

# Bayesian Data Analysis

*BIOS719 Generalized Linear Models*

## Bayesian logistic and Poisson regression: Revisit the Beetle data example

```
#### 7.3.1 Beetle data example
## Create the dataset
beetle <- as.data.frame(cbind( dose=c(1.6907,1.7242,1.7552,1.7842,1.8113,1.8369,1.861,1.8839),
                                n=c(59,60,62,56,63,59,62,60),
                                y=c(6,13,18,28,52,53,61,60)
                                ))

## Print the data
beetle

##   dose  n  y
## 1 1.69 59  6
## 2 1.72 60 13
## 3 1.76 62 18
## 4 1.78 56 28
## 5 1.81 63 52
## 6 1.84 59 53
## 7 1.86 62 61
## 8 1.88 60 60
```

We will use R2jags package to fit Bayesian models. Before installing the package in R, you need to download and install JAGS to your laptop (<https://sourceforge.net/projects/mcmc-jags/files/JAGS/4.x/>).

We will fit a logistic regression model:  $\text{logit}(\pi_i) = \beta_0 + \beta_1 * \text{Dose}_i$

```
#### Frequentist logistic regression
fit.logit <- glm(cbind(y, n-y)~dose, family=binomial(link="logit"), data=beetle)
summary(fit.logit)

##
## Call:
## glm(formula = cbind(y, n - y) ~ dose, family = binomial(link = "logit"),
##      data = beetle)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.594  -0.394   0.833   1.259   1.594
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -60.72      5.18  -11.7   <2e-16 ***
## dose           34.27      2.91   11.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 284.202  on 7  degrees of freedom
## Residual deviance:  11.232  on 6  degrees of freedom
## AIC: 41.43
```

```
##  
## Number of Fisher Scoring iterations: 4  
library(R2jags)  
  
## Loading required package: rjags  
## Loading required package: coda  
## Linked to JAGS 4.3.0  
## Loaded modules: basemod,bugs  
##  
## Attaching package: 'R2jags'  
## The following object is masked from 'package:coda':  
##  
##      traceplot
```

```

## JAGS model
logit.jags <- function(){

  for(i in 1:N) {
    r[i] ~ dbin(p[i], n[i])          # Binomial likelihood
    logit(p[i]) <- beta[1] + beta[2]*x[i] # Logistic regression

    #r[i] ~ dpois(p[i]*n[i])          # Poisson likelihood
    #log(p[i]) <- beta[1] + beta[2]*x[i] # Poisson regression
  }

  # Priors
  beta[1] ~ dnorm(0, 0.0001) # Second argument is precision (= 1/variance)
  beta[2] ~ dnorm(0, 0.0001)

  # If you want to allow correlatio between beta1 and beta2...
  # Set "mean" and "invR" as part of your data
  #beta[1:2] ~ dmmnorm(mean[1:2], invR[1:2,1:2])

  # Obtain Odds ratio
  for (i in 1:2) {
    OR[i] <- exp(beta[i])
  }
}

## Prepare for MCMC
## Specify data
N <- nrow(beetle)      # number of obs
r <- beetle$y
n <- beetle$n
x <- beetle$dose
## Data entry
jdata <- list(N=N,r=r,n=n,x=x)

## Parameters to be saved
para <- c("beta", "OR")

## Initial values with two chains
init1 <- list(beta=c(-40,20))
init2 <- list(beta=c(-60,30))
init <- list(init1, init2)

## Run JAGS
Bayes.logit <- jags(data=jdata, inits=init, para,
                    n.iter=50000, n.burnin=20000, n.chains=2, n.thin=1,
                    DIC=TRUE, jags.seed=12345, progress.bar="text",
                    model.file=logit.jags)

## module glm loaded

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:

```

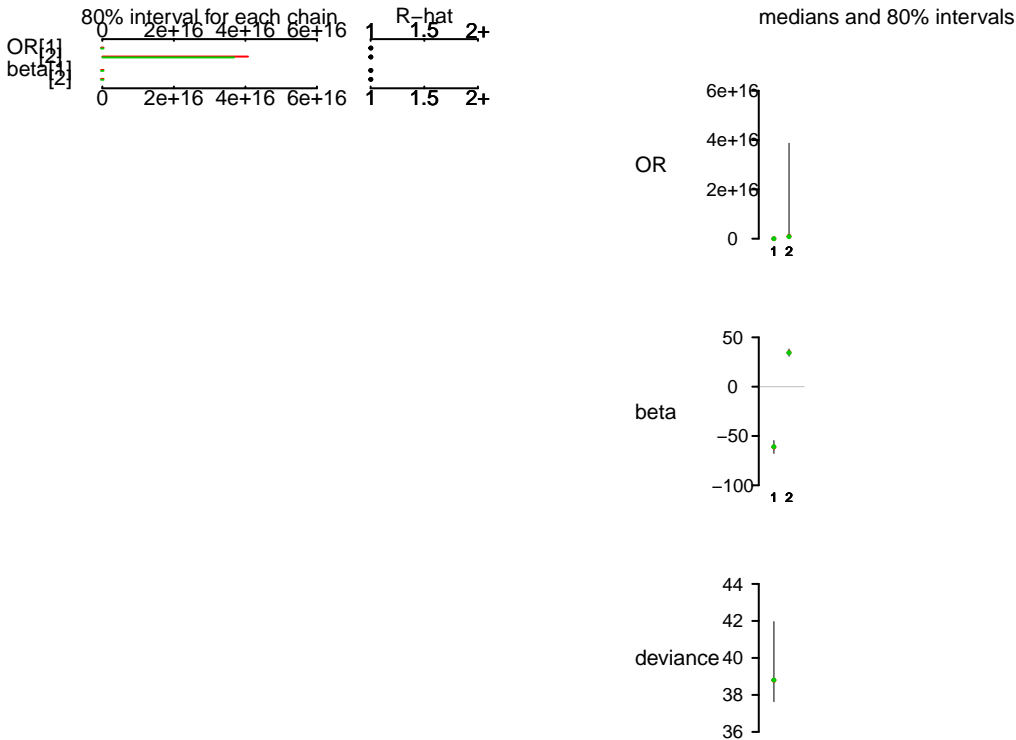
```
## Observed stochastic nodes: 8
## Unobserved stochastic nodes: 2
## Total graph size: 55
##
## Initializing model
```

```
## Check results
print(Bayes.logit)
```

