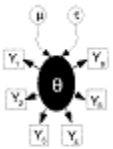
Examples Volume 1

References:

Sorry - an on-line version of the references is currently unavailable.

Please refer to the existing Examples documentation available from
`http://www.mrc-bsu.cam.ac.uk/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 ith 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

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

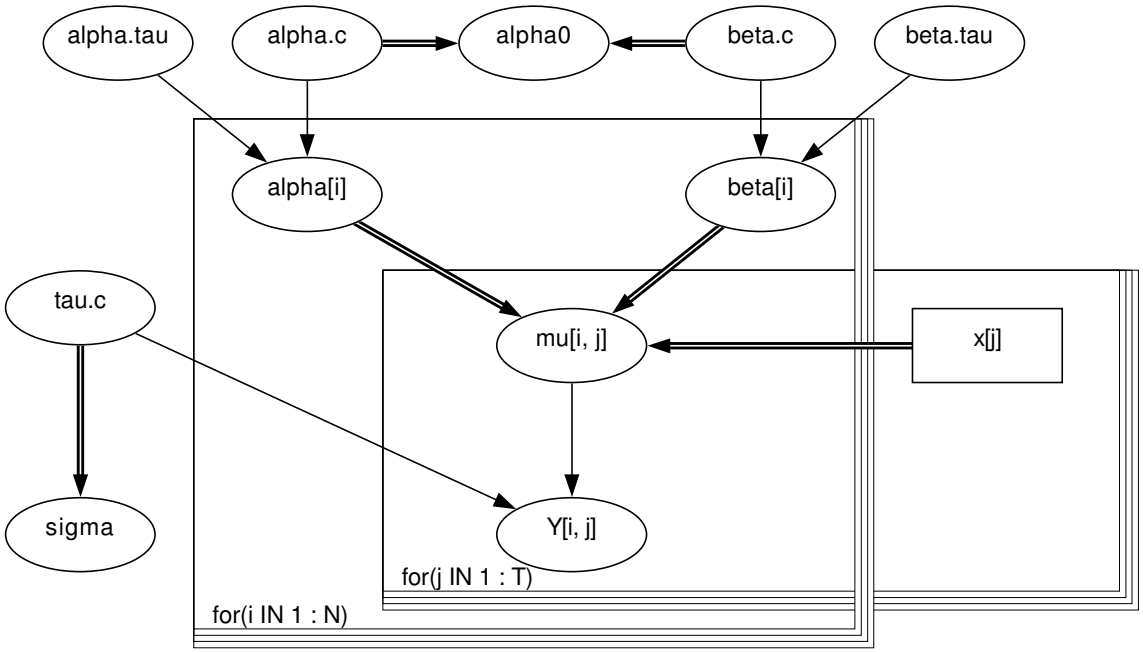
$$\alpha_i \sim \text{Normal}(\alpha_c, \tau_\alpha)$$

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

where x_{bar} = 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).

α_c , τ_α , β_c , τ_β , τ_c are given independent ``noninformative" priors. Interest particularly focuses on the intercept at zero time (birth), denoted $\alpha_0 = \alpha_c - \beta_c x_{bar}$.

Graphical model for rats example:



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,alpha.tau)
      beta[i] ~ dnorm(beta.c,beta.tau)
   }
   tau.c ~ dgamma(0.001,0.001)
   sigma <- 1 / sqrt(tau.c)
   alpha.c ~ dnorm(0.0,1.0E-6)
   alpha.tau ~ dgamma(0.001,0.001)
   beta.c ~ dnorm(0.0,1.0E-6)
   beta.tau ~ dgamma(0.001,0.001)
   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 (click to open)

Inits (click to open)

(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 "load data" from the "Specification" dialog.)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha0	106.6	3.655	0.04079	99.44	106.5	113.8	1001	10000
beta.c	6.185	0.1061	0.00132	5.975	6.185	6.394	1001	10000
sigma	6.086	0.4606	0.007398	5.255	6.061	7.049	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.

