ESRC Summer School Workshop on Multilevel Modelling using MLwiN and WinBUGS

Nicky Best (Imperial College)
Bill Browne (Nottingham University)

University of Southampton

7th July 2005

Workshop Outline

9.00 -10.30

- Session 1: Introduction to multilevel models
- Practical 1: Fitting Normal multilevel models in MLwiN
- Session 2: Introduction to Bayesian inference

BREAK

11.00 -12.30

- Session 3: Bayesian computation and WinBUGS
- Practical 2: Fitting Normal multilevel models in WinBUGS
- Practical 3: MCMC estimation in MLwiN and the MLwiN to WinBUGS interface

LUNCH

1.30 – 3.00

- Session 4: Generalised Linear Mixed Models (GLMMs)
- Fitting GLMMs in WinBUGS and MLwiN
- Session 5: More complex hierarchical / multilevel models
- Summary

Session 1 : Introduction to Multilevel Models

In this session we will cover

- What is a multilevel model?
- Why do we need them?
- Features of a multilevel model
- Random intercept and slopes models.
- The IGLS algorithm in MLwiN

Multilevel Models

Also known as random effects models, hierarchical linear models, variance components models

Most social systems have a nested structure, for example:

- Students within schools
- Kids within families
- Patients within hospitals
- Repeated measurements within pupils within schools

Multilevel models are good for exploring relationships between variables where data is collected from populations with a nested structure.

Variability across groups – the effect of group membership

In this summer school we will focus on a (tutorial) dataset taken from the field of education. The dataset consists of variables on 4059 pupils from 65 schools in the UK. Our response of interest is the pupil's exam scores at age 16 with the main predictor of interest being an earlier reading test taken at age 11.

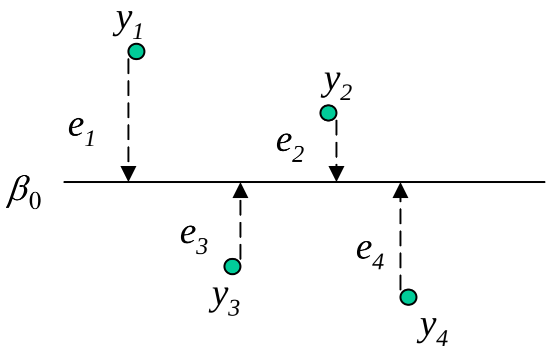
Now if you want to know what the average exam score in secondary school is, where we have a large number of secondary schools, then classical statistical techniques are fine.

However if you want to know how exam score **varies** across schools and what processes drive that variability then you need to use a multilevel model.

Single level models – fitting lines (and more) to data

Pretty much the simplest single level model we can fit is to estimate the mean of a variable (e.g. exam score) . In a regression framework this looks like this :

$$y_i = \beta_0 + e_i$$



This is shown here, schematically, for four data points, although in reality we have many more. β_0 is the mean of y and the departures of each point from the mean are given by $e_{i.}$

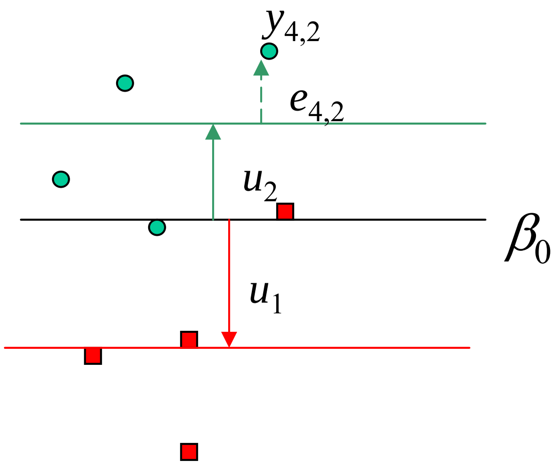
The amount of variation in the data is give by the sum of the squared values of the e_i . (divided by n-1).This quantity is known as the **variance** of $e_{i.}$

Multilevel regression models

Now suppose that we have data grouped into schools, we can write a model

$$y_{ij} = \beta_0 + u_j + e_{ij}$$

Where y_{ij} is now the attainment score for the *i*th child in the *jth* school. β_0+u_j is the mean for the school *j*. u_j is the school effect and e_{ij} is the departure of the *ith* pupil in school *j* from the mean for school *j*.



We can now work out the variance of the school effects, var(u_j)=σ_u^2 and also the variance of the between pupil within school effects, var(e_{ij})=σ_e^2. That is we have partitioned the unexplained variability into two components a part due to school level processes and a part due to pupil level processes.

This is a multilevel model.

Problems with not doing a multilevel analysis

Conceptual : the between school variability and what factors reduce it are generally of fundamental interest. A single level model gives us no estimate of between school variability.

Technical : If the higher level clustering is not properly accounted for in the model then inferences we make about other predictors will be incorrect. We will tend to infer a relationship where none exists.

Problems with not doing a multilevel analysis

Another way to think of the problem is to ask whether a single level model adequately describes the variation in the response, or whether the data are more variable than the model assumes.

Some reasons for excess variation in response data include:

- Individual heterogeneity i.e. systematic differences between units which are not attributable to random variation.
    - for binary/count data this is often termed over-dispersion.
- Repeated response measurements from the same unit tend to be correlated
    - 2 responses from the same unit will be more alike than 2 responses from different units.
- Failure to measure or include a relevant explanatory variable.
- Inaccurate measurement of relevant explanatory variables.

Random effects vs fixed effects

The basic multilevel model uses random effects, so actually the full equation is

$$y_{ij} = \beta_0 + u_j + e_{ij}$$

$$u_j \sim N(0,\sigma_u^2) \;\; e_{ij} \sim N(0,\sigma_e^2) \qquad (1)$$

This looks very similar to the basic analysis of variance model :

$$y_{ij} = \beta_0 + u_j + e_{ij}$$

$$e_{ij} \sim N(0,\sigma_e^2) \qquad (2)$$

The key difference being that in (1) the school effects are random variables and are assumed to have a Normal distribution with variance $\sigma_u{}^2$. In (2) schools are regarded as being independent. In (1) we often assume that we have taken a ***random sample*** of schools from a ***population*** of schools.

Random versus fixed effects continued

Random effects

$$y_{ij} = \beta_0 + u_j + e_{ij}$$

$$u_j \sim N(0,\sigma_u^2) \quad e_{ij} \sim N(0,\sigma_e^2)$$

It is either assumed we have a sample from a larger population of groups and we wish to make inferences to the wider sample or that our sample is exchangeable.

In estimating a school mean all the other schools are taken into account.

After fitting group effects further group level variables can be added.

Fixed effects

$$y_{ij} = \beta_0 + u_j + e_{ij}$$

$$e_{ij} \sim N(0,\sigma_e^2)$$

It is assumed **all** groups of interest have been sampled. Origins in agricultural statistics.

In estimating a school mean all other schools are ignored.

After fitting group effects further group level variables cannot be added

When is a variable a level?

Schools can be thought of as a level but gender can not. Why?

For a variable to be a level we must have a population of units from which we have taken a random sample. For example, people, schools, families etc. We regard the units as ***exchangeable*****.** Note exchangeability can often be justified when the sample is in fact the population.

A more general term that subsumes level is ***random classification***. If we have 2 levels in our structure a nested relationship is assumed. If we have 2 random classifications no nesting is assumed.

Where a classification has a small number of units we might prefer the term ***category*** to unit. For example, gender has categories male and female and social class is broken down as a set of discrete categories. These categories are not exchangeable and we refer to such classifications as ***fixed classifications.***

School is a random classification but school gender(mixed,boys, girls) is a fixed classification.

Adding explanatory variables

The basic single level regression model, for example, regressing child exam score at age 16 on intake reading score is

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

We can extend this to a multilevel model :

$$y_{ij} = \beta_{0j} + \beta_1 x_{ij} + e_{ij}$$

$$\beta_{0j} = \beta_0 + u_{0j}$$

$$u_{0j} \sim N(0, \sigma^2_{u0}) \;\; e_{ij} \sim N(0, \sigma^2_e)$$

y_{ij} and x_{ij} are the exam score and reading scores for child *i* in school *j*.

In this model we have an average linear relationship between attainment and intake score given by the intercept(β_0) and slope (β_1) . Schools depart from this average line in terms of the intercept only by an amount $u_{0j.}$

Random intercept model (parallel lines)

The summary line for the *j*th school is given by

$$\beta_0 + \beta_1 x_{ij} + u_{0j}$$

Pupils are distributed around their school's summary line with departure $e_{ij.}$

Terminology

β_0, β_1 : fixed effects

u_{0j}, e_{ij} : random effects or multilevel residuals

$\sigma_{u0}^2, \sigma_{\mathrm{e}}^2$: random parameters

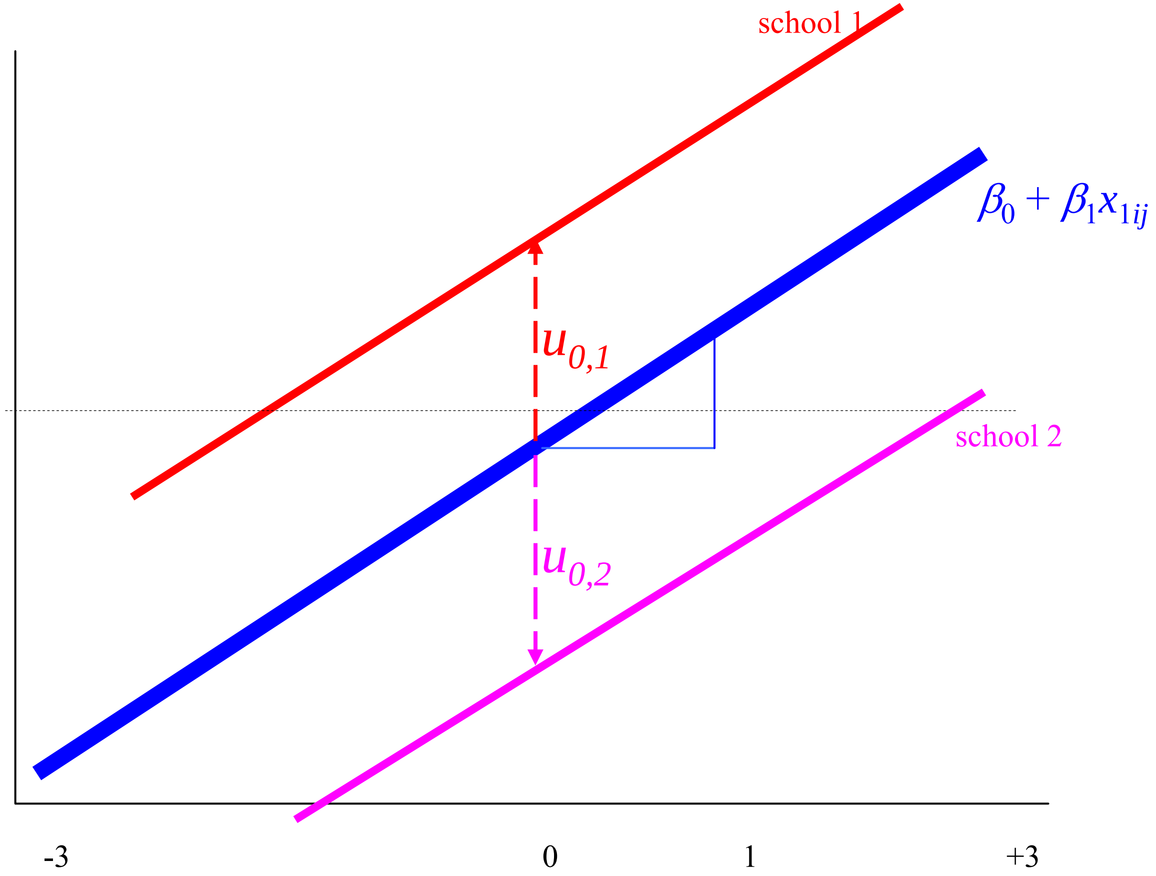
Random intercept models(parallel lines)

$$y_{ij} = \beta_{0j} + \beta_1 x_{ij} + e_{ij}$$

$$\beta_{0j} = \beta_0 + u_{0j}$$

$$u_{0j} \sim N(0, \sigma_{u0}^2)$$

$$e_{ij} \sim N(0, \sigma_e^2)$$



Variance partition coefficient(VPC)

Given the basic random intercept model

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + u_j + e_{ij}$$

$$u_j \sim N(0,\sigma_u^2) \qquad e_{ij} \sim N(0,\sigma_e^2)$$

Then the VPC is the proportion of the total variance at the higher level. That is

$$\frac{\sigma_u^2}{\sigma_u^2 + \sigma_e^2}$$

This is the clustering effect of higher level units. As this increases it becomes more important for the technical and substantive reasons given earlier to use multilevel modelling. Note that in this model the formula for the VPC is the same as for another concept called the intra-class correlation or ICC but this is not true for all multilevel models.

Multilevel residuals

In single level models there is only 1 residual that can be obtained by

$$\hat{e}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$$

In a multilevel model we have a raw residual

$$r_{ij} = y_{ij} - (\hat{\beta}_0 + \hat{\beta}_1 x_{ij})$$

Which we wish to decompose into 2 parts a school residual and a pupil residual, that is

$$\hat{u}_{0j}, \hat{e}_{ij}$$

Shrinkage

Let r_{+j} be the mean of the raw residuals over the pupils in school *j*. Then the predicted level 2 residual for school *j* is obtained by multiplying r_{+j} by a factor as follows

$$\hat{u}_{0j} = \frac{\sigma_{u0}^2}{\sigma_{u0}^2 + \sigma_e^2 / n_j} r_{+j}$$

Where n_j is the number of pupils in school *j*. The multiplier is always less than one and we say the raw residual has been multiplied by a *shrinkage factor*.

The shrinkage will be large if n_j is small or σ_e^2 is large compared to σ_u^2 or both.

In either case we have relatively little information about the school because its students are very variable or few in number.

Random slope model (crossing lines)

The assumption that all the school's lines are parallel is probably unrealistic, we can extend the model to allow schools to have different slopes:

$$y_{ij} = \beta_{0j} + \beta_{1j}x_{ij} + e_{ij}$$

$$\beta_{0j} = \beta_0 + u_{0j}$$

$$\beta_{1j} = \beta_1 + u_{1j}$$

$$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N(0, \Omega_u) \;\; \Omega_u = \begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u01} & \sigma^2_{u1} \end{bmatrix}$$

$$e_{ij} \sim N\left(0, \sigma^2_e\right)$$

Now each school has an intercept residual(u_{0j}) and a slope residual(u_{1j}). We can therefore estimate the school level intercept/slope covariance(σ_{u01})

Random slopes model (crossing lines)

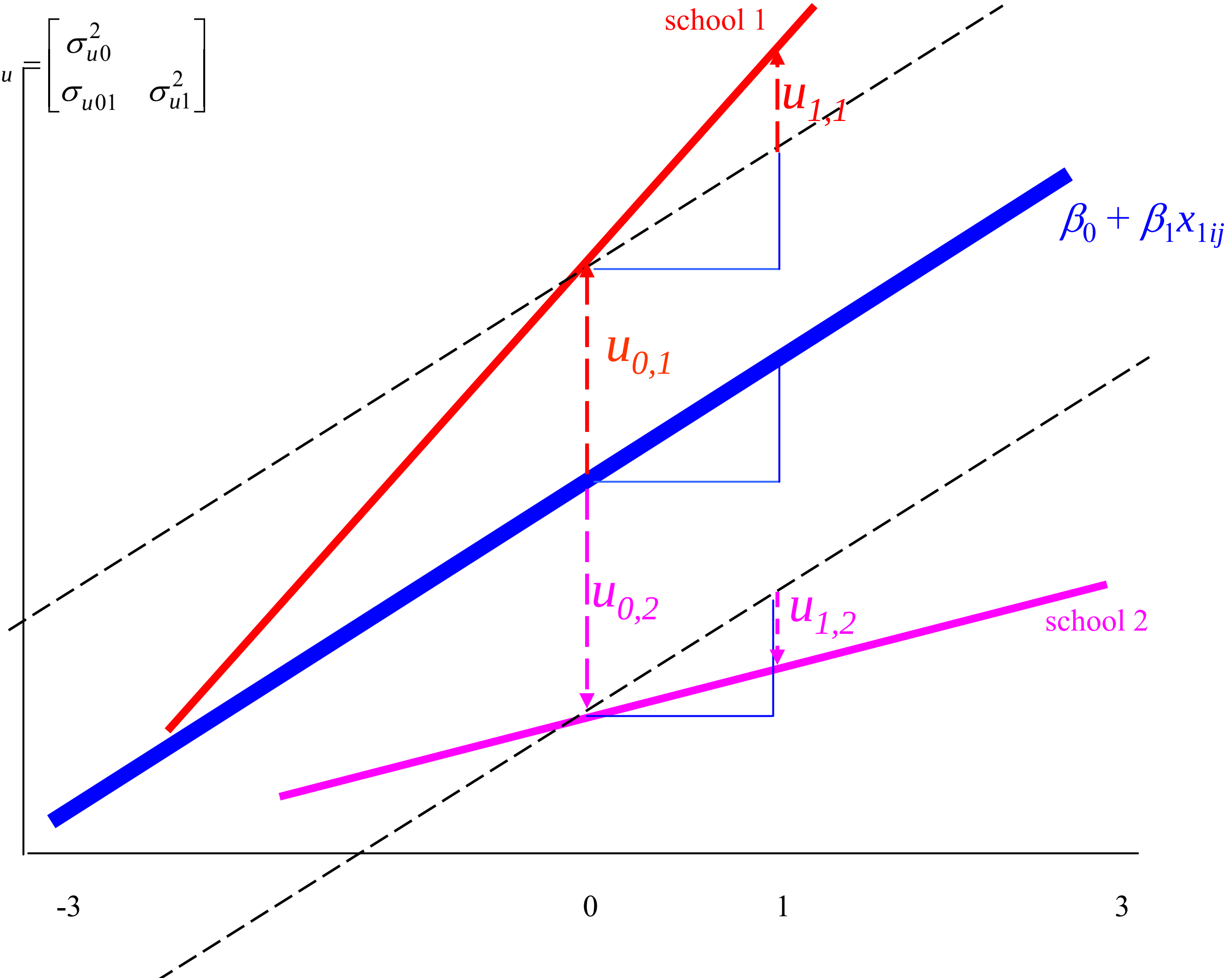
$$y_{ij} = \beta_{0j} + \beta_{1j}x_{ij} + e_{ij}$$

$$\beta_{0j} = \beta_0 + u_{0j}$$

$$\beta_{1j} = \beta_1 + u_{1j}$$

$$\begin{bmatrix} u_{0j} \\ u_{1j} \end{bmatrix} \sim N(0,\Omega_u) : \Omega_u = \begin{bmatrix} \sigma^2_{u0} & \\ \sigma_{u01} & \sigma^2_{u1} \end{bmatrix}$$

$$e_{ij} \sim N(0,\sigma^2_e)$$



Fitting a Multilevel Model in MLwiN : The IGLS algorithm

- Linear (1-level) models can be estimated by simple matrix formulae.
- The introduction of a 2nd level and hence more terms that describe the variance means such simple formulae are no longer available.
- Instead an alternative approach is needed.
- MLwiN uses the IGLS (Iterative generalised least squares) algorithm.
- Here the fixed and random parameters are estimated in turn (conditioning on each other) resulting in 2 steps that converge to the ML estimates for the model.

