bayes_week5

hyunjeonglee

2020 년 11 월 2 일

hierarchical modeling

yi,j | θj ind. ~ N(θj, σ2) - 각각의 observation 은 속한 그룹의 분포로부터 관측된다.

θj iid~ N(μ, τ2) - 각 그룹의 평균은 공통의 분포로 부터 발생했다고 가정한다.

anova model 과의 차이점: -각 그룹의 평균이 공통의 분포에서 생성됐다고 가정하지 않음. - anoval model 은 effect 로 인해 평균에 차이가 생긴다고 가정.

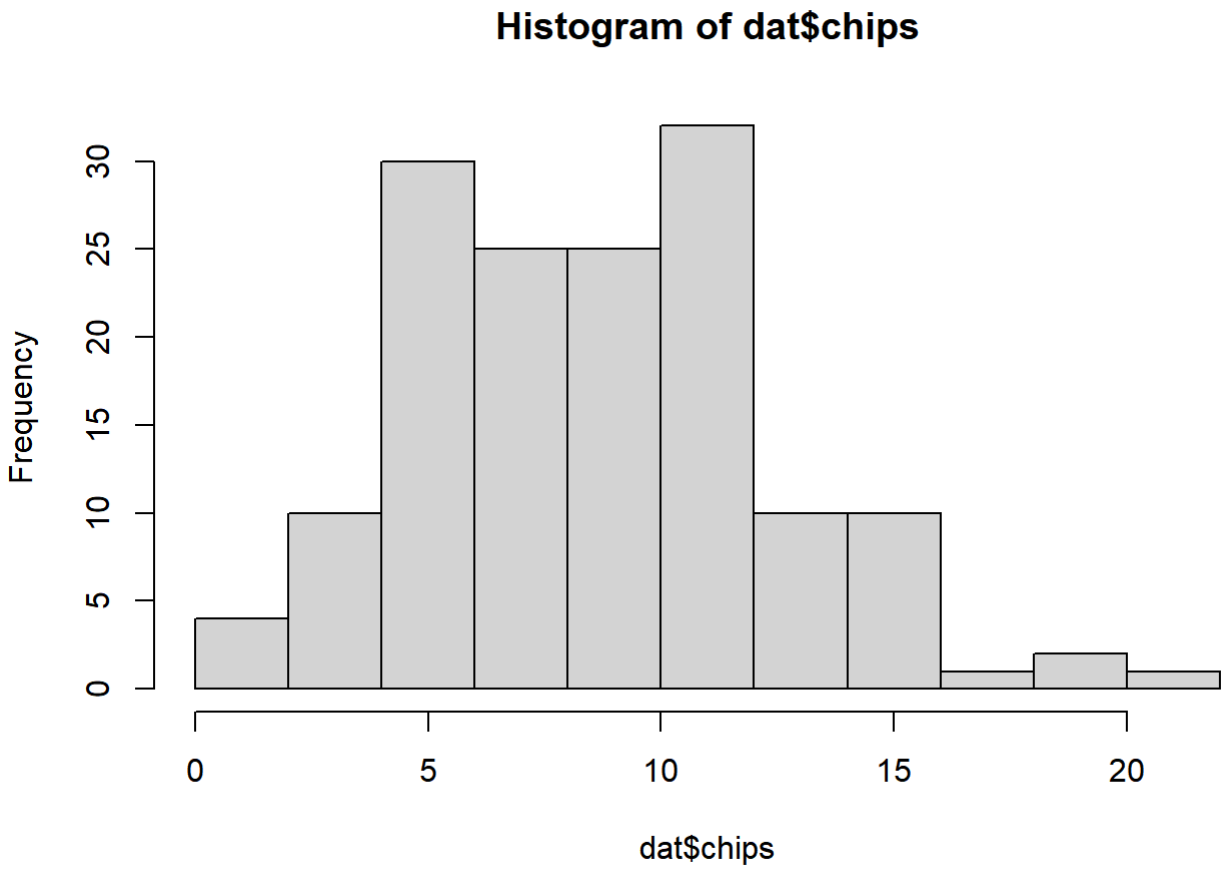
```
dat = read.table(file="cookies.txt", header=TRUE)
head(dat)
```

```
##   chips location
## 1    12        1
## 2    12        1
## 3     6        1
## 4    13        1
## 5    12        1
## 6    12        1
```

