

Mth422a-Assignment-4

Vijay Soren (211163)

2024-03-28

Ques - 1

part 1

I have written the JAGS code for this question

```
set.seed(2)

library(rjags)

## Warning: package 'rjags' was built under R version 4.3.3

## Loading required package: coda

## Warning: package 'coda' was built under R version 4.3.3

## Linked to JAGS 4.3.1

## Loaded modules: basemod,bugs

Y <- c(64,13,33,18,30,20)
n <- length(Y)
data <- list(Y = Y, n=n)
model_string <- textConnection("model{
  # likelihood
  for(i in 1:n){
    Y[i] ~ dpois(exp(alpha + i*beta))
  }

  #priors
  alpha ~ dnorm(0,0.0001)
  beta ~ dnorm(0,0.0001)
}")

inits <- list( alpha = 0.5,beta = 2)
model <- jags.model(model_string,data = data,inits = inits,n.chains = 2,quiet = TRUE)

# burn in
update(model,10000, progress.bar = "none")
```

```

params <- c("alpha","beta")
samples <- coda.samples(model = model,variable.names = params,n.iter = 20000,progress.bar = "none")

summary(samples)

```

```

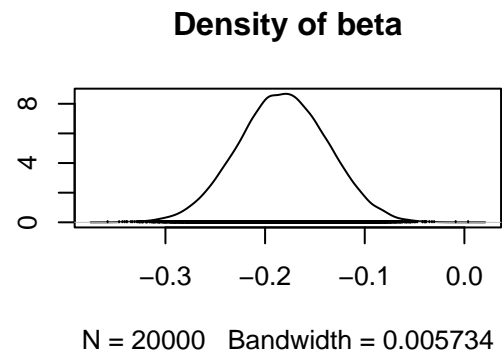
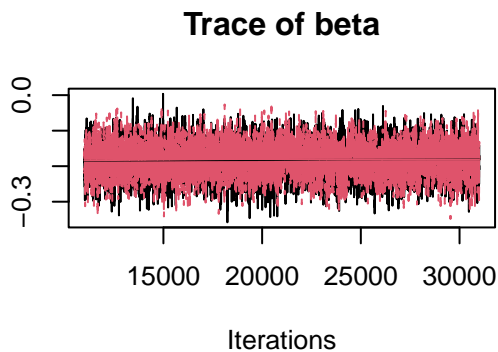
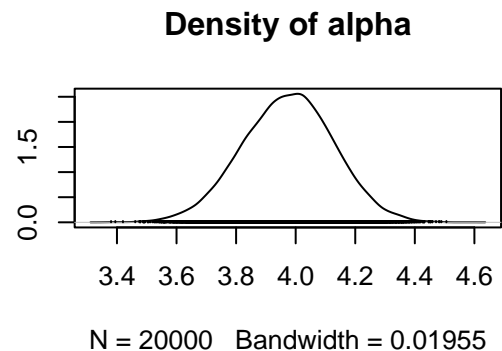
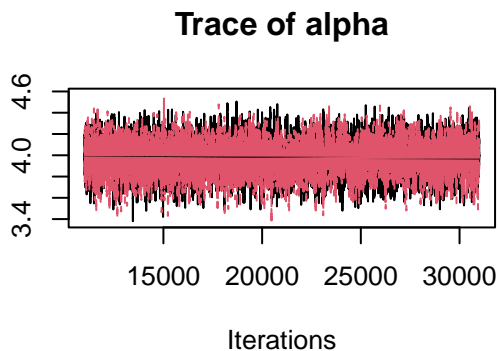
##
## Iterations = 11001:31000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##      Mean      SD Naive SE Time-series SE
## alpha  3.9738 0.15358 0.0007679      0.0026106
## beta  -0.1822 0.04504 0.0002252      0.0007631
##
## 2. Quantiles for each variable:
##
##      2.5%    25%    50%    75%    97.5%
## alpha  3.6687 3.8697 3.977 4.0783 4.26840
## beta  -0.2707 -0.2125 -0.182 -0.1514 -0.09489

```

```

plot(samples)

```



Yes, MCMC sampler has converged w can see from above trace plot.
 and mean values of α is 3.9806 and β is -0.1841
 so, YES ,the rate of discovery is changing over time and it is decreasing.

