lab3

Weng Hang Wong

4/15/2020

1. Normal model, mixture of normal model with semi-conjugate prior.

The data rainfall.dat consist of daily records, from the beginning of 1948 to the 1/100 inch, and records of zero end of 1983, of precipitation (rain or snow in units of 100 precipitation are excluded) at Snoqualmie Falls, Washington. Analyze the data using the following two models.

(a) Normal model.

Assume the daily precipitation $\{y_1, ..., y_n\}$ are independent normally distributed, $y_1, ..., y_n|\mu, \sigma^2 \sim N(\mu, \sigma^2)$ where both μ and σ^2 are unknown. Let $\mu \sim N(\mu_0, \tau_0^2)$ independently of $\sigma^2 \sim Inv - \chi^2(\nu_0, \sigma_0^2)$

i. Implement (code!) a Gibbs sampler that simulates from the joint posterior $p(\mu, \sigma^2|y_1, ..., y_n)$. The full conditional posteriors are given on the slides from Lecture 7.

The conditionally conjugate prior:

$$\mu \sim N(\mu_0, \tau_0^2)$$

$$\sigma^2 \sim Inv - \chi^2(v_0, \sigma_0^2)$$

The full conditional posteriors:

$$\mu|\sigma, x \sim N(\mu_n, \tau_n^2)$$

$$\sigma^2|\mu, x \sim Inv - \chi^2(v_n, \frac{v_0\sigma_0^2 + \sum_{i=1}^n (x_i - \mu)^2}{n + v_0})$$