IGLS algorithm

- Designed for structured multivariate Normal response models i.e. Y~MVN(Xβ,*V*).
- The *n*×*n* variance matrix *V* is structured to correspond to the multilevel model and is block diagonal.
- For a 2-level variance components model

$$V_{ii} = \sigma_e^2 + \sigma_u^2,$$

$$V_{ij} = \sigma_u^2 \text{ if } i,\ j \text{ in same level 2 unit,}$$

$$V_{ij} = 0 \text{ if } i,\ j \text{ in different level 2 units.}$$

IGLS algorithm (continued)

Two steps that are alternated between until convergence:

- Step 1: Update β assuming *V* known using GLS.

$$\hat{\beta} = (X^T V^{-1} X)^{-1} X^T V^{-1} Y$$

- Step 2: Update *V* assuming β known.

Calculate the vector of residuals,

$\widetilde{R} = \{r_i\}$ where $r_i = Y_i - X_i \hat{\beta}$

Vectorise the cross-product of $\widetilde{R}, vec(\widetilde{R}\widetilde{R}^T)$

This vector has expectation, *vec*(*V*) and so we can use least squares to find estimates of σ_e^2 and σ_u^2

IGLS algorithm (continued)

- Note that algorithm can give negative variance estimates and non-positive definite variance matrices in random slopes models.
- However in MLwiN positivity constraints for variances are added.

Further Reading

- Goldstein (2003) Multilevel Statistical Modelling (3rd Edition)
- Rasbash, Steele, Browne & Prosser (2004). A User’s Guide to MLwiN (version 2.0)
- Snijders & Bosker (1999) Multilevel Analysis.
- Browne (2004) MCMC Estimation in MLwiN.

Session 2. Introduction to Bayesian inference

In this session we will cover

- What is Bayesian inference
- Simple example of conjugate Bayesian analysis
- Link between Bayesian inference, exchangeability and multilevel models
- Bayesian prior distributions
- Comparison on Bayesian and classical multilevel models

Bayes theorem and its link with Bayesian inference

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}.$$

If A_i is a set of mutually exclusive and exhaustive events (*i.e.* $p(\bigcup_i A_i) = \sum_i p(A_i) = 1$), then

$$p(A_i|B) = \frac{p(B|A_i)p(A_i)}{\sum_j p(B|A_j)p(A_j)}.$$

Example: use of Bayes theorem in diagnostic testing

- A new HIV test is claimed to have "95% sensitivity and 98% specificity"
- In a population with an HIV prevalence of 1/1000, what is the chance that patient testing positive actually has HIV?

Let A be the event that patient is truly HIV positive, $\overline{A}$ be the event that they are truly HIV negative.

Let B be the event that they test positive.

We want $p(A|B)$.

"95% sensitivity" means that $p(B|A) = .95$.

"98% specificity" means that $p(B|\overline{A}) = .02$.

Now Bayes theorem says

$$p(A|B) = \frac{p(B|A)p(A)}{p(B|A)p(A) + p(B|\overline{A})p(\overline{A})}.$$

Hence $p(A|B) = \frac{.95 \times .001}{.95 \times .001 + .02 \times .999} = .045$.

Thus over 95% of those testing positive will, in fact, not have HIV.

- Our intuition is poor when processing probabilistic evidence

- The vital issue is *how should this test result change our belief that patient is HIV positive?*

- The disease prevalence can be thought of as a '*prior*' probability ($p = 0.001$)

- Observing a positive result causes us to modify this probability to $p = 0.045$. This is our '*posterior*' probability that patient is HIV positive.

- Bayes theorem applied to *observables* (as in diagnostic testing) is uncontroversial and established

- More controversial is the use of Bayes theorem in general statistical analyses, where *parameters* are the unknown quantities, and their prior distribution needs to be specified — this is **Bayesian inference**

Bayesian inference

Makes fundamental distinction between

- Observable quantities x, i.e. the data
- Unknown quantities θ

θ can be statistical parameters, missing data, mismeasured data ...

$\longrightarrow$ parameters are treated as random variables

$\longrightarrow$ in the Bayesian framework, we make probability statements about model parameters

! in the frequentist framework, parameters are fixed non-random quantities and the probability statements concern the data

As with any statistical analysis, we start by positing a model which specifies

$$p(x \mid \theta)$$

This is the **likelihood**, which relates all variables into a **'full probability model'**

From a Bayesian point of view

- θ is unknown so should have a **probability distribution** reflecting our uncertainty about it before seeing the data
  $\rightarrow$ need to specify a **prior distribution** $p(\theta)$

- x is known so we should condition on it
  $\rightarrow$ use Bayes theorem to obtain conditional probability distribution for unobserved quantities of interest given the data:

  $$p(\theta \mid x) = \frac{p(\theta)\, p(x \mid \theta)}{\int p(\theta)\, p(x \mid \theta)\, d\theta} \propto p(\theta)\, p(x \mid \theta)$$

  This is the **posterior distribution**

The prior distribution $p(\theta)$, expresses our uncertainty about θ **before** seeing the data.

The posterior distribution $p(\theta \mid x)$, expresses our uncertainty about θ **after** seeing the data.

Bayesian inference using the Normal distribution

Known variance, unknown mean

Suppose we have a sample of Normal data $x_i \sim \mathsf{N}(\theta, \sigma^2) \quad (i = 1, ..., n)$. For now assume σ^2 is known and θ has a Normal prior $\theta \sim \mathsf{N}(\mu, \sigma^2/n_0)$

Then the posterior distribution is

$$\begin{aligned} p(\theta|\boldsymbol{x}) &\propto \prod_i p(x_i \,|\, \theta)\; p(\theta) \\ &\propto \exp\left[-\frac{\sum_i (x_i - \theta)^2}{2\sigma^2}\right] \times \exp\left[-\frac{(\theta - \mu)^2 n_0}{2\sigma^2}\right] \end{aligned}$$

By matching terms in θ and writing $\sum x_i = n\overline{x}$ it can be shown that

$$\sum_i (x_i - \theta)^2 + (\theta - \mu)^2 n_0 = \left(\theta - \frac{n_0\mu + n\overline{x}}{n_0 + n}\right)^2 (n_0 + n) + \text{constant}$$

The term involving θ is exactly that arising from a Normal distribution, so

$$p(\theta|\boldsymbol{x}) = \mathsf{N}\left(\frac{n_0\mu + n\overline{x}}{n_0 + n}, \frac{\sigma^2}{n_0 + n}\right)$$

- Same standard deviation σ is used in the likelihood and the prior

- Prior variance is based on an 'implicit' sample size n_0

- As n_0 tends to 0, the variance becomes larger and the distribution becomes 'flatter', and in the limit the distribution becomes essentially uniform over $-\infty, \infty$

- Posterior mean $(n_0\mu + n\overline{x})/(n_0 + n)$ is a weighted average of the prior mean μ and parameter estimate $\overline{x}$, weighted by their precisions (relative 'sample sizes'), and so is always a compromise between the two

- Posterior variance is based on an implicit sample size equivalent to the sum of the prior 'sample size' n_0 and the sample size of the data n

Large sample properties

As $n \to \infty$,

$$\begin{aligned} \text{posterior mean,}(n_0\mu + n\bar{x})/(n_0 + n) &\to \bar{x} \\ \text{posterior variance,}\sigma^2/(n_0 + n) &\to \sigma^2/n \\ \text{and so posterior distribution,}p(\theta|\boldsymbol{x}) &\to \mathsf{N}(\bar{x}, \sigma^2/n) \end{aligned}$$

which do not depend on the prior

In the frequentist setting, the MLE is $\widehat{\theta} = \bar{x}$ with $\mathsf{SE}(\widehat{\theta}) = \sigma/\sqrt{n}$, and sampling distribution

$$p(\widehat{\theta} \mid \theta) = p(\bar{x}|\theta) = \mathsf{N}(\theta,\, \sigma^2/n),$$

whereas in the Bayesian framework, the "dual statement" is made:

$$p(\theta \mid \bar{x}) \to \mathsf{N}(\bar{x},\, \sigma^2/n)$$

Example: SBP — Bayesian analysis for Normal data

Interested in the long-term systolic blood pressure in mmHg (SBP) of a particular 60-year old female

Take 2 independent readings 6 weeks apart and their mean is 130

We know that systolic blood pressure is measured with a standard deviation $\sigma = 5$

What should we estimate her SBP to be?

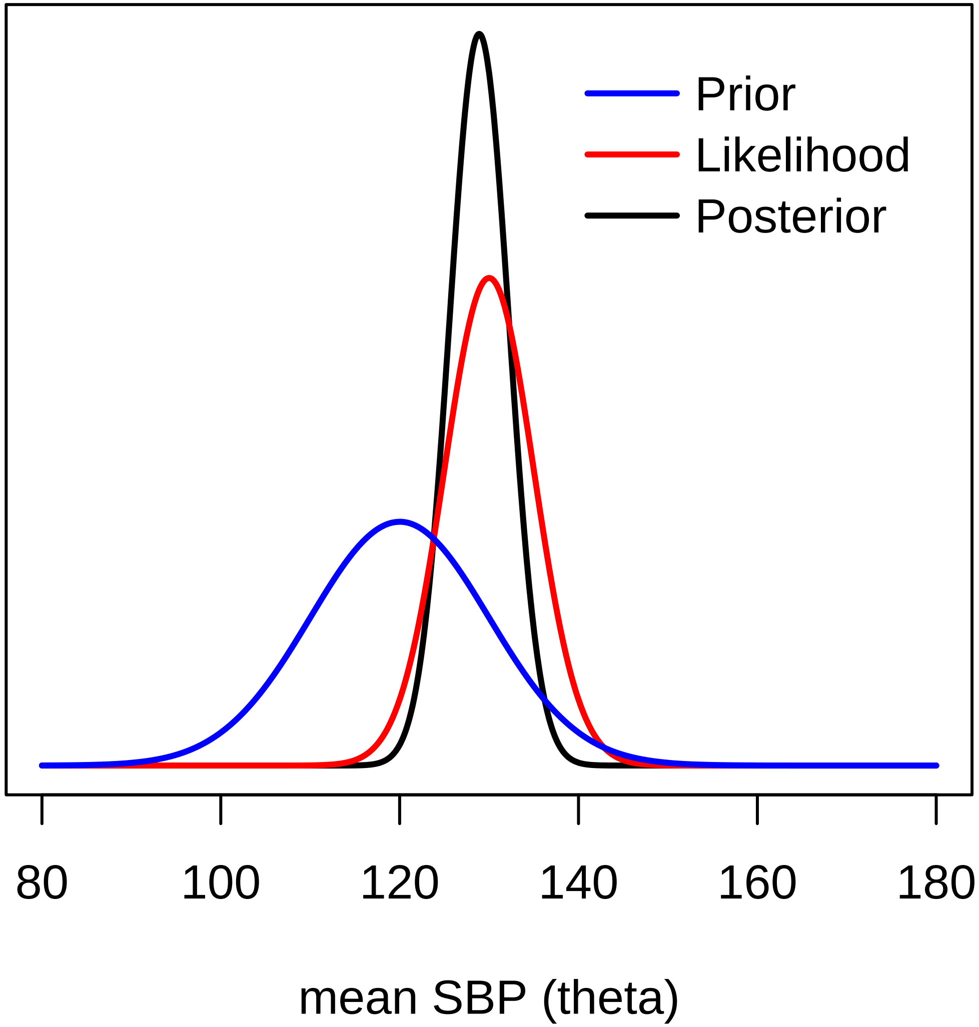
Let her long-term SBP be denoted θ. A standard analysis would use the sample mean $\overline{x} = 130$ as an estimate, with standard error $\sigma/\sqrt{n} = 5/\sqrt{2} = 3.5$: a 95% confidence interval is $\overline{x} \pm 1.96 \times \sigma/\sqrt{n}$, i.e. 123.1 to 136.9.

Suppose a survey in the same population revealed females aged 60 had a mean long-term SBP of 120 with standard deviation 10

- suggests Normal$(120, 10^2)$ prior form θ
- if we express the prior standard deviation as $\sigma/\sqrt{n_0}$, we can solve to find $n_0 = (\sigma/10)^2 = 0.25$
- so our prior can be written as $\theta \sim$ Normal$(120, \sigma^2/0.25)$

Posterior for θ is then

$$\begin{aligned} p(\theta|\boldsymbol{x}) &= \text{Normal}\left(\frac{0.25 \times 120 + 2 \times 130}{0.25 + 2},\ \frac{5^2}{0.25 + 2}\right) \\ &= \text{Normal}(128.9,\ 11.1) \quad \text{giving 95\% interval for } \theta \text{ of } 122.4 \text{ to } 135.4 \end{aligned}$$


Prior
Likelihood
Posterior
80
100
120
140
160
180
mean SBP (theta)

Summary

When the posterior is in the same family as the prior then we have what is known as *conjugacy*. Examples include:

Likelihood	Parameter	Prior	Posterior
Normal	mean	Normal	Normal
Normal	precision	Gamma	Gamma
Binomial	success prob.	Beta	Beta
Poisson	rate or mean	Gamma	Gamma

In all cases

- the posterior mean is a compromise between the prior mean and the MLE
- the posterior s.d. is less than each of the prior s.d. and the s.e.(MLE)

> *'A Bayesian is one who, vaguely expecting a horse and catching a glimpse of a donkey, strongly concludes he has seen a mule'* (Senn, 1997)

As $n \to \infty$,

- the posterior mean → the MLE
- the posterior s.d. → the s.e.(MLE)
- the posterior does not depend on the prior.

These observations are generally true, when the MLE exists and is unique

Further reading

Berry (1996) (Introductory text on Bayesian methods, with medical slant)

Lee (2004) (Good intro to Bayesian inference; more mathematical than Berry)

Bernardo and Smith (1994) (Advanced text on Bayesian theory)

Exchangeability

'Exchangeability' is a formal expression of the idea that we find no systematic reason to distinguish the individual random variables $X_1, ..., X_n$

→ A *judgement* that they are 'similar' but not identical

We judge that $X_1, ..., X_n$ are exchangeable if the probability that we assign to any set of potential outcomes, $p(x_1, ..., x_n)$, is unaffected by permutations of the labels attached to the variables

e.g. suppose X_1, X_2, X_3 are the first three tosses of a (possibly biased) coin, where $X_1 = 1$ indicates a head, and $X_1 = 0$ indicates a tail

We might judge $p(X_1 = 1, X_2 = 0, X_3 = 1) = p(X_2 = 1, X_1 = 0, X_3 = 1) = p(X_1 = 1, X_3 = 0, X_2 = 1)$: i.e. the probability of getting 2 heads and a tail is unaffected by the particular toss on which the tail comes

This is a natural judgement to make if we have no reason to think that one toss is systematically any different from another

Note that it does *not* mean we believe that $X_1, ..., X_n$ are independent: this would not allow us to learn about the chance of a head

Representation theorem

de Finetti (1930) showed that if a set of binary variables $X_1, ..., X_n$ were judged exchangeable, then it implied that

$$p(x_1, ..., x_n) = \int \prod_{i=1}^{n} p(x_i|\theta)\ p(\theta)\ d\theta$$

Easy if argue from ‘right to left’

From ‘left to right’ is remarkable: exchangeable random quantities can be thought of as being *independently and identically distributed* drawn from some common *parametric distribution* depending on an unknown parameter θ, which itself has a *prior distribution* $p(\theta)$

Thus, from a subjective judgment about observable quantities, one derives the whole apparatus of parametric models and Bayesian statistics!

Link between exchangeability, representation thm and hierarchical models

Recall the represenation theorem for exchangeable random variables:
$p(x_1, ..., x_n) = \int \prod_{i=1}^n \theta^{x_i}(1-\theta)^{(1-x_i)} p(\theta) d\theta$

Suppose x_{ij} is outcome for individual j, unit i, with unit-specific parameter θ_i

- Assumption of partial exchangeability of individuals within units can be represented by the following model:

$$\begin{array}{rcl} x_{ij} & \sim & p(x_{ij}|\theta_i) \\ \theta_i & \sim & p(\theta_i) \end{array}$$

- Assumption of exchangeability of the units can be represented by the model:

$$\begin{array}{rcl} \theta_i & \sim & p(\theta_i|\phi) \\ \phi & \sim & p(\phi) \end{array}$$

  – can be considered as a common prior for all units, but one with unknown parameters

Note that there does not need to be any actual sampling — perhaps these I units are the only ones that exist — since the probability structure is a consequence of the belief in exchangeability rather than a physical randomisation mechanism

We emphasise that an assumption of exchangeability is a *judgement* based on our knowledge of the context.