part 2

here code for MCMC sampler

Here , the candidate distribution is $N(\beta_j, \sigma_j^2)$

```
Y      <- c(64,13,33,18,30,20)

post   <- function(Y,t,beta,pri.sd=10){
  mn <- exp(beta[1] + t*beta[2])
  l  <- prod(dpois(Y,mn))
  p  <- prod(dnorm(beta,0,pri.sd))
  return(l*p)}

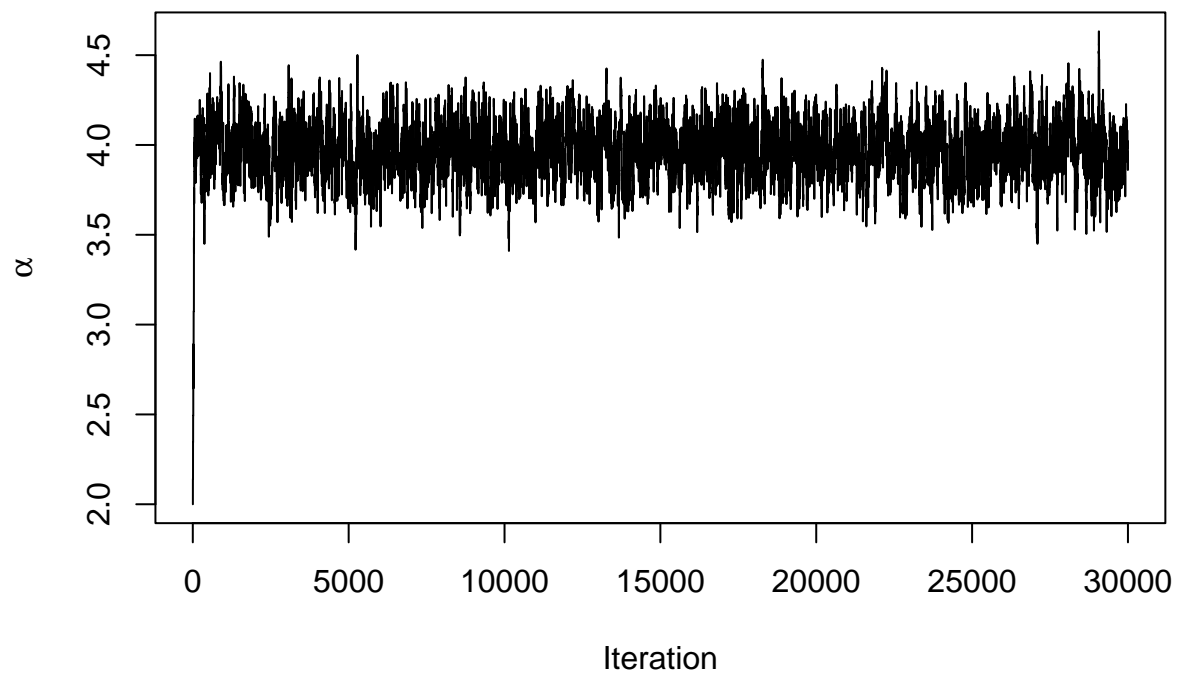
MCMC <- function(Y, beta.init, iters,cansd){

  # chain initiation
  beta <- beta.init
  # define chains
  beta.chain <- matrix(NA, iters,2)

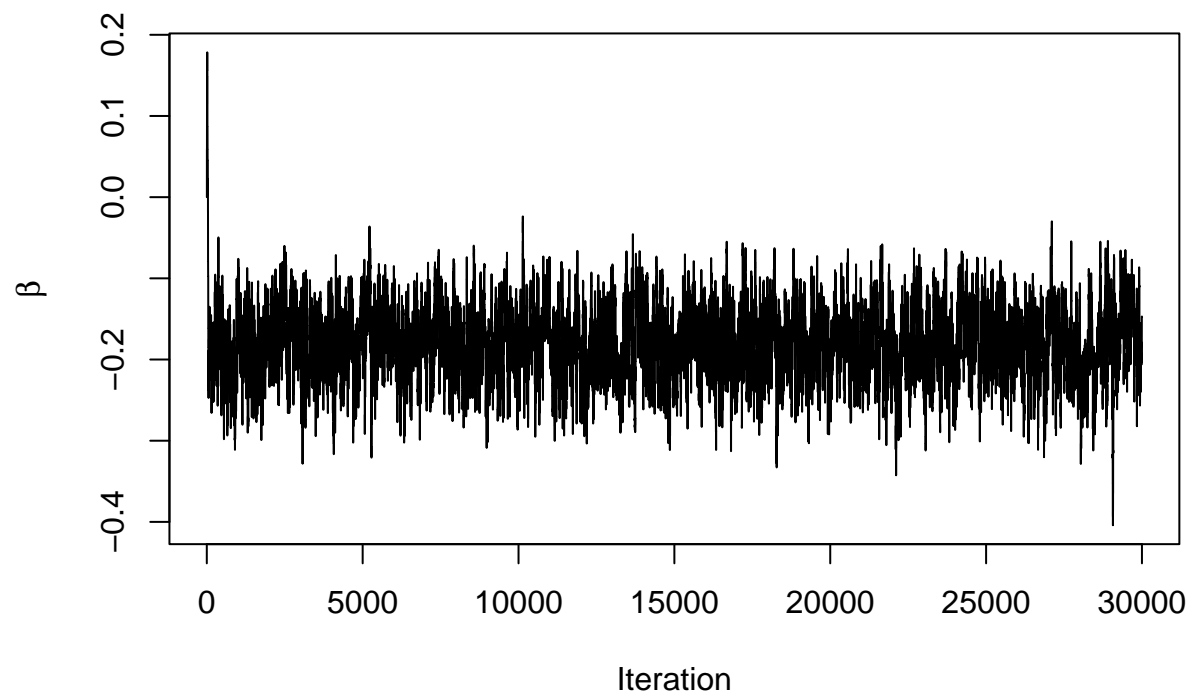
  t <- 1:length(Y)
  # start MCMC
  for(iter in 1:iters){
    for(j in 1:2){
      can <- beta
      can[j] <- rnorm(1,beta[j],cansd[j])
      R <- post(Y,t,can)/post(Y,t,beta)
      if(runif(1)<R){
        beta <- can
      }
    }
    beta.chain[iter,] <- beta
  }

  # return chains
  out <- list(beta.chain = beta.chain)
  return(out)}

MCMC.out <- MCMC(Y, beta.init = c(2,0),cansd = c(0.2,0.05), iters = 30000)
plot(MCMC.out$beta.chain[,1],type = "l",ylab = expression(alpha), xlab = "Iteration")
```



```
plot(MCMC.out$beta.chain[,2],type = "l",ylab = expression(beta), xlab = "Iteration")
```



```
### acceptance ratio
S <- 30000
print("Acceptance Ratio for alpha and beta")
```

```
## [1] "Acceptance Ratio for alpha and beta"
```

```
acc_rate <- colMeans(MCMC.out$beta.chain[-1,] != MCMC.out$beta.chain[-S,])
acc_rate
```

```
## [1] 0.4065469 0.4548485
```

