

BUGS Rats: a normal hierarchical model

This example is taken from section 6 of Gelfand *et al* (1990), and concerns 30 young rats whose weights were measured weekly for five weeks. Part of the data is shown below, where Y_{ij} is the weight of the i th rat measured at age x_j .

	Weights Y_{ij} of rat i on day x_j				
	$x_j = 8$	15	22	29	36
Rat 1	151	199	246	283	320
Rat 2	145	199	249	293	354
.....					
Rat 30	153	200	244	286	324

A plot of the 30 growth curves suggests some evidence of downward curvature.

The model is essentially a random effects linear growth curve

$$Y_{ij} \sim \text{Normal}(\alpha_i + \beta_i(x_j - \bar{x}), \tau_c)$$

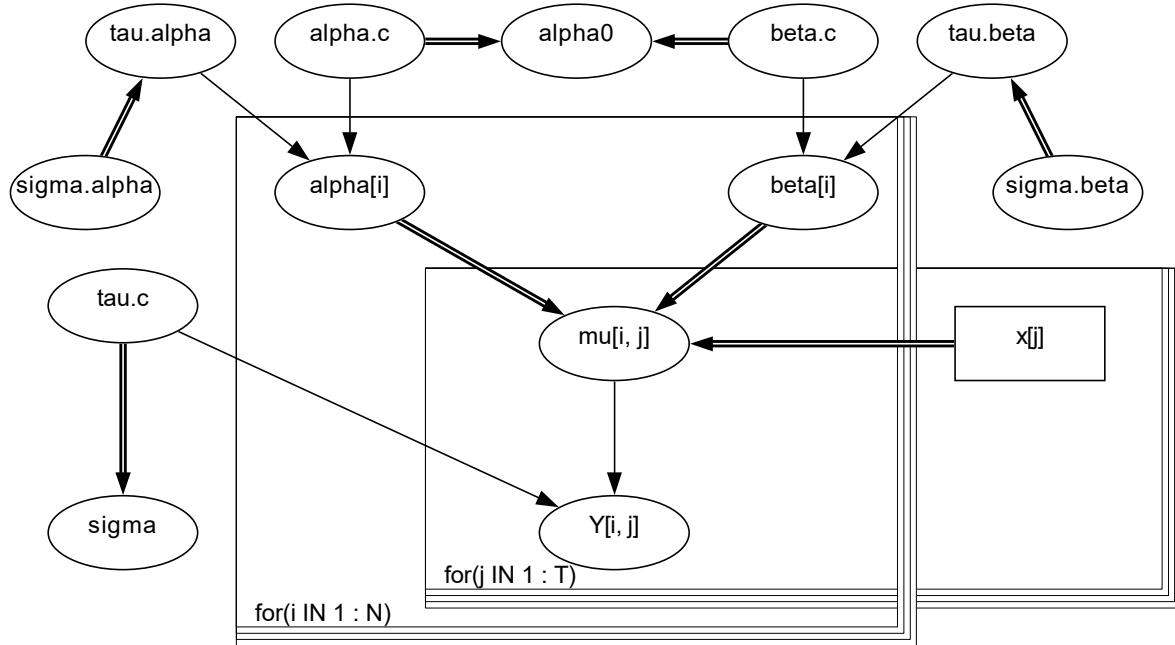
$$\alpha_i \sim \text{Normal}(\alpha_c, \tau_\alpha)$$

$$\beta_i \sim \text{Normal}(\beta_c, \tau_\beta)$$

where $\bar{x} = 22$, and τ represents the *precision* (1/variance) of a normal distribution. We note the absence of a parameter representing correlation between α_i and β_i unlike in Gelfand *et al* 1990. However, see the Birats example in Volume 2 which does explicitly model the covariance between α_i and β_i . For now, we standardise the x_j 's around their mean to reduce dependence between α_i and β_i in their likelihood: in fact for the full balanced data, complete independence is achieved. (Note that, in general, prior independence does not force the posterior distributions to be independent).

$\alpha_c, \tau_\alpha, \beta_c, \tau_\beta, \tau_c$ are given independent "noninformative" priors, with two alternatives considered for τ_α and τ_β : prior 1 is uniform on the scale of the standard deviations $\sigma_\alpha = 1/\sqrt{\tau_\alpha}$ and $\sigma_\beta = 1/\sqrt{\tau_\beta}$, and prior 2 is a gamma(0.001, 0.001) on the precisions τ_α and τ_β . Interest particularly focuses on the intercept at zero time (birth), denoted $\alpha_0 = \alpha_c - \beta_c \bar{x}$.

Graphical model for rats example (using prior 1):



BUGS language for rats example:

```

model
{
  for( i in 1 : N ) {
    for( j in 1 : T ) {
      Y[i , j] ~ dnorm(mu[i , j],tau.c)
      mu[i , j] <- alpha[i] + beta[i] * (x[j] - xbar)
    }
    alpha[i] ~ dnorm(alpha.c,tau.alpha)
    beta[i] ~ dnorm(beta.c,tau.beta)
  }
  tau.c ~ dgamma(0.001,0.001)
  sigma <- 1 / sqrt(tau.c)
  alpha.c ~ dnorm(0.0,1.0E-6)
  # Choice of prior of random effects variances
  # Prior 1: uniform on SD
  sigma.alpha~ dunif(0,100)
  sigma.beta~ dunif(0,100)
  tau.alpha<-1/(sigma.alpha*sigma.alpha)
  tau.beta<-1/(sigma.beta*sigma.beta)

  #Prior 2: (not recommended)
  #tau.alpha ~ dgamma(0.001,0.001)
  #tau.beta ~ dgamma(0.001,0.001)

  beta.c ~ dnorm(0.0,1.0E-6)

  alpha0 <- alpha.c - xbar * beta.c
}

```

Note the use of a very flat but conjugate prior for the population effects: a locally uniform prior could also have been used.

Data \Rightarrow list(x = c(8.0, 15.0, 22.0, 29.0, 36.0), xbar = 22, N = 30, T = 5,

```

Y = structure(
  .Data = c(151, 199, 246, 283, 320,
           145, 199, 249, 293, 354,
           147, 214, 263, 312, 328,
           155, 200, 237, 272, 297,
           135, 188, 230, 280, 323,
           159, 210, 252, 298, 331,
           141, 189, 231, 275, 305,
           159, 201, 248, 297, 338,
           177, 236, 285, 350, 376,
           134, 182, 220, 260, 296,
           160, 208, 261, 313, 352,
           143, 188, 220, 273, 314,
           154, 200, 244, 289, 325,
           171, 221, 270, 326, 358,
           163, 216, 242, 281, 312,
           160, 207, 248, 288, 324,

```

142, 187, 234, 280, 316,
 156, 203, 243, 283, 317,
 157, 212, 259, 307, 336,
 152, 203, 246, 286, 321,
 154, 205, 253, 298, 334,
 139, 190, 225, 267, 302,
 146, 191, 229, 272, 302,
 157, 211, 250, 285, 323,
 132, 185, 237, 286, 331,
 160, 207, 257, 303, 345,
 169, 216, 261, 295, 333,
 157, 205, 248, 289, 316,
 137, 180, 219, 258, 291,
 153, 200, 244, 286, 324),
 .Dim = c(30,5))) \leftarrow

(Note: the response data (Y) for the rats example can also be found in the file ratsy.odc in rectangular format. The covariate data (X) can be found in S-Plus format in file ratsx.odc. To load data from each of these files, focus the window containing the open data file before clicking on "Data" from the "Model" menu.)

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha0	106.6	3.65	0.04151	99.43	106.5	113.9	1001	10000
beta.c	6.185	0.1102	0.001294	5.967	6.185	6.404	1001	10000
sigma	6.074	0.4673	0.007724	5.247	6.044	7.068	1001	10000

Using Prior 2 (not recommended) has negligible impact

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha0	106.6	3.625	0.03477	99.32	106.6	113.6	1001	10000
beta.c	6.185	0.1068	0.001354	5.979	6.184	6.398	1001	10000
sigma	6.082	0.4714	0.007308	5.248	6.052	7.093	1001	10000

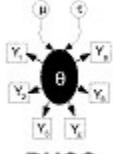
These results may be compared with Figure 5 of Gelfand *et al* 1990 --- we note that the mean gradient of independent fitted straight lines is 6.19.

Gelfand *et al* 1990 also consider the problem of missing data, and delete the last observation of cases 6-10, the last two from 11-20, the last 3 from 21-25 and the last 4 from 26-30. The appropriate data file is obtained by simply replacing data values by NA (see below). The model specification is unchanged, since the distinction between observed and unobserved quantities is made in the data file and not the model specification.

Gelfand *et al* 1990 focus on the parameter estimates and the predictions for the final 4 observations on rat 26. These predictions are obtained automatically in *BUGS* by monitoring the relevant *Y[]* nodes. The following estimates were obtained:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
Y[26,2]	204.5	8.74	0.1159	187.0	204.4	221.7	1001	10000
Y[26,3]	250.0	10.27	0.1642	229.7	249.9	270.1	1001	10000
Y[26,4]	295.4	12.64	0.2092	270.3	295.3	320.3	1001	10000
Y[26,5]	340.6	15.32	0.284	310.2	340.5	370.5	1001	10000
beta.c	6.575	0.1507	0.003708	6.281	6.573	6.875	1001	10000

We note that our estimate 6.58 of bc is substantially greater than that shown in Figure 6 of Gelfand *et al* 1990. However, plotting the growth curves indicates some curvature with steeper gradients at the beginning: the mean of the estimated gradients of the reduced data is 6.66, compared to 6.19 for the full data. Hence we are inclined to believe our analysis. The observed weights for rat 26 were 207, 257, 303 and 345, compared to our predictions of 204, 250, 295 and 341.



BUGS

Pumps: conjugate gamma-Poisson hierarchical model

George *et al* (1993) discuss Bayesian analysis of hierarchical models where the conjugate prior is adopted at the first level, but for any given prior distribution of the hyperparameters, the joint posterior is not of closed form. The example they consider relates to 10 power plant pumps. The number of failures x_i is assumed to follow a Poisson distribution

$$x_i \sim \text{Poisson}(\theta_i t_i) \quad i = 1, \dots, 10$$

where θ_i is the failure rate for pump i and t_i is the length of operation time of the pump (in 1000s of hours). The data are shown below.

Pump	t_i	x_i
1	94.5	5
2	15.7	1
3	62.9	5
4	126	14
5	5.24	3
6	31.4	19
7	1.05	1
8	1.05	1
9	2.1	4
10	10.5	22

A conjugate gamma prior distribution is adopted for the failure rates:

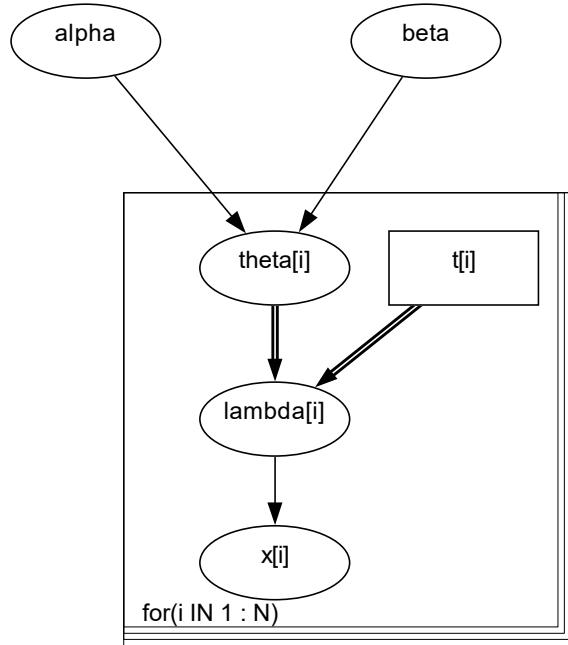
$$\theta_i \sim \text{Gamma}(\alpha, \beta), \quad i = 1, \dots, 10$$

George *et al* (1993) assume the following prior specification for the hyperparameters α and β

$$\begin{aligned}\alpha &\sim \text{Exponential}(1.0) \\ \beta &\sim \text{Gamma}(0.1, 1.0)\end{aligned}$$

They show that this gives a posterior for β which is a gamma distribution, but leads to a non-standard posterior for α . Consequently, they use the Gibbs sampler to simulate the required posterior densities.

Graphical model for pump example:



BUGS *language for pump example*:

```

model
{
  for (i in 1 : N) {
    theta[i] ~ dgamma(alpha, beta)
    lambda[i] <- theta[i] * t[i]
    x[i] ~ dpois(lambda[i])
  }
  alpha ~ dexp(1)
  beta ~ dgamma(0.1, 1.0)
}

```

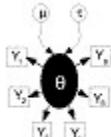
Data \Rightarrow list(t = c(94.3, 15.7, 62.9, 126, 5.24, 31.4, 1.05, 1.05, 2.1, 10.5),
 $x = c(5, 1, 5, 14, 3, 19, 1, 1, 4, 22), N = 10)$ \Leftrightarrow

Inits \Rightarrow list(alpha = 1, beta = 1) \Leftrightarrow

Results

A burn in of 1000 updates followed by a futher 10000 updates gave the parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	0.7001	0.2699	0.004706	0.2851	0.6634	1.338	1001	10000
beta	0.929	0.5325	0.00978	0.1938	0.8315	2.205	1001	10000
theta[1]	0.0598	0.02542	2.68E-4	0.02128	0.05627	0.1195	1001	10000
theta[2]	0.1008	0.07855	8.177E-4	0.00838	0.08181	0.3023	1001	10000
theta[3]	0.08927	0.03759	3.702E-4	0.0316	0.08469	0.1762	1001	10000
theta[4]	0.116	0.03048	3.17E-4	0.06363	0.1132	0.1825	1001	10000
theta[5]	0.6056	0.315	0.003087	0.1529	0.5529	1.359	1001	10000
theta[6]	0.6105	0.1393	0.0014	0.3668	0.5996	0.9096	1001	10000
theta[7]	0.9025	0.7252	0.007937	0.07559	0.7167	2.751	1001	10000
theta[8]	0.8964	0.725	0.008262	0.07614	0.7098	2.785	1001	10000
theta[9]	1.59	0.7767	0.009004	0.4828	1.452	3.452	1001	10000
theta[10]	1.993	0.4251	0.004915	1.264	1.958	2.916	1001	10000



BUGS Dogs: loglinear model for binary data

Lindley (19??) analyses data from Kalbfleisch (1985) on the Solomon-Wynne experiment on dogs, whereby they learn to avoid an electric shock. A dog is put in a compartment, the lights are turned out and a barrier is raised, and 10 seconds later an electric shock is applied. The results are recorded as success ($Y=1$) if the dog jumps the barrier before the shock occurs, or failure ($Y=0$) otherwise.

Thirty dogs were each subjected to 25 such trials. A plausible model is to suppose that a dog learns from previous trials, with the probability of success depending on the number of previous shocks and the number of previous avoidances. Lindley thus uses the following model

$$\pi_j = A^{x_j} B^{j-x_j}$$

for the probability of a shock (failure) at trial j , where x_j = number of success (avoidances) before trial j and $j-x_j$ = number of previous failures (shocks). This is equivalent to the following log linear model

$$\log \pi_j = \alpha x_j + \beta (j - x_j)$$

Hence we have a generalised linear model for binary data, but with a log-link function rather than the canonical logit link. This is trivial to implement in BUGS:

```

model
{
  for (i in 1 : Dogs) {
    xa[i, 1] <- 0; xs[i, 1] <- 0 p[i, 1] <- 0
    for (j in 2 : Trials) {
      xa[i, j] <- sum(Y[i, 1 : j - 1])
      xs[i, j] <- j - 1 - xa[i, j]
      log(p[i, j]) <- alpha * xa[i, j] + beta * xs[i, j]
      y[i, j] <- 1 - Y[i, j]
      y[i, j] ~ dbern(p[i, j])
    }
  }
  alpha ~ dnorm(0, 0.00001)
  beta ~ dnorm(0, 0.00001)
  A <- exp(alpha)
  B <- exp(beta)
}

```

Data

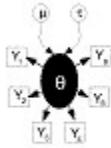
Inits

```
list(alpha = -1, beta = -1)
```

Results

Time for 10000 updates was 158s on 200MHz Pentium Pro.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
A	0.7831	0.01916	2.72E-4	0.7448	0.7837	0.8203	1001	10000
B	0.9243	0.01086	1.436E-4	0.9019	0.9246	0.9446	1001	10000
alpha	-0.2448	0.02451	3.489E-4	-0.2947	-0.2437	-0.1981	1001	10000
beta	-0.07875	0.01176	1.555E-4	-0.1032	-0.07842	-0.05704	1001	10000



Seeds: Random effect logistic regression

This example is taken from Table 3 of Crowder (1978), and concerns the proportion of seeds that germinated on each of 21 plates arranged according to a 2 by 2 factorial layout by seed and type of root extract. The data are shown below, where r_i and n_i are the number of germinated and the total number of seeds on the i th plate, $i = 1, \dots, N$. These data are also analysed by, for example, Breslow: and Clayton (1993).

seed O. aegyptiaco 75			seed O. aegyptiaco 73								
Bean			Cucumber			Bean			Cucumber		
r	n	r/n	r	n	r/n	r	n	r/n	r	n	r/n
10	39	0.26	5	6	0.83	8	16	0.50	3	12	0.25
23	62	0.37	53	74	0.72	10	30	0.33	22	41	0.54
23	81	0.28	55	72	0.76	8	28	0.29	15	30	0.50
26	51	0.51	32	51	0.63	23	45	0.51	32	51	0.63
17	39	0.44	46	79	0.58	0	4	0.00	3	7	0.43
			10	13	0.77						

The model is essentially a random effects logistic, allowing for over-dispersion. If p_i is the probability of germination on the i th plate, we assume

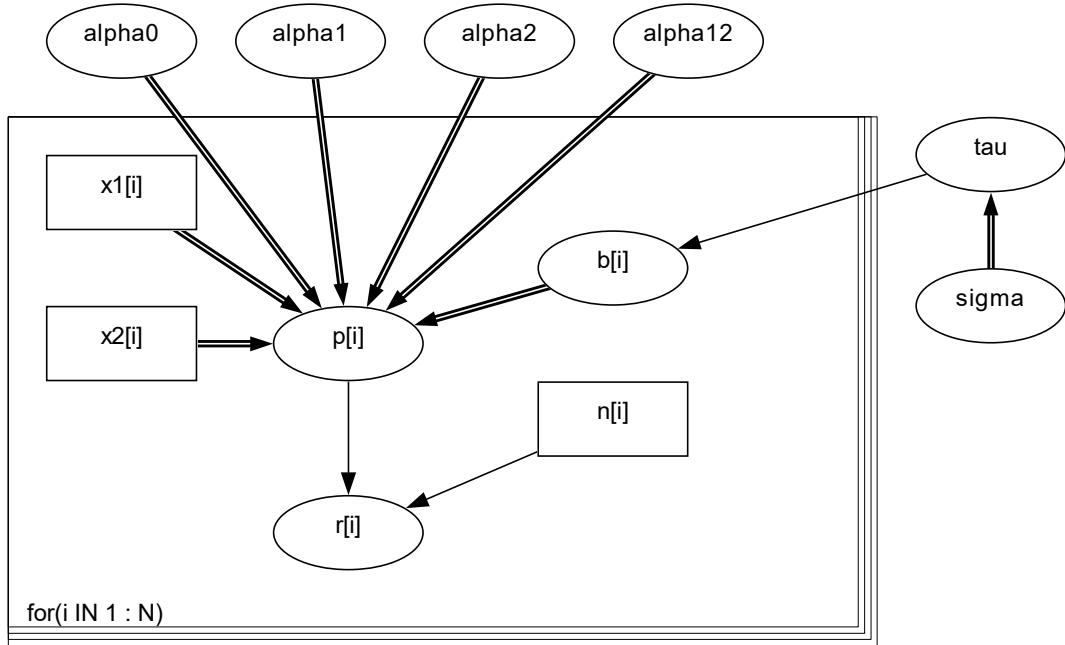
$$r_i \sim \text{Binomial}(p_i, n_i)$$

$$\text{logit}(p_i) = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_{12} x_{1i} x_{2i} + b_i$$

$$b_i \sim \text{Normal}(0, \tau)$$

where x_{1i} , x_{2i} are the seed type and root extract of the i th plate, and an interaction term $\alpha_{12} x_{1i} x_{2i}$ is included. α_0 , α_1 , α_2 , α_{12} are given independent "noninformative" priors; two alternative "noninformative" priors are considered for the random effects variance: prior 1 is a uniform prior on the standard deviation, and prior 2 is a gamma(0.001, 0.001) prior on the precision.

Graphical model for seeds example (assuming prior 1)



BUGS language for seeds example

```

model
{
  for( i in 1 : N ) {
    r[i] ~ dbin(p[i],n[i])
    b[i] ~ dnorm(0.0,tau)
    logit(p[i]) <- alpha0 + alpha1 * x1[i] + alpha2 * x2[i] +
      alpha12 * x1[i] * x2[i] + b[i]
  }
  alpha0 ~ dnorm(0.0,1.0E-6)
  alpha1 ~ dnorm(0.0,1.0E-6)
  alpha2 ~ dnorm(0.0,1.0E-6)
  alpha12 ~ dnorm(0.0,1.0E-6)
  # Choice of priors for random effects variance
  # Prior 1: uniform on SD
  sigma~ dunif(0,100)
  tau<-1/(sigma*sigma)

  #Prior 2:
  #tau ~ dgamma(1.0E-3, 1.0E-3);
  #sigma <- 1/sqrt(tau); # s.d. of random effects
}

```

Data ↳

```
list(r = c(10, 23, 23, 26, 17, 5, 53, 55, 32, 46, 10, 8, 10, 8, 23, 0, 3, 22, 15, 32, 3),
n = c(39, 62, 81, 51, 39, 6, 74, 72, 51, 79, 13, 16, 30, 28, 45, 4, 12, 41, 30, 51, 7),
x1 = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
x2 = c(0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1),
N = 21) ↳
```

Inits1 ↳ list(alpha0 = 0, alpha1 = 0, alpha2 = 0, alpha12 = 0, sigma = 1) ↳

Inits2 ↳ list(alpha0 = 0, alpha1 = 0, alpha2 = 0, alpha12 = 0, tau = 1) ↳

Results

A burn in of 1000 updates followed by a further 10000 updates gave the following parameter estimates:

Results using prior 1 (uniform on SD):

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha0	-0.5581	0.2104	0.008852	-0.9755	-0.5608	-0.1423	1001	10000
alpha1	0.0749	0.345	0.01313	-0.6258	0.0785	0.7355	1001	10000
alpha12	-0.8525	0.4713	0.01798	-1.8	-0.8466	0.05573	1001	10000
alpha2	1.372	0.2901	0.01141	0.7929	1.368	1.953	1001	10000
sigma	0.353	0.1478	0.006523	0.08363	0.3425	0.6775	1001	10000

Prior 2 (not recommended) leads to a slightly increased precision:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha0	-0.5546	0.1941	0.007696	-0.9353	-0.5577	-0.1597	1001	10000
alpha1	0.08497	0.3127	0.01283	-0.5814	0.09742	0.6679	1001	10000
alpha12	-0.8229	0.4321	0.01785	-1.697	-0.8218	0.01641	1001	10000
alpha2	1.356	0.2743	0.01236	0.8257	1.347	1.909	1001	10000
sigma	0.2731	0.1437	0.007956	0.04133	0.2654	0.5862	1001	10000

We may compare simple logistic, maximum likelihood (from EGRET), penalized quasi-likelihood (PQL) Breslow and Clayton (1993) with the BUGS results

variable	Logistic regression		maximum likelihood		PQL	
	β	SE	β	SE	β	SE
α_0	-0.558	0.126	-0.546	0.167	-0.542	0.190
α_1	0.146	0.223	0.097	0.278	0.77	0.308
α_2	1.318	0.177	1.337	0.237	1.339	0.270
α_{12}	-0.778	0.306	-0.811	0.385	-0.825	0.430
σ	---	--	0.236	0.110	0.313	0.121

Heirarchical centering is an interesting reformulation of random effects models. Introduce the variables

$$\mu_i = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_{12} x_{1i} x_{2i}$$

$$\beta_i = \mu_i + b_i$$

the model then becomes

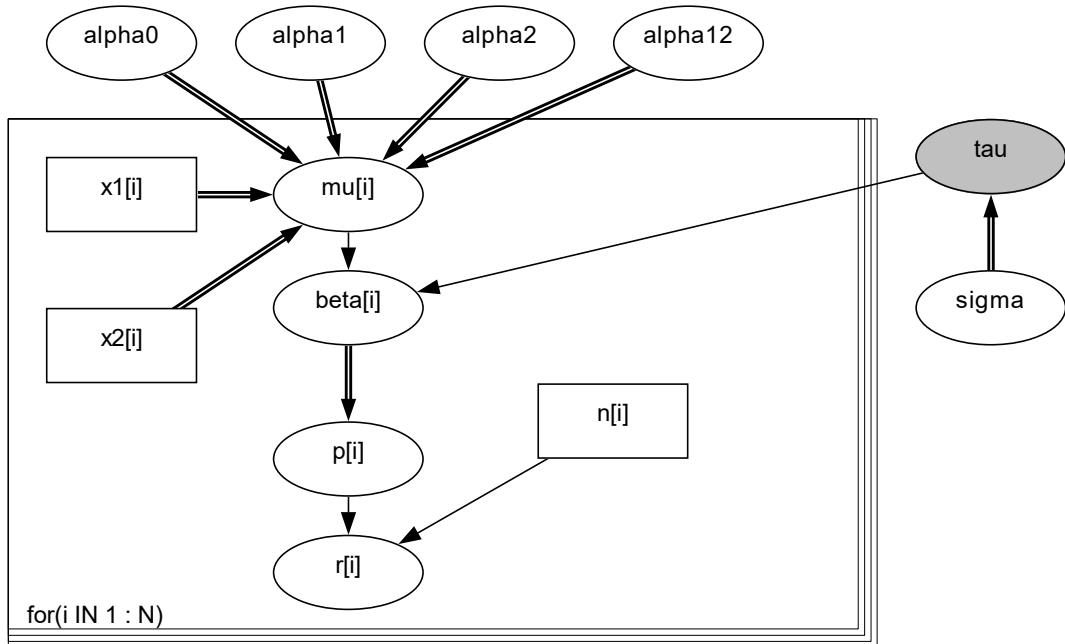
$$r_i \sim \text{Binomial}(p_i, n_i)$$

$$\text{logit}(p_i) = \beta_i$$

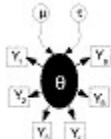
$$\beta_i \sim \text{Normal}(\mu_i, \tau)$$

The graphical model is shown below

name:	tau	type:	logical	link:	identity
value:	1/(sigma*sigma)				



This formulation of the model has two advantages: the sequence of random numbers generated by the Gibbs sampler has better correlation properties and the time per update is reduced because the updating for the α parameters is now conjugate.



BUGS

Surgical: Institutional ranking

This example considers mortality rates in 12 hospitals performing cardiac surgery in babies. The data are shown below.

Hospital	No of ops	No of deaths
A	47	0
B	148	18
C	119	8
D	810	46
E	211	8
F	196	13
G	148	9
H	215	31
I	207	14
J	97	8
K	256	29
L	360	24

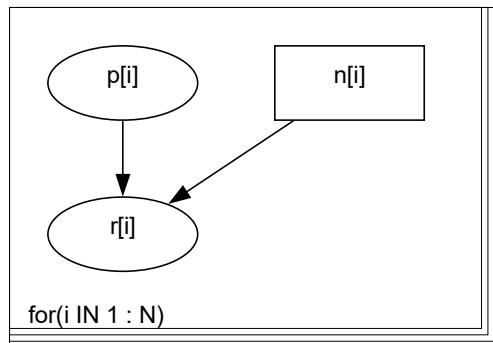
The number of deaths r_i for hospital i are modelled as a binary response variable with 'true' failure probability p_i :

$$r_i \sim \text{Binomial}(p_i, n_i)$$

We first assume that the true failure probabilities are *independent* (i.e. fixed effects) for each hospital. This is equivalent to assuming a standard non-informative prior distribution for the p_i 's, namely:

$$p_i \sim \text{Beta}(1.0, 1.0)$$

Graphical model for fixed effects surgical example:



BUGS language for fixed effects surgical model:

```

model
{
  for( i in 1 : N ) {
    p[i] ~ dbeta(1.0, 1.0)
    r[i] ~ dbin(p[i], n[i])
  }
}

```

Data \Rightarrow

```

list(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),
     N = 12)  $\Leftarrow$ 

```

Inits \Rightarrow

```

list(p = c(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1))  $\Leftarrow$ 

```

A more realistic model for the surgical data is to assume that the failure rates across hospitals are *similar* in some way. This is equivalent to specifying a *random effects* model for the true failure probabilities p_i as follows:

$$\text{logit}(p_i) = b_i$$

$$b_i \sim \text{Normal}(\mu, \tau)$$

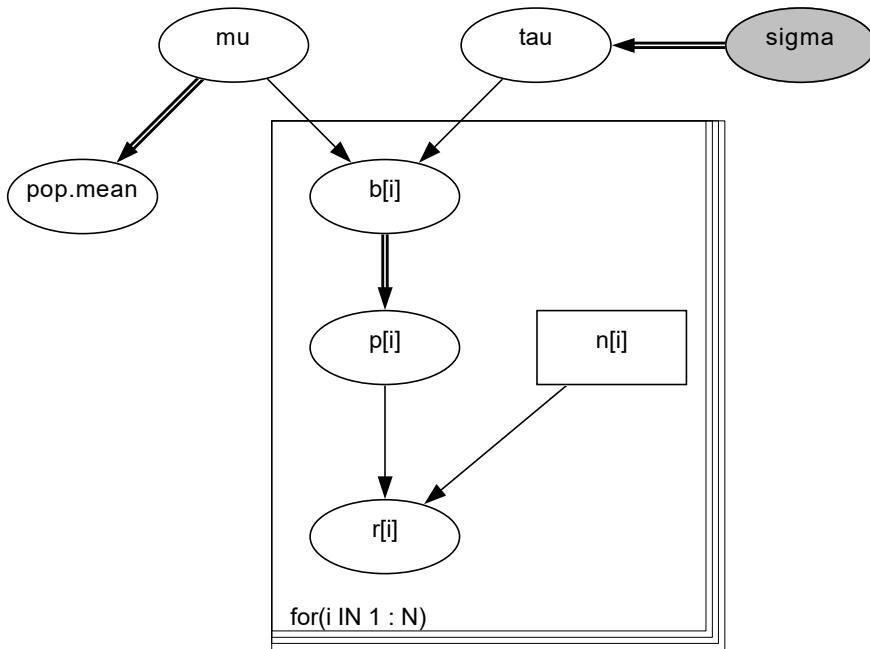
A standard non-informative prior is then specified for the population mean (logit) probability of failure, μ , with two alternative "noninformative" priors considered for the random effects variance: prior 1 is a uniform prior on the standard deviation, and prior 2 is a gamma(0.001, 0.001) prior on the precision, τ .

Graphical model for random effects surgical example (prior 1):

```

name: sigma      type: stochastic   density: dunif
lower bound 0     upper bound 100

```



BUGS language for random effects surgical model:

```

model
{
  for( i in 1 : N ) {
    b[i] ~ dnorm(mu,tau)
    r[i] ~ dbin(p[i],n[i])
    logit(p[i]) <- b[i]
  }
  pop.mean <- exp(mu) / (1 + exp(mu))
  mu ~ dnorm(0.0,1.0E-6)
  # Choice of prior on random effects variance
  # Prior 1: uniform on SD
  sigma~ dunif(0,100)
  tau<-1/(sigma*sigma)

  #Prior 2:
  #tau ~ dgamma(1.0E-3, 1.0E-3);
  #sigma <- 1/sqrt(tau); # s.d. of random effects
}

```

Inits1 ➔ click on one of the arrows to open the initial values (prior 1) ←

Inits2 ➔ click on one of the arrows to open the initial values (prior 2) ←

Results

A burn in of 1000 updates followed by a further 10000 updates gave the following estimates of surgical mortality

in each hospital for the fixed effect analysis

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p[1]	0.02032	0.01989	1.95E-4	5.62E-4	0.01443	0.07396	1001	10000
p[2]	0.1267	0.02674	2.634E-4	0.07905	0.125	0.1849	1001	10000
p[3]	0.07441	0.02414	2.263E-4	0.03388	0.07197	0.1285	1001	10000
p[4]	0.0579	0.008179	9.032E-5	0.04291	0.05751	0.07502	1001	10000
p[5]	0.0421	0.01378	1.434E-4	0.01993	0.04048	0.0736	1001	10000
p[6]	0.07082	0.01805	1.862E-4	0.03966	0.06951	0.1095	1001	10000
p[7]	0.06681	0.02036	2.058E-4	0.03209	0.06505	0.1116	1001	10000
p[8]	0.1475	0.02409	2.136E-4	0.1034	0.1465	0.1973	1001	10000
p[9]	0.07181	0.01781	1.89E-4	0.04072	0.0703	0.1099	1001	10000
p[10]	0.09129	0.02921	2.638E-4	0.04221	0.08845	0.1553	1001	10000
p[11]	0.1164	0.01996	2.042E-4	0.08016	0.1155	0.1579	1001	10000
p[12]	0.06925	0.01323	1.434E-4	0.04554	0.06839	0.09723	1001	10000

and for the random effects analysis using prior 1:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	-2.564	0.1731	0.002066	-2.927	-2.557	-2.238	1001	10000
p[1]	0.04996	0.02046	3.488E-4	0.01473	0.04877	0.09364	1001	10000
p[2]	0.1056	0.02225	3.02E-4	0.06799	0.1037	0.1542	1001	10000
p[3]	0.07013	0.01831	1.976E-4	0.03857	0.06875	0.1092	1001	10000
p[4]	0.05882	0.007996	9.063E-5	0.04407	0.05849	0.07547	1001	10000
p[5]	0.05002	0.01343	2.124E-4	0.02638	0.04923	0.07813	1001	10000
p[6]	0.06863	0.0153	1.823E-4	0.0416	0.06772	0.102	1001	10000
p[7]	0.06584	0.01653	1.842E-4	0.03643	0.06473	0.1019	1001	10000
p[8]	0.1266	0.02241	3.47E-4	0.08726	0.1255	0.1739	1001	10000
p[9]	0.06943	0.01495	1.639E-4	0.0427	0.06845	0.1013	1001	10000
p[10]	0.07882	0.02121	2.201E-4	0.04334	0.07692	0.1255	1001	10000
p[11]	0.1038	0.01789	2.473E-4	0.07246	0.1026	0.1418	1001	10000
p[12]	0.06807	0.01181	1.256E-4	0.04676	0.06756	0.09279	1001	10000
pop.mean	0.07234	0.01151	1.354E-4	0.05083	0.07193	0.09642	1001	10000
sigma	0.4615	0.1795	0.004137	0.197	0.4335	0.9033	1001	10000

Prior 2 estimates a slightly smaller random effects SD, but has little impact on the overall results.

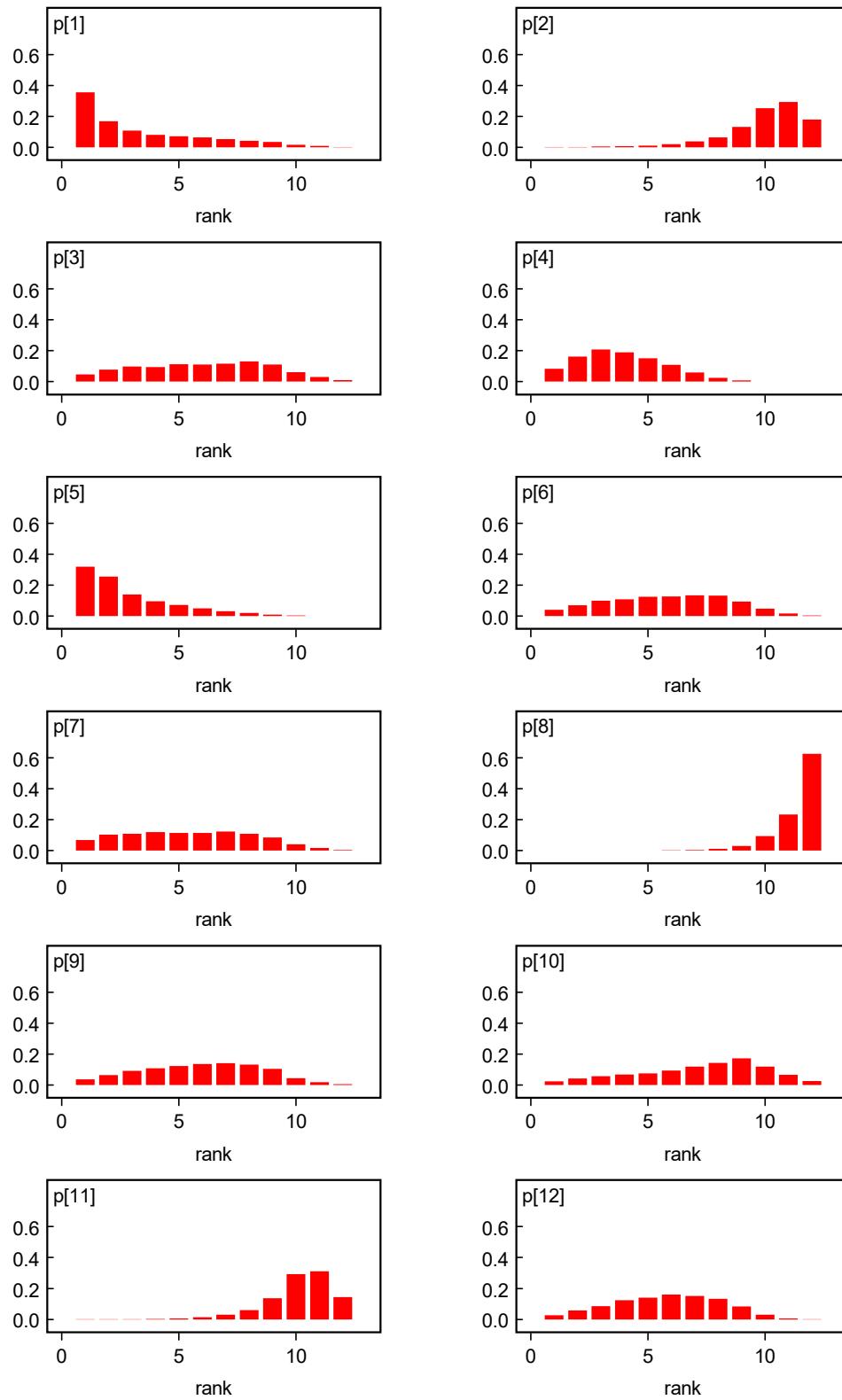
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	-2.554	0.1524	0.002556	-2.877	-2.547	-2.27	1001	10000
p[1]	0.05339	0.01949	3.605E-4	0.0183	0.05288	0.09332	1001	10000
p[2]	0.1032	0.02207	2.985E-4	0.06643	0.1011	0.1527	1001	10000
p[3]	0.07053	0.01718	2.089E-4	0.0404	0.06958	0.1073	1001	10000
p[4]	0.05937	0.007905	1.148E-4	0.04484	0.05904	0.07583	1001	10000
p[5]	0.05181	0.01327	2.429E-4	0.02797	0.05114	0.08004	1001	10000
p[6]	0.06956	0.01475	1.598E-4	0.04355	0.0686	0.1014	1001	10000
p[7]	0.06671	0.01582	1.967E-4	0.03879	0.0655	0.1008	1001	10000
p[8]	0.123	0.02208	4.144E-4	0.08326	0.1219	0.1696	1001	10000
p[9]	0.06993	0.01463	1.779E-4	0.04388	0.0691	0.1017	1001	10000
p[10]	0.07854	0.01998	1.95E-4	0.04476	0.07686	0.1232	1001	10000
p[11]	0.102	0.0175	2.548E-4	0.07154	0.1008	0.1397	1001	10000
p[12]	0.06857	0.0118	1.183E-4	0.04716	0.06799	0.09341	1001	10000
pop.mean	0.07283	0.01013	1.693E-4	0.05333	0.07263	0.09365	1001	10000
sigma	0.4021	0.1577	0.003781	0.159	0.379	0.7801	1001	10000

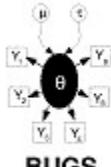
A particular strength of the Markov chain Monte Carlo (Gibbs sampling) approach implemented in *BUGS* is the ability to make inferences on arbitrary functions of unknown model parameters. For example, we may compute the *rank* probability of failure for each hospital at each iteration. This yields a sample from the posterior distribution of the ranks.

