Problem 7.2

Chen Bo Calvin Zhang

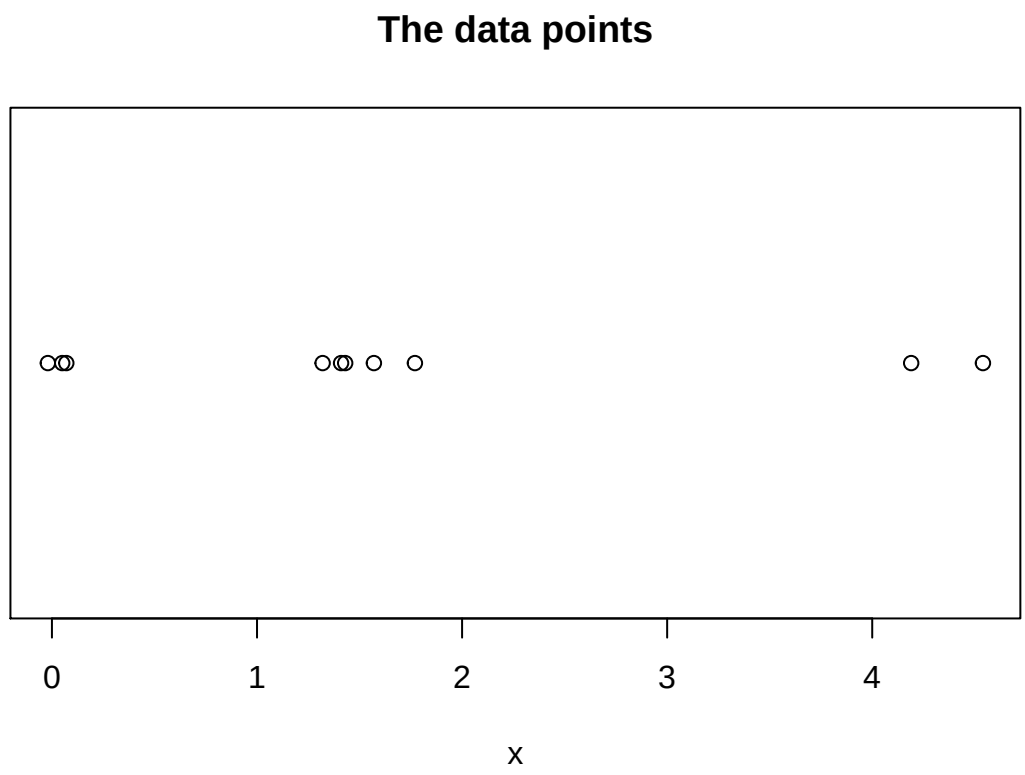
23/11/2020

Implement the EM algorithm for a K-component univariate Gaussian mixture model.

```
EM = function(x, K, num_iter = 100)
{
  # number of samples
  n = length(x)

  # random initialization of z
  z = matrix(runif(K*n), n, K)
  # make rows add up to 1
  # take each row (1) of z and divide each element ("/") by the rowSum
  rowStd = function(z) sweep(z, 1, rowSums(z), "/")
  z = rowStd(z)

  # mean vector
  mu = numeric(K)
  # standard deviation vector
  s2 = numeric(K)

  # loop num_iter times
  for (r in 1:num_iter)
  {

    # M step: maximize the expected complete data log-likelihood
    n.group = colSums(z)  # vector of n_k_hat
    pi = n.group / n  # vector of pi_k_hat

    mu = colSums(x*z) / n.group
    s2 = colSums((replicate(K, x) - t(replicate(n, mu)))^2*z) / n.group

    # E step: compute the probabilities of allocation for each sample X_i
    for (i in 1:n)
    {
      z[i, ] = pi * dnorm(x[i], mean = mu, sd = sqrt(s2))
    }
    z = rowStd(z)
  }

  return (list(pi=pi, mu=mu, s2=s2, z=zapsmall(z)))
}
```

