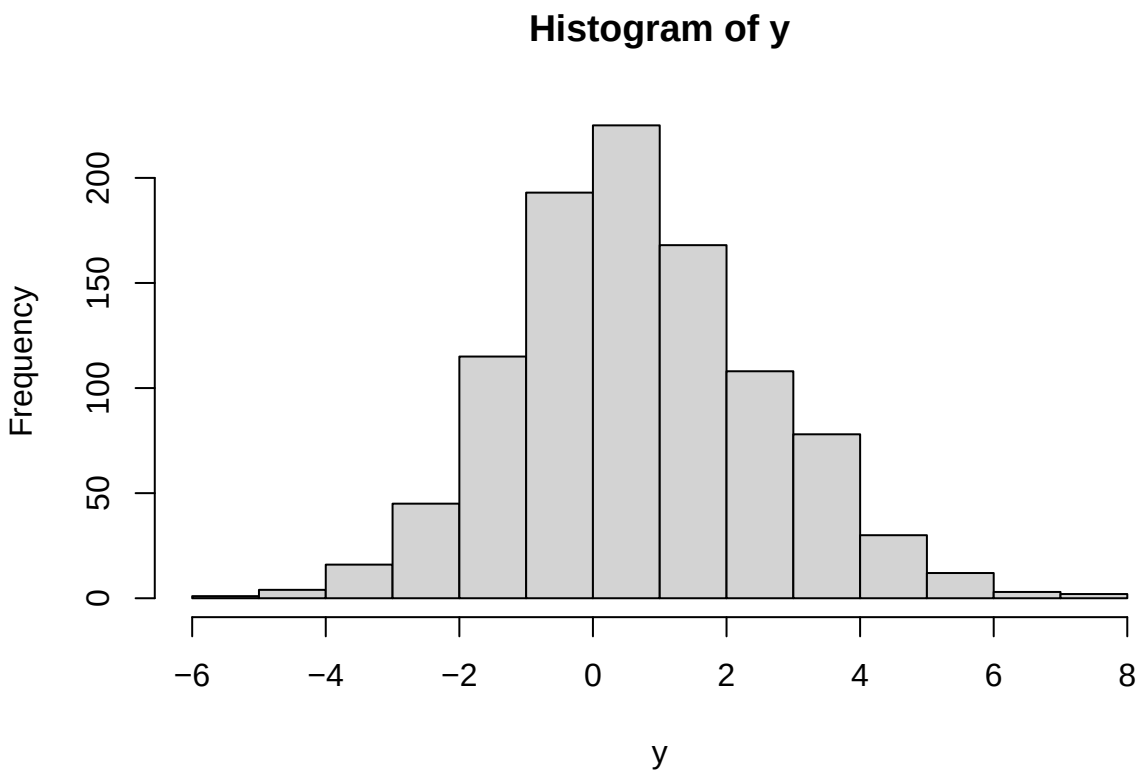
```
bestmu0
```

```
## [1] 0.2
```

```
bestmu1
```

```
## [1] 2.12
```

```
##############################################
##### EM ALGORITHM FOR A MIXTURE MODEL #####
##############################################

## input data: mixture of normals with true mu0=0,mu1=1,sigsq0=1,sigsq1=4,alpha=0.75
data <- read.table("data/normnorm2.txt")
y <- data[,2]
par(mfrow=c(1,1))
hist(y)
```




```
n <- length(y)

## Expectation function:
Estep <- function(alpha,mu0,mu1,sigsq0,sigsq1){
  ind <- rep(NA,n)
  for (i in 1:n){
    prob0 <- (1-alpha)*dnorm(y[i],mean=mu0,sd=sqrt(sigsq0))
    prob1 <- alpha*dnorm(y[i],mean=mu1,sd=sqrt(sigsq1))
    ind[i] <- prob1/(prob0+prob1)
  }
  ind
}

## Maximization function
Mstep <- function(ind){
  alpha <- sum(ind)/n
  mu1 <- sum(ind*y)/sum(ind)
  mu0 <- sum((1-ind)*y)/sum(1-ind)
  sigsq1 <- sum(ind*((y-mu1)^2))/sum(ind)
  sigsq0 <- sum((1-ind)*((y-mu0)^2))/sum(1-ind)
  c(alpha,mu0,mu1,sigsq0,sigsq1)
}

## Starting values for EM algorithm:
curalpha <- 0.5
curmu0 <- -1
curmu1 <- 1
cursigsq0 <- 1
cursigsq1 <- 1
```

```r
itermat <- c(curalpha,curmu0,curmu1,cursigsq0,cursigsq1)
```

```r