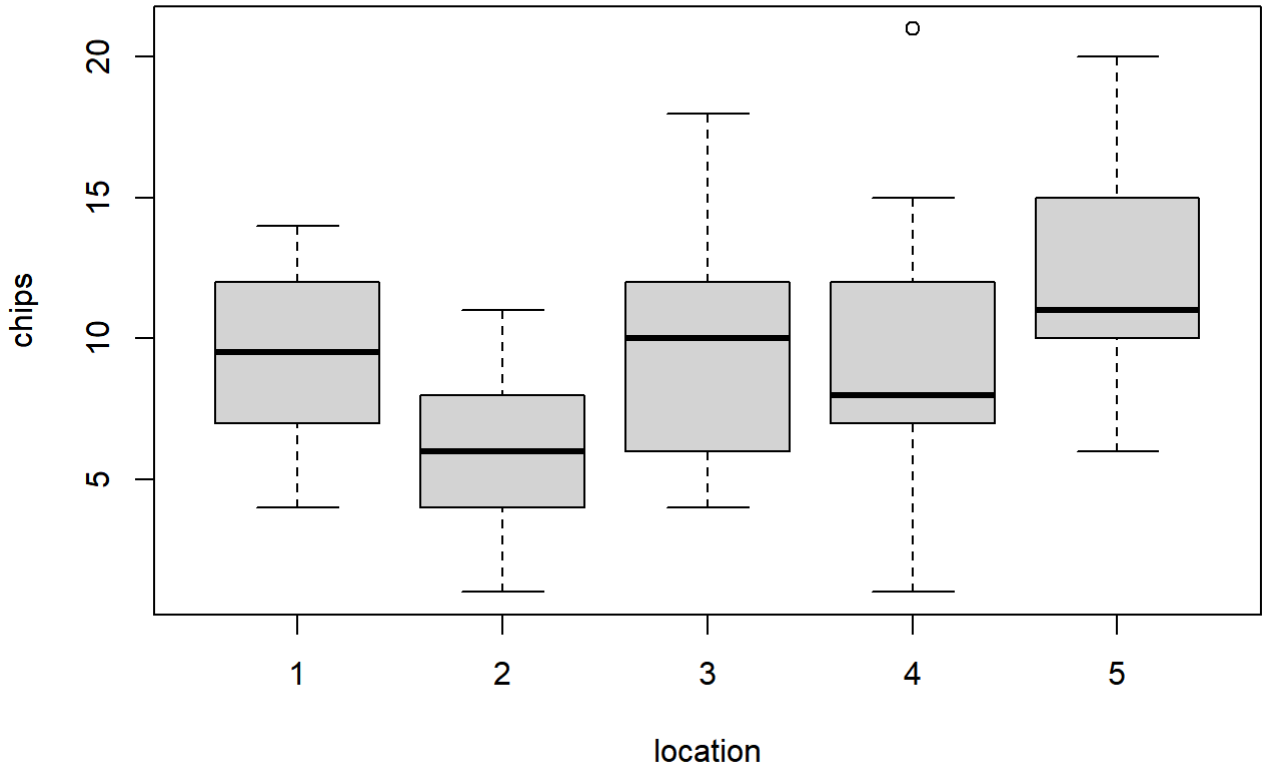
```
table(dat$location)
```

```
##
##  1  2  3  4  5
## 30 30 30 30 30
```

```
hist(dat$chips)
```




```
boxplot(chips ~ location, data=dat)
```




```
library("rjags")
```

```
## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs
```

