Bayesian Estimation of Multilevel Hierarchical Linear & Logistic Regression Models

Edps 590BAY

Carolyn J. Anderson

Department of Educational Psychology
ILLINOIS

Fall 2022

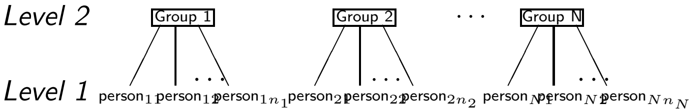
Overview

- Hierarchical data
- Simple example: anorexia data
- Bayesian Estimation
- Anorexia data
- More complex example: 20 schools of NELS
- Logistic Regression: GSS 5 vocabulary items

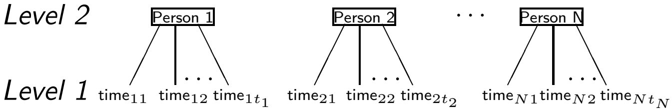
There is a bit in Kruschke (see chapter on hierarchial models) and a nice example in Gelman (look for schools example).

Examples of Hierarchies or Clustering

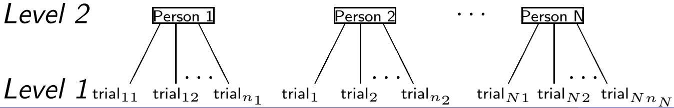
(a) Individuals within groups



(b) Longitudinal



(c) Repeated Measures



More Examples of Hierarchies

peer groups	schools	person	companies
kids	students	test items	employees

neighborhoods	schools	clinics
families	classes	doctors
children	students	patients

The Problem of Clustered Data

- Observations within groups are more similar than observations from other groups.
- Mathematically, let
  - y_{ij} be a response for person i within group j.
  - $y_{i'j}$ be a response for person i' within group j.
  - $y_{kj'}$ be a response for person k within group k.
- Typically, $r(y_{ij}, y_{i'j}) > r(y_{ij}, y_{kj'})$
- Furthermore, $r(y_{ij}, y_{i'j'}) \neq 0$
- We have violated the independence assumption needed for nearly all classical statistical methods
- The standard errors of means or regression coefficients are too small.
- This leads to inflated Type I errors.

Advantages of Taking in Account Clustering

- Takes care of dependencies in data and gives correct standard errors, confidence intervals, and significance tests.
- Statistically efficient estimates of regression coefficients.
- With clustered/multilevel/hierarchially structured data, can use covariates measured at any of the levels of the hierarchy.
- Model all levels simultaneously.
- Study contextual effects.
- Theories can be rich.

Same Model, Different Names

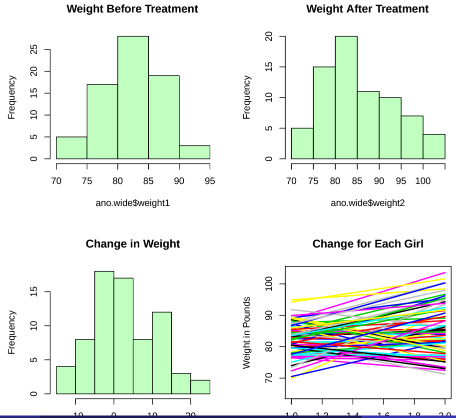
- Hierarchical Linear Models
- Multilevel Analysis using Linear Mixed Models
- Variance Components Analysis
- Random coefficients Models
- Growth curve analysis

All are special cases of Generalized Linear Mixed Models (GLMMs)

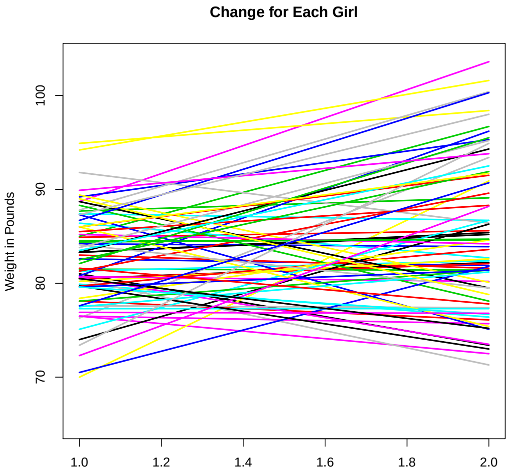
They can be estimated using Bayesian methods.

Graphically

Let's re-consider the Anorexia example but add a more multilevel aspect to this example:



Different intercepts and slopes



Different intercepts and slopes

- With only 2 time points, we cannot even estimate a linear model for each girl because 2 points define a line.
- We assume a random distribution for both the intercept and slope.
- We use all the girl's data to find this intercept and slope for the "average" girl
- In particular, for girl j and time i

$$y_{ij} = \beta_{0j} + \beta_{1j}\text{Time}_{ij} + \epsilon_{ij}$$

where the models for regression coefficients are

$$\begin{pmatrix} \beta_{0j} \\ \beta_{1j} \\ \epsilon_{ij} \end{pmatrix} \sim MVN\left(\begin{pmatrix} \gamma_{00} \\ \gamma_{10} \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{10} & 0 \\ \tau_{10} & \tau_{22} & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix}\right) \; i.i.d.$$

Usual Presentation of the Model

- Level 1

$$y_{ij} = \beta_{0j} + \beta_{1j}\text{Time}_{ij} + \epsilon_{ij}$$

where $\epsilon_{ij} \sim N(0, \sigma^2)$ $i.i.d.$
Level 2:

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + U_{0j} \\ \beta_{1j} &= \gamma_{10} + U_{1j} \end{aligned}$$

where

$$\begin{pmatrix} U_{0j} \\ U_{1j} \end{pmatrix} \sim MVN\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{22} \end{pmatrix}\right) \; i.i.d.$$

- Linear Mixed Model:

$$y_{ij} = \underbrace{\gamma_{00} + \gamma_{10}}_{fixed} + \underbrace{U_{0j} + U_{1j} + \sigma^2}_{random}$$

Bayesian Estimation of Simple: Model 0

Random effects ANOVA (null or empty HLM)

$$\text{weight}_{ij} = \beta_{0j} + \epsilon_{ij} = \gamma_{00} + U_{0j} + \epsilon_{ij}$$

where i is time and j is girl.

```
dataList ← list(y = ano$weight,
                sdY = sd(ano$weight),
                n = length(ano$weight),
                ng= length(ano$weight)/2,
                girl= ano$girl
           )
```

