XIAOYU WANG

```
H <- matrix(readBin("~/Desktop/data/histograms.bin", "double", 640000), 40000, 16)
```

part 1): Implement the EMalgorithm in R

```
multinomialEM <-function(hist,K,tau){

# number of obs
i <- nrow(hist)

# number of dimensions
j <- ncol(hist)

# choose random value of cluster proportions (weights)
ck <- rep(1,K)/K

# choose random value of (cluster parameters)
tk <- hist[sample(1:i,K), ] + 0.01

diff <- 5
while (diff > tau){
# E-Step:
Phi <- exp( hist %*% t(log(tk)))
sum <- rowSums((ck * Phi))
aik <- (ck * Phi)/sum

# M-Step:
#compute new weight c
ck <- colSums(aik)/i

tk_old <- tk

bk <- t(aik) %*% hist
sum2 <- rowSums(bk)
tk <- bk/sum2

# Compute difference:
diff <- norm(tk - tk_old)

}

a.max <- apply(aik,1,which.max)
return(a.max)

}
```

part 2): Run the algorithm on the input data for K=3, K=4 and K=5.

```
set.seed(444)
head(multinomialEM(H, 3, 0.01), n = 50)
```

```
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 2 2 2 2 2 3 3 3 3 3 2 3 2 2 2 2
## [36] 2 2 3 2 2 2 3 3 3 3 3 3 2 3 3
```

```
head(multinomialEM(H, 4, 0.01), n = 50)
```

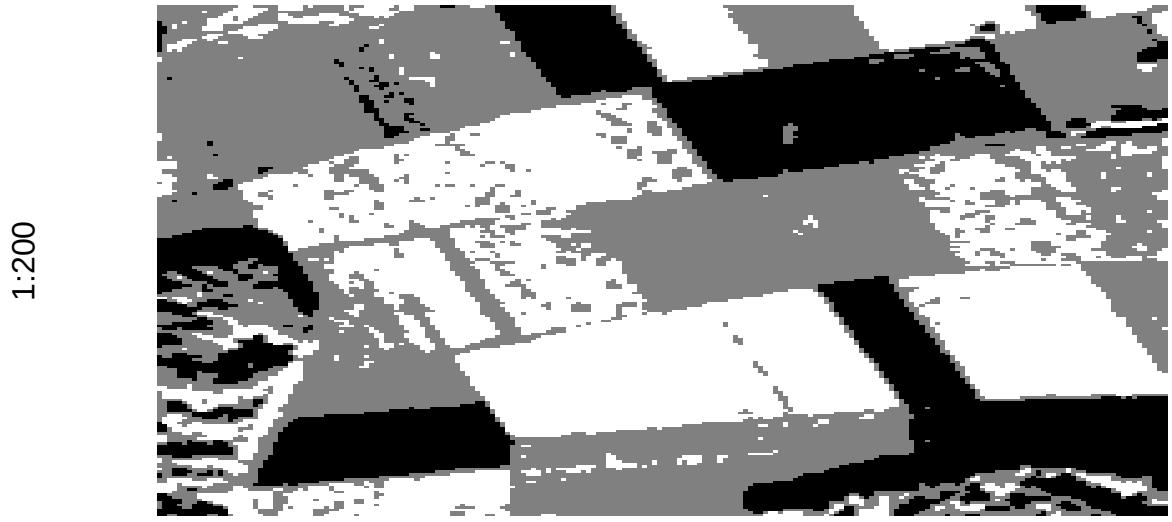
```
##  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 3 3 3 3 3 3 4 3 4 3 3 3 3 3 3 3 3
## [36] 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3
```

```
head(multinomialEM(H, 5, 0.01), n = 50)
```