Assuming $\theta_1, ...\theta_I$ are drawn from some common prior distribution whose parameters are unknown is known as a **hierarchical** or **multi-level** model

General form of a Bayesian hierarchical model

Observables x,

Parameters $\theta = (\theta_1, \ldots, \theta_n)$

- likelihood $p(x|\theta)$ models the structure of observables (1st level)
- prior $p(\theta)$ is decomposed into conditional distributions expressing judgements about exchangeability:
  $p(\theta|\phi_2)$ (2nd level),
  $p(\phi_2|\phi_3)$ (3rd level),
  ...
  and a marginal distribution $p(\phi_m)$, such that

$$p(\theta) = \int p(\theta|\phi_2)p(\phi_2|\phi_3)\ldots p(\phi_{m-1}|\phi_m)p(\phi_m)d\phi_2 d\phi_3 \ldots d\phi_m$$

  ϕ_k are called the hyperparameters of level k

Can view hierarchical models as a way of simplifying specification of the joint prior on θ

Provides a way of ‘estimating’ the prior distribution

Schools example

$$\begin{array}{rcl} y_{ij} & \sim & \text{Normal}(\mu_{ij}, \sigma_e^2) \quad \text{child } i\text{, school } j \\ \mu_{ij} & = & \beta_0 + \beta_1 x_{ij} + u_{0j} \\ u_{0j} & \sim & \text{Normal}(0, \sigma_{u0}^2) \end{array}$$

- Can think of random effects distribution as prior on u_{0j} which itself depends on unknown parameters (σ_{u0}^2)

- Bayesian framework → need priors on *all* unknown parameters, so also need to specify priors for β_0, β_1, σ_e^2, σ_{u0}^2

Specifying priors: some recommendations

Distinguish

- *primary* parameters of interest in which one may want minimal influence of priors
- *secondary* structure used for smoothing *etc.* in which informative priors may be more acceptable

Invariance arguments can suggest suitable scale on which to be 'uniform'

Prior best placed on interpretable parameters

Great caution needed in complex models that an apparently innocuous uniform prior is not introducing substantial information

'There is no such thing as a 'noninformative' prior. Even improper priors give information: all possible values are equally likely' (Fisher, 1996)

Location parameters

e.g. means, regression coefficients:

$$\begin{aligned}\beta &\sim \text{Unif}(-100, 100)\\ \beta &\sim \text{Normal}(0, 100000)\end{aligned}$$

Prior will be locally uniform over the region supported by the likelihood

Scale parameters

- Sample variance σ_e^2: standard 'reference' prior

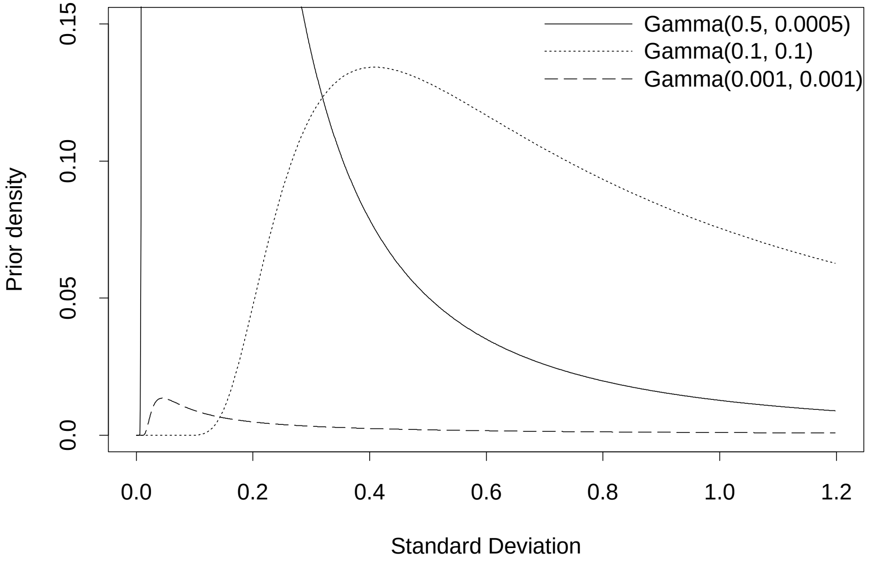
  $$\begin{aligned}p(\text{log}(\sigma_e^2)) &= \text{Uniform}(-\infty, \infty)\\ \text{which is equivalent to} & \\ p(\sigma_e^2) &= \text{Gamma}(0, 0) \quad (\text{i.e. } p(\sigma_e^2) \propto \tfrac{1}{\sigma_e^2})\end{aligned}$$

  Note that this is an **improper prior**, but when combined with an informative likelihood will result in a **proper posterior**

- Variance of random effects σ_{u0}^2: standard 'reference' prior will give an **improper posterior** distribution since $\sigma_{u0}^2 = 0$, is supported by non-negligible likelihood
  - A number of alternatives have been suggested (see next slide)

Priors on random effects variances

- Gamma(ϵ, ϵ), with ϵ small and positive, is 'just proper' form of reference prior
  - $1/\sigma^2_{u0} \sim \text{Gamma}(0.001, 0.001)$ (i.e. just proper gamma prior for the random effects *precision*) is often used, as it also has nice conjugacy properties with the Normal distribution for the random effects
  - But inference may still be sensitive to choice of ϵ
    - $*$ sensitivity particularly a problem if data (likelihood) supports small values of σ^2 (i.e. little evidence of heterogeneity between units)
    - $*$ See Gelman (2005) for further discussion

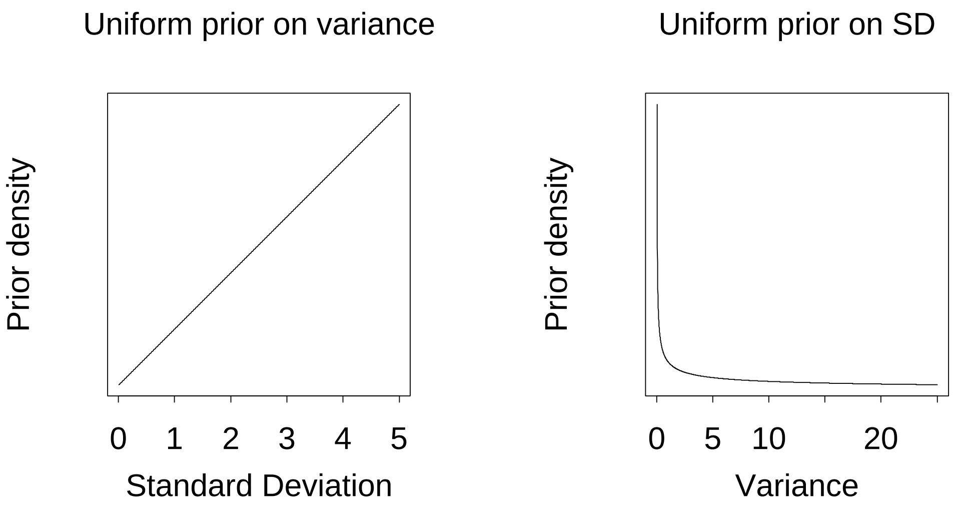


Some different gamma(a, b) priors for the precision, shown on the scale of the standard deviation

- Uniform priors over a finite range on the variance or standard deviation, e.g.

$$\sigma^2_{u0} \sim \text{Uniform}(0,1000); \quad \sigma_{u0} \sim \text{Uniform}(0,1000)$$

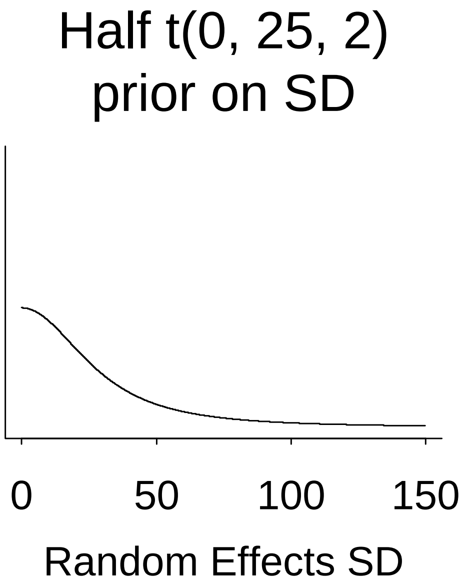
  Appropriate upper bound will depend on scale of measurement of random effects



- Half-normal or half-t on standard deviation, e.g.

  $$\sigma_{u0} \sim \text{Normal}(0, 100) I(0,)$$

  Note, value chosen for variance of half-normal or half-t will depend on scale of measurements for continuous data



Sensitivity analysis plays a crucial role in assessing the impact of particular prior distributions on the conclusions of an analysis.

Bayesian vs ‘classical’ multilevel models

Differences in specification

Bayesian approach

- *all* unknown quantities are treated as random variables, and so must be assigned prior probability distributions

Classical approach

- only the random effects are treated as random variables and given a probability distribution
- other parameters (e.g. regression coefficients, variances) are treated as fixed (but unknown)

Differences in estimation

Bayesian approach

- Inference based on posterior distribution
- Obtained by multiplying together likelihood and priors
- Resulting joint posterior then summarised to give e.g.
  - marginal posterior distribution of particular parameter
  - expected value of posterior distribution (point estimate of parameter)
  - probability that parameter lies within particular interval

→ need to **integrate** joint posterior distribution

→ except in simple cases, need to use simulation based methods (Markov chain Monte Carlo integration)

Classical approach

- Inference based on likelihood (or approximation to it)
- Random effects usually treated as nuisance parameters that do not explicitly appear likelihood
- Requires iterative algorithm to obtain maximum likelihood estimators
- Interval estimates based on assumed asymptotic normality of likelihood

For large samples (units and observations per unit), classical and Bayesian (with vague priors) methods typically give similar results

Bayesian approach generally better for small samples (no need for approximations or asymptotics; weakly informative priors can help stablise model)

Session 3. Bayesian computation and WinBUGS

In this session we will cover

- Monte Carlo and Markov chain Monte Carlo (MCMC) simulation methods
- Interpreting the output from an MCMC simulation
- Introduction to WinBUGS

Why is computation important?

- Bayesian inference centres around the posterior distribution

  $$p(\theta, \phi | x) \propto p(x | \theta, \phi) \times p(\theta, \phi)$$

  where θ is of interest, ϕ is nuisance

- $p(x|\theta, \phi)$ and $p(\theta, \phi)$ will often be available in closed form, but $p(\theta, \phi|x)$ is usually not analytically tractable, and we want to
  - obtain marginal posterior $p(\theta|x) = \int p(\theta, \phi|x)\ d\phi$
  - calculate properties of $p(\theta|x)$, such as mean, tail areas etc.

→ numerical integration becomes vital

Example: a Monte Carlo approach to estimating tail-areas of distributions

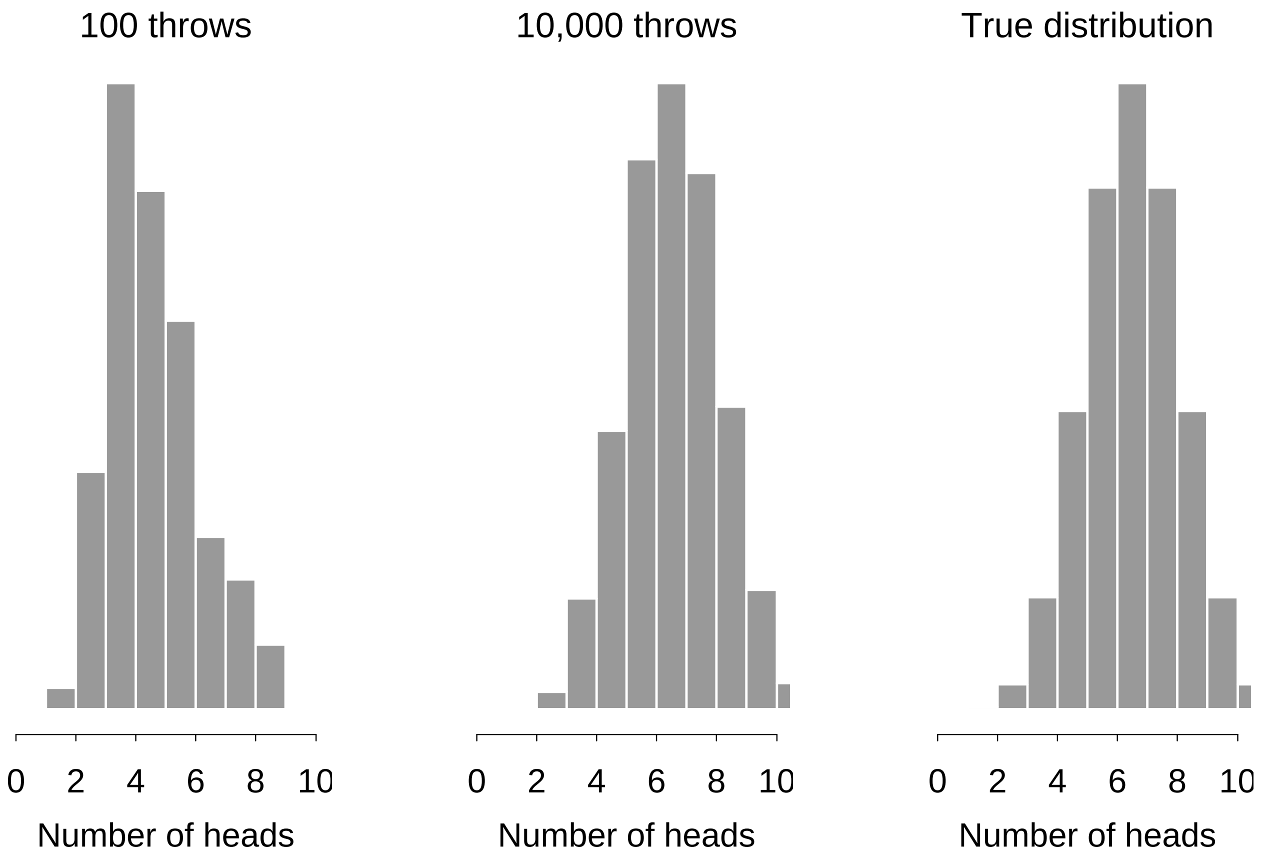
Suppose we want to know the probability of getting 8 or more heads when we toss a fair coin 10 times.

An *algebraic* approach:

$$
\begin{aligned}
\Pr(\geq 8 \text{ heads}) &= \sum_{z=8}^{10} p\left(z|\pi = \frac{1}{2}, n = 10\right) \\
&= \begin{pmatrix} 10 \\ 8 \end{pmatrix} \left(\frac{1}{2}\right)^{8} \left(\frac{1}{2}\right)^{2} + \begin{pmatrix} 10 \\ 9 \end{pmatrix} \left(\frac{1}{2}\right)^{9} \left(\frac{1}{2}\right)^{1} + \begin{pmatrix} 10 \\ 10 \end{pmatrix} \left(\frac{1}{2}\right)^{10} \left(\frac{1}{2}\right)^{0} \\
&= 0.0547.
\end{aligned}
$$

A *physical* approach would be to repeatedly throw a set of 10 coins and count the proportion of throws that there were 8 or more heads.

A *simulation* approach uses a computer to toss the coins!



Proportion with 8 or more ’heads’ in 10 tosses:

(a) After 100 ’throws’ (0.017); (b) after 10,000 throws (0.0577); (c) the true Binomial distribution (0.0547)

General Monte Carlo Integration

Standard software packages such as Splus, R have in-built algorithms for sampling from binomial and other standard distributions

If we had algorithms for sampling from arbitrary (typically high-dimensional) posterior distributions, we could use Monte Carlo methods for Bayesian estimation:

- Suppose we can draw samples from the joint posterior distribution for (θ, ϕ), *i.e.*

$$(\theta^{(1)}, \phi^{(1)}), (\theta^{(2)}, \phi^{(2)}), ..., (\theta^{(N)}, \phi^{(N)}) \quad \sim \quad p(\theta, \phi | x)$$

- Then
  - $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}$ are a sample from the marginal posterior $p(\theta|x)$
  - $E(g(\theta)) \; = \; \int g(\theta)p(\theta|x)d\theta \; \approx \; \frac{1}{N}\sum_{i=1}^{N} g(\theta^{(i)})$

    → this is Monte Carlo integration

    → theorems exist which prove convergence in the limit as $N \to \infty$ even if the sample is dependent (crucial to the success of MCMC)

How do we sample from the posterior?

- In general, we want samples from the joint posterior distribution $p(\boldsymbol{\theta}|x)$ (where now we use $\boldsymbol{\theta}$ to denote vector of all model parameters, including nuisance parameters)

- *Independent* sampling from $p(\boldsymbol{\theta}|x)$ may be difficult

- **BUT** *dependent* sampling from a *Markov chain* with $p(\boldsymbol{\theta}|x)$ as its stationary (equilibrium) distribution is easier

- A sequence of random variables $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, ...$ forms a Markov chain if

$$\theta^{(i+1)} \quad \sim \quad p(\theta|\theta^{(i)})$$

  i.e. conditional on the value of $\theta^{(i)}$, $\theta^{(i+1)}$ is independent of $\theta^{(i-1)}, ..., \theta^{(0)}$

- Theorems exist which show that

  $\frac{1}{n}\sum_{i=1}^{n} g(\theta^{(i)}) \rightarrow E(g(\theta))$ as $n \rightarrow \infty$

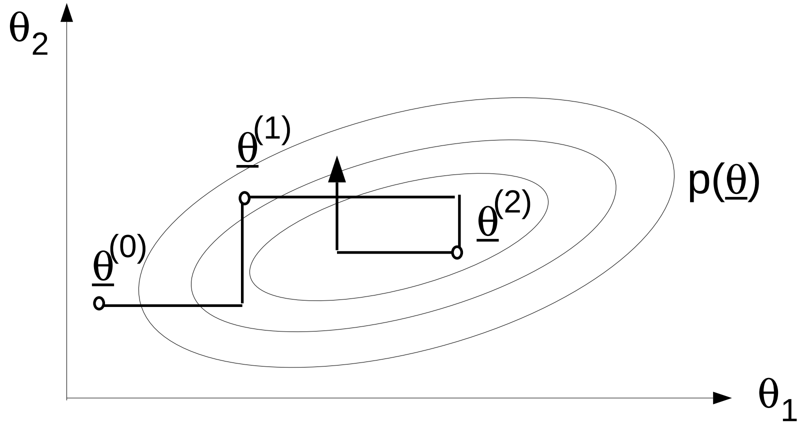
  when $\theta^{(1)}, ..., \theta^{(n)}$ are sampled from a suitable Markov chain

How do we design a Markov chain with $p(\boldsymbol{\theta}|x)$ as its unique stationary distribution?

- This is surprisingly easy and several standard ‘recipes’ are available
- Metropolis *et al.* (1953) showed how to do this
- This method was generalized by Hastings (1970)
- **Gibbs Sampling** (see Geman and Geman (1984), Gelfand and Smith (1990), Casella and George (1992)) is a special case of the Metropolis-Hastings algorithm which generates a Markov chain by sampling from **full conditional distributions**
- See Gilks, Richardson and Spiegelhalter (1996) for a full introduction and many worked examples.

