hw 2

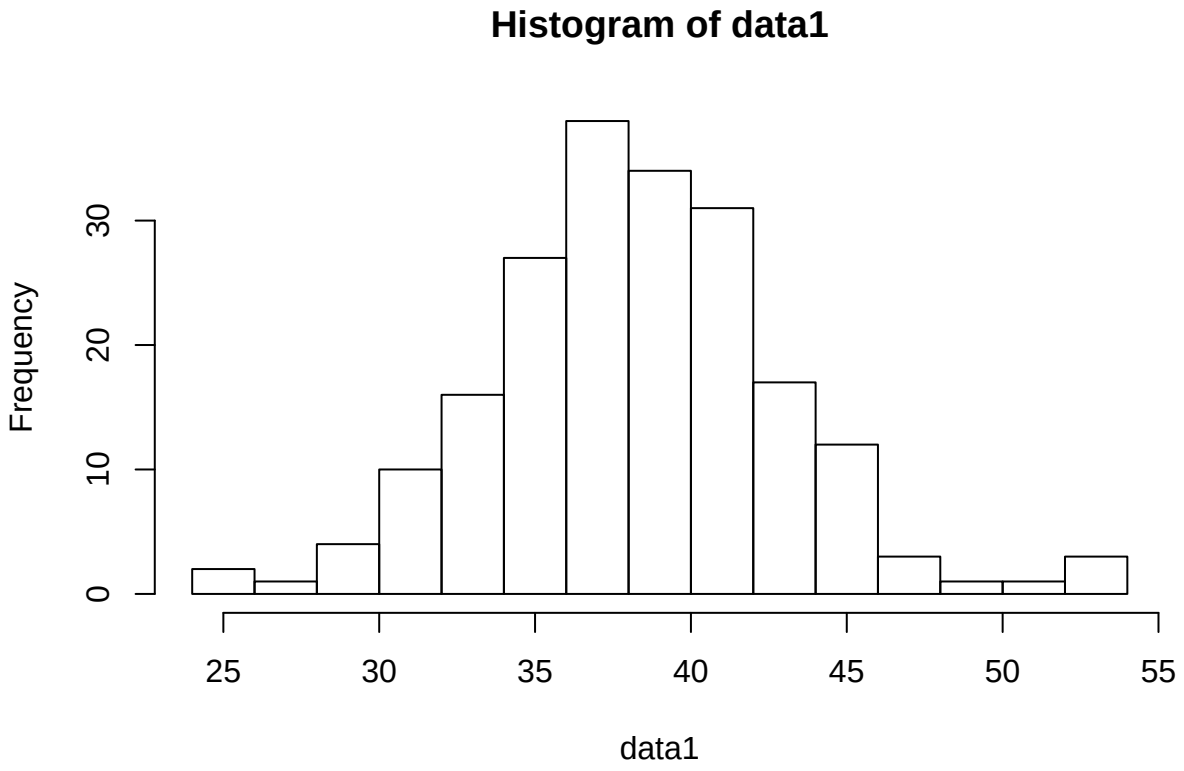
Sina Sanei

November 2, 2018

Question 1

let's see the data:

```
hist(data1, breaks = 15)
```




```
summary(data1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.03   35.37   38.07   38.35   41.18   52.73
```

density function is :

```
f_x= function(x, mu, gamma ){
  exp(-(x-mu)/gamma)/ (gamma*(1+exp(-(x-mu)/gamma))^2)
}
#likelihood:
lik = function(xvec , mu , gamma ){
  li = 1
  for( i in 1:length(xvec)){
    ll= f_x(xvec[i],mu,gamma)
    li = li * ll
  }
  return(li)
```

```
}
# unnormalized pi will be :
pi = function(xvec , mu , gamma ){
  lik(xvec, mu,gamma) * exp((-mu^2)/50) * exp(-gamma/10)
}
# now we compute full conditionals : f(mu| X,gamma)
f_mu = function(xvec, mu , gamma ){
  lik(xvec, mu,gamma) * exp((-mu^2)/50)
}
# full conditional : f(gamma| X,mu)
f_gamma = function(xvec , mu , gamma ){
  lik(xvec, mu,gamma) * exp(-gamma/10)
}
```

Now we can consrtruct metropolis-hastings algortithm :

```
#initialize markov chain:(mu,gamma), take the mean of prior distributions for initial
set.seed(501)
chain_length= 5000
MH = matrix(NA, chain_length, 3,dimnames = list(NULL,c("index","mu","gamma")))
MH[1,]= c(1, 38.2924, 2.6)
proposal = function(mu,gamma){
  p1=rnorm(1,mu,1)
  p2=rnorm(1,gamma,1)
  return(c(p1,p2))
}
naccept = 1
i = 2
for (i in 2:chain_length){
  MH[i,1]=i
  propose = proposal(MH[i-1,2],MH[i-1,3]) # make a proposal for new mu and gamma
  alpha_1 = min(f_mu(data1,propose[1],MH[i-1,3])*dnorm(MH[i-1,2],MH[i-1,2],1)/
                f_mu(data1,MH[i-1,2],MH[i-1,3])/dnorm(propose[1],MH[i-1,2],1) , 1)
  alpha_2 = min(f_gamma(data1,MH[i-1,2],propose[2])*dnorm(MH[i-1,3],MH[i-1,3],1)/
                f_gamma(data1,MH[i-1,2],MH[i-1,3])/dnorm(propose[2],MH[i-1,3],1) , 1)
  if (runif(1)< alpha_1){ MH[i,2]= propose[1] }else{MH[i,2]= MH[i-1,2]}
   if (runif(1)< alpha_2){ MH[i,3]= propose[2] }else{MH[i,3]= MH[i-1,3]}
}
```