Acceptance ratio for α and β is 0.409 and 0.456 respectively which is acceptable and good.

Que 2

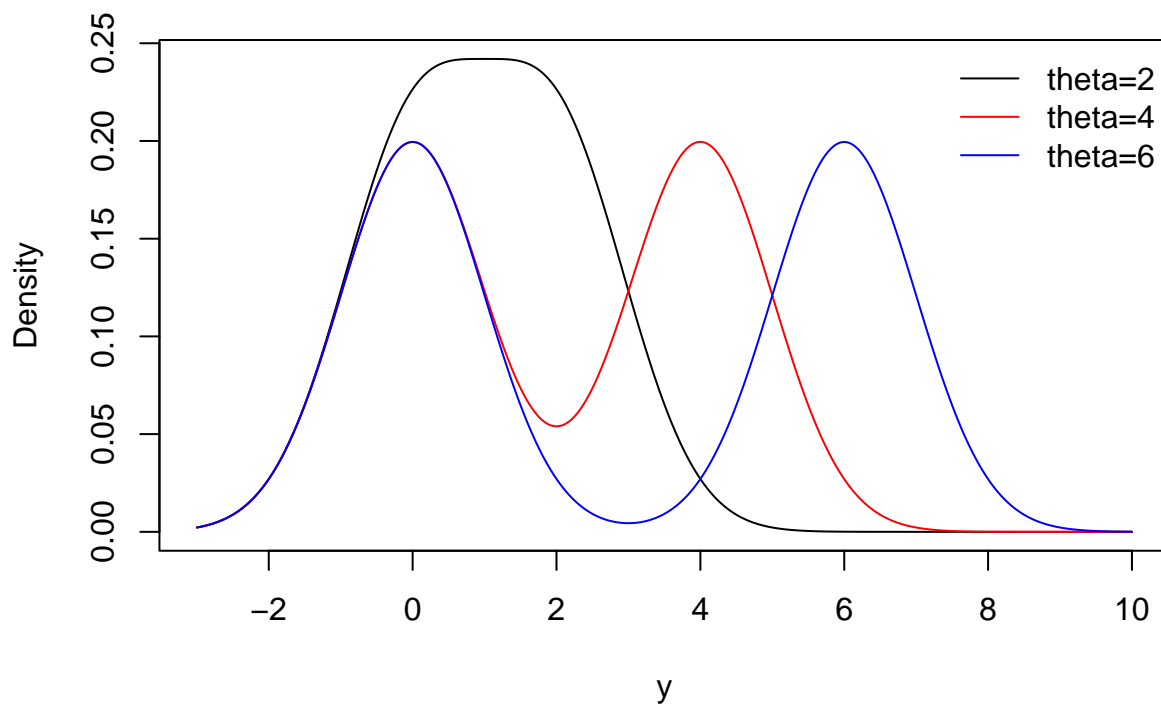
part (a)

```
set.seed(27695)
theta_true <- 4
n <- 30
B <- rbinom(n, 1, 0.5)
Y <- rnorm(n, B*theta_true, 1)
```

