Mixture Models

You ZUO

2019/10/14

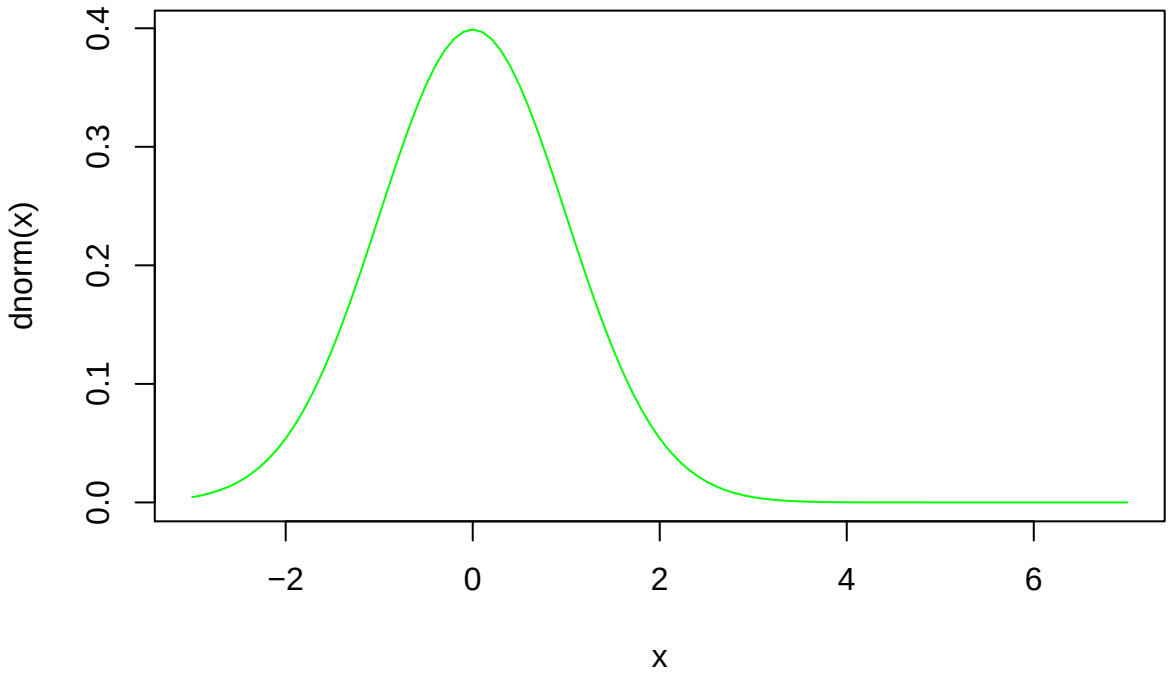
exo1

```
library(mclust)
```

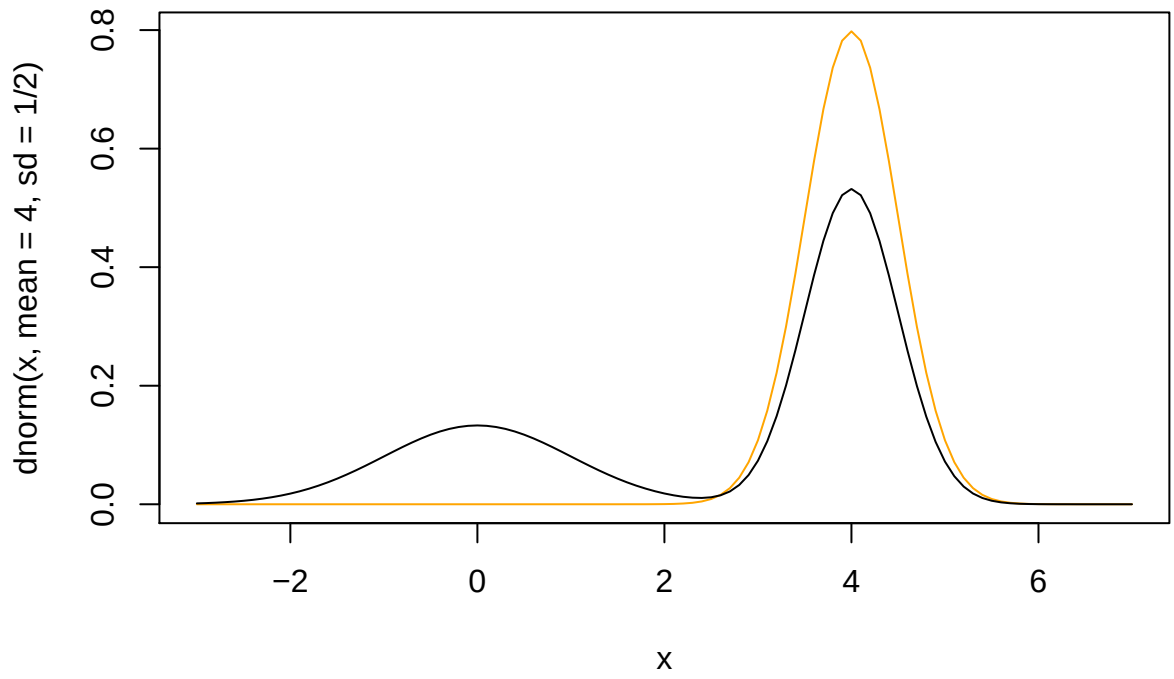
```
## Package 'mclust' version 5.4.3
## Type 'citation("mclust")' for citing this R package in publications.
```

```