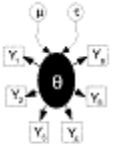
Data (click to open)

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:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
Y[26,2]	204.6	8.689	0.1145	187.6	204.7	221.4	1001	10000
Y[26,3]	250.2	10.21	0.1732	230.1	250.2	270.5	1001	10000
Y[26,4]	295.6	12.5	0.228	270.6	295.5	319.7	1001	10000
Y[26,5]	341.2	15.29	0.2936	310.7	341.3	370.9	1001	10000
beta.c	6.578	0.1497	0.003415	6.284	6.578	6.87	1001	10000

We note that our estimate 6.58 of β_c 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.



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:

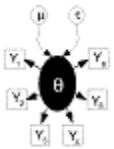
```
model
{
   for (i in 1 : Dogs) {
      xa[i, 1] <- 0; xs[i, 1] <- 0 p[i, 1] <- 0
      for (j in 2 : Trials) {
         xa[i, j] <- sum(Y[i, 1 : j - 1])
         xs[i, j] <- j - 1 - xa[i, j]
         log(p[i, j]) <- alpha * xa[i, j] + beta * xs[i, j]
         y[i, j] <- 1 - Y[i, j]
         y[i,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

Data (click to open)

Inits(click to open)

Results

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
A	0.7833	0.01917	2.665E-4	0.746	0.7833	0.8207	1001	10000
B	0.9242	0.01089	1.573E-4	0.9018	0.9247	0.9445	1001	10000
alpha	-0.2446	0.02451	3.4E-4	-0.293	-0.2442	-0.1976	1001	10000
beta	-0.07886	0.0118	1.705E-4	-0.1033	-0.07825	-0.05711	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,...,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

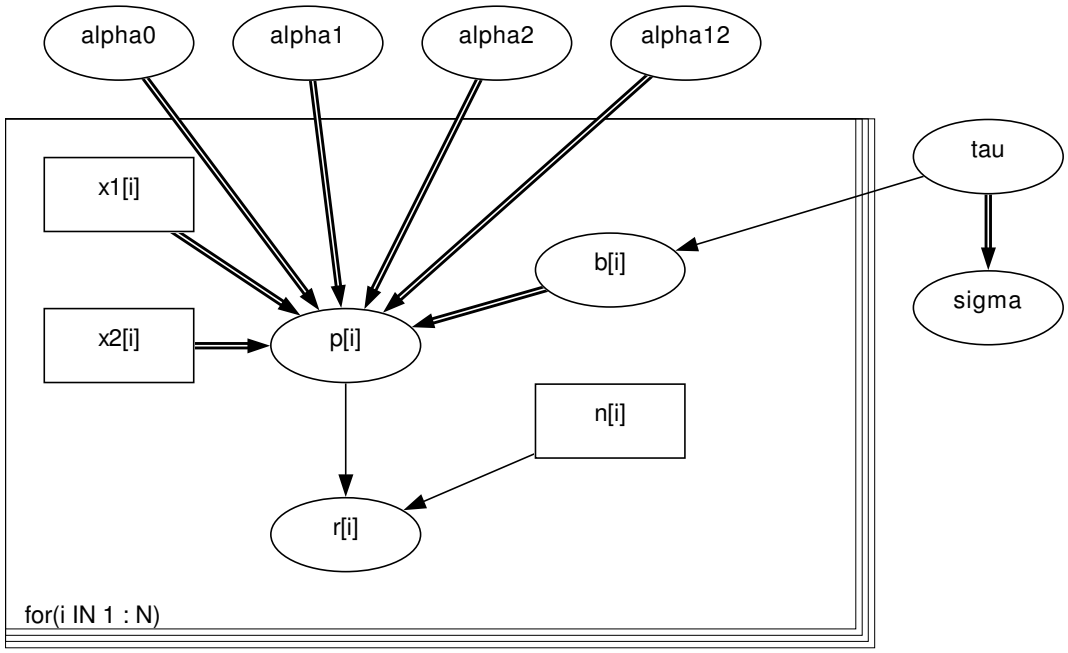
$$r_i \sim \text{Binomial}(p_i, n_i)$$

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

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

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} , τ are given independent "noninformative" priors.

Graphical model for seeds example



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)
   tau ~ dgamma(0.001,0.001)
   sigma <- 1 / sqrt(tau)
}
```