We use $B \sim \text{bernoulli}(0.5)$ which produce 1, or 0 of size n with probability 0.5 and $f(y|\theta)$ is mixture of two Normal distribution one is $N(0, 1)$ and other is $N(\theta, 1)$ and both are 50-50 chance of drawing sample of model.

part (b)

```
y <- seq(-3,10,0.01)
plot(y,0.5*dnorm(y,0,1) + 0.5*dnorm(y,2,1),type="l",ylab="Density")
lines(y,0.5*dnorm(y,0,1) + 0.5*dnorm(y,4,1),col="red")
lines(y,0.5*dnorm(y,0,1) + 0.5*dnorm(y,6,1),col="blue")
legend("topright",c("theta=2","theta=4","theta=6"),lty=1,col=c("black","red","blue"),bty="n")
```



part (c)

```
library(stats4)

set.seed(27695)
theta_true <- 4
n <- 30
B <- rbinom(n,1,0.5)
Y <- rnorm(n,B*theta_true,1)
```

```

nlp <- function(theta)
{
  like <- 0.5*dnorm(Y, 0, 1) + 0.5*dnorm(Y, theta, 1)
  prior <- dnorm(theta, 0, 10)
  neg_log_post <- -sum(log(like)) - log(prior)

  return(neg_log_post)
}

map_est <- mle(nlp, start = list(theta=1))
map_est_val <- map_est@coef
sd <- as.numeric(sqrt(vcov(map_est)))
map <- 4.18

```

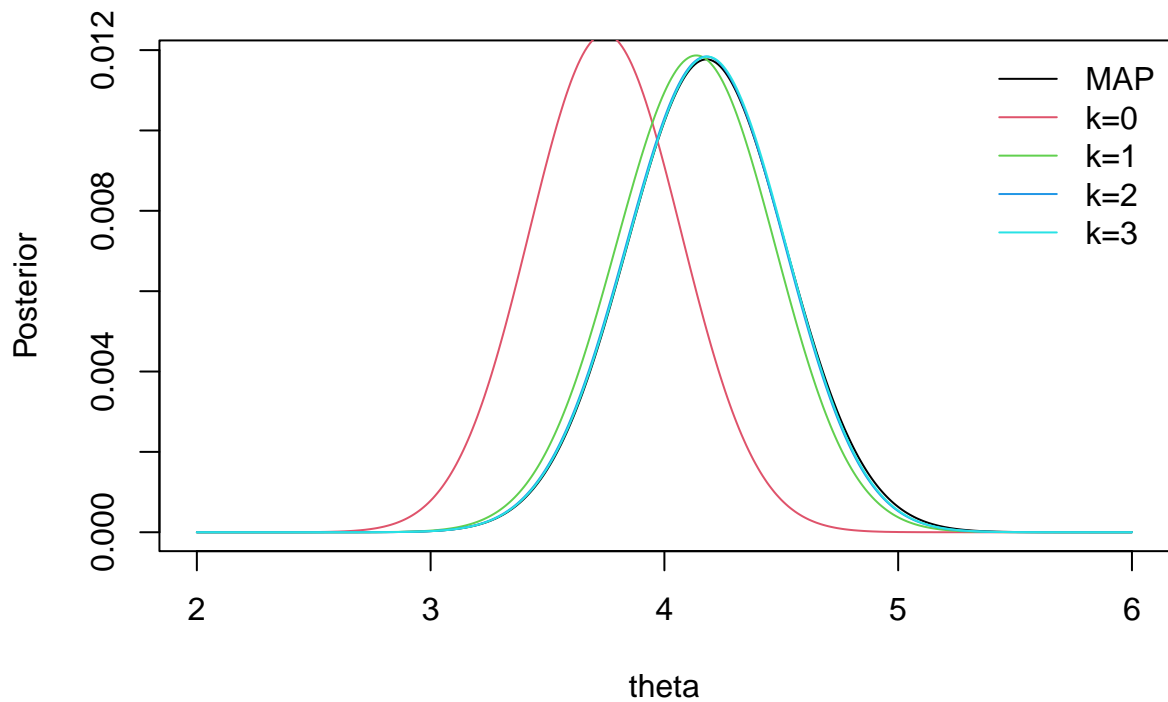
part (d)

```

posterior <- function(theta,Y,k){
  post <- dnorm(theta,0,sqrt(10^k))
  for(i in 1:length(Y)){
    post<-post*(0.5*dnorm(Y[i],0,1)+
                0.5*dnorm(Y[i],theta,1))
  }
  return(post/sum(post))}

theta <- seq(2,6,0.01)
map <- dnorm(theta,map,sd)
plot(theta,map/sum(map),type="l",ylab="Posterior")
lines(theta,posterior(theta,Y,0),col=2)
lines(theta,posterior(theta,Y,1),col=3)
lines(theta,posterior(theta,Y,2),col=4)
lines(theta,posterior(theta,Y,3),col=5)
legend("topright",c("MAP", "k=0", "k=1", "k=2", "k=3"),
      col=1:5,lty=1,bty="n")