#modeling with jags

```
mod_string = " model {
for (i in 1:length(chips)) {
  chips[i] ~ dpois(lam[location[i]])
}

for (j in 1:max(location)) {
  lam[j] ~ dgamma(alpha, beta)
}
```

```
alpha = mu^2 / sig^2
beta = mu / sig^2

mu ~ dgamma(2.0, 1.0/5.0)
sig ~ dexp(1.0)

} "

set.seed(113)

data_jags = as.list(dat)

params = c("lam", "mu", "sig")

mod = jags.model(textConnection(mod_string), data=data_jags,
n.chains=3)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 150
##    Unobserved stochastic nodes: 7
##    Total graph size: 315
##
## Initializing model
```

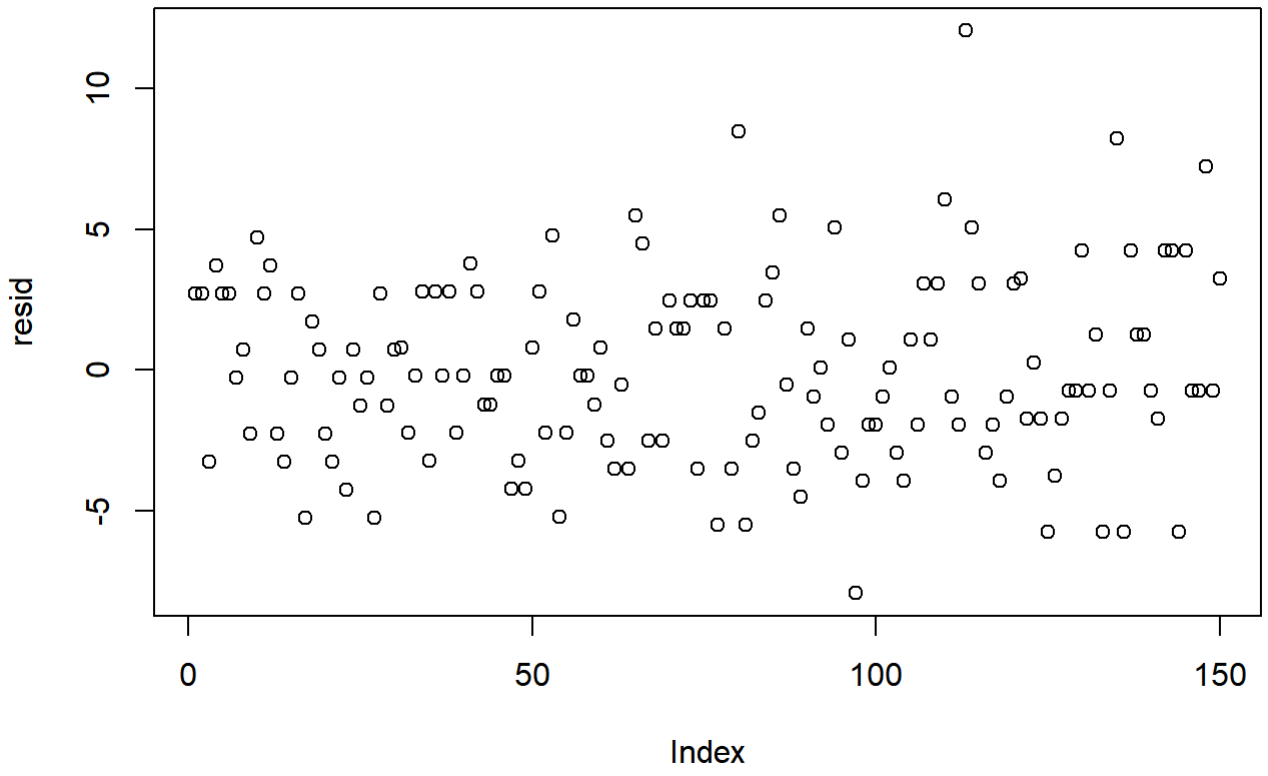
```
update(mod, 1e3)

mod_sim = coda.samples(model=mod,
                       variable.names=params,
                       n.iter=5e3)
mod_csim = as.mcmc(do.call(rbind, mod_sim))
effectiveSize(mod_sim)
```

```
##    lam[1]    lam[2]    lam[3]    lam[4]    lam[5]        mu
sig
## 15000.000  9692.978 14752.947 14180.430 12747.275  6535.484
3294.897
```