nks <- rmultinom(n = 1, size = 1000, prob = c(1/3, 2/3))

means <- c(0, 4)
sds <- c(1, 1/2)
sample <- mapply(function(nk, mean, sd){
  rnorm(nk, mean, sd)}, nks, means, sds)

sample <- unlist(sample)
```

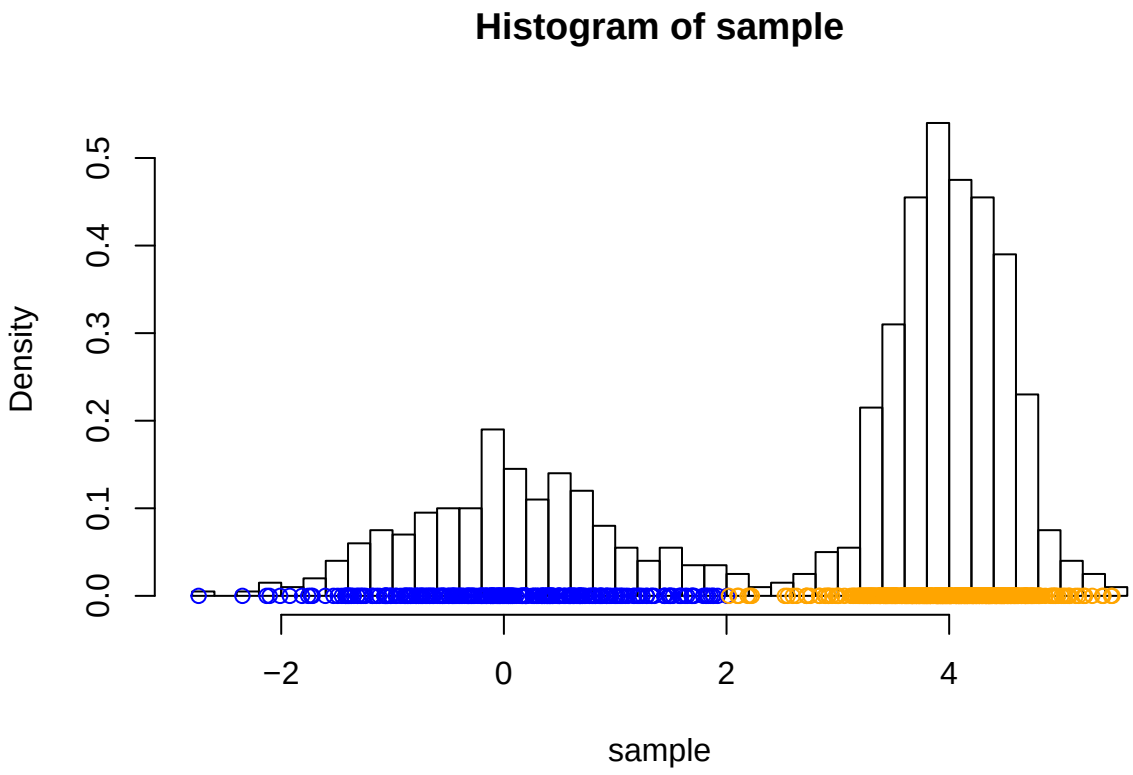
```
curve(dnorm(x), from = -3, to = 7, col = "green")
```