```



part (e)

```
library(rjags)

data <- list(n=n,Y=Y)
model_string <- textConnection("model{
  # Likelihood
  for(i in 1:n){
    Y[i] ~ dnorm(B[i]*theta,1)
  }

  # Priors
  for(i in 1:n){
    B[i] ~ dbern(0.5)
  }
  theta ~ dnorm(0, 0.01)
}")

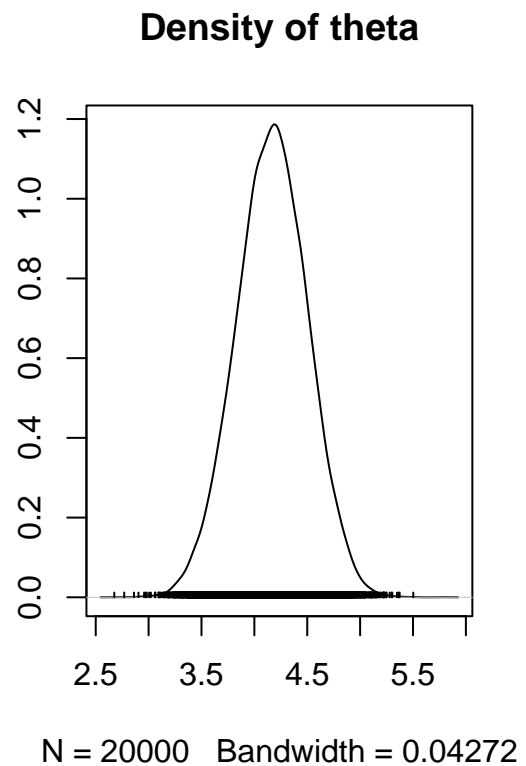
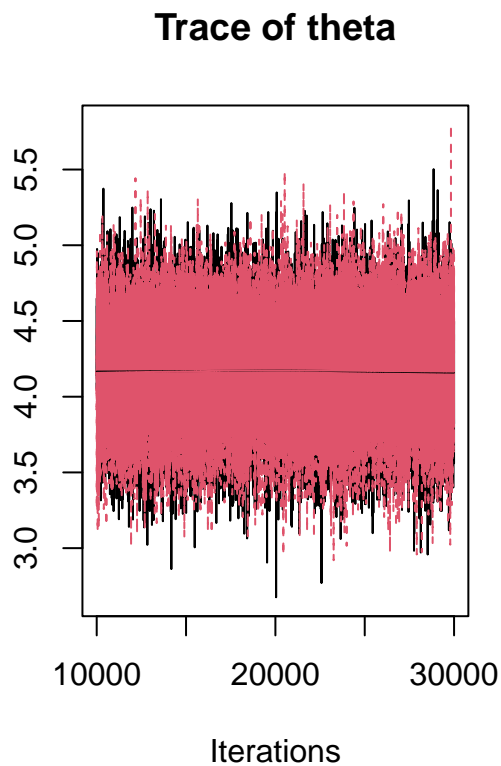
inits <- list(theta=1)
model <- jags.model(model_string,data = data, inits=inits, n.chains=2,quiet=TRUE)
update(model, 10000, progress.bar="none")
params <- c("theta")
samples <- coda.samples(model,
                        variable.names=params,
```



```
n.iter=20000, progress.bar="none")
summary(samples)
```

```
##
## Iterations = 10001:30000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean           SD      Naive SE Time-series SE
##          4.166915      0.335536      0.001678      0.002140
##
## 2. Quantiles for each variable:
##
##  2.5%   25%   50%   75%  97.5%
##  3.501  3.942  4.170  4.396  4.817
```

```
plot(samples)
```



Que 3

part (a)

Convergence may be slow due to number of parameters are more than observation so, it is likely that not all parameters are identifiable.

part (b)

