Working with JAGS and R

Andrei Zhirnov
A.Zhirnov@exeter.ac.uk

May 12, 2020

Agenda

- Writing models in JAGS
- Working with JAGS from R
- Example: A logistic regression
- Example: A multilevel logistic regression
- Example: Estimating policy positions from roll-call votes

JAGS

JAGS = Just Another Gibbs Sampler

- created in 2007 by Martyn Plummer
- uses Gibbs sampler
- works with different platforms
- needs an appropriate package to work with R (`rjags` or `R2jags`) or Python (`pyjags`)

Installing JAGS

Installation instructions for different platforms:
`http://mcmc-jags.sourceforge.net/`
Windows and Mac installers:
`https://sourceforge.net/projects/mcmc-jags/files/JAGS/4.x/`

Workflow

1. write down the model for JAGS
2. prepare the data
3. send requests to JAGS (e.g., with `rjags` package)
    1. compilation + initialization + adaptation
    2. a run with burn-in iterations
    3. a run with iterations to be used for analysis
4. check if the MCMC converged – if not, run more iterations or revise model/data
5. study the posterior distribution

Useful R packages

`rjags` – for sending requests to JAGS
`mcmcplots` – for visual convergence diagnostics
`coda` – for diagnostics and summary of MCMC output

```
install.packages(c("rjags","coda","mcmcplots"))
```

Writing down your model for JAGS

2 blocks:

- `model { ... }`
- `data { ... }`

`model` block

Draw a tree that connects the root (dependent variable) to other variables and parameters.

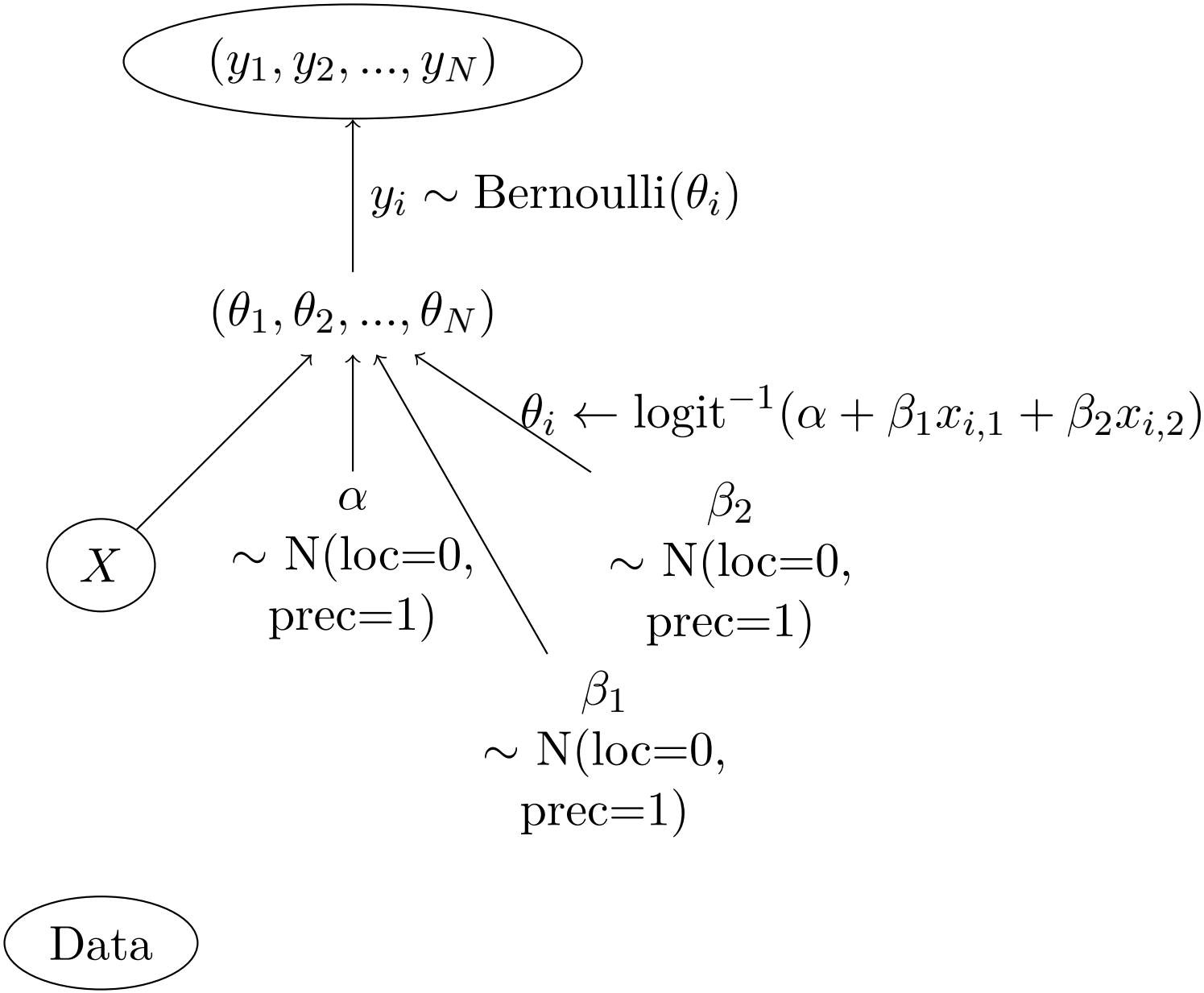
Types of relations between a node and the linked branches:

- deterministic
  `A <- B` means "A is defined as equal to B"
- stochastic
  `A ~ B` means "A is drawn from distribution B"

Anything on the right hand side of a statement has to be

- defined in the model (be on the left-hand side), or
- included in the data

Logit


$(y_1, y_2, ..., y_N)$
$y_i \sim \text{Bernoulli}(\theta_i)$
$(\theta_1, \theta_2, ..., \theta_N)$
$\theta_i \leftarrow \text{logit}^{-1}(\alpha + \beta_1 x_{i,1} + \beta_2 x_{i,2})$
X
α
$\sim$ N(loc=0, prec=1)
β_2
$\sim$ N(loc=0, prec=1)
β_1
$\sim$ N(loc=0, prec=1)
Data

Logit

```
model{
  for (i in 1:[ number of observations ]) {
    y[i] ~ dbern(theta[i])
    }
  for (i in 1:[ number of observations ]) {
    logit(theta[i]) <- alpha + inprod(x[i,],beta)
    }
  alpha ~ dnorm(0,1)
  for (g in 1:[ number of indep variables ]) {
    beta[g] ~ dnorm(0,1)
    }
   }
```

Logit

```
data {
  N <- length(y)
  G <- dim(x)
  }
model{
  for (i in 1:N) {
    y[i] ~ dbern(theta[i])
    logit(theta[i]) <- alpha + inprod(x[i,],beta)
    }
alpha ~ dnorm(0,1)
for (g in 1:G[2]) {
    beta[g] ~ dnorm(0,1)
    }
  }
```