The figures below show the posterior ranks for the estimated surgical mortality rate in each hospital for the random effect models. These are obtained by setting the rank monitor for variable p (select the "Rank" option

from the "Statistics" menu) after the burn-in phase, and then selecting the "histogram" option from this menu after a further 10000 updates. These distributions illustrate the considerable uncertainty associated with 'league tables': there are only 2 hospitals (H and K) whose intervals exclude the median rank and none whose intervals fall completely within the lower or upper quartiles.

Plots of distribution of ranks of true failure probability for random effects model:





Salm: extra - Poisson variation in dose - response study

Breslow (1984) analyses some mutagenicity assay data (shown below) on salmonella in which three plates have been processed at each dose i of quinoline and the number of revertant colonies of TA98 Salmonella measured. A certain dose-response curve is suggested by theory.

dose of quinoline (μg per plate)						
0	10	33	100	333	1000	
15	16	16	27	33	20	
21	18	26	41	38	27	
29	21	33	69	41	42	

This is assumed to be a random effects Poisson model allowing for over-dispersion. Let x_i be the dose on the plates $i1$, $i2$ and $i3$. Then we assume

$$y_{ij} \sim \text{Poisson}(\mu_{ij})$$

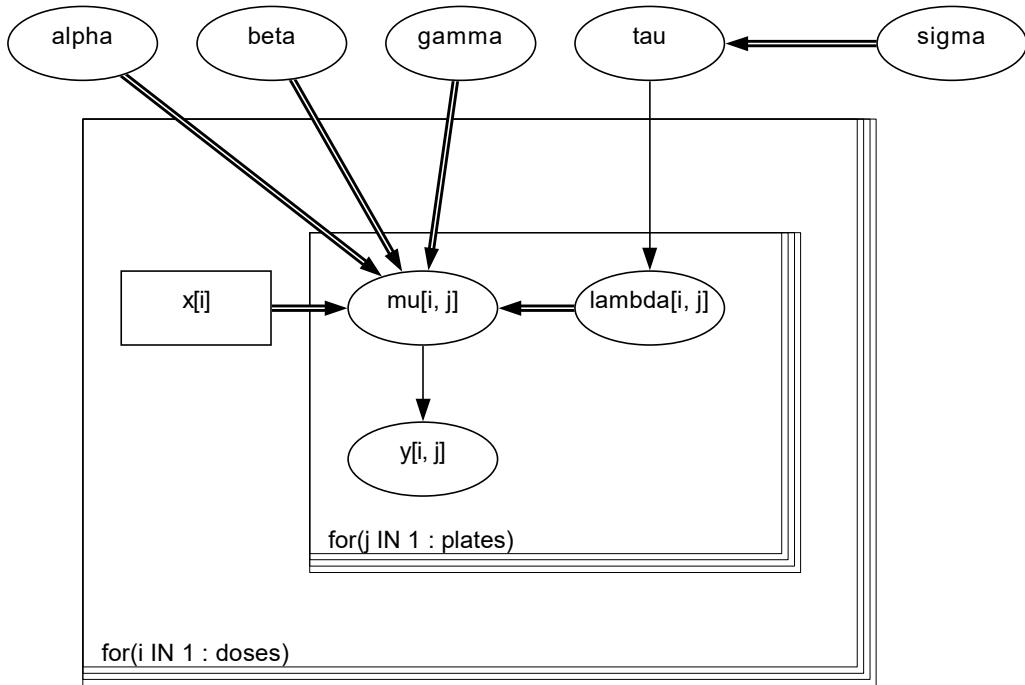
$$\log(\mu_{ij}) = \alpha + \beta \log(x_i + 10) + \gamma x_i + \lambda_{ij}$$

$$\lambda_{ij} \sim \text{Normal}(0, \tau)$$

α , β , γ are given independent "noninformative" priors; two alternative "noninformative" priors are considered for the random effects variance: prior 1 is a uniform prior on the standard deviation, and prior 2 is a gamma(0.001, 0.001) prior on the precision.

The appropriate graph (assuming prior 1) is shown below:

Graphical model for salm example (with prior 1)



BUGS language for salm example

```

for( i in 1 : doses ) {
  for( j in 1 : plates ) {
    y[i , j] ~ dpois(mu[i , j])
    log(mu[i , j]) <- alpha + beta * log(x[i] + 10) +
      gamma * x[i] + lambda[i , j]
    lambda[i , j] ~ dnorm(0.0, tau)
  }
}
alpha ~ dnorm(0.0,1.0E-6)
beta ~ dnorm(0.0,1.0E-6)
gamma ~ dnorm(0.0,1.0E-6)
# Choice of priors for random effects variance
# Prior 1: uniform on SD
sigma~ dunif(0,100)
tau<-1/(sigma*sigma)

#Prior 2:
#tau ~ dgamma(1.0E-3, 1.0E-3);
#sigma <- 1/sqrt(tau); # s.d. of random effects
}

```

Data ↴

```
list(doses = 6, plates = 3,
  y = structure(.Data = c(15,21,29,16,18,21,16,26,33,27,41,60,33,38,41,20,27,42),
  .Dim = c(6, 3)),
  x = c(0, 10, 33, 100, 333, 1000))
```

Inits1 ↴

```
list(alpha = 0, beta = 0, gamma = 0, sigma = 1)
```

Inits2 ↴

```
list(alpha = 0, beta = 0, gamma = 0, tau = 0.1)
```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

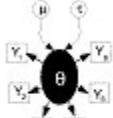
Results using prior 1 (uniform on SD)

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	2.116	0.3714	0.03158	1.323	2.14	2.82	1001	10000
beta	0.3271	0.1021	0.008889	0.1198	0.3227	0.5382	1001	10000
gamma	-0.001038	4.591E-4	3.542E-5	-0.001936	-0.001028	-9.741E-5	1001	10000
sigma	0.2833	0.08556	0.002732	0.1424	0.2731	0.483	1001	10000

Assuming Prior 2 has little impact:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	2.156	0.4076	0.03642	1.346	2.164	3.056	1001	10000
beta	0.3153	0.1118	0.0101	0.07164	0.3132	0.5445	1001	10000
gamma	-9.818E-4	4.823E-4	3.885E-5	-0.001977	-9.648E-4	-8.369E-6	1001	10000
sigma	0.2614	0.08064	0.002819	0.1254	0.2527	0.4438	1001	10000

These estimates can be compared with the quasi-likelihood estimates of Breslow (1984) who reported $\alpha = 2.203 \pm 0.363$, $\beta = 0.311 \pm 0.099$, $\gamma = -9.74E-4 \pm 4.37E-4$, $\sigma = 0.268$



BUGS

Equiv: bioequivalence in a cross-over trial

The table below shows some data from a two-treatment, two-period crossover trial to compare 2 tablets A and B, as reported by Gelfand *et al* (1990).

Subject i	Sequence	seq	Period 1	T_{i1}	Period 2	T_{i2}
1	AB	1	1.40	1	1.65	2
2	AB	1	1.64	1	1.57	2
3	BA	-1	1.44	2	1.58	1
....						
8	AB	1	1.25	1	1.44	2
9	BA	-1	1.25	2	1.39	1
10	BA	-1	1.30	2	1.52	1

The response Y_{ik} from the i th subject ($i = 1, \dots, 10$) in the k th period ($k = 1, 2$) is assumed to be of the form

$$Y_{ik} \sim \text{Normal}(m_{ik}, \tau_1)$$

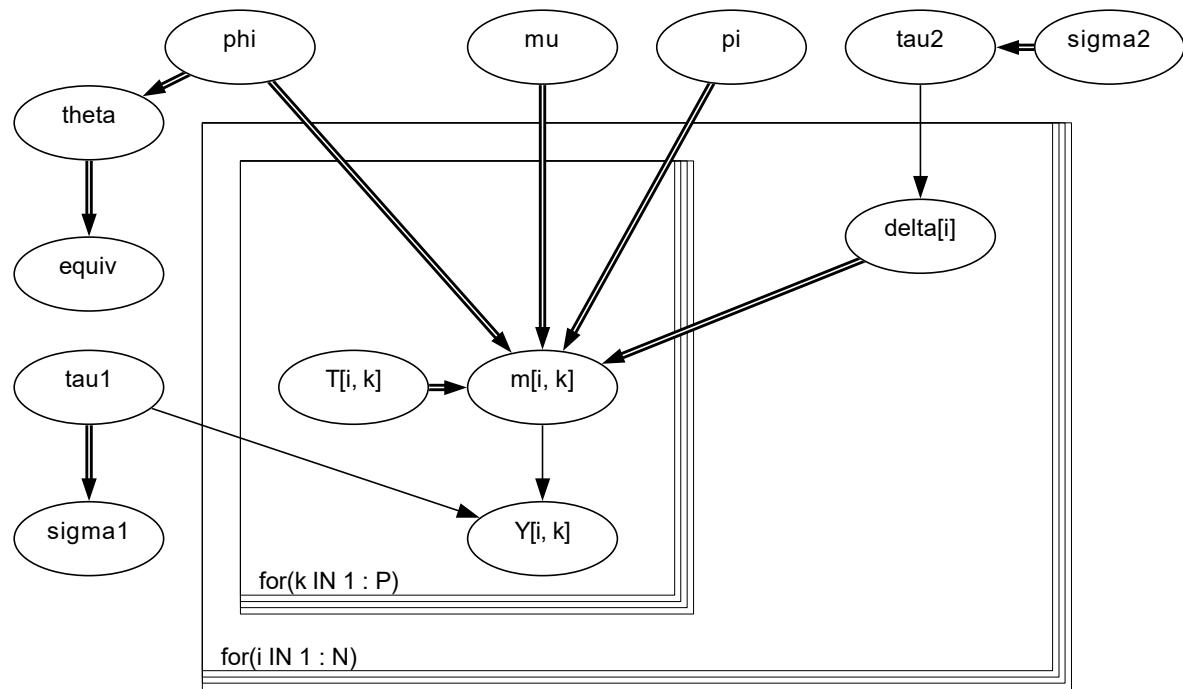
$$m_{ik} = \mu + (-1)^{T_{ik}-1} \phi / 2 + (-1)^{k-1} \pi / 2 + \delta_i$$

$$\delta_i \sim \text{Normal}(0, \tau_2)$$

where $T_{ik}=1,2$ denotes the treatment given to subject i in period k , μ, ϕ, π are the overall mean, treatment and period effects respectively, and δ_i represents the random effect for subject i .

Two alternative "noninformative" priors are considered for the random effects variance: prior 1 is a uniform prior on the standard deviation, and prior 2 is a gamma(0.001, 0.001) prior on the precision. The graph of this model (assuming prior 1) and its BUGS language description are shown below

Graphical model for equiv example (with prior 1)



BUGS language for equiv example

```

model
{
  for( k in 1 : P ){
    for( i in 1 : N ){
      Y[i , k] ~ dnorm(m[i , k], tau1)
      m[i , k] <- mu + sign[T[i , k]] * phi / 2 + sign[k] * pi / 2 + delta[i]
      T[i , k] <- group[i] * (k - 1.5) + 1.5
    }
  }
  for( i in 1 : N ){
    delta[i] ~ dnorm(0.0, tau2)
  }
  tau1 ~ dgamma(0.001, 0.001) sigma1 <- 1 / sqrt(tau1)
  # Choice of prior on random effects variance
  # Prior 1: uniform on SD
  sigma2~ dunif(0,100)
  tau2<-1/(sigma2*sigma2)
  # Prior 2:
  #tau2 ~ dgamma(0.001, 0.001)
  #sigma2 <- 1 / sqrt(tau2)
  mu ~ dnorm(0.0, 1.0E-6)
  phi ~ dnorm(0.0, 1.0E-6)
  pi ~ dnorm(0.0, 1.0E-6)
  theta <- exp(phi)
  equiv <- step(theta - 0.8) - step(theta - 1.2)
}

```

Note the use of the step function to indicate whether $\theta = e^\phi$ lies between 0.8 and 1.2 which traditionally determines bioequivalence.

Data \Rightarrow

```

list(N = 10, P = 2,
  group = c(1, 1, -1, -1, 1, 1, 1, -1, -1),
  Y = structure(.Data = c(1.40, 1.65,
    1.64, 1.57,
    1.44, 1.58,
    1.36, 1.68,
    1.65, 1.69,
    1.08, 1.31,
    1.09, 1.43,
    1.25, 1.44,
    1.25, 1.39,
    1.30, 1.52), .Dim = c(10, 2)),
  sign = c(1, -1)) $\Leftarrow$ 

```

Inits1 \Rightarrow

```
list(mu=0, phi=0, pi=0, tau1= 1, sigma2 = 1) $\Leftarrow$ 
```

Inits2 \Rightarrow

list(mu=0, phi=0, pi=0, tau1= 1, tau2 = 1) ↳

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates.

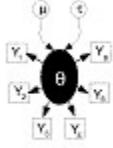
In this analysis the prior on the random-effects variability has minimal impact.

Results using prior 1: uniform on SD

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
equiv	0.9976	0.04893	5.341E-4	1.0	1.0	1.0	1001	10000
mu	1.435	0.0583	0.00194	1.316	1.436	1.547	1001	10000
phi	-0.007634	0.05051	5.296E-4	-0.1085	-0.007785	0.0933	1001	10000
pi	-0.1802	0.05046	5.365E-4	-0.2828	-0.1798	-0.07947	1001	10000
sigma1	0.1082	0.03213	8.047E-4	0.06481	0.1015	0.1887	1001	10000
sigma2	0.1558	0.06028	0.001586	0.0542	0.1477	0.2972	1001	10000
theta	0.9937	0.05021	5.342E-4	0.8971	0.9922	1.098	1001	10000

Results using prior 2: gamma on precision

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
equiv	0.9983	0.0412	4.277E-4	1.0	1.0	1.0	1001	10000
mu	1.436	0.05185	0.001662	1.33	1.437	1.538	1001	10000
phi	-0.008479	0.05106	4.526E-4	-0.1121	-0.008476	0.09185	1001	10000
pi	-0.18	0.0507	5.224E-4	-0.2826	-0.1802	-0.07813	1001	10000
sigma1	0.11	0.03254	9.043E-4	0.06546	0.1038	0.1894	1001	10000
sigma2	0.139	0.05252	0.001376	0.04945	0.1329	0.2596	1001	10000
theta	0.9929	0.05074	4.427E-4	0.894	0.9916	1.096	1001	10000



Dyes: variance components model

Box and Tiao (1973) analyse data first presented by Davies (1967) concerning batch to batch variation in yields of dyestuff. The data (shown below) arise from a balanced experiment whereby the total product yield was determined for 5 samples from each of 6 randomly chosen batches of raw material.

Batch	Yield (in grams)				
1	1545	1440	1440	1520	1580
2	1540	1555	1490	1560	1495
3	1595	1550	1605	1510	1560
4	1445	1440	1595	1465	1545
5	1595	1630	1515	1635	1625
6	1520	1455	1450	1480	1445

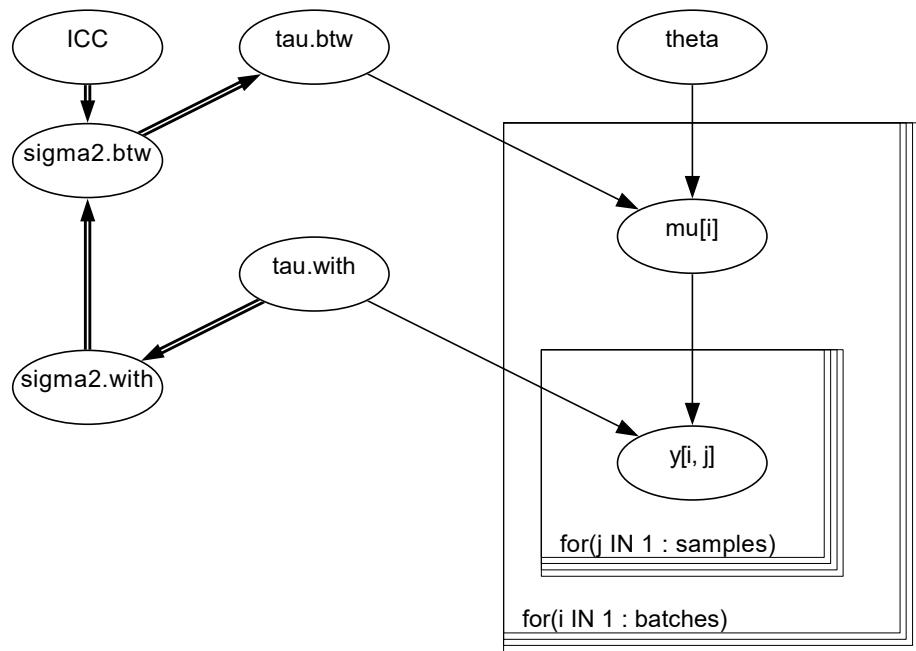
The object of the study was to determine the relative importance of between batch variation versus variation due to sampling and analytic errors. On the assumption that the batches and samples vary independently, and contribute additively to the total error variance, we may assume the following model for dyestuff yield:

$$y_{ij} \sim \text{Normal}(\mu_i, \tau_{\text{within}})$$

$$\mu_i \sim \text{Normal}(\theta, \tau_{\text{between}})$$

where y_{ij} is the yield for sample j of batch i , μ_i is the true yield for batch i , τ_{within} is the inverse of the within-batch variance σ^2_{within} (i.e. the variation due to sampling and analytic error), θ is the true average yield for all batches and τ_{between} is the inverse of the between-batch variance $\sigma^2_{\text{between}}$. The total variation in product yield is thus $\sigma^2_{\text{total}} = \sigma^2_{\text{within}} + \sigma^2_{\text{between}}$ and the relative contributions of each component to the total variance are $f_{\text{within}} = \sigma^2_{\text{within}} / \sigma^2_{\text{total}}$ and $f_{\text{between}} = \sigma^2_{\text{between}} / \sigma^2_{\text{total}}$. The latter is also referred to as the intra-class correlation coefficient (ICC). We assume standard non-informative priors for θ , τ_{within} and three alternative priors for the between-batch variance: prior 1 is a uniform prior on the between batch standard deviation σ_{between} , prior 2 is a uniform prior on the ICC, and prior 3 is a gamma(0.001, 0.001) prior on the between batch precision τ_{between} .

Graphical model for dyes example (with prior 2)



BUGS code for Dyes example:

```

model
{
  for( i in 1 : batches ) {
    mu[i] ~ dnorm(theta, tau.btw)
    for( j in 1 : samples ) {
      y[i , j] ~ dnorm(mu[i], tau.with)
    }
  }
  theta ~ dnorm(0.0, 1.0E-10)
  # prior for within-variation
  sigma2.with <- 1 / tau.with
  tau.with ~ dgamma(0.001, 0.001)

  # Choice of priors for between-variation
  # Prior 1: uniform on SD
  #sigma.btw~ dunif(0,100)
  #sigma2.btw<-sigma.btw*sigma.btw
  #tau.btw<-1/sigma2.btw

  # Prior 2: Uniform on intra-class correlation coefficient,
  #           ICC=sigma2.btw / (sigma2.btw+sigma2.with)
  ICC ~ dunif(0,1)
  sigma2.btw <- sigma2.with *ICC/(1-ICC)
  tau.btw<-1/sigma2.btw

  # Prior 3: gamma(0.001, 0.001) NOT RECOMMENDED
  #tau.btw ~ dgamma(0.001, 0.001)
  #sigma2.btw <- 1 / tau.btw
}

```

Bugs language for dyes example

Data \Rightarrow

```

list(batches = 6, samples = 5,
y = structure(
  .Data = c(1545, 1440, 1440, 1520, 1580,
  1540, 1555, 1490, 1560, 1495,
  1595, 1550, 1605, 1510, 1560,
  1445, 1440, 1595, 1465, 1545,
  1595, 1630, 1515, 1635, 1625,
  1520, 1455, 1450, 1480, 1445), .Dim = c(6, 5))) $\Leftarrow$ 

```

Inits1 \Rightarrow

```
list(theta=1500, tau.with=1, sigma.btw=1) $\Leftarrow$ 
```

Inits2 \Rightarrow

```
list(theta=1500, tau.with=1,ICC=0.5) $\Leftarrow$ 
```

Inits3 \Rightarrow

list(theta=1500, tau.with=1, tau.btw=1) \Leftarrow

Results

1. With uniform prior on between SD:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
sigma2.btw	2929.0	2191.0	12.0	263.9	2306.0	8658.0	10001	100000
sigma2.with	2725.0	880.3	4.112	1505.0	2562.0	4897.0	10001	100000
theta	1528.0	23.92	0.08894	1479.0	1528.0	1576.0	10001	100000

2. With uniform prior on ICC:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
ICC	0.4356	0.1812	0.001155	0.1036	0.4312	0.791	10001	100000
sigma2.btw	2620.0	2760.0	13.55	357.1	1907.0	9158.0	10001	100000
sigma2.with	2682.0	823.2	4.039	1515.0	2538.0	4680.0	10001	100000
theta	1527.0	23.03	0.0843	1481.0	1527.0	1574.0	10001	100000

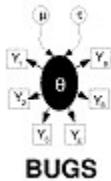
3. Gamma(0.001,0.001) prior on between precision:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
sigma2.btw	2231.0	4326.0	47.83	0.009035	1306.0	10290.0	25001	75000
sigma2.with	3023.0	1097.0	22.54	1557.0	2796.0	5737.0	25001	75000
theta	1527.0	21.63	0.1893	1484.0	1527.0	1572.0	25001	75000

The different priors have negligible impact on the estimation of theta. The (not recommended) Gamma(0.001, 0.01) prior leads to considerably increased uncertainty concerning the variance components.

Note that a relatively long run was required because of the high autocorrelation between successively sampled values of some parameters. Such correlations reduce the 'effective' size of the posterior sample, and hence a longer run is needed to ensure sufficient precision of the posterior estimates.

Note that the posterior distribution for $\sigma^2_{\text{between}}$ has a very long upper tail: hence the posterior mean is considerably larger than the median. Box and Tiao (B&T) estimate $\sigma^2_{\text{within}} = 2451$ and $\sigma^2_{\text{between}} = 1764$ by classical analysis of variance. Here, $\sigma^2_{\text{between}}$ is estimated by the difference of the between- and within-batch mean squares divided by the number of batches - 1. In cases where the between-batch mean square < within-batch mean square, this leads to the unsatisfactory situation of a *negative* variance estimate. Computing a confidence interval for $\sigma^2_{\text{between}}$ is also difficult using the classical approach due to its complicated sampling distribution



Stacks: robust regression

Birkes and Dodge (1993) apply different regression models to the much-analysed stack-loss data of Brownlee (1965). This features 21 daily responses of stack loss y , the amount of ammonia escaping, with covariates being air flow x_1 , temperature x_2 and acid concentration x_3 . Part of the data is shown below.

Day	Stack loss y	air flow x_1	temperature x_2	acid x_3
1	42	80	27	89
2	37	80	27	88
....				
21	15	70	20	91

We first assume a linear regression on the expectation of y , with a variety of different error structures. Specifically

$$\mu_i = \beta_0 + \beta_1 z_{1i} + \beta_2 z_{2i} + \beta_3 z_{3i}$$

$$y_i \sim \text{Normal}(\mu_i, \tau)$$

$$y_i \sim \text{Double exp}(\mu_i, \tau)$$

$$y_i \sim t(\mu_i, \tau, d)$$

where $z_{ij} = (x_{ij} - \bar{x}_{j\cdot}) / \text{sd}(x_j)$ are covariates standardised to have zero mean and unit variance. $\beta_1, \beta_2, \beta_3$ are initially given independent "noninformative" priors.

Maximum likelihood estimates for the double exponential (Laplace) distribution are essentially equivalent to minimising the sum of absolute deviations (LAD), while the other options are alternative heavy-tailed distributions. A t on 4 degrees of freedom has been chosen, although with more data it would be possible to allow this parameter also to be unknown.

We also consider the use of 'ridge regression', intended to avoid the instability due to correlated covariates. This has been shown Lindley and Smith (1972) to be equivalent to assuming the regression coefficients of the standardised covariates to be exchangeable, so that

$$\beta_j \sim \text{Normal}(0, \phi), j = 1, 2, 3.$$

In the following example we extend the work of Birkes and Dodge (1993) by applying this ridge technique to each of the possible error distributions.

Birkes and Dodge (1993) suggest investigating outliers by examining residuals $y_i - \mu_i$ greater than 2.5 standard deviations. We can calculate standardised residuals for each of these distributions, and create a variable `outlier[i]` taking on the value 1 whenever this condition is fulfilled. Mean values of `outlier[i]` then show the confidence with which this definition of outlier is fulfilled.

The *BUGS* language for all the models is shown below, with all models except the normal linear regression commented out:

```

model
{
# Standardise x's and coefficients
  for (j in 1 : p) {
    b[j] <- beta[j] / sd(x[ , j ])
    for (i in 1 : N) {
      z[i, j] <- (x[i, j] - mean(x[, j])) / sd(x[, j])
    }
  }
  b0 <- beta0 - b[1] * mean(x[, 1]) - b[2] * mean(x[, 2]) - b[3] * mean(x[, 3])

# Model
  d <- 4;                      # degrees of freedom for t
  for (i in 1 : N) {
    Y[i] ~ dnorm(mu[i], tau)
#    Y[i] ~ ddexp(mu[i], tau)
#    Y[i] ~ dt(mu[i], tau, d)

    mu[i] <- beta0 + beta[1] * z[i, 1] + beta[2] * z[i, 2] + beta[3] * z[i, 3]
    stres[i] <- (Y[i] - mu[i]) / sigma
    outlier[i] <- step(stres[i] - 2.5) + step(-(stres[i] + 2.5) )
  }
# Priors
  beta0 ~ dnorm(0, 0.00001)
  for (j in 1 : p) {
    beta[j] ~ dnorm(0, 0.00001) # coeffs independent
#    beta[j] ~ dnorm(0, phi)      # coeffs exchangeable (ridge regression)
  }
  tau ~ dgamma(1.0E-3, 1.0E-3)
  phi ~ dgamma(1.0E-2, 1.0E-2)
# standard deviation of error distribution
  sigma <- sqrt(1 / tau)          # normal errors
#  sigma <- sqrt(2) / tau          # double exponential errors
#  sigma <- sqrt(d / (tau * (d - 2))); # t errors on d degrees of freedom
}

```

Data \Rightarrow list(p = 3, N = 21,
 $Y = c(42, 37, 37, 28, 18, 18, 19, 20, 15, 14, 14, 13, 11, 12, 8, 7, 8, 8, 9, 15, 15),$
 $x = structure(.Data = c($
 $80, 27, 89,$
 $80, 27, 88,$
 $75, 25, 90,$
 $62, 24, 87,$
 $62, 22, 87,$
 $62, 23, 87,$
 $62, 24, 93,$
 $62, 24, 93,$
 $58, 23, 87,$
 $58, 18, 80,$
 $58, 18, 89,$
 $58, 17, 88,$
 $58, 18, 82,$
 $58, 19, 93,$

```

50, 18, 89,
50, 18, 86,
50, 19, 72,
50, 19, 79,
50, 20, 80,
56, 20, 82,
70, 20, 91), .Dim = c(21, 3))) $\leftarrow$ 

```

Inits \leftarrow

```
list(beta0 = 10, beta=c(0,0, 0), tau = 0.1, phi = 0.1) $\leftarrow$ 
```

Results

a) Normal error

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
b[1]	0.7199	0.1457	0.003132	0.4271	0.7216	1.004	1001	10000
b[2]	1.276	0.3968	0.008096	0.4859	1.275	2.072	1001	10000
b[3]	-0.1494	0.1672	0.002384	-0.4738	-0.1496	0.183	1001	10000
b0	-40.0	12.69	0.1544	-65.49	-39.91	-14.85	1001	10000
outlier[3]	0.0115	0.1066	9.361E-4	0.0	0.0	0.0	1001	10000
outlier[4]	0.0523	0.2226	0.002265	0.0	0.0	1.0	1001	10000
outlier[21]	0.333	0.4713	0.006568	0.0	0.0	1.0	1001	10000
sigma	3.405	0.6252	0.008113	2.451	3.311	4.889	1001	10000

b) Double exponential error

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
b[1]	0.8298	0.1323	0.003019	0.5651	0.8305	1.098	1001	10000
b[2]	0.7633	0.3425	0.007578	0.1628	0.7271	1.503	1001	10000
b[3]	-0.1179	0.1211	0.001852	-0.3672	-0.115	0.1181	1001	10000
b0	-38.65	8.956	0.1024	-57.03	-38.68	-20.67	1001	10000
outlier[3]	0.0552	0.2284	0.002372	0.0	0.0	1.0	1001	10000
outlier[4]	0.284	0.4509	0.005483	0.0	0.0	1.0	1001	10000
outlier[21]	0.5837	0.4929	0.007195	0.0	1.0	1.0	1001	10000
sigma	3.497	0.865	0.009522	2.188	3.371	5.517	1001	10000

c) t4 error

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
b[1]	0.8328	0.1425	0.003777	0.537	0.8352	1.113	1001	10000
b[2]	0.8639	0.379	0.009468	0.1656	0.8424	1.677	1001	10000
b[3]	-0.1228	0.131	0.001939	-0.391	-0.1198	0.1339	1001	10000
b0	-40.39	9.879	0.1168	-60.65	-40.39	-20.89	1001	10000
outlier[3]	0.0391	0.1938	0.002708	0.0	0.0	1.0	1001	10000
outlier[4]	0.2262	0.4184	0.007121	0.0	0.0	1.0	1001	10000
outlier[21]	0.5801	0.4935	0.009352	0.0	1.0	1.0	1001	10000
sigma	3.512	0.876	0.01613	2.134	3.401	5.501	1001	10000

d) Normal error ridge regression

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
b[1]	0.6854	0.1346	0.002574	0.4153	0.6861	0.946	1001	10000
b[2]	1.3	0.3657	0.006805	0.5882	1.299	2.021	1001	10000
b[3]	-0.124	0.1642	0.001773	-0.4487	-0.1263	0.205	1001	10000
b0	-40.62	12.44	0.1193	-65.49	-40.72	-15.94	1001	10000
outlier[3]	0.0157	0.1243	0.001289	0.0	0.0	0.0	1001	10000
outlier[4]	0.0492	0.2163	0.002448	0.0	0.0	1.0	1001	10000
outlier[21]	0.2924	0.4549	0.006248	0.0	0.0	1.0	1001	10000
sigma	3.385	0.6103	0.006601	2.422	3.305	4.782	1001	10000

e) Double exponential error ridge regression

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

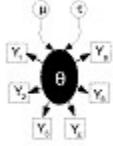
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
b[1]	0.7887	0.1286	0.002965	0.5211	0.792	1.036	1001	10000
b[2]	0.8026	0.3343	0.006942	0.2263	0.7713	1.543	1001	10000
b[3]	-0.09646	0.1183	0.001925	-0.3341	-0.09265	0.1376	1001	10000
b0	-38.94	8.791	0.1117	-56.88	-38.87	-21.47	1001	10000
outlier[3]	0.0812	0.2731	0.002672	0.0	0.0	1.0	1001	10000
outlier[4]	0.2829	0.4504	0.00556	0.0	0.0	1.0	1001	10000
outlier[21]	0.5177	0.4997	0.007443	0.0	1.0	1.0	1001	10000
sigma	3.513	0.895	0.01198	2.176	3.372	5.644	1001	10000

f) t4 error ridge regression

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
b[1]	0.7862	0.1411	0.003918	0.4908	0.7917	1.048	1001	10000
b[2]	0.9246	0.3601	0.009679	0.2784	0.9023	1.705	1001	10000
b[3]	-0.1055	0.13	0.001928	-0.3705	-0.1042	0.1491	1001	10000
b0	-40.44	9.798	0.1159	-60.28	-40.32	-20.98	1001	10000
outlier[3]	0.0512	0.2204	0.003358	0.0	0.0	1.0	1001	10000
outlier[4]	0.2214	0.4152	0.008136	0.0	0.0	1.0	1001	10000
outlier[21]	0.5143	0.4998	0.01019	0.0	1.0	1.0	1001	10000
sigma	3.516	0.8844	0.01688	2.133	3.391	5.571	1001	10000

We note the similar results between the Birkes and Dodge methods and BUGS, and the lack of influence of the ridge technique in this context.



BUGS Epil: repeated measures on Poisson counts

Breslow and Clayton (1993) analyse data initially provided by Thall and Vail (1990) concerning seizure counts in a randomised trial of anti-convulsant therapy in epilepsy. The table below shows the successive seizure counts for 59 patients. Covariates are treatment (0,1), 8-week baseline seizure counts, and age in years. The structure of this data is shown below

Patient	y_1	y_2	y_3	y_4	Trt	Base	Age
1	5	3	3	3	0	11	31
2	3	5	3	3	0	11	30
3	2	4	0	5	0	6	25
4	4	4	1	4	0	8	36
....							
8	40	20	21	12	0	52	42
9	5	6	6	5	0	12	37
....							
59	1	4	3	2	1	12	37

We consider model III of Breslow and Clayton (1993), in which Base is transformed to $\log(\text{Base}/4)$ and Age to $\log(\text{Age})$, and a Treatment by $\log(\text{Base}/4)$ interaction is included. Also present are random effects for both individual subjects b_{1j} and also subject by visit random effects b_{jk} to model extra-Poisson variability within subjects. $V4$ is an indicator variable for the 4th visit.

$$y_{jk} \sim \text{Poisson}(\mu_{jk})$$

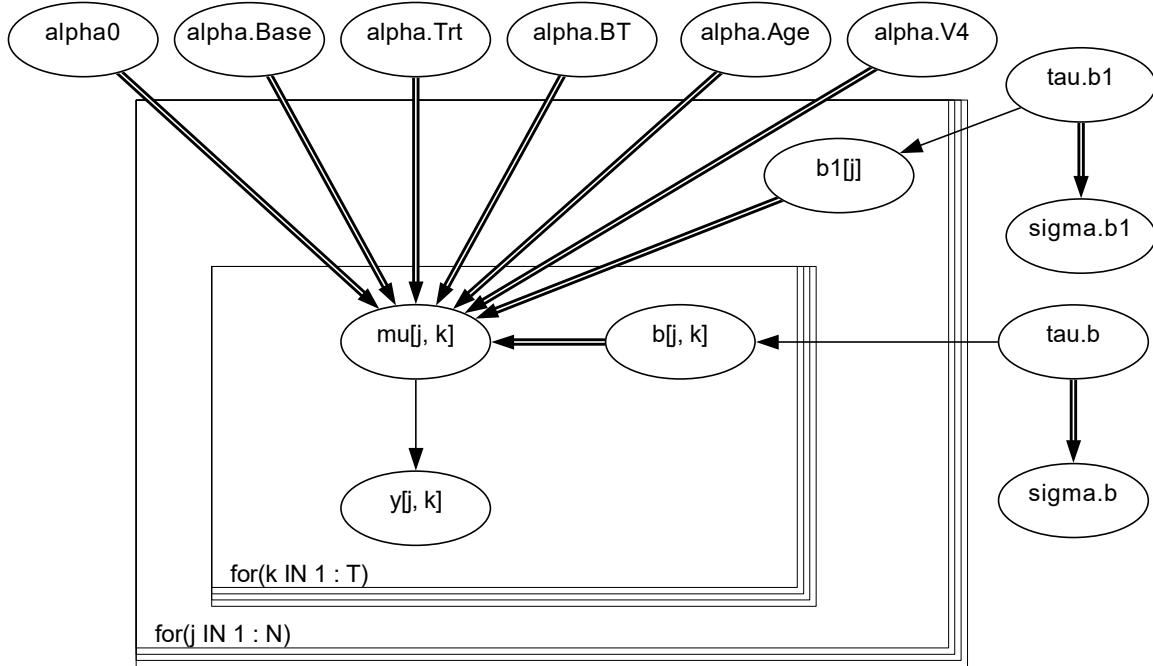
$$\begin{aligned} \log\mu_{jk} = & \alpha_0 + \alpha_{\text{Base}} \log(\text{Base}_j / 4) + \alpha_{\text{Trt}} \text{Trt}_j + \alpha_{\text{BT}} \text{Trt}_j \log(\text{Base}_j / 4) + \\ & \alpha_{\text{Age}} \text{Age}_j + \alpha_{V4} V_4 + b_{1j} + b_{jk} \end{aligned}$$

$$b_{1j} \sim \text{Normal}(0, \tau_{b1})$$

$$b_{jk} \sim \text{Normal}(0, \tau_b)$$

Coefficients and precisions are given independent "noninformative" priors.

