

# MTH422A Assignment5

Vijay Soren (211163)

2024-04-15

2

3

```
### que3
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(563,10)
N_s <- c(2820,27)
c <- 1
n <- 2
```

```
# Baye factor
```

```
lgconst <- lgamma(Y[1] + Y[2] + 1) + (Y[1] + 1) * log(N_s[1]) + (Y[2] + 1) * log(N_s[2]) -
  ((Y[1] + Y[2] + 1) * log(N_s[1] + N_s[2]) + lgamma(Y[1] + 1) + lgamma(Y[2] + 1))
```

```
# Calculate the exponential of lgconst
```

```
result <- exp(lgconst)
```

```
bf21_c1 <- result*c*pgamma(c,shape = Y[1]+Y[2]+1, rate = N_s[1] + N_s[2])/(pgamma(c,Y[1] + 1,N_s[1])*pgamma(c,Y[2] + 1,N_s[2]))
```

```
data <- list(Y = Y, n = length(Y), N_s = N_s,c= c)
```

```
model_string <- textConnection("model{
```

```
# Likelihood
```

```
for(i in 1:n){
```

```

Y[i] ~ dpois(N_s[i]*lambda[i])
like[i] <- dpois(Y[i],N_s[i]*lambda[i])
}

#prior
for(i in 1:n){
  lambda[i] ~ dunif(0,c)
}
})

model <- jags.model(model_string, data = data, n.chains = 2,quiet = TRUE)
update(model,10000,progress.bar = "none")

samples <- coda.samples(model, variable.names = c("like"),
  n.iter = 50000,progress.bar = "none")

## dic
DIC1 <- dic.samples(model,n.iter = 50000 ,progress.bar = "none")
DIC1

```

```

## Mean deviance: 14.34
## penalty 1.986
## Penalized deviance: 16.32

```

*# Compute WAIC*

```

like <- rbind(samples[[1]],samples[[2]]) # Combine the two chains
fbar <- colMeans(like)
Pw <- sum(apply(log(like),2,var))
WAIC1 <- -2*sum(log(fbar)) + 2*Pw
WAIC1

```

```
## [1] 15.74768
```

### c=10

```
c <- 10
```

*# Baye factor*

```

lgconst <- lgamma(Y[1] + Y[2] + 1) + (Y[1] + 1) * log(N_s[1]) + (Y[2] + 1) * log(N_s[2]) -
  ((Y[1] + Y[2] + 1) * log(N_s[1] + N_s[2]) + lgamma(Y[1] + 1) + lgamma(Y[2] + 1))

```

*# Calculate the exponential of lgconst*

```
result <- exp(lgconst)
```

```
bf21_c10 <- result * c * pgamma(c,shape = Y[1]+Y[2]+1, rate = N_s[1] + N_s[2])/(pgamma(c,Y[1] + 1,N_s[1]
```

```
data <- list(Y = Y, n = length(Y), N_s = N_s,c= c)
```

```
model_string <- textConnection("model{
```

```

# Likelihood
for(i in 1:n){
Y[i] ~ dpois(N_s[i]*lambda[i])
like[i] <- dpois(Y[i],N_s[i]*lambda[i])
}

#prior
for(i in 1:n){
  lambda[i] ~ dunif(0,c)
}
})

model <- jags.model(model_string, data = data, n.chains = 2,quiet = TRUE)
update(model,10000,progress.bar = "none")

samples_10 <- coda.samples(model, variable.names = c("like"),
                           n.iter = 50000,progress.bar = "none")

## dic
DIC_10 <- dic.samples(model,n.iter = 50000 ,progress.bar = "none")
DIC_10

```

```

## Mean deviance: 14.33
## penalty 1.986
## Penalized deviance: 16.32

```

*# Compute WAIC*

```

like <- rbind(samples_10[[1]],samples_10[[2]]) # Combine the two chains
fbar <- colMeans(like)
Pw <- sum(apply(log(like),2,var))
WAIC_10 <- -2*sum(log(fbar)) + 2*Pw
WAIC_10

```

```

## [1] 15.74132

```

```

BF21 <- c(bf21_c1,bf21_c10)
C <- c(1,10)
DIC <- c(14.35,14.35)
Waic <- c(WAIC1,WAIC_10)
tab <- data.frame(C,BF21,DIC,Waic)
print(tab)

```

```

##      C      BF21    DIC    Waic
## 1   1 0.7155665 14.35 15.74768
## 2  10 7.1544884 14.35 15.74132