$$where, \frac{1}{\tau_n^2} = \frac{n}{\sigma^2} + \frac{1}{\tau_0^2}$$

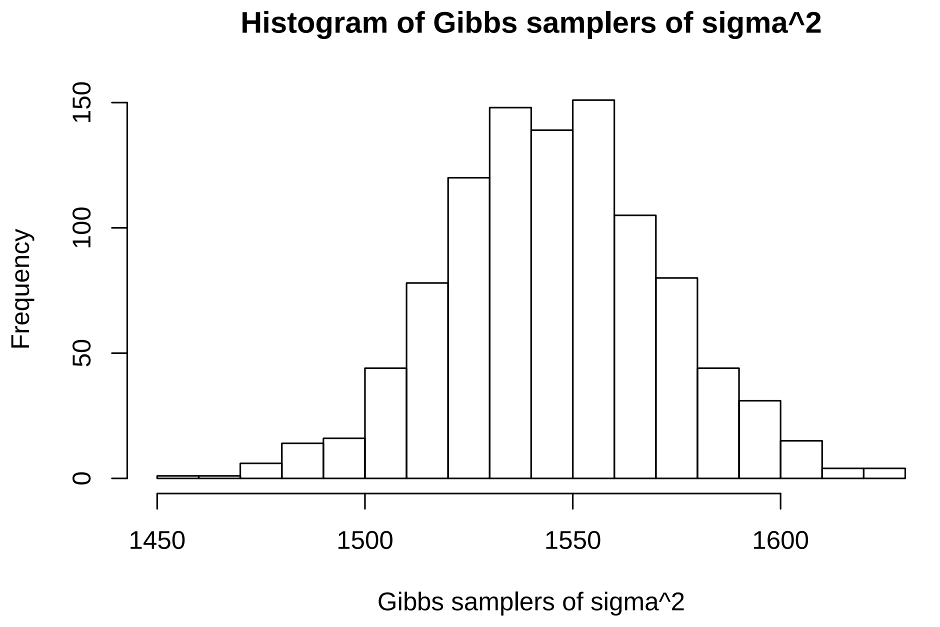
$$\mu_n = w\bar{x} + (1 - w)\mu_0$$

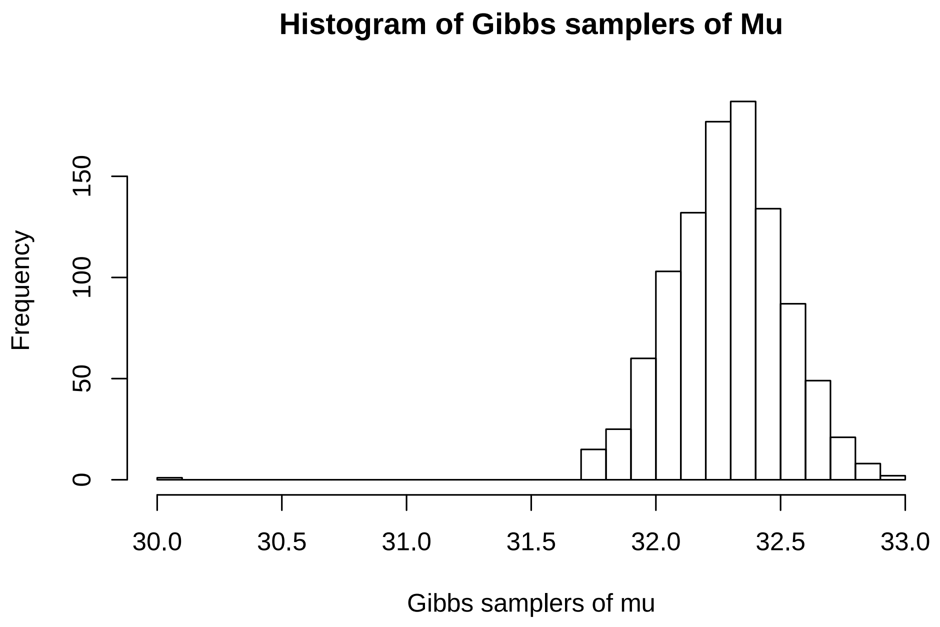
$$where, w = \frac{n/\sigma^2}{n/\sigma^2 + 1/\tau_0^2}$$

Since we don't have enough information about the percipitaion in Washington, base on lack of knowledge, we set our parameters as following: $\mu_0 = 30, \tau_0^2 = 50, \nu_0 = 3, \sigma_0^2 = var(givendata)$.

By using Gibbs Sampler, we simulate the mean and the variance from the joint posterior in a 1000 draws.

```
## [1] 1547.103
```