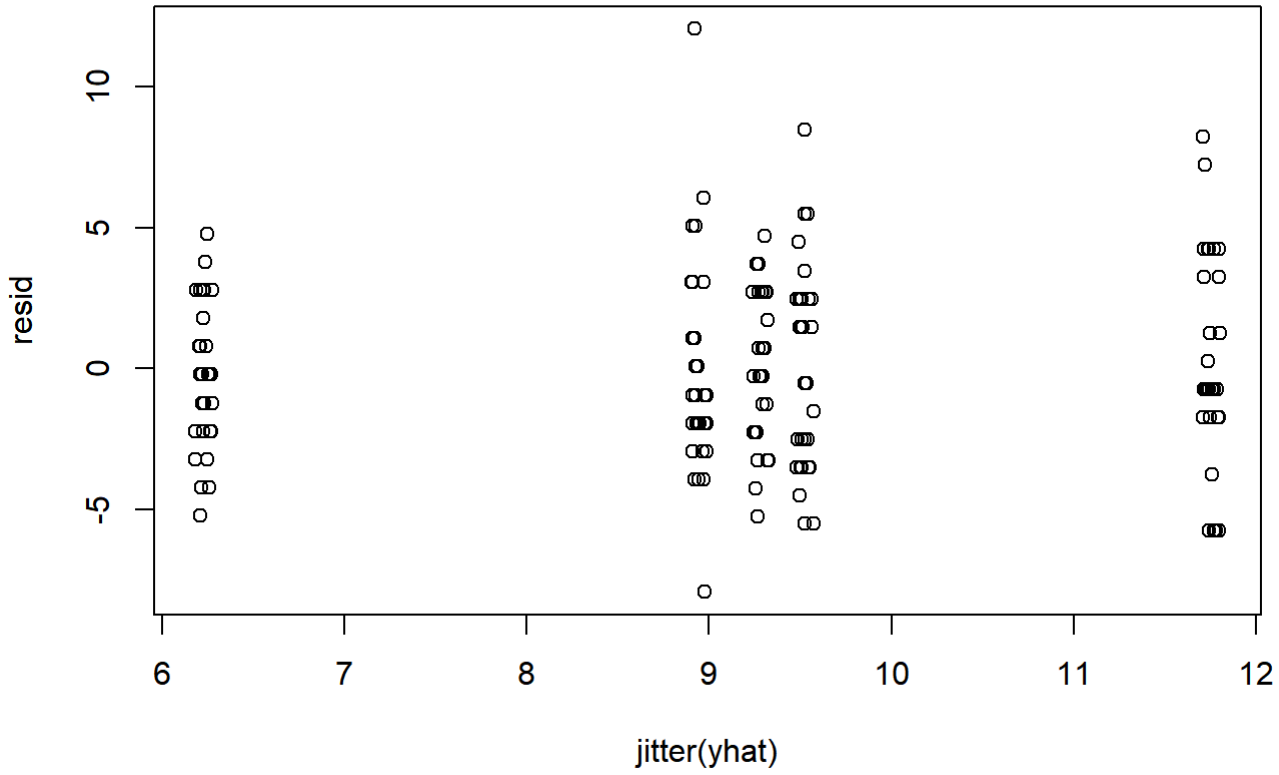
```
## observation level residuals
(pm_params = colMeans(mod_csim))
```

```
##    lam[1]    lam[2]    lam[3]    lam[4]    lam[5]        mu
sig
##  9.282398  6.227439  9.525941  8.942437 11.756637  9.128033
2.089608
```