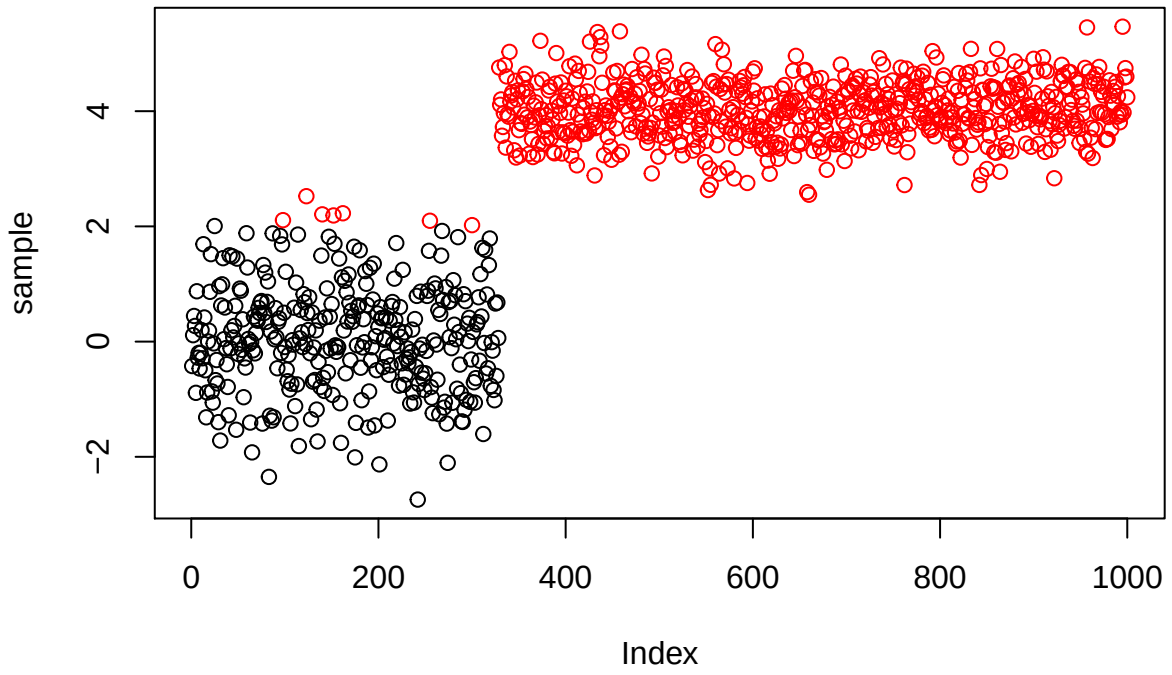

```
curve(dnorm(x, mean = 4, sd = 1/2), from = -3, to = 7, col = "orange")

mixture <- function(x){
  1/3*dnorm(x) + 2/3 * dnorm(x,4,1/2)
}

curve(mixture(x), -3, 7, add = T)
```




```
cl <- kmeans(x = sample, centers = 2, nstart = 20)
hist(x = sample, probability = T, breaks = 30)
points(sample, rep(0, 1000), col = c("blue", "orange")[cl$cluster])
```




```
plot(x = sample, col = cl$cluster)
```




```
list_of_samples <- split(sample, cl$cluster)
lapply(list_of_samples, mean)
```

```
## $`1`
## [1] 0.0273904
##
## $`2`
## [1] 3.995648
```

```
lapply(list_of_samples, sd)
```

```
## $`1`
## [1] 0.9030341
##
## $`2`
## [1] 0.5251815
```

```
# E variances partagées, variances libres
modelE <- Mclust(data = sample, G = 2, modelNames = "E", warn = F)
modelV <- Mclust(data = sample, G = 2, modelNames = "V", warn = F)

summary(modelE)
```