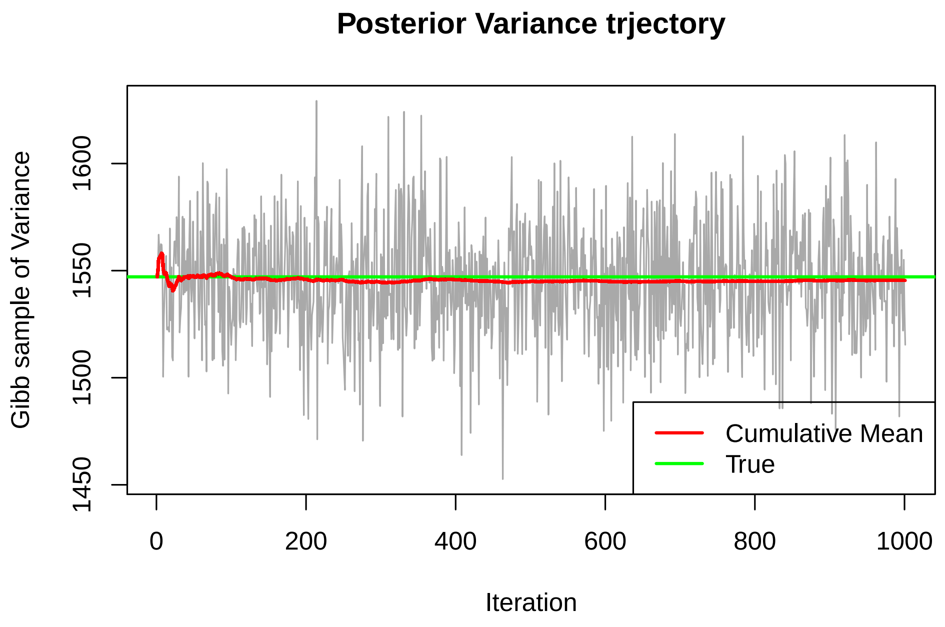


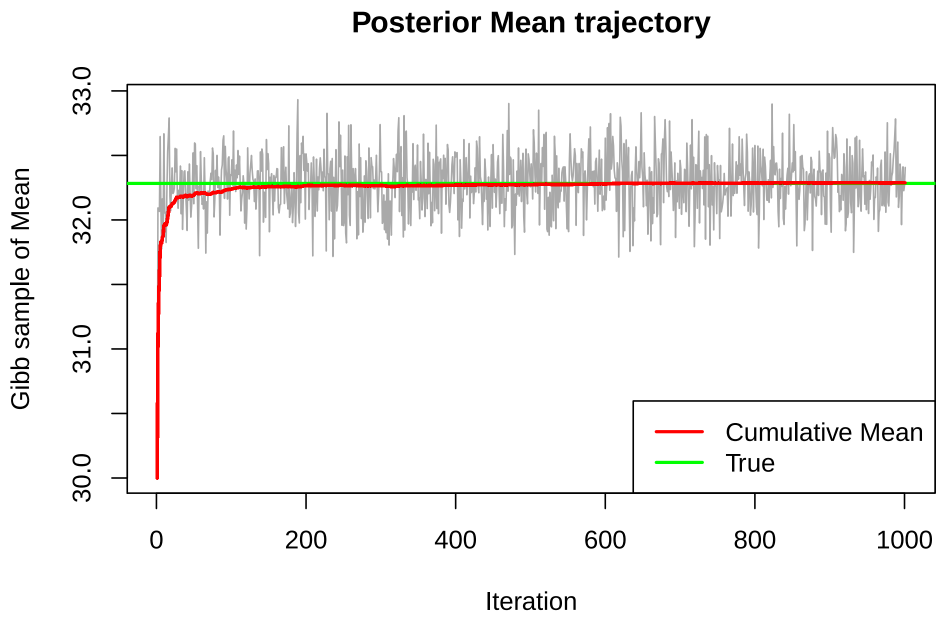
ii. Analyze the daily precipitation using your Gibbs sampler in (a)-i. Evaluate the convergence of the Gibbs sampler by suitable graphical methods, for example by plotting the trajectories of the sampled Markov chains.

The below two graphs are shown the trajectory plot of the sampled Markov chains. From the first graph of Posterior Variance trajectory, after around 50 iterations burning period, it is towards converaged to the true variance of the data which is about 1547.

From the second graph of Posterior Mean trajectory, the burning-period of which is faster than the varanice, which within around 50 iterations. After that, it is converaged to the true mean of the data which is about 32.28.

From the result, we can say that the Gibbs Sampling is a success, since both Gibbs mean and variance are converaged to the true mean and variance. Moreover, the burning period is fast and within 50 iterations, based on the initiative parameters are set close to the given data.





(b). Mixture normal model.

Let us now instead assume that the daily precipitation $y_1, ..., y_n$ follow an iid two-component mixture of normals model:

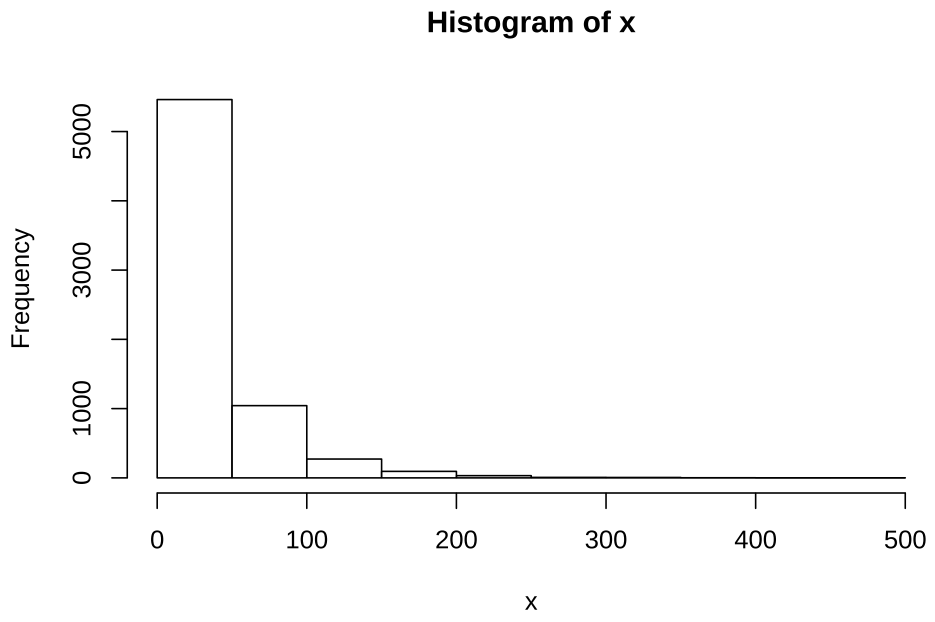
$$p(y_i|\mu, \sigma^2, \pi) = \pi N(y_i|\mu_1, \sigma_1^2) + (1-\pi)N(y_i|\mu_2, \sigma_2^2),$$