Model 0

```
model0 <- ‘‘model {
              for (i in 1:n) {
                  y[i] ∼ dnorm(mu[i],precision)
                  mu[i] ← beta0j[girl[i]]
              }
             for (j in 1:ng) {
                  beta0j[j] ∼ dnorm(g0,ptau)
              }
              g0 ∼ dnorm(0,1/(100*sdYˆ2))
              tau ∼ dunif(0.0001,200)
              ptau ← 1/tauˆ2
              sigma ∼ dunif(0.0001,2000)
              precision ← 1/sigmaˆ2
              icc ← tauˆ2 /(sigmaˆ2 + tauˆ2)
              }...
```

ICC for Random Intercept Model(s)

A measure of within class dependency (correlation), which is also interpretable as the proportion of variance due to random intercept is the Intra-class correlation

$$\text{ICC} = \frac{\tau^2}{\tau^2 + \sigma^2}$$

This can get a posterior estimate of this by

- Monte carlo after model fitting
- Within jags model code (see anorexia data, model 1)
- Use estimated posteriors and compute ICC. (see anorexia data model0 or model 1)

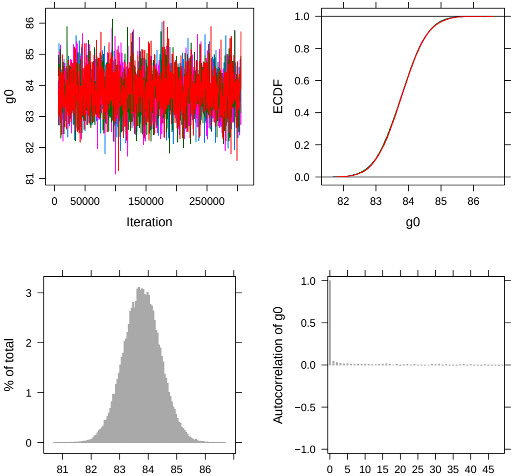
Starting Values

```
start1 = list("g0"=mean(ano$weight),
"sigma"=sd(ano$weight), "tau"=.5,
 .RNG.name="base::Wichmann-Hill", .RNG.seed=523)
start2 = list("g0"=rnorm(1,0,3), "sigma"=5, "tau"=1,
 .RNG.name="base::Marsaglia-Multicarry", .RNG.seed=57)
start3 = list("g0"=rnorm(1,3,4), "sigma"=10, "tau"=5,
 .RNG.name="base::Super-Duper", .RNG.seed=24)
start4 = list("g0"=rnorm(1,-3,10), "sigma"=50,
"tau"=20, .RNG.name="base::Mersenne-Twister",
 .RNG.seed=72100)

start <- list(start1,start2,start3,start4)
```

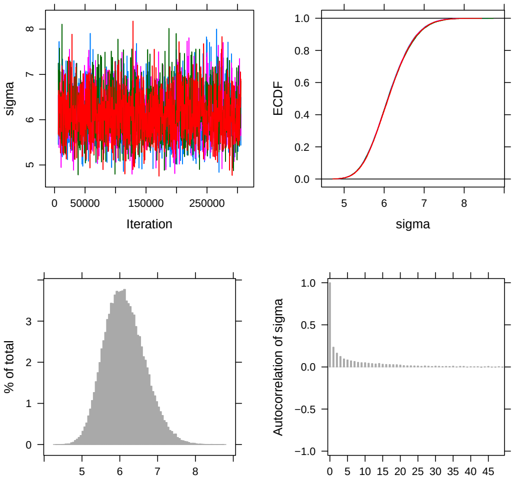
Run Model 0

```
model0.runjags ← run.jags(model=model0,
                method="parallel",
                monitor=c("g0", "sigma", "tau","icc"),
                data=dataList,
                n.chains=4,
                sample=20000,
                burnin=5000,
                inits=start,
                thin=15)

print(model0.runjags)
summary(model0.reml)
plot(model0.runjags)
```

Trace & Density: g_{00}

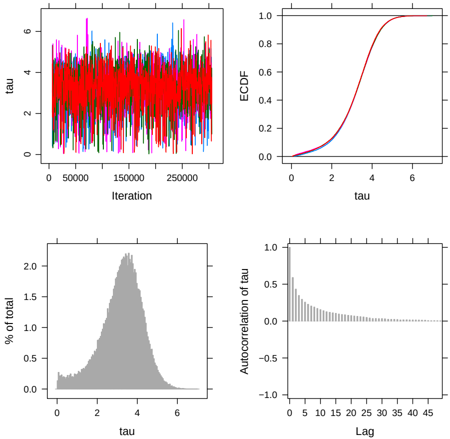


Trace & Density: σ



Trace & Density: τ

Could be better (see lower left)



Results Model 0

JAGS model summary statistics from 80000 samples (thin = 15; chains = 4; adapt+burnin = 6000):

	Lower95	Median	Upper95	Mean	SD	Mode
g0	82.488	83.784	85.058	83.783	0.65056	–
sigma	5.1143	6.0916	7.1706	6.1273	0.53325	–
tau	0.8393	3.3142	5.1464	3.1954	1.0518	–
icc	1.6166e-06	0.23103	0.41407	0.22849	0.11401	–

	MCerr	MC%ofSD	SSeff	AC.150	psrf
g0	0.0034139	0.5	36315	0.011125	0.99999
sigma	0.0039559	0.7	18171	0.050108	1
tau	0.011611	1.1	8206	0.15418	1.0003
icc	0.001154	1	9752	0.14466	1.0004

Total time taken: 44.1 seconds

Results Model 0

For the average girl,

$$\text{weight}_{ij} = 83.783$$

and sample mean $\bar{y} = 83.785$ with variances equal to

$$\begin{aligned} \text{within girl:} \quad & \hat{\sigma}^2 = 6.1273^2 = 37.54381 \\ \text{between girls:} \quad & \hat{\tau}^2 = 3.1954^2 = 10.21058 \\ \text{total:} \quad & \hat{\sigma}^2 + \hat{\tau}^2 = 47.75439 \qquad \text{vs } s^2 = 47.3145 \end{aligned}$$

and

$$\text{ICC} = \frac{10.21058}{47.75493} = .2138$$

Alternative Way to Compute ICC

After fitting model,...

```
    samps ← combine.mcmc(mcmc.objects = model0.runjags
(mcmc))
    # I need to know which columns I need
    colnames(samps)
    sigma2 ← samps[,2]**2
    tau2 ← samps[,3]**2
    icc ← tau2/(sigma2+tau2)
    summary(icc)
```

$$\text{icc} = 0.2271$$

(note: Values may differ slightly from results in example online)

Add some Complexity: Model 1

Adding a fixed effect for time:

$$\begin{aligned} \text{weight}_{ij} &= \beta_{0j} + \beta_1 \text{Time}_{ij} + \epsilon_{ij} \\ &= \gamma_{00} + \gamma_{10} \text{Time}_{ij} + U_{0j} + \epsilon_{ij} \end{aligned}$$

```
dataList ← list(y = ano$weight,
                time = ano$time,
                sdY = sd(ano$weight),
                n = length(ano$weight),
                ng= length(ano$weight)/2,
                girl= ano$girl
           )
```