## Running EM algorithm
diff <- 1
numiters <- 1
while (diff > 0.000001 || numiters <= 100){
  numiters <- numiters+1
  curind <- Estep(curalpha,curmu0,curmu1,cursigsq0,cursigsq1)
  curparam <- Mstep(curind)
  curalpha <- curparam[1]
  curmu0 <- curparam[2]
  curmu1 <- curparam[3]
  cursigsq0 <- curparam[4]
  cursigsq1 <- curparam[5]
  itermat <- rbind(itermat,curparam)
  diff <- max(abs(itermat[numiters,]-itermat[numiters-1,]))
  print (numiters)
}
```

```r
parametertext <- c("alpha","mu0","mu1","sigsq0","sigsq1")
par(mfrow=c(2,3))
for (i in 1:5){
  plot(1:length(itermat[,1]),itermat[,i],main=parametertext[i],xlab="Iterations",ylab="Value")
}

lastiter <- length(itermat[,1])
itermat[lastiter,]
```

```
## [1]  0.71703897 -0.06442663  0.98226740  1.20671276  4.23380604
```

```r
## EM code above with different Starting values:
curalpha <- 0.5
curmu0 <- -10
curmu1 <- 10
cursigsq0 <- 2
cursigsq1 <- 2
itermat <- c(curalpha,curmu0,curmu1,cursigsq0,cursigsq1)

###############################################
#### EM ALGORITHM FOR BASEBALL EXAMPLE #####
###############################################

#Reading in Data:
data <- read.table("data/hitters.post1970.txt",header=T,sep="\t")