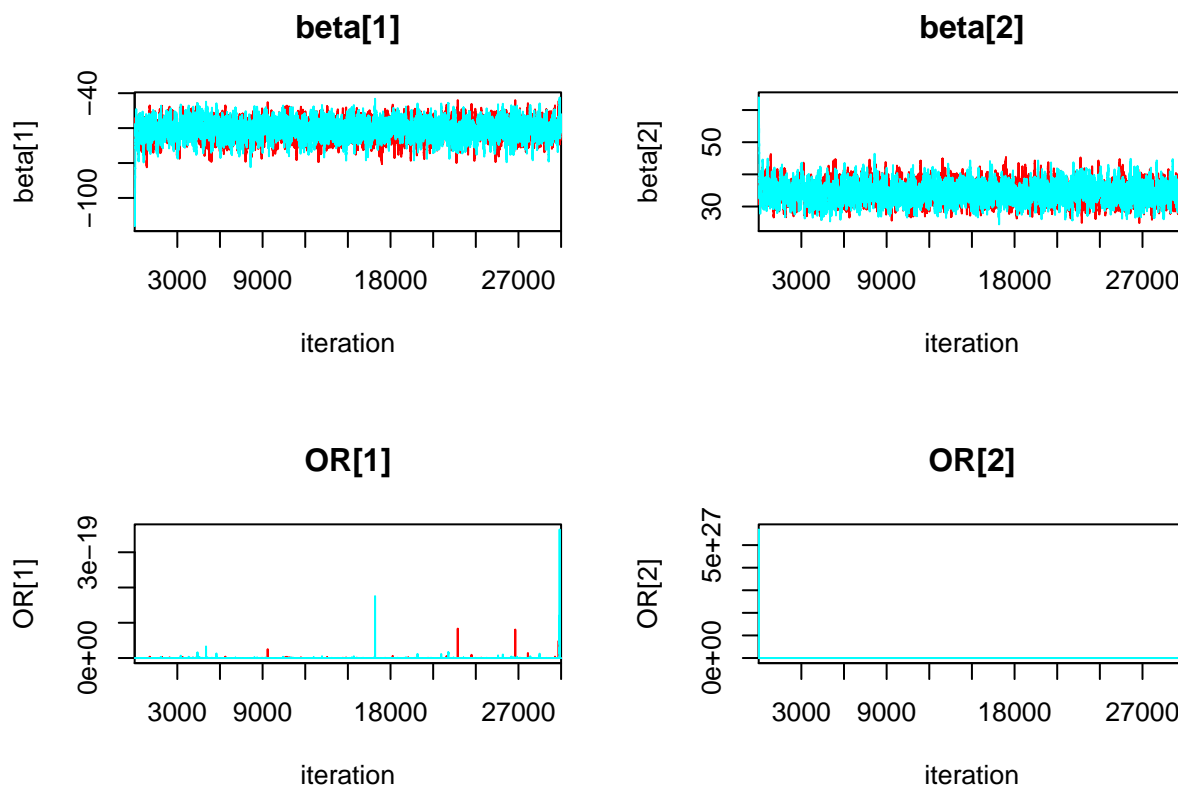
```
## Inference for Bugs model at "/var/folders/77/20b0kw_97nz5bkpgpl4j63q4ww81c2/T//RtmpbxaXyu/model4a524e590f25.txt", fit using jags, 2 chains, each with 50000 iterations (first 20000 discarded)
## n.sims = 60000 iterations saved
##          mu.vect  sd.vect      2.5%      25%      50%      75%      97.5% Rhat n.eff
## OR[1]      0.00e+00 0.00e+00  0.00e+00  0.00e+00  0.00e+00  0.00e+00  0.00e+00 1.00  2000
## OR[2]      4.05e+23 3.54e+25  4.25e+12  1.32e+14  8.98e+14  6.09e+15  3.43e+17 1.00  1900
## beta[1]    -6.11e+01 5.20e+00 -7.16e+01 -6.44e+01 -6.10e+01 -5.76e+01 -5.15e+01 1.00  2000
## beta[2]     3.45e+01 2.92e+00  2.91e+01  3.25e+01  3.44e+01  3.63e+01  4.04e+01 1.00  1600
## deviance    3.96e+01 1.05e+01  3.75e+01  3.80e+01  3.88e+01  4.02e+01  4.48e+01 1.03  3400
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 54.9 and DIC = 94.5
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```
plot(Bayes.logit)
```

rs/77/20b0kw\_97nz5bkpgpl4j63q4ww81c2/T//RtmpbxaXyu/model4a524e590f25.txt", fit using jags, 2 chains, each with 50000 iteration



```
traceplot(Bayes.logit, var=c("beta", "OR"), mfrow=c(2,2), ask=FALSE)
```