JAGS with $\lambda = 10$ and $a = 1, b = 1$

```
library(rjags)
Y <- 10
a <- 1
b <- 1
lambda <- 10

data <- list(Y = Y, a = a, b = b, lambda = lambda)

model_string <- textConnection("model{
  # likelihood
  Y ~ dbin(p,n)

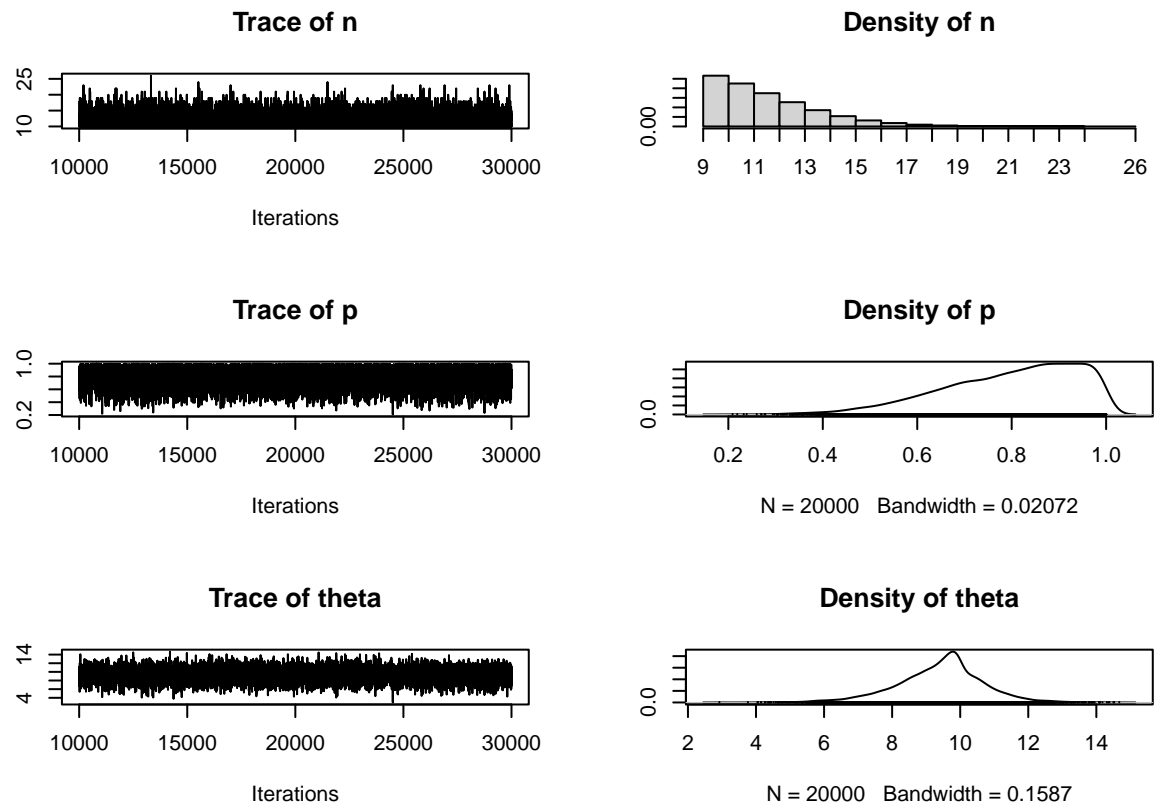
  #priors
  n ~ dpois(lambda)
  p ~ dbeta(a,b)
  theta <- n*p
}")
inits <- list(p = 0.8, n = 1)
model <- jags.model(model_string, data = data, inits = inits, quiet = TRUE)
# burn in
update(model, 10000, progress.bar = "none")
params <- c("n", "p", "theta")
samples <- coda.samples(model = model, variable.names = params, n.iter = 20000, progress.bar = "none")

summary(samples)

##
## Iterations = 10001:30000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD Naive SE Time-series SE
## n      12.0424  2.0207 0.014288      0.025318
## p       0.7975  0.1417 0.001002      0.001764
## theta   9.4022  1.2648 0.008943      0.007497
##
## 2. Quantiles for each variable:
```

```
##
##          2.5%    25%    50%    75%  97.5%
## n      10.0000 10.0000 12.0000 13.0000 17.000
## p       0.4788 0.7033 0.8214 0.9126 0.992
## theta   6.6289 8.6789 9.5264 10.1330 11.815
```

```
plot(samples)
```



part (c)

JAGS with $\lambda = 10$ and $a = 10, b = 10$

```
Y <- 10
a <- 10
b <- 10
lambda <- 10

data <- list(Y = Y, a = a, b = b, lambda = lambda)
model_string <- textConnection("model{
  # likelihood
  Y ~ dbin(p,n)

  #priors
  n ~ dpois(lambda)
```

```

    p ~ dbeta(a,b)
    theta <- n*p
  }")

inits <- list(p = 0.8,n = 1)
model <- jags.model(model_string,data = data,inits = inits, quiet = TRUE)
# burn in
update(model,10000, progress.bar = "none")
params <- c("n","p","theta")
samples <- coda.samples(model = model,variable.names = params,n.iter = 20000,progress.bar = "none")

summary(samples)