dim(data)
```

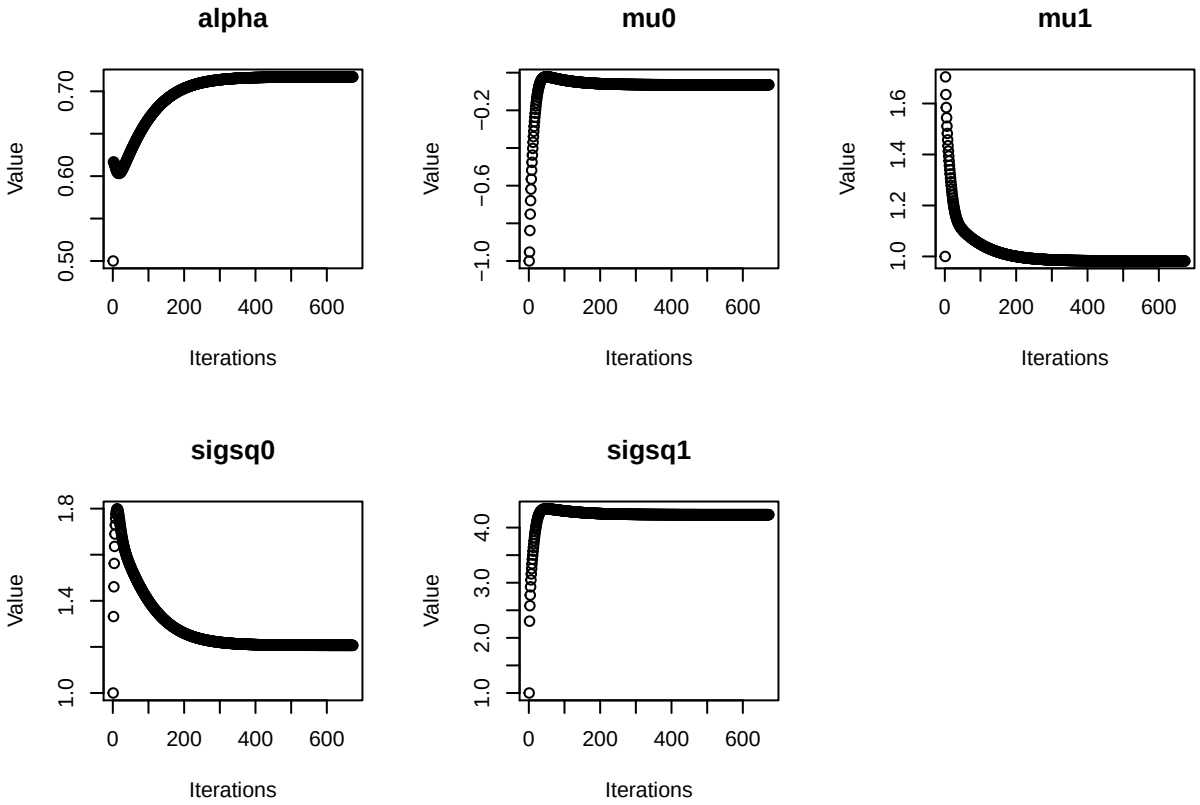
```
## [1] 20810    26
```

```r
data <- data[data$AB>100,]
dim(data)
```

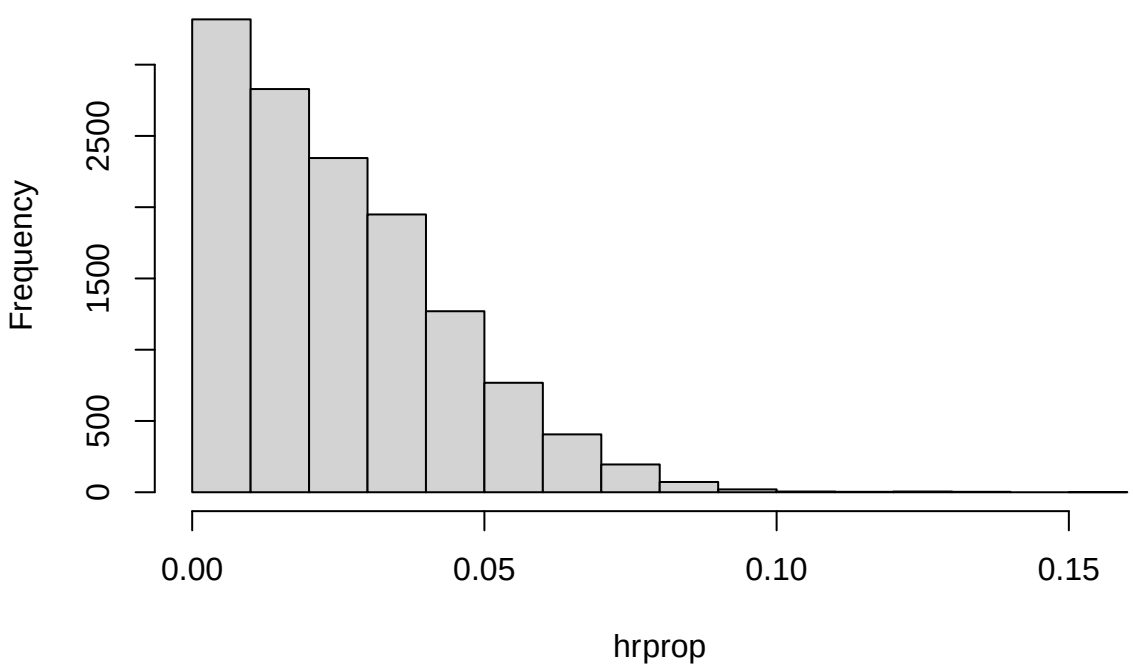
```
## [1] 13189    26
```

```r
hr <- data$HR
ab <- data$AB
player <- data$player
year <- data$year
```

```r
#Calculating homerun proportion:
hrprop <- hr/ab
par(mfrow=c(1,1))
```




```r
hist(hrprop)
```

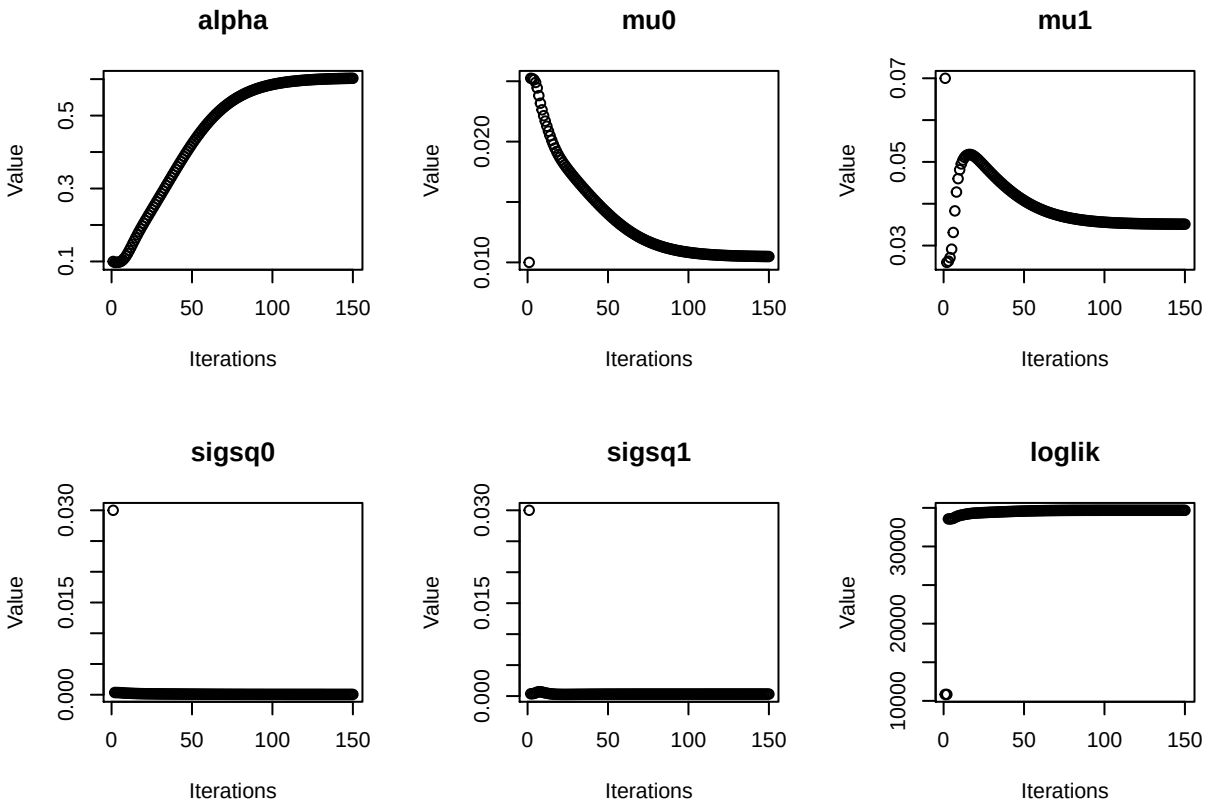



```r
#Expectation function
Estep <- function(y,alpha,mu0,mu1,sigsq0,sigsq1){
  n <- length(y)
  ind <- rep(NA,n)
  for (i in 1:n){
    prob0 <- (1-alpha)*dnorm(y[i],mean=mu0,sd=sqrt(sigsq0))
    prob1 <- alpha*dnorm(y[i],mean=mu1,sd=sqrt(sigsq1))
    ind[i] <- prob1/(prob0+prob1)
  }
  ind
}

#Maximization function
Mstep <- function(y,ind){
  n <- length(y)
  alpha <- sum(ind)/n
  mu1 <- sum(ind*y)/sum(ind)
  mu0 <- sum((1-ind)*y)/sum(1-ind)
  sigsq1 <- sum(ind*((y-mu1)^2))/sum(ind)
  sigsq0 <- sum((1-ind)*((y-mu0)^2))/sum(1-ind)
  c(alpha,mu0,mu1,sigsq0,sigsq1)
}

##observed data loglikelihood function
loglik.mix <- function(y,ind,alpha,mu0,mu1,sigsq0,sigsq1){
  loglik <- sum(log(alpha*dnorm(y,mu1,sqrt(sigsq1))+(1-alpha)*dnorm(y,mu0,sqrt(sigsq0))))