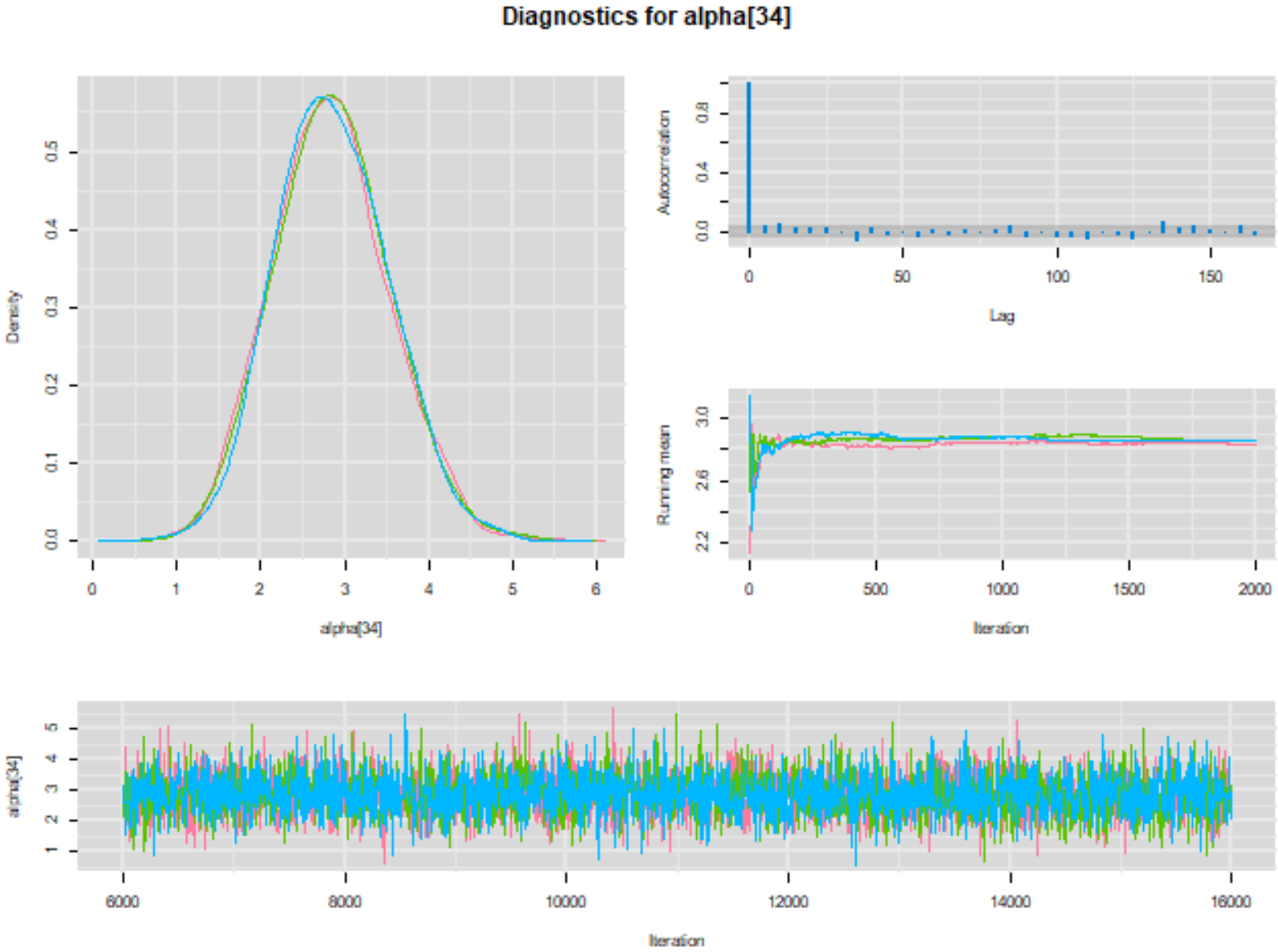
Check convergence

Check if the MCMC chains approximately converged!