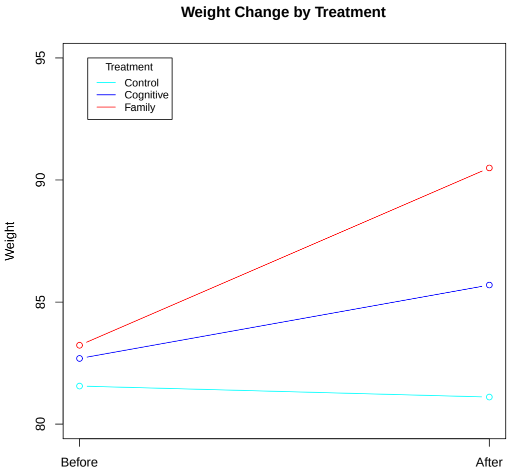
Model 1

```
model1 <- ``model {
          for (i in 1:n) {
             y[i] ~ dnorm(mu[i],precision)
             mu[i] ← beta0j[girl[i]] + g1*time[i]
          }
             for (j in 1:ng) {
          beta0j[j] ~ dnorm(g0,ptau)
          }
          g0 ~ dnorm(0,1/(100*sdY^2))
          g1 ~ dnorm(0,1/(100*sdY^2))
          tau ~ dunif(0.0001,200)
          ptau ← 1/tau^2
          sigma ~ dunif(0.0001,2000)
          precision ← 1/sigma^2
          }''
```

Run Model 1

- Add $g1$ to list of starting values.
- Suggestion: thin=10
- See Rmarkdown to check it.

Model 2: Let's Add treatment



Run Model 2: Add treatment

$$\text{weight}_{ij} = \underbrace{(g0 + U_{0j})}_{\beta_{0j}} + g1 * \text{time}_{ij} + g2 * \text{Rx1}_{ij} + g3 * \text{Rx3}_{ij} + \epsilon_{ij}$$

- Create dummy codes for treatment
```
ano$Rx1 <- ifelse(ano$Rx==1,1,0)
ano$Rx3 <- ifelse(ano$Rx==3,1,0)
```
  Note: Rx1 = Cognitive, and Rx3 = Family.
- Add Rx1 and Rx2 to dataList
- Add to model just like in linear regression
- Add starting values for g2 and g3.
- Run model.
- Did it converge? How does it look? See Rmarkdown

More Models

- Model 3: Random intercept but add interactions between time and treatments.
- Model 4: Model 3 except add random slope (and NO random intercept)
- Model 5: Random intercept and slope but τ_{01} are un-correlated (NO interactions) — let's look at this one
- Really can't do correlated random intercept and slope because not enough time points.

Model 5

$$
\begin{aligned}
\text{weight}_{ij} &= g_{00} + g1 * \text{time}_{ij} + g2 * \text{Rx1}_{ij} + g3 * \text{Rx3}_{ij} + \\
&\quad + U_{0j} + U_{1j} * \text{time}_{ij} + \epsilon_{ij}
\end{aligned}
$$

```
dataList ← list(
            y = ano$weight,
            time = ano$time,
            rx1 = ano$Rx1,
            rx3 = ano$Rx3,
            sdY = sd(ano$weight),
            n = length(ano$weight),
            ng= length(ano$weight)/2,
            girl= ano$girl
            )
```

Model 5

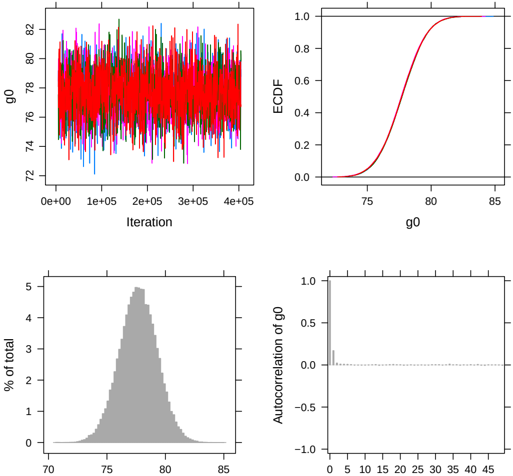
```
model5 ← "model {
for (i in 1:n) {
y[i] ∼ dnorm(mu[i],precision)
mu[i] ← g0 + g1*time[i] + g2*rx1[i] + g3*rx3[i]
             + U0j[girl[i]] + U1j[girl[i]]*time[i]
}

for (j in 1:ng) {
U0j[j] ∼ dnorm(0,ptau0)
U1j[j] ∼ dnorm(0,ptau1)
}

g0 ∼ dnorm(0,1/(100*sdYˆ2))
g1 ∼ dnorm(0,1/(100*sdYˆ2))
g2 ∼ dnorm(0,1/(100*sdYˆ2))
g3 ∼ dnorm(0,1/(100*sdYˆ2))

ptau0 ∼ dgamma(0.001,0.001)
tau0 ← 1/sqrt(ptau0)

tau1 ∼ dunif(0.0001,200)
ptau1 ← 1/tau1ˆ2

sigma ∼ dunif(0.0001,2000)
precision ← 1/sigmaˆ2
}"
```

Model 5

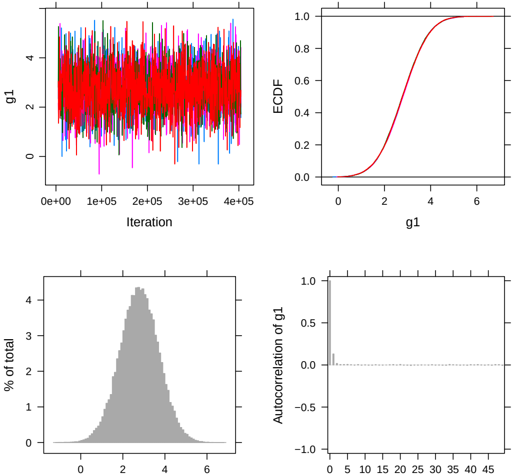
After adding starting values for all parameters,

```
model5.runjags ← run.jags(model=model5,
             method="parallel",
             monitor=c("g0", "g1", "g2", "g3",
                    "sigma", "ptau0", "tau1"),
             data=dataList,
             sample=20000,
             n.chains=4,
             thin=20,
             inits=start)

print(model5.runjags)
plot(model5.runjags)
```