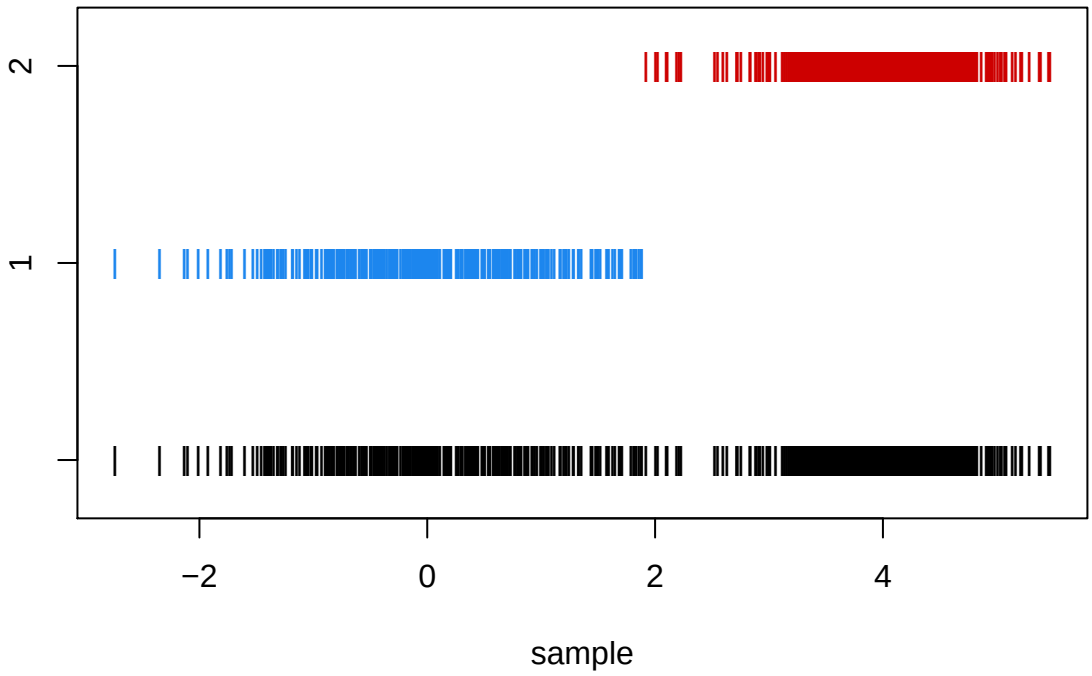
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust E (univariate, equal variance) model with 2 components:
##
##  log-likelihood    n df       BIC       ICL
##       -1638.218 1000  4 -3304.067 -3316.516
##
## Clustering table:
##   1   2
```