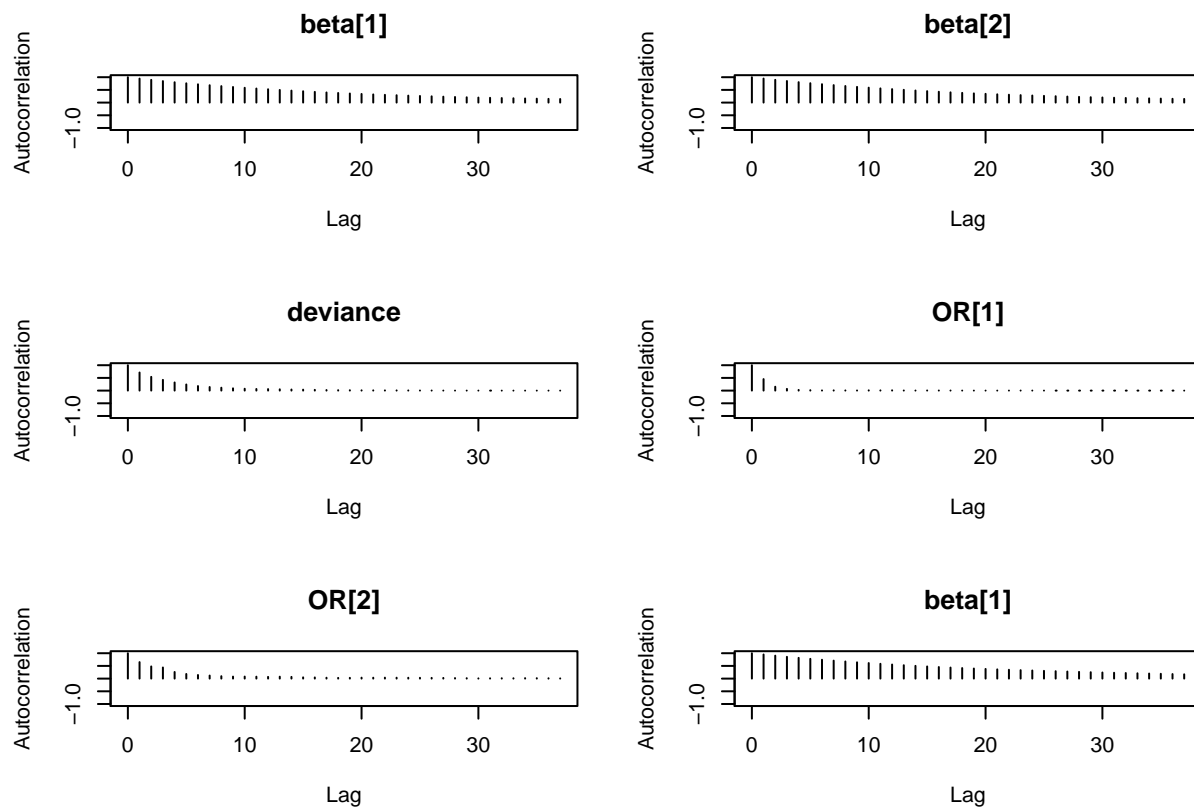
```
logit.mcmc <- as.mcmc(Bayes.logit)
summary(logit.mcmc)
```

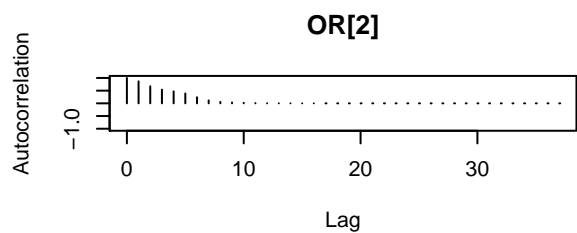
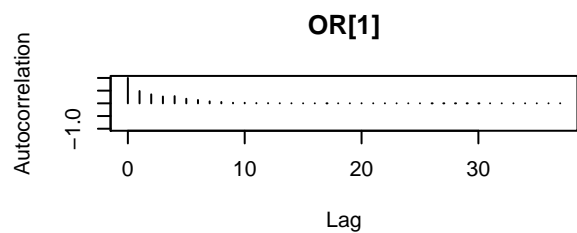
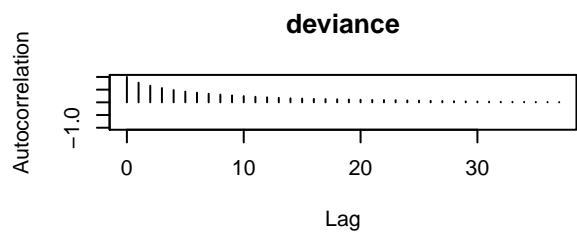
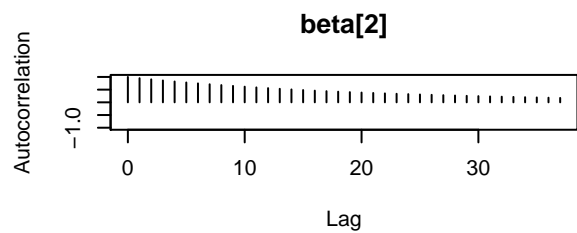
```
##
## Iterations = 20001:50000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 30000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD Naive SE Time-series SE
## beta[1]  -6.11e+01  5.20e+00  2.12e-02      1.31e-01
## beta[2]   3.45e+01  2.92e+00  1.19e-02      7.40e-02
## deviance  3.96e+01  1.05e+01  4.28e-02      1.49e-01
## OR[1]     4.46e-23  2.06e-21  8.41e-24      0.00e+00
## OR[2]     4.05e+23  3.54e+25  1.44e+23      4.00e+23
##
## 2. Quantiles for each variable:
##
##              2.5%      25%      50%      75%      97.5%
## beta[1]  -7.16e+01 -6.44e+01 -6.10e+01 -5.76e+01 -5.15e+01
## beta[2]   2.91e+01  3.25e+01  3.44e+01  3.63e+01  4.04e+01
## deviance  3.75e+01  3.80e+01  3.88e+01  4.02e+01  4.48e+01
## OR[1]     8.19e-32  1.08e-28  3.24e-27  9.64e-26  4.49e-23
## OR[2]     4.25e+12  1.32e+14  8.98e+14  6.09e+15  3.43e+17
```

```
head(logit.mcmc[[1]])    ## MCMC samples from the first chain
```