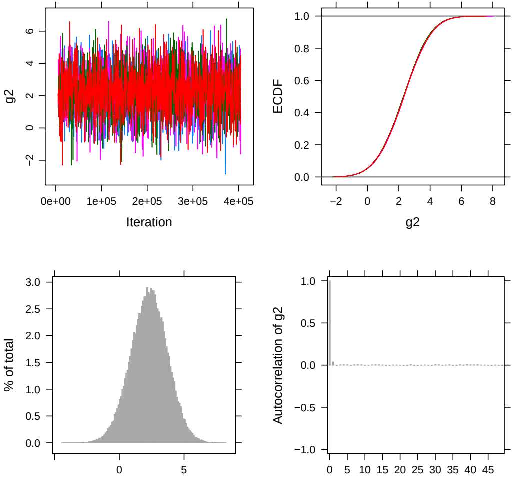
Model 5: Trace & Density g0



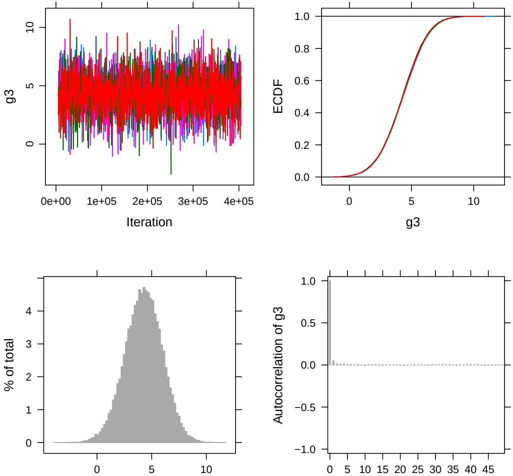
Model 5: Trace & Density g1



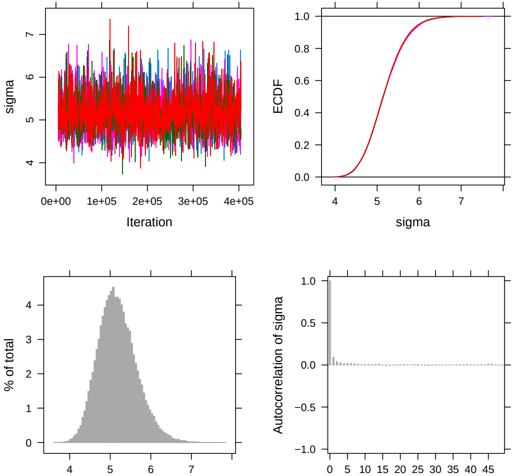
Model 5: Trace & Density g2



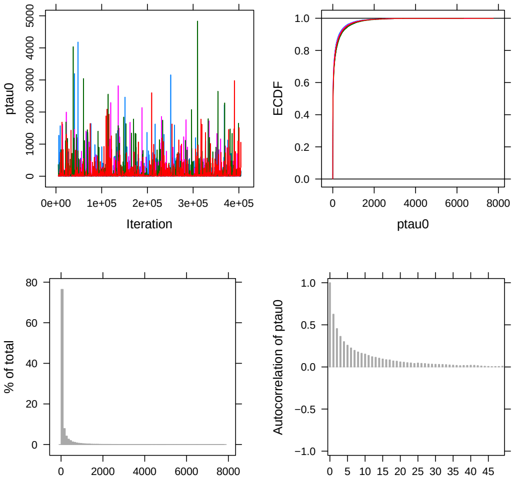
Model 5: Trace & Density g3



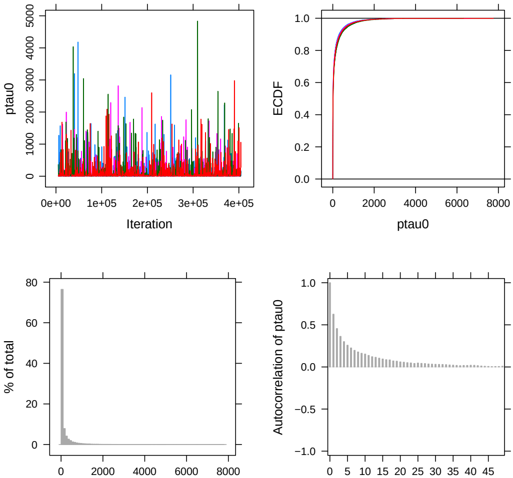
Model 5: Trace & Density sigma



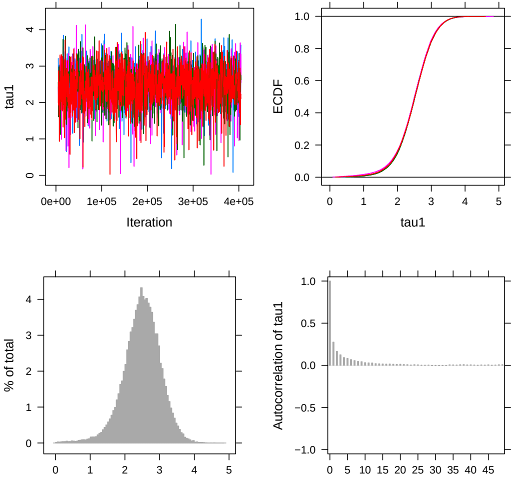
Model 5: Trace & Density ptau1



Model 5: Trace & Density ptau0



This “looks” better: tau1



Model 5: Statistics

JAGS model summary statistics from 80000 samples (thin = 20; chains = 4; adapt+burnin = 5000):

	Lower95	Median	Upper95	Mean	SD
g0	74.503	77.678	80.821	77.675	1.6103
g1	0.98671	2.7887	4.5941	2.7846	0.92155
g2	-0.44021	2.3134	5.1091	2.3	1.4138
g3	0.91706	4.2921	7.6316	4.2688	1.7173
sigma	4.2966	5.1457	6.1231	5.1802	0.47197
ptau0	0.032431	11.653	659.89	126.71	330.29
tau1	1.4083	2.5115	3.5236	2.4871	0.53629

	MCerr	MC%ofSD	SSeff	AC.400	psrf
g0	0.0083055	0.5	37589	-0.0047055	1.0002
g1	0.0046962	0.5	38508	0.0012152	1.0001
g2	0.0070581	0.5	40126	-0.0026701	1.0001
g3	0.0087671	0.5	38369	0.0033132	1.0001
sigma	0.0025477	0.5	34320	0.0087721	1.0001
ptau0	3.7299	1.1	7841	0.18711	1.0018
tau1	0.0036089	0.7	22082	0.016333	1.0009