The Gibbs sampler

Suppose we have only two unknown parameters, θ_1 and θ_2



- Sample $\theta_1^{(1)}$ from $p(\theta_1|\theta_2^{(0)}, x)$
- Sample $\theta_2^{(1)}$ from $p(\theta_2|\theta_1^{(1)}, x)$
- Sample $\theta_1^{(2)}$ from $p(\theta_1|\theta_2^{(1)}, x)$
- ......

$\boldsymbol{\theta}^{(n)}$ forms a Markov chain with (*eventually*) a stationary distribution $p(\boldsymbol{\theta}|x)$.

Note, the user needs to provide starting values for the algorithm, $\theta_1^{(0)}$ and $\theta_2^{(0)}$

Performance of MCMC methods

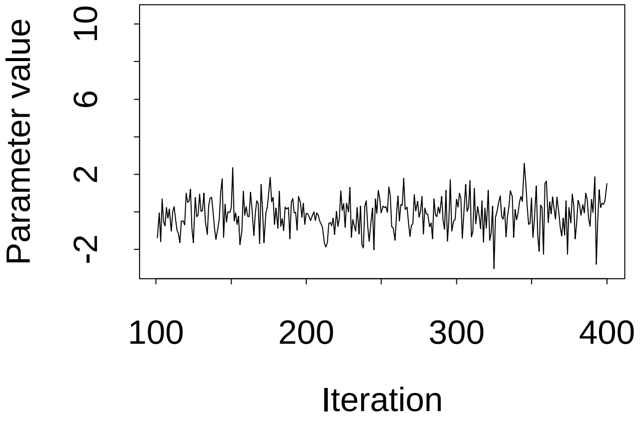
There are two main issues to consider

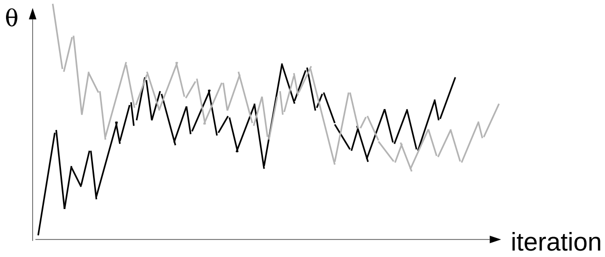
- Convergence (how quickly does the distribution of $\boldsymbol{\theta}^{(t)}$ approach $p(\boldsymbol{\theta}|x)$?)
- Efficiency (how well are functionals of $p(\boldsymbol{\theta}|x)$ estimated from $\{\boldsymbol{\theta}^{(t)}\}$?)

Checking convergence

This is the users responsibility!

- Note: Convergence is to target **distribution** (the required posterior), not to a single value.
- Once convergence reached, samples should look like a random scatter about a stable mean value.





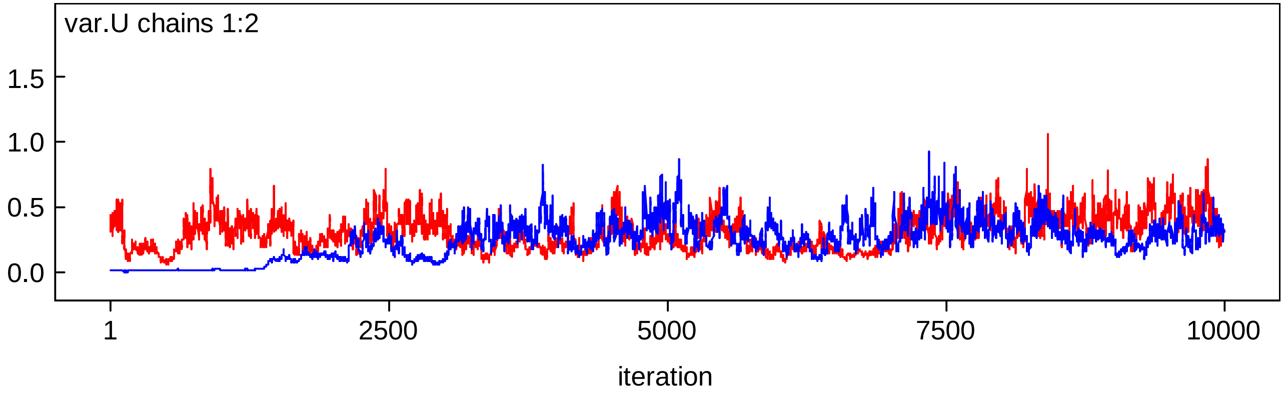
- One approach is to run many long chains with widely differing starting values.

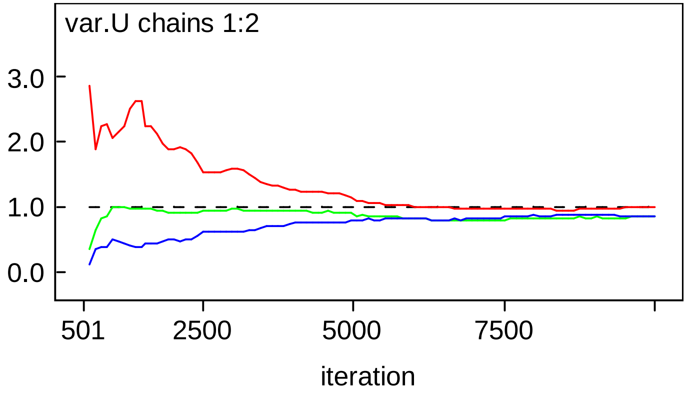
Formally, we can use the Brooks-Gelman-Rubin diagnostic which is based on ratio of between:within chain variances (ANOVA)

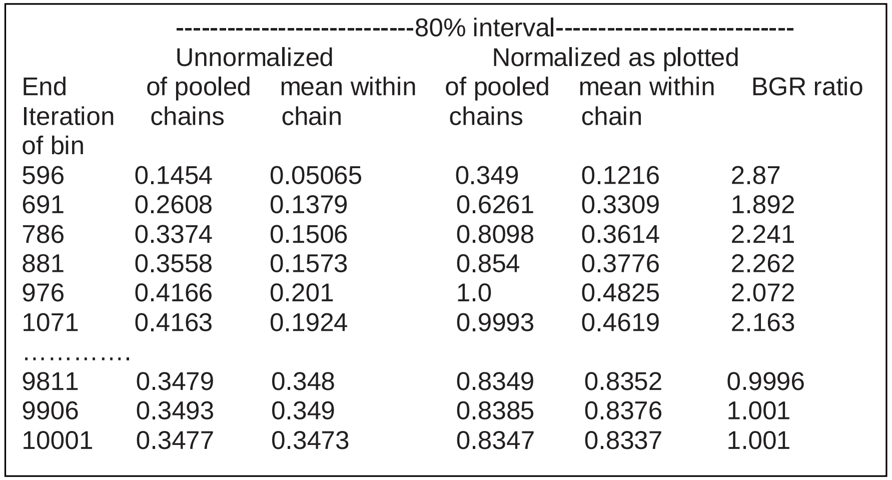
WinBUGS produces plots of:

- Average 80% interval within-chains (blue) and pooled 80% interval between chains (green) — should both converge to stable values
- Ratio pooled:average interval widths (red) — should converge to 1.0
- To print values of the GR-diagnostic: double click on the plot, and press CRTL left-hand-mouse button.

Just one of many potential diagnostics – see, for example, Cowles and Carlin (1996) for further discussion.







	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End Iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
596	0.1454	0.05065	0.349	0.1216	2.87
691	0.2608	0.1379	0.6261	0.3309	1.892
786	0.3374	0.1506	0.8098	0.3614	2.241
881	0.3558	0.1573	0.854	0.3776	2.262
976	0.4166	0.201	1.0	0.4825	2.072
1071	0.4163	0.1924	0.9993	0.4619	2.163
…………					
9811	0.3479	0.348	0.8349	0.8352	0.9996
9906	0.3493	0.349	0.8385	0.8376	1.001
10001	0.3477	0.3473	0.8347	0.8337	1.001

How many iterations after convergence?

- After convergence, further iterations are needed to obtain samples for posterior inference.

- More iterations = more accurate posterior estimates.

- Accuracy of the posterior estimates can be assessed by the Monte Carlo standard error for each parameter (i.e. difference between the mean of the sampled values of the parameter and the true posterior mean):
  - Posterior mean estimated by sample mean $\mathbb{E}(\theta) \approx \frac{1}{n}\sum \theta^{(i)}$
  - If samples were generated independently, could estimate SE of the mean as $\sqrt{S^2/n}$ where $S^2 = \frac{1}{n-1}\sum(\theta^{(i)} - \overline{\theta})^2$ is the sample variance
  - But, this will underestimate the true MC standard error due to autocorrelation in the samples generated using MCMC
  - Various remedies to obtain better estimate of MC error
    - ∗ WinBUGS uses a 'batch means' method — replaces sample variance S^2 by variance of batched means, which are assumed independent
    - ∗ Alternatively, replace actual posterior sample size n in calculation of MC error by 'effective sample size' n/δ where $\delta = 1 + 2\sum_{k=1}^{\infty} \rho(k)$ is the autocorrelation time and $\rho(k)$ is the lag k autocorrelation in the sample of $\theta^{(i)}$'s (see MLwiN)

Inference using posterior samples from MCMC runs

A powerful feature of the Bayesian approach is that all inference is based on the joint posterior distribution

$\Rightarrow$ can address wide range of substantive questions by appropriate summaries of the posterior

- Typically report either mean or median of the posterior samples for each parameter of interest as a point estimate

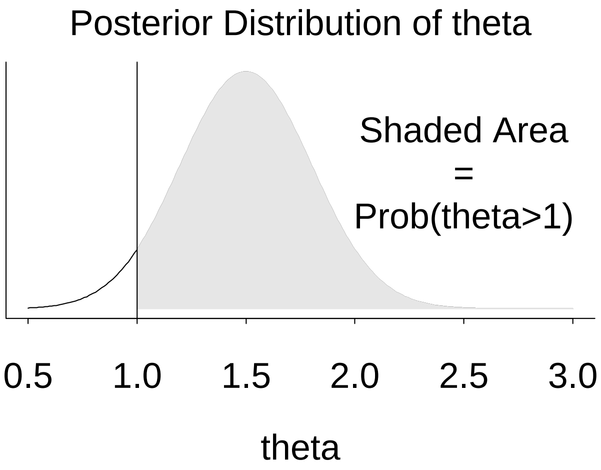
- 2.5% and 97.5% percentiles of the posterior samples for each parameter give a 95% posterior credible interval (interval within which the parameter lies with probability 0.95)

Probability statements about parameters

- Already noted that classical inference cannot provide probability statements about parameters
- In contrast, in Bayesian inference, it is simple to calculate e.g. $\Pr(\theta > 1)$:

  $=$ Area under posterior distribution curve to the right of 1

  $=$ Proportion of values in the posterior sample of `theta` which are > 1



- In `WinBUGS` use the `step` function:

  `p.theta <- step(theta - 1)`

- For discrete parameters, may also be interested in $\Pr(\delta = \delta_0)$:

  `p.delta <- equals(delta, delta0)`

- Posterior means of `p.theta` and `p.delta` give the required probabilities

Complex functions of parameters

- Classical inference about a function of the parameters $g(\theta)$ requires construction of a specific estimator of $g(\theta)$
  - not always possible, e.g. attributable risk = function of RR and prob. of exposure

- Easy using MCMC: just calculate required function $g(\theta)$ as a logical node at each iteration and summarise posterior samples of $g(\theta)$

$\Rightarrow$ in `WinBUGS`, include variables representing required functions as extra terms in the model code, and set sample monitors on these functions

Bayesian model comparison using the Deviance Information Criterion

- Natural way to compare models is to use criterion based on trade-off between the fit of the data to the model and the corresponding complexity of the model
- Spiegelhalter et al (2002) proposed a Bayesian model comparison criterion based on this principle:

  Deviance Information Criterion, DIC = 'goodness of fit' + 'complexity'

- They measure fit via the deviance

$$D(\theta) = -2 \log L(\text{data}|\theta)$$

- Complexity is measured by an estimate of the 'effective number of parameters', defined as

$$\begin{aligned} p_D &= E_{\theta|y}[D] - D(E_{\theta|y}[\theta]) \\ &= \overline{D} - D(\overline{\theta}); \end{aligned}$$

  i.e. posterior mean deviance minus deviance evaluated at the posterior mean of the parameters

- The DIC is then defined analagously to AIC as

$$\begin{aligned} \text{DIC} &= D(\overline{\theta}) + 2p_D \\ &= \overline{D} + p_D \end{aligned}$$

  Models with smaller DIC are better supported by the data

- DIC can be monitored in `WinBUGS` from `Inference/DIC` menu

Residual diagnostics for Bayesian models

Can adapt standard ideas, such as

- *residuals*: plot versus covariates or fitted values, checks for autocorrelation, distributional shape, outliers,....

- *prediction*: compare predictions with external validation set, or cross-validation

e.g. *standardised Pearson residuals*

$$r_i = \frac{x_i - \mathbb{E}(x_i)}{\sqrt{\mathsf{Var}(x_i)}}$$

where $\mathbb{E}(x_i)$ and $\mathsf{Var}(x_i)$ are functions of parameters

Key difference with Bayesian methods is that parameters are unknown quantities with distributions, so residuals also have posterior distribution.

Diagnostics for hierarchical models still area on ongoing research
→ various approaches proposed based on predictive methods (see O'Hagan (2003); Gelman et al (2004); Marshall and Spiegelhalter (2003))

The BUGS program

Bayesian inference Using Gibbs Sampling

- Language for specifying complex Bayesian models
- Constructs object-oriented internal representation of the model
- Builds up an arbitrarily complex model through specification of local structure
- Simulation from full conditionals using Gibbs sampling
- Current version (WinBUGS 1.4) runs in Windows, and incorporates a script language for running in batch mode
- 'Classic' BUGS available for UNIX but this is an old version

WinBUGS is freely available from `http://www.mrc-bsu.cam.ac.uk/bugs`

- An open source version of BUGS (called OpenBUGS) is under development, and includes versions of BUGS that run under LINUX (LinBUGS) and that can be run directly from R (BRugs). See `http://www.rni.helsinki.fi/openbugs`

Running WinBUGS

1. Open *Specification tool* and *Update* from *Model* menu, and *Samples* from *Inference* menu.

2. Program responses are shown on bottom-left of screen.

3. Highlight `model` by double-click. Click on *Check model*.

4. Highlight start of data. Click on *Load data*.

5. Click on *Compile*.

6. Highlight start of initial values. Click on *Load inits*.

7. Click on *Gen Inits* if more initial values needed.

8. Click on *Update* to burn in.

9. Type nodes to be monitored into *Sample Monitor*, and click *set* after each.

10. Perform more updates.

11. Type * into *Sample Monitor*, and click *stats* etc to see results on all monitored nodes.

Example: Schools — implemented using WinBUGS

```
# Model description held in file ‘schools-mod.odc’
model {
# Level 1 definition
for(i in 1:N) {     # N = total number of observations
  normexam[i] ~ dnorm(mu[i], tau.e)   # tau.e = PRECISION
  mu[i]<- beta[1] + beta[2] * standlrt[i] + u0[school[i]]
}

# Higher level definitions
for (j in 1:n2) {   # n2 schools
   u0[j] ~ dnorm(0, tau.u0)   # tau.u0 = PRECISION
}

# Priors for regression coefficients (’fixed’ effects)
for (k in 1:2) { beta[k] ~ dnorm(0, 0.000001) }

# Priors for random effects variances
tau.e ~ dgamma(0.001, 0.001);    sigma2.e <- 1/tau.e
sigma.u0 ~ dunif(0, 1000);    sigma2.u0 <- pow(sigma.u0,2); tau.u0 <- 1/sigma2.u0
}
```

- Note `dnorm` parameterised in terms on mean and precision (1/variance)
- Note use of double indexing `u0[school[i]]` — useful for multilevel models with different numbers of level 1 observations per level 2 unit

Data

```
# Data held in file ‘schools-dat.odc’
list(
N= 4059,
n2 = 65,
school = c(1,1,1,1,,...,2,2,,...3,3,,...,65),
standlrt = c(0.619059,0.205802,-1.364576,.....),
normexam = c(0.261324,0.134067,-1.723882,.....)
)
```

Initial values

- Winbugs can automatically generate initial values for the MCMC analysis from the prior distribution of each unknown parameter
- Better to provide a file with reasonable values for all parameters that have been given a vague prior.

```
# initial values held in file ‘schools-in1.odc’
list(beta = c(-1, 1), tau.e = 0.5, sigma2.u0 = 3)

# initial values held in file ‘schools-in1.odc’
list(beta = c(0.6, -2), tau.e = 4, sigma2.u0 = 0.1)

# Then click ‘gen inits’ option in WinBUGS to generate initial values
# for random effects u0[j]
```

Running from ‘scripts’

Once a program is working it is more convenient to use ’scripts’ to carry out a simulation in the background.

```
# Script for running analysis
display(‘log’)
check(‘c:/winbugs/schools-mod’)    # check syntax of model
data(‘c:/winbugs/schools-dat’)     # load data file
compile(2)                         # generate code for 3 simulations
inits(1,‘c:/winbugs/schools-in1’)  # load initial values 1
inits(2,‘c:/winbugs/schools-in2’)  # load initial values 2
gen.inits()                        # generate initial value u0
set(sigma2.e)   # monitor level 1 (residual) variance
set(sigma2.u0)  # monitor level 2 (between schools) variance
set(beta)       # monitor regression coefficients
update(11000)   # perform 11000 simulations
history(*)      # trace plot of samples for each monitored parameter
gr(*)           # Gelman-Rubin diagnostic for convergence
beg(1001)       # Discard first 1000 iterations as burn-in
stats(*)        # Calculate summary statistics for all monitored quantities
density(sigma2.u0)  # Plot posterior distribution of sigma2.u0
```

Some aspects of the BUGS language

- `<-` represents logical dependence, *e.g.* `m <- a + b*x`
- `~` represents stochastic dependence, *e.g.* `r ~ dunif(a,b)`
- Can use arrays and loops

```
 for (i in 1:n){
   r[i] ~ dbin(p[i],n[i])
   p[i] ~ dunif(0,1)
   }
```

- Some functions can appear on left-hand-side of an expression, *e.g.*

```
logit(p[i])<- a + b*x[i]
log(m[i])  <- c + d*y[i]
```

- `mean(p[])` to take mean of whole array, `mean(p[m:n])` to take mean of elements m to n. Also for `sum(p[])`.
- `dnorm(0,1)I(0,)` means the prior will be restricted to the range $(0, \infty)$.