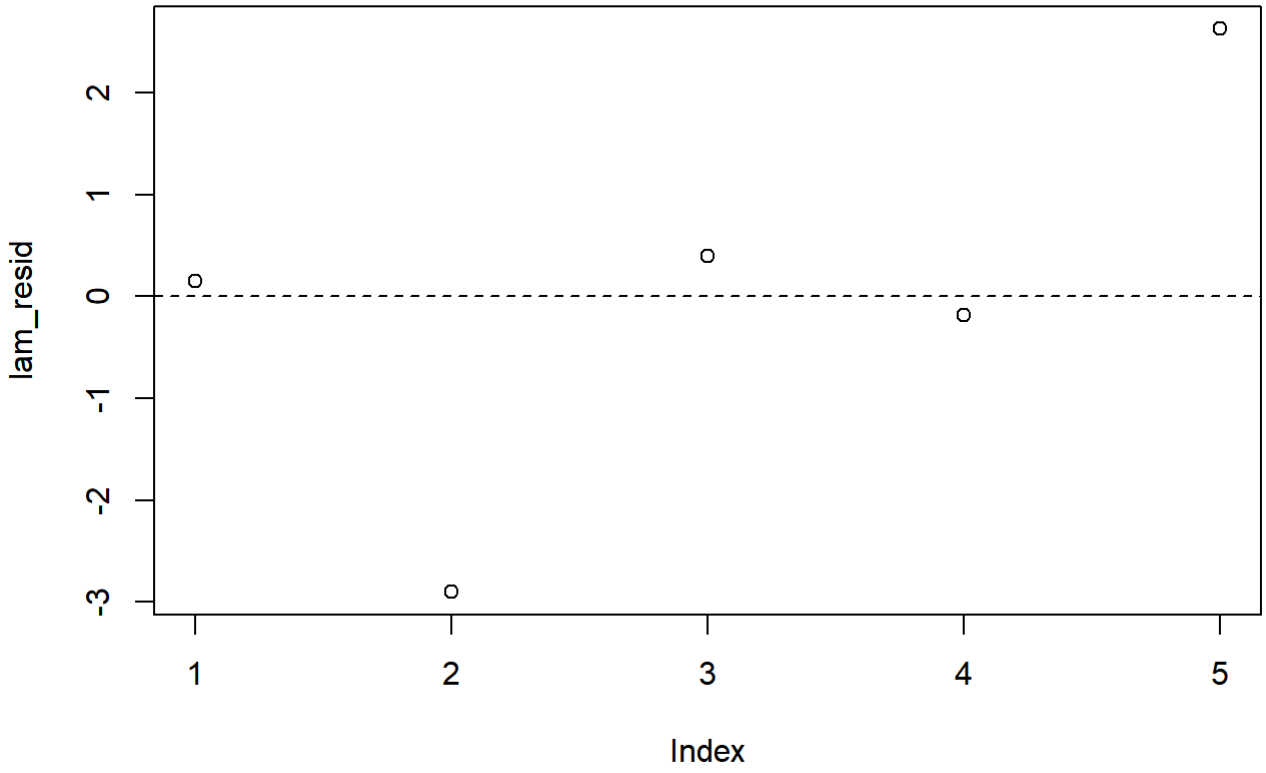
```
yhat = rep(pm_params[1:5], each=30)
resid = dat$chips - yhat
plot(resid)
```




```
plot(jitter(yhat), resid)
```




```
var(resid[yhat<7])
## [1] 6.447126
var(resid[yhat>11])
## [1] 13.72414
lam_resid = pm_params[1:5] - pm_params["mu"]
plot(lam_resid)
abline(h=0, lty=2)
```




```
growth = read.csv(file="pctgrowth.csv", header=TRUE)
head(growth)
```

```
##      y grp
## 1  1.2   1
## 2  1.4   1
## 3 -0.5   1
## 4  0.3   1
## 5  0.9   1
## 6  2.3   1
```

