STAT431_HW5

Taiga Hasegawa(taigah2)

2019/4/10

Question1

(a)

$\sigma_1^2 = \sigma_2^2 = \sigma^2$ In this case, $Y_i \mid Z, \sigma^2 \sim i.i.d. N(0, 1+\sigma^2)$ and Y is exchangeable.

(b)

$Cov(Y_1, Y_2) = E[Y_1 Y_2] = E[(Z+\epsilon_1)(Z+\epsilon_2)] = E[Z^2] = Var(Z) = 1$

Z is independent each other.

Question2

(b)

model {

for (i in 1:length(psihat)) {

```
psihat[i] ~ dnorm(psi[i],tausq[i])

psi[i] ~ dnorm(psi0,tausq0)

tausq[i] <- 1 / sigma[i]^2
```

}

psi0 ~ dnorm(0,0.001)

tausq0 ~ dgamma(0.001,0.001)

sigmasq0 <- 1/tausq0

}

(c)

```
library(rjags)
```

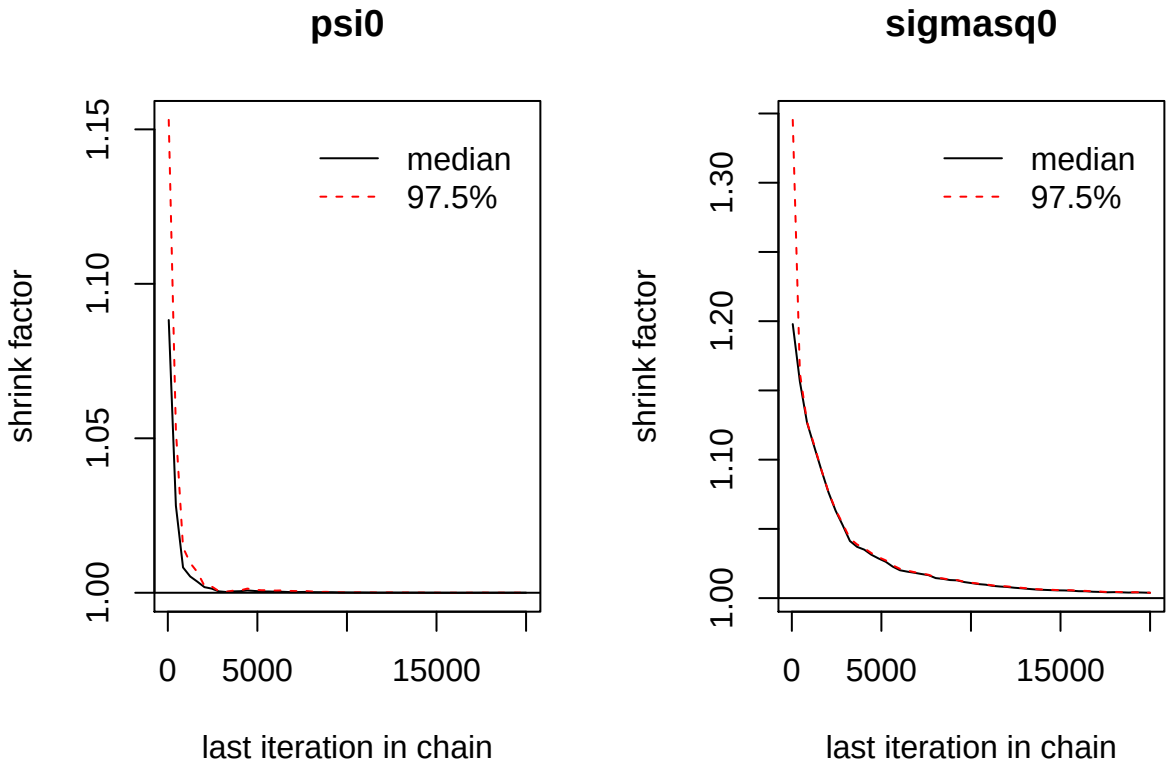
```
## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs
```

```
d <- read.table("hw5.txt", header=TRUE)
inits=list(list(psi0=0.1,tausq0=0.0001),list(psi0=5,tausq0=1),list(psi0=10,tausq0=0.01))
m=jags.model("prob2.bug",d,inits,n.chains = 3)
```