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha0	-0.5525	0.1852	0.00402	-0.9312	-0.5505	-0.1879	1001	10000
alpha1	0.08382	0.3031	0.005803	-0.5238	0.09076	0.6794	1001	10000
alpha12	-0.8165	0.4109	0.008128	-1.671	-0.8073	-0.0287	1001	10000
alpha2	1.346	0.2564	0.00553	0.8501	1.34	1.881	1001	10000
sigma	0.267	0.1471	0.007996	0.03842	0.2552	0.5929	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 β	SE	maximum likelihood β	SE	PQL 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

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

$$\beta_i = \mu_i + b_i$$

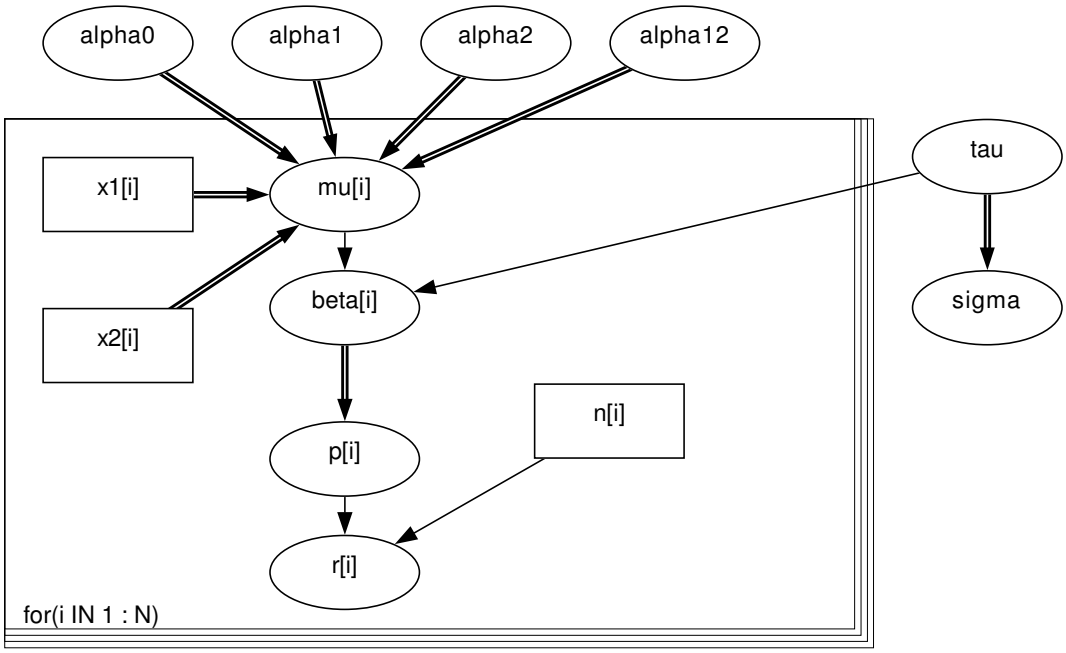
the model then becomes

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

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

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

The graphical model is shown below



This formulation of the model has two advantages: the squence 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.

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

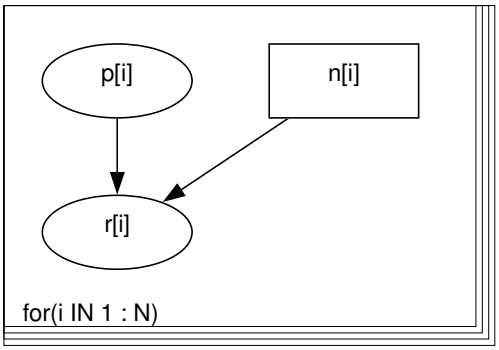
The number of deaths r_i for hospital *i* are modelled as a binary response variable with `true' failure probability p_i:

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

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 (click to open)

Inits (click to open)