  loglik
}
```

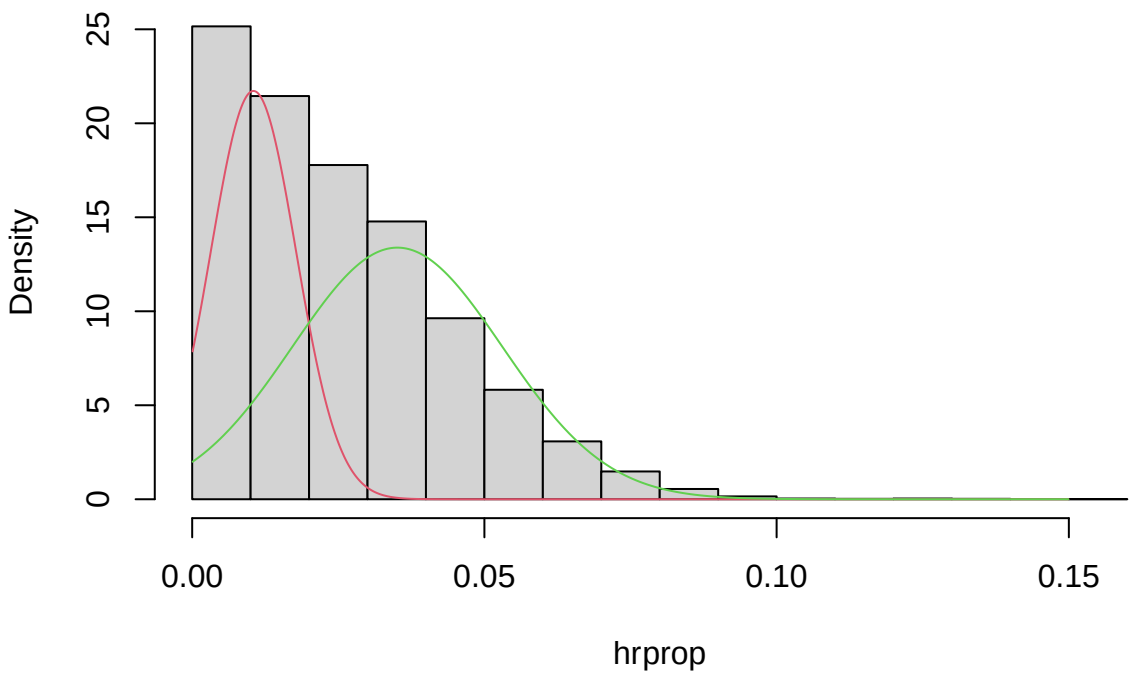
```r
#Running EM iterations
curalpha <- 0.1
curmu0 <- 0.01
curmu1 <- 0.07
cursigsq0 <- 0.03
cursigsq1 <- 0.03
curind <- Estep(hrprop,curalpha,curmu0,curmu1,cursigsq0,cursigsq1)
loglik <- loglik.mix(hrprop,curind,curalpha,curmu0,curmu1,cursigsq0,cursigsq1)
itermat <- c(curalpha,curmu0,curmu1,cursigsq0,cursigsq1,loglik)
diff <- 1
numiters <- 1
```

```r
while (diff > 0.001 || numiters <= 100){
  curind <- Estep(hrprop,curalpha,curmu0,curmu1,cursigsq0,cursigsq1)
  curparam <- Mstep(hrprop,curind)
  curalpha <- curparam[1]
  curmu0 <- curparam[2]
  curmu1 <- curparam[3]
  cursigsq0 <- curparam[4]
  cursigsq1 <- curparam[5]
  itermat <- rbind(itermat,c(curparam,loglik))
  loglik <- loglik.mix(hrprop,curind,curalpha,curmu0,curmu1,cursigsq0,cursigsq1)
  numiters <- numiters + 1
  diff <- max(abs(itermat[numiters,]-itermat[numiters-1,]))
  print (c(numiters,loglik))
}
```

```r
parametertext <- c("alpha","mu0","mu1","sigsq0","sigsq1","loglik")
par(mfrow=c(2,3))
for (i in 1:6){
  plot(1:numiters,itermat[,i],main=parametertext[i],xlab="Iterations",ylab="Value")
}
```




```