```

So, We can conclude that  $M1$  is best model

```
### Assign-5 que-4
```

```
library(geoR)
```

```
## Warning: package 'geoR' was built under R version 4.3.3
```

```
## -----
## Analysis of Geostatistical Data
## For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
## geoR version 1.9-4 (built on 2024-02-14) is now loaded
## -----
```

```
library(rjags)
Y <- gambia$pos
```

```
X <- gambia[,4:8]
X <- scale(X)
data <- list(Y=Y,X=X,n=length(Y))
```

```
# Fit logistic model
model_string <- textConnection("model{
  for(i in 1:n){
    Y[i] ~ dbern(pi[i])
    logit(pi[i]) <- beta[1] + X[i,1]*beta[2] +
                      X[i,2]*beta[3] + X[i,3]*beta[4] +
                      X[i,4]*beta[5] + X[i,5]*beta[6]

  }

  #priors
  for(j in 1:6){
    beta[j] ~ dnorm(0,0.01)
  }

  # Posterior predictive checks
  for(i in 1:n){
    Y1[i] ~ dbern(pi[i])
  }
  D[1] <- mean(Y1[]^2) - mean(Y1[])^2
  D[2] <- mean(Y1[])

}")

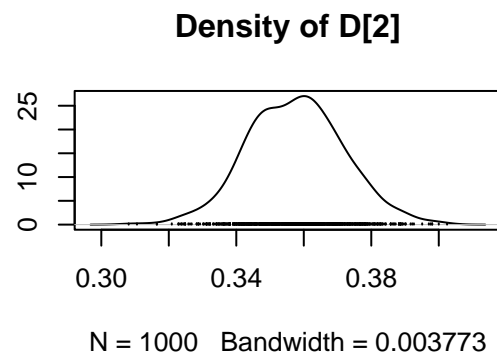
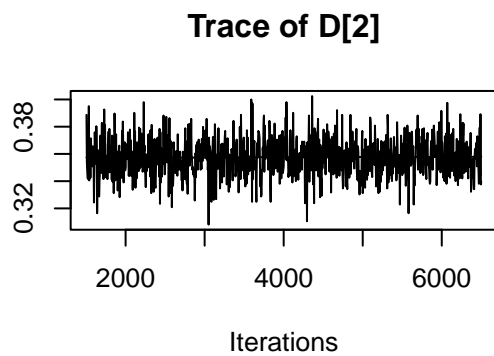
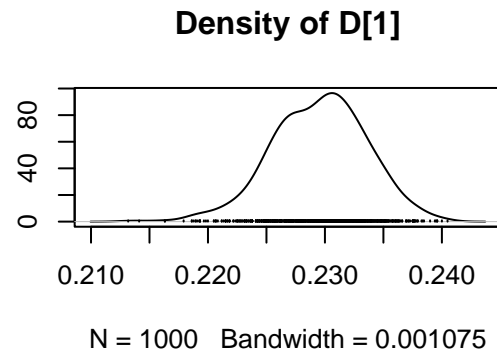
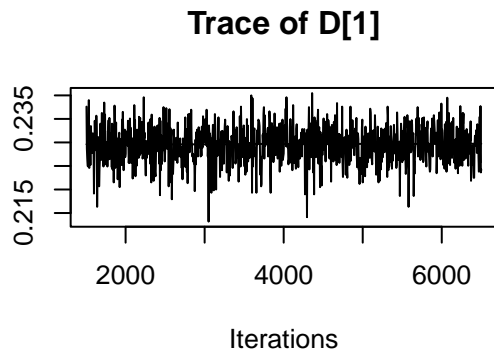
model <- jags.model(model_string,data = data, n.chains=1,quiet=TRUE)
```

```

update(model, 500, progress.bar = "none")

samples <- coda.samples(model, variable.names = c("D"),
                        n.iter = 5000, thin = 5, progress.bar = "none")
plot(samples)

```



```

D.m <- samples[[1]]

##Compute the Bayesian p-values
D0 <- c(var(Y), mean(Y))
Dnames <- c("Var Y", "Mean Y")

# Compute the test stats for the models
pval1 <- rep(NA, 2)
names(pval1) <- c("Var Y", "Mean Y")

for(j in 1:2){

  pval1[j] <- mean(D.m[, j] > D0[j])

}

```

```
print(pval1)
```

```
## Var Y Mean Y  
## 0.506 0.506
```

## 5

```
### assign5 ques5  
Y <- WWUsage  
Y <- as.numeric(Y)  
data1 <- list( Y = Y )  
  
## ar1  
Ar1_model <- textConnection("model{  
  for(t in 2:100){  
    Y[t] ~ dnorm(mu[t],tau)  
    mu[t] <- beta[1] + beta[2]*Y[t-1]  
    like[t] <- dnorm(Y[t],mu[t],tau)  
  }  
  ## priors  
  ## priors  
  for(j in 1:2){  
    beta[j] ~ dnorm(0,0.01)  
  }  
  
  tau ~ dgamma(0.1,0.1)  
  
}")  
  
model <- jags.model(Ar1_model,data = data1, n.chains=2,quiet=TRUE)  
  
update(model, 10000, progress.bar = "none")  
  
## dic  
dic1 <- dic.samples(model,n.iter = 50000)  
print(dic1)  
  
## Mean deviance: 626.5  
## penalty 3.012  
## Penalized deviance: 629.5  
  
samps1 <- coda.samples(model, variable.names = c("like"),  
  n.iter = 50000, thin = 5, progress.bar = "none")
```

```

# Compute WAIC

like <- rbind(samps1[[1]],samps1[[2]]) # Combine the two chains
fbar <- colMeans(like)
Pw <- sum(apply(log(like),2,var))
WAIC1 <- -2*sum(log(fbar)) + 2*Pw
WAIC1