Functions in the BUGS language

- `p <- step(x-.7)` $= 1$ if `x` ≥ 0.7, 0 otherwise. Hence monitoring `p` and recording its mean will give the probability that `x`≥ 0.7.

- `p <- equals(x,.7)` $= 1$ if `x` $= 0.7$, 0 otherwise.

- `tau <- 1/pow(s,2)` sets $\tau = 1/s^2$.

- `s <-   1/ sqrt(tau)` sets $s = 1/\sqrt{\tau}$.

- `p[i,k] <- inprod(pi[], Lambda[i,,k])` sets $p_{ik} = \sum_j \pi_j \Lambda_{ijk}$.

- See 'Model Specification/Logical nodes' in manual for full syntax.

Some common Distributions

Expression	Distribution	Usage
`dbin`	binomial	`r ~ dbin(p,n)`
`dnorm`	normal	`x ~ dnorm(mu,tau)`
`dpois`	Poisson	`r ~ dpois(lambda)`
`dunif`	uniform	`x ~ dunif(a,b)`
`dgamma`	gamma	`x ~ dgamma(a,b)`

NB. The normal is parameterised in terms of its mean and *precision* = 1/ variance = 1/sd^2.

See 'Model Specification/The BUGS language: stochastic nodes/Distributions' in manual for full syntax.

Functions cannot be used as arguments in distributions (you need to create new nodes).

The `WinBUGS` data formats

`WinBUGS` accepts data files in:

1. Rectangular format

```
 n[] r[]
 47  0
148 18
 ...
360 24
END
```

2. S-Plus format:

```
list(N=12,n = c(47,148,119,810,211,196,
          148,215,207,97,256,360),
     r = c(0,18,8,46,8,13,9,31,14,8,29,24))
```

Generally need a ‘list’ to give size of datasets etc.

Calling WinBUGS from other software

- Scripts enable WinBUGS 1.4 to be called from other software
- Interfaces developed for R, Splus, SAS, Matlab
- See `www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml`
- Andrew Gelman's `bugs` function for R is most developed - reads in data, writes script, monitors output etc.
- OpenBUGS site http://mathstat.helsinki.fi/openbugs/ provides an open source version, including BRugs which works from within R.

Further reading

Gelfand and Smith (1990) (key reference to use of Gibbs sampling for Bayesian calculations)

Casella and George (1992) (Explanation of Gibbs sampling)

Brooks (1998) (tutorial paper on MCMC)

Spiegelhalter, Gilks and Richardson (1996) (Comprehensive coverage of practical aspects of MCMC)

Session 4. Generalised Linear Mixed Models (GLMMs)

In this session we will cover specification and implementation of multilevel models for non-normal response data

- Conceptually straightforward to extend idea of multilevel or hierarchical models to situations where response data are not Normally distributed

- Convenient to work within Generalised Linear Models (GLM) framework → extend to Generalised Linear Mixed Models (GLMMs) where 'mixed' implies mixture of *fixed* and *random* effects (although remember that, actually, all parameters are regarded as random variables in the Bayesian paradigm)

Example: Bangladesh — Hierarchical models for binary data

Data

- 1988 Bangladesh Fertility Survey
- 1934 women in N =60 districts
- Response of interest = binary indicator of whether or not each woman was using contraception at time of survey
- Covariates include age, number of children, education, religion, district

Single level logistic regression model accounting for effects of age and district on contraceptive rates

$$\begin{aligned} y_{ij} &\sim \text{Bernoulli}(p_{ij}) \quad (\text{woman } i, \text{ district } j) \\ \text{logit} p_{ij} &= \beta_0 + \beta_1 \text{AGE}_{ij} + \delta_j \end{aligned}$$

- In Bayesian framework, we would also need to specify priors on β_0, β_1 and δ_j $(j = 1, ..., N)$
- Note — δ_j is regarded as an independent effect for district j in this model, so we would typically specify vague independent priors such as $\delta_j \sim \text{Normal}(0, 100000)$ or $\delta_j \sim \text{Uniform}(-1000, 1000)$ for each j (usually with constraint that $\delta_1 = 0$ for identifiability)

Multilevel / hierarchical / GLMM model

- Treating district as a 'fixed' effect ⇒ contraceptive rates are modelled as being completely independent in different districts

- May be more reasonable to assume exchangeability between districts ⇒ contraceptive rates assumed to be similar but not identical in different districts

$$\begin{aligned} y_{ij} &\sim \text{Bernoulli}(p_{ij}) \quad (\text{woman } i, \text{ district } j) \\ \text{logit} p_{ij} &= \beta_0 + \beta_1 \text{AGE}_{ij} + \delta_j \\ \delta_j &\sim \text{N}(0, \sigma^2) \end{aligned}$$

- Again, in Bayesian framework, also need to specify priors on β_0, β_1 and σ^2

Example: Small area disease counts — hierarchical models for count data

Aim: to estimate relative risk of disease in small areas and look for evidence of geographical variation in risk that may indicate presence of environmental risk factor

Data: Counts of cases of childhood leukaemia and population in 873 electoral wards in London:
y_i : observed number of leukaemias in area i,
E_i : expected number of leukaemias in area i, adjusted for age, sex

Parameters
θ_i : underlying relative risk of leukaemia in area i

- Likelihood (sampling variability within area):

$$y_i \sim \text{Poisson}(E_i\theta_i)$$

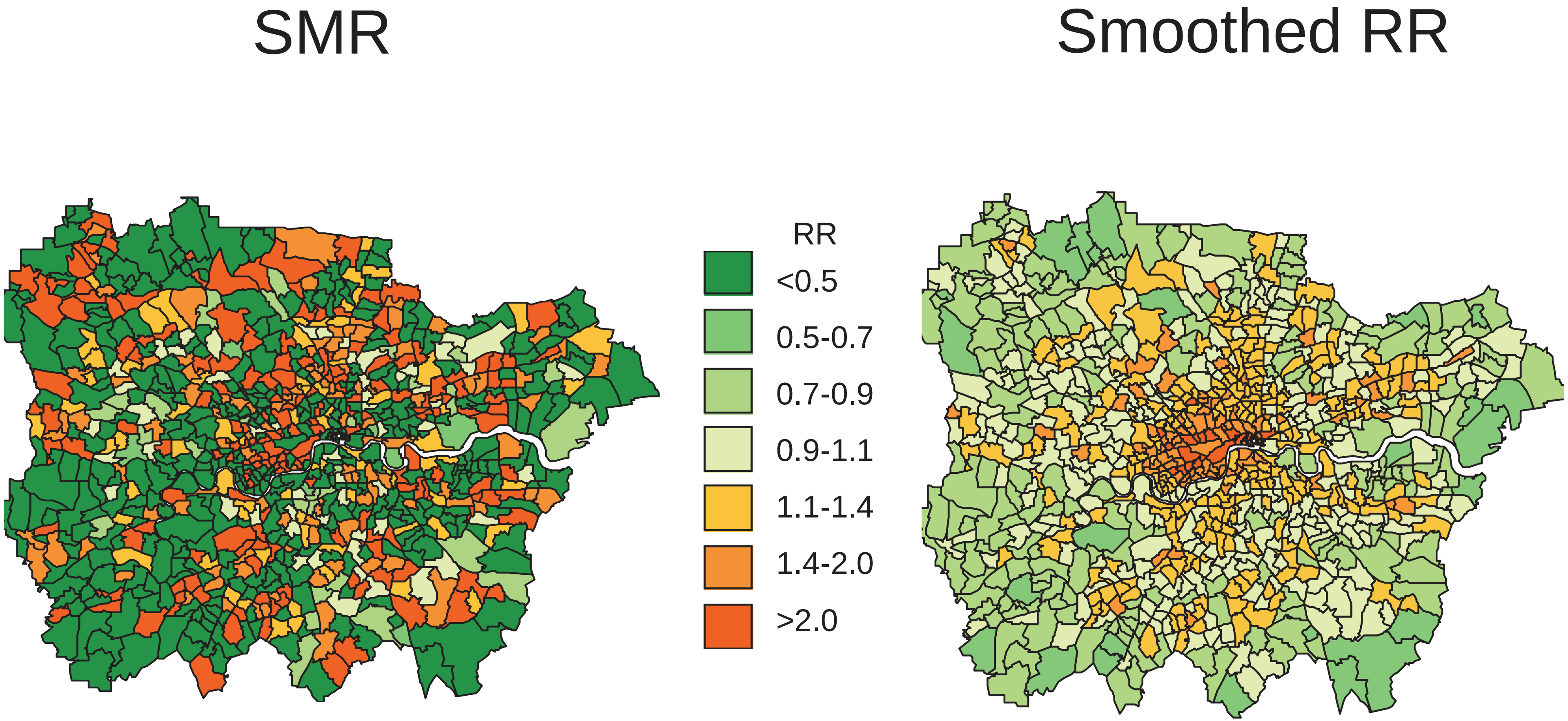
- Exchangeable relative risks across areas:

$$\log\theta_i \sim \text{N}(\mu, \sigma^2)$$

- Hyper-priors

$$\mu \sim \text{Normal}(0, 10000); \quad \sigma^2 \sim \text{Uniform}(0, 10)$$

! Is the assumption of exchangeability reasonable here?


SMR
Smoothed RR
RR
<0.5
0.5-0.7
0.7-0.9
0.9-1.1
1.1-1.4
1.4-2.0
>2.0

Estimation of GLMMs

Bayesian framework

- Same as before — use MCMC to generate samples from exact joint posterior distribution

Classical framework

- Quasi-likelihood (MQL/PQL 1st and 2nd order)
  - Model linearised and IGLS applied
  - MQL1 crudest approximation
    - ∗ Estimates may be biased downwards (esp. if clusters small). But stable.
  - PQL2 best approximation, but may not converge.
    - ∗ Tip: Start with MQL1 to get starting values for PQL.

- Iterative bootstrap

- Quadrature (e.g. PROCNLMIXED in SAS, Stata, aML)

Further reading

WinBUGS examples volumes I and II (lots of examples of Bayesian hierarchical models)

Congdon (2001) (lots of examples of Bayesian hierarchical models)

Gelman et al (2004) Chapters 5, 13, 14

Session 5. More complex hierarchical models

In this session we will briefly cover

- Graphical models as a tool for building complex hierarchical models
- Missing data
- Covariate measurement error
- Multilevel models for variance parameters, and complex variance functions
- Autoregressive models (temporal and spatial)
- Cross-classified and multiple membership models

Graphical Models

Model building

Statistical modelling of complex systems involve usually many interconnected random variables.

How to build the connections ?

Key idea: conditional independence

It is helpful to represent the modelling process by a graph

- nodes: all random quantities
- links (directed or undirected): association between the nodes

Directed edges: natural ordering of association, “causal” influence

Undirected edges: symmetric association, correlation

The graph is used to represent a set of *conditional independence* statements

Independence and Conditional independence

Two variables, X and Y, are *statistically independent* if

$$p(X,Y) \;=\; p(X)\,p(Y).$$

Equivalently, variables X and Y are statistically independent if

$$p(Y \mid X) \;=\; p(Y).$$

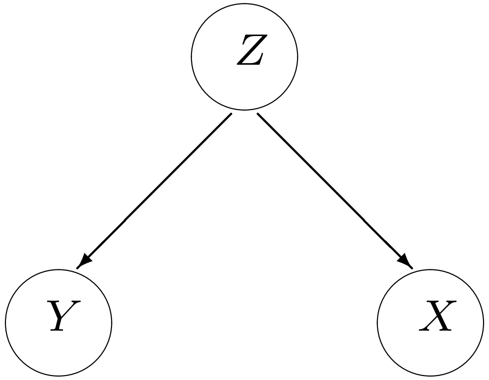
Conditional independence:

Given 3 variables X, Y and Z, we say that X and Y are conditionally independent given Z, denoted by $X \perp\!\!\!\perp Y \mid Z$,

if

$$p(X,Y \mid Z) \;=\; p(X \mid Z)p(Y \mid Z)$$

We can draw this relationship in a *graph*:



Genetic Example:

Consider a family with 2 parents and 2 children.

Let X and Y denote the genotype of the 2 children and Z the genotype of the parents.

If we know the genotypes of the parents, the genotypes of the children are conditionally independent: $X \perp\!\!\!\perp Y \mid Z$

$$p(X, Y \mid Z) \;\; = \;\; p(X \mid Z)p(Y \mid Z)$$

But, if we have no information on the parents, the genotypes of the children are marginally (unconditionally) dependent.

Directed Acyclic Graphs, (DAG)

DAG: set of nodes $V = \{v\} +$ a set of directed edges

- Only contain directed edges
- Used to build models directionally, e.g. disease → symptoms, parameters → data, cause→effect
- We also suppose that there are no directed cycles
  $\Rightarrow$ each node has a well defined set of parents: parents$[v]$ and descendants

The joint distribution associated with the graph is specified by:

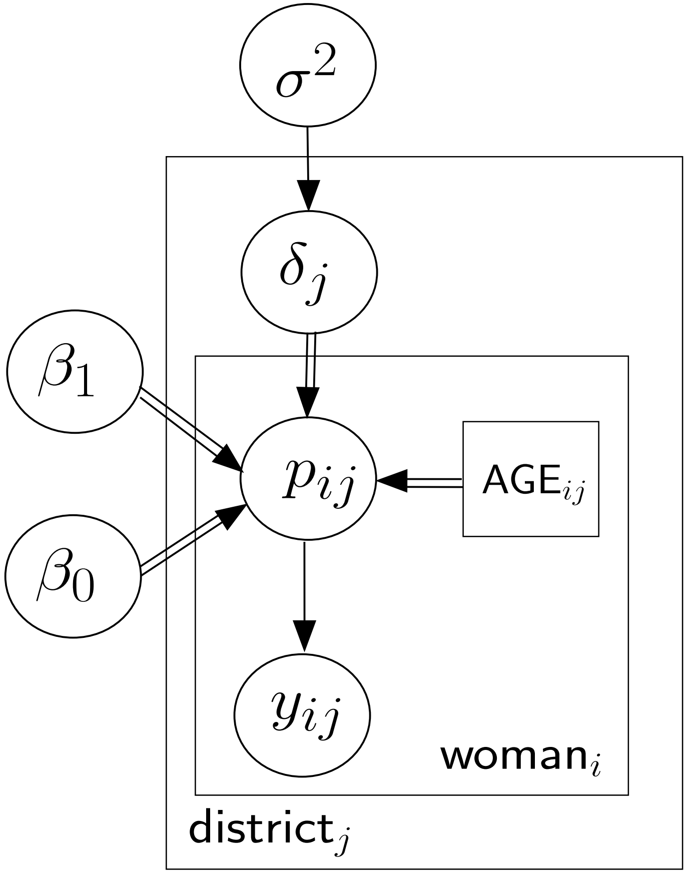
$$p(V) \quad = \quad \prod_{v \epsilon V} p(v \,|\, \text{parents}[v])$$

This is a recursive factorisation that is be extensively used in Bayesian computations for hierarchical models

For the simple genetics example, graph implies the following joint distribution:

$$p(X,Y,Z) = p(X|Z)p(Y|Z)p(Z)$$

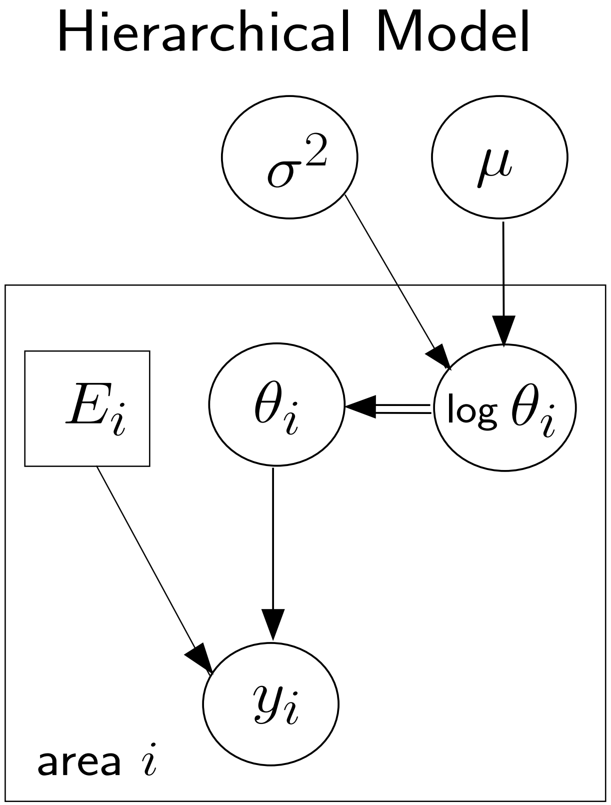
DAG for Bangladesh example (hierarchical model)



Joint distribution:

$$p(\boldsymbol{y}, \boldsymbol{\delta}, \beta_0, \beta_1, \sigma^2) = \prod_{ij} p(y_{ij}|\beta_0, \beta_1, \delta_i, \mathsf{AGE}_{ij}) \ \ \prod_{j} p(\delta_j|\sigma^2) \ \ p(\beta_0)p(\beta_1)p(\sigma^2)$$

DAG for disease mapping model



Joint distribution:

$$p(\boldsymbol{y}, \log\boldsymbol{\theta}, \mu, \sigma^2) = \prod_i p(y_i | \log\theta_i) \;\; \prod_i p(\log\theta_i | \mu, \sigma^2) \;\; p(\mu) p(\sigma^2)$$

Further reading

Spiegelhalter (1998) (Tutorial on Bayesian graphical models)

Spiegelhalter et al (1995) (Discussion of link between graphical models and Bayesian computation)

Richardson and Best (2003) (Use of Bayesian graphical models to build complex models in environmental epidemiology)

Missing data

1. Classical approach:

- Complete case analysis
  - Inefficient since throwing away data
  - Can be biased
- Imputation
  - 'Fill in' missing data with imputed values, then estimate parameters assuming imputed values were actually observed
  - Naive approach: replace missing data by mean of observed responses
    - underestimates true variation in response
    - may be biased
  - Multiple imputation (Rubin, 1978)
    - Generate $K > 1$ sets of imputations
    - Re-estimate model using each 'completed' data set
    - Pool parameter estimates to obtain single estimate
    - Estimate variance by combining within and between-imputation variances

2. Bayesian approach:

- Inference based on joint posterior distribution of the parameters and missing data given the observed data and modelling assumptions
- Using MCMC ⇒ obtain samples of all the unknowns (*i.e.* parameters **and missing data**)

⇒
  - Need to specify prior distribution (or more elaborate prior model) for missing values
    (note — if missing *response* values, likelihood automatically acts as model for missing data)
  - Missing values then 'automatically' imputed at each MCMC iteration
  - Posterior estimates of model parameters will be fully adjusted for uncertainty in the imputed observations (conditional on the assumed model)

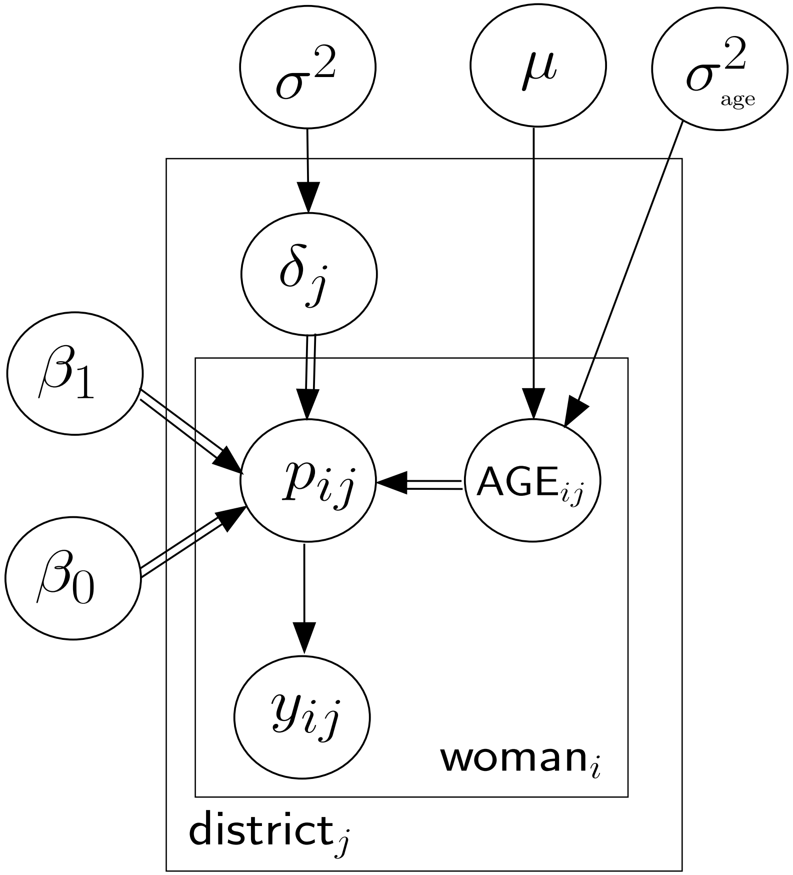
- Missing value code in `BUGS` is `NA`

Note: also possible to fit models for non-ignorable missing data. See Best et al (1996) for an example.