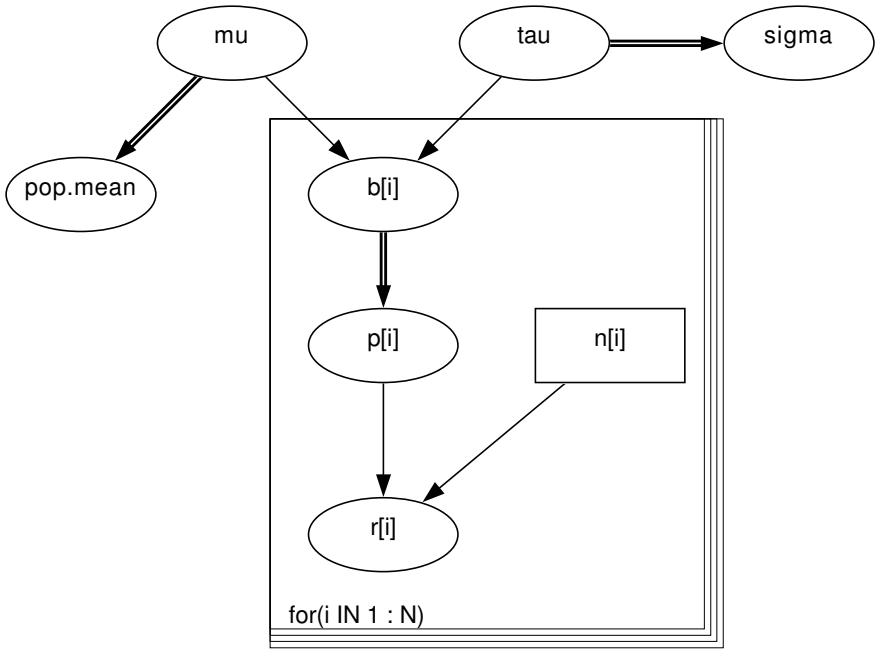
A more realistic model for the surgical data is to assume that the failure rates across hospitals are *similar* in some way. This is equivalent to specifying a *random effects* model for the true failure probabilities p_i as follows:

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

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

Standard non-informative priors are then specified for the population mean (logit) probability of failure, μ, and precision, τ.

Graphical model for random effects surgical example:



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)
   sigma <- 1 / sqrt(tau)
   tau ~ dgamma(0.001,0.001)
}
```

Data (click to open)

Inits (click to open)

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
p[1]	0.02009	0.01946	2.085E-4	6.091E-4	0.01441	0.07178	1001	10000
p[2]	0.1266	0.0271	2.67E-4	0.07853	0.125	0.1845	1001	10000
p[3]	0.07436	0.02371	2.349E-4	0.03492	0.07181	0.1265	1001	10000
p[4]	0.05789	0.00824	8.136E-5	0.04264	0.05762	0.07487	1001	10000
p[5]	0.04237	0.01388	1.096E-4	0.01972	0.04086	0.07362	1001	10000
p[6]	0.07081	0.01811	1.935E-4	0.0397	0.06931	0.1098	1001	10000
p[7]	0.06686	0.02025	1.872E-4	0.03259	0.06493	0.111	1001	10000
p[8]	0.1473	0.02393	2.681E-4	0.1039	0.146	0.1983	1001	10000
p[9]	0.07216	0.0179	1.59E-4	0.04093	0.07071	0.1104	1001	10000
p[10]	0.09078	0.0288	3.122E-4	0.04274	0.08817	0.1531	1001	10000
p[11]	0.1165	0.02009	2.074E-4	0.08	0.1155	0.1589	1001	10000
p[12]	0.06906	0.01345	1.261E-4	0.04518	0.06816	0.0977	1001	10000

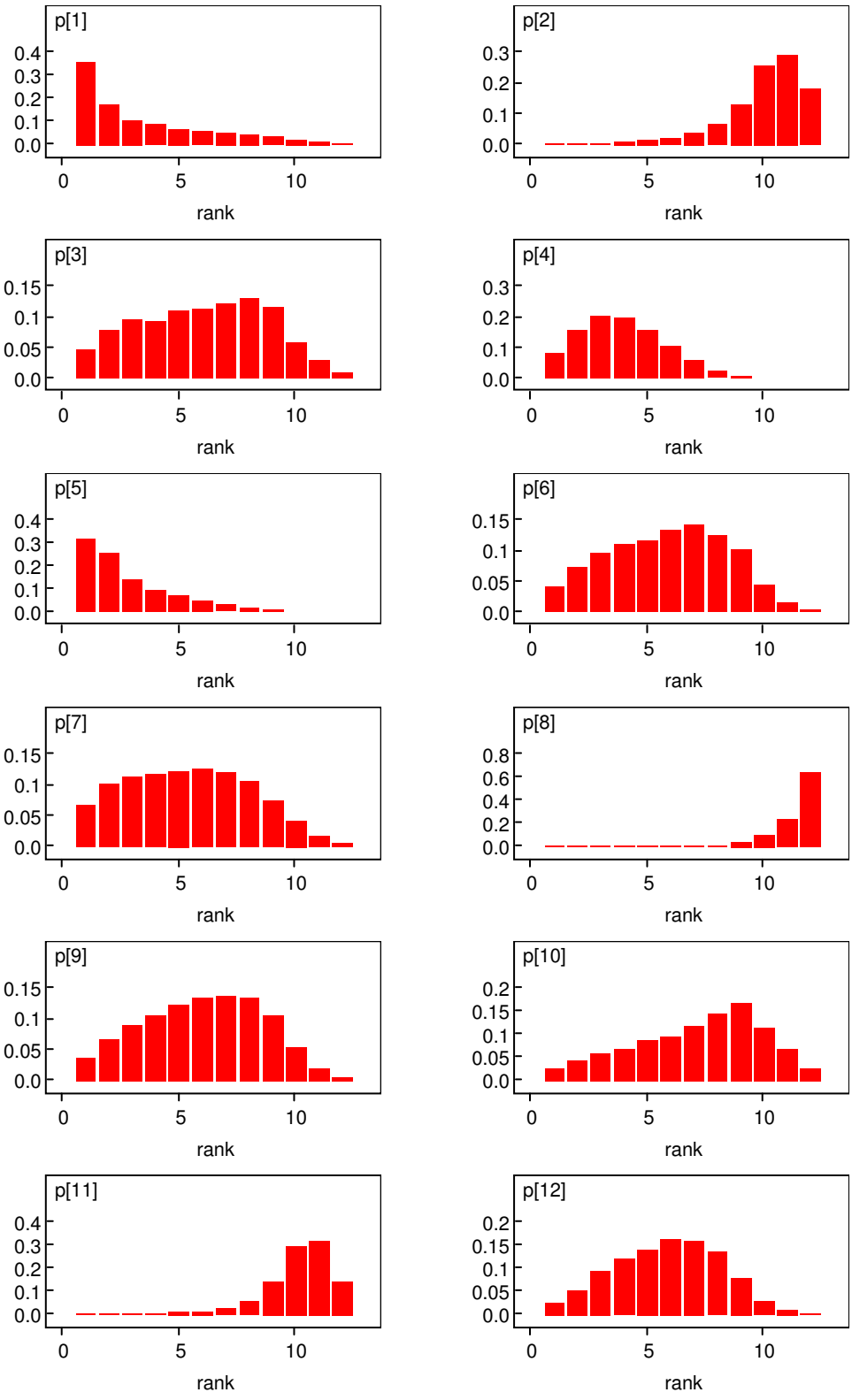
and for the random effects analysis

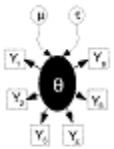
	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
mu	-2.558	0.1554	0.002585	-2.884	-2.551	-2.266	1001	10000
p[1]	0.05302	0.01948	3.565E-4	0.01802	0.05221	0.09348	1001	10000
p[2]	0.1029	0.02196	2.976E-4	0.06712	0.1006	0.152	1001	10000
p[3]	0.07044	0.01727	1.978E-4	0.0397	0.06916	0.1079	1001	10000