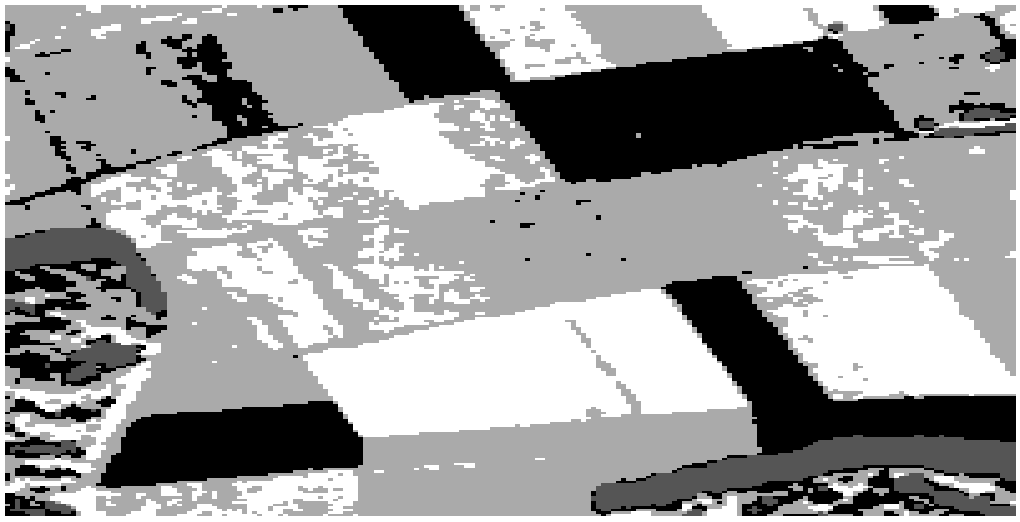
```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 4 4 4 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1
## [36] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
```

part 3): Visualize the results as an image.

```
set.seed(444)
Visualize <- function(x){

#converte the result to matrix form
data <- matrix(x, nrow = 200, byrow = T)
data2 <- t(data)
data2 <- data2[,ncol(data2):1]

image(x = 1:200, y = 1:200, data2, axes = FALSE, col = grey((0:256)/256))

}


Visualize(multinomialEM(H, 3, 0.01))
```

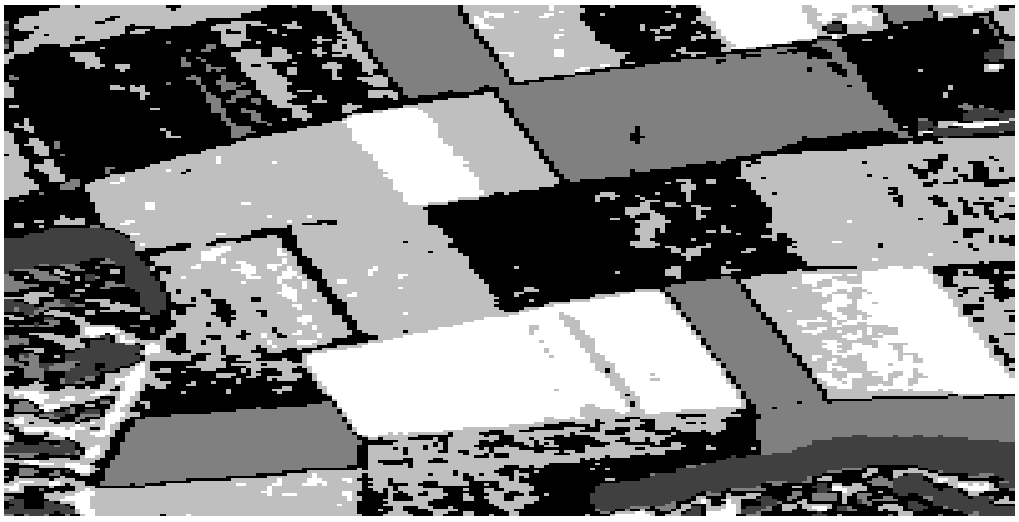



```
Visualize(multinomialEM(H, 4, 0.01))
```




```
Visualize(multinomialEM(H, 5, 0.01))
```

1:200



1:200