Example: Bangladesh — missing covariate data

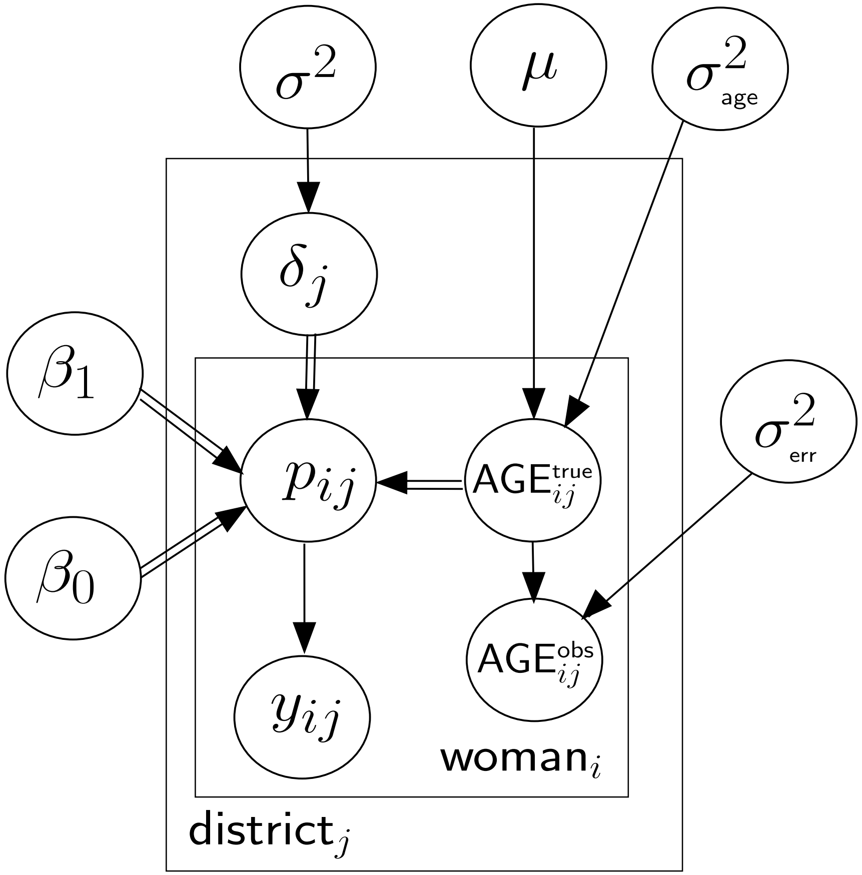
- Suppose age was not recorded for some women

- Assuming that there is no systematic reason why age was not recorded, we might specify a simple Normal prior distribution for the age covariate: $\text{AGE}_{ij} \sim \text{Normal}(\mu, \sigma^2_{\text{age}})$



Classical measurement error

Handled in similar way to missing covariate data, but regress on true value not observed.



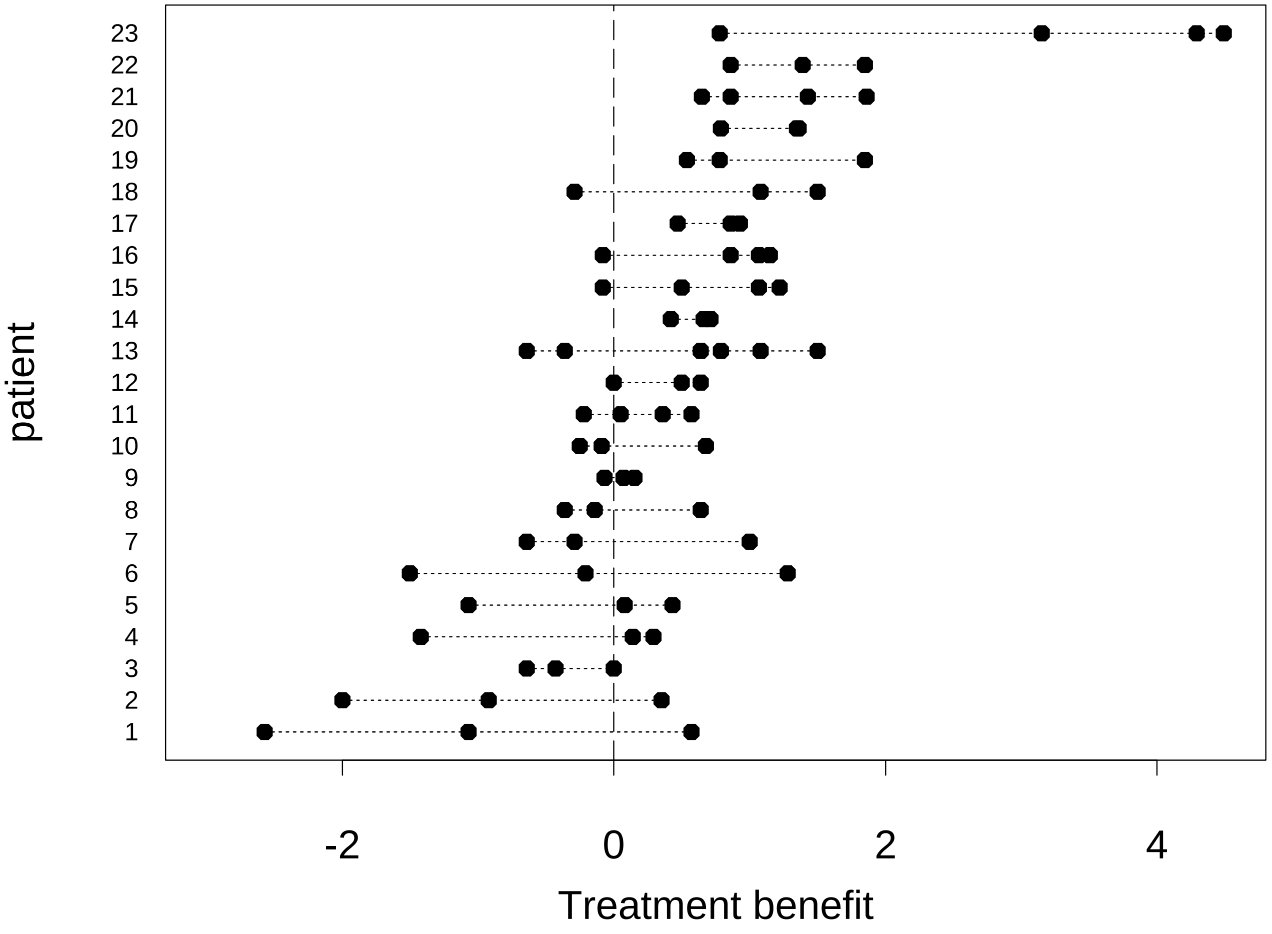
Hierarchical models for variances

Example: N-of-1 trials

Spiegelhalter et al (2004) Example 6.10

- N-of-1 trials → repeated within-person crossover trials
- Often suitable for investigating short-term symptom relief in chronic conditions
- Example:
  - **Intervention:** Amitriptyline for treatment of fibromyalgia to be compared with placebo.
  - **Study design:** 23 N-of-1 studies - each patient treated for a number of periods (3 to 6 per patient), and in each period both amitriptyline and placebo were administered in random order
  - **Outcome measure:** Difference in response to a symptom questionnaire in each paired crossover period. A positive difference indicates Amitriptyline is superior
  - **Evidence from study:** 7/23 experienced benefit from the new treatments in all their periods

Raw data for each patient



Statistical model

If y_{kj} is the j^{th} measurement on the k^{th} individual, we assume

$$y_{kj} \sim \mathsf{N}(\theta_k, \sigma_k^2)$$

Assume both θ_k's and σ_k^2's are *exchangeable*, in the sense there is no reason to expect systematic differences and we act as if they are drawn from some common prior distribution.

Note: alternative assumptions are either that θ_k and σ_k^2 are same for all patients (pooled model) or that they are independent (fixed effects) for each patient

We make the specific distributional assumption that

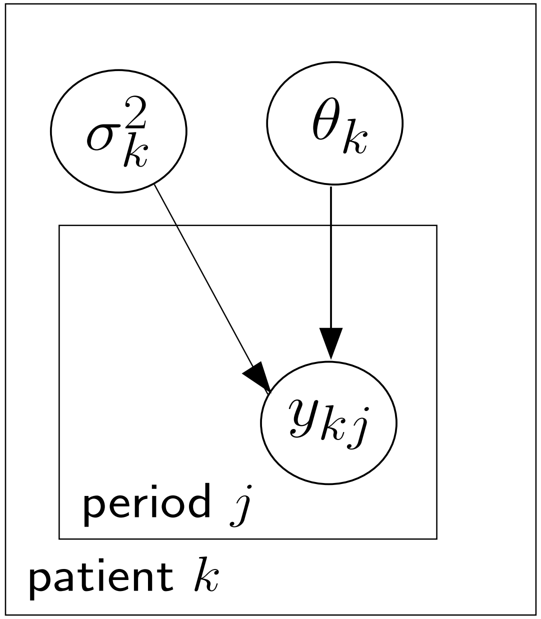
$$\begin{aligned} \theta_k &\sim \mathsf{N}(\mu_\theta, \phi_\theta^2) \\ \log(\sigma_k^2) &\sim \mathsf{N}(\mu_\sigma, \phi_\sigma^2) \end{aligned}$$

A normal distribution for the log-variances is equivalent to a log-normal distribution for the variances

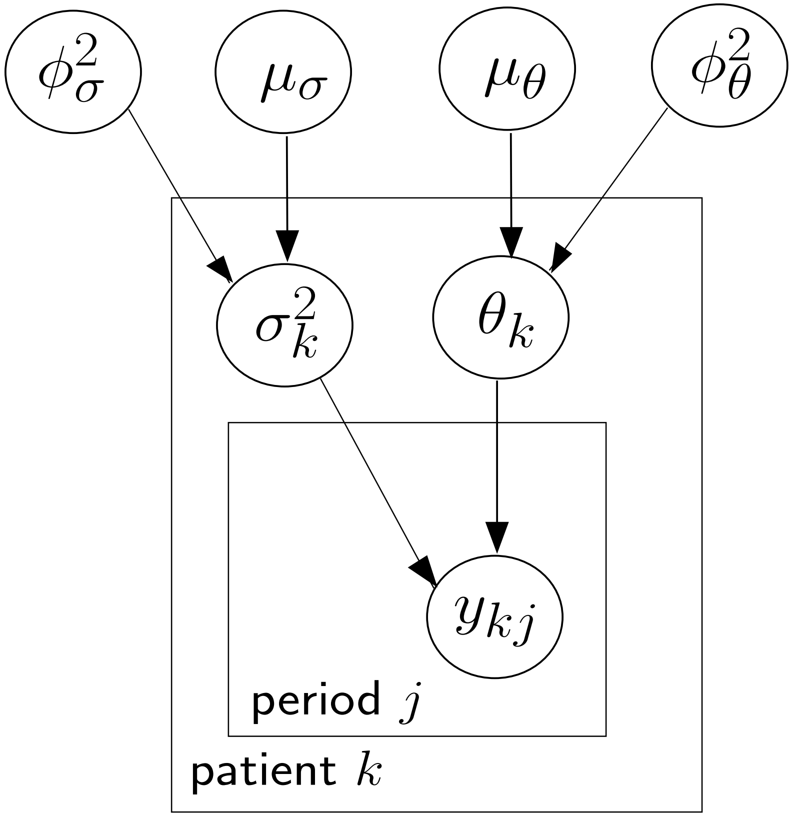
Uniform priors adopted for $\mu_\theta, \phi_\theta, \mu_\sigma$ and ϕ_σ.

Graphical model

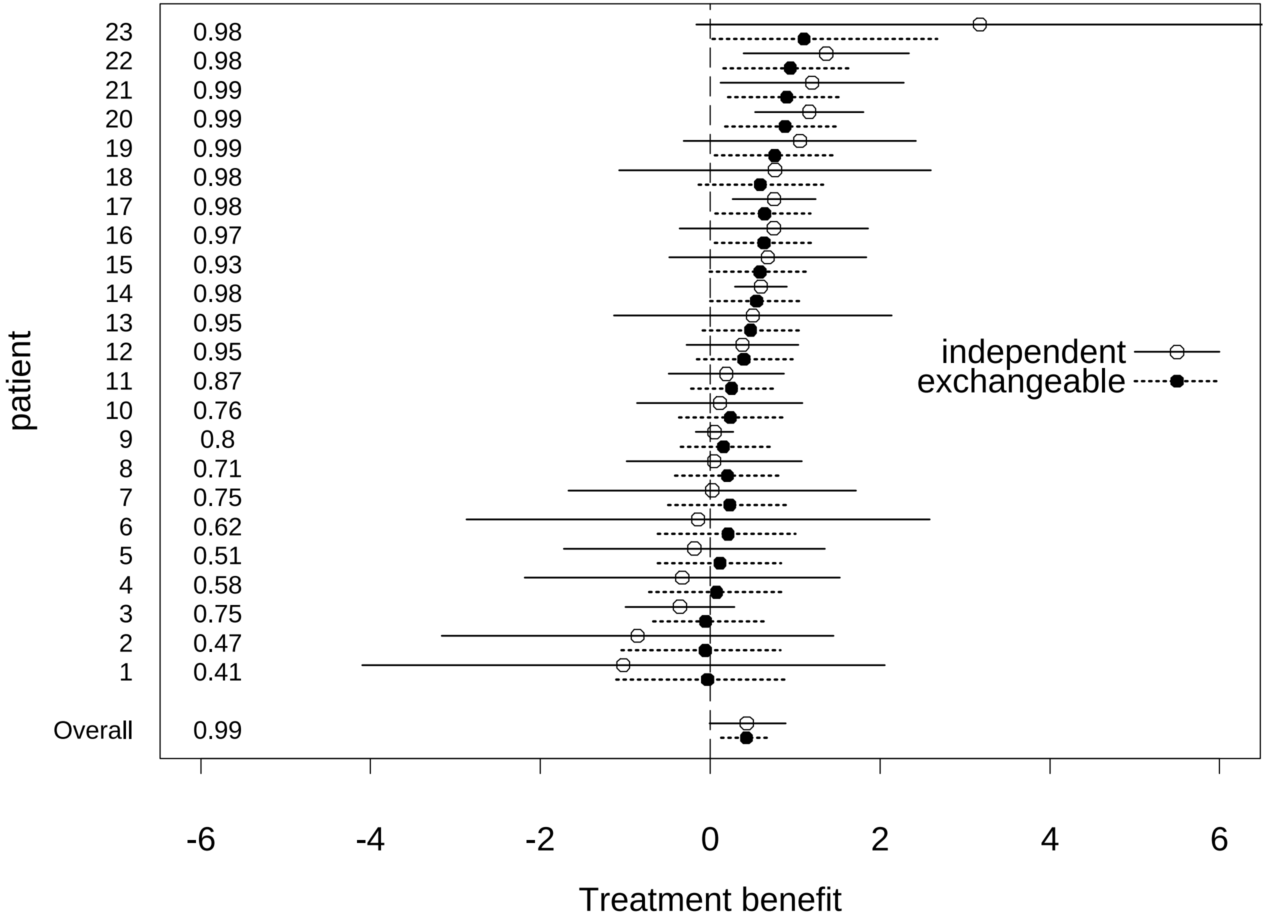
Independent effect



Exchangeable means and variances



Estimates and 95% intervals for treatment effect, and posterior probability that effect > 0



Autoregressive models for temporal dependence

Sometimes necessary to explicitly model temporal dependence of data or of parameters.

