```r
y = read.csv("gammadat_assign6.txt", sep = " ")
y = y$y
n = length(y)

loglik = function(theta, y){
  alpha = theta[1]
  beta = theta[2]
  if(alpha <= 0 | beta <= 0) return(-Inf)
  n = length(y)
  #if(dgamma(y[i], shape = alpha, rate = beta,  log = TRUE)) print()
  return(sum(sapply(1:n, function(i) dgamma(y[i], shape = alpha, rate = beta,  log = TRUE))))
}

loglik2 = function(theta, y){
  alpha = theta[1]
  beta = theta[2]
  if(alpha <= 0 | beta <= 0) return(-Inf)
  n = length(y)
  #if(dgamma(y[i], shape = alpha, rate = beta,  log = TRUE)) print()
  return(sum(sapply(1:n, function(i) alpha * log(beta) - log(gamma(alpha)) + (alpha - 1) * log(y[i]) - 
}


loglik_slice = function(alpha, beta, y){
  if(alpha <= 0 | beta <= 0) return(-Inf)
  n = length(y)
  #if(dgamma(y[i], shape = alpha, rate = beta,  log = TRUE)) print()
  return(sum(sapply(1:n, function(i) dgamma(y[i], shape = alpha, rate = beta,  log = TRUE))))
}

loglik_neg = function(theta, y){
  return(-loglik(theta, y))
}

loglik_slice_neg = function(alpha, beta, y){
  return(-loglik_slice(alpha, beta, y))
}


alphahat = mean(y)^2 / var(y)
betahat = mean(y) / var(y)

theta_ini = c(alphahat, betahat)

res = optim(par = theta_ini, fn = loglik_neg, y = y)

res$par
alphahat = res$par[1]
betahat = res$par[2]

mean(rgamma(1000, alphahat, rate = betahat))
var(rgamma(1000, alphahat, rate = betahat))
```

```r
B = 2 * alphahat
lambda = 10
gamma = lambda * betahat

#alphavec = seq(from = 0.1, to = B, length = 3000)
#sapply(alphavec, function(x) betahat^(n*x) / gamma(x)^n * prod(y)^(x - 1))
#plot(alphavec , sapply(alphavec, function(x) betahat^(n*x) / gamma(x)^n * prod(y)^(x - 1)))
#plot(alphavec , sapply(alphavec, function(x) n*x*log(betahat) - n * log(gamma(x)) + (x - 1) * sum(log(

lik_alpha = function(x, beta_t){
  if(x <= 0 | x >= B) return(0)
  return(exp(n*x*log(beta_t) - n * log(gamma(x)) + (x - 1) * sum(log(y)) - 350))
}

lik_alpha_neg = function(x, beta_t, ..., y2 = y, n2 = n){
  if(x <= 0 | x >= B) return(0)
  return(-exp(n2*x*log(beta_t) - n2 * log(gamma(x)) + (x - 1) * sum(log(y2)) - 350))
}

optimize(interval = c(0, B), lik_alpha_neg, beta = betahat)

Gibbs <- function(alpha_ini, beta_ini, MCsize1){
  alpha_t = alpha_ini
  beta_t = beta_ini
  alpha_sample = c()
  beta_sample = c()
  #alphavec = seq(from = 0.1, to = B, length = 300)
  for(i in 1:MCsize1){
    #max_idx = which.max(sapply(alphavec, function(x) loglik_alpha(x, beta_t, y2 = y, n2 = n)))
    optim_res = optimize(interval = c(0, B), lik_alpha_neg, beta = beta_t)
    alpha_max = optim_res$minimum
    lik_max = -optim_res$objective

    while(TRUE){
      Y = runif(1, 0, B)
      U = runif(1)
      if(U <= lik_alpha(Y, beta_t) / lik_max){
        alpha_t = Y
        break
      }
    }

    beta_t <- rgamma(1, shape = n * alpha_t + gamma, rate = sum(y) + lambda)
    alpha_sample <- c(alpha_sample, alpha_t)
    beta_sample <- c(beta_sample, beta_t)
  }
  return(list(alpha_sample, beta_sample))
}
res <- Gibbs(alphahat, betahat, 50100)
alpha_sampled <- res[[1]][-(1:100)]
beta_sampled <- res[[2]][-(1:100)]

summary(alpha_sampled)
```

```r
summary(beta_sampled)

#alphahat
#mean(alpha_sampled)
#betahat
#mean(beta_sampled)

hist(alpha_sampled, main = "posterior of alpha")
hist(beta_sampled, main = "posterior of beta")

plot(alpha_sampled, type = "l", main = "posterior of alpha")
plot(beta_sampled, type = "l", main = "posterior of beta")

acf(alpha_sampled, lag.max = 1000, main = "ACF of alpha")
acf(beta_sampled, lag.max = 1000, main = "ACF of beta")

#loglik(c(mean(res[[1]]), mean(res[[2]])), y)
#loglik(c(alphahat, betahat), y)

cor(res[[1]], res[[2]])

mean(alpha_sampled / beta_sampled)

acf(alpha_sampled / beta_sampled, lag.max = 1000)
acf(alpha_sampled - beta_sampled, lag.max = 1000)
plot(alpha_sampled / beta_sampled, type = "l")
```