# plotting fitted mixture density
finalparam<-itermat[numiters,]
alpha <- finalparam[1]
mu0 <- finalparam[2]
mu1 <- finalparam[3]
sigsq0 <- finalparam[4]
sigsq1 <- finalparam[5]
par(mfrow=c(1,1))
hist(hrprop,prob=T)
x <- ppoints(1000)*0.15
y1 <- (1-alpha)*dnorm(x,mu0,sqrt(sigsq0))
y2 <- alpha*dnorm(x,mu1,sqrt(sigsq1))
lines(x,y1,col=2)
lines(x,y2,col=3)
```

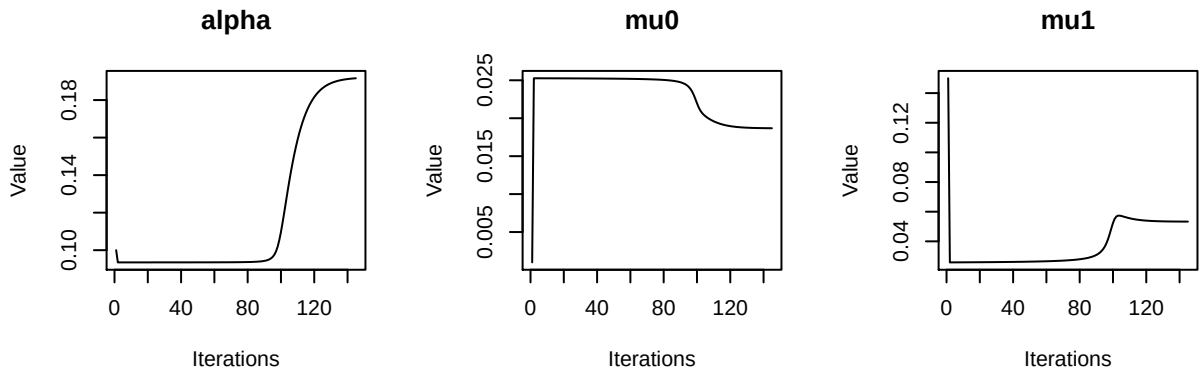


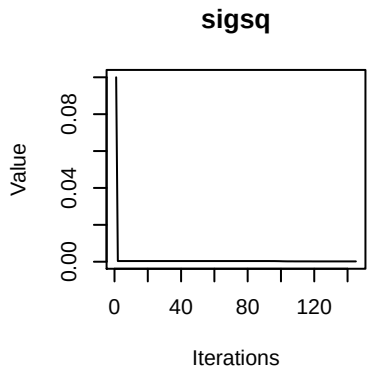

```
#EM algorithm for equal-variance model
Estep2 <- function(y,alpha,mu0,mu1,sigsq){
  n <- length(y)
  ind <- rep(NA,n)
  for (i in 1:n){
    prob0 <- (1-alpha)*dnorm(y[i],mean=mu0,sd=sqrt(sigsq))
    prob1 <- alpha*dnorm(y[i],mean=mu1,sd=sqrt(sigsq))
    ind[i] <- prob1/(prob0+prob1)
  }
  ind
}
Mstep2 <- function(y,ind){
  n <- length(y)
  alpha <- sum(ind)/n
  mu1 <- sum(ind*y)/sum(ind)
  mu0 <- sum((1-ind)*y)/sum(1-ind)
  sigsq <- sum(ind*((y-mu1)^2))
  sigsq <- sigsq+sum((1-ind)*((y-mu0)^2))
  sigsq <- sigsq/n
  c(alpha,mu0,mu1,sigsq)
}
##observed data loglikelihood function for equal variance model
loglik.mix2 <- function(y,ind,alpha,mu0,mu1,sigsq){
  loglik <- sum(log(alpha*dnorm(y,mu1,sqrt(sigsq))+(1-alpha)*dnorm(y,mu0,sqrt(sigsq))))