```
## 319 681
```

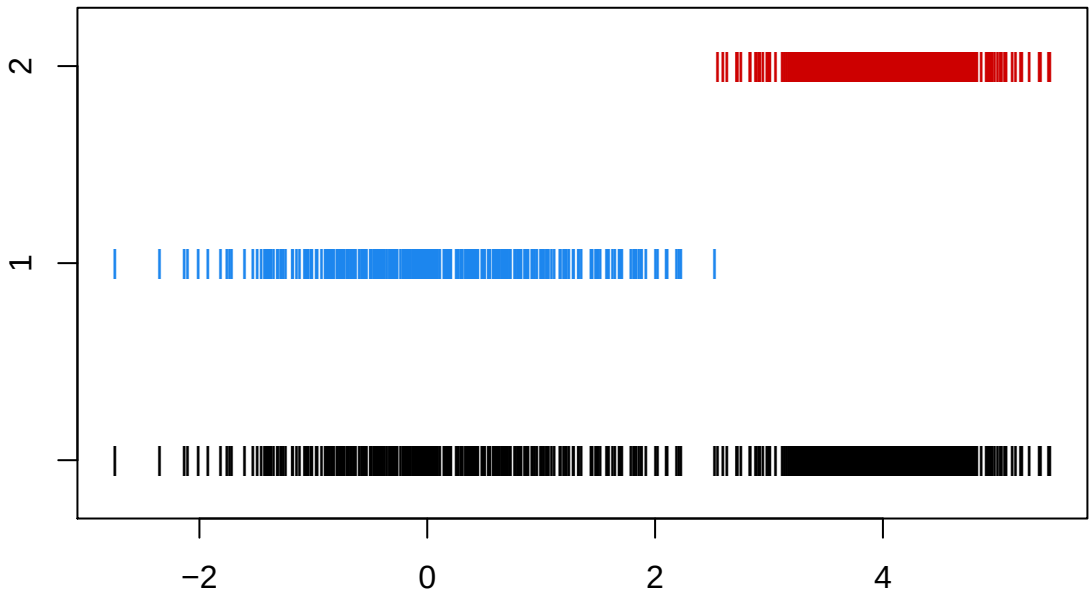
```
summary(modelV)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust V (univariate, unequal variance) model with 2 components:
##
##  log-likelihood    n df      BIC       ICL
##       -1555.845 1000  5 -3146.23 -3153.958
##
## Clustering table:
##   1   2
## 328 672
```

```
plot(modelE, what = "classification")
```




```
plot(modelV, what = "classification")
```




```
modelV$parameters$mean
```

```
##          1          2
## 0.09037535 4.01760735
```

```
modelV$parameters$variance$sigmasq
```

```
## [1] 0.9361713 0.2410452
```

```
nks <- rmultinom(n = 1, size = 1000, prob = c(1/3, 2/3))
means <- c(0, 4)
sds <- c(1, 1/2)
sample <- mapply(function(nk, mean, sd){
  rnorm(nk, mean, sd)}, nks, means, sds)
x <- data.frame(unlist(sample))

Init.EM <- function(x, k=2) {
  # init
  proportions <- rep(x = 1/k, times = k)
  variances <- rep(x = 1, times = k)
  means <- x[sample(1:nrow(x), k), ]
  parameters <- list(proportions = proportions, variances = variances, means = means)

  return(parameters)
}
```

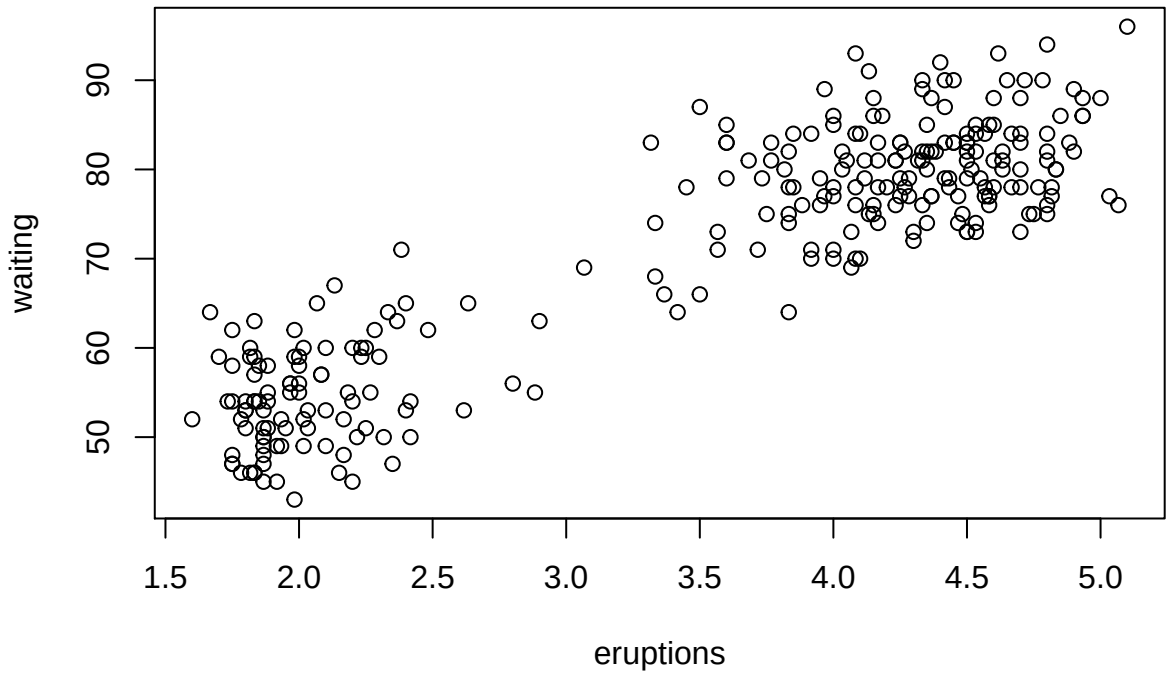
```
x <- as.matrix(x)