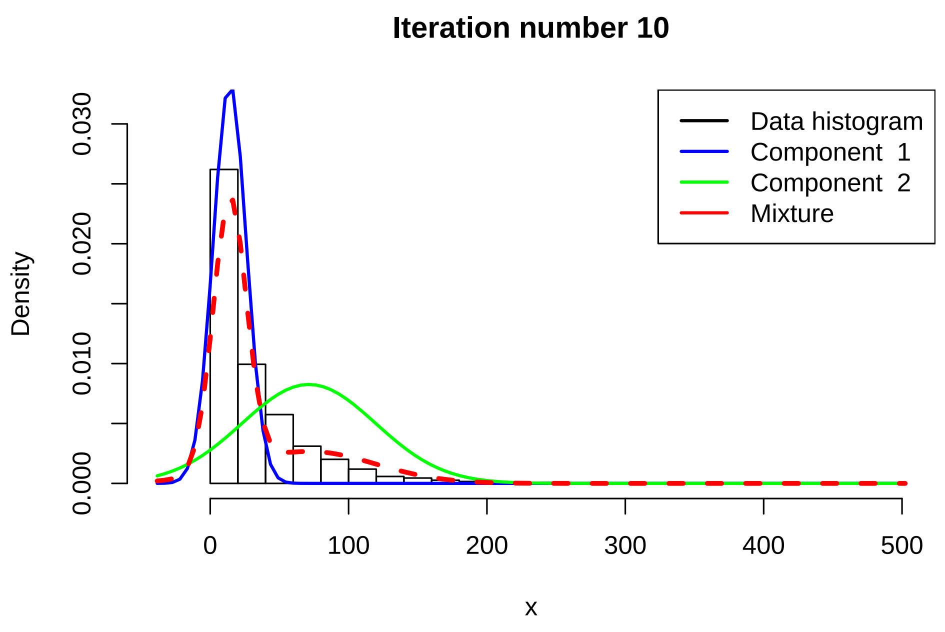
where

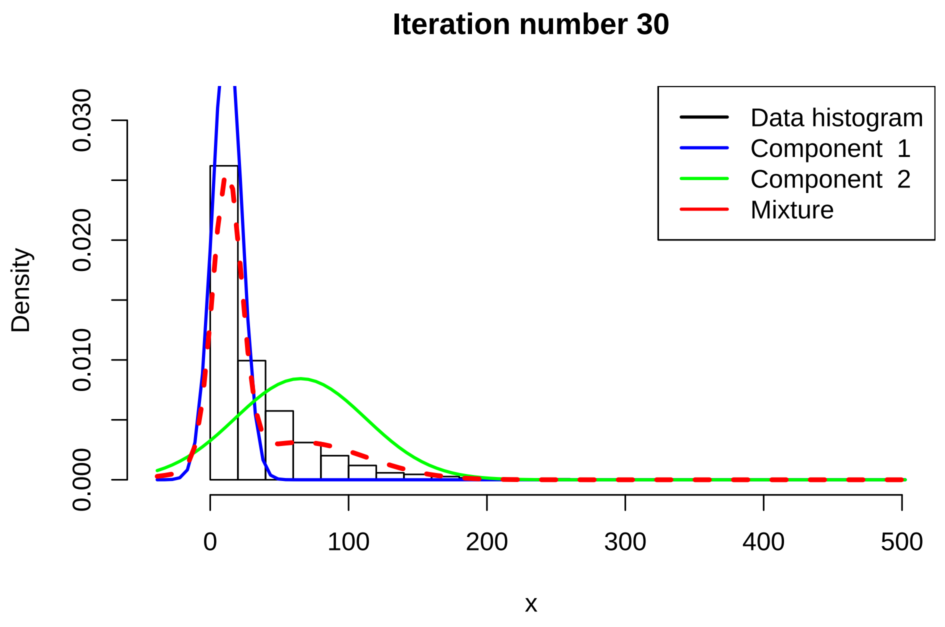
$$\mu = (\mu_1, \mu_2) \ \ and \ \ \sigma^2 = (\sigma_1^2, \sigma_2^2)$$