```

```

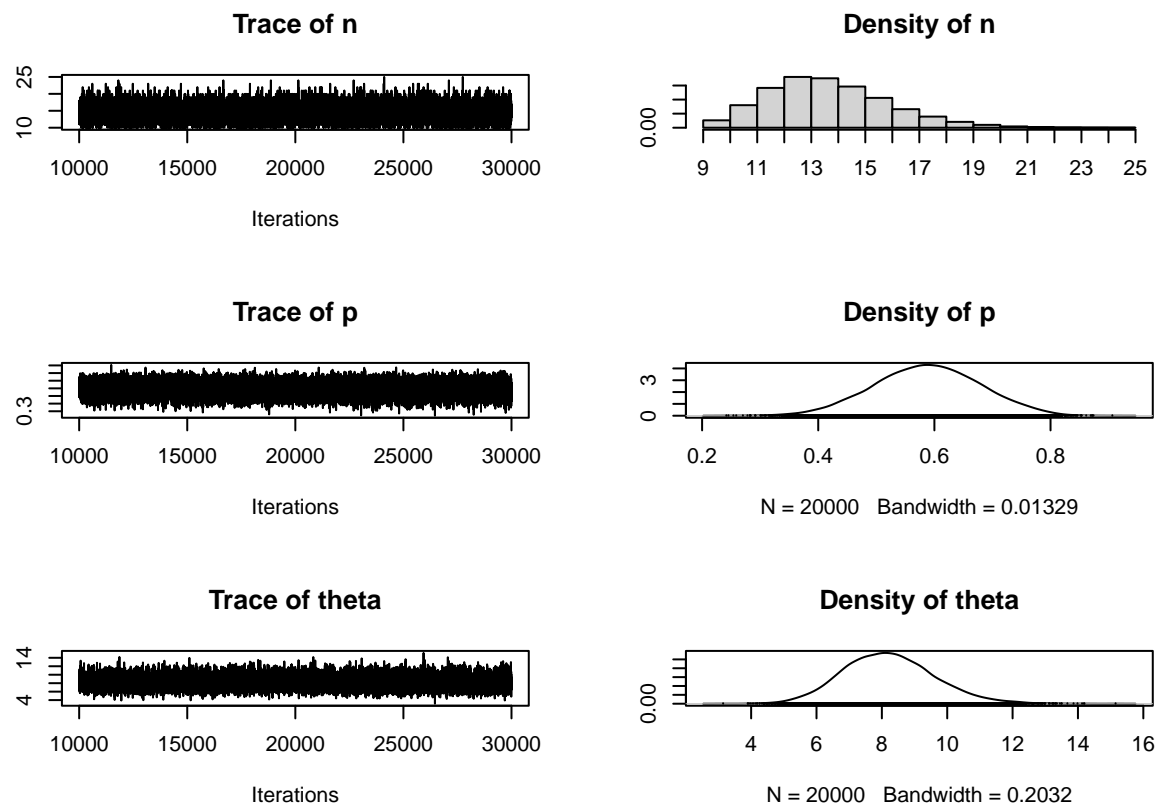
##
## Iterations = 10001:30000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean          SD Naive SE Time-series SE
## n      14.1247 2.22981 0.0157672      0.0190205
## p       0.5876 0.09085 0.0006424      0.0007622
## theta   8.2154 1.38966 0.0098264      0.0083014
##
## 2. Quantiles for each variable:
##
##           2.5%      25%      50%      75%      97.5%
## n      10.000 13.0000 14.0000 15.0000 19.0000
## p       0.408  0.5258  0.5881  0.6501  0.7626
## theta   5.663  7.2450  8.1526  9.1066 11.1536

```

```

plot(samples)

```



Now the convergence is better than previous case

Que 4

we assume for placebo observation $N(\mu, \sigma^2)$ and for treatment is $N(\mu + \delta, \sigma^2)$

We have to whether $\delta = 0$ or not

We use Bayesian two-sample t test and

the jeffrey's priors for μ, δ, σ^2 is directly proportional to $1/(\sigma^2)^2$ and marginal distributio of δ given data is $t_n[\bar{Y}_2 - \bar{Y}_1, \sigma^2(1/n_1 + 1/n_2)]$ and $\hat{\sigma}^2 = (n_1 \hat{s}_1^2 + n_2 \hat{s}_2^2)/n$, where $n = n_1 + n_2$ and $\hat{s}_1^2 = (\sum_{i=1}^{n_1} (Y_i - \bar{Y}_1)^2)/n_1$ and $\hat{s}_2^2 = (\sum_{i=1}^{n_1} (Y_i - \bar{Y}_2)^2)/n_2$

```
### assgn4 - ques4
Y1 <- c(2,-3.1,-1,0.2,0.3,0.4)
Y2 <- c(-3.5,-1.6,-4.6,-0.9,-5.1,0.1)

# Statistics from group 1
Ybar1 <- mean(Y1)
s21 <- sum((Y1 - Ybar1)^2)/length(Y1)
n1 <- length(Y1)

# Statistics from group 2
Ybar2 <- mean(Y2)
s22 <- sum((Y2 - Ybar2)^2)/length(Y2)
```

```

n2 <- length(Y2)

# Posterior of the difference assuming equal variance
delta_hat <- Ybar2 - Ybar1
delta_hat

## [1] -2.4

s2 <- (n1*s21 + n2*s22)/(n1+n2)
scale <- sqrt(s2)*sqrt(1/n1 + 1/n2)
df <- n1+n2
cred_int <- delta_hat + scale * qt(c(0.025, 0.975), df = df)
cred_int