E.step <- function(x, parameters) {
  K <- length(parameters$means)
  Tik <- matrix(NA, nrow = nrow(x), ncol = K)

  for (k in 1:K) {
    Tik[,k] <- parameters$proportions[k] * dnorm(x[,1], mean = parameters$means[k], sd = parameters$vari
```

```
  }

  return(Tik <- Tik/rowSums(Tik))
}


M.step <- function(x, Tik, parameters) {
  K <- length(parameters$means)
  parameters$proportions <- colSums(Tik)/nrow(x)

  for (k in 1:K) {
    parameters$means[k] <- sum(Tik[,k]*x) / sum(Tik[,k])
    parameters$variances[k] <- sum(Tik[,k]*(x - parameters$means[k])^2)/sum(Tik[,k])
  }

  return(parameters)
}


EM <- function(x, k) {

  parameters <- Init.EM(x, k)
  iter <- 0
  parameters.new <- parameters
  repeat{
    Tik <- E.step(x, parameters)
    parameters <- M.step(x, Tik, parameters)
    if((sum(unlist(parameters.new) - unlist(parameters))^2) / sum(unlist(parameters.new))^2 < 1e-20) br
    parameters.new <- parameters
  }
  return(list(parameters = parameters.new, Tik = Tik))
}
```

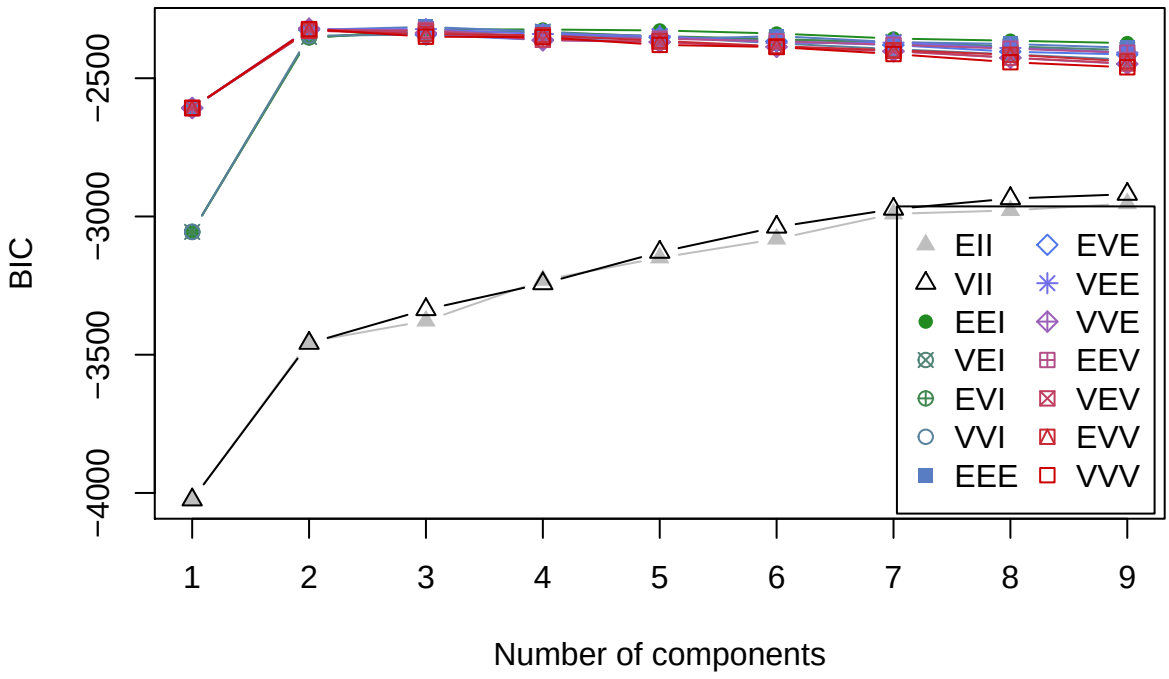
```
x <- data.frame(x)
# EM(x,2)
```

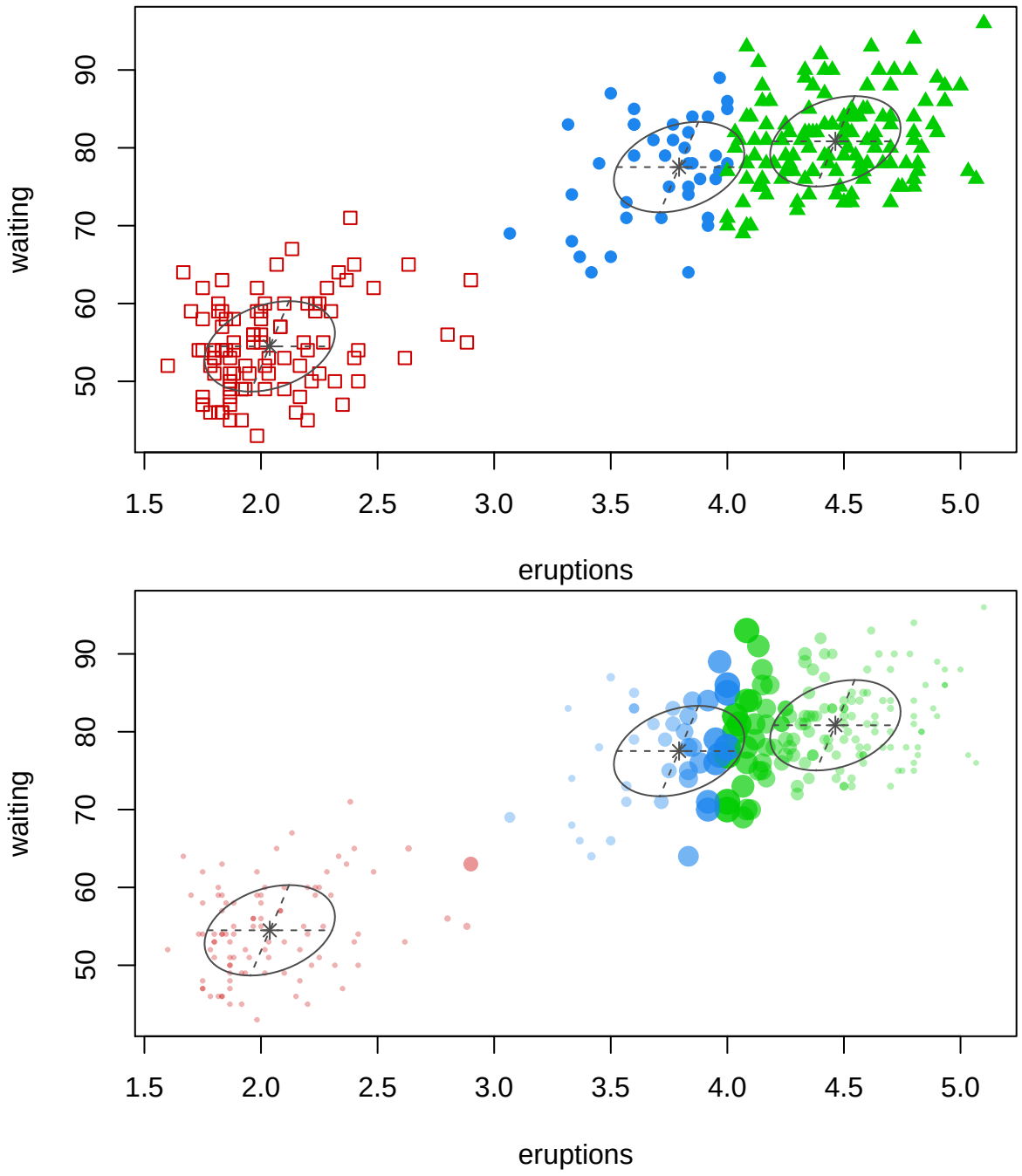
Exo2