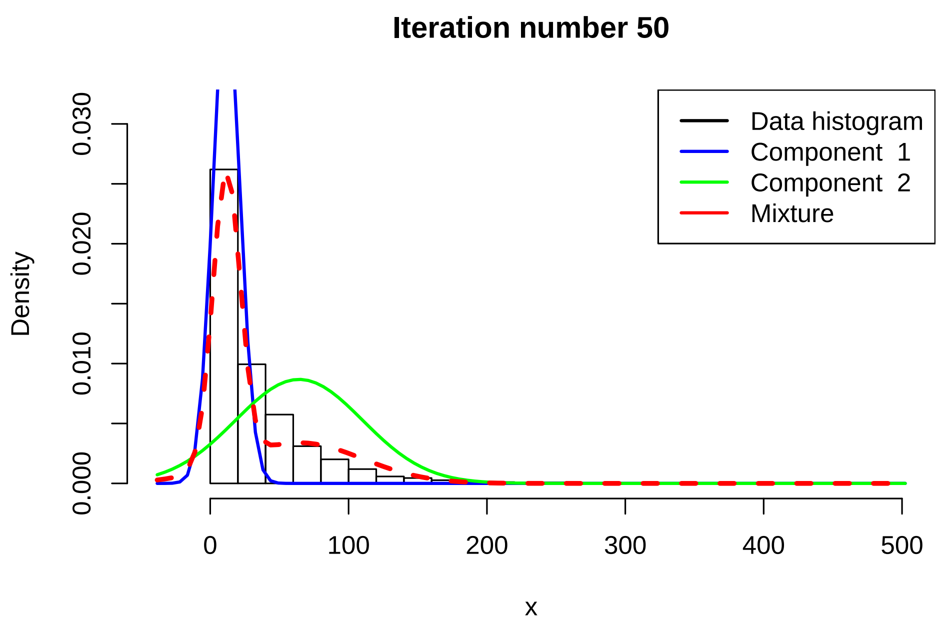
Use the Gibbs sampling data augmentation algorithm in NormalMixtureGibbs.R (available under Lecture 7 on the course page) to analyze the daily precipitation data. Set the prior hyperparameters suitably. Evaluate the convergence of the sampler.

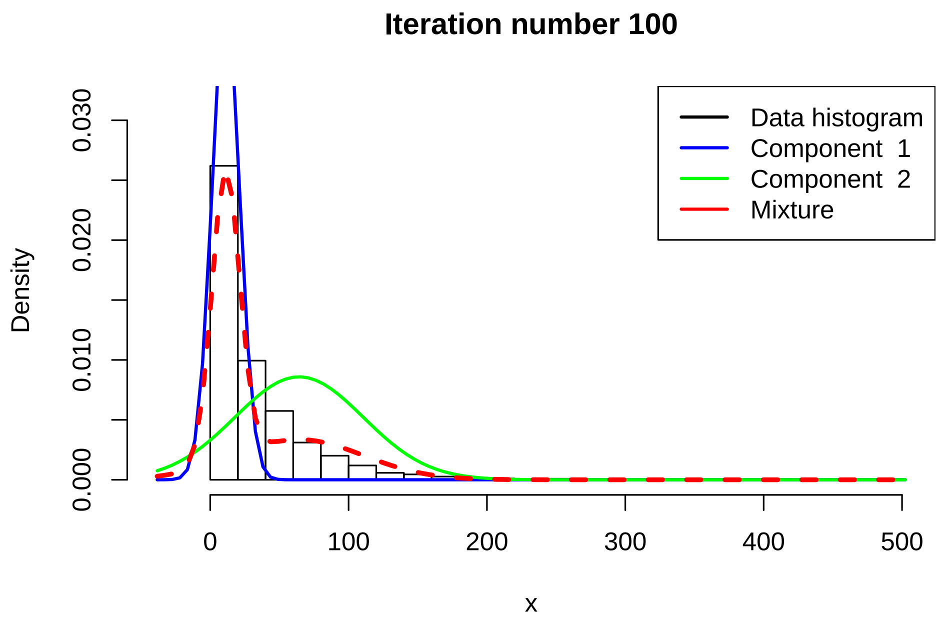
We set the same prior hyperparmeters as above. From 100 iterations, the sampler converaged very fast from the first 20 iterations, and after 30th iterations the components do not change much. Therefore, we can say in the algorithm of normal mixture, Gibbs sampling with the initial hyperparameters work well in fitting the data.