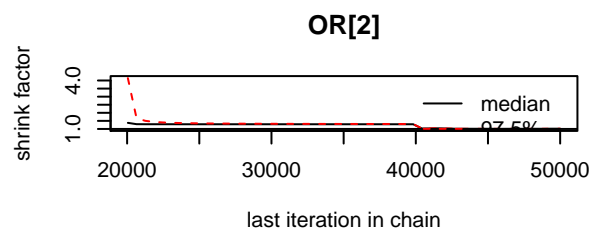
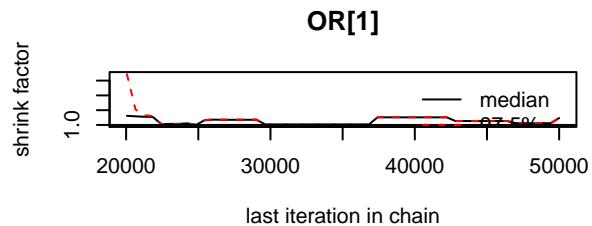
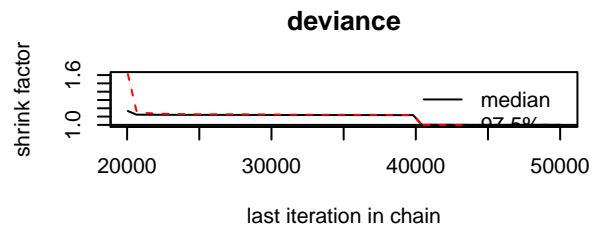
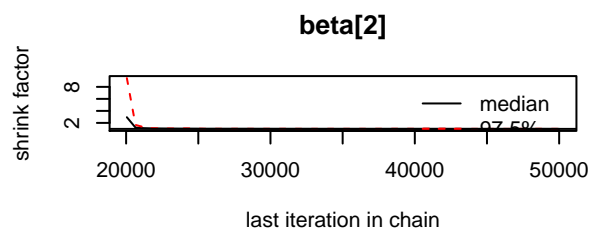
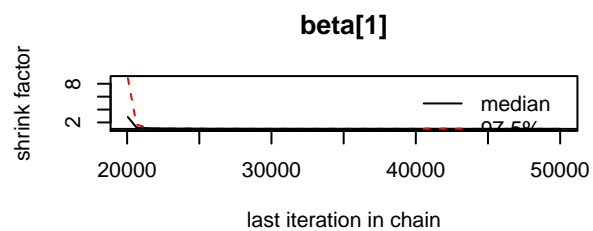
```
## Markov Chain Monte Carlo (MCMC) output:
## Start = 20001
## End = 20007
## Thinning interval = 1
##      beta[1] beta[2] deviance   OR[1]   OR[2]
## [1,]  -63.9   34.0     893 1.70e-28 5.68e+14
## [2,]  -73.1   39.5     596 1.71e-32 1.37e+17
## [3,]  -73.7   40.0     447 9.94e-33 2.39e+17
## [4,]  -78.7   42.9     366 6.84e-35 4.45e+18
## [5,]  -77.9   42.7     289 1.51e-34 3.43e+18
## [6,]  -77.1   42.4     224 3.17e-34 2.70e+18
## [7,]  -76.1   42.0     164 9.23e-34 1.80e+18
```

```
autocorr.plot(logit.mcmc)
```





```
gelman.plot(logit.mcmc)
```