Total time taken: 49.7 seconds

tau0 = 1/sqrt(ptau0) = 1/sqrt(126.71) = 0.0888

More Complex Models and Data

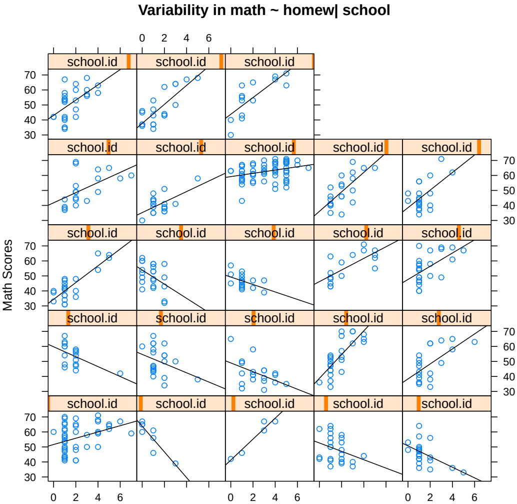
Mini outline:

- The NELS Data from Kreft & deLeeuw:
  - $N = 23$ schools
  - Lots of possible variables
  - Graph of some data
- Start with some simple models ... and then
- Correlated random intercept and random slope
  - Non-conjugate priors
  - Conjugate priors: Wishart distribution

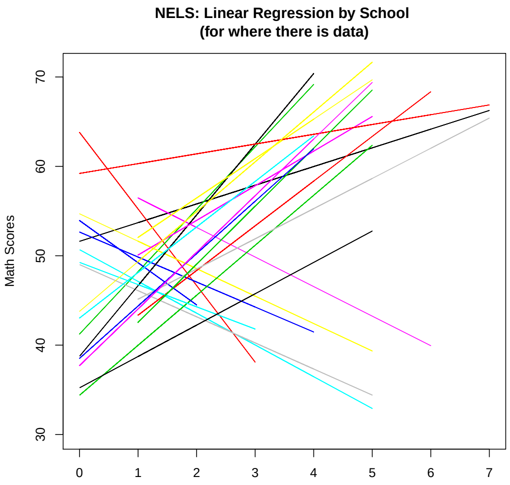
Nels Data:Possible Variables

Level 1 (student)	Level 2 (school)
STUDENT = STUDENT ID	SCHOOL = SCHOOL ID
SEX = STUDENT SEX	SCHTYPE = SCHOOL TYPE
RACE = STUDENT RACE	CLASSSTR= CLASS STRUCTURE
HOMEW = TIME ON MATH HOMEWORK	SCHSIZE = SCHOOL SIZE
SES = SOCIOECONOMIC STATUS	URBAN = URBANICITY
PARED = PARENTAL EDUCATION	GEO = GEOGRAPHIC REGION
MATH = MATH SCORE	MINORITY= PERCENT MINORITY
WHITE = WHITE RACE BINARY	RATIO = STUDENT-TEACHER RATIO

Math x Homework by School



Math x Homework: Expectation for Model?



The Models

The models that are in the R script on course web-site (in lmer model formula):

- Model 1: math∼ 1 + homew + (1 | school.id)
- Model 2: math∼ 1 + homew + (1 |school.id)
  + (0 + homew | school.id)
- Model 3: math∼ 1 + homew + (1 + homew | school.id)
  Conjugate priors, which means we need to know about the Wishart distribution.
- Model 4: math ∼ 1 + homew + ses + public
  + homew*public + (1 + homew | school.id)

Note: Before running Gibbs sampling, we'll use rjags for small number of iterations to do quick checks that model complies and initializes OK (i.e., easy to make mistakes).

NELS Model 1 as HLM

- Level 1:

$$\text{math}_{ij} = \beta_{0j} + \beta_{1j}\text{homework}_{ij} + \epsilon_{ij}$$

- Level 2:

$$\begin{aligned}\beta_{0j} &= \gamma_{00} + U_{0j}\\ \beta_{1j} &= \gamma_{10}\end{aligned}$$

where

$$\begin{pmatrix} U_{0j} \\ \epsilon_{ij} \end{pmatrix} \sim \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & 0 \\ 0 & \sigma^2 \end{pmatrix} \right) \; iid$$

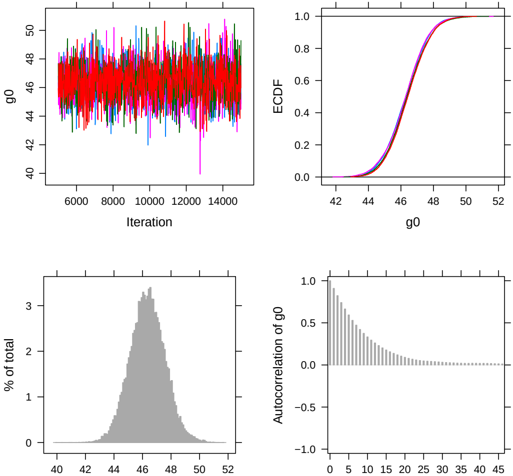
- Linear Mixed Model:

$$\text{math}_{ij} = \underbrace{\gamma_{00} + \gamma_{10}\text{homework}_{ij} + U_{0j}}_{school\ conditional\ mean} + \epsilon_{ij}$$

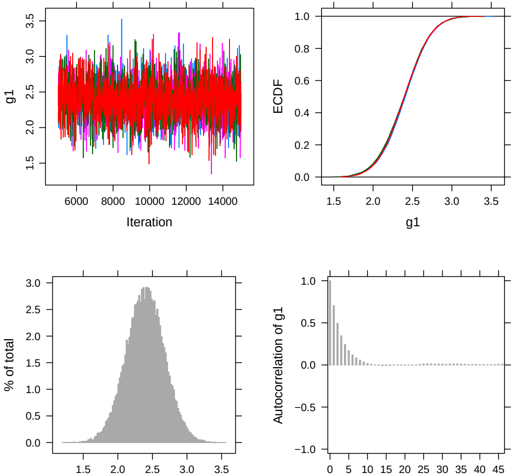
NELS Model 1 in R

```
ri.mod1 ← "model {
    for (i in 1:n) {
       y[i] ∼ dnorm(mu[i],precision)
       mu[i] ← g0 + U0j[school.id[i]] + g1*hmwk[i]
    }
    for (j in 1:N) {
       U0j[j] ∼ dnorm(0,ptau)
    }
    g0 ∼ dnorm(0,1/(100*sdY^2))
    g1 ∼ dnorm(0,1/(100*sdY^2))

    tau ∼ dunif(0.0001,200)
    ptau ← 1/tau^2

    sigma ∼ dunif(0.0001,2000)
    precision ← 1/sigma^2
}"
```