Iteration number 10
Density
0.000
0.010
0.020
0.030
0
100
200
300
400
500
x
Data histogram
Component 1
Component 2
Mixture


Iteration number 30
Density
0.000
0.010
0.020
0.030
0
100
200
300
400
500
x
Data histogram
Component 1
Component 2
Mixture





(c) Graphical comparison.

Plot the following densities in one figure: 1) a histogram or kernel density estimate of the data. 2) Normal density $N(y_i|\mu, \sigma^2)$ in (a); 3) Mixture of normals density $p(y_i|\mu, \sigma^2, \pi)$ in (b). Base your plots on the mean over all posterior draws.

Appendix

```
library(LaplacesDemon)
# 1.
## (a)
###.i. Gibbs sampler from joint posterior
data = read.table("rainfall.dat")

# para from data
n = length(row(data)) #6920
xbar = mean(data$V1) #32.28
var(data$V1)

#set up parameters
mu0 = 30 #True mean =32.28
tau20 = 150 # we dont have prior knowledge
nu0 = 3 #df set it to a small value
sigma20 = var(data$V1) #True var=1547.103

## Gibbs sampling
Ndraws = 1000
GibbMu = c()
GibbMu[1] = mu0
GibbSigma2 = c()
GibbSigma2[1] = sigma20

for(i in 1:Ndraws){
  # parameters w mu tau, update mu_n, tau_n
  w = (n/GibbSigma2[i]) / ( (n/GibbSigma2) + (1/tau20) )
  mu_n = w*xbar + (1-w)*mu0
  tau_n = 1/ ( (n/GibbSigma2[i]) + (1/tau20)  )

  # sampling posterior mu, add in mu
  GibbMu[i+1] = rnorm(1, mean=mu_n, sd=tau_n)

  #sampling posterior sigma2, add in sigma2
  nu_n = nu0 + n
  Scale =(nu0*sigma20 + sum((data$V1 - GibbMu[i+1])^2) )/ (n+nu0)
  GibbSigma2[i+1] = rinvchisq(1, df= nu_n, scale= Scale)
}

hist(GibbSigma2,breaks=20, main="Histogram of Gibbs samplers of sigma^2", xlab="Gibbs samplers of sigma
hist(GibbMu, breaks=30, main="Histogram of Gibbs samplers of Mu", xlab="Gibbs samplers of mu")


### ii.
cumMu=c()
cumSigma2=c()

# plot variance MC converage
plot(GibbSigma2, type="l",col="darkgrey", main="Posterior Variance trjectory", xlab="Iteration",ylab="G
abline(h=var(data$V1), col="green", lwd=2)
legend("bottomright",legend=c("Cumulative Mean", "True"), col=c("red","green"),lwd=2)
```

```
for(i in 1:length(GibbSigma2)){
  cumSigma2[i] = mean(GibbSigma2[1:i])
  lines(cumSigma2, col="red", lwd=2)
}

# plot mean MC converage
plot(GibbMu, type="l", col="darkgrey", main="Posterior Mean trajectory", xlab="Iteration", ylab="Gibb s
abline(h=xbar, col="green", lwd=2)
legend("bottomright",legend=c("Cumulative Mean", "True"), col=c("red","green"),lwd=2)
for(i in 1:length(GibbMu)){
  cumMu[i]= mean(GibbMu[1:i])
  lines(cumMu,col="red", lwd=2)
}


# 1.b

# Estimating a simple mixture of normals
# Author: Mattias Villani, IDA, Linkoping University. http://mattiasvillani.com

##########    BEGIN USER INPUT #################
# Data options
#data(faithful)
#rawData <- faithful
x <- as.matrix(data$V1)

# Model options
nComp <- 2    # Number of mixture components

# Prior options
alpha <- 10*rep(1,nComp) # Dirichlet(alpha)
muPrior <- rep(30,nComp) # Prior mean of mu
tau2Prior <- rep(150,nComp) # Prior std of mu
sigma2_0 <- rep(var(x),nComp) # s20 (best guess of sigma2)
nu0 <- rep(3,nComp) # degrees of freedom for prior on sigma2

# MCMC options
nIter <- 100 # Number of Gibbs sampling draws

# Plotting options
plotFit <- TRUE
lineColors <- c("blue", "green", "magenta", 'yellow')
#sleepTime <- 0.1 # Adding sleep time between iterations for plotting