To determine the statring values I used the R package : `fitdist` Which is used to fit the logistic distribution to the observed data via : `si = fitdist(data1,"logis")` this uses Mle to estimate the parameters distribution with observed data, then I get :

```
library(fitdistrplus)
```

```
## Loading required package: MASS
```

```
## Loading required package: survival
```

```
## Loading required package: npsurv
```

```
## Loading required package: lsei
```

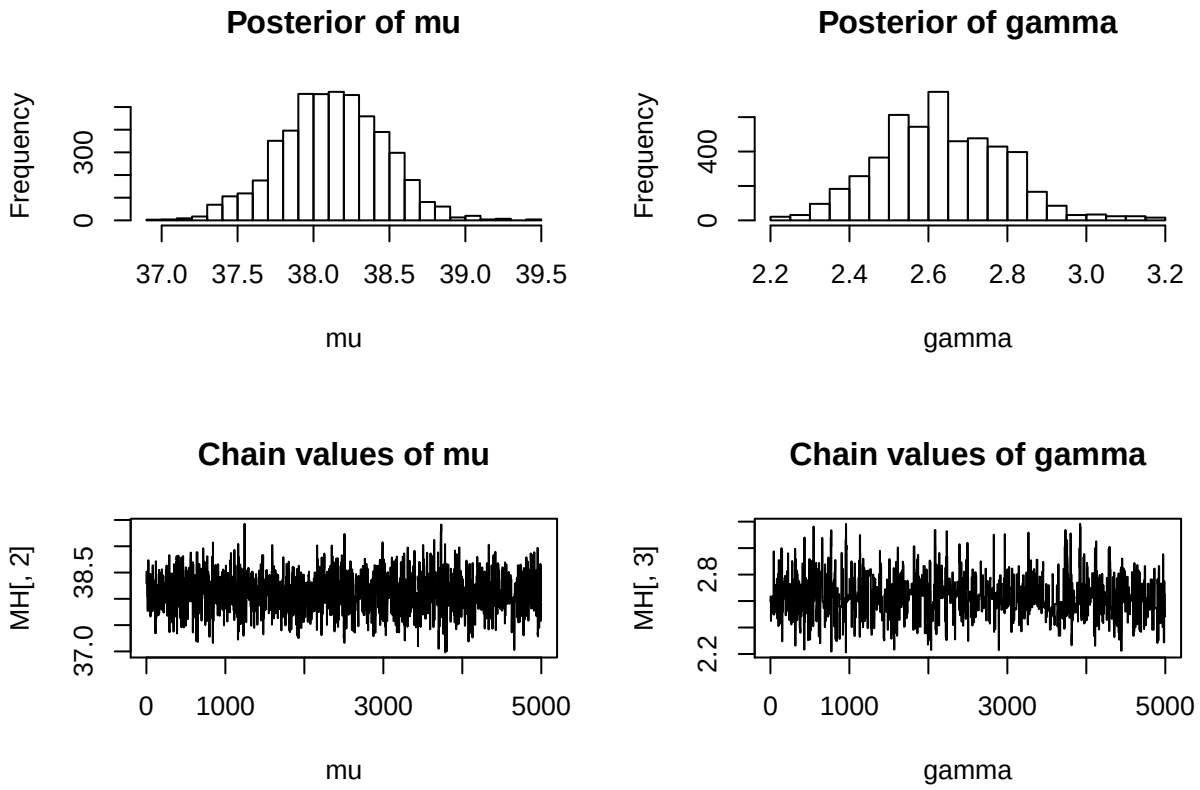
```
si = fitdist(data1,"logis")
si
```

```
## Fitting of the distribution ' logis ' by maximum likelihood
## Parameters:
```

```
##           estimate Std. Error
## location 38.292433  0.3194642
## scale     2.604135  0.1536939
```

Which i used as starting values.
looking at the chain :

```
par(mfrow=c(2,2))
hist(MH[,2],breaks =30, main="Posterior of mu" ,xlab="mu")
hist(MH[,3],breaks =30, main="Posterior of gamma" ,xlab="gamma")
plot(MH[,2], type = "l", xlab="mu" , main = "Chain values of mu")
plot(MH[,3], type = "l", xlab="gamma" , main = "Chain values of gamma")
```



This chain seems to mix well, mainlt due to the choice of starting values, and does not seem to have a problem.
Using `mcmcse` package to compute standard errors for parameter estimates :

```
library(mcmcse)
```

```
## mcmcse: Monte Carlo Standard Errors for MCMC
## Version 1.3-2 created on 2017-07-03.
## copyright (c) 2012, James M. Flegal, University of California, Riverside
##                     John Hughes, University of Colorado, Denver
##                     Dootika Vats, University of Warwick
##                     Ning Dai, University of Minnesota
##  For citation information, type citation("mcmcse").
##  Type help("mcmcse-package") to get started.
```

```
gamma_square= MH[,3]^2
mu_gamma = MH[,2]/MH[,3]
## mu
mcse(MH[,2],size="sqroot")
```

```
## $est
## [1] 38.12313
##
## $se
## [1] 0.008264322
```

```
##Gamma
mcse(MH[,3],size="sqroot")
```

```
## $est
## [1] 2.635685
##
## $se
## [1] 0.005905182
```

```
### Gamma squared
mcse(gamma_square,size="sqroot")
```

```
## $est
## [1] 6.971933
##
## $se
## [1] 0.03135177
```

```
###mu/gamma
mcse(mu_gamma,size="sqroot")
```

```
## $est
## [1] 14.51638
##
## $se
## [1] 0.03321528
```