The graphical model is below



The model shown above leads to a Markov chain that is highly correlated with poor convergence properties. This can be overcome by standardizing each covariate about its mean to ensure approximate prior independence between the regression coefficients as show below:

*BUGS language for epil example model III with covariate centering
(centering interaction term BT about mean(BT)):*

```

model
{
  for(j in 1 : N) {
    for(k in 1 : T) {
      log(mu[j, k]) <- a0 + alpha.Base * (log.Base4[j] - log.Base4.bar)
        + alpha.Trt * (Trt[j] - Trt.bar)
        + alpha.BT * (BT[j] - BT.bar)
        + alpha.Age * (log.Age[j] - log.Age.bar)
        + alpha.V4 * (V4[k] - V4.bar)
        + b1[j] + b[j, k]
      y[j, k] ~ dpois(mu[j, k])
      b[j, k] ~ dnorm(0.0, tau.b);    # subject*visit random effects
    }
    b1[j] ~ dnorm(0.0, tau.b1)    # subject random effects
    BT[j] <- Trt[j] * log.Base4[j] # interaction
    log.Base4[j] <- log(Base[j] / 4) log.Age[j] <- log(Age[j])
  }
}

# covariate means:
log.Age.bar <- mean(log.Age[])
Trt.bar <- mean(Trt[])
BT.bar <- mean(BT[])
log.Base4.bar <- mean(log.Base4[])
V4.bar <- mean(V4[])
# priors:
a0 ~ dnorm(0.0, 1.0E-4)
alpha.Base ~ dnorm(0.0, 1.0E-4)
alpha.Trt ~ dnorm(0.0, 1.0E-4);
alpha.BT ~ dnorm(0.0, 1.0E-4)
alpha.Age ~ dnorm(0.0, 1.0E-4)
alpha.V4 ~ dnorm(0.0, 1.0E-4)
tau.b1 ~ dgamma(1.0E-3, 1.0E-3); sigma.b1 <- 1.0 / sqrt(tau.b1)
tau.b ~ dgamma(1.0E-3, 1.0E-3); sigma.b <- 1.0 / sqrt(tau.b)

# re-calculate intercept on original scale:
alpha0 <- a0 - alpha.Base * log.Base4.bar - alpha.Trt * Trt.bar
- alpha.BT * BT.bar - alpha.Age * log.Age.bar - alpha.V4 * V4.bar
}

```

Data \Leftrightarrow list(N = 59, T = 4,

```

y = structure(.Data = c( 5, 3, 3, 3,
3, 5, 3, 3,
2, 4, 0, 5,
4, 4, 1, 4,
7, 18, 9, 21,
5, 2, 8, 7,
6, 4, 0, 2,
40, 20, 21, 12,
5, 6, 6, 5,
14, 13, 6, 0,

```

```

26, 12, 6, 22,
12, 6, 8, 4,
4, 4, 6, 2,
7, 9, 12, 14,
16, 24, 10, 9,
11, 0, 0, 5,
0, 0, 3, 3,
37, 29, 28, 29,
3, 5, 2, 5,
3, 0, 6, 7,
3, 4, 3, 4,
3, 4, 3, 4,
2, 3, 3, 5,
8, 12, 2, 8,
18, 24, 76, 25,
2, 1, 2, 1,
3, 1, 4, 2,
13, 15, 13, 12,
11, 14, 9, 8,
8, 7, 9, 4,
0, 4, 3, 0,
3, 6, 1, 3,
2, 6, 7, 4,
4, 3, 1, 3,
22, 17, 19, 16,
5, 4, 7, 4,
2, 4, 0, 4,
3, 7, 7, 7,
4, 18, 2, 5,
2, 1, 1, 0,
0, 2, 4, 0,
5, 4, 0, 3,
11, 14, 25, 15,
10, 5, 3, 8,
19, 7, 6, 7,
1, 1, 2, 3,
6, 10, 8, 8,
2, 1, 0, 0,
102, 65, 72, 63,
4, 3, 2, 4,
8, 6, 5, 7,
1, 3, 1, 5,
18, 11, 28, 13,
6, 3, 4, 0,
3, 5, 4, 3,
1, 23, 19, 8,
2, 3, 0, 1,
0, 0, 0, 0,
1, 4, 3, 2), .Dim = c(59, 4)),
Trt = c( 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1),
Base = c( 11, 11, 6, 8, 66, 27, 12, 52, 23, 10,
52, 33, 18, 42, 87, 50, 18, 111, 18, 20,
12, 9, 17, 28, 55, 9, 10, 47, 76, 38,
19, 10, 19, 24, 31, 14, 11, 67, 41, 7,
22, 13, 46, 36, 38, 7, 36, 11, 151, 22,
41, 32, 56, 24, 16, 22, 25, 13, 12),
Age = c(31,30,25,36,22,29,31,42,37,28,
36,24,23,36,26,26,28,31,32,21,
29,21,32,25,30,40,19,22,18,32,
20,30,18,24,30,35,27,20,22,28,
23,40,33,21,35,25,26,25,22,32,
25,35,21,41,32,26,21,36,37),
V4 = c(0, 0, 0, 1))
```

Inits ↵

```
list(a0 = 1, alpha.Base = 0, alpha.Trt = 0, alpha.BT = 0,
alpha.Age = 0, alpha.V4 = 0, tau.b1 = 1, tau.b =1) ↵
```

Results

A burn in of 5000 updates followed by a further 10000 updates gave the following parameter estimates

a) Without over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
alpha.Age	0.4921	0.3721	0.0162	-0.223	0.4843	1.263
alpha.BT	0.3394	0.2104	0.01922	-0.04543	0.3529	0.7535
alpha.Base	0.8806	0.1396	0.01068	0.5825	0.883	1.153
alpha.Trt	-0.9319	0.4157	0.03647	-1.757	-0.949	-0.1474
alpha.V4	-0.1032	0.08565	0.001771	-0.2692	-0.1028	0.0647
alpha0	-1.421	1.254	0.05049	-3.966	-1.399	1.031
sigma.b	0.3642	0.04391	0.001713	0.2843	0.3622	0.4548
sigma.b1	0.4976	0.07162	0.002429	0.3685	0.4929	0.6504

b) With over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
alpha.Age	0.4925	0.3546	0.008292	-0.1951	0.4931	1.192
alpha.BT	0.365	0.1831	0.0121	0.01216	0.3604	0.7317
alpha.Base	0.8745	0.1287	0.006416	0.6267	0.8755	1.134
alpha.Trt	-0.9807	0.3679	0.02319	-1.71	-0.9779	-0.2629
alpha.V4	-0.1008	0.08732	7.052E-4	-0.2722	-0.1014	0.07285
alpha0	-1.415	1.192	0.03038	-3.783	-1.412	0.8756
sigma.b	0.3639	0.04442	0.001133	0.281	0.3626	0.4546
sigma.b1	0.495	0.06981	0.001072	0.3714	0.4902	0.6428

These estimates can be compared with those of Breslow and Clayton (1993) who reported $\alpha_0 = -1.27 \pm 1.2$, $\alpha_{\text{Base}} = 0.86 \pm 0.13$, $\alpha_{\text{Trt}} = -0.93 \pm 0.40$, $\alpha_{\text{BT}} = 0.34 \pm 0.21$, $\alpha_{\text{Age}} = 0.47 \pm 0.35$, $\alpha_{\text{V4}} = -0.10 \pm 0.90$, $\sigma_{b1} = 0.48 \pm 0.06$, $\sigma_b = 0.36 \pm 0.04$.

*BUGS language for epil example model III with covariate centering
(alternative centering of interaction term BT):*

Ross (1990; p12) uses the parameterization $(x_i - m)^2 - \overline{(x_i - m)^2}$, where $m = \text{mean}(x)$ for centering 2nd order terms (i.e. x_i^2) in a polynomial regression. We consider a similar parameterisation for centering an interaction term $AB = A \times B$.

$$\begin{aligned}
AB.\text{centered}_i &= (A_i - A.\bar{b})(B_i - B.\bar{b}) - \text{mean}((A_i - A.\bar{b})(B_i - B.\bar{b})) \\
&= A_i B_i - A_i B.\bar{b} - B_i A.\bar{b} + A.\bar{b} B.\bar{b} - 1/N (\sum_i A_i B_i - A_i B.\bar{b} - B_i A.\bar{b} + A.\bar{b} B.\bar{b}) \\
&= AB_i - A_i B.\bar{b} - B_i A.\bar{b} + A.\bar{b} B.\bar{b} - AB.\bar{b} + A.\bar{b} B.\bar{b} + A.\bar{b} B.\bar{b} - A.\bar{b} B.\bar{b} \\
&= AB_i - AB.\bar{b} - A_i B.\bar{b} - B_i A.\bar{b} + 2A.\bar{b} B.\bar{b}
\end{aligned}$$

We apply this parameterisation to the Trt x log.Base4 interaction term for the Epilepsy example below:

```

model
{
  for(j in 1 : N) {
    for(k in 1 : T) {
      log(mu[j, k]) <- a0 + a.Base * (log.Base4[j] - log.Base4.bar)
      + a.Trt * (Trt[j] - Trt.bar)
      + alpha.BT * (BT[j] - BT.bar - Trt[j] * log.Base4.bar - log.Base4[j] * Trt.bar
                    + 2 * log.Base4.bar * Trt.bar)
      + alpha.Age * (log.Age[j] - log.Age.bar)
      + alpha.V4 * (V4[k] - V4.bar)
      + b1[j] + b[j, k]
      y[j, k] ~ dpois(mu[j, k])
      b[j, k] ~ dnorm(0.0, tau.b);    # subject*visit random effects
    }
    b1[j] ~ dnorm(0.0, tau.b1)    # subject random effects
    BT[j] <- Trt[j] * log.Base4[j]  # interaction
    log.Base4[j] <- log(Base[j] / 4) log.Age[j] <- log(Age[j])
  }
}

# covariate means:
log.Age.bar <- mean(log.Age[])
Trt.bar <- mean(Trt[])
BT.bar <- mean(BT[])
log.Base4.bar <- mean(log.Base4[])
V4.bar <- mean(V4[])
# priors:
a0 ~ dnorm(0.0, 1.0E-4)
a.Base ~ dnorm(0.0, 1.0E-4)
a.Trt ~ dnorm(0.0, 1.0E-4);
alpha.BT ~ dnorm(0.0, 1.0E-4)
alpha.Age ~ dnorm(0.0, 1.0E-4)
alpha.V4 ~ dnorm(0.0, 1.0E-4)
tau.b1 ~ dgamma(1.0E-3, 1.0E-3); sigma.b1 <- 1.0 / sqrt(tau.b1)
tau.b ~ dgamma(1.0E-3, 1.0E-3); sigma.b <- 1.0/ sqrt(tau.b)

# re-calculate intercept on original scale:
alpha0 <- a0 - a.Base * log.Base4.bar - a.Trt * Trt.bar
+ alpha.BT * (2 * log.Base4.bar * Trt.bar - BT.bar) - alpha.Age * log.Age.bar -
alpha.V4 * V4.bar
alpha.Base <- a.Base - alpha.BT * Trt.bar
alpha.Trt <- a.Trt - alpha.BT * log.Base4.bar
}

```

Initial values

```
list(a0 = 1, a.Base = 0, a.Trt = 0, alpha.BT = 0,
alpha.Age = 0, alpha.V4 = 0, tau.b1 = 1, tau.b = 1)
```

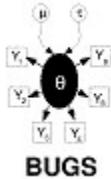
A burn in of 5000 updates followed by a further 10000 updates gave the following parameter estimates. Note the reduced autocorrelation for alpha.BT under this parameterisation.

a) Without over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
alpha.Age	0.4788	0.3676	0.01484	-0.2471	0.4899	1.196
alpha.BT	0.3428	0.2207	0.01229	-0.08613	0.3406	0.7982
alpha.Base	0.8795	0.1395	0.007519	0.5854	0.8854	1.135
alpha.Trt	-0.9413	0.4276	0.02191	-1.819	-0.9286	-0.1366
alpha.V4	-0.1005	0.08807	0.001625	-0.2736	-0.1002	0.07373
alpha0	-2.004	1.351	0.0588	-4.593	-2.037	0.7087
sigma.b	0.3658	0.04409	0.001732	0.2859	0.3629	0.4567
sigma.b1	0.4976	0.06934	0.001788	0.373	0.493	0.6476

b) With over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
alpha.Age	0.4841	0.3601	0.00654	-0.2259	0.4847	1.207
alpha.BT	0.3573	0.2121	0.005481	-0.0624	0.361	0.7664
alpha.Base	0.8817	0.1353	0.003002	0.618	0.8819	1.146
alpha.Trt	-0.9715	0.4138	0.009561	-1.789	-0.9756	-0.1633
alpha.V4	-0.103	0.08846	6.67E-4	-0.2755	-0.1022	0.07119
alpha0	-1.397	1.224	0.02257	-3.816	-1.387	1.007
sigma.b	0.3644	0.04392	0.001099	0.2833	0.3631	0.455
sigma.b1	0.4988	0.071	0.001151	0.3729	0.4941	0.6505



Blocker: random effects meta-analysis of clinical trials

Carlin (1992) considers a Bayesian approach to meta-analysis, and includes the following examples of 22 trials of beta-blockers to prevent mortality after myocardial infarction.

Study	Mortality: deaths / total	
	Treated	Control
1	3/38	3/39
2	7/114	14/116
3	5/69	11/93
4	102/1533	127/1520
.....		
20	32/209	40/218
21	27/391	43/364
22	22/680	39/674

In a random effects meta-analysis we assume the true effect (on a log-odds scale) δ_i in a trial i is drawn from some population distribution. Let r_{C_i} denote number of events in the control group in trial i , and r_{T_i} denote events under active treatment in trial i . Our model is:

$$r_{C_i} \sim \text{Binomial}(p_{C_i}, n_{C_i})$$

$$r_{T_i} \sim \text{Binomial}(p_{T_i}, n_{T_i})$$

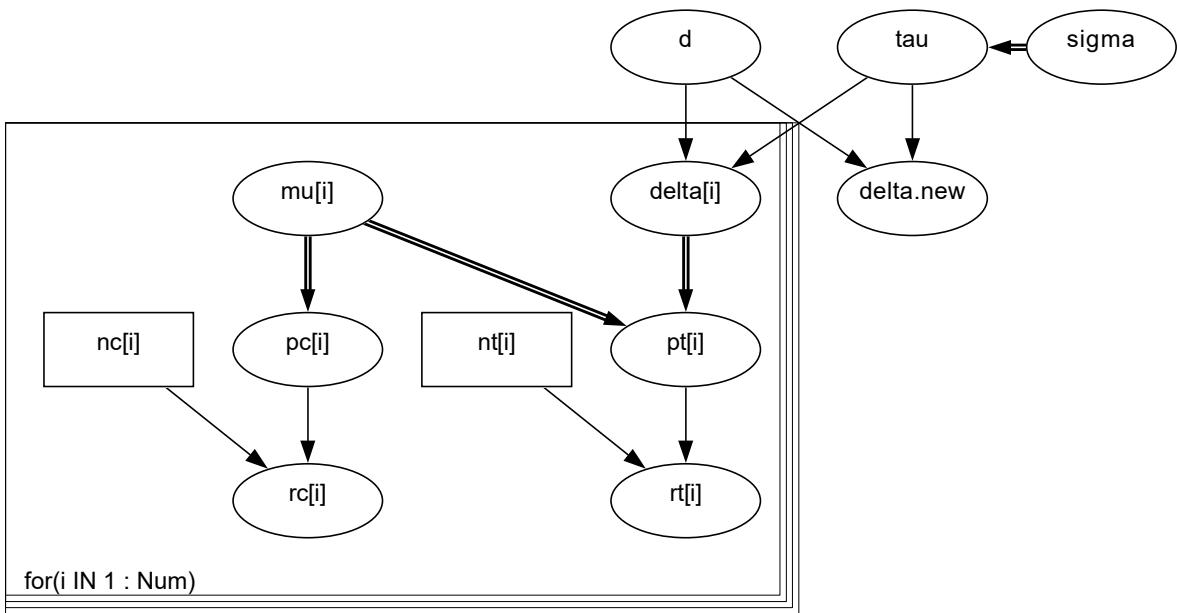
$$\text{logit}(p_{C_i}) = \mu_i$$

$$\text{logit}(p_{T_i}) = \mu_i + \delta_i$$

$$\delta_i \sim \text{Normal}(d, \tau)$$

"Noninformative" priors are given for the μ_i 's and d ; two alternative "noninformative" priors are considered for the random effects variance: prior 1 uses a $\text{Gamma}(0.001, 0.001)$ prior on the precision τ , while prior 2 assumes a proper uniform prior on the standard deviation σ . The graph for this model (with prior 2) is shown below. We want to make inferences about the population effect d , and the predictive distribution for the effect δ_{new} in a new trial. *Empirical Bayes* methods estimate d and τ by maximum likelihood and use these estimates to form the predictive distribution $p(\delta_{\text{new}} | d_{\hat{\text{hat}}}, \tau_{\hat{\text{hat}}})$. *Full Bayes* allows for the uncertainty concerning d and τ .

Graphical model for blocker example (with prior 2):



BUGS language for blocker example:

```

model
{
  for( i in 1 : Num ) {
    rc[i] ~ dbin(pc[i], nc[i])
    rt[i] ~ dbin(pt[i], nt[i])
    logit(pc[i]) <- mu[i]
    logit(pt[i]) <- mu[i] + delta[i]
    mu[i] ~ dnorm(0.0,1.0E-5)
    delta[i] ~ dnorm(d, tau)
  }
  d ~ dnorm(0.0,1.0E-6)
  # Choice of priors for random effects variance
  #tau ~ dgamma(0.001,0.001)
  #sigma <- 1 / sqrt(tau)
  tau<-1/(sigma*sigma)
  sigma~dunif(0,10)
  delta.new ~ dnorm(d, tau)
}

```

Data \Rightarrow

```

list(rt = c(3, 7, 5, 102, 28, 4, 98, 60, 25, 138, 64, 45, 9, 57, 25, 33, 28, 8, 6, 32, 27, 22),
  nt = c(38, 114, 69, 1533, 355, 59, 945, 632, 278, 1916, 873, 263, 291, 858, 154, 207, 251, 151, 174, 209, 391, 680),
  rc = c(3, 14, 11, 127, 27, 6, 152, 48, 37, 188, 52, 47, 16, 45, 31, 38, 12, 6, 3, 40, 43, 39),
  nc = c(39, 116, 93, 1520, 365, 52, 939, 471, 282, 1921, 583, 266, 293, 883, 147, 213, 122, 154, 134, 218, 364, 674),
  Num = 22)  $\Leftarrow$ 

```

Inits2 \Rightarrow

```
list(d = 0, delta.new = 0, sigma=1, mu = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),  
     delta = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)) \Leftrightarrow
```

Inits \Rightarrow

```
list(d = 0, delta.new = 0, tau=1, mu = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),  
     delta = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)) \Leftrightarrow
```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates using the gamma prior on τ

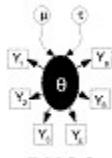
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
d	-0.2489	0.06282	0.002297	-0.3734	-0.248	-0.1239	1001	10000
delta.new	-0.2496	0.1576	0.002582	-0.5773	-0.2514	0.07974	1001	10000
sigma	0.1243	0.06834	0.002835	0.02878	0.1142	0.2796	1001	10000

Our estimates are lower and with tighter precision - in fact similar to the values obtained by Carlin for the empirical Bayes estimator. The discrepancy appears to be due to Carlin's use of a uniform prior for σ^2 in his analysis, which will lead to increased posterior mean and standard deviation for d, as compared to our use of a $\text{gamma}(0.001, 0.001)$ prior on the precision which is approximately equivalent to assuming $p(\sigma^2) \sim 1 / \sigma^2$ (see his Figure 1).

If we use a uniform prior on σ , the estimate of σ is slightly increased but there is little influence on the overall conclusions.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
d	-0.2484	0.06517	0.002568	-0.3699	-0.2493	-0.1161	1001	10000
delta.new	-0.2489	0.1692	0.002666	-0.5963	-0.2532	0.1073	1001	10000
sigma	0.1334	0.07934	0.00502	0.009759	0.1263	0.305	1001	10000

In some circumstances it might be reasonable to assume that the population distribution has heavier tails, for example a t distribution with low degrees of freedom. This is easily accomplished in BUGS by using the dt distribution function instead of dnorm for δ and δ_{new} .



Oxford: smooth fit to log-odds ratios

Breslow and Clayton (1993) re-analyse 2 by 2 tables of cases (deaths from childhood cancer) and controls tabulated against maternal exposure to X-rays, one table for each of 120 combinations of age (0-9) and birth year (1944-1964). The data may be arranged to the following form.

Strata	Exposure: X-ray / total Cases	Controls	age	year - 1954
1	3/28	0/28	9	-10
.....				
120	7/32	1/32	1	10

Their most complex model is equivalent to expressing the log(odds-ratio) ψ_i for the table in stratum i as

$$\log \psi_i = \alpha + \beta_1 \text{year}_i + \beta_2 (\text{year}_i^2 - 22) + b_i$$

$$b_i \sim \text{Normal}(0, \tau)$$

They use a quasi-likelihood approximation of the full hypergeometric likelihood obtained by conditioning on the margins of the tables.

We let $r_{\cdot i}^0$ denote number of exposures among the $n_{\cdot i}^0$ controls in stratum i , and $r_{\cdot i}^1$ denote number of exposures for the $n_{\cdot i}^1$ cases. Then we assume

$$r_{\cdot i}^0 \sim \text{Binomial}(p_{\cdot i}^0, n_{\cdot i}^0)$$

$$r_{\cdot i}^1 \sim \text{Binomial}(p_{\cdot i}^1, n_{\cdot i}^1)$$

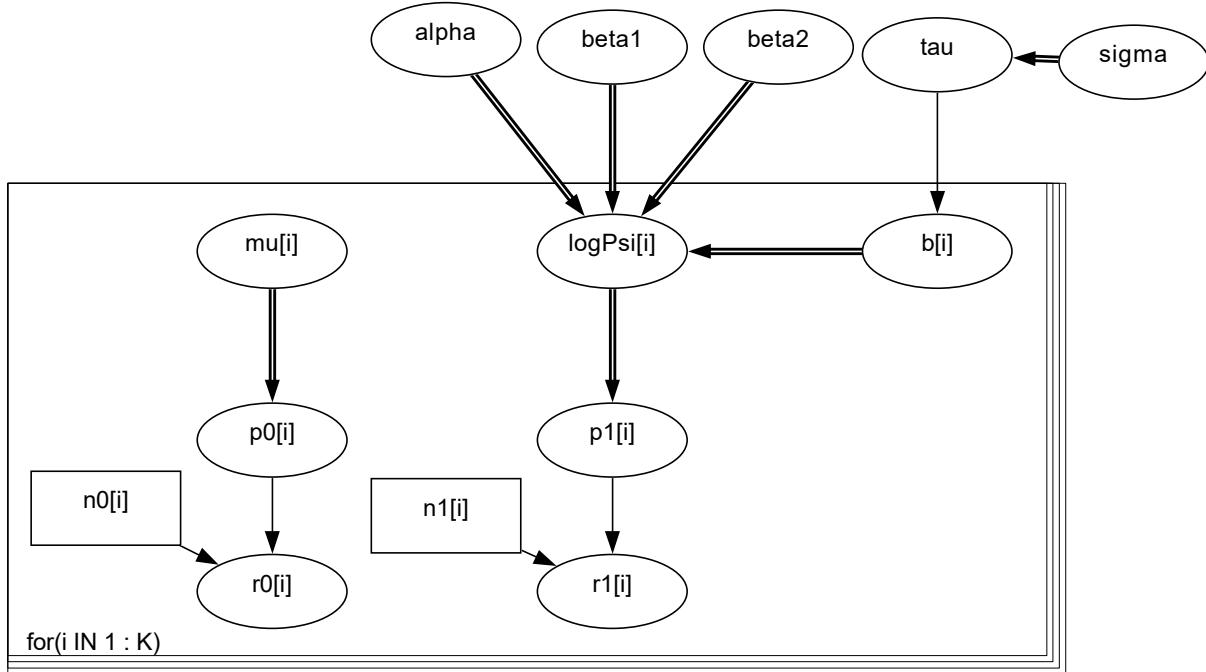
$$\text{logit}(p_{\cdot i}^0) = \mu_i$$

$$\text{logit}(p_{\cdot i}^1) = \mu_i + \log \psi_i$$

Assuming this model with independent vague priors for the μ_i 's provides the correct conditional likelihood.

Two alternative "noninformative" priors are considered for the random effects variance: prior 1 assumes a proper uniform prior on the standard deviation σ , while prior 2 uses a Gamma(0.001, 0.001) prior on the precision τ .

The appropriate graph (assuming prior 1) is shown below



BUGS language for Oxford example:

```

model
{
  for (i in 1 : K) {
    r0[i] ~ dbin(p0[i], n0[i])
    r1[i] ~ dbin(p1[i], n1[i])
    logit(p0[i]) <- mu[i]
    logit(p1[i]) <- mu[i] + logPsi[i]
    logPsi[i] <- alpha + beta1 * year[i] + beta2 * (year[i] * year[i] - 22) + b[i]
    b[i] ~ dnorm(0, tau)
    mu[i] ~ dnorm(0.0, 1.0E-6)
  }
  alpha ~ dnorm(0.0, 1.0E-6)
  beta1 ~ dnorm(0.0, 1.0E-6)
  beta2 ~ dnorm(0.0, 1.0E-6)
  # Choice of prior for random effects variance
  # Prior 1: uniform on SD
  sigma ~ dunif(0,100)
  tau <- 1/(sigma*sigma)

  #Prior 2:
  #tau ~ dgamma(1.0E-3, 1.0E-3);
  #sigma <- 1/sqrt(tau); # s.d. of random effects
}

```

```

Data ⇒list(r1 = c(3, 5, 2, 7, 7, 2, 5, 3, 5, 11, 6, 6, 11, 4, 4, 2, 8, 8, 6, 5, 15, 4, 9, 9, 4, 12, 8, 8,
  6, 8, 12, 4, 7, 16, 12, 9, 4, 7, 8, 11, 5, 12, 8, 17, 9, 3, 2, 7, 6, 5, 11, 14, 13, 8,
  6, 4, 8, 4, 8, 7, 15, 15, 9, 9, 5, 6, 3, 9, 12, 14, 16, 17, 8, 8, 9, 5, 9, 11, 6, 14,
  21, 16, 6, 9, 8, 9, 8, 4, 11, 11, 6, 9, 4, 4, 9, 9, 10, 14, 6, 3, 4, 6, 10, 4, 3, 3,
  10, 4, 10, 5, 4, 3, 13, 1, 7, 5, 7, 6, 3, 7),
n1 = c(28, 21, 32, 35, 35, 38, 30, 43, 49, 53, 31, 35, 46, 53, 61, 40, 29, 44, 52, 55, 61, 31, 48, 44, 42, 53, 56, 71,
  43, 43, 43, 40, 44, 70, 75, 71, 37, 31, 42, 46, 47, 55, 63, 91, 43, 39, 35, 32, 53, 49, 75, 64, 69, 64,
  49, 29, 40, 27, 48, 43, 61, 77, 55, 60, 46, 28, 33, 32, 46, 57, 56, 78, 58, 52, 31, 28, 46, 42, 45, 63,
  71, 69, 43, 50, 31, 34, 54, 46, 58, 62, 52, 41, 34, 52, 63, 59, 88, 62, 47, 53, 57, 74, 68, 61, 45, 45,
  62, 73, 53, 39, 45, 51, 55, 41, 53, 51, 42, 46, 54, 32),
r0 = c(0, 2, 2, 1, 2, 0, 1, 1, 1, 2, 4, 4, 2, 1, 7, 4, 3, 5, 3, 2, 4, 1, 4, 5, 2, 7, 5, 8,
  2, 3, 5, 4, 1, 6, 5, 11, 5, 2, 5, 8, 5, 6, 6, 10, 7, 5, 5, 2, 8, 1, 13, 9, 11, 9,
  4, 4, 8, 6, 8, 6, 8, 14, 6, 5, 5, 2, 4, 2, 9, 5, 6, 7, 5, 10, 3, 2, 1, 7, 9, 13,
  9, 11, 4, 8, 2, 3, 7, 4, 7, 5, 6, 6, 5, 6, 9, 7, 7, 4, 2, 3, 4, 10, 3, 4, 2,
  10, 5, 4, 5, 4, 6, 5, 3, 2, 2, 4, 6, 4, 1),
n0 = c(28, 21, 32, 35, 35, 38, 30, 43, 49, 53, 31, 35, 46, 53, 61, 40, 29, 44, 52, 55, 61, 31, 48, 44, 42, 53, 56, 71,
  43, 43, 43, 40, 44, 70, 75, 71, 37, 31, 42, 46, 47, 55, 63, 91, 43, 39, 35, 32, 53, 49, 75, 64, 69, 64,
  49, 29, 40, 27, 48, 43, 61, 77, 55, 60, 46, 28, 33, 32, 46, 57, 56, 78, 58, 52, 31, 28, 46, 42, 45, 63,
  71, 69, 43, 50, 31, 34, 54, 46, 58, 62, 52, 41, 34, 52, 63, 59, 88, 62, 47, 53, 57, 74, 68, 61, 45, 45,
  62, 73, 53, 39, 45, 51, 55, 41, 53, 51, 42, 46, 54, 32),
year = c(-10, -9, -9, -8, -8, -7, -7, -7, -6, -6, -6, -6, -5, -5, -5, -5, -5, -5, -4, -4, -4, -4, -4, -4, -4,
  -3, -3, -3, -3, -3, -3, -2, -2, -2, -2, -2, -2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
  -1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 6,
  6, 6, 6, 6, 7, 7, 7, 7, 8, 8, 8, 9, 9, 10), K = 120) ⇫

```

Inits1 \Rightarrow list(alpha=0,beta1=0,beta2=0,sigma=1,

) ←

Inits2 \Rightarrow list(alpha=0 beta1=0 beta2=0 tau=1)

```

b=c(0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0)
) ⇔

```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

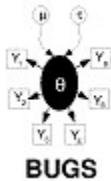
Prior 1: uniform on SD (preferred prior)

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	0.5825	0.06404	0.0013	0.4574	0.5826	0.707	1001	10000
beta1	-0.04678	0.01539	3.202E-4	-0.07628	-0.04678	-0.01669	1001	10000
beta2	0.00709	0.003127	7.765E-5	9.631E-4	0.007054	0.01333	1001	10000
sigma	0.1323	0.07116	0.005771	0.02874	0.1168	0.2912	1001	10000

There is little impact in using Prior 2:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	0.5795	0.06281	0.001574	0.4567	0.5794	0.7028	1001	10000
beta1	-0.04597	0.01541	3.267E-4	-0.0763	-0.04573	-0.01637	1001	10000
beta2	0.006996	0.003105	7.918E-5	0.00106	0.006959	0.01317	1001	10000
sigma	0.1173	0.07473	0.006238	0.02641	0.09903	0.2889	1001	10000

These estimates compare well with Breslow and Clayton (1993) PQL estimates of $\alpha = 0.566 \pm 0.070$, $\beta_1 = -0.469 \pm 0.0167$, $\beta_2 = 0.0071 \pm 0.0033$, $\sigma = 0.15 \pm 0.10$.



BUGS LSAT: item response

Section 6 of the Law School Aptitude Test (LSAT) is a 5-item multiple choice test; students score 1 on each item for the correct answer and 0 otherwise, giving $R = 32$ possible response patterns. Boch and Lieberman (1970) present data on LSAT for $N = 1000$ students, part of which is shown below.

Pattern index	Item response pattern	Freq (m)
1	0 0 0 0 0	3
2	0 0 0 0 1	6
3	0 0 0 1 0	2
.
.
.
30	1 1 1 0 1	61
31	1 1 1 1 0	28
32	1 1 1 1 1	298

The above data may be analysed using the one-parameter Rasch model (see Andersen (1980), pp.253-254; Boch and Aitkin (1981)). The probability p_{jk} that student j responds correctly to item k is assumed to follow a logistic function parameterized by an 'item difficulty' or threshold parameter α_k and a latent variable θ_j representing the student's underlying ability. The ability parameters are assumed to have a Normal distribution in the population of students. That is:

$$\text{logit}(p_{jk}) = \theta_j - \alpha_k, \quad j = 1, \dots, 1000; k = 1, \dots, 5$$

$$\theta_j \sim \text{Normal}(0, \tau)$$

The above model is equivalent to the following random effects logistic regression:

$$\text{logit}(p_{jk}) = \beta\theta_j - \alpha_k, \quad j = 1, \dots, 1000; k = 1, \dots, 5$$

$$\theta_j \sim \text{Normal}(0, 1)$$

where β corresponds to the scale parameter ($\beta^2 = \tau$) of the latent ability distribution. We assume a half-normal distribution with small precision for β ; this represents vague prior information but constrains β to be positive. Standard vague normal priors are assumed for the α_k 's. Note that the location of the α_k 's depend upon the mean of the prior distribution for θ_j which we have arbitrarily fixed to be zero. Alternatively, Boch and Aitkin ensure identifiability by imposing a sum-to-zero constraint on the α_k 's. Hence we calculate $a_k = \alpha_k - \bar{\alpha}$ to enable comparison of the BUGS posterior parameter estimates with the Boch and Aitkin marginal maximum likelihood estimates.

```

model
{
# Calculate individual (binary) responses to each test from multinomial data
  for (j in 1 : culm[1]) {
    for (k in 1 : T) { r[j, k] <- response[1, k] }
  }
  for (i in 2 : R) {
    for (j in culm[i - 1] + 1 : culm[i]) {
      for (k in 1 : T) { r[j, k] <- response[i, k] }
    }
  }
# Rasch model
  for (j in 1 : N) {
    for (k in 1 : T) {
      logit(p[j, k]) <- beta * theta[j] - alpha[k]
      r[j, k] ~ dbern(p[j, k])
    }
    theta[j] ~ dnorm(0, 1)
  }
# Priors
  for (k in 1:T) {
    alpha[k] ~ dnorm(0, 0.0001);  a[k] <- alpha[k] - mean(alpha[])
  }
  beta ~ dnorm(0,0.0001) I(0,)
}

```

Note that the data are read into *BUGS* in the original multinomial format to economize on space and effort. The 5 times 1000 individual binary responses for each item and student are then created within *BUGS* using the index variable *culm* (read in from the data file), where *culm*[*i*] = cumulative number of students recording response patterns 1, 2, ..., *i*: *i* <= R.

Data \Rightarrow list($N = 1000$, $R = 32$, $T = 5$)

```

culm = c(3, 9, 11, 22, 23, 24, 27, 31, 32, 40, 40, 56, 56, 59, 61, 76, 86, 115, 129, 210, 213, 241, 256, 336, 352, 408,
       429, 602, 613, 674, 702, 1000),
response = structure(.Data = c(
  0, 0, 0, 0, 0,
  0, 0, 0, 0, 1,
  0, 0, 0, 1, 0,
  0, 0, 0, 1, 1,
  0, 0, 1, 0, 0,
  0, 0, 1, 0, 1,
  0, 0, 1, 1, 0,
  0, 0, 1, 1, 1,
  0, 1, 0, 0, 0,
  0, 1, 0, 0, 1,
  0, 1, 0, 1, 0,
  0, 1, 0, 1, 1,
  0, 1, 1, 0, 0,
  0, 1, 1, 0, 1,
  0, 1, 1, 1, 0,
  0, 1, 1, 1, 1,
  1, 0, 0, 0, 0,
  1, 0, 0, 0, 1,
  1, 0, 0, 1, 0,
  1, 0, 0, 1, 1,
  1, 0, 1, 0, 0,
  1, 0, 1, 0, 1
), .Label = c("0", "1"))

```

```

1, 0, 1, 1, 0,
1, 0, 1, 1, 1,
1, 1, 0, 0, 0,
1, 1, 0, 0, 1,
1, 1, 0, 1, 0,
1, 1, 0, 1, 1,
1, 1, 1, 0, 0,
1, 1, 1, 0, 1,
1, 1, 1, 1, 0,
1, 1, 1, 1, 1), .Dim = c(32, 5))
)♀

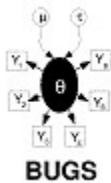
```

Inits ↳list(alpha = c(0,0,0,0,0), beta = 1)♀

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
a[1]	-1.262	0.1043	9.784E-4	-1.469	-1.26	-1.062	1001	10000
a[2]	0.4782	0.07003	6.559E-4	0.3401	0.4782	0.6163	1001	10000
a[3]	1.241	0.06878	8.928E-4	1.107	1.24	1.376	1001	10000
a[4]	0.1695	0.07322	7.178E-4	0.02613	0.1695	0.3144	1001	10000
a[5]	-0.6268	0.0855	8.627E-4	-0.7925	-0.6266	-0.4633	1001	10000
beta	0.7632	0.07082	0.001667	0.6242	0.7639	0.9023	1001	10000



BUGS

Bones: latent trait model for multiple ordered catagorical responses

The concept of skeletal age (SA) arises from the idea that individuals mature at different rates: for any given chronological age (CA), the average SA in a sample of individuals should equal their CA, but with an inter-individual spread which reflects the differential rate of maturation. Roche et al (1975) have developed a model for predicting SA by calibrating 34 indicators (items) of skeletal maturity which may be observed in a radiograph. Each indicator is categorized with respect to its degree of maturity: 19 are binary items (i.e. 0 = immature or 1 = mature); 8 items have 3 grades (i.e. 0 = immature; 1 = partially mature; 2 = fully mature); 1 item has 4 ordered grades and the remaining 6 items have 5 ordered grades of maturity. Roche et al. calculated threshold parameters for the boundaries between grades for each indicator. For the binary items, there is a single threshold representing the CA at which 50% of individuals are mature for the indicator. Three-category items have 2 threshold parameters: the first corresponds to the CA at which 50% of individuals are either partially or fully mature for the indicator; the second is the CA at which 50% of individuals are fully mature. Four and five-category items have 3 and 4 threshold parameters respectively, which are interpreted in a similar manner to those for 3-category items. In addition, Roche et al. calculated a discriminability (slope) parameter for each item which reflects its rate of maturation. Part of this data is shown below. Columns 1–4 represent the threshold parameters (note the use of the missing value code NA to 'fill in' the columns for items with fewer than 4 thresholds); column 5 is the discriminability parameter; column 6 gives the number of grades per item.

Threshold parameters				Discriminability	Num grades
0.7425	NA	NA	NA	2.9541	2
10.2670	NA	NA	NA	0.6603	2
10.5215	NA	NA	NA	0.7965	2
9.3877	NA	NA	NA	1.0495	2
0.2593	NA	NA	NA	5.7874	2
.
0.3887	1.0153	NA	NA	8.1123	3
3.2573	7.0421	NA	NA	0.9974	3
.
15.4750	16.9406	17.4944	NA	1.4297	4
.
5.0022	6.3704	8.2832	10.4988	1.0954	5
4.0168	5.1537	7.1053	10.3038	1.5329	5

Thissen (1986) (p.71) presents the following graded radiograph data on 13 boys whose chronological ages range from 6 months to 18 years. (Note that for ease of implementation in BUGS we have listed the items in a different order to that used by Thissen):

ID	CA	Maturity grades for items 1 - 32
1	0.6	1 1
2	1.0	2 1 1 1 2 2 1
.
12	16.0	2 3 3 3 1 NA 2 1 3 2 5 5 5 5 5
13	18.0	2 2 2 2 2 2 2 2 2 NA 2 2 2 2 2 2 2 3 3 3 NA 2 NA 2 3 4 5 5 5 5 5

Some items have missing data (represented by the code NA in the table above). This does not present a problem for *BUGS*: the missing grades are simply treated as unknown parameters to be estimated along with the other parameters of interest such as the SA for each boy.

This model uses the logistic function. For each item j and each grade k , the cumulative probability Q_{jk} that a boy with skeletal age θ is assigned a more mature grade than k is given by

$$\text{logit}Q_{jk} = \delta_j(\theta - \gamma_{jk})$$

where δ_j is the discriminability parameter and the γ_{jk} are the threshold parameters for item j . Hence the probability of observing an immature grade (i.e. $k = 1$) for a particular skeletal age θ is $p_{j,1} = 1 - Q_{j,1}$. The probability of observing a fully mature grade (i.e. $k = K_j$, where K_j is the number of grades for item j) is $p_{j,K_j} = Q_{j,K_j} - Q_{j,K_j-1}$. For items with 3 or more categories, the probability of observing an intermediate grade is $p_{j,k} = Q_{j,k} - Q_{j,k-1}$ (i.e. the difference between the cumulative probability of being assigned grade k or more, and of being assigned grade $k+1$ or more).

The *BUGS* language for this model is given below. Note that the θ_i for each boy i is assigned a vague, independent normal prior $\text{theta}[i] \sim \text{dnorm}(0.0, 0.001)$. That is, each boy is treated as a separate problem with no 'learning' or 'borrowing strength' across individuals, and hence no hierarchical structure on the θ_i 's.