- weekly, monthly, quarterly number of reported cases of an infectious disease — often exhibit short term dependence, plus possibly seasonal patterns

- daily / weekly values of economic indicators — necessary to distinguish short term dependence from systematic trends

Autoregressive models of order 1, AR(1)

Let $\boldsymbol{z} = (z_1, \ldots, z_T)$ be a time ordered sequence of observations

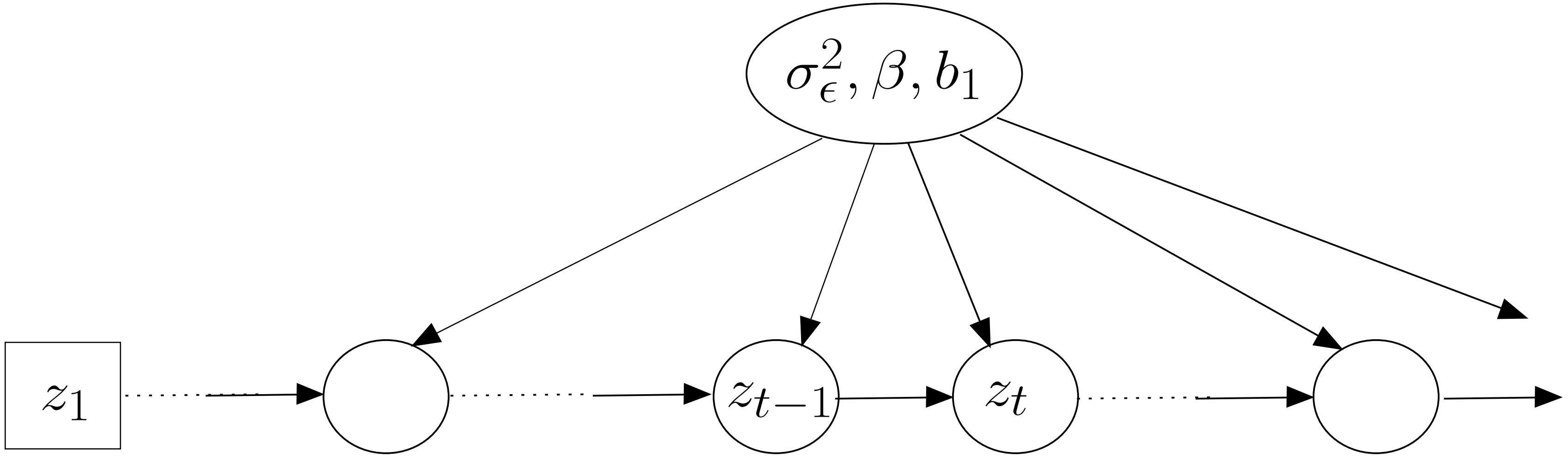
A first order autoregressive Gaussian model for $\boldsymbol{z}$ can be defined as

$$z_t = b_1 z_{t-1} + \epsilon_t; \quad \epsilon_t \sim N(0, \sigma^2_\epsilon)$$

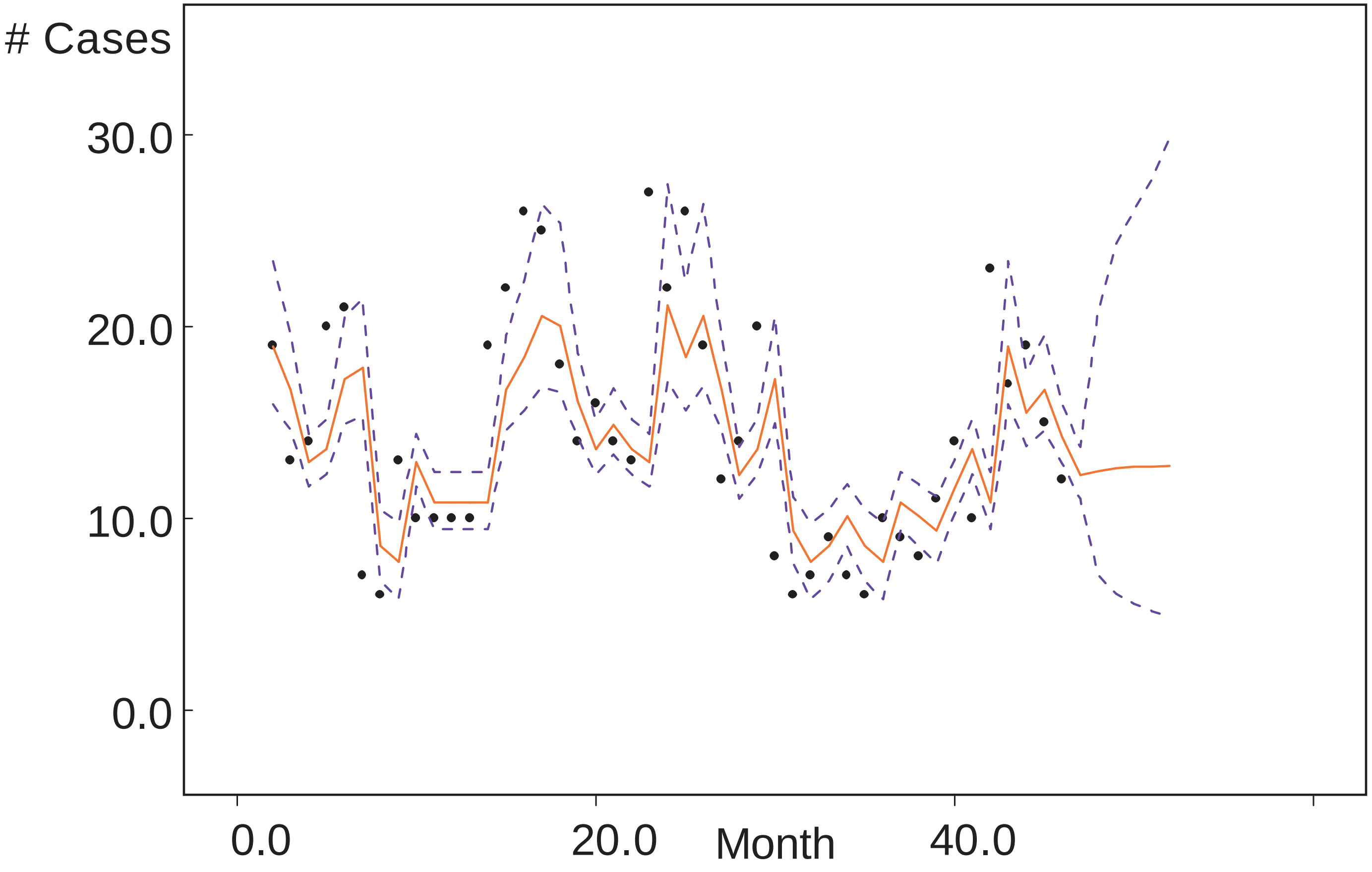
or equivalently

$$z_t \sim N(b_1 z_{t-1}, \sigma^2_\epsilon)$$

DAG of AR(1) model



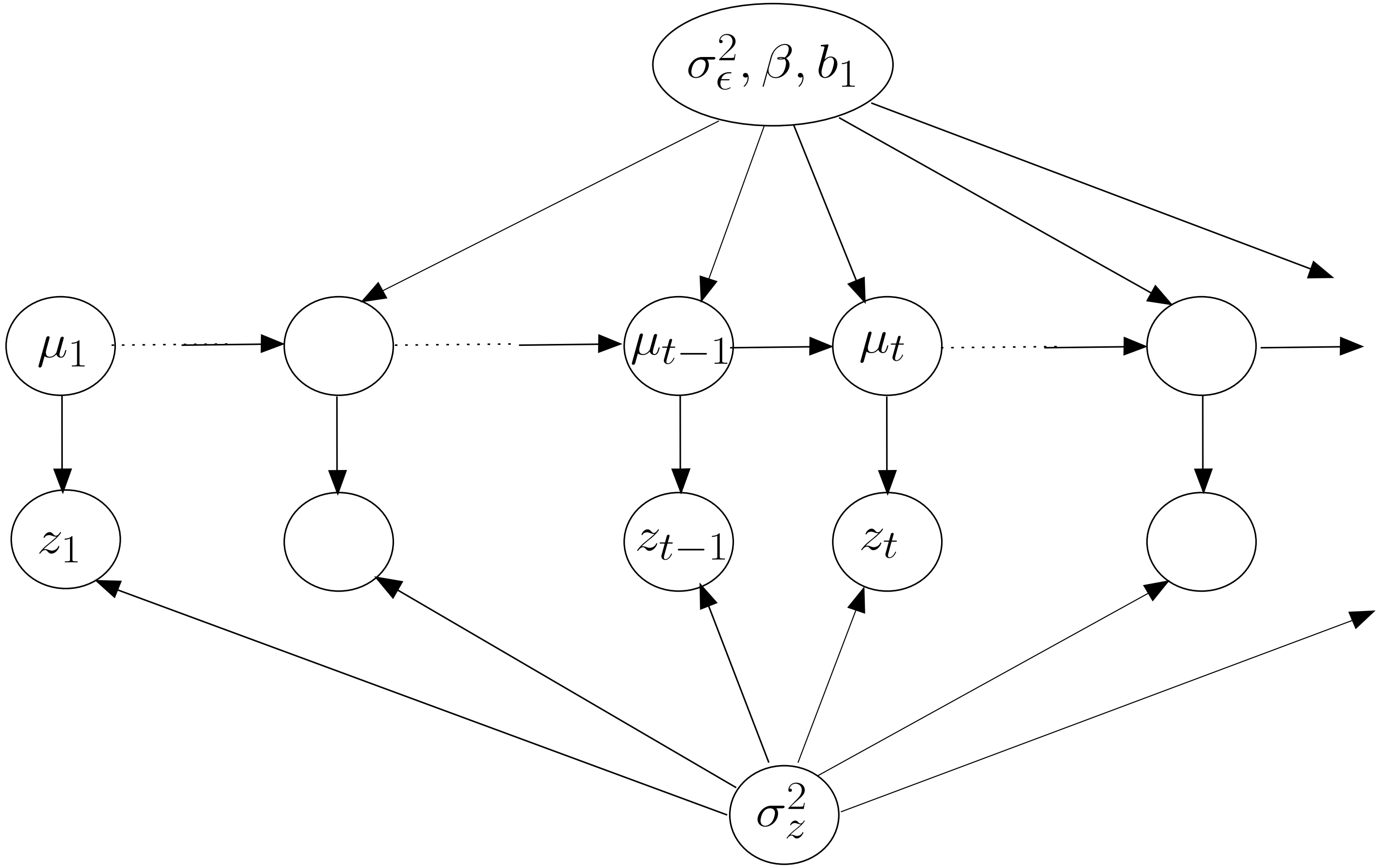
Posterior median and 95% intervals for the estimated true number of disease cases per month (`y.fitted`), plus posterior predictive values for the next 6 months – AR(1) model



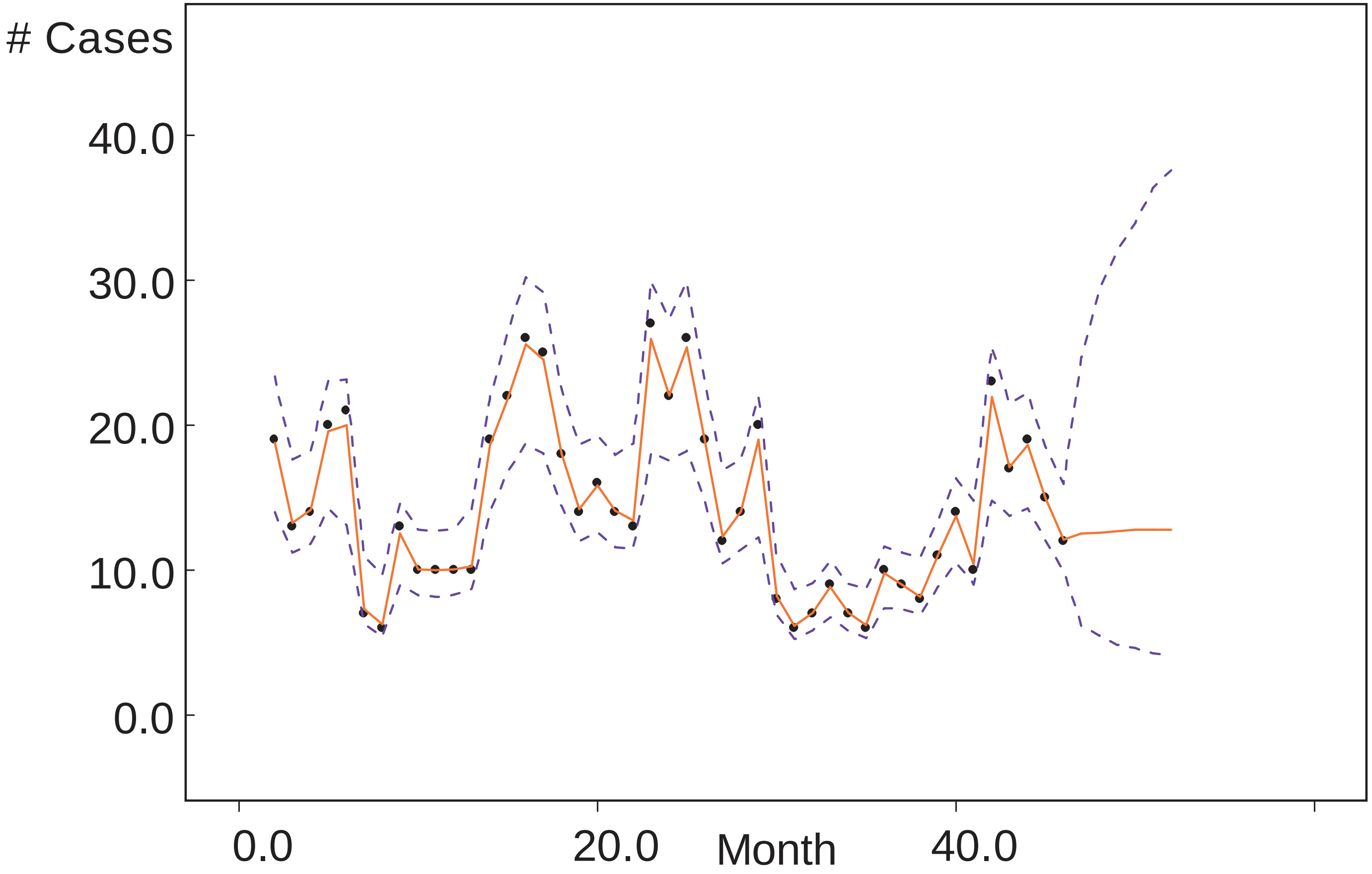
Can easily extend to state-space model — allows for measurement error (and non-Normal sampling likelihoods) on response

$$\begin{aligned} z_t &\sim N(\mu_t, \sigma_z^2) \\ \mu_t &\sim N(b_1\mu_{t-1}, \sigma_\epsilon^2) \end{aligned}$$

DAG for state-space model



Posterior median and 95% intervals for the estimated true number of disease cases per month (`y.fitted`), plus posterior predictive values for the next 6 months – State-space model with independent Gaussian observation process and AR(1) system process



Model	DIC	p_D
AR(1)	36.1	1.7
SS	-41.5	53.3

More complex models in MLwiN

Here we discuss in more detail other models that can be fitted in MLwiN using MCMC, in particular:

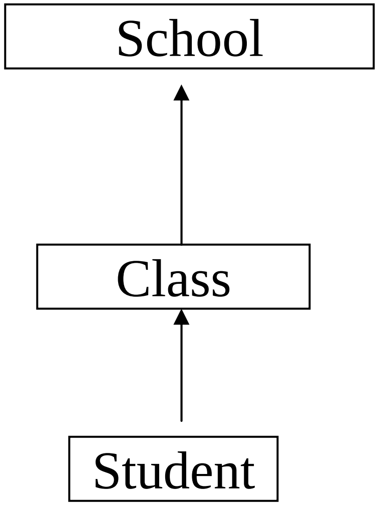
- Cross-classified & Multiple membership models.
- Spatial models.
- Missing data & multiple imputation.
- Measurement error models.
- Complex variance functions.

Cross-classified and multiple membership models

- The basic (nested) multilevel model can be easily extended (using MCMC) in two directions: (See Browne (2003) chapters 14 & 15)
- **Cross-classified models**: Here we remove the restriction that higher classifications are nested e.g. students nested in a crossing of schools and neighbourhoods
- **Multiple membership models**: Here an observation may be affected by more than one higher level unit in a classification e.g. students moving school during the period of their schooling.

Classification diagrams

A useful way of conveying the underlying structure of the dataset is with the ***classification diagram***. This has one node per classification and nodes linked by arrows have a nested relationship and unlinked nodes have a crossed relationship. (Note that classification diagrams are slightly different from DAGs in that they do not formally represent conditional independence assumptions)

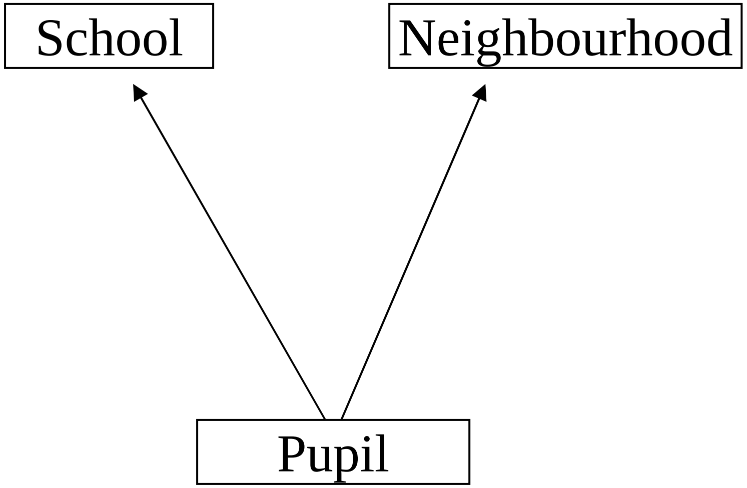


School

Class

Student

Nested structure where classes are nested within schools.