p[4]	0.0593	0.007985	1.212E-4	0.04458	0.05897	0.07591	1001	10000
p[5]	0.05187	0.01329	2.269E-4	0.02791	0.05102	0.07961	1001	10000
p[6]	0.06903	0.01448	1.564E-4	0.04284	0.06854	0.1004	1001	10000
p[7]	0.06682	0.01602	1.773E-4	0.03835	0.06595	0.1009	1001	10000
p[8]	0.1226	0.02244	4.014E-4	0.08196	0.1217	0.1698	1001	10000
p[9]	0.0698	0.01432	1.508E-4	0.04432	0.06901	0.1004	1001	10000
p[10]	0.07851	0.01955	2.03E-4	0.04506	0.07662	0.1217	1001	10000
p[11]	0.1021	0.01761	2.283E-4	0.07158	0.1009	0.1398	1001	10000
p[12]	0.06858	0.01168	1.301E-4	0.04745	0.06805	0.09349	1001	10000
pop.mean	0.07259	0.01028	1.696E-4	0.05293	0.07235	0.09401	1001	10000
sigma	0.4028	0.16	0.003672	0.1577	0.3793	0.7872	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* probabilty 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:


p[1]
p[2]
p[3]
p[4]
p[5]
p[6]
p[7]
p[8]
p[9]
p[10]
p[11]
p[12]
rank



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 *i*1, *i*2 and *i*3. Then we assume

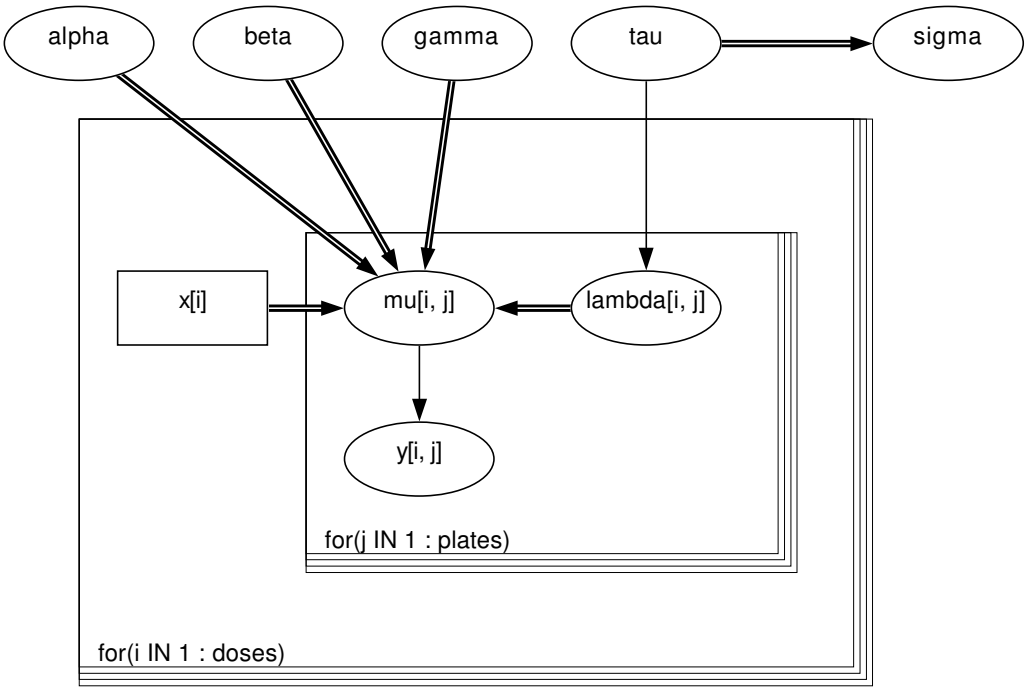
$$y_{ij} \sim \text{Poisson}(\mu_{ij})$$

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

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

α , β , γ , τ are given independent ``noninformative" priors. The appropriate graph is shown

Graphical model for salm example



BUGS *language for salm example*