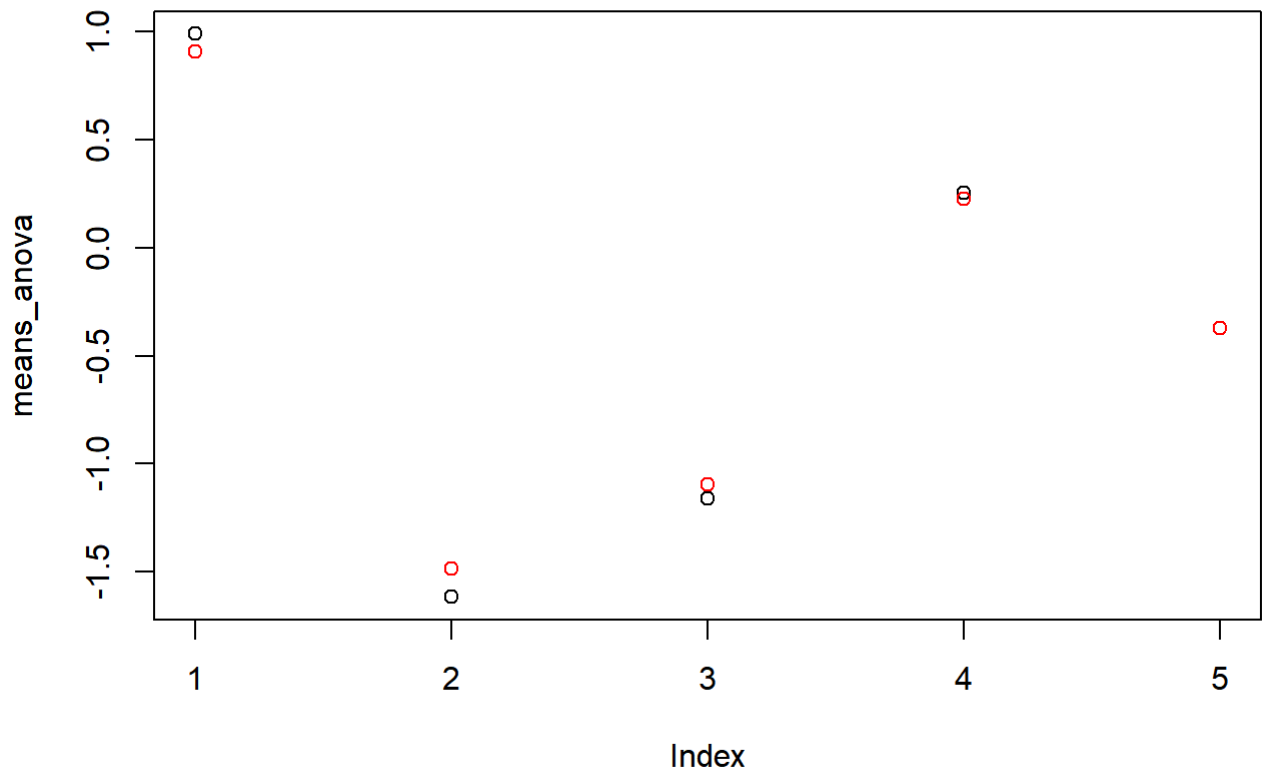
```
mod_string_G = " model {
for (i in 1:length(y)) {
  y[i] ~ dnorm(lam[grp[i]], prec1)
}

for (j in 1:max(grp)) {
```

```
  lam[j] ~ dnorm(mu, prec2)
}

mu ~ dnorm(0,1.0/1.0e6)
prec2 ~ dgamma(1/2.0, 1*3.0/2.0)
prec1 ~ dgamma(1/2.0, 2*1.0/2.0)

sig1_2 = 1.0 / prec1
sig1 = sqrt(sig1_2)
sig2_2 = 1.0 / prec2
sig2 = sqrt(sig2_2)

} "

set.seed(113)

data_jags_G = as.list(growth)

params = c("lam", "mu", "sig1" ,"sig2")

modG = jags.model(textConnection(mod_string_G),
data=data_jags_G, n.chains=3)
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 53
##    Unobserved stochastic nodes: 8
##    Total graph size: 129
##
## Initializing model
update(modG, 1e3)
```

```
mod_simG = coda.samples(model=modG,
                        variable.names=params,
                        n.iter=5e3)
mod_csimG = as.mcmc(do.call(rbind, mod_simG))
(pm_params = colMeans(mod_csimG))
```

```
##     lam[1]     lam[2]     lam[3]     lam[4]     lam[5]         mu
sig1
##  0.9066279 -1.4844417 -1.0958431  0.2248365 -0.3700138 -
0.3590470  1.0048068
##       sig2
##  1.4199282
```

```
means_theta = pm_params[1:5]
means_anova = tapply(growth$y, INDEX=growth$grp, FUN=mean)
plot(means_anova)
points(means_theta, col="red")
```



```r
library("MASS")
data("OME")


dat = subset(OME, OME != "N/A")
dat$OME = factor(dat$OME) # relabel OME
dat$ID = as.numeric(factor(dat$ID)) # relabel ID so there are no gaps in numbers (they now go from 1 to 63)
head(dat)
```

```
##   ID Age OME Loud     Noise Correct Trials
## 1  1  30 low   35   coherent       1      4
## 2  1  30 low   35 incoherent       4      5
## 3  1  30 low   40   coherent       0      3
## 4  1  30 low   40 incoherent       1      1
## 5  1  30 low   45   coherent       2      4
## 6  1  30 low   45 incoherent       2      2
```

```r
## Original reference model and covariate matrix
mod_glm = glm(Correct/Trials ~ Age + OME + Loud + Noise, data=dat, weights=Trials, family="binomial")
X = model.matrix(mod_glm)[,-1]
head(X)
```

```