```

```
## [1] 629.1302
```

```

## ar2
Ar2_model <- textConnection("model{
  for(t in 3:100){
    Y[t] ~ dnorm(mu[t],tau)
    mu[t] <- beta[1] + beta[2]*Y[t-1] + beta[3]*Y[t-2]
    like[t] <- dnorm(Y[t],mu[t],tau)
  }
  ## priors
  for(j in 1:3){
    beta[j] ~ dnorm(0,0.01)
  }

  tau ~ dgamma(0.1,0.1)

}")

model <- jags.model(Ar2_model,data = data1, n.chains=2,quiet=TRUE)

update(model, 10000, progress.bar = "none")

## dic
dic2 <- dic.samples(model,n.iter = 50000)
print(dic2)

```

```

## Mean deviance: 517.7
## penalty 4.025
## Penalized deviance: 521.8

```

```

# Compute WAIC
samps2 <- coda.samples(model, variable.names = c("like"),
                        n.iter = 50000, thin = 5, progress.bar = "none")

like <- rbind(samps2[[1]],samps2[[2]]) # Combine the two chains
fbar <- colMeans(like)
Pw <- sum(apply(log(like),2,var))
WAIC2 <- -2*sum(log(fbar)) + 2*Pw
WAIC2

```

```
## [1] 521.3996
```

```
## ar3
Ar3_model <- textConnection("model{
  for(t in 4:100){
    Y[t] ~ dnorm(mu[t],tau)
    mu[t] <- beta[1] + beta[2]*Y[t-1] + beta[3]*Y[t-2] + beta[4]*Y[t-3]
    like[t] <- dnorm(Y[t],mu[t],tau)
  }
  ## priors
  for(j in 1:4){
    beta[j] ~ dnorm(0,0.01)
  }

  tau ~ dgamma(0.1,0.1)

}")

model <- jags.model(Ar3_model,data = data1, n.chains=2,quiet=TRUE)

update(model, 10000, progress.bar = "none")

## dic
dic3 <- dic.samples(model,n.iter = 50000)
print(dic3)
```

```
## Mean deviance: 507.4
## penalty 5.503
## Penalized deviance: 512.9
```

```
# Compute WAIC
samps3 <- coda.samples(model, variable.names = c("like"),
  n.iter = 50000, thin = 5, progress.bar = "none")

like <- rbind(samps3[[1]],samps3[[2]]) # Combine the two chains
fbar <- colMeans(like)
Pw <- sum(apply(log(like),2,var))
WAIC3 <- -2*sum(log(fbar)) + 2*Pw
WAIC3
```

```
## [1] 511.6005
```

```
## ar4
Ar4_model <- textConnection("model{
  for(t in 5:100){
    Y[t] ~ dnorm(mu[t],tau)
    mu[t] <- beta[1] + beta[2]*Y[t-1] + beta[3]*Y[t-2] + beta[4]*Y[t-3] + beta[5]*Y[t-4]
    like[t] <- dnorm(Y[t],mu[t],tau)

  }
  ## priors
  for(j in 1:5){
    beta[j] ~ dnorm(0,0.01)
```



```

}

tau ~ dgamma(0.1,0.1)

}"))

model <- jags.model(Ar4_model,data = data1, n.chains=2,quiet=TRUE)

update(model, 10000, progress.bar = "none")

## dic
dic4 <- dic.samples(model,n.iter = 50000)
print(dic4)

## Mean deviance: 489.6
## penalty 5.65
## Penalized deviance: 495.2

# Compute WAIC
samps4 <- coda.samples(model, variable.names = c("like"),
                        n.iter = 50000, thin = 5, progress.bar = "none")

like <- rbind(samps4[[1]],samps4[[2]]) # Combine the two chains
fbar <- colMeans(like)
Pw <- sum(apply(log(like),2,var))
WAIC4 <- -2*sum(log(fbar)) + 2*Pw
WAIC4

## [1] 493.1161

### table for comparision
L <- c(1:4)
DIC <- c(626.5,517.9 ,506.9,487.8)
WAIC <- c(WAIC1,WAIC2,WAIC3,WAIC4)
model_compar <- data.frame(L,DIC,WAIC)
print(model_compar)

```

```

##   L   DIC     WAIC
## 1 1 626.5 629.1302
## 2 2 517.9 521.3996
## 3 3 506.9 511.6005
## 4 4 487.8 493.1161

```

From above information L=4 best