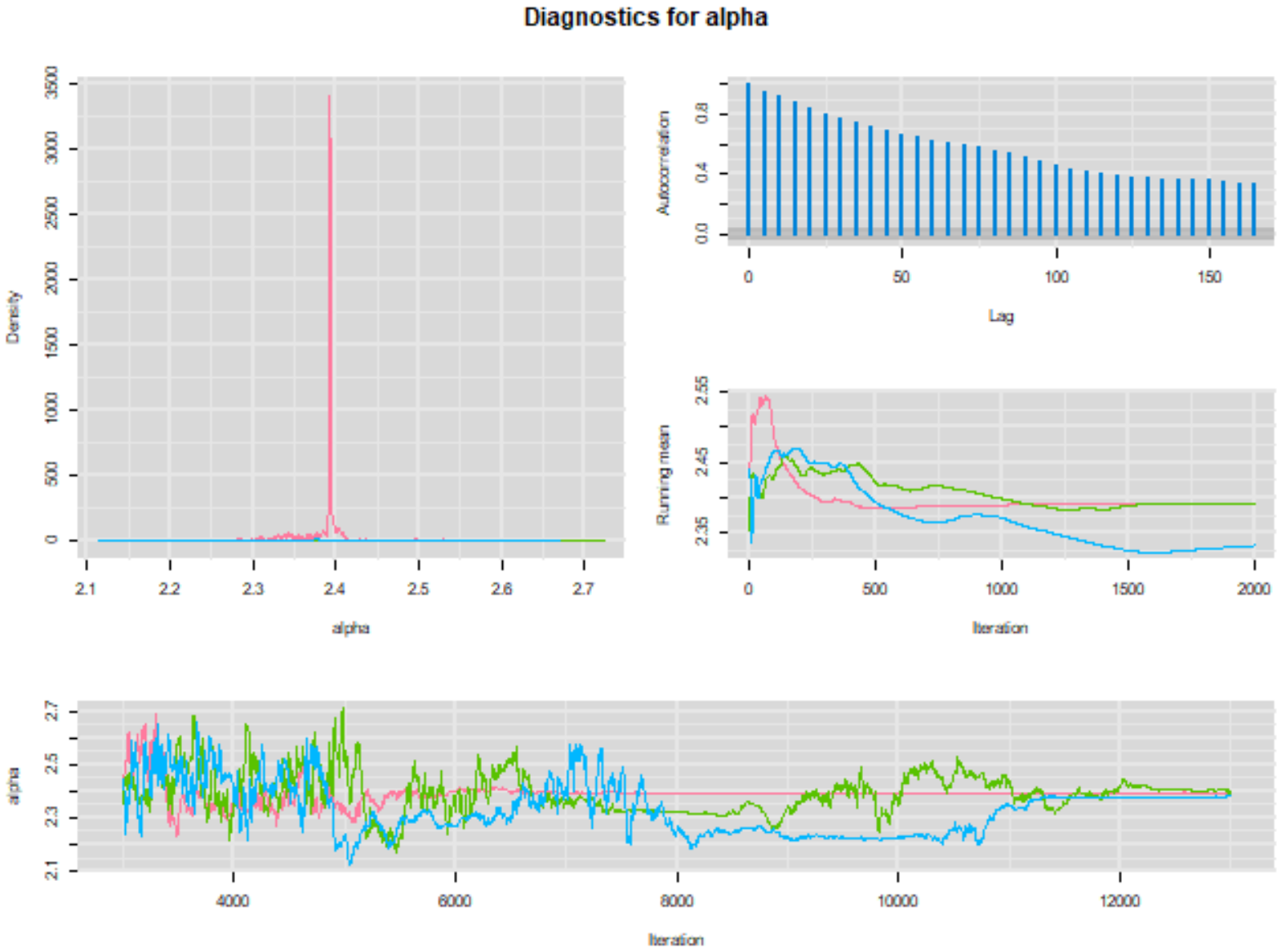
- they are well-mixed
- they are stationary