BUGS language for bones example

```

model
{
  for (i in 1 : nChild) {
    theta[i] ~ dnorm(0.0, 0.001)
    for (j in 1 : nlnd) {
      # Cumulative probability of > grade k given theta
      for (k in 1: ncat[j] - 1) {
        logit(Q[i, j, k]) <- delta[j] * (theta[i] - gamma[j, k])
      }
    }

    # Probability of observing grade k given theta
    for (j in 1 : nlnd) {
      p[i, j, 1] <- 1 - Q[i, j, 1]
      for (k in 2 : ncat[j] - 1) {
        p[i, j, k] <- Q[i, j, k - 1] - Q[i, j, k]
      }
      p[i, j, ncat[j]] <- Q[i, j, ncat[j] - 1]
      grade[i, j] ~ dcat(p[i, j, 1 : ncat[j]])
    }
  }
}

```

Data \Rightarrow list(nChild = 13, nlnd = 34,
 $\gamma = \text{structure}($

Inits \Rightarrow list(theta = c(0.5, 1.2, 3.5, 6.7, 8.9, 12, 13, 16, 18))

We note a couple of tricks used in the above code. Firstly, the variable `p` has been declared as a 3-way rectangular array with the size of the third dimension equal to the maximum number of possible grades (i.e.5) for all items (even though items 1–28 have fewer than 5 categories). The statement

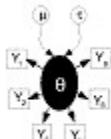
`grade[i, j] ~ dcat(p[i, j, 1 :ngrade[j]])`

is then used to select the relevant elements of $p[i,j,]$ for item j , thus ignoring any 'empty' spaces in the array for items with fewer than the maximum number of grades. Secondly, the final section of the above code includes a loop indexed as follows

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
theta[1]	0.3296	0.2053	0.002019	-0.0967	0.3371	0.7175	1001	10000
theta[2]	1.365	0.2548	0.002603	0.9055	1.35	1.912	1001	10000
theta[3]	2.348	0.2787	0.002852	1.804	2.346	2.902	1001	10000
theta[4]	2.902	0.2922	0.003024	2.34	2.901	3.478	1001	10000
theta[5]	5.523	0.5161	0.004614	4.518	5.513	6.567	1001	10000
theta[6]	6.755	0.6013	0.006727	5.639	6.742	7.977	1001	10000
theta[7]	6.449	0.5903	0.005852	5.337	6.436	7.629	1001	10000
theta[8]	8.917	0.6958	0.006558	7.56	8.914	10.29	1001	10000
theta[9]	8.972	0.6792	0.007061	7.625	8.978	10.26	1001	10000
theta[10]	11.96	0.6955	0.007166	10.63	11.96	13.36	1001	10000
theta[11]	11.57	0.9	0.0091	9.907	11.53	13.45	1001	10000
theta[12]	15.79	0.5696	0.006159	14.71	15.78	16.93	1001	10000
theta[13]	16.95	0.7378	0.006839	15.61	16.92	18.53	1001	10000



Inhaler: ordered categorical data

Ezzet and Whitehead (1993) analyse data from a two-treatment, two-period crossover trial to compare 2 inhalation devices for delivering the drug salbutamol in 286 asthma patients. Patients were asked to rate the clarity of leaflet instructions accompanying each device, using a 4-point ordinal scale. In the table below, the first entry in each cell (r,c) gives the number of subjects in Group 1 (who received device A in period 1 and device B in period 2) giving response r in period 1 and response c in period 2. The entry in brackets is the number of Group 2 subjects (who received the devices in reverse order) giving this response pattern.

		Response in period 2				TOTAL
		1 Easy	2 Only clear after re-reading	3 Not very clear	4 Confusing	
Response in period 1	1	59 (63)	35 (13)	3 (0)	2 (0)	99 (76)
	2	11 (40)	27 (15)	2 (0)	1 (0)	41 (55)
	3	0 (7)	0 (2)	0 (1)	0 (0)	0 (10)
	4	1 (2)	1 (0)	0 (1)	0 (0)	2 (3)
TOTAL		71 (112)	63 (30)	5 (2)	3 (0)	142 (144)

The response R_{it} from the i th subject ($i = 1, \dots, 286$) in the t th period ($t = 1, 2$) thus assumes integer values between 1 and 4. It may be expressed in terms of a continuous latent variable Y_{it} taking values on $(-\infty, \infty)$ as follows:

$$R_{it} = j \text{ if } Y_{it} \text{ in } [a_{j-1}, a_j), \quad j = 1, \dots, 4$$

where $a_0 = -\infty$ and $a_4 = \infty$. Assuming a logistic distribution with mean μ_{it} for Y_{it} , then the cumulative probability Q_{itj} of subject i rating the treatment in period t as worse than category j (i.e. $\text{Prob}(Y_{it} \geq a_j)$) is given by

$$\text{logit}Q_{itj} = -(a_j + \mu_{st} + b_i)$$

where b_i represents the random effect for subject i . Here, μ_{st} depends only on the period t and the sequence $s_i = 1, 2$ to which patient i belongs. It is defined as

$$\mu_{11} = \beta / 2 + \pi / 2$$

$$\mu_{12} = -\beta / 2 - \pi / 2 - \kappa$$

$$\mu_{21} = -\beta / 2 + \pi / 2$$

$$\mu_{22} = \beta / 2 - \pi / 2 + \kappa$$

where β represents the treatment effect, π represents the period effect and κ represents the carryover effect. The probability of subject i giving response j in period t is thus given by $p_{itj} = Q_{itj} - Q_{it0}$, where $Q_{it0} = 1$ and

$Q_{it4} = 0$ (see also the Bones example).

The *BUGS* language for this model is shown below. We assume the b_i 's to be normally distributed with zero mean and common precision τ . Two alternative "noninformative" priors are considered for the random effects variance: prior 1 is a uniform prior on the standard deviation, and prior 2 is a gamma(0.001, 0.001) prior on the precision. The fixed effects β , π and κ are given vague normal priors, as are the unknown cut points a_1 , a_2 and a_3 . We also impose order constraints on the latter using the $I(,)$ notation in *BUGS*, to ensure that $a_1 < a_2 < a_3$.

```

model
{
#
# Construct individual response data from contingency table
#
  for (i in 1 : Ncum[1, 1]) {
    group[i] <- 1
    for (t in 1 : T) { response[i, t] <- pattern[1, t] }
  }
  for (i in (Ncum[1,1] + 1) : Ncum[1, 2]) {
    group[i] <- 2 for (t in 1 : T) { response[i, t] <- pattern[1, t] }
  }

  for (k in 2 : Npattern) {
    for(i in (Ncum[k - 1, 2] + 1) : Ncum[k, 1]) {
      group[i] <- 1 for (t in 1 : T) { response[i, t] <- pattern[k, t] }
    }
    for(i in (Ncum[k, 1] + 1) : Ncum[k, 2]) {
      group[i] <- 2 for (t in 1 : T) { response[i, t] <- pattern[k, t] }
    }
  }
#
# Model
#
  for (i in 1 : N) {
    for (t in 1 : T) {
      for (j in 1 : Ncut) {
#
# Cumulative probability of worse response than j
#
        logit(Q[i, t, j]) <- -(a[j] + mu[group[i], t] + b[i])
      }
#
# Probability of response = j
#
        p[i, t, 1] <- 1 - Q[i, t, 1]
        for (j in 2 : Ncut) { p[i, t, j] <- Q[i, t, j - 1] - Q[i, t, j] }
        p[i, t, (Ncut+1)] <- Q[i, t, Ncut]

        response[i, t] ~ dcat(p[i, t, ])
      }
#
# Subject (random) effects
#
      b[i] ~ dnorm(0.0, tau)
    }

#
# Fixed effects
#
    for (g in 1 : G) {

```

Note that the data is read into *BUGS* in the original contingency table format to economize on space and effort. The individual responses for each of the 286 patients are then constructed within *BUGS*.

Data \Rightarrow list(N = 286, T = 2, G = 2, Npattern = 16, Ncut = 3,

pattern = structure(.Data =
c(1, 1,

1, 2,
1, 3,
1, 4,
2, 1,
2, 2,
2, 3,
2, 4,
3, 1,
3, 2,
3, 3,
3, 4,
4, 1,
4, 2,
4, 3,
4, 4), .Dim = c(16, 2)),

Ncum = structure(.Data =
c(59, 122,

157, 170,
173, 173,
175, 175,
186, 226,
253, 268,
270, 270,
271, 271,
271, 278,
278, 280,
280, 281,
281, 281,
282, 284,

285, 285,

285, 286,
286, 286), .Dim = c(16, 2)),

treat = structure(.Data =

c(1, -1,

-1, 1), .Dim = c(2, 2)),

period = structure(.Data =

c(1, -1,

1, -1), .Dim = c(2, 2)),

carry = structure(.Data =

c(0, -1,

0, 1), .Dim = c(2, 2))

) \Leftarrow

Inits1 \Rightarrow

list(beta = 0, pi = 0, kappa = 0, a = c(2, 3, 4), sigma = 1) \Leftarrow

Inits2 \Rightarrow

list(beta = 0, pi = 0, kappa = 0, a = c(2, 3, 4), tau = 1) \Leftarrow

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

Results using prior 1: uniform on SD

node	mean	sd	MC error 2.5%	median	97.5%	start	sample
------	------	----	---------------	--------	-------	-------	--------

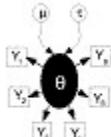
a[1]	0.7228	0.1402	0.004395	0.4621	0.7168	1.015	1001	10000
a[2]	3.964	0.3298	0.01577	3.361	3.951	4.661	1001	10000
a[3]	5.316	0.476	0.01889	4.446	5.293	6.297	1001	10000
beta	1.083	0.3246	0.008275	0.4594	1.074	1.73	1001	10000
kappa	0.2423	0.2532	0.005772	-0.2472	0.2422	0.7488	1001	10000
log.sigma	0.2174	0.1995	0.0129	-0.2546	0.2386	0.5544	1001	10000
pi	-0.2413	0.2	0.002374	-0.6431	-0.2361	0.1479	1001	10000
sigma	1.267	0.2405	0.01513	0.7753	1.269	1.741	1001	10000

Results using prior 2 (Gamma(0..001,0.001) show that prior has negligible impact on the results.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
a[1]	0.7079	0.1377	0.004319	0.4547	0.7022	0.9886	1001	10000
a[2]	3.91	0.3322	0.01624	3.299	3.899	4.597	1001	10000
a[3]	5.256	0.4678	0.0186	4.394	5.237	6.22	1001	10000
beta	1.047	0.3264	0.008527	0.4203	1.039	1.707	1001	10000
kappa	0.2532	0.2524	0.006044	-0.2383	0.2547	0.7513	1001	10000
log.sigma	0.1667	0.2279	0.01523	-0.3635	0.1984	0.5194	1001	10000
pi	-0.237	0.199	0.002517	-0.6342	-0.2365	0.1586	1001	10000
sigma	1.21	0.2539	0.01669	0.6953	1.219	1.681	1001	10000

The estimates can be compared with those of Ezzet and Whitehead, who used the Newton-Raphson method and numerical integration to obtain maximum-likelihood estimates of the parameters. They reported

$$\beta = 1.17 \pm 0.75, \pi = -0.23 \pm 0.20, \kappa = 0.21 \pm 0.49, \log\sigma = 0.17 \pm 0.23, a1 = 0.68, a2 = 3.85, a3 = 5.10$$



BUGS Mice: Weibull regression

Dellaportas and Smith (1993) analyse data from Grieve (1987) on photocarcinogenicity in four groups, each containing 20 mice, who have recorded a survival time and whether they died or were censored at that time. A * indicates censoring.

Mouse	Irradiated control	Vehicle control	Test substance	Positive control
1	12	32	22	27
.....				
18	*40	30	24	12
19	31	37	37	17
20	36	27	29	26

The survival distribution is assumed to be Weibull. That is

$$f(t_i, z_i) = r e^{\beta z_i} t_i^{r-1} \exp(-e^{\beta z_i} t_i^r)$$

where t_i is the failure time of an individual with covariate vector z_i and β is a vector of unknown regression coefficients. This leads to a baseline hazard function of the form

$$\lambda_0(t_i) = r t_i^{r-1}$$

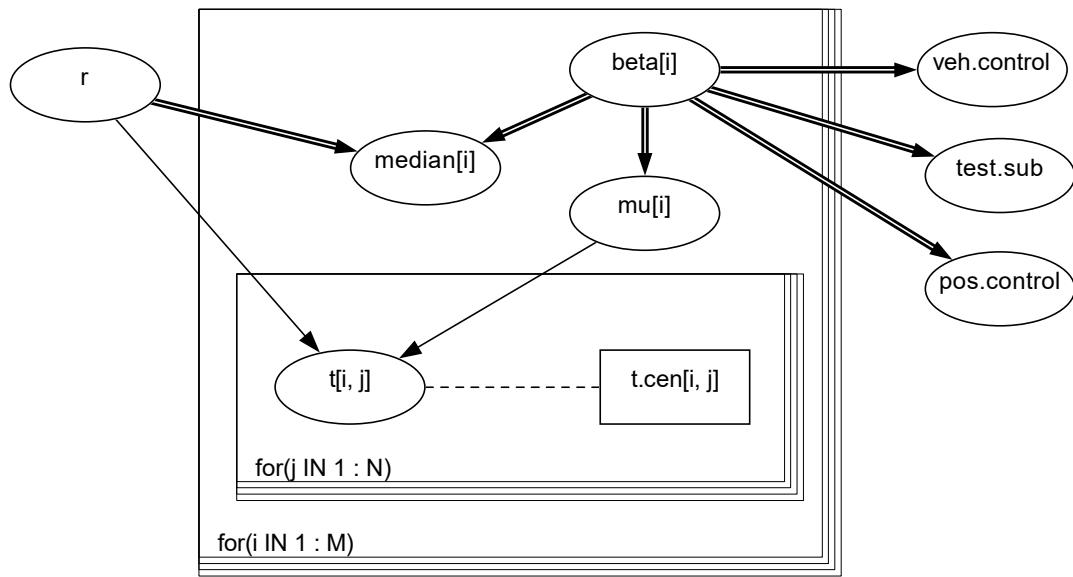
Setting $\mu_i = e^{\beta z_i}$ gives the parameterisation

$$t_i \sim \text{Weibull}(\tau, \mu_i)$$

For censored observations, the survival distribution is a truncated Weibull, with lower bound corresponding to the censoring time. The regression β coefficients were assumed a priori to follow independent Normal distributions with zero mean and "vague" precision 0.0001. The shape parameter r for the survival distribution was given a Gamma(1, 0.0001) prior, which is slowly decreasing on the positive real line.

Median survival for individuals with covariate vector z_i is given by $m_i = (\log 2 e^{-\beta z_i})^{1/r}$

The appropriate graph and BUGS language are below, using an undirected dashed line to represent a logical range constraint.



```

model
{
  for(i in 1 : M) {
    for(j in 1 : N) {
      t[i, j] ~ dweib(r, mu[i])I(t.cen[i, j],)
    }
    mu[i] <- exp(beta[i])
    beta[i] ~ dnorm(0.0, 0.001)
    median[i] <- pow(log(2) * exp(-beta[i]), 1/r)
  }
  r ~ dexp(0.001)
  veh.control <- beta[2] - beta[1]
  test.sub <- beta[3] - beta[1]
  pos.control <- beta[4] - beta[1]
}

```

Data \Rightarrow list(t)

```

c(12, 1, 21, 25, 11, 26, 27, 30, 13, 12, 21, 20, 23, 25, 23, 29, 35, NA, 31, 36,
 32, 27, 23, 12, 18, NA, NA, 38, 29, 30, NA, 32, NA, NA, NA, NA, 25, 30, 37, 27,
 22, 26, NA, 28, 19, 15, 12, 35, 35, 10, 22, 18, NA, 12, NA, NA, 31, 24, 37, 29,
 27, 18, 22, 13, 18, 29, 28, NA, 16, 22, 26, 19, NA, NA, 17, 28, 26, 12, 17, 26),
.Dim = c(4, 20)),
cen = structure(.Data =
  c( 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 40, 0, 0,
    0, 0, 0, 0, 40, 40, 0, 0, 0, 40, 0, 40, 40, 40, 40, 0, 0, 0, 0,
    0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 0, 40, 40, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 20, 0, 0, 0, 29, 10, 0, 0, 0, 0, 0, 0),

```

.Dim = c(4, 20)),
M = 4, N = 20) ↳

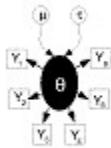
Inits ↳ list(beta = c(0, 0, 0, 0), r = 3) ↳

We note a number of tricks in setting up this model. First, individuals who are censored are given a missing value in the vector of failure times t, whilst individuals who fail are given a zero in the censoring time vector t.cen (see data file listing below). The truncated Weibull is modelled using $\text{I}(t.cen[i],)$ to set a lower bound. Second, we set a parameter beta[j] for each treatment group j. The contrasts beta[j] with group 1 (the irradiated control) are calculated at the end. Alternatively, we could have included a grand mean term in the relative risk model and constrained beta[1] to be zero.

Results

A burn in of 1000 updates followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
median[1]	23.94	1.967	0.05889	20.3	23.89	28.09	1001	10000
median[2]	35.21	3.359	0.04757	29.46	34.93	42.64	1001	10000
median[3]	26.91	2.383	0.0582	22.62	26.79	31.91	1001	10000
median[4]	21.43	1.799	0.03362	18.2	21.32	25.36	1001	10000
pos.control	0.3409	0.3457	0.007235	-0.3272	0.3429	1.009	1001	10000
r	3.037	0.3182	0.02749	2.388	3.045	3.64	1001	10000
test.sub	-0.3516	0.3459	0.004433	-1.035	-0.3541	0.3303	1001	10000
veh.control	-1.16	0.3679	0.005974	-1.893	-1.156	-0.4446	1001	10000



BUGS Kidney: Weibull regression with random effects

McGilchrist and Aisbett (1991) analyse time to first and second recurrence of infection in kidney patients on dialysis using a Cox model with a multiplicative frailty parameter for each individual. The risk variables considered are age, sex and underlying disease (coded other, GN, AN and PKD). A portion of the data are shown below.

Patient Number	Recurrence time t	Event (2 = cens)	Age at time t	Sex (1 = female)	Disease (0 = other; 1 = GN 2 = AN; 3 = PKD)
1	8,16	1,1	28,28	0	0
2	23,13	1,2	48,48	1	1
3	22,28	1,1	32,32	0	0
4	447,318	1,1	31,32	1	0
.....					
35	119,8	1,1	22,22	1	1
36	54,16	2,2	42,42	1	1
37	6,78	2,1	52,52	1	3
38	63,8	1,2	60,60	0	3

We have analysed the same data assuming a parametric Weibull distribution for the survivor function, and including an additive random effect b_i for each patient in the exponent of the hazard model as follows

$$t_{ij} \sim \text{Weibull}(r, \mu_{ij}) \quad i = 1, \dots, 38; j = 1, 2$$

$$\log \mu_{ij} = \alpha + \beta_{\text{age}} \text{AGE}_{ij} + \beta_{\text{sex}} \text{SEX}_i + \beta_{\text{disease1}} \text{DISEASE}_{i1} + \\ \beta_{\text{disease2}} \text{DISEASE}_{i2} + \beta_{\text{disease3}} \text{DISEASE}_{i3} + b_i$$

$$b_i \sim \text{Normal}(0, \tau)$$

where AGE_{ij} is a continuous covariate, SEX_i is a 2-level factor and DISEASE_{ik} ($k = 1, 2, 3$) are dummy variables representing the 4-level factor for underlying disease. Note that the survival distribution is a truncated Weibull for censored observations as discussed in the mice example. The regression coefficients are given "non-informative" priors, namely

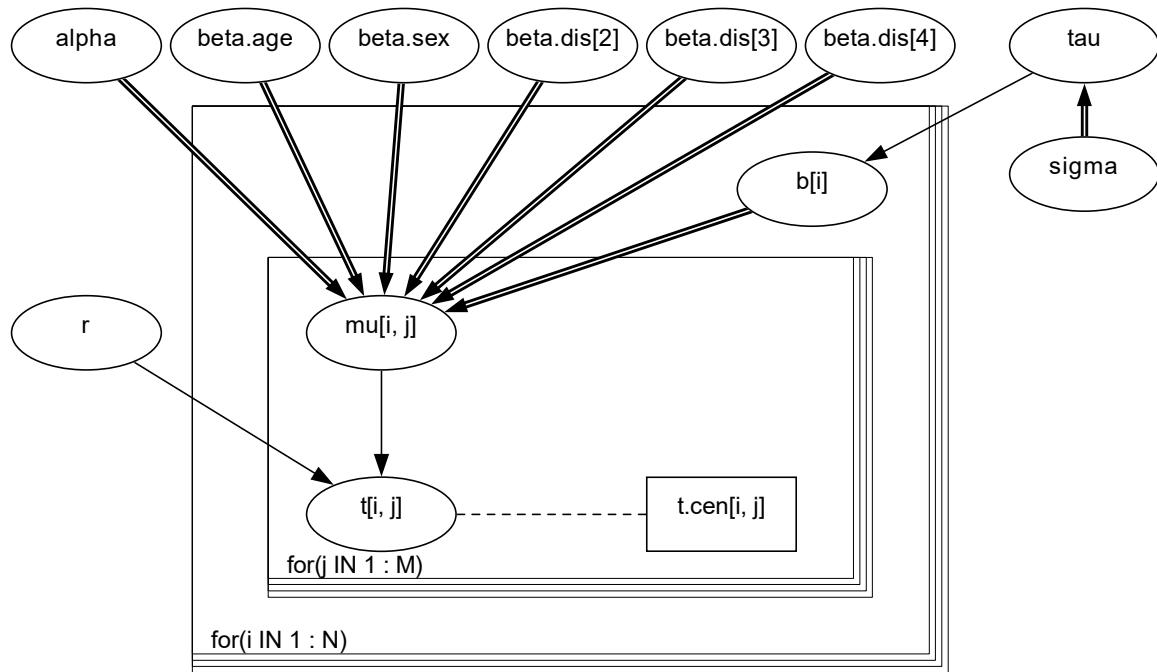
$$b_k \sim \text{Normal}(0, 0.0001)$$

Two alternative "noninformative" priors are considered for the random effects variance: prior 1 is a uniform prior on the standard deviation, and prior 2 is a gamma(0.001, 0.001) prior on the precision.

The shape parameter of the survival distribution r is given a Gamma(1, 0.0001) prior which is slowly decreasing on the positive real line.

The graphical model (with prior 1) and BUGS language are given below.

Graphical model for kidney example (with prior 1):



BUGS language for kidney example

```

model
{
  for (i in 1:N) {
    for (j in 1:M) {
      # Survival times bounded below by censoring times:
      t[i,j] ~ dweib(r,mu[i,j]) I(t.cen[i,j],);
      log(mu[i,j]) <- alpha + beta.age*age[i,j]
                  + beta.sex *sex[i]
                  + beta.dis[disease[i]] + b[i];
    }
  }
  # Random effects:
  b[i] ~ dnorm(0.0, tau)
}
# Priors:
alpha ~ dnorm(0.0, 0.0001);
beta.age ~ dnorm(0.0, 0.0001);
beta.sex ~ dnorm(0.0, 0.0001);
# beta.dis[1] <- 0; # corner-point constraint
for(k in 2 : 4) {
  beta.dis[k] ~ dnorm(0.0, 0.0001);
}
# Choice of priors for random effects variance
# Prior 1: uniform on SD
sigma~ dunif(0,100)
tau<-1/(sigma*sigma)

#Prior 2:
#tau ~ dgamma(1.0E-3, 1.0E-3);
#sigma <- 1/sqrt(tau); # s.d. of random effects

r ~ dgamma(1.0, 1.0E-3);

}

```

Data \Rightarrow list(N = 38, M = 2,

```

t = structure(
.Data = c( 8, 16,
23, NA,
22, 28,
447, 318,
30, 12,
24, 245,
7, 9,
511, 30,
53, 196,
15, 154,
7, 333,
141, NA,
96, 38,
NA, NA,
```

```

536, NA,
17, NA,
185, 177,
292, 114,
NA, NA,
15, NA,
152, 562,
402, NA,
13, 66,
39, NA,
12, 40,
NA, 201,
132, 156,
34, 30,
2, 25,
130, 26,
27, 58,
NA, 43,
152, 30,
190, NA,
119, 8,
NA, NA,
NA, 78,
63, NA), .Dim = c(38, 2)),
t.cen = structure(
.Data = c( 0,  0,
          0, 13,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0,
          0,  0),
.Dim = c(38, 2)),
age = structure(
.Data = c(28,  28,
        48, 48,
        32, 32,
        31, 32,
        10, 10,
        15, 15,
        12, 12,
        18, 18,
        22, 22,
        25, 25,
        20, 20,
        24, 24,
        26, 26,
        28, 28,
        30, 30,
        33, 33,
        35, 35,
        37, 37,
        39, 39,
        41, 41,
        43, 43,
        45, 45,
        47, 47,
        49, 49,
        51, 51,
        53, 53,
        55, 55,
        57, 57,
        59, 59,
        61, 61,
        63, 63,
        65, 65,
        67, 67,
        69, 69,
        71, 71,
        73, 73,
        75, 75,
        77, 77,
        79, 79,
        81, 81,
        83, 83,
        85, 85,
        87, 87,
        89, 89,
        91, 91,
        93, 93,
        95, 95,
        97, 97,
        99, 99),
.Dim = c(38, 2))

```

```

16, 17,
51, 51,
55, 56,
69, 69,
51, 52,
44, 44,
34, 34,
35, 35,
42, 42,
17, 17,
60, 60,
60, 60,
43, 44,
53, 53,
44, 44,
46, 47,
30, 30,
62, 63,
42, 43,
43, 43,
57, 58,
10, 10,
52, 52,
53, 53,
54, 54,
56, 56,
50, 51,
57, 57,
44, 45,
22, 22,
42, 42,
52, 52,
60, 60), .Dim = c(38, 2),
beta.dis = c(0, NA, NA, NA),
sex = c(0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1,
0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0),
disease = c(1, 2, 1, 1, 1, 1, 2, 2, 3, 2, 3, 1, 3, 3, 1, 3, 1, 1, 2, 1, 4, 1, 3, 3,
3, 3, 2, 3, 2, 3, 4, 2, 1, 1, 4, 4))↳
```

Inits1 \Rightarrow list(beta.age = 0, beta.sex = 0, beta.dis=c(NA,0,0,0),
alpha = 0, r=1, sigma=1)[↳]

Inits2 \Rightarrow list(beta.age = 0, beta.sex = 0, beta.dis=c(NA,0,0,0),
alpha = 0, r=1, tau=0.3)[↳]

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

Prior 1: Uniform on SD (preferred prior)

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	-5.128	1.129	0.09498	-7.514	-5.019	-3.156	1001	10000
beta.age	0.005224	0.01886	0.001422	-0.03003	0.005588	0.03987	1001	10000
beta.dis[2]	0.1631	0.633	0.03253	-0.9871	0.1367	1.516	1001	10000
beta.dis[3]	0.6814	0.6549	0.03616	-0.5294	0.6532	2.097	1001	10000
beta.dis[4]	-1.129	0.9339	0.04462	-2.908	-1.17	0.7911	1001	10000
beta.sex	-2.109	0.5718	0.03101	-3.246	-2.099	-1.026	1001	10000
r	1.333	0.1948	0.01601	1.007	1.314	1.757	1001	10000
sigma	0.923	0.3425	0.02356	0.2855	0.8982	1.659	1001	10000

Prior 2 (Gamma(0.001,0.001) leads to a slightly smaller estimate of sigma, and so provides slightly more precise estimates of the other parameters.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	-4.6	0.8962	0.07002	-6.541	-4.541	-3.065	1001	10000
beta.age	0.003027	0.01475	9.703E-4	-0.02411	0.002423	0.03368	1001	10000
beta.dis[2]	0.1329	0.5393	0.02148	-0.9443	0.1276	1.237	1001	10000
beta.dis[3]	0.6444	0.5301	0.02364	-0.4158	0.6428	1.711	1001	10000
beta.dis[4]	-1.168	0.8335	0.03382	-2.772	-1.182	0.5535	1001	10000
beta.sex	-1.938	0.4854	0.02524	-2.952	-1.917	-1.033	1001	10000
r	1.215	0.1623	0.01293	0.9275	1.204	1.536	1001	10000
sigma	0.6374	0.357	0.02847	0.04597	0.653	1.322	1001	10000



BUGS Leuk: Cox regression

Several authors have discussed Bayesian inference for censored survival data where the integrated baseline hazard function is to be estimated non-parametrically Kalbfleisch (1978), Kalbfleisch and Prentice (1980), Clayton (1991), Clayton (1994). Clayton (1994) formulates the Cox model using the counting process notation introduced by Andersen and Gill (1982) and discusses estimation of the baseline hazard and regression parameters using MCMC methods. Although his approach may appear somewhat contrived, it forms the basis for extensions to random effect (frailty) models, time-dependent covariates, smoothed hazards, multiple events and so on. We show below how to implement this formulation of the Cox model in *BUGS*.

For subjects $i = 1, \dots, n$, we observe processes $N_i(t)$ which count the number of failures which have occurred up to time t . The corresponding intensity process $I_i(t)$ is given by

$$I_i(t)dt = E(dN_i(t) | F_{t-})$$

where $dN_i(t)$ is the increment of N_i over the small time interval $[t, t+dt]$, and F_{t-} represents the available data just before time t . If subject i is observed to fail during this time interval, $dN_i(t)$ will take the value 1; otherwise $dN_i(t) = 0$. Hence $E(dN_i(t) | F_{t-})$ corresponds to the probability of subject i failing in the interval $[t, t+dt]$. As $dt \rightarrow 0$ (assuming time to be continuous) then this probability becomes the instantaneous hazard at time t for subject i . This is assumed to have the proportional hazards form

$$I_i(t) = Y_i(t)\lambda_0(t)\exp(\beta z_i)$$

where $Y_i(t)$ is an observed process taking the value 1 or 0 according to whether or not subject i is observed at time t and $\lambda_0(t)\exp(\beta z_i)$ is the familiar Cox regression model. Thus we have observed data $D = N_i(t)$, $Y_i(t)$, z_i , $i = 1, \dots, n$ and unknown parameters β and $\Lambda_0(t) = \text{Integral}(\lambda_0(u), u, t, 0)$, the latter to be estimated non-parametrically.

The joint posterior distribution for the above model is defined by

$$P(\beta, \Lambda_0() | D) \sim P(D | \beta, \Lambda_0()) P(\beta) P(\Lambda_0())$$

For *BUGS*, we need to specify the form of the likelihood $P(D | \beta, \Lambda_0())$ and prior distributions for β and $\Lambda_0()$. Under non-informative censoring, the likelihood of the data is proportional to

$$\prod_{i=1}^n \left[\prod_{t \geq 0} I_i(t)^{dN_i(t)} \right] \exp(-\int_0^{\infty} I_i(t)dt)$$

This is essentially as if the counting process increments $dN_i(t)$ in the time interval $[t, t+dt]$ are independent Poisson random variables with means $I_i(t)dt$:

$$dN_i(t) \sim \text{Poisson}(l_i(t)dt)$$

We may write

$$l_i(t)dt = Y_i(t)\exp(\beta z_i)d\Lambda_0(t)$$

where $d\Lambda_0(t) = \Lambda_0(t)dt$ is the increment or jump in the integrated baseline hazard function occurring during the time interval $[t, t+dt]$. Since the conjugate prior for the Poisson mean is the gamma distribution, it would be convenient if $\Lambda_0()$ were a process in which the increments $d\Lambda_0(t)$ are distributed according to gamma distributions. We assume the conjugate independent increments prior suggested by Kalbfleisch (1978), namely

$$d\Lambda_0(t) \sim \text{Gamma}(cd\Lambda_0^*(t), c)$$

Here, $d\Lambda_0^*(t)$ can be thought of as a prior guess at the unknown hazard function, with c representing the degree of confidence in this guess. Small values of c correspond to weak prior beliefs. In the example below, we set $d\Lambda_0^*(t) = r dt$ where r is a guess at the failure rate per unit time, and dt is the size of the time interval.

The above formulation is appropriate when genuine prior information exists concerning the underlying hazard function. Alternatively, if we wish to reproduce a Cox analysis but with, say, additional hierarchical structure, we may use the multinomial-Poisson trick described in the *BUGS* manual. This is equivalent to assuming independent increments in the cumulative 'non-informative' priors. This formulation is also shown below.

The fixed effect regression coefficients β are assigned a vague prior

$$\beta \sim \text{Normal}(0.0, 0.000001)$$

BUGS language for the Leuk example:

```

model
{
# Set up data
  for(i in 1:N) {
    for(j in 1:T) {
# risk set = 1 if obs.t >= t
      Y[i,j] <- step(obs.t[i] - t[j] + eps)
# counting process jump = 1 if obs.t in [ t[j], t[j+1] )
#           i.e. if t[j] <= obs.t < t[j+1]
      dN[i, j] <- Y[i, j] * step(t[j + 1] - obs.t[i] - eps) * fail[i]
    }
  }
# Model
  for(j in 1:T) {
    for(i in 1:N) {
      dN[i, j] ~ dpois(ldt[i, j])          # Likelihood
      ldt[i, j] <- Y[i, j] * exp(beta * Z[i]) * dL0[j] # Intensity
    }
    dL0[j] ~ dgamma(mu[j], c)
    mu[j] <- dL0.star[j] * c   # prior mean hazard
  }
# Survivor function = exp(-Integral{l0(u)du})^exp(beta*z)
  S.treat[j] <- pow(exp(-sum(dL0[1 : j])), exp(beta * -0.5));
  S.placebo[j] <- pow(exp(-sum(dL0[1 : j])), exp(beta * 0.5));
}
c <- 0.001
r <- 0.1
for (j in 1 : T) { dL0.star[j] <- r * (t[j + 1] - t[j]) }
beta ~ dnorm(0.0,0.0000001)
}

```

Data \Rightarrow list(N = 42, T = 17, eps = 1.0E-10,

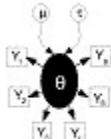
Inits \Rightarrow list(beta = 0.0,

```
dL0 = c(1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,  
      1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0)) $\leftarrow$ 
```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample	
S.placebo[1]	0.9257	0.05092	5.019E-4	0.7988	0.937	0.9904	1001	10000	
S.placebo[2]	0.8519	0.0686	6.338E-4	0.6929	0.8619	0.9582	1001	10000	
S.placebo[3]	0.8147	0.07487	6.828E-4	0.6476	0.8227	0.9365	1001	10000	
S.placebo[4]	0.7413	0.08489	7.565E-4	0.5577	0.7486	0.885	1001	10000	
S.placebo[5]	0.6676	0.09183	9.088E-4	0.4734	0.6731	0.8312	1001	10000	
S.placebo[6]	0.56	0.09717	8.407E-4	0.3644	0.5622	0.7397	1001	10000	
S.placebo[7]	0.5269	0.09772	8.701E-4	0.3333	0.5274	0.7112	1001	10000	
S.placebo[8]	0.4113	0.09431	8.231E-4	0.2359	0.4098	0.6011	1001	10000	
S.placebo[9]	0.378	0.09366	8.078E-4	0.2023	0.3751	0.566	1001	10000	
S.placebo[10]	0.318	0.09024	8.049E-4	0.1572	0.3134	0.5074	1001	10000	
S.placebo[11]	0.2568	0.0855	6.886E-4	0.1112	0.2499	0.4414	1001	10000	
S.placebo[12]	0.2237	0.08225	6.879E-4	0.08616	0.2156	0.399	1001	10000	
S.placebo[13]	0.1942	0.07836	6.721E-4	0.06663	0.1859	0.3652	1001	10000	
S.placebo[14]	0.1642	0.07302	6.817E-4	0.0491	0.1547	0.326	1001	10000	
S.placebo[15]	0.1386	0.06776	6.091E-4	0.03615	0.1289	0.2955	1001	10000	
S.placebo[16]	0.08536	0.05401	4.756E-4	0.01341	0.07397	0.217	1001	10000	
S.placebo[17]	0.04334	0.03805	3.89E-4	0.002476	0.03214	0.1436	1001	10000	
S.treat[1]	0.9826	0.01426	1.676E-4	0.9451	0.9863	0.9982	1001	10000	
S.treat[2]	0.9642	0.02168	2.587E-4	0.9095	0.9688	0.9921	1001	10000	
S.treat[3]	0.9545	0.02514	3.11E-4	0.8913	0.9595	0.9882	1001	10000	
S.treat[4]	0.9342	0.03235	4.251E-4	0.8565	0.9401	0.9795	1001	10000	
S.treat[5]	0.9124	0.03906	5.61E-4	0.8202	0.9186	0.97	1001	10000	
S.treat[6]	0.8769	0.04935	7.001E-4	0.7626	0.8835	0.9532	1001	10000	
S.treat[7]	0.8648	0.05292	7.705E-4	0.7414	0.8722	0.9472	1001	10000	
S.treat[8]	0.8177	0.06533	9.487E-4	0.6724	0.8255	0.9226	1001	10000	
S.treat[9]	0.8022	0.06898	0.001035	0.6496	0.8096	0.9145	1001	10000	
S.treat[10]	0.7714	0.07658	0.001155	0.6015	0.7793	0.9001	1001	10000	
S.treat[11]	0.7346	0.0851	0.001242	0.5488	0.7422	0.8778	1001	10000	
S.treat[12]	0.7117	0.08988	0.001286	0.5204	0.7192	0.8648	1001	10000	
S.treat[13]	0.6888	0.09417	0.001366	0.4879	0.6965	0.8517	1001	10000	
S.treat[14]	0.6626	0.09837	0.001414	0.4569	0.6685	0.837	1001	10000	
S.treat[15]	0.6368	0.1023	0.001492	0.4242	0.6424	0.8201	1001	10000	
S.treat[16]	0.5667	0.1128	0.001537	0.34	0.5695	0.7772	1001	10000	
S.treat[17]	0.4765	0.1208	0.001524	0.2444	0.4763	0.7093	1001	10000	
beta	1.546	0.4186	0.005352	0.7653	1.533	2.395	1001	10000	



BUGS LeukFr: Cox regression with random effects

Freireich et al (1963)'s data presented in the Leuk example actually arise via a *paired* design. Patients were matched according to their remission status (partial or complete). One patient from each pair received the drug 6-MP whilst the other received the placebo. We may introduce an additional vector (called `pair`) in the BUGS data file to indicate each of the 21 pairs of patients.

We model the potential 'clustering' of failure times within pairs of patients by introducing a group-specific random effect or frailty term into the proportional hazards model. Using the counting process notation introduced in the Leuk example, this gives

$$\begin{aligned} l_i(t) dt &= Y_i(t) \exp(\beta' z_i + b_{\text{pair}_i}) d\Lambda_0(t) & i = 1, \dots, 42; \quad \text{pair}_i = 1, \dots, 21 \\ b_{\text{pair}_i} &\sim \text{Normal}(0, \tau) \end{aligned}$$

Two alternative "noninformative" priors are considered for the variance of the frailty parameters: prior 1 is a uniform prior on the standard deviation ($\sigma = 1/\sqrt{\tau}$), and prior 2 is a $\text{gamma}(0.001, 0.001)$ prior on the precision τ . Note that the above 'additive' formulation of the frailty model is equivalent to assuming multiplicative frailties with a log-Normal population distribution. Clayton (1991) discusses the Cox proportional hazards model with multiplicative frailties, but assumes a Gamma population distribution.