Apply the EM function with K = 3 to the following data vector containing n = 10 observations.

```r
x = c(4.54, 1.57, 1.41, 1.77, 1.43, 0.07, 0.05, 4.19, -0.02, 1.32)
plot(x, rep(1, length(x)), ylab="", yaxt="n", main="The data points")
```



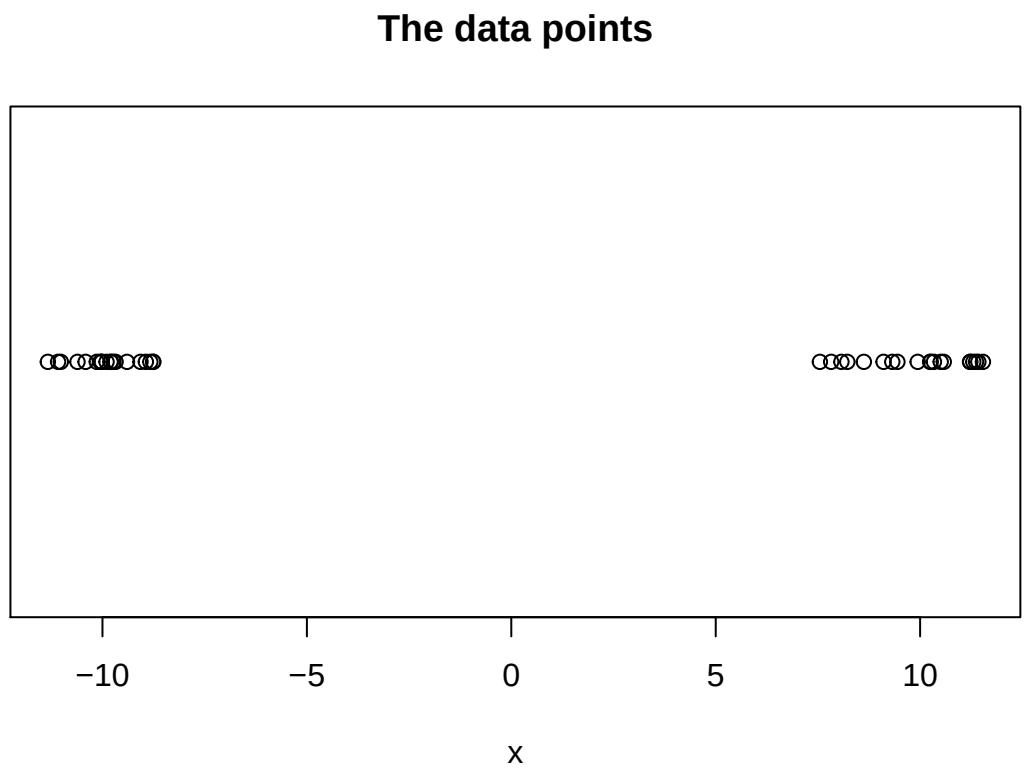

```r
em.out = EM(x, 3)
print(em.out)
```

```
## $pi
## [1] 0.3 0.2 0.5
##
## $mu
## [1] 0.03333333 4.36500000 1.50000000
##
## $s2
## [1] 0.001488889 0.030625000 0.024640000
##
## $z
##       [,1] [,2] [,3]
##  [1,]    0    1    0
##  [2,]    0    0    1
##  [3,]    0    0    1
##  [4,]    0    0    1
##  [5,]    0    0    1
##  [6,]    1    0    0
##  [7,]    1    0    0
##  [8,]    0    1    0
```