  loglik
}
curalpha <- 0.1
curmu0 <- 0.001
curmu1 <- 0.15
```

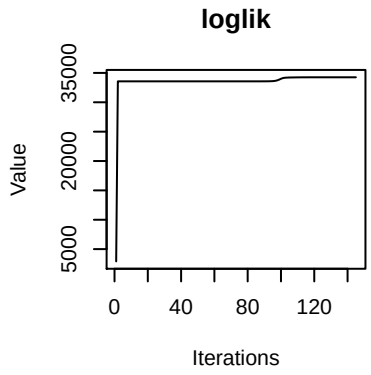
```
cursigsq <- 0.1
curind <- Estep2(hrprop,curalpha,curmu0,curmu1,cursigsq)
loglik <- loglik.mix2(hrprop,curind,curalpha,curmu0,curmu1,cursigsq)
itermat2 <- c(curalpha,curmu0,curmu1,cursigsq,loglik)
diff <- 1
numiters <- 1
```

```
while (diff > 0.001 || numiters <= 100){
  curind <- Estep2(hrprop,curalpha,curmu0,curmu1,cursigsq)
  curparam <- Mstep2(hrprop,curind)
  curalpha <- curparam[1]
  curmu0 <- curparam[2]
  curmu1 <- curparam[3]
  cursigsq <- curparam[4]
  loglik <- loglik.mix2(hrprop,curind,curalpha,curmu0,curmu1,cursigsq)
  itermat2 <- rbind(itermat2,c(curparam,loglik))
  numiters <- numiters + 1
  diff <- max(abs(itermat2[numiters,]-itermat2[numiters-1,]))
  print (c(numiters,loglik))
}
```

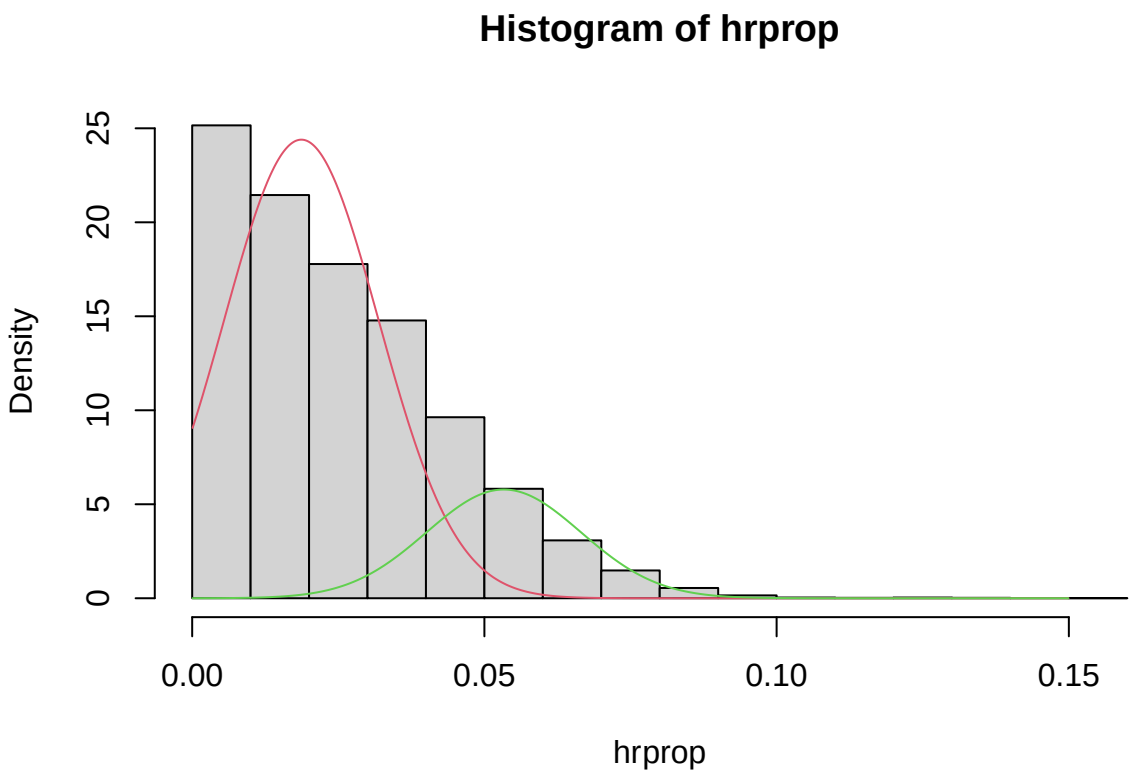
```
#Tracking iterations
parametertext <- c("alpha","mu0","mu1","sigsq","loglik")
par(mfrow=c(2,3))
for (i in 1:5){
  plot(1:numiters,itermat2[,i],type="l",main=parametertext[i],xlab="Iterations",ylab="Value")
}

# plotting equal-variances fitted mixture density
finalparam<-itermat2[numiters,]