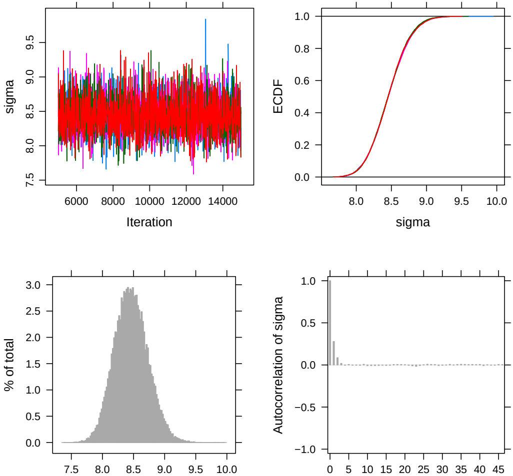
Trace & Density: g0



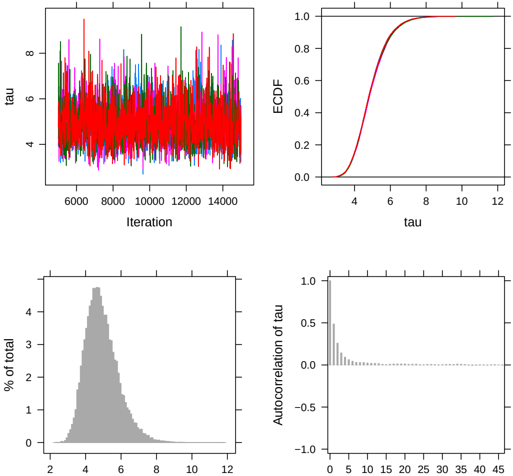
Trace & Density: g1



Trace & Density: sigma



Trace & Density: tau



Model 1: Statistics

JAGS model summary statistics from 40000 samples (chains = 4; adapt+burnin = 5000):

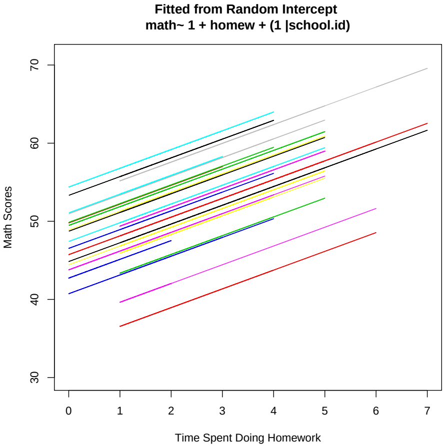
	Lower95	Median	Upper95	Mean	SD
g0	43.909	46.344	48.737	46.346	1.2327
g1	1.8331	2.3992	2.9263	2.3967	0.27808
sigma	7.9444	8.4543	8.9964	8.4615	0.2708
tau	3.2825	4.8041	6.8194	4.9202	0.92725

	MCerr	MC%ofSD	SSeff	AC.10	psrf
g0	0.02616	2.1	2221	0.33119	1.0029
g1	0.0033394	1.2	6934	0.016903	1.0004
sigma	0.0018039	0.7	22536	-0.0071621	1
tau	0.0084287	0.9	12103	0.021921	1.0002

Total time taken: 10.1 seconds

e.g., Fitted Regression Lines

The school conditional regressions. . .



NELS Model 2 as HLM

- Level 1:

$$\text{math}_{ij} = \beta_{0j} + \beta_{1j}\text{homework}_{ij} + \epsilon_{ij}$$

- Level 2:

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + U_{0j} \\ \beta_{1j} &= \gamma_{10} + U_{1j} \end{aligned}$$

where

$$\begin{pmatrix} U_{0j} \\ U_{1j} \\ \epsilon_{ij} \end{pmatrix} \sim \left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & 0 & 0 \\ 0 & \tau_{11} & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix} \right) \; iid$$

- Linear Mixed Model:

$$\text{math}_{ij} = \gamma_{00} + \gamma_{10}\text{homework}_{ij} + U_{0j} + U_{1j}\text{homework}_{ij} + \epsilon_{ij}$$

NELS Model 3 as HLM

- Level 1:

$$\text{math}_{ij} = \beta_{0j} + \beta_{1j}\text{homework}_{ij} + \epsilon_{ij}$$

- Level 2:

$$\begin{aligned} \beta_{0j} &= \gamma_{00} + U_{0j} \\ \beta_{1j} &= \gamma_{10} + U_{1j} \end{aligned}$$

where

$$\begin{pmatrix} U_{0j} \\ U_{1j} \\ \epsilon_{ij} \end{pmatrix} \sim \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{10} & 0 \\ \tau_{10} & \tau_{11} & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix} \right) \; iid$$

- Linear Mixed Model:

$$\text{math}_{ij} = \gamma_{00} + \gamma_{10}\text{homework}_{ij} + U_{0j} + U_{1j}\text{homework}_{ij} + \epsilon_{ij}$$

Prior for τs

For conjugate priors, we need a distribution for the matrix $\boldsymbol{T}$,

$$\boldsymbol{T} = \begin{pmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{11} \end{pmatrix}.$$

However, similar to the univariate case where we use precision, we do so here. For the multivariate case we use

$$\boldsymbol{T}^{-1} = \Omega \sim \text{Wishart}$$

- For a single variance, we know that

$$1/\sigma^2 \sim \text{chi-square}$$

The Wishart Distribution

- Let $z_j \sim N(0, \sigma^2)$ iid , we know
  - $z_j^2 \sim \chi^2$
  - $\sum_{j=1}^m z_j^2 \sim \chi_m^2$
  - $1/\sigma^2 \sim \chi^2$.
- A multivariate generalization to the chi-square is the Wishart distribution and gives of the sampling distribution of covariance matrix. Let $\boldsymbol{Z}_j = (z_1, \ldots, z_p)' \sim MVN(\mathbf{0}, \boldsymbol{\Sigma})$. The Wishart is defined by

$$\begin{aligned} W_m(\cdot|\boldsymbol{\Sigma}) &= \text{Wishart distribution with } m \text{ degrees of freedom} \\ &= \text{The distribution of } \sum_{j=1}^{m} \boldsymbol{Z_j} \boldsymbol{Z_j'} \end{aligned}$$

  where $\boldsymbol{Z_j} \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$ and independent.
- For us,

$$\boldsymbol{T}^{-1} = \boldsymbol{\Omega} \sim \text{Wishart} \qquad \text{and} \qquad \boldsymbol{T} = \boldsymbol{\Omega}^{-1}$$

R model 3

```
re.mod2 ← "model {
# Likelihood:  the data model
    for (i in 1:n) {
       y[i] ∼ dnorm(meanY[i],precision)
       meanY[i] ← betaj[school.id[i],1]
             + betaj[school.id[i],2]*hmwk[i]
    }
# Random Effects:  dmnorm is multivariate normal density
    for (j in 1:N) {
       betaj[j,1:2] ∼ dmnorm(mu[1:2],Omega[1:2,1:2])
    }
# Priors
    precision ∼ dgamma(0.01,0.01)
    sigma ← 1/sqrt(precision)
    mu[1] ∼ dnorm(0,1/(100*sdY^2))
    mu[2] ∼ dnorm(0,1/(100*sdY^2))

    Omega[1:2,1:2]   dwish(R[,],2.1)
       R[1,1] ← 1/2.1
       R[1,2] ← 0
       R[2,1] ← 0
       R[2,2] ← 1/2.1

    Tau ← inverse(Omega)
}"
```

NELS Model 3

Run the R script for this model.