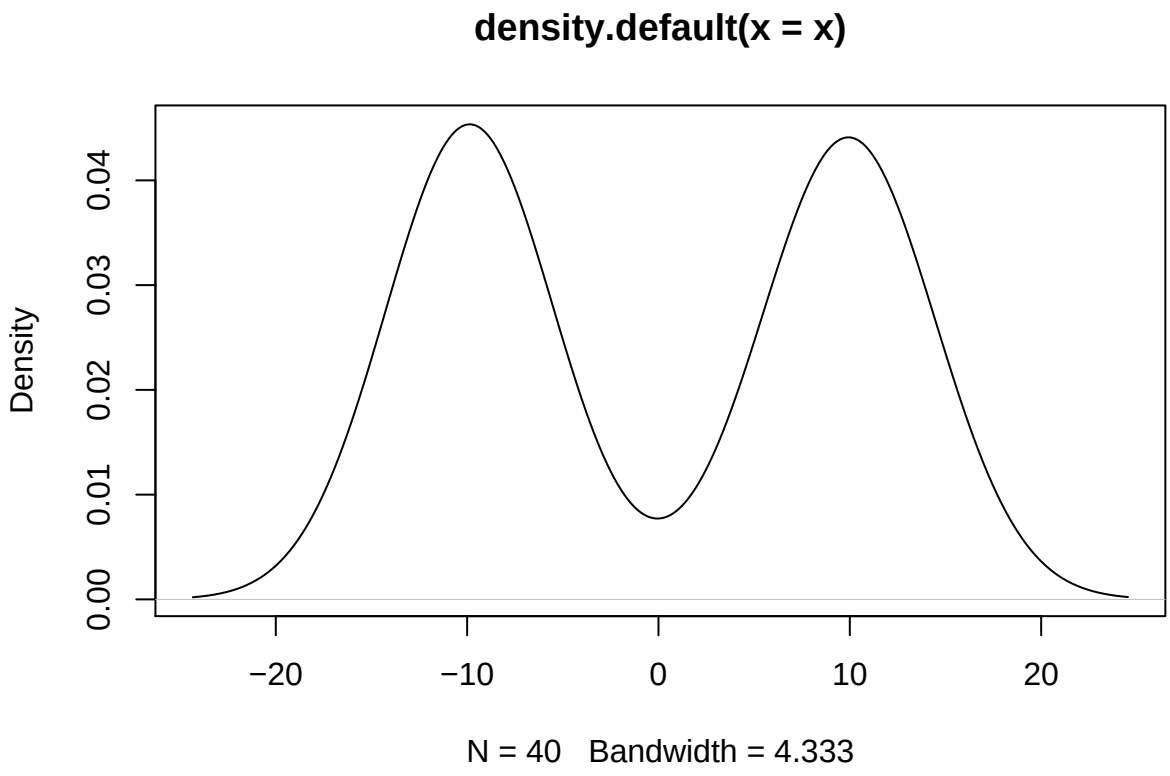
```
##  [9,]    1    0    0
## [10,]    0    0    1
```

Apply the EM function to data simulated as follows (i.e. two clearly separated groups).

```
x = c(rnorm(20, mean=-10), rnorm(20, mean=10))
plot(x, rep(1, length(x)), ylab="", yaxt="n", main="The data points")
```




```
plot(density(x))
```




```
em.out = EM(x, 2, 5000)
print(em.out)
```

```
## $pi
## [1] 0.5 0.5
##
## $mu
## [1] -9.867814  9.904950
##
## $s2
## [1] 0.5647151 1.6343843
##
## $z
##       [,1] [,2]
##  [1,]    1    0
##  [2,]    1    0
##  [3,]    1    0
##  [4,]    1    0
##  [5,]    1    0
##  [6,]    1    0
##  [7,]    1    0
##  [8,]    1    0
##  [9,]    1    0
## [10,]    1    0
## [11,]    1    0
```

```
## [12,]    1    0
## [13,]    1    0
## [14,]    1    0
## [15,]    1    0
## [16,]    1    0
## [17,]    1    0
## [18,]    1    0
## [19,]    1    0
## [20,]    1    0
## [21,]    0    1
## [22,]    0    1
## [23,]    0    1
## [24,]    0    1
## [25,]    0    1
## [26,]    0    1
## [27,]    0    1
## [28,]    0    1
## [29,]    0    1
## [30,]    0    1
## [31,]    0    1
## [32,]    0    1
## [33,]    0    1
## [34,]    0    1
## [35,]    0    1
## [36,]    0    1
## [37,]    0    1
## [38,]    0    1
## [39,]    0    1
## [40,]    0    1
```