##   Age OMElow Loud Noiseincoherent
## 1  30      1   35               0
## 2  30      1   35               1
## 3  30      1   40               0
## 4  30      1   40               1
## 5  30      1   45               0
## 6  30      1   45               1
```

```r
##  reconstructed model
mod_string_OME = " model {
    for (i in 1:length(y)) {
        y[i] ~ dbin(phi[i], n[i])
        logit(phi[i]) = a[ID[i]] + b[1]*Age[i] + b[2]*OMElow[i] + b[3]*Loud[i] + b[4]*Noiseincoherent[i]
```

```
    }

  for (j in 1:max(ID)) {
    a[j] ~ dnorm(a0, prec_a)
  }

  a0 ~ dnorm(0.0, 1.0/1.0e6)
  prec_a ~ dgamma(1/2.0, 1*10.0/2.0)
  tau = sqrt( 1.0 / prec_a )

    for (k in 1:4) {
       b[k] ~ dnorm(0.0, 1.0/4.0^2)
    }

} "

data_jags = as.list(as.data.frame(X))
data_jags$y = dat$Correct
data_jags$n = dat$Trials
data_jags$ID = dat$ID
params = c("a0", "a", "b", "tau")

modO = jags.model(textConnection(mod_string_OME),
data=data_jags, n.chains=3)
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 712
##    Unobserved stochastic nodes: 69
##    Total graph size: 6502
##
## Initializing model
```

```
update(modO, 1e3) # burn-in


mod_simO = coda.samples(model=modO,
                        variable.names=params,
                        n.iter=5e3)


mod_csimO = as.mcmc(do.call(rbind, mod_simO)) # combine
multiple chains
dic.samples(modO, n.iter=1e3)
## Mean deviance:  1237
## penalty 45.08
## Penalized deviance: 1282
mod_string_origin = " model {
    for (i in 1:length(y)) {
        y[i] ~ dbin(phi[i], n[i])
        logit(phi[i]) = b0 + b[1]*Age[i] + b[2]*OMElow[i] +
b[3]*Loud[i] + b[4]*Noiseincoherent[i]
    }


    b0 ~ dnorm(0.0, 1.0/5.0^2)
    for (j in 1:4) {
        b[j] ~ dnorm(0.0, 1.0/4.0^2)
    }


} "
params = c("b0", "b")


modorigin = jags.model(textConnection(mod_string_origin),
data=data_jags, n.chains=3)
## Warning in jags.model(textConnection(mod_string_origin),
data = data_jags, :
## Unused variable "ID" in data
## Compiling model graph
##    Resolving undeclared variables
```

```
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 712
##    Unobserved stochastic nodes: 5
##    Total graph size: 4377
##
## Initializing model
dic.samples(modorigin, n.iter=1e3)
## Mean deviance:  1258
## penalty 5.561
## Penalized deviance: 1263
```