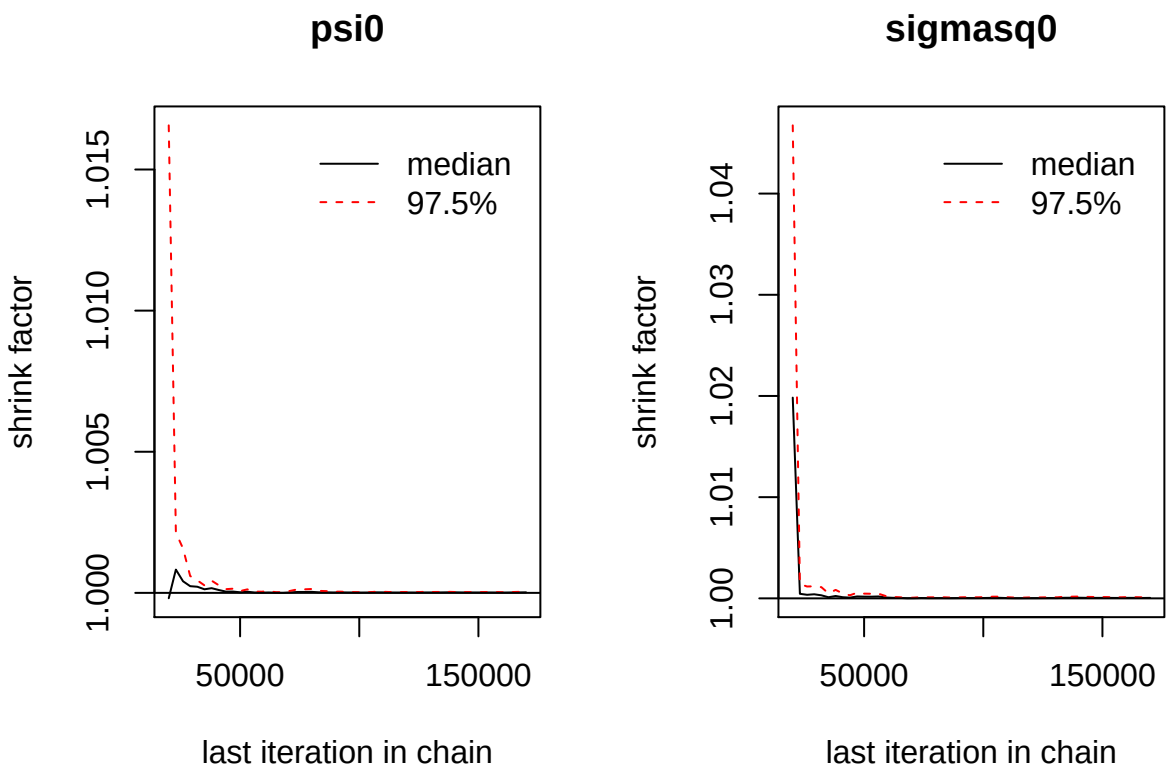
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
```

```
## Graph information:
##    Observed stochastic nodes: 12
##    Unobserved stochastic nodes: 14
##    Total graph size: 67
##
## Initializing model
```

```
x <- coda.samples(m, c("psi0","sigmasq0"), n.iter=20000)
gelman.plot(x, autoburnin=FALSE, ask=TRUE)
```




```
x <- coda.samples(m, c("psi0","sigmasq0"), n.ite=150000)
gelman.plot(x, autoburnin=FALSE, ask=TRUE)
```



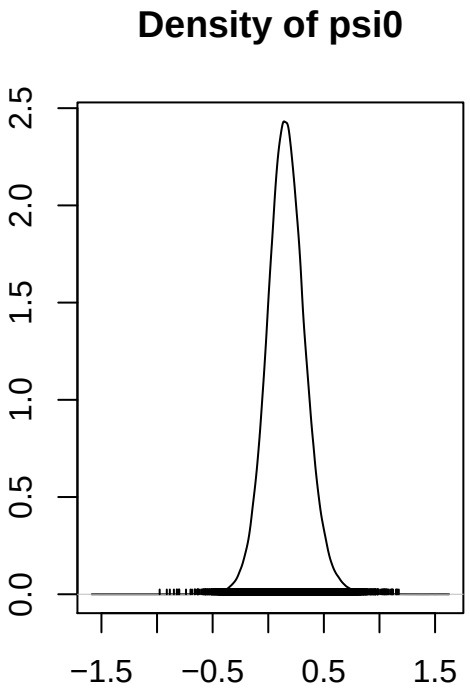
20000 iterations should be burned out because shrink factor of all variables becomes significantly below 1.05 after 20000 iterations.

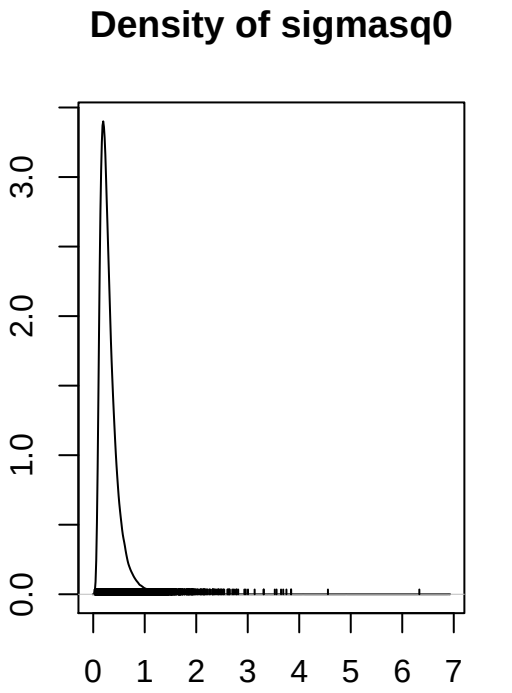
(d)

```
summary(window(x, 20001, 170000))
```

```
##
## Iterations = 20001:170000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 150000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean     SD  Naive SE Time-series SE
## psi0     0.1628 0.1757 0.0002619      0.0003196
## sigmasq0 0.3078 0.1959 0.0002921      0.0004107
##
## 2. Quantiles for each variable:
##
##               2.5%    25%    50%    75%  97.5%
## psi0     -0.17489 0.0497 0.1586 0.2719 0.5230
## sigmasq0  0.09609 0.1822 0.2592 0.3751 0.8081
```

```
plot(window(x, 20001, 170000), trace=FALSE, ask=TRUE)
```


Density of psi0
0.0
0.5
1.0
1.5
2.0
2.5
−1.5
−0.5
0.5
1.5
N = 150000 Bandwidth = 0.01301


Density of sigmasq0
0.0
1.0
2.0
3.0
0
1
2
3
4
5
6
7
N = 150000 Bandwidth = 0.0113