What do you think? Converged? Good Model?

How might we improve the model?

NELS Model 4 – add predictors

- Level 1:

$$\text{math}_{ij} = \beta_{0j} + \beta_{1j}\text{homework}_{ij} + \beta_{2j}\text{ses}_{ij} + \epsilon_{ij}$$

- Level 2:

$$\begin{aligned}\beta_{0j} &= \gamma_{00} + \gamma_{01}\text{public}_j + U_{0j}\\ \beta_{1j} &= \gamma_{10} + \gamma_{11}\text{public}_j + U_{1j}\\ \beta_{2j} &= \gamma_{20}\end{aligned}$$

and

$$\begin{pmatrix} U_{0j} \\ U_{1j} \\ \epsilon_{ij} \end{pmatrix} \sim \left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{10} & 0 \\ \tau_{10} & \tau_{11} & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix} \right) \; iid$$

- What is the Linear Mixed Model?

NELS Model 4 – add predictors

Let's look at R-code and fit the model to data.

Although I didn't do it, you can use Students t-distribution if you think this might be a better likelihood for the data (i.e., if you are concerned about outliers).

Preview to Stan and brms

We will cover Hamletonian sample, Stan and the package brms in a later lecture; however, I wanted to give you a little taste of how easy (good and bad thing) it is to estimate multilevel models using brms. Bare bones example using default priors:

```
library(lme4)
model2.lmer ← lmer(math ∼ 1 + homew + ses + public.fac
          + homew*public.fac + (1 + homew|school.id),
          data=nels, REML=TRUE,
          control = lmerControl(optimizer ="Nelder_Mead"))
```

```
library(brms)
model2.brm ← brm(math ∼ 1 + homew + ses + public.fac +
homew*public.fac
          + homew*public.fac + (1 + homew|school.id),
         data=nels, cores=4, save_all_pars=TRUE)
```

IRT as Multlevel Logistic Regression

Just change link to logit and use Bernoulli for likelihood.

We will use 5 (out of the 11 available) GSS vocabulary items from the 2004 General Social Survey (1155 respondents).

$$A = \begin{cases} 1 & \text{if item is A} \\ 0 & \text{otherwise} \end{cases} \quad \ldots \quad E = \begin{cases} 1 & \text{if item is E} \\ 0 & \text{otherwise} \end{cases}$$

The Random effects logistic regression model:

$$\log\left(\frac{Pr(Y_{ij}=1)}{1-Pr(Y_{ij}=1)}\right) = U_{0j} + \gamma_1 A_{ij} + \gamma_2 B_{ij} + \ldots + \gamma_{10} E_{ij}$$

or

$$Pr(Y_{ij}=1) = \frac{1}{1+\exp(-(U_{0j} + \gamma_1 A_{ij} + \gamma_2 B_{ij} + \ldots + \gamma_{10} E_{ij}))}$$

Multilevel Logistic Regression: R code

$$\log\left(\frac{Pr(Y_{ij}=1)}{Pr(Y_{ij}=0)}\right) = U_{0j} + \gamma_1 A_{ij} + \gamma_2 B_{ij} + \ldots + \gamma_{10} E_{ij}$$

```
dataList ← list(
             id=vo5$id,
             y=vo5$y,
             A=vo5$A,
             B=vo5$B,
             C=vo5$C,
             D=vo5$D,
             E=vo5$E,
             n=length(vo5$y),
             Nid=length(unique(vo5$id))
       )
```

The Model

```
logreg1 ← "model {
for (i in 1:n) {
y[i] ∼ dbern(p[i])
p[i] ← 1/(1 + exp(-eta[i]))
eta[i] ← theta[id[i]] + ba*A[i] + bb*B[i] + bc*C[i] +
bd*D[i] + be*E[i]
}
for (j in 1:Nid) {
theta[j] ∼ dnorm(0,ptau)
}
ptau ∼ dgamma(0.01,0.01)
tau ← 1/sqrt(ptau)
ba ∼ dnorm(0,1/1000)
bb ∼ dnorm(0,1/1000)
bc ∼ dnorm(0,1/1000)
bd ∼ dnorm(0,1/1000)
be ∼ dnorm(0,1/1000)
}"
writeLines(logreg1,con="logreg1.txt")
```

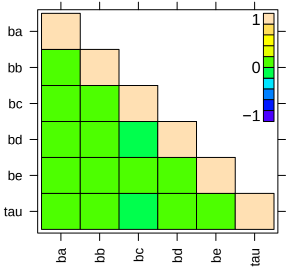
Test Run

Before running for real, even this takes some time because this is a very long data file.

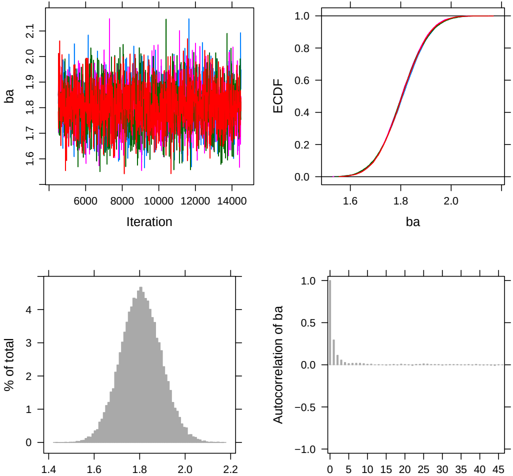
start1 ← list("ba"=2,"bb"=3.0,"bc"=-1.5,
"bd"=3.0, "be"=2.0, "ptau"=.001)

```
logreg1.chk ← run.jags(
            model=logreg1,
            sample=100,
            data=dataList,
            inits=start1,
            monitor=c("ba","bb", "bc", "bd",
                 "be","tau"),
           n.chains=1 )
```