alpha <- finalparam[1]
mu0 <- finalparam[2]
mu1 <- finalparam[3]
sigsq <- finalparam[4]
par(mfrow=c(1,1))
```



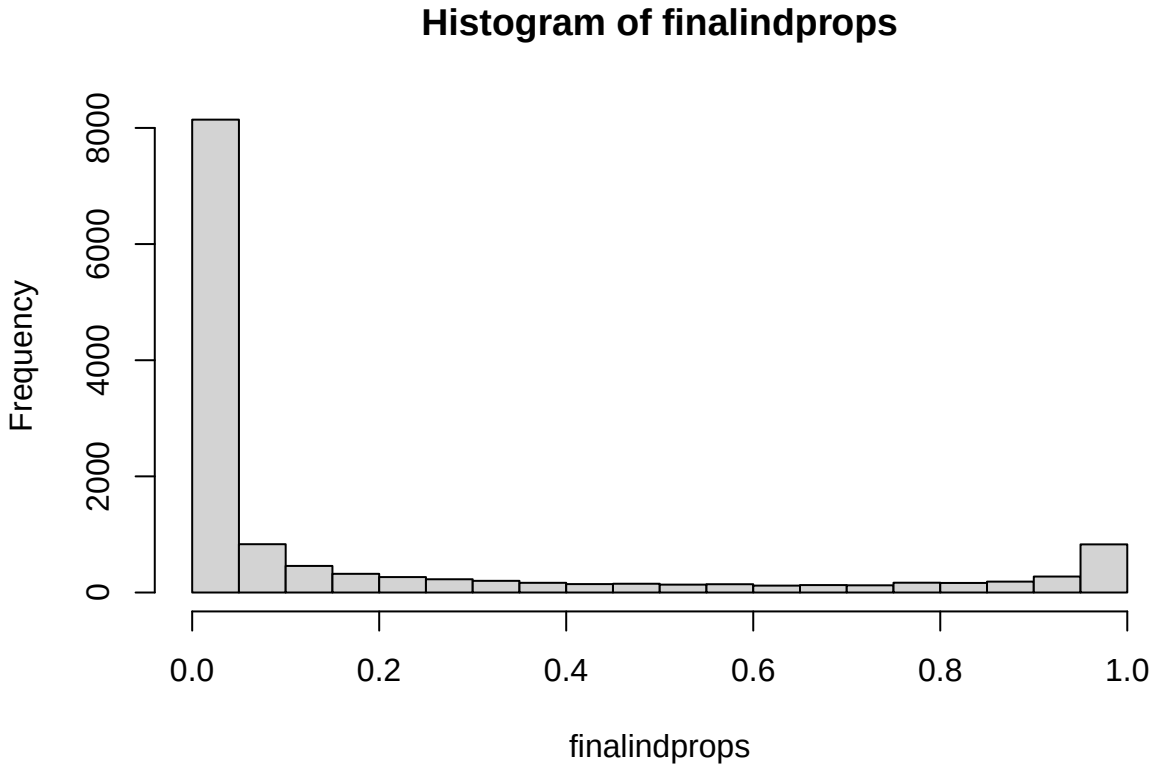





```
hist(hrprop,prob=T)
x <- ppoints(1000)*0.15
y1 <- (1-alpha)*dnorm(x,mu0,sqrt(sigsq))
y2 <- alpha*dnorm(x,mu1,sqrt(sigsq))
lines(x,y1,col=2)
lines(x,y2,col=3)
```




```
#Getting Individual probabilities for each player
finalindprops <- Estep2(hrprop,alpha,mu0,mu1,sigsq)

hist(finalindprops)
```



```
sum(finalindprops > 0.9999)
```

```
## [1] 39
```

```
players.topHR<-data[finalindprops > 0.9999,1:5]
players.topHR
```

```
##        playerID yearID stint teamID lgID
## 2     aaronha01   1971     1    ATL   NL
## 4     aaronha01   1973     1    ATL   NL
## 697   bagweje01   1994     1    HOU   NL
## 1183  belleal01   1995     1    CLE   AL
## 1795  bondsba01   1994     1    SFN   NL
## 1800  bondsba01   1999     1    SFN   NL
## 1801  bondsba01   2000     1    SFN   NL
## 1802  bondsba01   2001     1    SFN   NL
## 1803  bondsba01   2002     1    SFN   NL
## 1804  bondsba01   2003     1    SFN   NL
## 1805  bondsba01   2004     1    SFN   NL
## 6338  gamblos01   1979     2    NYA   AL
## 6923  gonzalu01   