If the MCMC chains did not converge you cannot interpret the estimates!

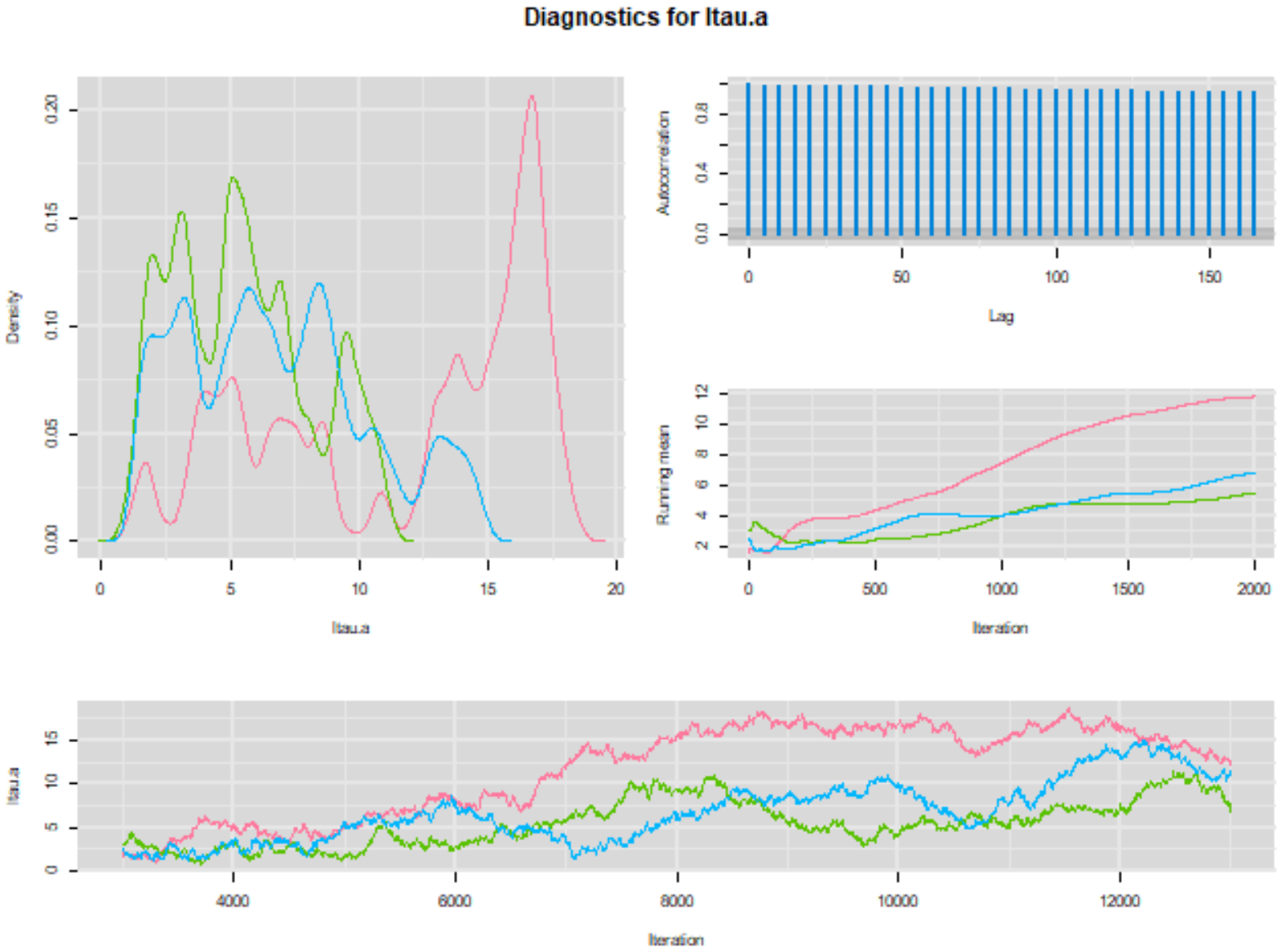
MCMC that approximately converged



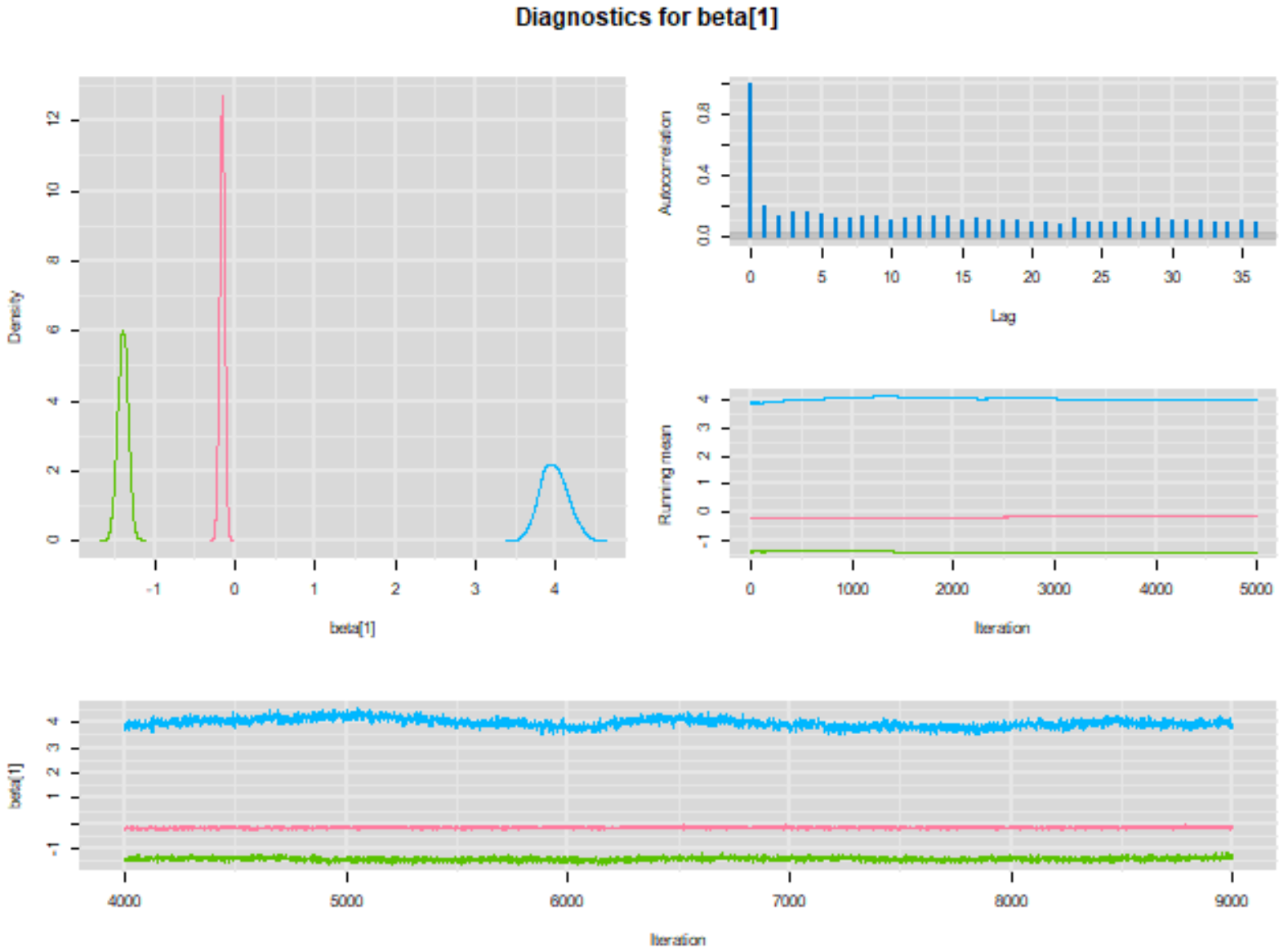