```
data(faithful)
plot(faithful)
```

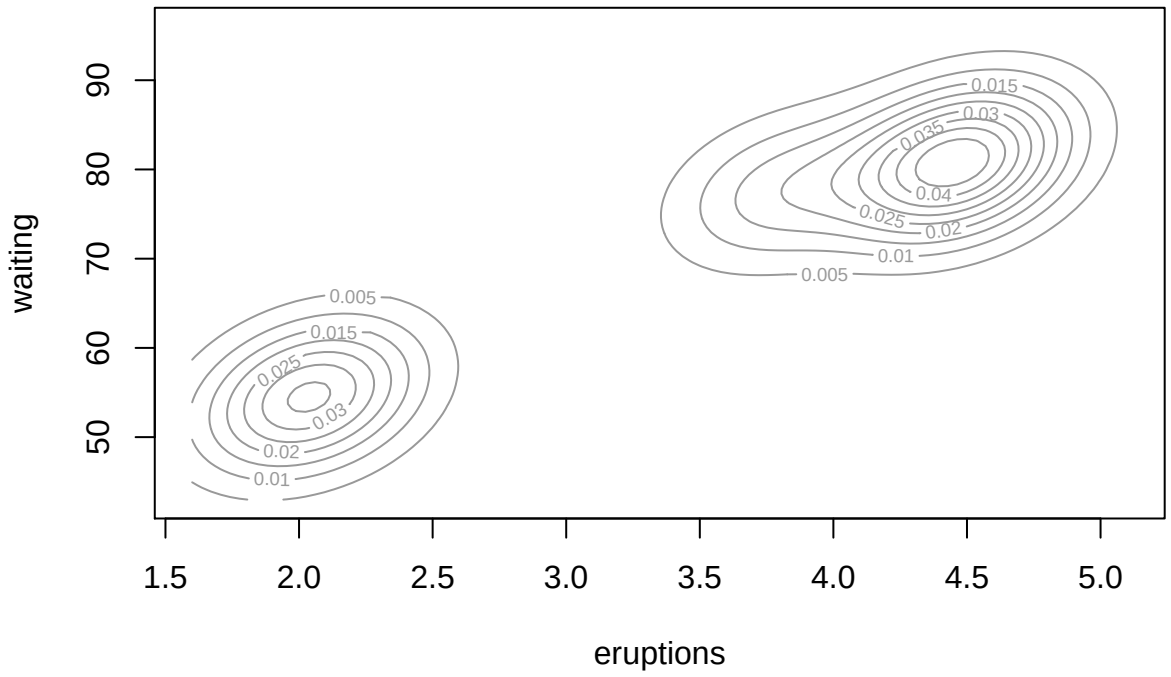



```
res <- Mclust(data = faithful)
plot(res)
```




waiting
90
80
70
60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions

waiting
90
80
70
60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions


waiting
50
60
70
80
90
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions
0.005
0.015
0.025
0.03
0.02
0.01
0.015
0.03
0.035
0.04
0.025
0.02
0.01
0.005