The modified BUGS code needed to include a frailty term in the Leuk example is shown below

```

model
{
# Set up data
for(i in 1 : N) {
  for(j in 1 : T) {
    # risk set = 1 if obs.t >= t
    Y[i, j] <- step(obs.t[i] - t[j]) + eps
    # counting process jump = 1 if obs.t in [ t[j], t[j]+1 ] )
    #           i.e. if t[j] <= obs.t < t[j+1]
    dN[i, j] <- Y[i, j] *step(t[j+1] - obs.t[i] - eps)*fail[i]
  }
}
# Model
for(j in 1 : T) {
  for(i in 1 : N) {
    dN[i, j] ~ dpois(ldt[i, j])
    ldt[i, j] <- Y[i, j] * exp(beta * Z[i]+b[pair[i]]) * dL0[j]
  }
  dL0[j] ~ dgamma(mu[j], c)
  mu[j] <- dL0.star[j] * c  # prior mean hazard
  # Survivor function = exp(-Integral{l0(u)du})^exp(beta * z)
  S.treat[j] <- pow(exp(-sum(dL0[1 : j])), exp(beta * -0.5))
  S.placebo[j] <- pow(exp(-sum(dL0[1 : j])), exp(beta * 0.5))
}
for(k in 1 : Npairs) {
  b[k] ~ dnorm(0.0, tau);
}

```

```

}

# Choice of priors on random effects (frailty) variance
# Prior 1: uniform on SD
sigma ~ dunif(0,100)
tau <- 1/(sigma*sigma)

#Prior 2:
#tau ~ dgamma(1.0E-3, 1.0E-3);
#sigma <- 1/sqrt(tau); # s.d. of random effects

c <- 0.001 r <- 0.1
for (j in 1:T) {
  dL0.star[j] <- r * (t[j+1]-t[j])
}
beta ~ dnorm(0.0,0.000001)
}

```

Data \Rightarrow list(N = 42, T = 17, eps = 0.00001, Npairs = 21,
 $t = c(1,2,3,4,5,6,7,8,10,11,12,13,15,16,17,22,23,35),$
 $obs.t = c(1,1,2,2,3,4,4,5,5,8,8,8,11,11,12,12,15,17,22,23,$
 $6,6,6,7,9,10,10,11,13,16,17,19,20,22,23,25,32,32,34,35),$
 $pair = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,$
 $19,18,8,1,20,6,2,10,3,14,4,11,7,9,12,16,17,5,13,15,21),$
 $fail = c(1,$
 $1,1,1,0,1,0,1,0,0,1,1,0,0,0,1,1,0,0,0,0,0,0),$
 $Z = c(-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,$
 $-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,$
 $0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,$
 $0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5)$
 $)\Leftarrow$

Inits1 \Rightarrow list(beta = 0.0,
 $dL0 = c(1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,$
 $1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0),$
 $sigma=1$
 $)\Leftarrow$

Inits2 \Rightarrow list(beta = 0.0,
 $dL0 = c(1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,$
 $1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0),$
 $tau=1$
 $)\Leftarrow$

Results

A 1000 update burn in followed by a further 10000 updates gave the following parameter estimates

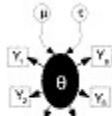
Prior 1: uniform on SD: preferred prior.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta	-1.674	0.4604	0.0105	-2.643	-1.652	-0.8429	1001	10000
sigma	0.3976	0.2965	0.01803	0.008496	0.338	1.098	1001	10000

Prior 2: Gamma(0.001,0.001) on precision

This prior leads to a substantially smaller estimate of sigma, with a slightly increased precision when estimating beta.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta	-1.585	0.4286	0.006553	-2.455	-1.576	-0.7727	1001	10000
sigma	0.2149	0.2027	0.01268	0.02829	0.1395	0.7565	1001	10000



BUGS Dugongs: nonlinear growth curve

Carlin and Gelfand (1991) present a nonconjugate Bayesian analysis of the following data set from Ratkowsky (1983):

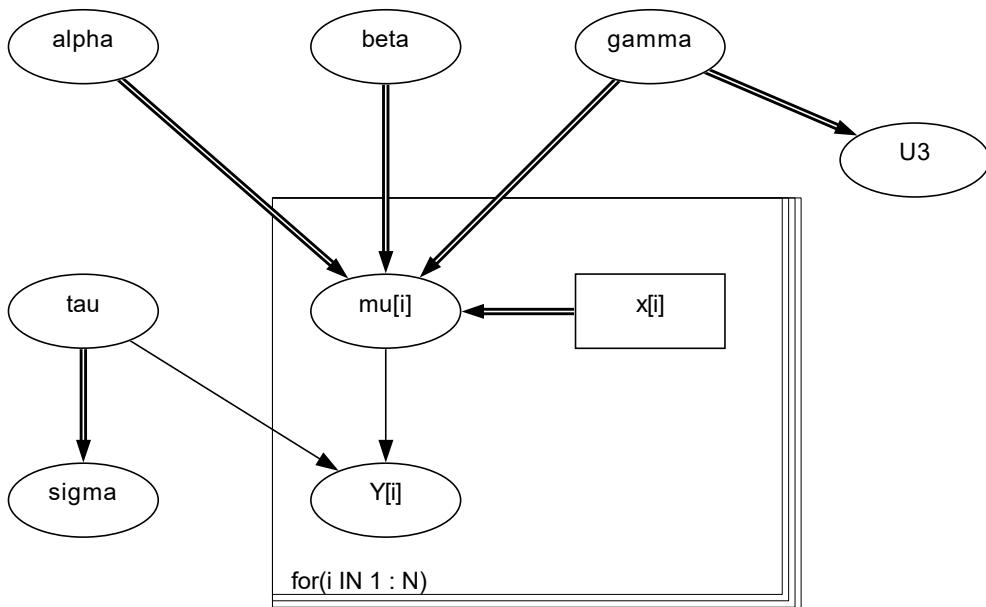
Dugong	1	2	3	4	5	26	27
Age (X)	1.0	1.5	1.5	1.5	2.5	29.0	31.5
Length (Y)	1.80	1.85	1.87	1.77	2.02	2.27	2.57

The data are length and age measurements for 27 captured dugongs (sea cows). Carlin and Gelfand (1991) model this data using a nonlinear growth curve with no inflection point and an asymptote as X_i tends to infinity:

$$Y_i \sim \text{Normal}(\mu_i, \tau), \quad i = 1, \dots, 27$$

$$\mu_i = \alpha - \beta \gamma^{X_i} \quad \alpha, \beta > 1; 0 < \gamma < 1$$

Standard noninformative priors are adopted for α , β and τ , and a uniform prior on (0,1) is assumed for γ . However, this specification leads to a non conjugate full conditional distribution for γ which is also non log-concave. The graph and corresponding BUGS code is given below



```

model
{
  for( i in 1 : N ){
    Y[i] ~ dnorm(mu[i], tau)
    mu[i] <- alpha - beta * pow(gamma,x[i])

  }
  alpha ~ dnorm(0.0, 1.0E-6)
  beta ~ dnorm(0.0, 1.0E-6)
  gamma ~ dunif(0.5, 1.0)
  tau ~ dgamma(0.001, 0.001)
  sigma <- 1 / sqrt(tau)
  U3 <- logit(gamma)
}

```

Data \Rightarrow

```

list(x = c( 1.0, 1.5, 1.5, 1.5, 2.5, 4.0, 5.0, 5.0, 7.0,
          8.0, 8.5, 9.0, 9.5, 9.5, 10.0, 12.0, 12.0, 13.0,
          13.0, 14.5, 15.5, 15.5, 16.5, 17.0, 22.5, 29.0, 31.5),
      Y = c(1.80, 1.85, 1.87, 1.77, 2.02, 2.27, 2.15, 2.26, 2.47,
            2.19, 2.26, 2.40, 2.39, 2.41, 2.50, 2.32, 2.32, 2.43,
            2.47, 2.56, 2.65, 2.47, 2.64, 2.56, 2.70, 2.72, 2.57), N = 27)  $\Leftarrow$ 

```

Inits \Rightarrow

```
list(alpha = 1, beta = 1, tau = 1, gamma = 0.9)  $\Leftarrow$ 
```

Results

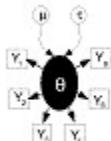
A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

a) Without over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
U3	1.885	0.2756	0.01228	1.313	1.89	2.432
alpha	2.658	0.07479	0.003493	2.531	2.651	2.83
beta	0.9736	0.07726	0.001784	0.8249	0.9719	1.131
gamma	0.865	0.03266	0.001417	0.788	0.8687	0.9192
sigma	0.09927	0.01549	2.192E-4	0.07478	0.09741	0.1352

b) With over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
U3	1.861	0.2738	0.005245	1.303	1.864	2.385
alpha	2.653	0.07324	0.001585	2.529	2.647	2.816
beta	0.9749	0.07716	9.589E-4	0.8278	0.9736	1.133
gamma	0.8623	0.03318	6.083E-4	0.7864	0.8658	0.9157
sigma	0.09905	0.01487	1.862E-4	0.07492	0.09734	0.1332



BUGS Orange Trees: Non-linear growth curve

This dataset was originally presented by Draper and Smith (1981) and reanalysed by Lindstrom and Bates (1990). The data Y_{ij} consist of trunk circumference measurements recorded at time x_j , $j=1,\dots,7$ for each of $i = 1,\dots, 5$ orange trees. We consider a logistic growth curve as follows:

$$\begin{aligned} Y_{ij} &\sim \text{Normal}(\eta_{ij}, \tau_c) \\ \eta_{ij} &= \frac{\theta_{i1}}{1 + \phi_{i2} \exp(\phi_{i3} x_j)} \\ \theta_{i1} &= \log(\phi_{i1}) \\ \theta_{i2} &= \log(\phi_{i2} + 1) \\ \theta_{i3} &= \log(-\phi_{i3}) \end{aligned}$$

The BUGS code is as follows

```

model {
  for (i in 1:K) {
    for (j in 1:n) {
      Y[i, j] ~ dnorm(eta[i, j], tauC)
      eta[i, j] <- phi[i, 1] / (1 + phi[i, 2] * exp(phi[i, 3] * x[j])))
    }
    phi[i, 1] <- exp(theta[i, 1])
    phi[i, 2] <- exp(theta[i, 2]) - 1
    phi[i, 3] <- -exp(theta[i, 3])
    for (k in 1:3) {
      theta[i, k] ~ dnorm(mu[k], tau[k])
    }
  }
  tauC ~ dgamma(1.0E-3, 1.0E-3)
  sigmaC <- 1 / sqrt(tauC)
  varC <- 1 / tauC
  for (k in 1:3) {
    mu[k] ~ dnorm(0, 1.0E-4)
    tau[k] ~ dgamma(1.0E-3, 1.0E-3)
    sigma[k] <- 1 / sqrt(tau[k])
  }
}

```

Data

```
⇒list(n = 7, K = 5, x = c(118.00, 484.00, 664.00, 1004.00, 1231.00, 1372.00, 1582.00),
      Y = structure(
        .Data = c(30.00, 58.00, 87.00, 115.00, 120.00, 142.00, 145.00,
                 33.00, 69.00, 111.00, 156.00, 172.00, 203.00, 203.00,
                 30.00, 51.00, 75.00, 108.00, 115.00, 139.00, 140.00,
```

```

32.00, 62.00, 112.00, 167.00, 179.00, 209.00, 214.00,
30.00, 49.00, 81.00, 125.00, 142.00, 174.00, 177.00),
.Dim = c(5, 7))) $\Leftarrow$ 

```

Inits

```

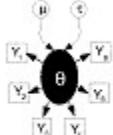
 $\Rightarrow$ list(theta = structure(
  .Data = c(5, 2, -6,
           5, 2, -6,
           5, 2, -6,
           5, 2, -6,
           5, 2, -6),
  .Dim = c(5, 3)),
  mu = c(5, 2, -6), tau = c(20, 20, 20), tauC = 20) $\Leftarrow$ 

```

Results

The Metropolis algorithm is used to sample the theta parameters in this model. The Gaussian proposal distribution used for this algorithm adapts for the first 4000 iterations and these samples are discarded from the summary statistics. A further 1000 update burn-in followed by 10000 updates gave the following parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[1]	5.263	0.1317	0.005271	5.004	5.262	5.505	5001	10000
mu[2]	2.195	0.1267	0.007693	1.959	2.189	2.443	5001	10000
mu[3]	-5.885	0.1212	0.009366	-6.189	-5.873	-5.695	5001	10000
sigma[1]	0.2257	0.1352	0.004796	0.05804	0.1982	0.5529	5001	10000
sigma[2]	0.1431	0.1267	0.005708	0.02453	0.1086	0.4596	5001	10000
sigma[3]	0.1137	0.1008	0.006555	0.02381	0.08349	0.3696	5001	10000
sigmaC	7.993	1.24	0.04616	6.006	7.85	10.82	5001	10000



BUGS Orange Trees: Non-linear growth curve

We repeat the Otrees example, replacing the 3 independent univariate Normal priors for each ϕ_{ik} , $k=1,2,3$ by a multivariate Normal prior $\phi_i \sim \text{MNV}(\mu, T)$

```

model {
  for (i in 1:K) {
    for (j in 1:n) {
      Y[i, j] ~ dnorm(eta[i, j], tauC)
      eta[i, j] <- phi[i, 1] / (1 + phi[i, 2] * exp(phi[i, 3] * x[j]))
    }
    phi[i, 1] <- exp(theta[i, 1])
    phi[i, 2] <- exp(theta[i, 2]) - 1
    phi[i, 3] <- -exp(theta[i, 3])
    theta[i, 1:3] ~ dmnorm(mu[1:3], tau[1:3, 1:3])
  }
  mu[1:3] ~ dmnorm(mean[1:3], prec[1:3, 1:3])
  tau[1:3, 1:3] ~ dwish(R[1:3, 1:3], 3)
  sigma2[1:3, 1:3] <- inverse(tau[1:3, 1:3])
  for (i in 1 : 3) {sigma[i] <- sqrt(sigma2[i, i])}
  tauC ~ dgamma(1.0E-3, 1.0E-3)
  sigmaC <- 1 / sqrt(tauC)
}

```

Data

```

⇒list(n = 7, K = 5, x = c(118.00, 484.00, 664.00, 1004.00, 1231.00, 1372.00, 1582.00),
Y = structure(
  .Data = c(30.00, 58.00, 87.00, 115.00, 120.00, 142.00, 145.00,
           33.00, 69.00, 111.00, 156.00, 172.00, 203.00, 203.00,
           30.00, 51.00, 75.00, 108.00, 115.00, 139.00, 140.00,
           32.00, 62.00, 112.00, 167.00, 179.00, 209.00, 214.00,
           30.00, 49.00, 81.00, 125.00, 142.00, 174.00, 177.00),
  .Dim = c(5, 7)),
  mean = c(0, 0, 0),
  R = structure(.Data = c(0.1, 0, 0,
                         0, 0.1, 0,
                         0, 0, 0.1), .Dim = c(3, 3)),
  prec = structure(.Data = c(1.0E-6, 0, 0,
                            0, 1.0E-6, 0,
                            0, 0, 1.0E-6), .Dim = c(3, 3))) ↵

```

Inits

```

⇒list(theta = structure(
  .Data = c(5, 2, -6,
           5, 2, -6,
           5, 2, -6,

```

```

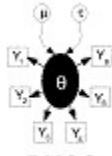
5, 2, -6,
5, 2, -6),
.Dim = c(5, 3),
mu = c(5, 2, -6),
tau = structure(.Data = c(0.1, 0, 0,
                           0, 0.1, 0,
                           0, 0, 0.1), .Dim = c(3, 3)),
tauC = 20)◻

```

Results

Time for 10000 updates 32s on 200MHz Pentium Pro. A 4000 iteration Metropolis adaptive phase plus 1000 update burn in followed by a further 10000 updates gave the parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[1]	5.257	0.1356	0.003816	4.986	5.257	5.519	5001	10000
mu[2]	2.2	0.1685	0.008079	1.872	2.198	2.55	5001	10000
mu[3]	-5.87	0.1416	0.008107	-6.144	-5.872	-5.591	5001	10000
sigma[1]	0.259	0.1209	0.002427	0.13	0.231	0.5474	5001	10000
sigma[2]	0.268	0.1284	0.003944	0.12	0.2382	0.6097	5001	10000
sigma[3]	0.2191	0.1026	0.003375	0.1061	0.1945	0.4783	5001	10000
sigmaC	7.8	1.166	0.03427	5.884	7.685	10.43	5001	10000



BUGS

Biopsies: discrete variable latent class model

Spiegelhalter and Stovin (1983) presented data on repeated biopsies of transplanted hearts, in which a total of 414 biopsies had been taken at 157 sessions. Each biopsy was graded on evidence of rejection using a 4 category scale of none (O), minimal (M), mild (+) and moderate-severe (++) . Part of the data is shown below.

Combination	Multinomial response	Session frequency
O O	(2, 0, 0, 0)	12
M M O	(1, 2, 0, 0)	10
+ + O	(1, 0, 2, 0)	17
++ ++ ++	(0, 0, 0, 3)	5

The sampling procedure may not detect the area of maximum rejection, which is considered the true underlying state at the time of the session and denoted t_i — the underlying probability distribution of the four true states is denoted by the vector p . It is then assumed that each of the observed biopsies are conditionally independent given this truestate with the restriction that there are no 'false positives': i.e. one cannot observe a biopsy worse than the true state. We then have the sampling model

$$b_i \sim \text{Multinomial}(e_{t_i}, n_i)$$

$$t_i \sim \text{Categorical}(p)$$

where b_i denotes the multinomial response at session i where n_i biopsies have been taken, and e_{jk} is the probability that a true state $t_i = j$ generates a biopsy in state k . The no-false-positive restriction means that $e_{12} = e_{13} = e_{14} = e_{23} = e_{24} = e_{34} = 0$. Spiegelhalter and Stovin (1983) estimated the parameters e_j and p using the EM algorithm, with some smoothing to avoid zero estimates.

The appropriate graph is shown below, where the role of the true state t_i is simply to pick the appropriate row from the 4×4 error matrix e . Here the probability vectors e_j ($j = 1, \dots, 4$) and p are assumed to have uniform priors on the unit simplex, which correspond to Dirichlet priors with all parameters being 1.

The BUGS code for this model is given below. No initial value file is provided, since the forward sampling procedure will find a configuration of starting values that is compatible with the expressed constraints. We also note the apparent "cycle" in the graph created by the expression `nbiops[i] <- sum(biopsies[i,])`. This will lead Such "cycles" are permitted provided that they are only data transformation statements, since this does not affect the essential probability model.

```

model
{
  for (i in 1:ns){
    nbiops[i] <- sum(biopsies[i, ])
    true[i] ~ dcat(p[])
    biopsies[i, 1:4] ~ dmulti(error[true[i], ], nbiops[i])
  }
  error[2,1:2] ~ ddirch(prior[1:2])
  error[3,1:3] ~ ddirch(prior[1:3])
  error[4,1:4] ~ ddirch(prior[1:4])
  p[1:4] ~ ddirch(prior[]); # prior for p
}

```

Data

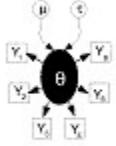
Inits

```
⇒list(p = c(0.25, 0.25, 0.25, 0.25),  
      error = structure(.Data = c(NA, NA, NA, NA,  
                                 0.5, 0.5, NA, NA,  
                                 0.33333333, 0.33333333, 0.33333333, NA,  
                                 0.25, 0.25, 0.25, 0.25), Dim = c(4, 4)))
```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
error[2,1]	0.5889	0.06761	0.001531	0.4574	0.5895	0.7167	1001	10000
error[2,2]	0.4111	0.06761	0.001531	0.2834	0.4106	0.5427	1001	10000
error[3,1]	0.3412	0.04587	7.03E-4	0.2577	0.3392	0.4369	1001	10000
error[3,2]	0.03692	0.01767	2.685E-4	0.009082	0.03458	0.07752	1001	10000
error[3,3]	0.6219	0.04761	7.629E-4	0.5245	0.6234	0.7087	1001	10000
error[4,1]	0.09886	0.04177	4.823E-4	0.0332	0.09287	0.1953	1001	10000
error[4,2]	0.02173	0.02229	3.216E-4	5.25E-4	0.01456	0.08197	1001	10000
error[4,3]	0.2064	0.06015	8.509E-4	0.1053	0.2013	0.3389	1001	10000
error[4,4]	0.673	0.07173	0.001009	0.5225	0.6775	0.8029	1001	10000
p[1]	0.1521	0.05134	0.001353	0.04446	0.1539	0.2482	1001	10000
p[2]	0.3124	0.05657	0.001383	0.2138	0.3083	0.4352	1001	10000
p[3]	0.3879	0.04391	5.475E-4	0.3037	0.3872	0.4768	1001	10000
p[4]	0.1475	0.03002	3.383E-4	0.09309	0.1459	0.2111	1001	10000



BUGS Eyes: Normal Mixture Model

Bowmaker et al (1985) analyse data on the peak sensitivity wavelengths for individual microspectrophotometric records on a small set of monkey's eyes. Data for one monkey (S14 in the paper) are given below (500 has been subtracted from each of the 48 measurements).

29.0	30.0	32.0	33.1	33.4	33.6	33.7	34.1	34.8	35.3
35.4	35.9	36.1	36.3	36.4	36.6	37.0	37.4	37.5	38.3
38.5	38.6	39.4	39.6	40.4	40.8	42.0	42.8	43.0	43.5
43.8	43.9	45.3	46.2	48.8	48.7	48.9	49.0	49.4	49.9
50.6	51.2	51.4	51.5	51.6	52.8	52.9	53.2		

Part of the analysis involves fitting a mixture of two normal distributions with common variance to this distribution, so that each observation y_i is assumed drawn from one of two groups. $T_i = 1, 2$ be the true group of the i th observation, where group j has a normal distribution with mean λ_j and precision τ . We assume an unknown fraction P of observations are in group 2, $1 - P$ in group 1. The model is thus

$$y_i \sim \text{Normal}(\lambda_{T_i}, \tau)$$

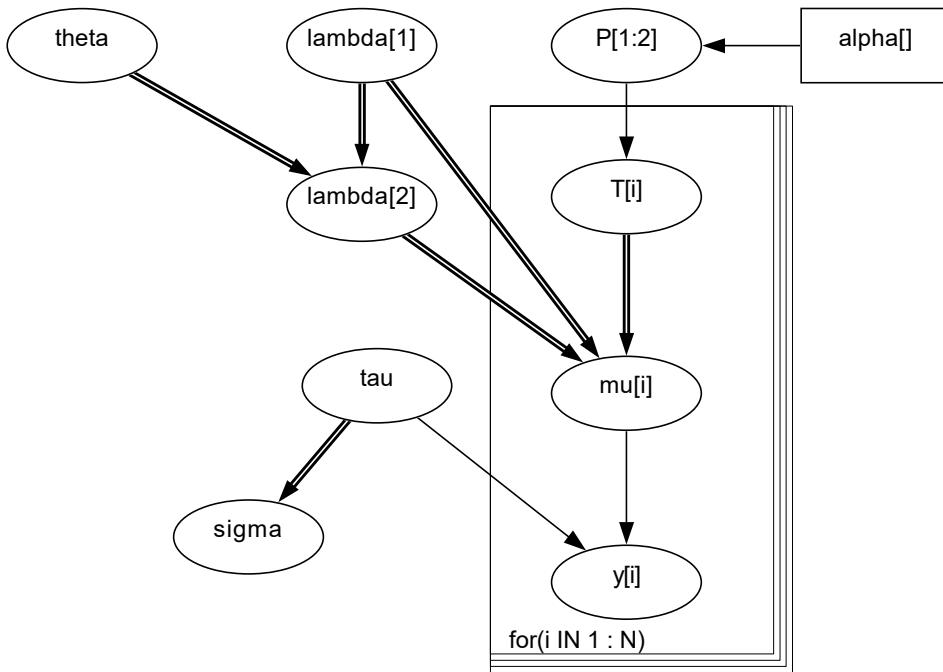
$$T_i \sim \text{Categorical}(P).$$

We note that this formulation easily generalises to additional components to the mixture, although for identifiability an order constraint must be put onto the group means.

Robert (1994) points out that when using this model, there is a danger that at some iteration, *all* the data will go into one component of the mixture, and this state will be difficult to escape from --- this matches our experience. Robert suggests a re-parameterisation, a simplified version of which is to assume

$$\lambda_2 = \lambda_1 + \theta, \quad \theta > 0.$$

$\lambda_1, \theta, \tau, P$, are given independent "noninformative" priors, including a uniform prior for P on $(0,1)$. The appropriate graph and the BUGS code are given below.



```

model
{
  for( i in 1 : N ){
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- lambda[T[i]]
    T[i] ~ dcat(P[])
  }
  P[1:2] ~ ddirch(alpha[])
  theta ~ dnorm(0.0, 1.0E-6)I(0.0, )
  lambda[2] <- lambda[1] + theta
  lambda[1] ~ dnorm(0.0, 1.0E-6)
  tau ~ dgamma(0.001, 0.001) sigma <- 1 / sqrt(tau)
}

```

Data \Leftrightarrow

```

list(y = c(529.0, 530.0, 532.0, 533.1, 533.4, 533.6, 533.7, 534.1, 534.8, 535.3,
       535.4, 535.9, 536.1, 536.3, 536.4, 536.6, 537.0, 537.4, 537.5, 538.3,
       538.5, 538.6, 539.4, 539.6, 540.4, 540.8, 542.0, 542.8, 543.0, 543.5,
       543.8, 543.9, 545.3, 546.2, 548.8, 548.7, 548.9, 549.0, 549.4, 549.9,
       550.6, 551.2, 551.4, 551.5, 551.6, 552.8, 552.9, 553.2), N = 48, alpha = c(1, 1),
  T = c(1, NA, NA, NA, NA, NA, NA, NA, NA,
        NA, NA, NA, NA, NA, NA, NA, NA, NA,
        NA, NA, NA, NA, NA, NA, NA, NA, NA,
        NA, NA, NA, NA, NA, NA, NA, NA, NA,
        NA, NA, NA, NA, NA, NA, 2)) $\Leftrightarrow$ 

```

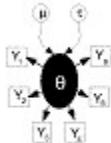
Inits \Leftrightarrow

```
list(lambda = c(535, NA), theta = 5, tau = 0.1) ↵
```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
P[1]	0.5982	0.08526	0.001663	0.426	0.6016	0.7554	1001	10000
P[2]	0.4018	0.08526	0.001663	0.2447	0.3984	0.5741	1001	10000
lambda[1]	536.7	0.9265	0.02124	535.0	536.7	538.6	1001	10000
lambda[2]	548.9	1.256	0.0314	546.3	548.9	551.2	1001	10000
sigma	3.768	0.6137	0.01967	2.935	3.668	5.25	1001	10000



BUGS Hearts: a mixture model for count data

The table below presents data given by Berry (1987) on the effect of a drug used to treat patients with frequent premature ventricular contractions (PVCs) of the heart.

number (i)	PVC's per minute		
	Pre-drug (x_i)	Post-drug (y_i)	Decrease
1	6	5	1
2	9	2	7
3	17	0	17
.
11	9	13	-4
12	51	0	51

Farewell and Sprott (1988) model these data as a mixture distribution of Poisson counts in which some patients are "cured" by the drug, whilst others experience varying levels of response but remain abnormal. A zero count for the post-drug PVC may indicate a "cure", or may represent a sampling zero from a patient with a mildly abnormal PVC count. The following model thus is assumed:

$$\begin{aligned}
 x_i &\sim \text{Poisson}(\lambda_i) && \text{for all patients} \\
 y_i &\sim \text{Poisson}(\beta\lambda_i) && \text{for all } \textit{uncured} \text{ patients} \\
 P(\text{cure}) &= \theta
 \end{aligned}$$

To eliminate nuisance parameters λ_i , Farewell and Sprott use the conditional distribution of y_i given $t_i = x_i + y_i$. This is equivalent to a binomial likelihood for y_i with denominator t_i and probability $p = b / 1+b$ (see Cox and Hinkley, 1974 pp. 136-137 for further details of the conditional distribution for Poisson variables). Hence the final mixture model may be expressed as follows:

$$\begin{aligned}
 P(y_i = 0 | t_i) &= \theta + (1 - \theta)(1 - p)t_i \\
 P(y_i | t_i) &= (1 - \theta)(t_i! / (y_i!(t_i - y_i)!)) (p^{y_i}(1 - p)^{(t_i - y_i)}} \quad y_i = 1, 2, \dots, t_i
 \end{aligned}$$

The BUGS code for this model is given below:

```

model
{
  for (i in 1 : N) {
    y[i] ~ dbin(P[state1[i]], t[i])
    state[i] ~ dbern(theta)
    state1[i] <- state[i] + 1
    t[i] <- x[i] + y[i]
    prop[i] <- P[state1[i]]
  }
  P[1] <- p
  P[2] <- 0
  logit(p) <- alpha
  alpha ~ dnorm(0, 1.0E-4)
  beta <- exp(alpha)
  logit(theta) <- delta
  delta ~ dnorm(0, 1.0E-4)
}

```

Data \Rightarrow

```

list(x = c(6, 9, 17, 22, 7, 5, 5, 14, 9, 7, 9, 51),
     y = c(5, 2, 0, 0, 2, 1, 0, 0, 0, 0, 13, 0), N = 12)

```

Inits \Rightarrow

```

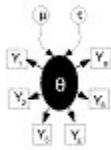
list(delta = 0, alpha = 0)

```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	-0.4787	0.274	0.002674	-1.019	-0.4789	0.04994	1001	10000
beta	0.6431	0.1783	0.001731	0.3611	0.6195	1.051	1001	10000
delta	0.3177	0.6288	0.006831	-0.9098	0.3108	1.584	1001	10000
theta	0.5721	0.1407	0.001528	0.287	0.5771	0.8298	1001	10000



BUGS Air: Berkson measurement error

Whittemore and Keller (1988) use an approximate maximum likelihood approach to analyse the data shown below on reported respiratory illness versus exposure to nitrogen dioxide (NO_2) in 103 children. Stephens and Dellaportas (1992) later use Bayesian methods to analyse the same data.

Respiratory illness (y)	Bedroom NO_2 level in ppb (z)			
	<20	20–40	40+	Total
Yes	21	20	15	56
No	27	14	6	47
Total	48	34	21	103

A discrete covariate z_j ($j = 1, 2, 3$) representing NO_2 concentration in the child's bedroom classified into 3 categories is used as a surrogate for true exposure. The nature of the measurement error relationship associated with this covariate is known precisely via a calibration study, and is given by

$$x_j = \alpha + \beta z_j + \varepsilon_j$$

where $\alpha = 4.48$, $\beta = 0.76$ and ε_j is a random element having normal distribution with zero mean and variance $\sigma^2 (= 1/\tau) = 81.14$. Note that this is a Berkson (1950) model of measurement error, in which the true values of the covariate are expressed as a function of the observed values. Hence the measurement error is independent of the latter, but is correlated with the true underlying covariate values. In the present example, the observed covariate z_j takes values 10, 30 or 50 for $j = 1, 2$, or 3 respectively (i.e. the mid-point of each category), whilst x_j is interpreted as the "true average value" of NO_2 in group j . The response variable is binary, reflecting presence/absence of respiratory illness, and a logistic regression model is assumed. That is

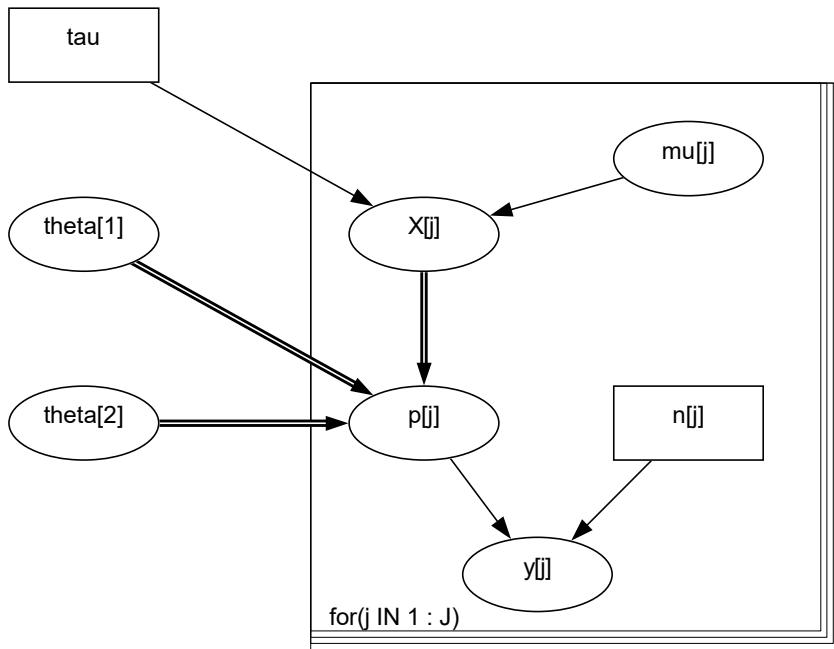
$$y_j \sim \text{Binomial}(p_j, n_j)$$

$$\text{logit}(p_j) = \theta_1 + \theta_2 x_j$$

where p_j is the probability of respiratory illness for children in the j th exposure group.

The regression coefficients θ_1 and θ_2 are given vague independent normal priors.

The graphical model is shown below:



```

model
{
  for(j in 1 : J) {
    y[j] ~ dbin(p[j], n[j])
    logit(p[j]) <- theta[1] + theta[2] * X[j]
    X[j] ~ dnorm(mu[j], tau)
    mu[j] <- alpha + beta * Z[j]
  }
  theta[1] ~ dnorm(0.0, 0.001)
  theta[2] ~ dnorm(0.0, 0.001)
}

```

Data

```
list(J = 3, y = c(21, 20, 15), n = c(48, 34, 21), Z = c(10, 30, 50), tau = 0.01234, alpha = 4.48, beta = 0.76)
```

Inits

```
list(theta = c(0.0, 0.0), X = c(0.0, 0.0, 0.0))
```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

a) Without over-relaxation.

node	mean	sd	MC error	2.5%	median	97.5%
X[1]	12.57	7.979	0.204	-4.034	12.83	27.17
X[2]	27.17	7.494	0.1085	12.66	27.08	42.2
X[3]	41.33	8.4	0.1784	25.43	41.28	57.97
theta[1]	-0.8276	0.7515	0.03839	-2.812	-0.6843	0.2217
theta[2]	0.04355	0.02906	0.001501	0.003022	0.03805	0.1201

b) With over-relaxation.

node	mean	sd	MC error	2.5%	median	97.5%
X[1]	12.87	8.185	0.1593	-3.643	13.08	27.74
X[2]	27.42	7.432	0.05592	12.81	27.44	42.42
X[3]	41.43	8.472	0.1318	25.31	41.26	58.49
theta[1]	-0.893	0.898	0.03741	-3.445	-0.6819	0.2371
theta[2]	0.04524	0.03292	0.001379	0.00294	0.03772	0.1404

Re-parameterised model with centred covariates:

```

model
{
  for( j in 1 : J ) {
    y[j] ~ dbin(p[j],n[j])
    logit(p[j]) <- theta0+ theta[2] * (X[j] - mean(mu[]))
    X[j] ~ dnorm(mu[j],tau)
    mu[j] <- alpha + beta * Z[j]
  }
  theta0 ~ dnorm(0.0,0.001)
  theta[2] ~ dnorm(0.0,0.001)
  theta[1] <- theta0 - theta[2] * mean(mu[])
}

```

Data

```
list(J = 3, y = c(21, 20, 15), n = c(48, 34, 21), Z = c(10, 30, 50), tau = 0.01234,
alpha = 4.48, beta = 0.76)
```

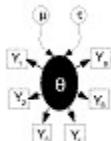
Inits

```
list(theta = c(NA, 0.0), theta0 = 0.0, X = c(0.0, 0.0, 0.0))
```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates, with over-relaxation.

node	mean	sd	MC error	2.5%	median	97.5%
X[1]	13.49	8.595	0.144	-3.65	13.65	29.75
X[2]	27.37	7.4	0.06966	13.04	27.31	42.18
X[3]	40.8	8.612	0.1284	24.35	40.69	57.81
theta[1]	-1.027	1.842	0.06678	-5.001	-0.7077	0.3557
theta[2]	0.05012	0.0671	0.002496	-0.003214	0.03884	0.1966



Cervix: case - control study with errors in covariates

Carroll, Gail and Lubin (1993) consider the problem of estimating the odds ratio of a disease d in a case-control study where the binary exposure variable is measured with error. Their example concerns exposure to herpes simplex virus (HSV) in women with invasive cervical cancer ($d=1$) and in controls ($d=0$). Exposure to HSV is measured by a relatively inaccurate western blot procedure w for 1929 of the 2044 women, whilst for 115 women, it is also measured by a refined or "gold standard" method x . The data are given in the table below. They show a substantial amount of misclassification, as indicated by low sensitivity and specificity of w in the "complete" data, and Carroll, Gail and Lubin also found that the degree of misclassification was significantly higher for the controls than for the cases ($p=0.049$ by Fisher's exact test).

d	x	w	Count
<i>Complete data</i>			
1	0	0	13
1	0	1	3
1	1	0	5
1	1	1	18
0	0	0	33
0	0	1	11
0	1	0	16
0	1	1	16
<i>Incomplete data</i>			
1		0	318
1		1	375
1		0	701
1		1	535

They fitted a prospective logistic model to the case-control data as follows

$$\begin{aligned} d_i &\sim \text{Bernoulli}(p_i) & i = 1, \dots, 2044 \\ \text{logit}(p_i) &= \beta_0 C + \beta x_i & i = 1, \dots, 2044 \end{aligned}$$

where β is the log odds ratio of disease. Since the relationship between d and x is only directly observable in the 115 women with "complete" data, and because there is evidence of differential measurement error, the following parameters are required in order to estimate the logistic model

$$\begin{aligned} \phi_{1,1} &= P(w=1 | x=0, d=0) \\ \phi_{1,2} &= P(w=1 | x=0, d=1) \\ \phi_{2,1} &= P(w=1 | x=1, d=0) \\ \phi_{2,2} &= P(w=1 | x=1, d=1) \\ q &= P(x=1) \end{aligned}$$

The differential probability of being exposed to HSV ($x=1$) for cases and controls is calculated as follows

$$\begin{aligned}
\gamma_1 &= P(x=1 | d=1) \\
&= \frac{P(d=1 | x=1) P(x=1)}{P(d=1)} \\
&= \frac{1}{1 + (1 + \exp \beta_{0C} + \beta) / (1 + \exp \beta_{0C})} \quad \frac{1 - q}{q} \\
\gamma_2 &= P(x=1 | d=0) \\
&= \frac{P(d=0 | x=1) P(x=1)}{P(d=0)} \\
&= \frac{1}{1 + (1 + \exp -\beta_{0C} - \beta) / (1 + \exp -\beta_{0C})} \quad \frac{1 - q}{q}
\end{aligned}$$

The BUGS code is given below. The role of the variables x_1 and d_1 is to pick the appropriate value of ϕ (the incidence of w) for any given true exposure status x and disease status d . Since x and d take the values 0 or 1, and the subscripts for ϕ take values 1 or 2, we must first add 1 to each $x[i]$ and $d[i]$ in the BUGS code before using them as index values for ϕ . BUGS does not allow subscripts to be functions of variable quantities -- hence the need to create x_1 and d_1 for use as subscripts. In addition, note that γ_1 and γ_2 were not simulated directly in BUGS, but were calculated as functions of other parameters. This is because the dependence of γ_1 and γ_2 on d would have led to a cycle in the graphical model which would no longer define a probability distribution.

```

model
{
  for (i in 1 : N) {
    x[i] ~ dbern(q)      # incidence of HSV
    logit(p[i]) <- beta0C + beta * x[i] # logistic model
    d[i] ~ dbern(p[i])    # incidence of cancer
    x1[i] <- x[i] + 1
    d1[i] <- d[i] + 1
    w[i] ~ dbern(phi[x1[i], d1[i]]) # incidence of w
  }
  q ~ dunif(0.0, 1.0)      # prior distributions
  beta0C ~ dnorm(0.0, 0.00001);
  beta ~ dnorm(0.0, 0.00001);
  for(j in 1 : 2) {
    for(k in 1 : 2){
      phi[j, k] ~ dunif(0.0, 1.0)
    }
  }
}

# calculate gamma1 = P(x=1|d=0) and gamma2 = P(x=1|d=1)
gamma1 <- 1 / (1 + (1 + exp(beta0C + beta)) / (1 + exp(beta0C)) * (1 - q) / q)
gamma2 <- 1 / (1 + (1 + exp(-beta0C - beta)) / (1 + exp(-beta0C)) * (1 - q) / q)
}

```

Data

Inits

```
list(beta0C = 0, beta = 0, q = .5, phi = structure(.Data = c(.5,.5,.5,.5), .Dim = c(2, 2)))
```

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

a) Without over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
beta	0.6362	0.339	0.021	-0.03599	0.6471	1.275
beta0C	-0.9149	0.1892	0.01162	-1.293	-0.9149	-0.5555
gamma1	0.4387	0.05574	0.003244	0.3307	0.4375	0.548
gamma2	0.5938	0.06385	0.003815	0.4704	0.5937	0.7215
phi[1,1]	0.3166	0.05361	0.002939	0.2105	0.3169	0.4184
phi[1,2]	0.2198	0.08186	0.004233	0.07575	0.2142	0.3922
phi[2,1]	0.5697	0.06342	0.00327	0.4415	0.5707	0.6928
phi[2,2]	0.7615	0.06525	0.003439	0.6367	0.7624	0.8856
q	0.4942	0.04408	0.00248	0.4076	0.494	0.5799

b) With over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
beta	0.6047	0.3584	0.01814	-0.08128	0.5939	1.32
beta0C	-0.8995	0.1975	0.009534	-1.31	-0.8899	-0.5307
gamma1	0.4425	0.05506	0.002617	0.3337	0.4426	0.5465
gamma2	0.59	0.06539	0.002901	0.4581	0.59	0.7166
phi[1,1]	0.3138	0.05294	0.002304	0.208	0.3151	0.4134
phi[1,2]	0.2153	0.08102	0.00311	0.07448	0.2097	0.3857
phi[2,1]	0.5711	0.06162	0.00239	0.4514	0.5709	0.6922
phi[2,2]	0.767	0.06239	0.002461	0.6456	0.7677	0.8857
q	0.4953	0.04221	0.001749	0.4125	0.4947	0.5789

Re-parameterised model with centred covariates:

```

model
{
  for (i in 1 : N) {
    x[i] ~ dbern(q)      # incidence of HSV
    logit(p[i]) <- beta0 + beta * (x[i] - mean(w[])) # logistic model
    d[i] ~ dbern(p[i])    # incidence of cancer
  }
}

```

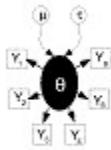
```

x1[i] <- x[i] + 1
d1[i] <- d[i] + 1
w[i] ~ dbern(phi[x1[i], d1[i]]) # incidence of w
}
q ~ dunif(0.0, 1.0) # prior distributions
beta0 ~ dnorm(0.0, 0.00001);
beta ~ dnorm(0.0, 0.00001);
for(j in 1 : 2) {
  for(k in 1 : 2){
    phi[j, k] ~ dunif(0.0, 1.0)
  }
}
# calculate gamma1 = P(x=1|d=0) and gamma2 = P(x=1|d=1)
gamma1 <- 1 / (1 + (1 + exp(beta0C + beta)) / (1 + exp(beta0C)) * (1 - q) / q)
gamma2 <- 1 / (1 + (1 + exp(-beta0C - beta)) / (1 + exp(-beta0C)) * (1 - q) / q)
beta0C <- beta0 - mean(w[])
beta
}
```

Inits

```
list(beta0 = 0, beta = 0, q = .5, phi = structure(.Data = c(.5,.5,.5,.5), .Dim = c(2, 2)))
```

node	mean	sd	MC error	2.5%	median	97.5%	start
beta	0.611	0.3568	0.01482	-0.0672	0.607	1.329	1001
beta0C	-0.9008	0.1974	0.007949	-1.311	-0.892	-0.5356	1001
gamma1	0.439	0.05494	0.002389	0.3302	0.4397	0.5465	1001
gamma2	0.5881	0.06518	0.002884	0.4583	0.5875	0.7143	1001
phi[1,1]	0.3181	0.05462	0.00209	0.2088	0.3199	0.4222	1001
phi[1,2]	0.2205	0.08035	0.003347	0.07782	0.2157	0.3852	1001
phi[2,1]	0.5675	0.06469	0.002403	0.4407	0.5678	0.6926	1001
phi[2,2]	0.7657	0.06314	0.0025	0.6417	0.7657	0.8871	1001
q	0.4924	0.04224	0.001933	0.4087	0.4929	0.5758	1001



BUGS Jaws: repeated measures analysis of variance

Elston and Grizzle (1962) present repeated measurements of ramus (jaw) bone height on a cohort of 20 boys over an 18 month period:

Subject	Age (in years)			
	8.0	8.5	9.0	9.5
1	47.8	48.8	49.0	49.7
2	46.4	47.3	47.7	48.4
3	46.3	46.8	47.8	48.5
.
.
19	46.2	47.5	48.1	48.4
20	46.3	47.6	51.3	51.8
Mean	48.7	49.6	50.6	51.5
Variance	6.4	6.5	6.9	7.5

Interest focuses on describing the average growth curve of the ramus bone. The 4 measurements $\mathbf{Y}_i = \{Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4}\}$ for each child i are assumed to be correlated and follow a multivariate normal (MVN) distribution with unknown population mean vector μ and precision matrix Ω . That is $\mathbf{Y}_i \sim \text{MVN}(\mu, \Omega)$

The following location models for the population mean μ were fitted in turn:

$$\begin{aligned} E(\mu_i) &= \beta_0 && \text{Constant height} \\ E(\mu_i) &= \beta_0 + \beta_1 x_j && \text{Linear growth curve} \\ E(\mu_i) &= \beta_0 + \beta_1 x_j + \beta_2 x_j^2 && \text{Quadratic growth curve} \end{aligned}$$

where x_j = age at j th measurement. Non-informative independent normal priors were specified for the regression coefficients β_0 , β_1 , and β_2 . The population precision matrix Ω was assumed to follow a Wishart(R, ρ) distribution. To represent vague prior knowledge, we chose the degrees of freedom ρ for this distribution to be as small as possible (i.e. 4, the rank of Ω). The scale matrix R was specified as a 4×4 diag(1) matrix which represents an assessment of the order of magnitude of the covariance matrix Ω^{-1} for \mathbf{Y}_i (see subsection on the use of the Wishart distribution in the "Multivariate normal nodes" section of the Classic BUGS manual (version 0.50). Note that except for cases with very few individuals, the choice of R has little effect on the posterior estimate of Ω^{-1} (Lindley, 1970).