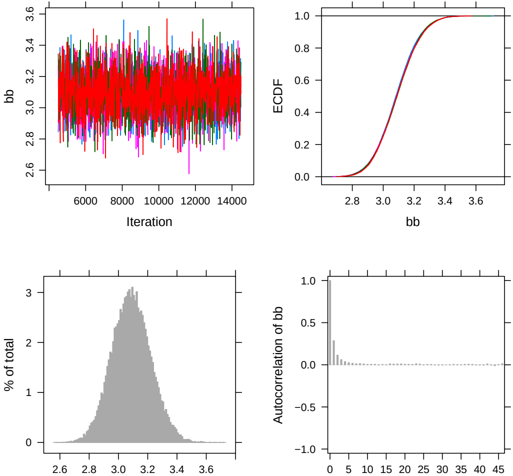
Correlations



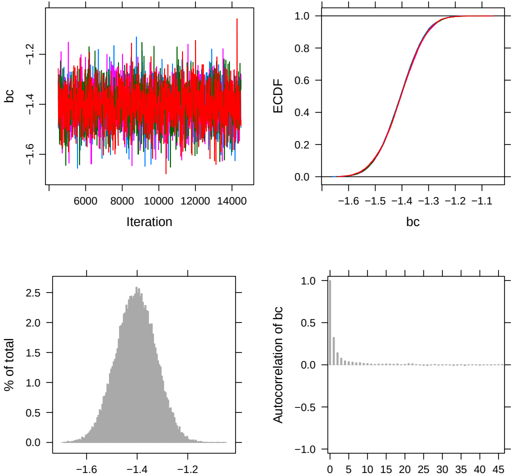
Trace and Density: ba



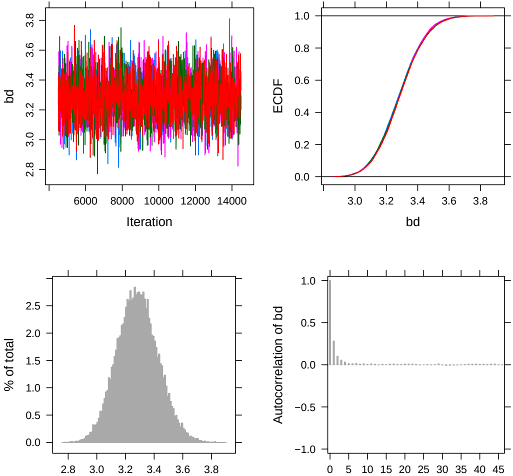
Trace and Density: bb



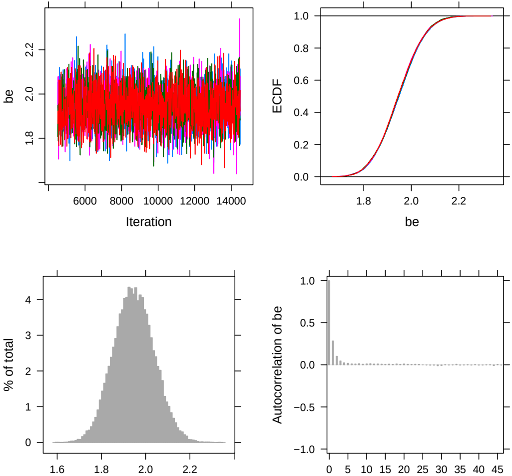
Trace and Density: bc



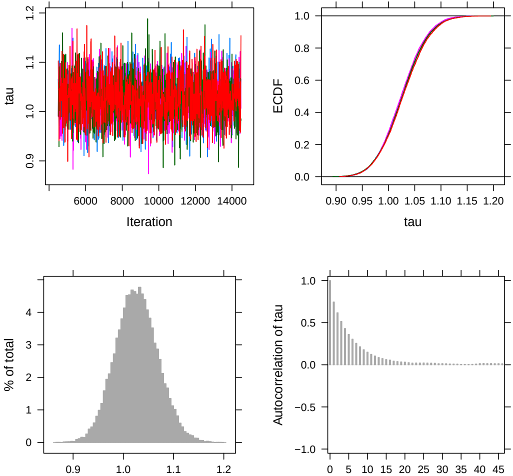
Trace and Density: bd



Trace and Density: be



Trace and Density: tau



From 40,000 Samples

JAGS model summary statistics from 40000 (chains = 4; adapt+burnin = 4500):

	Lower95	Median	Upper95	Mean	SD
ba	1.6311	1.8047	1.9782	1.8056	0.088319
bb	2.8249	3.0865	3.3483	3.088	0.13313
bc	-1.559	-1.4046	-1.2529	-1.4054	0.078671
bd	3.0128	3.2819	3.574	3.2844	0.14347
be	1.7719	1.9457	2.1286	1.9467	0.091723
tau	0.94462	1.0258	1.11	1.0265	0.042328

59.7 Minutes later. . .

	MCerr	MC%ofSD	SSeff	AC.10	psrf
ba	0.00063142	0.7	19565	0.0055738	1.0003
bb	0.00096278	0.7	19121	0.004305	1.0002
bc	0.0006287	0.8	15658	0.014761	1.0001
bd	0.00099655	0.7	20727	0.0043572	1.0003
be	0.00063587	0.7	20808	0.010457	1.0001
tau	0.00066735	1.6	4023	0.14941	1.0008

Multilevel Logistic Regression as an IRT model

$$Pr(Y_{ij}=1)=\frac{1}{1+\exp(-(U_{0j}+\gamma_1 A_{ij}+\gamma_2 B_{ij}+\ldots+\gamma_{10}E_{ij}))}$$

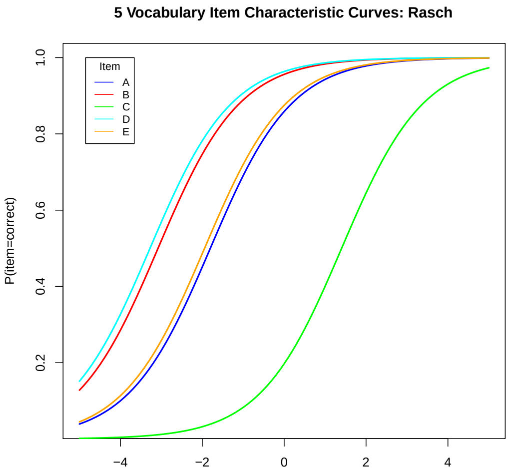
and for say item 2,

$$Pr(Y_{ij}=1)=\frac{1}{1+\exp(-(U_{0j}+\gamma_2))}$$

Set $U_{oj}=\theta$ and $\gamma_2=-b$

What is this model?

Average (i.e., $U_{0j} = \theta_j = 0$) Fitted ICCs



Multilevel Logistic Regression at IRT model

$$Pr(Y_{ij} = 1) = \frac{1}{1 + \exp(-(U_{0j} + \gamma_1 A_{ij} + \gamma_2 B_{ij} + \ldots + \gamma_{10} E_{ij}))}$$

and for say item 2,

$$Pr(Y_{ij} = 1) = \frac{1}{1 + \exp(-(U_{0j} + \gamma_2))}$$

This is a Rasch model.

Other IRT models can be fit using Bayesian methods (as multilevel models).