MCMC that did not converge



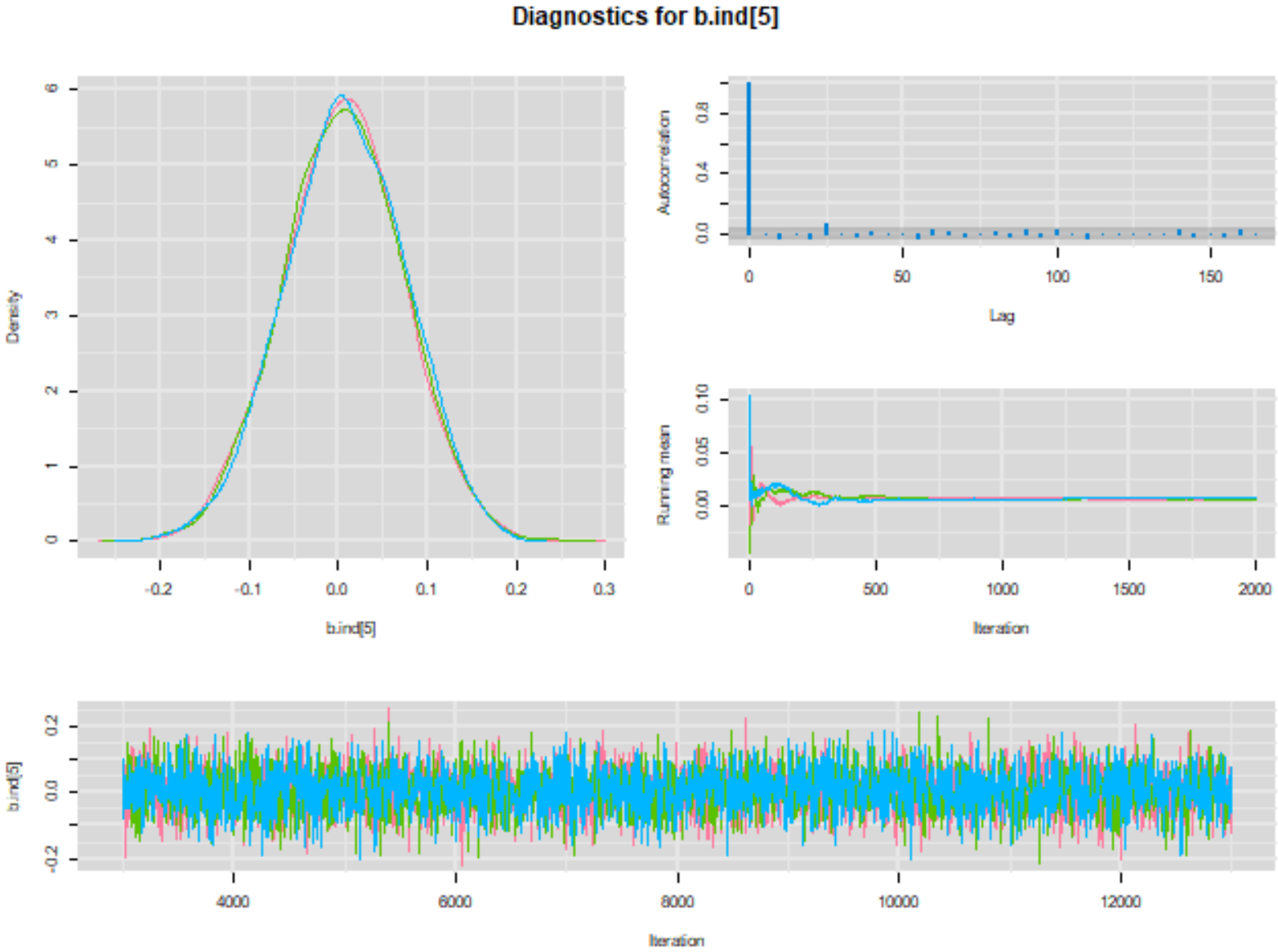
MCMC that did not converge



MCMC that did not converge



MCMC that approximately converged



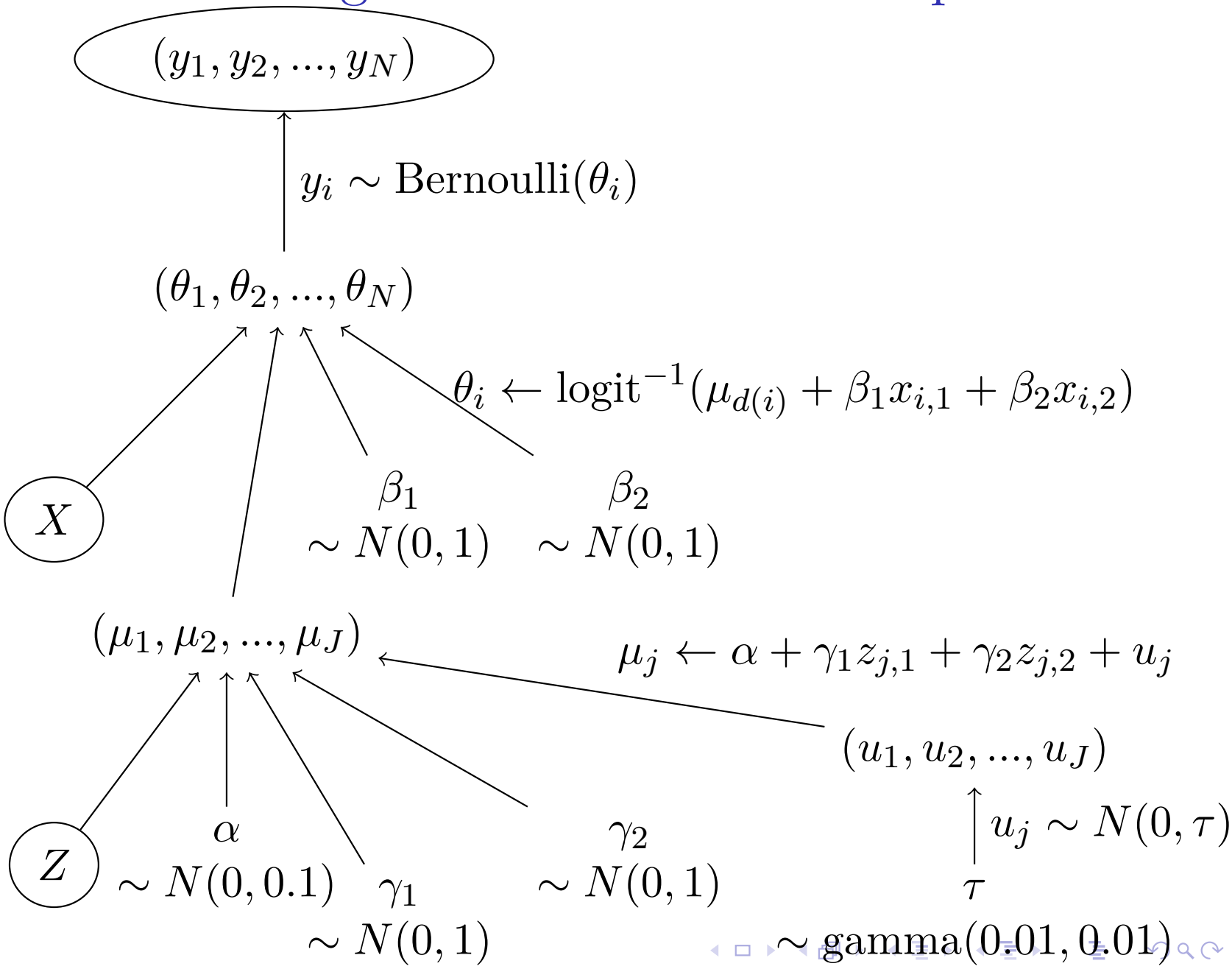
Gelman-Rubin convergence diagnostic

$\hat{R}$ - potential scale reduction factor

- is good if about 1 (<1.1 in most cases)
- requires at least 2 chains
- tests both for mixing and stationarity