School

Neighbourhood

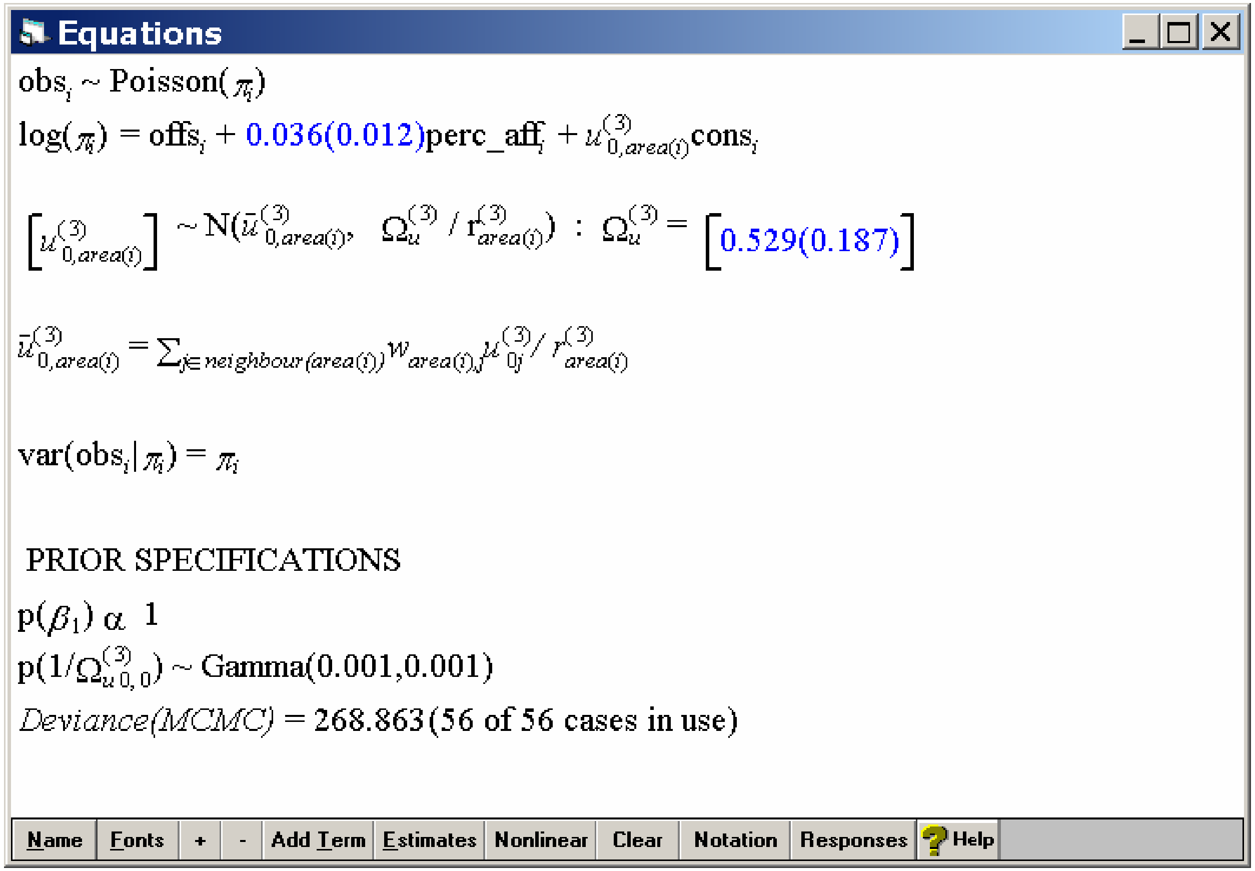
Pupil

Cross-classified structure where pupils from a school come from many neighbourhoods and people from a neighbourhood attend several schools.

Spatial Models

- Multiple membership models can be used to model spatial variation i.e. neighbouring region can be treated as an additional multiple membership classification.
- MLwiN can also fit CAR distributed residuals for spatial models.
- For further details see Browne(2003) chapter 16.

CAR Model



$\text{obs}_i \sim \text{Poisson}(\pi_i)$

$\log(\pi_i) = \text{offs}_i + 0.036(0.012)\text{perc\_aff}_i + u^{(3)}_{0,area(i)}\text{cons}_i$

$\left[u^{(3)}_{0,area(i)}\right] \sim \text{N}(\bar{u}^{(3)}_{0,area(i)},\ \Omega^{(3)}_u \,/\, \text{r}^{(3)}_{area(i)})\ :\ \Omega^{(3)}_u = \left[0.529(0.187)\right]$

$\bar{u}^{(3)}_{0,area(i)} = \sum_{j\in neighbour(area(i))} w_{area(i),j}\, u^{(3)}_{0j} / r^{(3)}_{area(i)}$

$\text{var}(\text{obs}_i|\pi_i) = \pi_i$

PRIOR SPECIFICATIONS

$\text{p}(\beta_1)\ \alpha\ 1$

$\text{p}(1/\Omega^{(3)}_{u\,0,0}) \sim \text{Gamma}(0.001,0.001)$

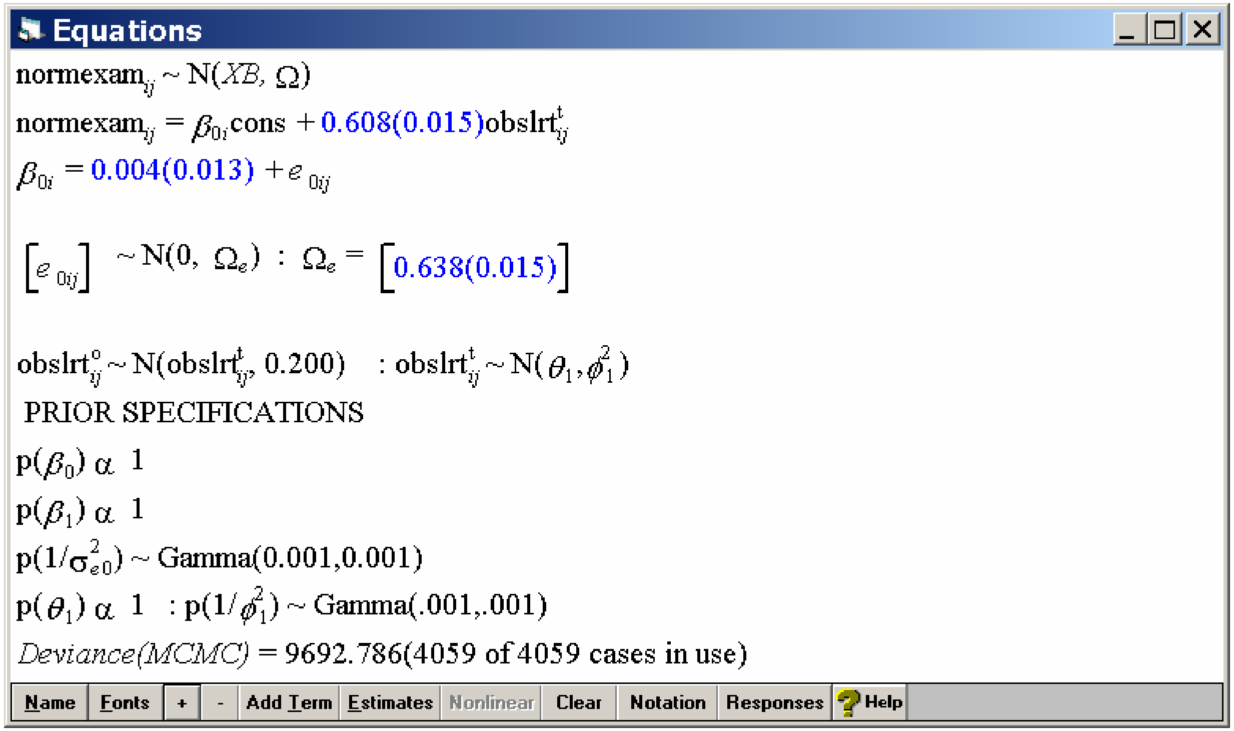
$Deviance(MCMC) = 268.863$(56 of 56 cases in use)

Missing Data & Multiple Imputation

- MLwiN can fit multivariate Normal response models with missing responses (See Browne 2003 chapter 17)
- It can also fit mixtures of Normal and Binomial responses using the probit link function and latent variables (See Browne 2003 chapter 18)
- The MCMC methods involve generating values for the missing data at each iteration and this means they can also be used to generate multiple imputation datasets.
- James Carpenter has written MLwiN macros to perform multiple imputation for (continuous) missing data (both responses and predictors). See http://www.missingdata.org.uk/ for more details.

Measurement Errors

- MLwiN can accommodate (known) measurement errors in predictors (see Browne 2003 chapter 13).



$\text{normexam}_{ij} \sim \text{N}(XB,\ \Omega)$

$\text{normexam}_{ij} = \beta_{0i}\text{cons} + 0.608(0.015)\text{obslrt}^{t}_{ij}$

$\beta_{0i} = 0.004(0.013) + e_{0ij}$

$\begin{bmatrix} e_{0ij} \end{bmatrix} \sim \text{N}(0,\ \Omega_e)\ :\ \Omega_e = \begin{bmatrix} 0.638(0.015) \end{bmatrix}$

$\text{obslrt}^{o}_{ij} \sim \text{N}(\text{obslrt}^{t}_{ij},\ 0.200)\quad :\ \text{obslrt}^{t}_{ij} \sim \text{N}(\theta_1, \phi^2_1)$

PRIOR SPECIFICATIONS

$\text{p}(\beta_0)\ \alpha\ 1$

$\text{p}(\beta_1)\ \alpha\ 1$

$\text{p}(1/\sigma^2_{e0}) \sim \text{Gamma}(0.001,0.001)$

$\text{p}(\theta_1)\ \alpha\ 1\ \ :\ \text{p}(1/\phi^2_1) \sim \text{Gamma}(.001,.001)$

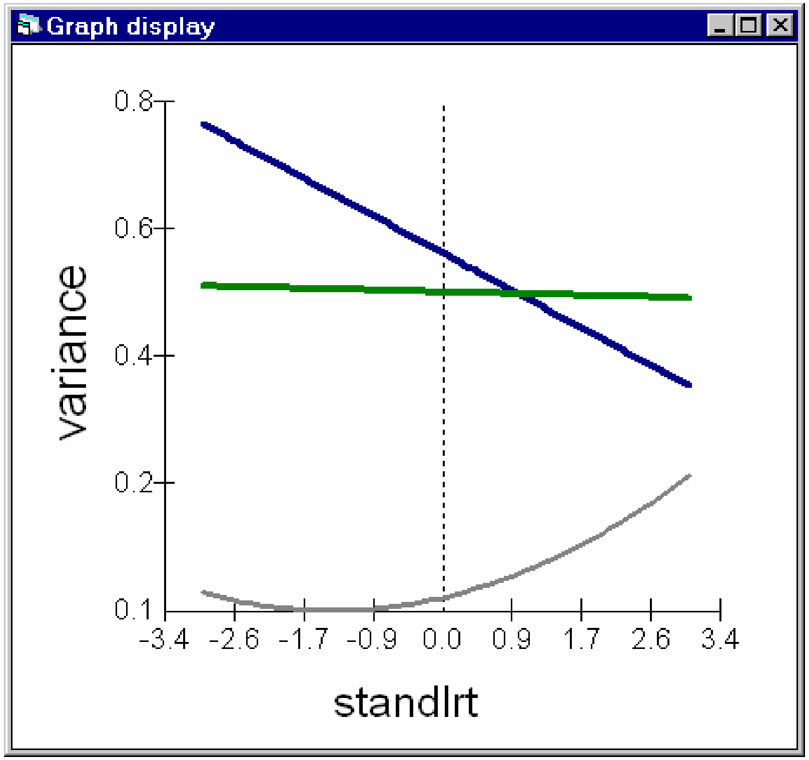
Deviance(MCMC) = 9692.786(4059 of 4059 cases in use)

Complex variance functions

- MLwiN allows for heteroskedasticity in a response i.e. the variance is a function of predictor variables (See Browne 2003 chapter 10)
- This can be fitted in both IGLS and MCMC.
- Complex variability is allowed at higher levels through for example random slopes models.

Complex variance functions

- Here we show partitioning the complex variance structure by level, intake score and gender. Grey represents the school level, blue boys at level 1 and green girls at level 1.



Summary Session

What have we covered?

- Normal response multilevel models
- Binary response multilevel models
- IGLS & MCMC estimation in MLwiN
- MCMC estimation in WinBUGS
- The MlwiN->WinBUGS interface
- Graphical models
- More complex hierarchical / multilevel models

MLwiN Summary

- MLwiN is a statistics package specifically designed for multilevel models
- It has two main estimation engines, the IGLS engine and the MCMC engine.
- It can also do bootstrapping using the IGLS engine
- It has many additional screens specifically designed to deal with features of multilevel models e.g. residual screens, hierarchy viewer
- It also has a macro language to allow users to run simulation studies.

WinBUGS Summary

- WinBUGS is a general purpose MCMC estimation engine with a Windows interface that allows model specification and some additional graphical summary tools.
- It can fit a far greater number of models than MLwiN.
- It can be embedded in other software, see for example Backbugs, the OpenBUGS project, BUGSXLA and BRUGS, R2WinBUGS.
- The experienced user can choose from many possible MCMC methods for their model.
- It allows the user to run multiple chains on the same model.

MLwiN – The IGLS engine

- IGLS (RIGLS) engine fits normal response multilevel models giving (restricted) maximum likelihood estimates.
- Macros incorporated in the software allow the IGLS engine to be used on other response types: Binomial, Poisson and Ordered/Unordered multinomial. Here the macro produces quasilikelihood (MQL/PQL) estimates.
- Other macros allow the IGLS algorithm to fit cross-classified and multiple membership models as constrained nested models.
- Other macros allow time series structures and correlated residuals to be fitted.

MLwiN – MCMC engine

- Developed later in MLwiN as an alternative to IGLS/RIGLS.
- By default using Gibbs sampling when the conditional posterior has a standard form or else uses MH sampling.
- Can fit all the response types that IGLS fits.
- In addition fits cross-classified and multiple membership models more efficiently.
- Can also fit models with measurement errors in predictors, missing responses and spatially dependent (CAR) random effects.
- Also fits multilevel factor analysis and correlated residuals (but only for balanced multivariate data).

WinBUGS

- Seemingly endless choice of responses!
- Considerable flexibility over choice of priors
- Can fit any model that can be expressed as a directed graph.
- Can fit some graphs with undirected links, e.g. CAR spatial residuals, by treating sets of parameters as nodes in the graph.
- Has extensive spatial modelling features and a mapping tool called GeoBUGS.
- Incorporates many MCMC techniques including AR sampling, slice sampling, conjugate Gibbs sampling and MH sampling.

Differences in approach - MLwiN

- MLwiN’s software development has focussed on an incremental approach.
- Just as multilevel models can be considered as an extension of linear models so other features have been bolted on!
- By focussing on families of models we can also consider related issues e.g. sample size calculations, graphical displays, residual and outlier analysis that are related to the particular model.
- The software is also aimed at social/medical scientists.
- In this way the estimation engines are only part of the whole package.

Differences in approach - WinBUGS

- WinBUGS has a much more a ‘the skies the limit’ approach!
- It has had some incremental developments in that the number of possible MCMC samplers has increased over time allowing more models to be fitted.
- It is mainly an estimation engine for fitting models and as such has attracted a more technically able user base.
- The advent of several interfaces from other packages, and more extensive documentation is likely to mean WinBUGS is more accessible to social/medical scientists.

Advantages/Disadvantages

- IGLS estimation is far quicker and for Normal response models gives good estimates.
- Model comparison is also easier with formal test statistics.
- MCMC in MLwiN is almost always faster than in WinBUGS.
- However the MCMC algorithms may be less efficient and the range of models is greatly reduced.
- Having two independent MCMC algorithms for fitting some models in common is a boon as programmers are not infallible!

Obtaining the software

WinBUGS is freely available from

http://www.mrc-bsu.cam.ac.uk/bugs

Information on multilevel modelling and obtaining MLwiN is available from

http://multilevel.ioe.ac.uk/index.html

References and further reading

Berry, DA (1996). *Statistics: A Bayesian Perspective*, Duxbury, London.

Best, NG, Spiegelhalter, DJ, Thomas, A and Brayne, CEG (1996). Bayesian analysis of realistically complex models. *J R Statist Soc A*, **159**, 323–342.

Brooks, SP (1998). Markov chain Monte Carlo method and its application. *The Statistician*, **47**, 69-100.

Brooks, SP and Gelman, A (1998). Alternative methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, **7**, 434-455.

Browne, WJ (2003). MCMC Estimation in MLwiN. London: Institute of Education, University of London, (currently available at `http://multilevel.ioe.ac.uk/dev/develop.html`)

Casella, G and George, EI (1992). Explaining the Gibbs sampler. *The American Statistician*, **46**, 167–174.

Congdon, P (2001) Bayesian statistical modelling. Wiley.

Cowles, MK and Carlin, BP (1996) Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, **91**, 883–904.

Fisher, LD (1996). Comments on Bayesian and frequentist analysis and interpretation of clinical trials — comment. *Controlled Clinical Trials*, **17**, 423–34.

Gelfand, AE and Smith, AFM (1990). Sampling-based approaches to calculating marginal densities. *J Amer Statistic Assoc*, **85**, 398–409.

Gelman, A (2005). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, to appear.

Gelman, A, Carlin, JC, Stern, H and Rubin, DB (2004). *Bayesian Data Analysis*, 2nd edition, Chapman & Hall, New York.

Lee, PM (2004). *Bayesian Statistics: An Introduction*, 3rd edition, Arnold, London.

Little RJA and Rubin DB (2002). *Statistical Analysis with Missing Data*, 2nd edition, Wiley, New Jersey.

Richardson, S (1996). Measurement error. In *Markov chain Monte Carlo in Practice*, (eds. DJ Spiegelhalter, WR Gilks, and S Richardson). Chapman & Hall, London, pp. 401-417.

Richardson, S and Best, NG (2003). Bayesian hierarchical models in ecological studies of health-environment effects, *Environmetrics*, **14**, 129-147.

Spiegelhalter, DJ (1998). Bayesian graphical modelling: a case-study in monitoring health outcomes. *Journal of the Royal Statistical Society, Series C*, **47**, 115–133.

Spiegelhalter, DJ, Thomas, A, and Best, NG (1995). Computation on Bayesian graphical models. In *Bayesian Statistics 5* (eds. JM Bernardo, JO Berger, AP Dawid and AFM Smith). Oxford University Press, Oxford), pp. 407-425.

Spiegelhalter, DJ, Gilks, WR and Richardson, S (1996). *Markov chain Monte Carlo in Practice*, Chapman & Hall, London.

Spiegelhalter, DJ, Abrams, K and Myles, JP (2004). *Bayesian Approaches to Clinical Trials and Health Care Evaluation*, Wiley, Chichester.

Spiegelhalter, DJ, Best, NG, Carlin, BP, and van der Linde, A (2002). Bayesian measures of model complexity and fit (with discussion). *J Roy Statist Soc B*, **64**, 583–639.