################   END USER INPUT ###############

###### Defining a function that simulates from the
rScaledInvChi2 <- function(n, df, scale){
  return((df*scale)/rchisq(n,df=df))
}

####### Defining a function that simulates from a Dirichlet distribution
rDirichlet <- function(param){
  nCat <- length(param)
```

```r
  piDraws <- matrix(NA,nCat,1)
  for (j in 1:nCat){
    piDraws[j] <- rgamma(1,param[j],1)
  }
  piDraws = piDraws/sum(piDraws) # Diving every column of piDraws by the sum of the elements in that co
  return(piDraws)
}

# Simple function that converts between two different representations of the mixture allocation
S2alloc <- function(S){
  n <- dim(S)[1]
  alloc <- rep(0,n)
  for (i in 1:n){
    alloc[i] <- which(S[i,] == 1)
  }
  return(alloc)
}

# Initial value for the MCMC
nObs <- length(x)
S <- t(rmultinom(nObs, size = 1 , prob = rep(1/nComp,nComp))) # nObs-by-nComp matrix with component all
mu <- quantile(x, probs = seq(0,1,length = nComp))
sigma2 <- rep(var(x),nComp)
probObsInComp <- rep(NA, nComp)

# Setting up the plot
xGrid <- seq(min(x)-1*apply(x,2,sd),max(x)+1*apply(x,2,sd),length = 100)
xGridMin <- min(xGrid)
xGridMax <- max(xGrid)
mixDensMean <- rep(0,length(xGrid))
effIterCount <- 0
ylim <- c(0,2*max(hist(x)$density))


for (k in 1:nIter){
  #message(paste('Iteration number:',k))
  alloc <- S2alloc(S) # Just a function that converts between different representations of the group al
  nAlloc <- colSums(S)
  #print(nAlloc)
  # Update components probabilities
  pi <- rDirichlet(alpha + nAlloc)

  # Update mu's
  for (j in 1:nComp){
    precPrior <- 1/tau2Prior[j]
    precData <- nAlloc[j]/sigma2[j]
    precPost <- precPrior + precData
    wPrior <- precPrior/precPost
    muPost <- wPrior*muPrior + (1-wPrior)*mean(x[alloc == j])
    tau2Post <- 1/precPost
    mu[j] <- rnorm(1, mean = muPost, sd = sqrt(tau2Post))
  }
  # Update sigma2's
  for (j in 1:nComp){
    sigma2[j] <- rScaledInvChi2(1, df = nu0[j] + nAlloc[j], scale = (nu0[j]*sigma2_0[j] + sum((x[alloc 
  }

  # Update allocation
  for (i in 1:nObs){
    for (j in 1:nComp){
      probObsInComp[j] <- pi[j]*dnorm(x[i], mean = mu[j], sd = sqrt(sigma2[j]))
    }
    S[i,] <- t(rmultinom(1, size = 1 , prob = probObsInComp/sum(probObsInComp)))
  }


  # Printing the fitted density against data histogram
  if(k %in% c(10,30,50,100) ){
  if (plotFit && (k%%1 ==0)){
    effIterCount <- effIterCount + 1
    hist(x, breaks = 20, freq = FALSE, xlim = c(xGridMin,xGridMax), main = paste("Iteration number",k), 
    mixDens <- rep(0,length(xGrid))
    components <- c()
    for (j in 1:nComp){
      compDens <- dnorm(xGrid,mu[j],sd = sqrt(sigma2[j]))
      mixDens <- mixDens + pi[j]*compDens
      lines(xGrid, compDens, type = "l", lwd = 2, col = lineColors[j])
      components[j] <- paste("Component ",j)
    }
    mixDensMean <- ((effIterCount-1)*mixDensMean + mixDens)/effIterCount

    lines(xGrid, mixDens, type = "l", lty = 2, lwd = 3, col = 'red')
    legend("topright", box.lty = 1, legend = c("Data histogram",components, 'Mixture'),
           col = c("black",lineColors[1:nComp], 'red'), lwd = 2)
    #Sys.sleep(sleepTime)
  }

  }
}

#########################    Helper functions    ##############################################
```