```

```
## [1] -4.6058799 -0.1941201
```

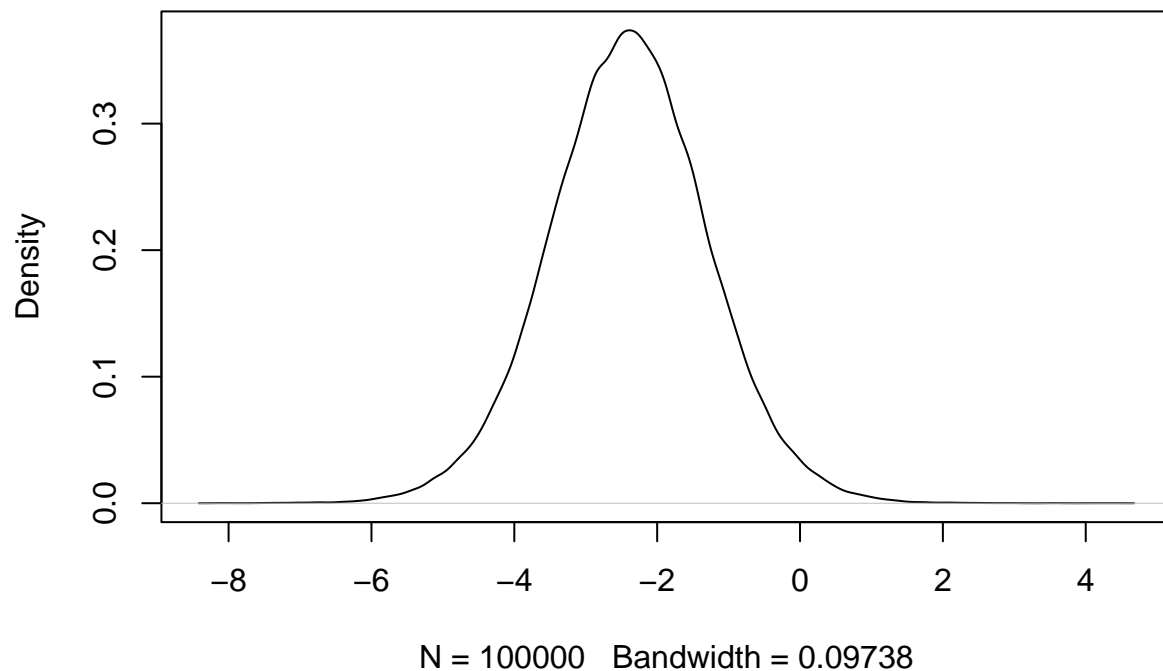
```

# Posterior of delta assuming unequal variance using MC sampling
mu1 <- Ybar1 + sqrt(s21/n1)*rt(1e5,df = df)
mu2 <- Ybar2 + sqrt(s22/n2)*rt(1e5,df = df)
delta <- mu2 - mu1

plot(density(delta), main = "Posterior distribution of the difference in means")

```

Posterior distribution of the difference in means



```
quantile(delta, c(0.025, 0.975)) # 95% credible set
```

```
##          2.5%          97.5%  
## -4.6101395 -0.1935554
```

```
#### sensitivity
```

```
data <- list(n = 6, Y1 = Y1, Y2 = Y2)  
library(rjags)
```

```
model_string <- textConnection("model{  
  # likelihood  
  for(i in 1:n){  
    Y1[i] ~ dnorm(mu, tau)  
    Y2[i] ~ dnorm(mu + delta,tau)  
  }  
  
  #priors  
  mu ~ dnorm(0,0.0001)  
  delta ~ dnorm(0,0.0001)  
  tau ~ dgamma(0.01,0.01)  
}")
```

```
model <- jags.model(model_string,data = data, n.chains=2,quiet=TRUE)  
update(model, 10000, progress.bar="none")  
params <- c("delta")  
samples <- coda.samples(model,  
  variable.names=params,  
  n.iter=20000, progress.bar="none")
```

```
summary(samples)
```

```
##  
## Iterations = 10001:30000  
## Thinning interval = 1  
## Number of chains = 2  
## Sample size per chain = 20000  
##  
## 1. Empirical mean and standard deviation for each variable,  
##    plus standard error of the mean:  
##  
##           Mean           SD      Naive SE Time-series SE  
##      -2.385376      1.249139      0.006246      0.010920  
##  
## 2. Quantiles for each variable:  
##  
##      2.5%      25%      50%      75%      97.5%  
## -4.8704 -3.1630 -2.3902 -1.6125  0.1139
```

```
plot(samples)
```