2001     1    ARI   NL
## 7251  griffke02   1994     1    SEA   AL
## 7253  griffke02   1996     1    SEA   AL
## 7254  griffke02   1997     1    SEA   AL
## 8411   hillgl01   2000     2    NYA   AL
## 10060 kingmda01   1979     1    CHN   NL
## 11168 lopezja01   2003     1    ATL   NL
## 12317 mcgwima01   1992     1    OAK   AL
## 12320 mcgwima01   1995     1    OAK   AL
## 12321 mcgwima01   1996     1    OAK   AL
## 12322 mcgwima01   1997     1    OAK   AL
## 12323 mcgwima01   1997     2    SLN   NL
## 12324 mcgwima01   1998     1    SLN   NL
## 12325 mcgwima01   1999     1    SLN   NL
## 12326 mcgwima01   2000     1    SLN   NL
## 12327 mcgwima01   2001     1    SLN   NL
## 12872 mitchke01   1994     1    CIN   NL
## 14957 phelpke01   1988     2    NYA   AL
## 16242 rodrial01   2002     1    TEX   AL
## 17746  sosasa01   1998     1    CHN   NL
## 17747  sosasa01   1999     1    CHN   NL
## 17749  sosasa01   2001     1    CHN   NL
## 17997 stargwi01   1971     1    PIT   NL
## 18657 thomafr04   1994     1    CHA   AL
## 18706 thomeji01   2001     1    CLE   AL
## 18707 thomeji01   2002     1    CLE   AL
## 20237 willima04   1994     1    SFN   NL
```