BUGS language for the Jaws example

```
model
{
  beta0 ~ dnorm(0.0, 0.001)
  beta1 ~ dnorm(0.0, 0.001)
  for (i in 1:N) {
    Y[i, 1:M] ~ dmnorm(mu[], Omega[ , ])
  }
}
```

```

for(j in 1:M) {
  mu[j] <- beta0 + beta1* age[j]
}
Omega[1 : M , 1 : M] ~ dwish(R[ , ], 4)
Sigma[1 : M , 1 : M] <- inverse(Omega[ , ])
}

}

```

Data

```

⇒list(M = 4, N = 20, Y = structure(
  .Data = c(47.8, 48.8, 49.0, 49.7,
          46.4, 47.3, 47.7, 48.4,
          46.3, 46.8, 47.8, 48.5,
          45.1, 45.3, 46.1, 47.2,
          47.6, 48.5, 48.9, 49.3,
          52.5, 53.2, 53.3, 53.7,
          51.2, 53.0, 54.3, 54.5,
          49.8, 50.0, 50.3, 52.7,
          48.1, 50.8, 52.3, 54.4,
          45.0, 47.0, 47.3, 48.3,
          51.2, 51.4, 51.6, 51.9,
          48.5, 49.2, 53.0, 55.5,
          52.1, 52.8, 53.7, 55.0,
          48.2, 48.9, 49.3, 49.8,
          49.6, 50.4, 51.2, 51.8,
          50.7, 51.7, 52.7, 53.3,
          47.2, 47.7, 48.4, 49.5,
          53.3, 54.6, 55.1, 55.3,
          46.2, 47.5, 48.1, 48.4,
          46.3, 47.6, 51.3, 51.8),
  .Dim = c(20, 4)),
  age = c(8.0, 8.5, 9.0, 9.5),
  R = structure(
    .Data = c(1, 0, 0, 0,
             0, 1, 0, 0,
             0, 0, 1, 0,
             0, 0, 0, 1), .
  Dim = c(4, 4)))⇒

```

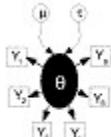
Inits

```
list(beta0 = 40, beta1 = 1)
```

Results

Time for 10000 updates 43s on 200MHz Pentium Pro. A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
Sigma[1,1]	6.694	2.345	0.02285	3.545	6.244	12.45	1001	10000
Sigma[1,2]	6.49	2.325	0.02302	3.355	6.039	12.23	1001	10000
Sigma[1,3]	6.068	2.3	0.02222	2.975	5.613	11.69	1001	10000
Sigma[1,4]	5.831	2.308	0.02223	2.687	5.379	11.52	1001	10000
Sigma[2,1]	6.49	2.325	0.02302	3.355	6.039	12.23	1001	10000
Sigma[2,2]	6.807	2.395	0.02401	3.591	6.344	12.79	1001	10000
Sigma[2,3]	6.463	2.38	0.02314	3.241	5.996	12.4	1001	10000
Sigma[2,4]	6.237	2.389	0.02312	2.985	5.767	12.22	1001	10000
Sigma[3,1]	6.068	2.3	0.02222	2.975	5.613	11.69	1001	10000
Sigma[3,2]	6.463	2.38	0.02314	3.241	5.996	12.4	1001	10000
Sigma[3,3]	7.307	2.568	0.02493	3.824	6.817	13.69	1001	10000
Sigma[3,4]	7.297	2.613	0.02551	3.733	6.804	13.82	1001	10000
Sigma[4,1]	5.831	2.308	0.02223	2.687	5.379	11.52	1001	10000
Sigma[4,2]	6.237	2.389	0.02312	2.985	5.767	12.22	1001	10000
Sigma[4,3]	7.297	2.613	0.02551	3.733	6.804	13.82	1001	10000
Sigma[4,4]	7.904	2.765	0.02782	4.119	7.352	14.71	1001	10000
beta0	33.67	1.987	0.1016	29.81	33.68	37.62	1001	10000
beta1	1.872	0.2267	0.01159	1.427	1.871	2.319	1001	10000
mu[1]	48.65	0.5447	0.01001	47.56	48.65	49.72	1001	10000
mu[2]	49.58	0.5358	0.005714	48.52	49.59	50.63	1001	10000
mu[3]	50.52	0.5505	0.005682	49.42	50.53	51.61	1001	10000
mu[4]	51.46	0.5871	0.009954	50.28	51.46	52.61	1001	10000



BUGS Birats: a bivariate normal hierarchical model

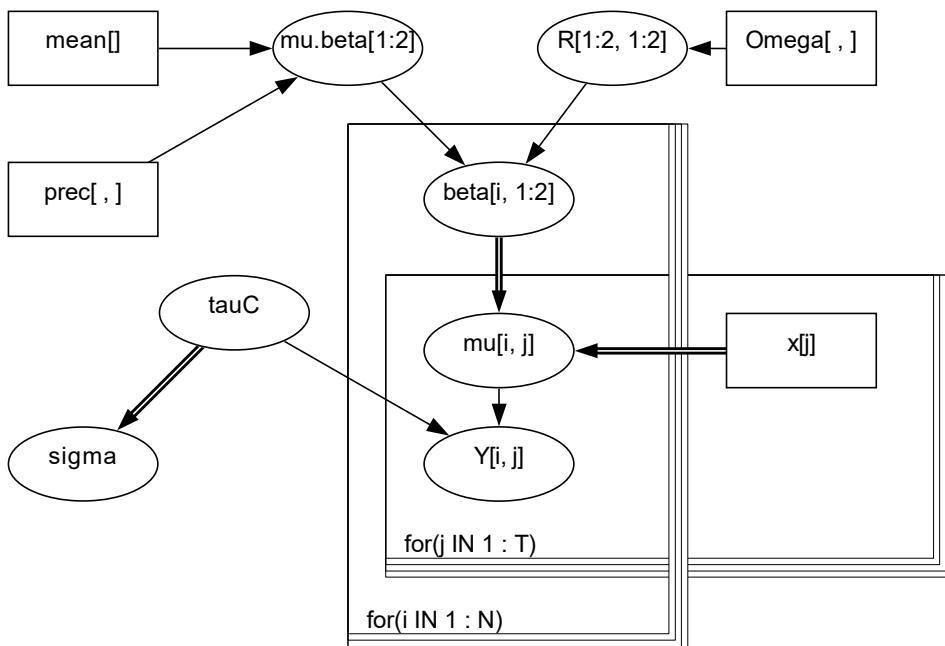
We return to the Rats example, and illustrate the use of a multivariate Normal (MVN) population distribution for the regression coefficients of the growth curve for each rat. This is the model adopted by Gelfand et al (1990) for these data, and assumes *a priori* that the intercept and slope parameters for each rat are correlated. For example, positive correlation would imply that initially heavy rats (high intercept) tend to gain weight more rapidly (steeper slope) than lighter rats. The model is as follows

$$\begin{aligned} Y_{ij} &\sim \text{Normal}(\mu_{ij}, \tau_c) \\ \mu_{ij} &= \beta_{1i} + \beta_{2i} x_j \\ \beta_i &\sim \text{MVN}(\mu_\beta, \Omega) \end{aligned}$$

where Y_{ij} is the weight of the i th rat measured at age x_j , and β_i denotes the vector (β_{1i}, β_{2i}) . We assume 'non-informative' independent univariate Normal priors for the separate components μ_{β_1} and μ_{β_2} . A Wishart(R, ρ) prior was specified for Ω , the population precision matrix of the regression coefficients. To represent vague prior knowledge, we chose the the degrees of freedom ρ for this distribution to be as small as possible (i.e. 2, the rank of Ω). The scale matrix was specified as

$$R = \begin{vmatrix} 200, & 0 \\ 0, & 0.2 \end{vmatrix}$$

This represents our prior guess at the order of magnitude of the covariance matrix Ω^{-1} for β_i (see Classic BUGS manual (version 0.5) section on Multivariate normal models), and is equivalent to the prior specification used by Gelfand et al. Finally, a non-informative Gamma(0.001, 0.001) prior was assumed for the measurement precision τ_c .



```

model
{
  for( i in 1 : N ) {
    beta[i , 1:2] ~ dmnorm(mu.beta[], R[ , ])
    for( j in 1 : T ) {
      Y[i , j] ~ dnorm(mu[i , j], tauC)
      mu[i , j] <- beta[i , 1] + beta[i , 2] * x[j]
    }
  }

  mu.beta[1:2] ~ dmnorm(mean[],prec[ , ])
  R[1:2 , 1:2] ~ dwish(Omega[ , ], 2)
  tauC ~ dgamma(0.001, 0.001)
  sigma <- 1 / sqrt(tauC)
}

```

Data

```

⇒list(x = c(8.0, 15.0, 22.0, 29.0, 36.0), N = 30, T = 5,
Omega = structure(.Data = c(200, 0, 0, 0.2), .Dim = c(2, 2)),
mean = c(0,0),
prec = structure(.Data = c(1.0E-6, 0, 0, 1.0E-6), .Dim = c(2, 2)),
Y = structure(
  .Data =  c(151, 199, 246, 283, 320,
            145, 199, 249, 293, 354,
            147, 214, 263, 312, 328,
            155, 200, 237, 272, 297,
            135, 188, 230, 280, 323,
            159, 210, 252, 298, 331,
            141, 189, 231, 275, 305,
            159, 201, 248, 297, 338,
            177, 236, 285, 350, 376,
            134, 182, 220, 260, 296,
            160, 208, 261, 313, 352,
            143, 188, 220, 273, 314,
            154, 200, 244, 289, 325,
            171, 221, 270, 326, 358,
            163, 216, 242, 281, 312,
            160, 207, 248, 288, 324,
            142, 187, 234, 280, 316,
            156, 203, 243, 283, 317,
            157, 212, 259, 307, 336,
            152, 203, 246, 286, 321,
            154, 205, 253, 298, 334,
            139, 190, 225, 267, 302,
            146, 191, 229, 272, 302,
            157, 211, 250, 285, 323,
            132, 185, 237, 286, 331,
            160, 207, 257, 303, 345,
            169, 216, 261, 295, 333,
            157, 205, 248, 289, 316,
            137, 180, 219, 258, 291,
            153, 200, 244, 286, 324),
  .Dim = c(30,5)))⇒

```

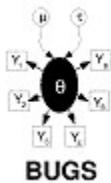
Inits

```
⇒list(mu.beta = c(0,0), tauC = 1,
      beta = structure(
        .Data = c(100,6,100,6,100,6,100,6,100,6,
                 100,6,100,6,100,6,100,6,100,6,
                 100,6,100,6,100,6,100,6,100,6,
                 100,6,100,6,100,6,100,6,100,6,
                 100,6,100,6,100,6,100,6,100,6),
        .Dim = c(30, 2)),
      R = structure(.Data = c(1,0,0,1), .Dim = c(2, 2)))⇒
```

Results

Time for 10000 updates was 17s on 200MHz Pentium Pro. A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu.beta[1]	106.6	2.355	0.03929	102.0	106.6	111.3	1001	10000
mu.beta[2]	6.183	0.1077	0.001501	5.97	6.183	6.397	1001	10000
sigma	6.151	0.4735	0.008216	5.315	6.12	7.166	1001	10000



Schools: ranking school examination results using multivariate hierarchical models

Goldstein et al. (1993) present an analysis of examination results from inner London schools. They use hierarchical or multilevel models to study the between-school variation, and calculate school-level residuals in an attempt to differentiate between 'good' and 'bad' schools. Here we analyse a subset of this data and show how to calculate a rank ordering of schools and obtain credible intervals on each rank.

Data

Standardized mean examination scores (Y) were available for 1978 pupils from 38 different schools. The median number of pupils per school was 48, with a range of 1–198. Pupil-level covariates included gender plus a standardized London Reading Test (LRT) score and a verbal reasoning (VR) test category (1, 2 or 3, where 1 represents the highest ability group) measured when each child was aged 11. Each school was classified by gender intake (all girls, all boys or mixed) and denomination (Church of England, Roman Catholic, State school or other); these were used as categorical school-level covariates.

Model

We consider the following model, which essentially corresponds to Goldstein et al.'s model 1.

$$\begin{aligned} Y_{ij} &\sim \text{Normal}(\mu_{ij}, \tau_{ij}) \\ \mu_{ij} &= \alpha_{1j} + \alpha_{2j} \text{LRT}_{ij} + \alpha_{3j} \text{VR1}_{ij} + \beta_1 \text{LRT}_{ij}^2 + \beta_2 \text{VR2}_{ij} + \beta_3 \text{Girl}_{ij} \\ &\quad + \beta_4 \text{Girls' school}_j + \beta_5 \text{Boys' school}_j + \beta_6 \text{CE school}_j \\ &\quad + \beta_7 \text{RC school}_j + \beta_8 \text{other school}_j \\ \log \tau_{ij} &= \theta + \phi \text{LRT}_{ij} \end{aligned}$$

where i refers to pupil and j indexes school. We wish to specify a regression model for the variance components, and here we model the logarithm of τ_{ij} (the inverse of the between-pupil variance) as a linear function of each pupil's LRT score. This differs from Goldstein et al.'s model which allows the variance σ_{ij}^2 to depend linearly on LRT. However, such a parameterization may lead to negative estimates of σ_{ij}^2 .

Prior distributions

The fixed effects β_k ($k=1,\dots,8$), θ and ϕ were assumed to follow vague independent Normal distributions with zero mean and low precision = 0.0001. The random school-level coefficients α_{kj} ($k = 1,2,3$) were assumed to arise from a multivariate normal population distribution with unknown mean γ and covariance matrix Σ . A non-informative multivariate normal prior was then specified for the population mean γ , whilst the inverse covariance matrix $T = \Sigma^{-1}$ was assumed to follow a Wishart distribution. To represent vague prior knowledge, we chose the degrees of freedom for this distribution to be as small as possible (i.e. 3, the rank of T). The scale matrix R was specified as

0.1	0.005	0.005
0.005	0.01	0.005
0.005	0.005	0.01

which represents our prior guess at the order of magnitude of Σ .

The BUGS code is given below:

model

```

    {
for(p in 1 : N) {

  Y[p] ~ dnorm(mu[p], tau[p])
  mu[p] <- alpha[school[p], 1] + alpha[school[p], 2] * LRT[p]
    + alpha[school[p], 3] * VR[p, 1] + beta[1] * LRT2[p]
    + beta[2] * VR[p, 2] + beta[3] * Gender[p]
    + beta[4] * School.gender[p, 1] + beta[5] * School.gender[p, 2]
    + beta[6] * School.denom[p, 1] + beta[7] * School.denom[p, 2]
    + beta[8] * School.denom[p, 3]
  log(tau[p]) <- theta + phi * LRT[p]
  sigma2[p] <- 1 / tau[p]
  LRT2[p] <- LRT[p] * LRT[p]

}

min.var <- exp(-(theta + phi * (-34.6193))) # lowest LRT score = -34.6193
max.var <- exp(-(theta + phi * (37.3807))) # highest LRT score = 37.3807

# Priors for fixed effects:
for (k in 1 : 8) { beta[k] ~ dnorm(0.0, 0.0001) }
theta ~ dnorm(0.0, 0.0001); phi ~ dnorm(0.0, 0.0001)

# Priors for random coefficients:
for (j in 1 : M) {
  alpha[j, 1:3 ] ~ dmnorm(gamma[1:3 ], T[1:3 ,1:3 ]);
    alpha1[j] <- alpha[j,1]
}

# Hyper-priors:
gamma[1:3 ] ~ dmnorm(mn[1:3 ], prec[1:3 ,1:3 ]);
T[1:3 ,1:3 ] ~ dwish(R[1:3 ,1:3 ], 3)
}

```

Data

),

$LRT = c$
 -10.6193, 8.3806999999999999, -12.6193, 14.3807, 9.3806999999999999, 2.3807, 9.3806999999999999, -7.6193, -4.6193, 0.3807,
 2.3807, -3.6193, 4.3807, -1.6193, 14.3807, -10.6193, 6.3807, 14.3807, -4.6193, 1.3807,
 -9.6193000000000001, -7.6193, -1.6193, -7.6193, 14.3807, -14.6193, 11.3807, -18.6193, 10.3807, 15.3807,
 -9.6193000000000001, -22.6193, 19.3807, 10.3807, 5.3807, -1.6193, 10.3807, 1.3807, -6.6193, -13.6193,
 -7.6193, -3.6193, 16.3807, 8.3806999999999999, 1.3807, -0.6193, 6.3807, -13.6193, -8.6193000000000001, -20.6193,
 -2.6193, 13.3807, 3.3807, -8.6193000000000001, 8.3806999999999999, -5.6193, -12.6193, 0.3807, -22.6193, -4.6193,
 -34.6193, -1.6193, 18.3807, 16.3807, -12.6193, 8.3806999999999999, 20.3807, -24.6193, -11.6193, -9.6193000000000001,
 -3.6193, -3.6193, -2.6193, -3.6193, 4.3807, -1.6193, -17.6193, -2.6193, 15.3807, -19.6193,
 -2.6193, -2.6193, 10.3807, 3.3807, 35.3807, 5.3807, 12.3807, -13.6193, 16.3807, -23.6193,
 -5.6193, 0.3807, -11.6193, -0.6193, -2.6193, -1.6193, 20.3807, 5.3807, 5.3807, 7.3807,
 20.3807, -0.6193, 9.3806999999999999, 1.3807, -10.6193, -0.6193, -4.6193, 1.3807, -4.6193, 16.3807,
 -3.6193, 8.3806999999999999, 3.3807, 9.3806999999999999, 3.3807, 3.3807, 1.3807, 9.3806999999999999, -2.6193, 11.3807,
 -24.6193, 7.3807, 10.3807, -6.6193, 27.3807, -1.6193, 18.3807, -5.6193, 6.3807, -7.6193,
 12.3807, -1.6193, 18.3807, 7.3807, 0.3807, 17.3807, 20.3807, 9.3806999999999999, -10.6193, -18.6193,
 -3.6193, -0.6193, -12.6193, 15.3807, 1.3807, 9.3806999999999999, -10.6193, 5.3807, 11.3807, -13.6193,
 5.3807, 17.3807, -1.6193, -5.6193, -10.6193, 10.3807, 7.3807, -7.6193, 19.3807, 11.3807,
 0.3807, 18.3807, -12.6193, 7.3807, -0.6193, 5.3807, 13.3807, -7.6193, -1.6193, -2.6193,
 26.3807, 0.3807, 6.3807, 7.3807, 4.3807, 23.3807, 12.3807, -7.6193, 18.3807, 10.3807,
 -12.6193, 0.3807, 6.3807, 15.3807, -0.6193, 9.3806999999999999, -15.6193, -3.6193, 17.3807, 8.3806999999999999,
 25.3807, 10.3807, -5.6193, 13.3807, 10.3807, -5.6193, 4.3807, 13.3807, 35.3807, -12.6193,
 3.3807, 23.3807, 19.3807, 19.3807, 20.3807, 4.3807, 0.3807, 9.3806999999999999, -22.6193, 5.3807,
 -4.6193, 3.3807, 6.3807, -21.6193, -12.6193, 5.3807, 5.3807, -17.6193, 7.3807, 24.3807,
 17.3807, 12.3807, -5.6193, 2.3807, 5.3807, 11.3807, 8.3806999999999999, 29.3807, -6.6193, 4.3807,
 22.3807, -1.6193, 5.3807, -12.6193, 12.3807, 15.3807, 23.3807, 7.3807, 8.3806999999999999, 23.3807,
 0.3807, -2.6193, -9.6193000000000001, 2.3807, 23.3807, -8.6193000000000001, 4.3807, 1.3807, 21.3807, 3.3807,
 8.3806999999999999, -6.6193, 18.3807, -12.6193, -7.6193, -0.6193, 12.3807, -9.6193000000000001, 4.3807, -8.6193000000000001,
 4.3807, 2.3807, 22.3807, 8.3806999999999999, 16.3807, 13.3807, 11.3807, 7.3807, 6.3807, 18.3807,
 6.3807, 10.3807, 10.3807, -11.6193, 7.3807, -0.6193, 18.3807, 4.3807, 4.3807, -7.6193,
 0.3807, 8.3806999999999999, 4.3807, 12.3807, 5.3807, 2.3807, 22.3807, 13.3807, -5.6193, 2.3807,
 5.3807, 2.3807, -9.6193000000000001, -1.6193, 3.3807, 6.3807, 3.3807, -11.6193, 1.3807, -12.6193,
 17.3807, 7.3807, -4.6193, 8.3806999999999999, -14.6193, 9.3806999999999999, 13.3807, -6.6193, 3.3807, -8.6193000000000001,
 0.3807, 11.3807, 12.3807, 20.3807, 13.3807, 13.3807, 21.3807, 17.3807, 1.3807, 3.3807,
 -1.6193, 0.3807, 3.3807, -8.6193000000000001, 14.3807, -11.6193, -9.6193000000000001, 14.3807, -6.6193, 4.3807,
 -1.6193, 15.3807, 17.3807, 15.3807, 9.3806999999999999, 16.3807, -2.6193, 1.3807, 2.3807, 0.3807,
 -10.6193, -2.6193, -3.6193, -5.6193, 14.3807, -1.6193, -1.6193, 7.3807, 7.3807, -11.6193,
 -15.6193, -10.6193, -0.6193, 4.3807, 2.3807, 10.3807, -6.6193, 3.3807, -11.6193, 18.3807,
 -5.6193, 20.3807, 15.3807, -6.6193, -7.6193, -4.6193, 26.3807, 16.3807, 9.3806999999999999, 5.3807,
 4.3807, 5.3807, 5.3807, -3.6193, 19.3807, -6.6193, 8.3806999999999999, 14.3807, 4.3807, 0.3807,
 12.3807, 1.3807, 7.3807, 20.3807, 10.3807, 0.3807, 1.3807, -4.6193, -6.6193, 20.3807,
 -11.6193, 0.3807, 18.3807, 9.3806999999999999, -0.6193, 8.3806999999999999, 1.3807, 8.3806999999999999, 12.3807, 6.3807,
 -14.6193, 7.3807, 14.3807, 8.3806999999999999, 11.3807, 18.3807, 24.3807, 17.3807, -5.6193, 28.3807,
 12.3807, 5.3807, 0.3807, 14.3807, 13.3807, -6.6193, -4.6193, 5.3807, 18.3807,
 5.3807, -2.6193, 1.3807, 6.3807, 0.3807, 8.3806999999999999, -6.6193, 16.3807, 8.3806999999999999, 7.3807,
 -1.6193, -8.6193000000000001, -6.6193, 11.3807, 4.3807, 7.3807, 2.3807, -3.6193, 7.3807, -2.6193,
 8.3806999999999999, -6.6193, 16.3807, -17.6193, -0.6193, -3.6193, 11.3807, 4.3807, 3.3807, -19.6193,
 20.3807, -15.6193, -0.6193, 8.3806999999999999, 1.3807, 18.3807, 3.3807, 9.3806999999999999, 15.3807, 0.3807,
 8.3806999999999999, 11.3807, -0.6193, 15.3807, 18.3807, 9.3806999999999999, -2.6193, 2.3807, 13.3807, -9.6193000000000001,
 13.3807, -18.6193, 3.3807, 2.3807, -10.6193, 8.3806999999999999, -6.6193, 24.3807, -2.6193, 15.3807,
 6.3807, 7.3807, 5.3807, 7.3807, 14.3807, -11.6193, -8.6193000000000001, -2.6193, 12.3807, -19.6193,
 -5.6193, -25.6193, -23.6193, -0.6193, 3.3807, -13.6193, -5.6193, 17.3807, -7.6193, -1.6193,
 -19.6193, -3.6193, 15.3807, 21.3807, 1.3807, -5.6193, -20.6193, 5.3807, 5.3807, -27.6193,
 -7.6193, -1.6193, 7.3807, 12.3807, -5.6193, -12.6193, 8.3806999999999999, 28.3807, -1.6193, -2.6193,
 -0.6193, -24.6193, 8.3806999999999999, -19.6193, 13.3807, 23.3807, 9.3806999999999999, 21.3807, -4.6193, 10.3807.

7.3807, -15.6193, -7.6193, -11.6193, -2.6193, -8.6193000000000001, -31.6193, 7.3807, -6.6193, -8.6193000000000001,
-8.6193000000000001, -14.6193, -1.6193, -16.6193, 0.3807, -4.6193, 3.3807, 5.3807, 11.3807, -7.6193,
-0.6193, 2.3807, -18.6193, -0.6193, 10.3807, -23.6193, -4.6193, 20.3807, -6.6193, -0.6193,
-9.6193000000000001, -9.6193000000000001, 3.3807, 5.3807, -2.6193, 4.3807, -1.6193, 10.3807, 2.3807, -12.6193,
-9.6193000000000001, 2.3807, -2.6193, 6.3807, -15.6193, -15.6193, 10.3807, 7.3807, 5.3807, -13.6193,
8.380699999999999, -12.6193, -9.6193000000000001, -19.6193, 5.3807, -1.6193, 4.3807, -4.6193, -7.6193, -25.6193,
0.3807, 16.3807, -34.6193, 5.3807, -17.6193, -18.6193, -3.6193, 1.3807, 4.3807, -6.6193,
-7.6193, -9.6193000000000001, -2.6193, -8.6193000000000001, 12.3807, -11.6193, 10.3807, -17.6193, -4.6193, 7.3807,
3.3807, 10.3807, -3.6193, -9.6193000000000001, -4.6193, -16.6193, 14.3807, -4.6193, -28.6193, 1.3807,
-8.6193000000000001, 4.3807, -9.6193000000000001, 4.3807, -22.6193, -4.6193, 2.3807, -11.6193, -11.6193, -8.6193000000000001,
-1.6193, 4.3807, 6.3807, 0.3807, -22.6193, -9.6193000000000001, 5.3807, 4.3807, -8.6193000000000001, -3.6193,
13.3807, -17.6193, -12.6193, 2.3807, -5.6193, -7.6193, -9.6193000000000001, -13.6193, -10.6193, 8.3806999999999999,
16.3807, -13.6193, -1.6193, -3.6193, -4.6193, 10.3807, 5.3807, -3.6193, -0.6193, -26.6193,
-13.6193, -13.6193, -12.6193, -11.6193, 8.3806999999999999, 8.3806999999999999, -4.6193, -6.6193, 35.3807, 0.3807,
7.3807, 7.3807, -22.6193, -12.6193, -22.6193, 6.3807, -0.6193, 4.3807, -4.6193, -6.6193,
10.3807, -8.6193000000000001, -4.6193, -2.6193, -0.6193, 2.3807, 1.3807, -23.6193, 8.3806999999999999, -5.6193,
23.3807, -25.6193, -8.6193000000000001, -3.6193, -2.6193, 5.3807, -10.6193, 20.3807, -4.6193, 10.3807,
3.3807, 18.3807, 8.3806999999999999, -1.6193, 10.3807, 2.3807, 10.3807, -8.6193000000000001, 26.3807, 16.3807,
7.3807, 14.3807, -0.6193, -11.6193, -19.6193, 9.3806999999999999, 12.3807, -4.6193, -3.6193, 35.3807,
-5.6193, 35.3807, 23.3807, 14.3807, -1.6193, 3.3807, -10.6193, -6.6193, -0.6193, -6.6193,
-0.6193, 3.3807, 1.3807, -3.6193, -10.6193, -0.6193, 3.3807, -16.6193, 6.3807, 3.3807,
-13.6193, -5.6193, -0.6193, -27.6193, -23.6193, -2.6193, 6.3807, -3.6193, -21.6193, -7.6193,
-0.6193, -2.6193, 0.3807, 2.3807, 8.3806999999999999, 3.3807, -9.6193000000000001, -15.6193, -6.6193, -4.6193,
-12.6193, -1.6193, -10.6193, 0.3807, -13.6193, -11.6193, -12.6193, -6.6193, 0.3807, -1.6193,
-20.6193, -2.6193, -0.6193, -19.6193, -21.6193, -1.6193, -7.6193, 14.3807, -30.6193, -34.6193,
7.3807, -1.6193, 13.3807, 6.3807, 5.3807, -14.6193, -20.6193, -20.6193, 7.3807, -12.6193,
-30.6193, -6.6193, -5.6193, -5.6193, 21.3807, -10.6193, -19.6193, 1.3807, 3.3807, -5.6193,
6.3807, -0.6193, -15.6193, 1.3807, 7.3807, -0.6193, -11.6193, -17.6193, -21.6193, -13.6193,
13.3807, -5.6193, -18.6193, 0.3807, -26.6193, 14.3807, 1.3807, 1.3807, -0.6193, -18.6193,
-14.6193, -10.6193, 1.3807, 17.3807, -13.6193, 7.3807, -4.6193, -2.6193, -5.6193, 17.3807,
8.3806999999999999, -2.6193, 15.3807, 26.3807, 1.3807, -0.6193, -6.6193, 7.3807, -4.6193, 1.3807,
-21.6193, -12.6193, 4.3807, 5.3807, -4.6193, 1.3807, -1.6193, -2.6193, 12.3807, 4.3807,
-33.6193, 1.3807, 20.3807, 29.3807, -1.6193, 5.3807, 3.3807, 0.3807, -4.6193, -5.6193,
3.3807, 3.3807, 15.3807, -1.6193, -4.6193, -12.6193, -1.6193, 4.3807, -5.6193, -0.6193,
3.3807, 11.3807, 1.3807, 2.3807, 14.3807, 11.3807, -2.6193, -1.6193, 1.3807, -2.6193,
23.3807, -5.6193, 4.3807, -21.6193, -30.6193, 2.3807, -14.6193, 2.3807, 11.3807, -1.6193,
-9.6193000000000001, 6.3807, -13.6193, -26.6193, 12.3807, 11.3807, 17.3807, -33.6193, -5.6193, 29.3807,
-8.6193000000000001, -26.6193, 3.3807, -16.6193, 5.3807, -4.6193, -5.6193, -6.6193, -1.6193, 6.3807,
-1.6193, 2.3807, 7.3807, 11.3807, -1.6193, -2.6193, 16.3807, -5.6193, -2.6193, -11.6193,
-2.6193, -0.6193, -4.6193, 1.3807, 20.3807, -11.6193, -2.6193, -13.6193, -15.6193, 7.3807,
-3.6193, 1.3807, -7.6193, 16.3807, -4.6193, 0.3807, -3.6193, -6.6193, 20.3807, -8.6193000000000001,
12.3807, 1.3807, -17.6193, -3.6193, -18.6193, -13.6193, 23.3807, -15.6193, -9.6193000000000001, -10.6193,
4.3807, -21.6193, 4.3807, -17.6193, -5.6193, -0.6193, 9.3806999999999999, -15.6193, -5.6193, -16.6193,
-17.6193, -0.6193, 13.3807, 13.3807, 3.3807, -7.6193, 17.3807, -3.6193, 11.3807, -3.6193,
-9.6193000000000001, 8.3806999999999999, 8.3806999999999999, 5.3807, -3.6193, 6.3807, -5.6193, -19.6193, -30.6193, 10.3807,
19.3807, 8.3806999999999999, 13.3807, -0.6193, 16.3807, 16.3807, 7.3807, 2.3807, 13.3807, -1.6193,
2.3807, -31.6193, 5.3807, -0.6193, 21.3807, 21.3807, -10.6193, -13.6193, 36.3807, -2.6193,
21.3807, -13.6193, -0.6193, -26.6193, -13.6193, -13.6193, -13.6193, -0.6193, 25.3807, -4.6193,
6.3807, 17.3807, -13.6193, 2.3807, 20.3807, -23.6193, -6.6193, -7.6193, -5.6193, 3.3807,
-24.6193, -24.6193, -13.6193, -25.6193, -13.6193, 13.3807, -6.6193, -25.6193, -28.6193, 7.3807,
-2.6193, 1.3807, -5.6193, 1.3807, -12.6193, -1.6193, -2.6193, -25.6193, 9.3806999999999999, 3.3807,
-1.6193, -10.6193, 11.3807, -10.6193, 7.3807, 1.3807, 23.3807, -9.6193000000000001, 7.3807, -9.6193000000000001,
5.3807, -7.6193, -4.6193, -6.6193, -11.6193, 25.3807, 15.3807, -22.6193, 7.3807, 17.3807,
-13.6193, -13.6193, -29.6193, 6.3807, 5.3807, -0.6193, -19.6193, -23.6193, 4.3807, 14.3807,
-2.6193, -16.6193, 20.3807, -11.6193, 7.3807, 7.3807, 8.3806999999999999, -8.6193000000000001, 18.3807, -4.6193,
3.3807, -3.6193, -4.6193, -0.6193, 5.3807, 2.3807, 5.3807, -24.6193, -18.6193, -13.6193,
-11.6193, -0.6193, 7.3807, -7.6193, 16.3807, 16.3807, 17.3807, -34.6193, 9.3806999999999999, -10.6193, -8.6193000000000001,
-23.6193, -11.6193, -1.6193, -10.6193, -1.6193, -6.6193, -2.6193, 5.3807, 24.3807, 1.3807,
19.3807, -23.6193, 21.3807, -15.6193, 17.3807, -4.6193, 14.3807, -8.6193000000000001, -20.6193, 0.3807,
-4.6193, -6.6193, 18.3807, -5.6193, 0.3807, -21.6193, 8.3806999999999999, 7.3807, -11.6193, -7.6193,
23.3807, -22.6193, -0.6193, 4.3807, -3.6193, -0.6193, -4.6193, 6.3807, 1.3807, -18.6193,
29.3807, 10.3807, -0.6193, -3.6193, -0.6193, 1.3807, 2.3807, -9.6193000000000001, 7.3807, -14.6193,
13.3807, -4.6193, 10.3807, 16.3807, -8.6193000000000001, -6.6193, -25.6193, 8.3806999999999999, -11.6193, 5.3807,
5.3807, -9.6193000000000001, 7.3807, -18.6193, 18.3807, 6.3807, 18.3807, 12.3807, 0.3807, -11.6193,
6.3807, -27.6193, -13.6193, 1.3807, -2.6193, 2.3807, -9.6193000000000001, 20.3807, -4.6193, 4.3807,
-6.6193, 4.3807, -14.6193, 8.3806999999999999, 9.3806999999999999, -13.6193, -0.6193, 5.3807, 11.3807, -25.6193,
-9.6193000000000001, 17.3807, -4.6193, 12.3807, -22.6193, -6.6193, 0.3807, 18.3807, 31.3807, 18.3807,
-7.6193, -8.6193000000000001, 16.3807, -4.6193, 4.3807, -4.6193, 8.3806999999999999, -6.6193, 15.3807, 9.3806999999999999,
3.3807, -8.6193000000000001, 5.3807, -29.6193, -34.6193, -11.6193, -13.6193, 5.3807, -6.6193, 7.3807,