## Poisson regression

```
#### Frequentist Poisson regression
fit.pois <- glm(y~dose+offset(log(n)), family=poisson(link="log"), data=beetle)
summary(fit.pois)
```

```
##
## Call:
## glm(formula = y ~ dose + offset(log(n)), family = poisson(link = "log"),
##      data = beetle)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.709  -0.918  -0.271   0.620   2.196
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -18.61      2.00   -9.30  <2e-16 ***
## dose             9.99      1.09    9.13  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 108.639  on 7  degrees of freedom
## Residual deviance:  12.255  on 6  degrees of freedom
## AIC: 57.73
##
## Number of Fisher Scoring iterations: 4
```

```
## JAGS model
pois.jags <- function(){

  for(i in 1:N) {
    r[i] ~ dpois(p[i]*n[i])          # Poisson likelihood
    log(p[i]) <- beta[1] + beta[2]*x[i] # Poisson regression
  }

  # Priors
  beta[1] ~ dnorm(0, 0.0001) # Second argument is precision (= 1/variance)
  beta[2] ~ dnorm(0, 0.0001)

  # If you want to allow correlatio between beta1 and beta2...
  # Set "mean" and "invR" as part of your data
  #beta[1:2] ~ dnmnorm(mean[1:2], invR[1:2,1:2])

  # Obtain relative rate
  for (i in 1:2) {
    RR[i] <- exp(beta[i])
  }
}

## Prepare for MCMC
## Specify data
N <- nrow(beetle)      # number of obs
```

```

r <- beetle$y
n <- beetle$n
x <- beetle$dose
## Data entry
jdata <- list(N=N,r=r,n=n,x=x)

## Parameters to be saved
para <- c("beta", "RR")

## Initial values with two chains
init1 <- list(beta=c(-20,10))
init2 <- list(beta=c(-25,5))
init <- list(init1, init2)

## Run JAGS
Bayes.pois <- jags(data=jdata, inits=init, para,
  n.iter=50000, n.burnin=20000, n.chains=2, n.thin=1,
  DIC=TRUE, jags.seed=12345, progress.bar="text",
  model.file=pois.jags)

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 8
##   Unobserved stochastic nodes: 2
##   Total graph size: 63
##
## Initializing model

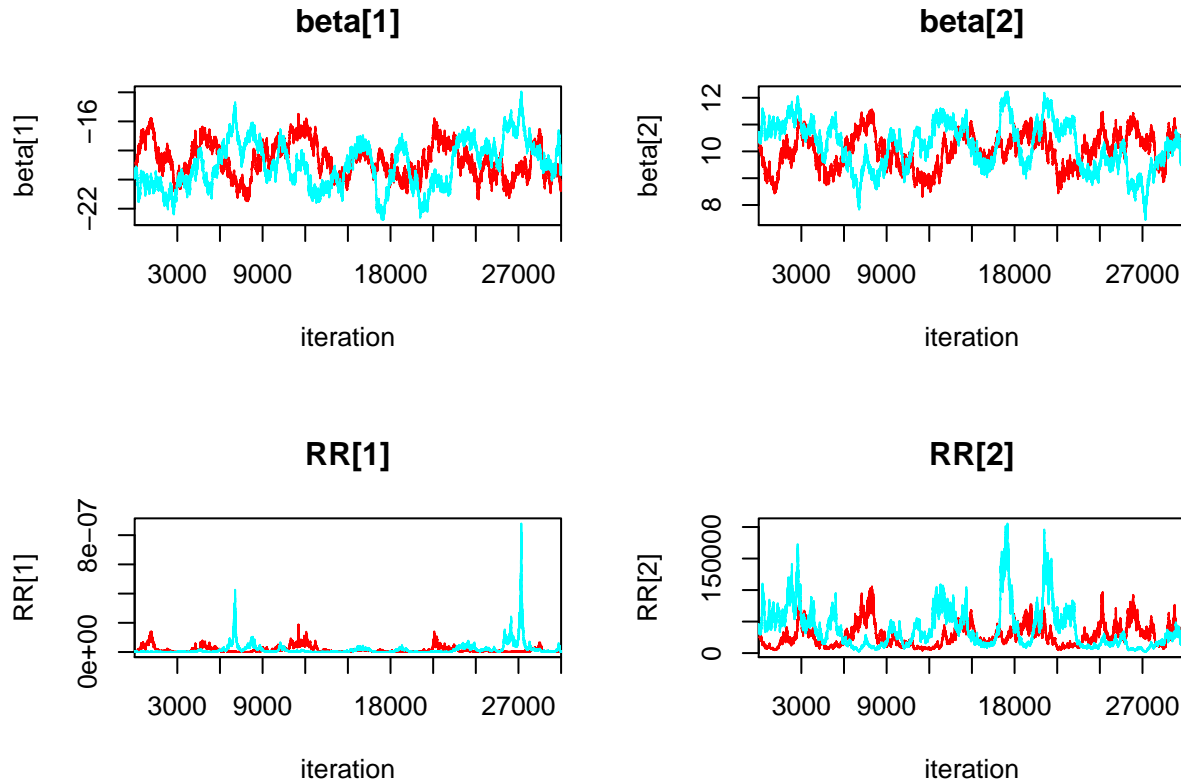
## Check results
print(Bayes.pois)

## Inference for Bugs model at "/var/folders/77/20b0kw_97nz5bkpgpl4j63q4ww81c2/T//RtmpbxaXyu/model14a525
## 2 chains, each with 50000 iterations (first 20000 discarded)
## n.sims = 60000 iterations saved

|          | mu.vect | sd.vect  | 2.5%    | 25%      | 50%     | 75%     | 97.5%    | Rhat | n.eff |
|----------|---------|----------|---------|----------|---------|---------|----------|------|-------|
| RR[1]    | 0.0     | 0.00     | 0.00    | 0.00     | 0.0     | 0.0     | 0.0      | 1.06 | 79    |
| RR[2]    | 32923.7 | 26638.15 | 6045.27 | 13931.17 | 25579.7 | 43758.2 | 104849.9 | 1.06 | 80    |
| beta[1]  | -18.8   | 1.41     | -21.48  | -19.88   | -18.9   | -17.8   | -16.3    | 1.06 | 79    |
| beta[2]  | 10.1    | 0.77     | 8.71    | 9.54     | 10.2    | 10.7    | 11.6     | 1.05 | 98    |
| deviance | 55.2    | 1.55     | 53.78   | 54.19    | 54.8    | 55.8    | 59.4     | 1.02 | 98    |

##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 1.2 and DIC = 56.4
## DIC is an estimate of expected predictive error (lower deviance is better).
traceplot(Bayes.pois, var=c("beta", "RR"), mfrow=c(2,2), ask=FALSE)