*measure whether the combined variance across and within chains exceeds the variance within chains)

Logit with random intercepts



Logit with random intercepts

`data` block

```
data {
J <- dim(x.dist)
G <- dim(x.ind)
}
```

J[1] - number of districts
J[2] - number of columns in the matrix with district-level data
G[1] - number of respondents
G[2] - number of columns in the matrix with respondent-level data

Logit with random intercepts

`model` block: individual-level pieces

```
for (i in 1:G[1]) {
     y[i] ~ dbern(theta[i])
     logit(theta[i]) <- mu[dcode[i]] +
                              inprod(x.ind[i,],b.ind)
     }
for (g in 1:G[2]) {
     b.ind[g] ~ dnorm(0,0.1)
     }
```

Logit with random intercepts

`model` block: district-level pieces

```
for (j in 1:J[1]){
    mu[j] <- alpha + inprod(x.dist[j,],b.dist) + u[j]
    u[j] ~ dnorm(aveu,tau)
    }
for (f in 1:J[2]) {
    b.dist[f] ~ dnorm(0,0.1)
    }
alpha ~ dnorm(0,0.1)
aveu ~ dnorm(0,0.1)
tau ~ dgamma(0.001, 0.001)
```

A measurement problem

Goal: measure senators' policy positions from roll-call data
Data: voting records for N senators on K motions

$$y_{ik} \in \{\text{yes}, \text{no}\}$$

Parameters:
policy positions of senators: $(\theta_1, \theta_2, ..., \theta_N)$
Policy state if the vote succeeds: $(\beta_1, \beta_2, .., , \beta_K)$
The status quo policy state: $(\beta_1^{SQ}, \beta_2^{SQ}, .., , \beta_K^{SQ})$

A measurement problem

Model:
(along the lines of the original NOMINATE procedure)
Senator i votes "yes" if the Euclidean distance between θ_i and β_k is smaller than the distance between θ_i and β_k^{SQ}

$$p(y_{ik} = \text{yes}) = \text{logit}^{-1}((\theta_i - \beta_k^{SQ})^2 - (\theta_i - \beta_k)^2)$$

Measuring policy positions

Starting point:

$$y_{ik} \sim \text{Bernoulli}(\hat{y}_{ik})$$
$$\hat{y}_{ik} = \text{logit}^{-1}((\theta_i - \beta_k^{SQ})^2 - (\theta_i - \beta_k)^2)$$
$$\theta_i \sim ...$$
$$\beta_k^{SQ} \sim ...$$
$$\beta_k \sim ...$$

To remove the correlated squared terms, we can use

$$\hat{y} = \text{logit}^{-1}((\beta_k - \beta_k^{SQ})(2\theta_i - \beta_k - \beta_k^{SQ}))$$

Measuring policy positions

To minimize calculations in the $N \times K$ loop, we can define

$$d_k = \beta_k - \beta_k^{SQ}$$
$$s_k = \beta_k + \beta_k^{SQ}$$

so that

$$\hat{y}_{ik} = \text{logit}^{-1}(2d_k\theta_i - d_k s_k)$$

Measuring policy positions

To fix the axis, set the prior on θ_i to

$$\theta_i \sim N(-1, 1) \text{ if } i \text{ is a Democrat}$$
$$\theta_i \sim N(1, 1) \text{ if } i \text{ is a Republican}$$

Measuring policy positions

`model` block:

```
for (i in 1:N){
    y[i] ~ dbern(y.hat[i])
    logit(y.hat[i]) <- 2*diff[cid[i]]*theta[rid[i]]
                                - diffsq[cid[i]]
    }
for (j in 1:K) {
    diffsq[k] <- sum[k]*diff[k]
    sum[k] ~ dnorm(0,0.5)
    diff[k] ~ dnorm(0,0.5)
    }
for (k in 1:J) {
    theta[j] ~ dnorm(ideol[j],1)
    }
```

Keep in touch with Exeter's Q-Step Centre...

For news and updates on our future events and activities, follow us on:

Twitter @ExeterQStep

Facebook @qstepexeter

To join the mailing list, email qstep@exeter.ac.uk