),.Dim = c(1978, 3)),
Gender = c(

Gender = c(

1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0,
0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0,
1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1,
0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1,

1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0,
 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1,
 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0,
 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0,
 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0,
 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0,
 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0,
 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0,
 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0,
 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1,
 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1,
 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0,
 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1,
 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1,
 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1,
 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1,
 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1,
 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1,
 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1,
 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1,
 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0
 $\text{).Dim} = \text{c}(1978, 2)$,
 $\text{Y} = \text{c}$ (
-0.6058763, 0.9911532, -1.5640939, 0.112787, 0.7515988, 0.5918959, 0.7515988, -1.0849851, -1.8036484, 1.3904106,
0.9113018000000001, 0.4321929, 1.1508562, -0.3663218, 1.5501135, -0.2066189, 0.3523414, 0.4321929, -0.9252821, 1.0710047,
-0.3663218, -0.1267674, -1.244688, -0.2864704, -0.4461733, -0.5260248, 0.1926385, -1.4842425, 1.1508562, -0.2066189,
-1.404391, -0.6857277000000001, 0.112787, 0.27249, 1.1508562, 0.112787, -0.5260248, 0.3523414, 1.1508562, 0.27249,
-0.1267674, 0.5120444, 0.5120444, 0.9113018000000001, 0.9113018000000001, -1.0051336, 0.1926385, -0.5260247, -
0.6857277000000001, -0.9252821,
0.4321929, 0.2724899, 0.7515988, -0.5260247, -0.3663218, -1.6439454, -1.5640939, 0.2724899, -1.4043909, 0.1926384,
-1.1648365, -0.6058762, 1.8695194, 1.3904105, -1.7237968, -1.244688, 2.6680341, -1.3245395, -1.3245395, -0.8454306,
-0.2066188, -1.0051336, 0.4321929, -0.3663218, -0.1267674, -1.1648365, -2.0432027, 0.5120443, 1.1508561, -0.5260247,
0.7515988, -0.3663218, 0.112787, -0.1267674, 1.2307076, 1.470262, 1.2307076, 1.3105591, 0.6717473, -1.3245395,
-0.3663218, 1.0710047, -0.0469159, 0.6717473, -0.6058762, -1.0849851, -0.5260247, 1.5501135, 0.2724899, 0.3523414,
2.1090738, -0.6058762, 1.0710047, 0.4321929, -1.1648365, -0.0469159, -1.244688, -0.1267674, -0.5260247, 2.0292223,
0.6717473, 0.5918958, -0.2066188, 1.470262, 0.4321929, -0.7655792, 0.5918958, -0.4461733, 0.0329355, 1.2307076,
-0.7655792, 0.4321929, -0.7655792, 0.3523414, 0.5918958, -1.4043909, 1.7896679, -0.4461733, 0.0329355, 0.0329355,
-0.3663218, -0.0469159, 1.470262, -0.5260247, 2.5881827, -0.0469159, 0.3523414, 1.629965, -1.3245395, -1.8834998,
-0.6857277000000001, -0.0469159, -1.8834998, -0.3663218, 0.5918958, -0.0469159, -1.8036483, 0.5120443, 0.9113017, -
1.4842424,
-1.4043909, 1.0710047, -0.2864703, -1.0849851, -0.2864703, 0.9911532, 1.629965, -0.5260247, 1.470262, 0.6717473,
-0.6857277000000001, 0.4321929, -0.6058762, 0.5918958, 0.112787, -0.6857277000000001, 1.2307076, -0.9252821, -0.0469159, -
1.244688,
0.112787, -1.1648365, -0.6857277000000001, -1.0849851, 1.3904105, 1.9493709, -1.0849851, -0.4461733, 2.1090738, -1.1648365,
-0.5260247, -0.2066188, 0.3523414, 1.7896679, 1.629965, 0.3523414, -1.0051336, 0.6717473, 1.5501135, -0.2066188,
2.0292223, -0.0469159, 0.3523414, 0.1926384, 0.9911532, -0.2864703, -1.244688, -0.1267674, 0.9911532, -0.6857277000000001,
-0.5260247, 0.8314502, 1.2307076, 0.9911532, 0.7515988, 1.0710047, -0.7655792, 0.5918958, -0.9252821, -0.2864703,
-1.5640939, 0.2724899, -0.4461733, 0.8314502, -1.1648365, 0.7515988, -0.0469159, -1.7237968, -1.3245395, 1.3904105,
1.0710047, 0.8314502, -1.0849851, 1.7098164, -0.1267674, -0.2066188, -1.0051336, 1.7098164, -0.3663218, 1.470262,
1.5501135, 0.9911532, 1.2307076, -0.8454306, -0.1267674, 0.5918958, 0.7515988, 1.3904105, 0.4321929, 0.112787,
0.3523414, 0.2724899, -1.4842424, 0.1926384, 2.1889253, 0.2724899, -1.4043909, -1.3245395, 1.470262, 1.0710047,
0.0329355, -0.2864703, 2.1889253, 0.1926384, -1.4043909, 0.0329355, 0.8314502, -1.8834998, -0.1267674, -0.1267674,
0.8314502, -0.0469159, 2.2687768, 0.5120443, 1.2307076, 0.7515988, 1.470262, 0.0329355, 1.470262, 0.1926384,
0.2724899, 0.0329355, 0.5918958, -1.0849851, 0.8314502, -1.7237968, 1.3904105, -1.1648365, 0.6717473, -0.6857277000000001,
-0.6857277000000001, -1.0051336, -0.6857277000000001, 0.9911532, 1.470262, -0.0469159, 1.9493709, 1.0710047, -0.2066188, -
0.0469159,
1.3904105, 0.6717473, 0.4321929, 0.8314502, -0.2066188, 1.1508561, -0.2864703, 1.2307076, -0.5260247, -0.6058762,
0.112787, 0.4321929, -1.4043909, -0.9252821, -1.6439454, 1.470262, 0.7515988, -0.8454306, -0.1267674, -0.6058762,
-0.0469159, 0.2724899, 2.5881827, 1.9493709, 0.4321929, -1.1648365, 1.9493709, 2.1889253, 0.8314502, 0.1926384,
0.3523414, 0.6717473, -0.4461733, -0.6058762, 0.8314502, -0.3663218, -1.7237968, 1.3904105, -0.8454306, 0.5120443,
-1.244688, 1.0710047, 0.5120443, 1.3904105, -0.4461733, -0.2864703, -0.1267674, -0.8454306, -1.4842424, 0.3523414,
-1.4842424, -0.0469159, -0.8454306, -0.1267674, 0.9113017, -0.9252821, -0.9252821, 0.2724899, -0.2066188, -1.0849851,
-1.0051336, -0.8454306, 0.0329355, 1.7896679, -0.2066188, 2.2687768, 0.0329355, -1.1648365, -1.0849851, 0.112787,
-0.6058762, 2.5083312, 1.2307076, -0.6857277000000001, -0.8454306, 0.6717473, 0.5918958, 1.3904105, 1.9493709, 0.0329355,
0.2724899, -0.3663218, 0.8314502, -0.1267674, 0.8314502, -1.9633513, 1.629965, 1.470262, -1.3245395, 0.3523414,
-0.6058762, -0.1267674, -0.5260247, -0.5260247, 1.9493709, -1.0051336, 0.0329355, -1.4842424, 0.9113017, 1.3904105,
-0.2066188, 0.8314502, 0.9911532, -0.3663218, -0.7655792, -0.6058762, -0.2066188, 0.112787, 0.6717473, -0.8454306,
0.9911532, 1.3105591, 1.0710047, -0.7655792, -0.0469159, 0.112787, 0.9113017, 0.0329355, 0.0329355, -0.8454306,
0.3523414, 0.112787, -0.2066188, -1.7237968, 0.8314502, 0.112787, -0.9252821, 0.5120443, -0.4461733, 0.5918958,
-0.0469159, -1.5640939, -1.8036483, -0.9252821, 0.5120443, 0.5120443, -0.6857277000000001, -0.3663218, -0.2066188,
0.0329355,
-0.4461733, -1.0051336, -1.7237968, 0.5120443, -1.4842424, 0.6717473, 1.0710047, -0.7655792, 0.8314502, -0.6058762,

-0.3663218, -1.4043909, -0.5260247, -1.244688, -0.2066188, -1.1648365, -0.1267674, -0.3663218, -0.0469159, -0.2864703,
-0.3663218, -0.7655792, -0.6058762, 0.2724899, 0.2724899, 0.9113017, 0.5918958, 0.8314502, 1.5501135, -1.1648365,
0.112787, 0.2724899, -0.2864703, 1.2307076, 0.8314502, 0.9113017, -0.5260247, -0.0469159, -0.2066188, -0.6857277000000001,
0.7515988, -1.1648365, -0.9252821, -0.7655792, -0.2864703, -0.2864703, -0.4461733, 0.112787, -0.8454306, -0.0469159,
-0.7655792, 1.1508561, -0.7655792, 0.112787, 0.5918958, -0.6857277000000001, -0.8454306, 0.7515988, 0.5120443, -1.9633513,
-0.2864703, -0.2066188, -1.4842424, -0.1267674, -0.5260247, -1.1648365, -1.4043909, 1.0710047, 0.1926384, 1.1508561,
-0.1267674, -0.8454306, 0.7515988, 1.7098164, -1.244688, -1.0849851, -1.9633513, 0.5120443, -0.3663218, 0.1926384,
-0.9252821, -0.2066188, 0.112787, -0.3663218, 0.5120443, -0.1267674, 0.5918958, 0.5120443, -0.3663218, -1.0051336,
0.8314502, -1.6439454, 1.470262, -0.3663218, 0.2724899, 1.470262, -0.5260247, 1.629965, 0.7515988, -0.1267674,
-1.9633513, 0.0329355, -1.0849851, -1.244688, 1.3904105, 0.5918958, -0.2864703, -0.2864703, -0.0469159, 0.4321929,
-0.7655792, -1.244688, 0.9911532, 1.1508561, -0.3663218, 1.5501135, -0.7655792, 0.5120443, 1.8695194, -0.1267674,
0.4321929, 0.5918958, -0.1267674, -1.3245395, -0.4461733, -0.9252821, 1.1508561, 1.7896679, 0.112787, -0.2864703,
-1.0849851, -1.3245395, 0.1926384, -1.1648365, -0.4461733, 0.112787, -0.2066188, 0.2724899, 0.3523414, -1.0051336,
-1.5640939, 0.112787, -0.2066188, -0.6058762, -0.6058762, -0.0469159, -0.2864703, -0.9252821, -0.9252821, -0.5260247,
1.3105591, -0.6058762, -1.6439454, 0.9113017, -1.4043909, 0.9911532, 1.3105591, -0.0469159, 0.4321929, 0.5918958,
0.5918958, 1.629965, 1.470262, -1.8036483, 0.0329355, -0.8454306, 0.1926384, 0.4321929, 0.9911532, -0.3663218,
-0.3663218, 1.3105591, 0.4321929, 0.5918958, -0.4461733, 0.8314502, -0.2864703, -0.6857277000000001, -0.8454306, -
1.6439454,
-0.7655792, -2.1230542, -1.244688, 2.5881827, -0.6857277000000001, -0.0469159, -0.3663218, 1.0710047, -0.3663218, -
0.4461733,
-0.6857277000000001, -0.0469159, 0.9113017, 0.9911532, -0.4461733, -0.6857277000000001, -1.0849851, -0.1267674, 0.5120443,
-1.0051336,
0.112787, 1.2307076, -0.2864703, 0.5918958, 0.112787, 0.7515988, -0.2066188, -0.3663218, -0.7655792, -1.5640939,
-1.0849851, 0.4321929, -0.6857277000000001, -0.7655792, 0.6717473, -1.244688, 0.3523414, 1.1508561, 2.0292223, -0.5260247,
0.0329355, 0.7515988, -0.0469159, 1.7896679, 0.112787, 0.112787, -0.2066188, 0.4321929, -1.0051336, 0.112787,
-0.6857277000000001, 0.0329355, -0.2066188, -0.4461733, -0.6058762, 1.7896679, -1.0849851, 0.9113017, -0.2066188, -
0.0469159,
1.0710047, -0.2066188, 1.8695194, 0.4321929, -0.0469159, 1.7896679, -0.9252821, 0.9113017, 1.2307076, 0.9113017,
-0.6857277000000001, 1.7098164, -0.3663218, -1.1648365, -0.3663218, -0.6058762, -1.3245395, -1.0051336, 0.8314502,
1.2307076,
1.7896679, -0.4461733, -0.5260247, 0.112787, 1.629965, 0.9911532, -1.0051336, 1.5501135, -0.0469159, -1.7237968,
0.6717473, -0.8454306, -1.0051336, 0.5120443, -1.8036483, -1.0051336, 0.112787, 1.470262, 0.5120443, -0.7655792,
1.0710047, 1.3105591, 1.1508561, 0.5120443, 0.9113017, 1.2307076, 1.0710047, -0.9252821, 0.1926384, 0.1926384,
-0.0469159, 0.6717473, 0.4321929, 0.1926384, 0.5918958, -0.0469159, -0.5260247, -0.2864703, 0.7515988, 0.112787,
-1.4842424, 0.5120443, 0.6717473, 1.7896679, -0.0469159, 1.9493709, 1.629965, -1.0051336, -0.2864703, 0.5120443,
0.4321929, 0.6717473, 0.8314502, -0.7655792, 1.7896679, -0.6058762, -1.244688, 1.7896679, 0.0329355, 1.629965,
-0.7655792, -0.6857277000000001, 2.34862828, 0.99115323, 0.03293554, -0.60587625, 1.23070765, -0.60587625, -0.7655792, -
0.12676741,
0.5918958600000001, -0.04691593, 0.67174733, -0.12676741, 0.11278701, 0.43219291, 0.7515988, 0.5918958600000001,
1.39041059, 1.62996502,
0.91130175, 0.99115323, -0.68572773, -0.7655792, 1.94937091, -1.16483657, -0.60587625, 2.10907386, 0.19263849,
0.5918958600000001,
0.7515988, -1.40439099, 2.02922239, 0.43219291, -0.68572773, -0.92528215, 1.15085617, 0.35234143, 0.43219291, -0.20661888,
1.55011354, -0.60587625, 2.18892533, 0.19263849, 0.67174733, 1.70981649, -1.88349983, 0.03293554, 1.62996502, -0.366321,
0.1926384, 0.7515988, 1.629965, -0.0469159, -0.1267674, 1.3904105, 1.3105591, -0.2066188, -0.2864703, 0.2724899,
2.0292223, 2.0292223, 1.5501135, 0.9113017, 1.5501135, 1.9493709, 0.3523414, 0.8314502, 1.5501135, 1.1508561,
1.3105591, -0.1267674, 1.2307076, 0.5120443, -0.3663218, 0.9113017, 0.9911532, -0.2864703, -0.4461733, 0.5120443,
0.5120443, 0.8314502, 0.5918958, 0.5918958, 1.3904105, 0.0329355, 1.629965, 1.7896679, 0.5918958, 1.470262,
2.2687768, 0.112787, 1.470262, -0.9252821, 1.3904105, 0.8314502, -0.28647, -1.0051336, 2.1090738, 0.2724899,
1.470262, 2.2687768, 0.5918958, -0.6857277000000001, 0.112787, 0.7515988, 0.4321929, 1.1508561, 1.3904105, 1.7098164,
0.9911532, -0.1267674, -1.0849851, 1.7896679, 1.3904105, 0.1926384, -1.1648365, 0.9113017, -0.2066188, 1.5501135,
-0.1267674, 1.2307076, -0.4461733, -1.4842424, 1.1508561, 0.4321929, 0.5918958, 0.3523414, 0.112787, 0.1926384,
0.112787, -0.6058762, 1.3904105, 0.3523414, 1.7098164, 0.3523414, 0.5918958, -0.0469159, 0.0329355, 1.8695194,
0.5918958, 0.9113017, 0.112787, 1.94937, -0.206618, -0.126767, -0.28647, -0.206618, 2.109073, -1.40439,
1.150856, -0.046915, -0.366321, 0.5918949999999999, 0.432192, 0.5120440000000001, 0.5120440000000001, -0.206618, -
1.084985, -0.605876,
-1.244688, -0.28647, -0.765579, 1.071004, -0.446173, -1.084985, 1.230707, 0.83145, -1.40439, -0.526024,
-0.446173, 0.032935, 0.751598, -1.244688, 0.751598, 0.83145, 0.112787, -0.046915, 0.032935, 0.352341,
-0.366321, -0.526024, 0.352341, 0.5918949999999999, -1.643945, -0.446173, -1.883499, -1.084985, -0.765579, -0.446173,
-0.92528, -0.84543, -0.605876, 1.789667, -0.925282, -0.84543, -0.84543, -0.605876, -1.643945, -0.126767,
-0.765579, -1.324539, -0.126767, 1.550113, -1.723796, -0.925282, -0.366321, -0.925282, -1.40439,
-1.40439, 0.83145, -1.883499, 0.5120440000000001, 0.352341, -0.28647, -1.723796, -0.685727, -1.963351, -1.164836,
-0.366321, -0.366321, 1.230707, -0.046915, -0.28647, 0.352341, -0.84543, -1.883499, -0.366321, -1.883499,
-1.164836, 0.272489, 0.192638, 0.192638, -1.244688, -1.244688, 0.192638, 0.5120440000000001, -1.8834998, -0.8454306,
-0.5260247, -1.0051336, 0.9911532, -0.2864703, -1.5640939, 0.7515988, -0.5260247, 0.9911532, -0.7655792, 0.8314502,
-1.8036483, -2.0432027, 0.9911532, -0.7655792, -1.0051336, -1.0051336, 0.0329355, -1.8834998, -0.2864703, -1.9633513,
1.0710047, -1.7237968, -0.9252821, 0.4321929, 2.2687768, -0.1267674, 1.1508561, 0.3523414, 1.7896679, -0.6857277000000001,
-1.1648365, -0.5260247, -0.0469159, 1.2307076, 2.1090738, -0.2066188, -0.4461733, 0.112787, -1.3245395, 1.8695194,
0.4321929, 2.1889253, 0.6717473, -1.9633513, -0.36632, 0.671747, 0.192638, -0.685727, -0.84543, 0.751598,

-0.685727, -1.40439, 0.83145, -0.765579, -0.526024, -0.925282, 1.310559, 0.991153, 0.83145, -1.084985,
-0.84543, -0.28647, 0.352341, -0.046915, 0.112787, -0.126767, -0.446173, 1.071004, -0.28647, -1.164836,
-0.84543, 0.192638, 0.272489, -0.046915, -0.206618, -1.643945, 0.671747, -1.963351, -2.123054, -2.123054,
-0.605876, -1.564093, 0.751598, -1.564093, -0.206618, -1.244688, -1.084985, 0.352341, 0.032935, 1.150856,
-0.126767, -0.84543, -1.084985, -0.765579, 0.5120440000000001, -0.126767, -0.84543, -0.28647, -0.366321, -0.685727,
-0.685727, -1.564093, 0.432192, -0.28647, 0.192638, -0.526024, -0.765579, -0.526024, 0.032935, 0.83145,
-0.046915, -0.605876, -0.925282, 0.911301, -0.685727, 0.5120440000000001, -1.084985, -0.206618, -0.685727, -2.123054,
-0.366321, 0.5918949999999999, -0.366321, -0.925282, -1.564093, -0.526024, -0.605876, -1.40439, 0.192638, -1.164836,
-1.164836, -1.963351, 0.272489, -0.605876, 0.352341, 1.310559, -0.046915, 0.192638, 0.5918949999999999, 0.911301, -0.446173,
-0.925282, -0.126767, 0.991153, -0.925282, -1.803648, -0.685727, -1.164836, 0.5918949999999999, 0.911301, -0.446173,
-0.446173, 1.310559, 0.27249, 0.192638, -1.324539, 0.27249, -0.605876, 1.470262, -0.685727, 1.470262,
-1.404391, 0.352341, 1.071004, -1.164836, -0.366321, -1.164836, -1.164836, -0.366321, -0.366321, -1.164836,
0.83145, -0.126767, -0.366321, -0.446173, -1.005133, -1.164836, -0.206618, -0.446173, -1.244688, -0.84543,
0.27249, -0.446173, -1.484242, 0.112787, -0.366321, 0.67174, -0.366321, 0.991153, 0.352341, -1.723796,
-0.526024, 0.671747, -0.28647, 0.5918949999999999, -1.40439, 1.94937, -0.446173, -1.005133, 0.5918949999999999, 0.911301,
-2.043202, -0.765579, -1.244688, -1.803648, -0.126767, -0.765579, -0.765579, -1.324539, 1.629965, 0.112787,
0.432192, 0.751598, -1.883499, 0.032935, -0.685727, -1.564093, 0.192638, 0.5120440000000001, 0.5918949999999999, -
1.005133,
-0.84543, 0.991153, -0.126767, 0.991153, -1.005133, -1.40439, -1.005133, 0.272489, -1.005133, 0.911301,
0.032935, -0.84543, 0.03293, -1.084985, 0.192638, 0.751598, -0.685727, 1.709816, -0.366321, -1.084985,
-1.40439, -0.605876, -1.244688, -1.324539, 0.352341, -0.206618, 0.991153, -1.005133, -0.925282, 0.352341,
-0.84543, -0.366321, -1.005133, -0.605876, 0.112787, -1.643945, -1.164836, -0.366321, -1.244688, -0.605876,
-0.765579, -1.324539, -1.084985, -1.084985, -0.526024, -1.244688, 0.032935, 0.192638, -0.366321, -1.40439,
-1.40439, -1.723796, -1.244688, 0.192638, -0.28647, -1.244688, -0.206618, -1.40439, -1.484242, -1.08498,
-0.28647, -1.164836, -1.244688, -1.484242, -1.244688, -0.84543, -0.206618, -0.366321, -0.765579, -1.40439,
0.112787, -0.046915, -1.963351, -0.526024, -0.605876, 0.671747, -0.925282, -1.164836, -1.005133, 0.192638,
-1.005133, -1.484242, -0.526024, -1.005133, 0.751598, -0.605876, 1.709816, 0.83145, -0.046915, -0.046915,
1.230707, 0.671747, 0.192638, -0.366321, 1.150856, -0.925282, 1.709816, -0.765579, -1.084985, -0.28647,
0.432192, 1.550113, 0.192638, 1.550113, -1.164836, 0.5120440000000001, -0.20661, -0.206618, -0.605876, 0.352341,
1.150856, 0.432192, 0.83145, -0.765579, -0.206618, 0.432192, 0.5120440000000001, -0.206618, 0.112787, 1.470262,
0.911301, 0.83145, 0.032935, 1.071004, 0.432192, 0.112787, 0.5120440000000001, -0.925282, -0.765579, 0.83145,
0.5918949999999999, 0.272489, 0.751598, -0.84543, -0.206618, 1.230707, 1.709816, 0.032935, 1.230707, 0.911301,
-0.28647, -1.164836, 1.150856, -1.244688, 0.5918949999999999, -0.526024, -0.605876, 0.112787, 2.029222, -0.526024,
-0.605876, 0.272489, -1.244688, 0.19263, -0.925282, 0.5918949999999999, -0.925282, -1.564093, 0.272489, 0.5918949999999999,
0.911301, 0.112787, 0.032935, 0.272489, 0.5120440000000001, 1.071004, 0.272489, -0.28647, 1.39041, -1.244688,
2.508331, 1.230707, -1.564093, -0.046915, -1.244688, 0.671747, -1.643945, 2.188925, -1.564093, -0.366321,
1.150856, 2.827737, 0.911301, -1.164836, 0.911301, 1.071004, 2.188925, -0.126767, 1.310559, 1.310559,
1.230707, 0.352341, -1.244688, -1.164836, -1.244688, 1.629965, 1.470262, -0.28647, -1.084985, 1.150856,
-0.52602, 2.98744, 3.5464, 2.188925, 0.272489, 0.751598, -0.925282, 0.272489, -1.084985, -1.005133,
0.432192, -0.28647, 0.991153, 0.991153, 0.352341, -0.925282, -0.366321, -0.685727, 0.671747, 0.112787,
-1.005133, -0.446173, -1.005133, -0.206618, 0.272489, -0.765579, 1.39041, 0.352341, -0.366321, -1.40439,
1.789667, 0.671747, -0.206618, 0.991153, 0.272489, 0.5918949999999999, -0.046915, -1.643945, -0.28647, -1.164836,
-0.765579, -1.005133, -0.765579, 0.432192, -0.925282, -0.206618, -0.84543, -0.68572, 0.032935, -0.685727,
0.192638, 0.991153, 0.112787, -0.925282, -0.28647, -0.526024, -0.84543, 0.352341, -1.324539, -1.084985,
0.112787, -0.366321, 1.310559, 1.310559, 0.5918949999999999, -0.206618, -1.484242, -1.324539, 1.550113, 0.272489,
-2.123054, 0.751598, 0.5120440000000001, -1.643945, 1.869519, -0.28647, -1.643945, -1.005133, -0.046915, -1.723796,
0.352341, 0.911301, -0.605876, -0.28647, 0.432192, -0.605876, 0.112787, -0.206618, -0.925282, -0.046915,
1.230707, 0.991153, -1.484242, 0.911301, -1.64394, 0.751598, 0.671747, 0.192638, -1.244688, -1.244688,
-0.526024, -0.28647, 0.112787, 0.991153, 0.432192, 1.150856, -0.84543, -0.765579, 0.112787, 0.991153,
0.192638, -0.28647, 1.150856, 1.310559, 0.911301, -0.526024, -0.28647, 0.5918949999999999, -0.126767, -0.366321,
-1.244688, -0.28647, 0.83145, 0.5918949999999999, -0.685727, -0.126767, -1.244688, -1.084985, 0.5918949999999999, 2.268776,
-1.324539, 1.39041, 0.032935, 0.911301, -0.685727, 0.192638, 0.991153, -0.925282, 0.5120440000000001, -0.206618,
1.789667, 1.31055, 1.709816, 1.789667, -0.28647, 0.352341, -0.84543, 2.188925, -0.84543, 0.032935,
0.432192, -0.765579, 0.751598, 0.112787, 0.911301, 1.39041, -0.84543, -0.28647, -0.765579, 0.432192,
0.671747, 0.192638, -0.28647, -2.043202, -1.883499, 0.83145, -1.643945, 1.789667, -0.605876, 1.230707,
-2.123054, 0.352341, -1.324539, -1.484242, 1.550113, 1.709816, 1.789667, -1.484242, -1.40439, 1.709816,
-0.28647, -1.164836, -0.685727, 0.671747, 0.671747, -0.366321, -0.366321, 0.5120440000000001, 0.112787, 1.071004,
0.751598, 1.629965, -0.046915, 0.991153, 0.5918949999999999, 0.911301, 0.83145, 0.5918949999999999, -0.28647, 0.192638,
-1.005133, -0.28647, -0.526024, 1.150856, 1.230707, -0.685727, -0.526024, -0.126767, 1.150856, -0.28647,
-0.685727, -1.005133, -1.564093, 1.150856, -1.084985, 0.272489, -0.685727, 1.230707, 1.629965, 0.5120440000000001,
0.83145, 0.432192, 0.272489, -0.206618, -0.206618, -1.564093, 1.310559, -1.564093, 0.112787, -1.564093,
1.789667, -0.84543, -0.925282, 1.39041, -0.366321, -0.84543, -0.765579, -0.765579, 0.5918949999999999, -1.723796,
-1.723796, -1.643945, 0.5120440000000001, -1.084985, 0.352341, 0.911301, 0.83145, -1.484242, -0.446173, -1.084985,
-1.803648, 0.5120440000000001, 0.5120440000000001, -0.685727, -0.765579, -0.046915, -0.526024, -1.484242, -1.164836,
2.268776,
1.071004, 1.150856, 0.192638, -0.925282, 0.911301, 0.751598, 0.5918949999999999, -0.28647, 0.991153, 0.112787,
-0.446173, 0.5120440000000001, 0.751598, 0.192638, 1.94937, -0.605876, 0.272489, 0.192638, 2.268776, 0.83145,
0.751598, -0.126767, -0.20661, -0.28647, -0.126767, -1.324539, -0.84543, 0.032935, 0.671747, -1.643945,

```

0.432192, 0.192638, -0.925282, -0.765579, 0.751598, 0.112787, -0.685727, -0.366321, 0.272489, -0.126767,
-1.803648, -1.484242, -1.164836, -0.765579, 0.112787, -0.765579, -0.685727, -0.765579, -1.244688, -0.925282,
0.112787, -1.643945, -1.40439, -0.28647, 0.192638, -0.046915, 0.5918949999999999, -1.564093, 1.230707, 0.911301,
-0.925282, -0.28647, 0.911301, -0.28647, 1.150856, -0.046915, 0.432192, -0.126767, 1.869519, -0.92528,
-0.925282, -0.84543, 0.5918949999999999, -1.244688, -0.925282, 2.188925, 0.751598, -1.723796, 1.629965, 0.991153,
0.112787, -0.84543, -1.723796, -0.605876, 0.112787, -1.084985, -1.244688, -1.244688, 0.032935, 0.432192,
-1.643945, -1.244688, 1.39041, -0.925282, -1.40439, 0.352341, 1.150856, -0.366321, 0.192638, -1.164836,
-0.366321, -1.244688, -1.084985, 1.550113, -0.685727, 0.911301, 1.230707, 0.192638, -0.925282, -1.40439,
0.751598, -0.366321, 0.751598, -0.206618, 0.5918949999999999, 0.751598, -0.36632, -0.126767, -0.685727, -0.446173,
-0.046915, -0.925282, 0.032935, -1.244688, 0.112787, -0.126767, -0.126767, 0.432192, 1.94937, 0.83145,
0.112787, -1.005133, 2.109073, -0.84543, 1.629965, -0.446173, 1.39041, 0.751598, -0.28647, 0.5120440000000001,
-1.005133, -0.28647, 0.432192, -0.685727, 0.272489, -0.446173, 1.470262, 0.751598, 0.5120440000000001, -0.526024,
1.230707, -1.005133, 1.550113, 0.911301, -0.366321, -1.084985, 0.352341, 0.751598, 0.432192, -1.963351,
1.071004, 1.550113, 0.991153, -1.64394, -0.765579, 1.550113, 1.071004, 0.5120440000000001, 1.869519, -1.164836,
0.991153, 0.112787, -0.84543, 0.192638, -0.526024, -1.164836, -2.043202, 1.869519, 0.032935, -1.40439,
0.192638, 0.272489, -0.765579, -1.084985, -0.126767, 1.150856, 1.629965, -0.366321, 1.150856, -0.925282,
0.671747, -1.963351, -1.803648, 0.83145, 0.112787, -1.803648, -0.446173, 0.991153, 1.230707, 0.991153,
-0.206618, 0.671747, -0.605876, 1.550113, -0.28647, -1.963351, 1.230707, 1.39041, -0.526024, -1.244688,
-0.36632, 1.310559, -1.963351, 1.629965, -1.324539, -1.803648, -0.28647, 1.869519, 1.470262, 1.550113,
-0.446173, -1.963351, 1.470262, 0.5120440000000001, -1.883499, -1.883499, -0.605876, -0.366321, -1.084985, -0.366321,
1.230707, 0.5120440000000001, 1.071004, -0.765579, -1.244688, -0.526024, -1.244688, -0.605876, -1.40439, -0.526024,
0.671747, 1.94937, 0.5120440000000001, 0.352341, -1.244688, -0.366321, 1.550113, -1.564093, -0.366321, 0.911301,
2.348628, 1.629965, 0.671747, 0.991153, -0.046915, 0.192638, -0.366321, -0.28647, 1.310559, -0.925282,
1.310559, 1.310559, 1.629965, 0.991153, -0.685727, -0.366321, -0.605876, 0.192638, 1.789667, -1.164836,
0.032935, 1.789667, 0.112787, 0.272489, 1.39041, -1.324539, -0.84543, 0.5918949999999999, -1.084985, 0.751598,
0.5120440000000001, -0.206618, 0.991153, 1.071004, -0.526024, 0.5120440000000001, 0.5918949999999999, 0.991153, -
0.366321, -0.843612,
-1.560598, -0.285956, 2.183663, 1.148016, 0.112369, 0.59036, -1.799594, 0.59036, -0.6842819999999999, -0.046961,
-1.799594, 0.829355, -0.6842819999999999, 1.227681, 2.422658, 0.431029, -1.560598, 0.032704
)
)⇨

```

Note that school is a 1978 x 3 matrix taking value 1 for all pupils in school 1, 2 for all pupils in school 2 and so on. For computational convenience, Y, mu and tau are indexed over a single dimension p = 1,...,1978 rather than as pupil i within school j as used in equations above. The appropriate school-level coefficients for pupil p are then selected using the school indicator in row p of the data array --- for example

`alpha[school[p],1].`

Inits

) ←

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

a) Without over-relaxation

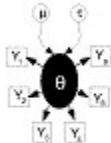
node	mean	sd	MC error	2.5%	median	97.5%
beta[1]	2.597E-4	9.65E-5	2.435E-6	7.334E-5	2.591E-4	4.497E-4
beta[2]	0.4099	0.06203	0.00341	0.2901	0.4093	0.53
beta[3]	0.17	0.04802	0.001491	0.07532	0.1698	0.2651
beta[4]	0.1147	0.1319	0.005749	-0.1406	0.112	0.3776
beta[5]	0.04825	0.1048	0.005128	-0.1645	0.04993	0.2435
beta[6]	-0.2871	0.1833	0.007287	-0.6438	-0.2852	0.07716
beta[7]	0.1477	0.1041	0.003529	-0.05753	0.1489	0.352
beta[8]	-0.1544	0.1885	0.007909	-0.5197	-0.1577	0.2302
gamma[1]	-0.6601	0.0982	0.006072	-0.8602	-0.6577	-0.4751
gamma[2]	0.03146	0.01012	1.33E-4	0.0113	0.03149	0.05154
gamma[3]	0.9447	0.08546	0.004669	0.7791	0.9433	1.111
phi	-0.002735	0.002834	3.034E-5	-0.008224	-0.002751	0.002699
theta	0.5804	0.03255	2.965E-4	0.5161	0.5807	0.6438

b) With over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
beta[1]	2.607E-4	9.624E-5	1.156E-6	7.034E-5	2.608E-4	4.495E-4
beta[2]	0.4156	0.05876	0.001563	0.2993	0.4156	0.5306
beta[3]	0.1701	0.04751	6.995E-4	0.07717	0.1705	0.2622
beta[4]	0.1173	0.1338	0.002555	-0.1455	0.1169	0.3794
beta[5]	0.05498	0.1036	0.002158	-0.143	0.05563	0.2628
beta[6]	-0.2895	0.1808	0.00313	-0.6516	-0.2876	0.06532
beta[7]	0.1443	0.1064	0.001706	-0.06485	0.1453	0.352
beta[8]	-0.1658	0.1849	0.003272	-0.5382	-0.1651	0.194
gamma[1]	-0.666	0.09297	0.002867	-0.8432	-0.6661	-0.4805
gamma[2]	0.03145	0.01023	5.744E-5	0.01158	0.03155	0.05129
gamma[3]	0.9519	0.08177	0.002135	0.7916	0.9517	1.111
phi	-0.002696	0.002861	1.616E-5	-0.00828	-0.002701	0.002937
theta	0.5803	0.03274	1.472E-4	0.5163	0.5806	0.6444

Estimating the ranks

The school-specific intercept α_{j1} measures the 'residual effect' for school j after adjusting for pupil- and school-level covariates. This might represent an appropriate quantity by which to rank schools' performance. We compute the ranks in BUGS using the "rank" option of the "Statistics" menu, which we set for the variable alpha at the same time as we set the "sample monitor" option. Since the rank is a function of stochastic nodes, its value will change at every iteration. Hence we may obtain a posterior distribution for the rank of $\alpha_{[k]}$ which may be summarized by posterior histograms as shown below:



BUGS Ice: non-parametric smoothing in an age-cohort model

Breslow and Clayton (1993) analyse breast cancer rates in Iceland by year of birth ($K = 11$ cohorts from 1840-1849 to 1940-1949) and by age ($J = 13$ groups from 20-24 to 80-84 years). Due to the number of empty cells we consider a single indexing over $I=77$ observed number of cases, giving data of the following form.

i	age _i	year _i	cases _i	person-years _i
1	1	6	2	41380
2	1	7	0	43650
...	...			
77	13	5	31	13600

In order to pull in the extreme risks associated with small birth cohorts, Breslow and Clayton first consider the exchangeable model

$$\begin{aligned} \text{cases}_i &\sim \text{Poisson}(\mu_i) \\ \log \mu_i &= \log \text{person-years}_i + \alpha_{\text{age}_i} + \beta_{\text{year}_i} \\ \beta_k &\sim \text{Normal}(0, \tau) \end{aligned}$$

Autoregressive smoothing of relative risks

They then consider the alternative approach of smoothing the rates for the cohorts by assuming an auto-regressive model on the β 's, assuming the second differences are independent normal variates. This is equivalent to a model and prior distribution

$$\begin{aligned} \text{cases}_i &\sim \text{Poisson}(\mu_i) \\ \log \mu_i &= \log \text{person-years}_i + \alpha_{\text{age}_i} + \beta_{\text{year}_i} \\ \beta_1 &\sim \text{Normal}(0, 0.000001\tau) \\ \beta_2 | \beta_1 &\sim \text{Normal}(0, 0.000001\tau) \\ \beta_k | \beta_1, \dots, \beta_{k-1} &\sim \text{Normal}(2\beta_{k-1} - \beta_{k-2}, \tau) \quad k > 2 \end{aligned}$$

We note that β_1 and β_2 are given "non-informative" priors, but retain a τ term in order to provide the appropriate likelihood for τ .