```



```
## Run JAGS with thin=10
## To improve convergence
Bayes.pois <- jags(data=jdata, inits=init, para,
  n.iter=500000, n.burnin=200000, n.chains=2, n.thin=10,
  DIC=TRUE, jags.seed=12345, progress.bar="text",
  model.file=pois.jags)
```

```
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 8
##   Unobserved stochastic nodes: 2
##   Total graph size: 63
##
## Initializing model
```

```
## Check results
print(Bayes.pois)
```

```
## Inference for Bugs model at "/var/folders/77/20b0kw_97nz5bkpgpl4j63q4ww81c2/T//RtmpbxaXyu/model14a522"
## 2 chains, each with 5e+05 iterations (first 2e+05 discarded), n.thin = 10
## n.sims = 60000 iterations saved
##      mu.vect  sd.vect   2.5%   25%   50%   75%   97.5% Rhat n.eff
## RR[1]      0.00    0.00    0.00    0.00    0.00    0.0    0.0    1  9700
## RR[2]  40506.57 54980.18 2712.57 9966.97 21231.29 46657.8 200293.8 1  9700
## beta[1]   -18.61    2.05  -22.68  -20.00  -18.56  -17.2   -14.8    1  9700
## beta[2]     9.99    1.12    7.91    9.21    9.96   10.8   12.2    1  7100
## deviance   55.78    2.00   53.78   54.33   55.20   56.6   61.0    1  7900
##
```

```
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 2.0 and DIC = 57.8
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```
traceplot(Bayes.pois, var=c("beta", "RR"), mfrow=c(2,2), ask=FALSE)
```