For computational reasons Breslow and Clayton impose constraints on their random effects β_k in order that their mean and linear trend are zero, and counter these constraints by introducing a linear term $b \times \text{year}_i$ and allowing unrestrained estimation of α_j . Since we allow free movement of the β 's we dispense with the linear term, and impose a "corner" constraint $\alpha_1 = 0$.

```
model
{
  for (i in 1:I) {
    cases[i] ~ dpois(mu[i])
    log(mu[i]) <- log(pyr[i]) + alpha[age[i]] + beta[year[i]]
```

```

}
betamean[1] <- 2 * beta[2] - beta[3]
Nneighs[1] <- 1
betamean[2] <- (2 * beta[1] + 4 * beta[3] - beta[4]) / 5
Nneighs[2] <- 5
for (k in 3 : K - 2) {
    betamean[k] <- (4 * beta[k - 1] + 4 * beta[k + 1] - beta[k - 2] - beta[k + 2]) / 6
    Nneighs[k] <- 6
}
betamean[K - 1] <- (2 * beta[K] + 4 * beta[K - 2] - beta[K - 3]) / 5
Nneighs[K - 1] <- 5
betamean[K] <- 2 * beta[K - 1] - beta[K - 2]
Nneighs[K] <- 1
for (k in 1 : K) {
    betaprec[k] <- Nneighs[k] * tau
}
for (k in 1 : K) {
    beta[k] ~ dnorm(betamean[k], betaprec[k])
    logRR[k] <- beta[k] - beta[5]
    tau.like[k] <- Nneighs[k] * beta[k] * (beta[k] - betamean[k])
}
alpha[1] <- 0.0
for (j in 2 : Nage) {
    alpha[j] ~ dnorm(0, 1.0E-6)
}
d <- 0.0001 + sum(tau.like[])
r <- 0.0001 + K / 2
tau ~ dgamma(r, d)
sigma <- 1 / sqrt(tau)
}

```

Data

```

⇒list(age = c(1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 6, 6, 6,
6, 6, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 10, 10, 11, 11, 11, 11,
11, 11, 12, 12, 12, 12, 12, 13, 13, 13, 13, 13),
year = c(6, 7, 8, 9, 10, 11, 6, 7, 8, 9, 10, 11, 5, 6, 7, 8, 9, 10, 10, 5, 6, 7, 8, 9, 10, 4, 5, 6, 7, 8, 9, 4, 5, 6, 7,
8, 9, 3, 4, 5, 6, 7, 8, 3, 4, 5, 6, 7, 8, 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4,
5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5),
cases = c(2, 0, 1, 1, 1, 2, 0, 2, 1, 1, 5, 5, 1, 1, 3, 7, 12, 10, 6, 11, 9, 14, 20, 14, 7, 14, 22, 25, 29, 37, 21, 11, 29, 33,
57, 24, 15, 8, 22, 27, 38, 52, 10, 15, 22, 26, 47, 31, 8, 11, 17, 23, 31, 38, 8, 10, 24, 30, 53, 26, 5, 3, 10,
18,
22, 30, 1, 7, 11, 26, 32, 17, 5, 8, 17, 32, 31),
pyr = c(41380, 43650, 49810, 58105, 57105, 76380, 39615, 42205, 48315, 56785, 55965, 33955, 29150, 38460,
40810, 47490, 55720, 55145, 27950, 37375, 39935, 46895, 54980, 27810, 25055, 27040, 36400,
39355,
46280, 54350, 24040, 26290, 35480, 38725, 45595, 25710, 22890, 23095, 25410, 34420, 37725, 44740,
21415, 21870, 24240, 33175, 36345, 21320, 17450, 19765, 20255, 22760, 31695, 34705, 15350, 17720,
18280, 20850, 29600, 15635, 9965, 12850, 15015, 15725, 18345, 26400, 8175, 11020, 13095, 14050,
16480, 10885, 7425, 10810, 12260, 14780, 13600),
I = 77, Nage = 13, K = 11)⇒

```

Init

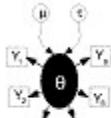
```
⇒list(tau=1,
```

```
alpha=c(NA,0,0,0,0,0,0,0,0,0,0,0),
beta =c(0.05,0.1,0,0,0,0,0,0,0,0,0))
```

Results

Time for 10000 updates 50s on 200MHz Pentium Pro. A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
logRR[1]	-1.069	0.2481	0.01745	-1.584	-1.051	-0.6877	1001	10000
logRR[2]	-0.7665	0.1575	0.01126	-1.092	-0.7602	-0.518	1001	10000
logRR[3]	-0.4678	0.08208	0.005319	-0.6461	-0.462	-0.3364	1001	10000
logRR[4]	-0.1986	0.03919	0.001245	-0.276	-0.1978	-0.1121	1001	10000
logRR[5]	0.0	1.0E-10	1.0E-12	0.0	0.0	0.0	1001	10000
logRR[6]	0.1557	0.04337	0.00183	0.04866	0.1651	0.2187	1001	10000
logRR[7]	0.3086	0.0652	0.003348	0.1578	0.3182	0.4103	1001	10000
logRR[8]	0.4639	0.08109	0.004742	0.2907	0.469	0.6001	1001	10000
logRR[9]	0.6128	0.106	0.006742	0.3947	0.6176	0.7956	1001	10000
logRR[10]	0.7857	0.1384	0.008817	0.4939	0.7954	1.041	1001	10000
logRR[11]	0.9663	0.2015	0.01177	0.553	0.9799	1.383	1001	10000
sigma	0.05392	0.04312	0.002532	0.007323	0.04302	0.1657	1001	10000



Beetles: choice of link function

Dobson (1983) analyses binary dose-response data published by Bliss (1935), in which the numbers of beetles killed after 5 hour exposure to carbon disulphide at N=8 different concentrations are recorded:

Concentration (x_i)	Number of beetles (n_i)	Number killed (r_i)
1.6907	59	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	52
1.8610	62	61
1.8839	60	60

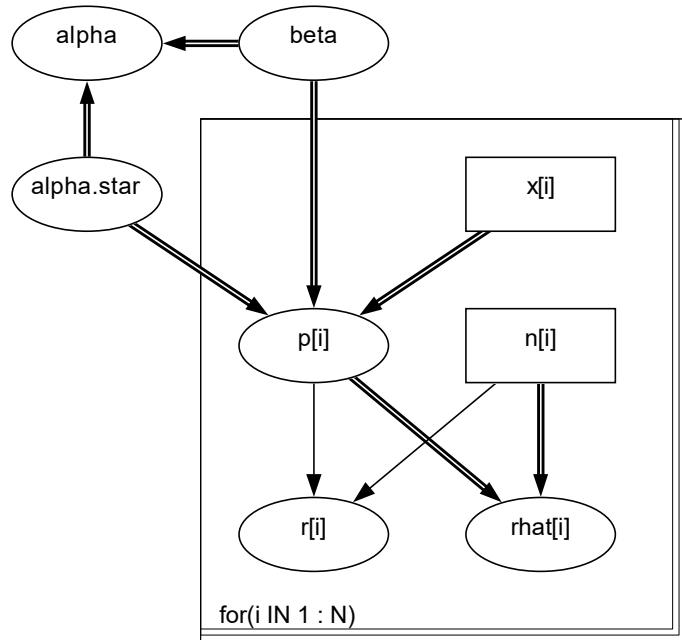
We assume that the observed number of deaths r_i at each concentration x_i is binomial with sample size n_i and true rate p_i . Plausible models for p_i include the logistic, probit and extreme value (complimentary log-log) models, as follows

$$p_i = \exp(\alpha + \beta x_i) / (1 + \exp(\alpha + \beta x_i))$$

$$p_i = \text{Phi}(\alpha + \beta x_i)$$

$$p_i = 1 - \exp(-\exp(\alpha + \beta x_i))$$

The corresponding graph is shown below:



```

model
{
  for( i in 1 : N ) {
    r[i] ~ dbin(p[i],n[i])
    logit(p[i]) <- alpha.star + beta * (x[i] - mean(x[]))
    rhat[i] <- n[i] * p[i]
  }
  alpha <- alpha.star - beta * mean(x[])
  beta ~ dnorm(0.0,0.001)
  alpha.star ~ dnorm(0.0,0.001)
}

```

Data

```

list( x = c(1.6907, 1.7242, 1.7552, 1.7842, 1.8113, 1.8369, 1.8610, 1.8839),
      n = c(59, 60, 62, 56, 63, 59, 62, 60),
      r = c(6, 13, 18, 28, 52, 53, 61, 60), N = 8)

```

Inits

```
list(alpha.star=0, beta=0)
```

Results

Time for 10000 updates was 7s on 200MHz Pentium Pro.A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

Logit model

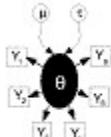
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	-60.8	5.177	0.05891	-71.38	-60.66	-51.11	1001	10000
beta	34.32	2.91	0.03336	28.89	34.24	40.27	1001	10000
rhat[1]	3.57	0.9597	0.00952	1.991	3.48	5.678	1001	10000
rhat[2]	9.957	1.7	0.01629	6.876	9.879	13.47	1001	10000
rhat[3]	22.51	2.126	0.02071	18.36	22.53	26.71	1001	10000
rhat[4]	33.91	1.777	0.02053	30.38	33.94	37.31	1001	10000
rhat[5]	50.06	1.659	0.02113	46.73	50.09	53.15	1001	10000
rhat[6]	53.22	1.105	0.01409	50.94	53.27	55.22	1001	10000
rhat[7]	59.15	0.7336	0.009167	57.57	59.21	60.4	1001	10000
rhat[8]	58.69	0.4233	0.005176	57.74	58.74	59.36	1001	10000

Probit model

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	-17.58	1.346	0.01519	-20.26	-17.55	-15.04	1001	10000
beta	9.928	0.757	0.008605	8.503	9.912	11.44	1001	10000
rhat[1]	3.405	1.021	0.0102	1.741	3.304	5.662	1001	10000
rhat[2]	10.71	1.709	0.01563	7.578	10.64	14.25	1001	10000
rhat[3]	23.45	1.925	0.01687	19.75	23.43	27.21	1001	10000
rhat[4]	33.82	1.622	0.01652	30.65	33.83	37.01	1001	10000
rhat[5]	49.64	1.644	0.01907	46.35	49.68	52.75	1001	10000
rhat[6]	53.31	1.168	0.01406	50.88	53.37	55.42	1001	10000
rhat[7]	59.62	0.7502	0.009075	57.96	59.7	60.85	1001	10000
rhat[8]	59.19	0.3685	0.004436	58.32	59.25	59.72	1001	10000

Extreme value (cloglog) model

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	-39.72	3.259	0.03582	-46.18	-39.64	-33.43	1001	10000
beta	22.12	1.809	0.01987	18.64	22.08	25.71	1001	10000
rhat[1]	5.652	1.144	0.01237	3.681	5.575	8.163	1001	10000
rhat[2]	11.32	1.613	0.01721	8.396	11.26	14.7	1001	10000
rhat[3]	20.96	1.929	0.01985	17.31	20.94	24.84	1001	10000
rhat[4]	30.36	1.705	0.01618	27.07	30.35	33.74	1001	10000
rhat[5]	47.77	1.777	0.01612	44.27	47.81	51.2	1001	10000
rhat[6]	54.08	1.255	0.01197	51.47	54.17	56.26	1001	10000
rhat[7]	61.02	0.546	0.005398	59.71	61.12	61.76	1001	10000
rhat[8]	59.92	0.1018	0.001028	59.64	59.95	60.0	1001	10000



BUGS Alligators: multinomial - logistic regression

Agresti (1990) analyses a set of data on the feeding choice of 221 alligators, where the response measure for each alligator is one of 5 categories: fish, invertebrate, reptile, bird, other. Possible explanatory factors are the length of alligator (two categories: ≤ 2.3 metres and > 2.3 metres), and the lake (4 categories: Hancock, Oklawaha, Trafford, George). The full data is shown below.

Lake	Size	Primary Food Choice				
		Fish	Invertebrate	Reptile	Bird	Other
Hancock	≤ 2.3	23	4	2	2	8
	> 2.3	7	0	1	3	5
Oklawaha	≤ 2.3	5	11	1	0	3
	> 2.3	13	8	6	1	0
Trafford	≤ 2.3	5	11	2	1	5
	> 2.3	8	7	6	3	5
George	≤ 2.3	16	19	1	2	3
	> 2.3	17	1	0	1	3

Each combination of explanatory factors is assumed to give rise to a multinomial response with a logistic link, so that for lake i , size j , the observed vector of counts $X_{ij\cdot} = X_{ij1}, \dots, X_{ij5}$ has distribution

$$X_{ij\cdot} \sim \text{Multinomial}(p_{ij\cdot}, n_{ij})$$

$$p_{ijk} = \phi_{ijk} / \sum_k \phi_{ijk}$$

$$\phi_{ijk} = e^{\alpha_k + \beta_{ik} + \gamma_{jk}}$$

where $n_{ij} = \sum_k X_{ijk}$, and $\alpha_1, \beta_{i1}, \beta_{1k}, \gamma_{j1}, \gamma_{1k} = 0$ for identifiability. This model is discussed in detail in the Classic BUGS manual (version 0.5) in the section on *Multinomial Logistic Models*. All unknown α 's, β 's, γ 's are initially given independent "noninformative" priors.

The Classic BUGS manual (version 0.5) discusses two ways of fitting this model: directly in the form given above or by using the multinomial-Poisson transformation which will be somewhat more efficient. Both techniques are illustrated in the code given below.

```

model
{
# PRIORS
alpha[1] <- 0;    # zero contrast for baseline food
for (k in 2 : K) { alpha[k] ~ dnorm(0, 0.00001) } # vague priors
# Loop around lakes:
for (k in 1 : K){ beta[1, k] <- 0 } # corner-point contrast with first lake
for (i in 2 : l){
  beta[i, 1] <- 0 ; # zero contrast for baseline food
  for (k in 2 : K){ beta[i, k] ~ dnorm(0, 0.00001) } # vague priors
}

```

```

# Loop around sizes:
for (k in 1 : K){ gamma[1, k] <- 0} # corner-point contrast with first size
for (j in 2 : J) {
  gamma[j, 1] <- 0 ; # zero contrast for baseline food
  for ( k in 2 : K){ gamma[j, k] ~ dnorm(0, 0.00001)} # vague priors
}

# LIKELIHOOD
for (i in 1 : I){ # loop around lakes
  for (j in 1 : J) { # loop around sizes

# Multinomial response
#   X[i,j,1:K] ~ dmulti( p[i,j,1:K] , n[i,j] )
#   n[i,j] <- sum(X[i,j,])
#   for (k in 1:K) { # loop around foods
#     p[i,j,k]      <- phi[i,j,k] / sum(phi[i,j,])
#     log(phi[i,j,k]) <- alpha[k] + beta[i,k] + gamma[j,k]
#   }

# Fit standard Poisson regressions relative to baseline
  lambda[i, j] ~ dnorm(0, 0.00001) # vague priors
  for (k in 1 : K) { # loop around foods
    X[i, j, k] ~ dpois(mu[i, j, k])
    log(mu[i, j, k]) <- lambda[i, j] + alpha[k] + beta[i, k] + gamma[j, k]
  }
}

# TRANSFORM OUTPUT TO ENABLE COMPARISON
# WITH AGRESTI'S RESULTS

for (k in 1:K) { # loop around foods
  for (i in 1:I) { # loop around lakes
    b[i,k] <- beta[i,k] - mean(beta[,k]); # sum to zero constraint
  }
  for (j in 1:J) { # loop around sizes
    g[j,k] <- gamma[j,k] - mean(gamma[,k]); # sum to zero constraint
  }
}
}

```

Data

```

list( I = 4, J = 2, K = 5,
X = structure(.Data = c(23, 4, 2, 2, 8, 7, 0, 1, 3, 5, 5, 11, 1, 0, 3, 13, 8, 6, 1, 0,
5, 11, 2, 1, 5, 8, 7, 6, 3, 5, 16, 19, 1, 2, 3, 17, 1, 0, 1, 3), .Dim = c(4, 2, 5)))

```

Inits

```

list(alpha = c(NA, 0, 0, 0, 0),

```

```

beta = structure(.Data = c(NA, NA, NA, NA, NA,
                         NA, 0, 0, 0, 0,
                         NA, 0, 0, 0, 0,
                         NA, 0, 0, 0, 0), .Dim = c(4, 5)),
gamma = structure(.Data = c(NA, NA, NA, NA, NA,
                           NA, 0, 0, 0, 0), .Dim = c(2, 5)),
lambda = structure(.Data = c(0, 0, 0, 0,
                            0, 0, 0, 0), .Dim = c(4, 2))
)

```

Results

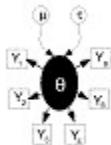
A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

a) Without over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
b[1,2]	-1.848	0.4383	0.02293	-2.731	-1.835	-1.027
b[1,3]	-0.3708	0.621	0.02243	-1.634	-0.3568	0.8245
b[1,4]	0.5511	0.5652	0.01348	-0.5269	0.5443	1.694
b[1,5]	0.2676	0.3589	0.009536	-0.4401	0.2678	0.9725
b[2,2]	0.8887	0.3354	0.009803	0.2404	0.8854	1.555
b[2,3]	0.9652	0.52	0.009951	-0.01176	0.9505	2.019
b[2,4]	-1.258	1.024	0.01253	-3.611	-1.119	0.3404
b[2,5]	-0.6495	0.5476	0.006817	-1.83	-0.6211	0.3395
b[3,2]	1.061	0.3424	0.01117	0.4008	1.056	1.75
b[3,3]	1.452	0.5171	0.01129	0.4805	1.431	2.522
b[3,4]	0.935	0.5997	0.008917	-0.2523	0.9276	2.121
b[3,5]	0.9879	0.3932	0.008232	0.2123	0.9894	1.776
b[4,2]	-0.1019	0.2885	0.008492	-0.6761	-0.1001	0.4526
b[4,3]	-2.046	1.005	0.01222	-4.469	-1.883	-0.4971
b[4,4]	-0.2282	0.6205	0.007964	-1.478	-0.2253	0.9928
b[4,5]	-0.606	0.4159	0.006381	-1.461	-0.592	0.1719

b) With over-relaxation

node	mean	sd	MC error	2.5%	median	97.5%
b[1,2]	-1.87	0.4603	0.01001	-2.889	-1.839	-1.024
b[1,3]	-0.3625	0.6262	0.008249	-1.594	-0.3546	0.8385
b[1,4]	0.559	0.564	0.004812	-0.5128	0.5474	1.704
b[1,5]	0.2789	0.3577	0.00332	-0.4257	0.2776	0.9785
b[2,2]	0.8878	0.3345	0.004201	0.2356	0.8869	1.551
b[2,3]	0.9535	0.5312	0.004329	-0.03954	0.9405	2.039
b[2,4]	-1.263	1.03	0.005909	-3.703	-1.134	0.3446
b[2,5]	-0.6623	0.5514	0.002845	-1.845	-0.6187	0.3122
b[3,2]	1.071	0.3432	0.004499	0.4037	1.065	1.761
b[3,3]	1.444	0.5236	0.004545	0.4579	1.428	2.524
b[3,4]	0.9336	0.5891	0.003824	-0.2111	0.9321	2.095
b[3,5]	0.9835	0.3962	0.002758	0.1984	0.9852	1.762
b[4,2]	-0.08908	0.2986	0.003159	-0.6707	-0.09152	0.5088
b[4,3]	-2.035	1.012	0.005156	-4.415	-1.89	-0.488
b[4,4]	-0.2297	0.6333	0.003405	-1.501	-0.2205	0.9989
b[4,5]	-0.6002	0.4079	0.00203	-1.437	-0.5893	0.1744



BUGS Endo: conditional inference in case-control studies

Breslow and Day (1980) analyse a set of data from a case-control study relating endometrial cancer with exposure to estrogens. 183 pairs of cases and controls were studied, and the full data is shown below.

Status of case	Status of control	
	Not exposed	Exposed
Not exposed	n00 = 121	n01 = 7
Exposed	n10 = 43	n11 = 12

We denote estrogen exposure as x_{ij} for the i th case-control pair, where $j=1$ for a case and $j=2$ for a control. The conditional likelihood for the log (odds ratio) β is then given by $\prod_i \exp \beta x_{i1} / (\exp \beta x_{i1} + \exp \beta x_{i2})$

We shall illustrate three methods of fitting this model. It is convenient to denote the fixed disease status as a variable $Y_{i1} = 1, Y_{i2} = 0$.

First, Breslow and Day point out that for case-control studies with a single control per case, we may obtain this likelihood by using unconditional logistic regression for each case-control pair. That is

$$\begin{aligned} Y_{i1} &\sim \text{Binomial}(p_i, 2) \\ \text{logit } p_i &= \beta (x_{i1} - x_{i2}) \end{aligned}$$

Second, the Classic BUGS manual (version 0.5) section on *Conditional likelihoods in case-control studies* discusses fitting this likelihood directly by assuming the model

$$\begin{aligned} Y_{i\cdot} &\sim \text{Multinomial}(p_{i\cdot}, 1) \\ p_{ij} &= e_{ij} / \sum_j e_{ij} \\ \text{log } e_{ij} &= \beta x_{ij} \end{aligned}$$

Finally, the Classic BUGS manual (version 0.5) shows how the multinomial-Poisson transformation can be used. In general, this will be more efficient than using the multinomial-logistic parameterisation above, since it avoids the time-consuming evaluation of $\sum_j e_{ij}$. However, in the present example this summation is only over $J=2$ elements, whilst the multinomial-Poisson parameterisation involves estimation of an additional intercept parameter for each of the 183 strata. Consequently the latter is less efficient than the multinomial-logistic in this case.

We note that all these formulations may be easily extended to include additional subject-specific covariates, and that the second and third methods can handle arbitrary numbers of controls per case. In addition, the Bayesian approach allows the incorporation of hierarchical structure, measurement error, missing data and so on.

All these techniques are illustrated in the code given below, which includes a transformation of the original summary statistics into full data. In this example,

all but the second conditional-likelihood approach are commented out.

```
model
{
# transform collapsed data into full
  for (i in 1:l){ Y[i,1] <- 1 Y[i,2] <- 0}
# loop around strata with case exposed, control not exposed (n10)
  for (i in 1:n10){ est[i,1] <- 1 est[i,2] <- 0}
# loop around strata with case not exposed, control exposed (n01)
  for (i in (n10+1):(n10+n01)){ est[i,1] <- 0 est[i,2] <- 1}
# loop around strata with case exposed, control exposed (n11)
  for (i in (n10+n01+1):(n10+n01+n11)){ est[i,1] <- 1 est[i,2] <- 1}
# loop around strata with case not exposed, control not exposed (n00)
  for (i in (n10+n01+n11+1):l){ est[i,1] <- 0 est[i,2] <- 0}

# PRIORS
  beta ~ dnorm(0,1.0E-6) ;

# LIKELIHOOD
  for (i in 1 : l){           # loop around strata
# METHOD 1 - logistic regression
  #   Y[i,1] ~ dbin( p[i,1], 1)
  #   logit(p[i,1]) <- beta * (est[i,1] - est[i,J])
# METHOD 2 - conditional likelihoods
  Y[i, 1 : J] ~ dmulti( p[i, 1 : J],1)
    for (j in 1:2){
      p[i, j] <- e[i, j] / sum(e[i, ])
      log( e[i, j] ) <- beta * est[i, j]
    }
# METHOD 3 fit standard Poisson regressions relative to baseline
  #   for (j in 1:J) {
  #     Y[i, j] ~ dpois(mu[i, j]);
  #     log(mu[i, j]) <- beta0[i] + beta*est[i, j];
  #   }
  #   beta0[i] ~ dnorm(0, 1.0E-6)
}

}
```

}

Data

```
list(n10=43, n01=7, n11=12, l = 183, J=2)
```

Inits

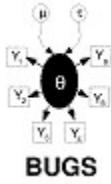
```
list(beta = 2)
```

Results

Time for 10000 updates 31s on 200MHz Pentium Pro. A 1000 update burn in followed by a further 10000

updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta	1.882	0.427	0.003892	1.104	1.858	2.754	1001	10000



Stagnant: a changepoint problem (and an illustration of how NOT to do MCMC!)

Carlin, Gelfand and Smith (1992) analyse data from Bacon and Watts (1971) concerning a changepoint in a linear regression.

i	x_i	Y_i	i	x_i	Y_i	i	x_i	Y_i
1	-1.39	1.12	11	-0.12	0.60	21	0.44	0.13
2	-1.39	1.12	12	-0.12	0.59	22	0.59	-0.01
3	-1.08	0.99	13	0.01	0.51	23	0.70	-0.13
4	-1.08	1.03	14	0.11	0.44	24	0.70	0.14
5	-0.94	0.92	15	0.11	0.43	25	0.85	-0.30
6	-0.80	0.90	16	0.11	0.43	26	0.85	-0.33
7	-0.63	0.81	17	0.25	0.33	27	0.99	-0.46
8	-0.63	0.83	18	0.25	0.30	28	0.99	-0.43
9	-0.25	0.65	19	0.34	0.25	29	1.19	-0.65
10	-0.25	0.67	20	0.34	0.24			

Note the repeated x's.

We assume a model with two straight lines that meet at a certain changepoint x_k --- this is slightly different from the model of Carlin, Gelfand and Smith (1992) who do not constrain the two straight lines to cross at the changepoint. We assume

$$Y_i \sim \text{Normal}(\mu_i, \tau) \\ \mu_i = \alpha + \beta_{J[i]} (x_i - x_k) \quad J[i]=1 \text{ if } i \leq k \quad J[i]=2 \text{ if } i > k$$

giving $E(Y) = \alpha$ at the changepoint, with gradient β_1 before, and gradient β_2 after the changepoint. We give independent "noninformative" priors to α , β_1 , β_2 and τ .

Note: alpha is $E(Y)$ at the changepoint, so will be highly correlated with k. This may be a very poor parameterisation.

Note way of constructing a uniform prior on the integer k, and making the regression parameter depend on a random changepoint.

```

model
{
  for( i in 1 : N ) {
    Y[i] ~ dnorm(mu[i],tau)
    mu[i] <- alpha + beta[J[i]] * (x[i] - x[k])
    J[i] <- 1 + step(i - k - 0.5)
    punif[i] <- 1/N
  }
  tau ~ dgamma(0.001,0.001)
  alpha ~ dnorm(0.0,1.0E-6)
  for( j in 1 : 2 ) {

```

```

        beta[j] ~ dnorm(0.0,1.0E-6)
    }
    k ~ dcat(punif[])
    sigma <- 1 / sqrt(tau)
}

```

Data

⇒

```

list(Y = c(1.12, 1.12, 0.99, 1.03, 0.92, 0.90, 0.81, 0.83, 0.65, 0.67, 0.60, 0.59, 0.51, 0.44, 0.43,
0.43, 0.33, 0.30, 0.25, 0.24, 0.13, -0.01, -0.13, -0.14, -0.30, -0.33, -0.46, -0.43, -0.65),
x = c(-1.39, -1.39, -1.08, -1.08, -0.94, -0.80, -0.63, -0.63, -0.25, -0.25, -0.12, -0.12, 0.01, 0.11, 0.11,
0.11, 0.25, 0.25, 0.34, 0.34, 0.44, 0.59, 0.70, 0.70, 0.85, 0.85, 0.99, 0.99, 1.19),
N = 29)⇒

```

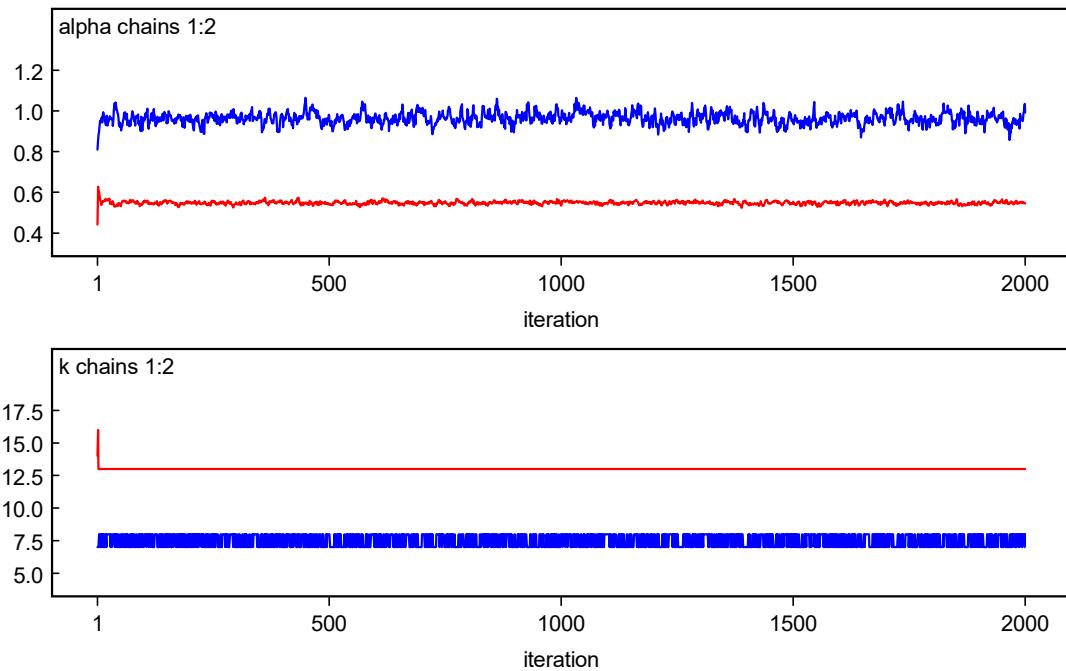
Inits

```

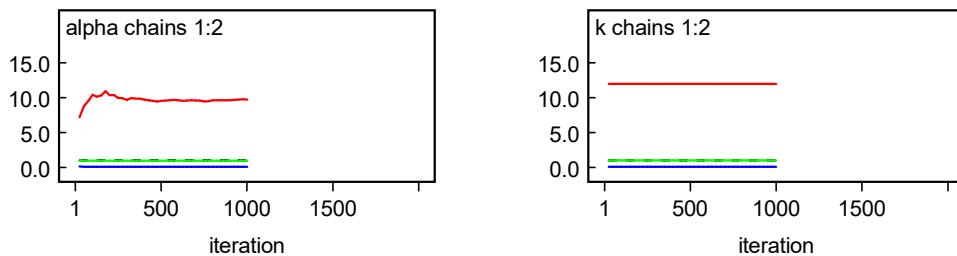
list(alpha = 0.2, beta = c(-0.45, -1.0), tau = 5, k = 16)
list(alpha = 0.6, beta = c(-0.45, -1.0), tau = 5, k = 8)

```

Traces of two chains shows complete dependence on starting values



Gelman-Rubin - red line should be near 1 !!



Results are hopeless - no mixing at all.

Note: alpha is E(Y) at the changepoint, so will be highly correlated with k. This may be a very poor parameterisation.

TRY USING CONTINUOUS PARAMETERISATION

```
model
{
  for(i in 1 : N) {
    Y[i] ~ dnorm(mu[i], tau)
    mu[i] <- alpha + beta[J[i]] * (x[i] - x.change)
    J[i] <- 1 + step(x[i] - x.change)
  }
  tau ~ dgamma(0.001, 0.001)
  alpha ~ dnorm(0.0, 1.0E-6)
  for(j in 1 : 2) {
    beta[j] ~ dnorm(0.0, 1.0E-6)
  }
  sigma <- 1 / sqrt(tau)
  x.change ~ dunif(-1.3, 1.1)
}
```

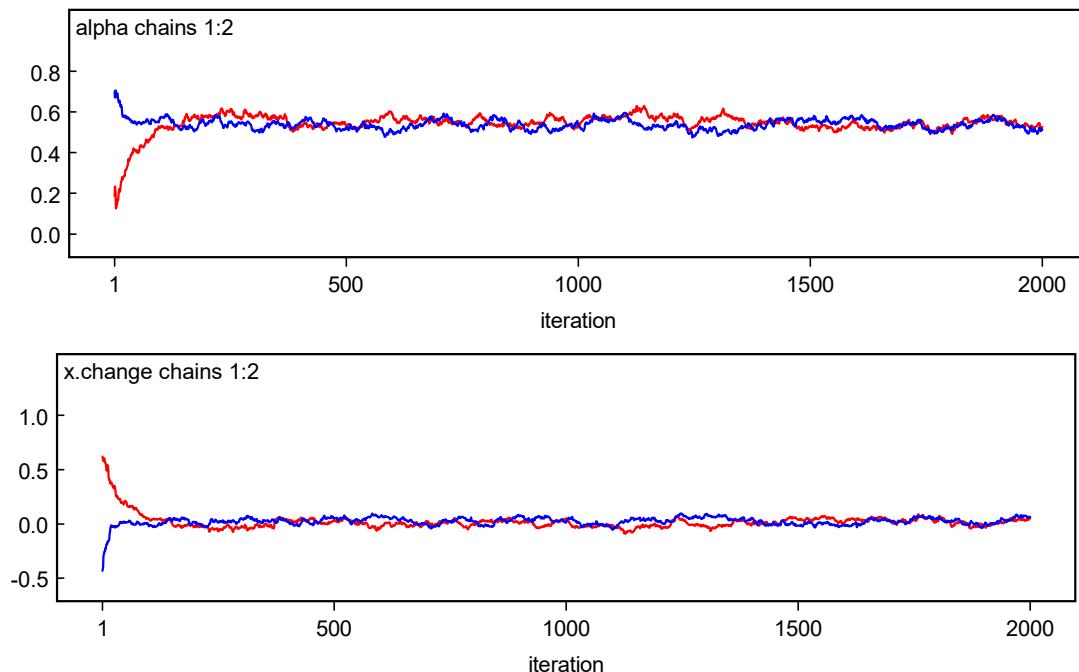
Data

```
⇒
list(Y = c(1.12, 1.12, 0.99, 1.03, 0.92, 0.90, 0.81, 0.83, 0.65, 0.67, 0.60, 0.59, 0.51, 0.44, 0.43,
0.43, 0.33, 0.30, 0.25, 0.24, 0.13, -0.01, -0.13, -0.14, -0.30, -0.33, -0.46, -0.43, -0.65),
x = c(-1.39, -1.39, -1.08, -1.08, -0.94, -0.80, -0.63, -0.63, -0.25, -0.25, -0.12, -0.12, 0.01, 0.11, 0.11,
0.11, 0.25, 0.25, 0.34, 0.34, 0.44, 0.59, 0.70, 0.70, 0.85, 0.85, 0.99, 0.99, 1.19),
N = 29)⇒
```

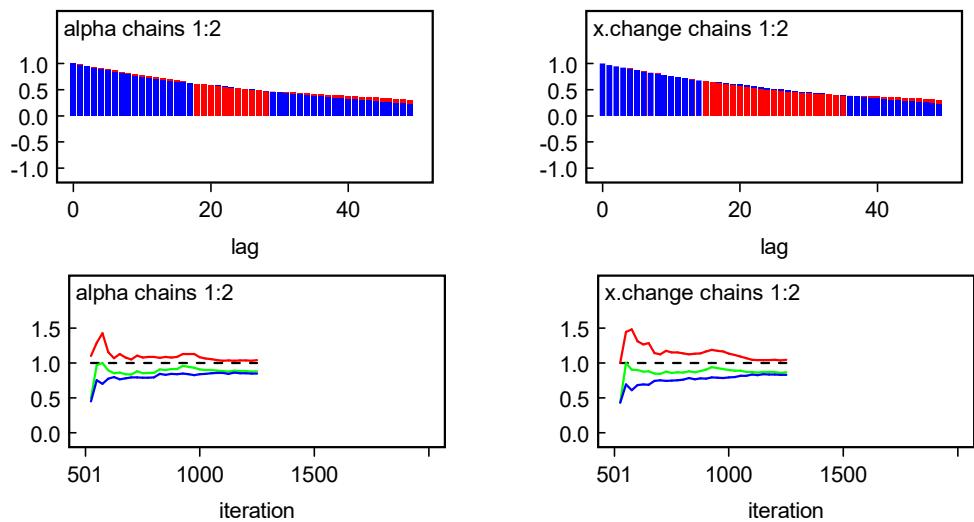
Inits using continuous changepoint

Using continuous changepoint
list(alpha = 0.47, beta = c(-0.45, -1.0), tau = 5, x.change = 0.5)
list(alpha = 0.47, beta = c(-0.45, -1.0), tau = 5, x.change = -0.5)

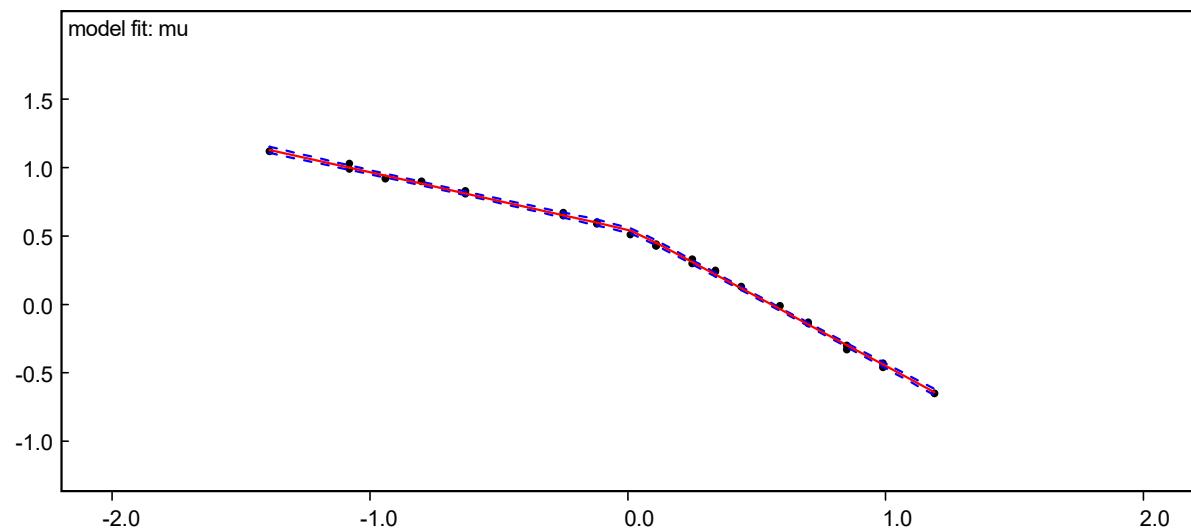
Results



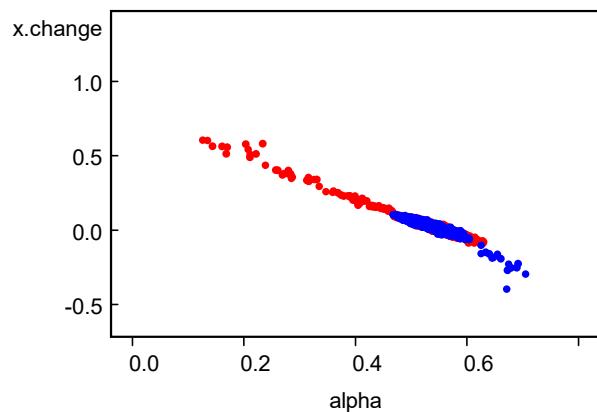
Not wonderful mixing, but reasonable



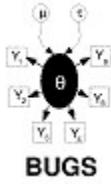
Good fit to data , (monitor mu and use as predicted values) use 'model fit' in Compare tool



Strong correlation of alpha and changepoint



alpha	x.change	r-value
		-0.932941



Asia: expert system

Evidence propagation

Lauritzen and Spiegelhalter (1988) introduce a fictitious "expert system" representing the diagnosis of a patient presenting to a chest clinic, having just come back from a trip to Asia and showing dyspnoea (shortness-of-breath). The BUGS code is shown below and the conditional probabilities used are given in Lauritzen and Spiegelhalter (1988). Note the use of `max` to do the logical-or. The `dcat` distribution is used to sample values with domain (1,2) with probability distribution given by the relevant entries in the conditional probability tables.

```

model
{
  smoking ~ dcat(p.smoking[1:2])
  tuberculosis ~ dcat(p.tuberculosis[asia,1:2])
  lung.cancer ~ dcat(p.lung.cancer[smoking,1:2])
  bronchitis ~ dcat(p.bronchitis[smoking,1:2])
  either <- max(tuberculosis,lung.cancer)
  xray ~ dcat(p.xray[either,1:2])
  dyspnoea ~ dcat(p.dyspnoea[either,bronchitis,1:2])
}

```

Data

```

list(asia = 2, dyspnoea = 2,
  p.tuberculosis = structure(.Data = c(0.99,0.01,0.95,0.05), .Dim = c(2,2)),
  p.bronchitis = structure(.Data = c(0.70,0.30,0.40,0.60), .Dim = c(2,2)),
  p.smoking = c(0.50,0.50),
  p.lung.cancer = structure(.Data = c(0.99,0.01,0.90,0.10), .Dim = c(2,2)),
  p.xray = structure(.Data = c(0.95,0.05,0.02,0.98), .Dim = c(2,2)),
  p.dyspnoea = structure(.Data = c(0.9,0.1,
  0.2,0.8,
  0.3,0.7,
  0.1,0.9), .Dim = c(2,2,2)))

```

Inits

```
list(smoking = 1, tuberculosis = 1, lung.cancer = 1, bronchitis = 1, xray = 1)
```

Results

Time for 10000 updates on was 200MHz PentiumPro 8s. A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
bronchitis	1.816	0.3874	0.004521	1.0	2.0	2.0	1001	10000
either	1.185	0.3881	0.004478	1.0	1.0	2.0	1001	10000
lung.cancer	1.107	0.309	0.003215	1.0	1.0	2.0	1001	10000
smoking	1.63	0.4828	0.005122	1.0	2.0	2.0	1001	10000
tuberculosis	1.084	0.2775	0.003069	1.0	1.0	2.0	1001	10000
xray	1.222	0.4153	0.004783	1.0	1.0	2.0	1001	10000