```
model
{
   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)
   tau ~ dgamma(0.001, 0.001)
   sigma <- 1 / sqrt(tau)
}
```

Data (click to open)

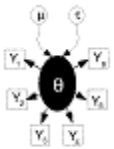
Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha	2.193	0.3874	0.01118	1.438	2.194	2.959	1001	10000
beta	0.3059	0.1054	0.003266	0.09692	0.3065	0.5131	1001	10000
gamma	-9.577E-4	4.525E-4	1.48E-5	-0.001837	-9.622E-4	-3.196E-5	1001	10000
sigma	0.2608	0.08077	0.002114	0.1305	0.2512	0.4472	1001	10000

These estimates can be compared with the quasi-likelihood estimates of Breslow (1984) who reported α = 2.203 +/- 0.363, β = 0.311 +/- 0.099, γ = -9.74E-4 +/- 4.37E-4, σ = 0.268



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 ($i = 1,...,10$) in the *k* th period ($k = 1,2$) is assumed to be of the form

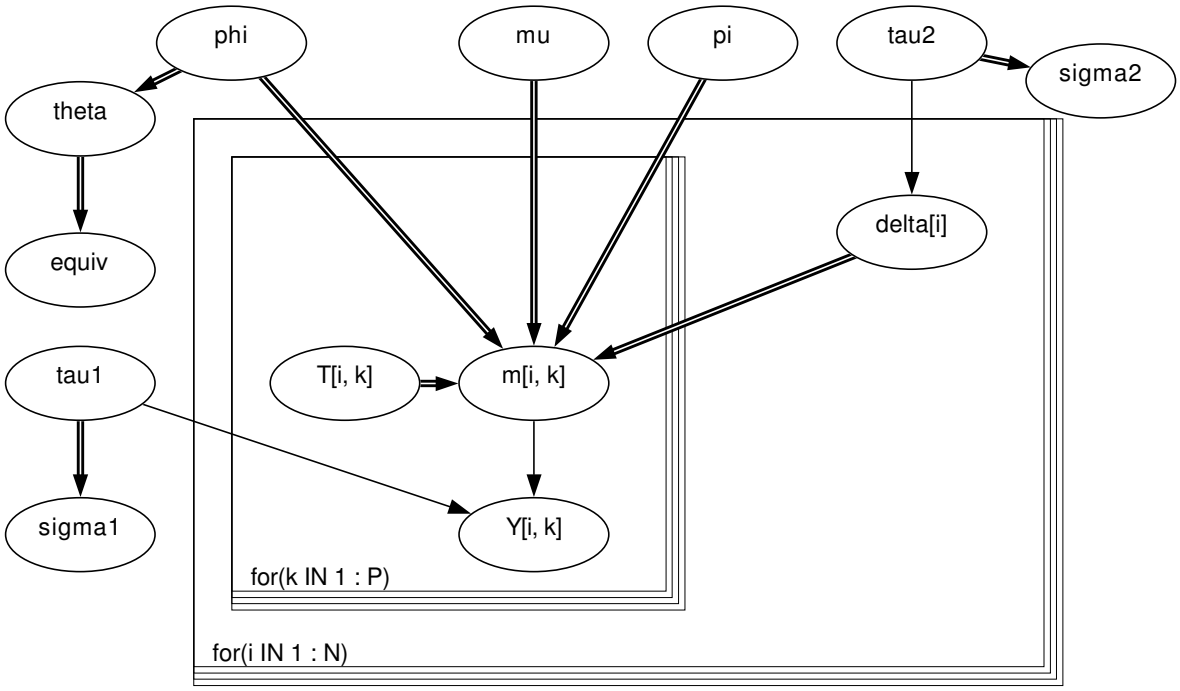
$$Y_{ik} \sim \text{Normal}(m_{ik}, \tau_1)$$

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

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

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*. The graph of this model and its *BUGS* language description are shown below

Graphical model for equiv example



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)
   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 bioequivelence.

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
equiv	0.9976	0.04893	5.148E-4	1.0	1.0	1.0	1001	10000
mu	1.437	0.05364	0.001822	1.329	1.437	1.542	1001	10000
phi	-0.008338	0.05201	5.151E-4	-0.1127	-0.008502	0.09926	1001	10000
pi	-0.1802	0.05189	4.793E-4	-0.2834	-0.1803	-0.07353	1001	10000
sigma1	0.1106	0.03374	9.345E-4	0.06492	0.1033	0.194	1001	10000
sigma2	0.1399	0.05357	0.001464	0.04614	0.1355	0.2624	1001	10000
theta	0.993	0.05181	5.125E-4	0.8934	0.9915	1.104	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

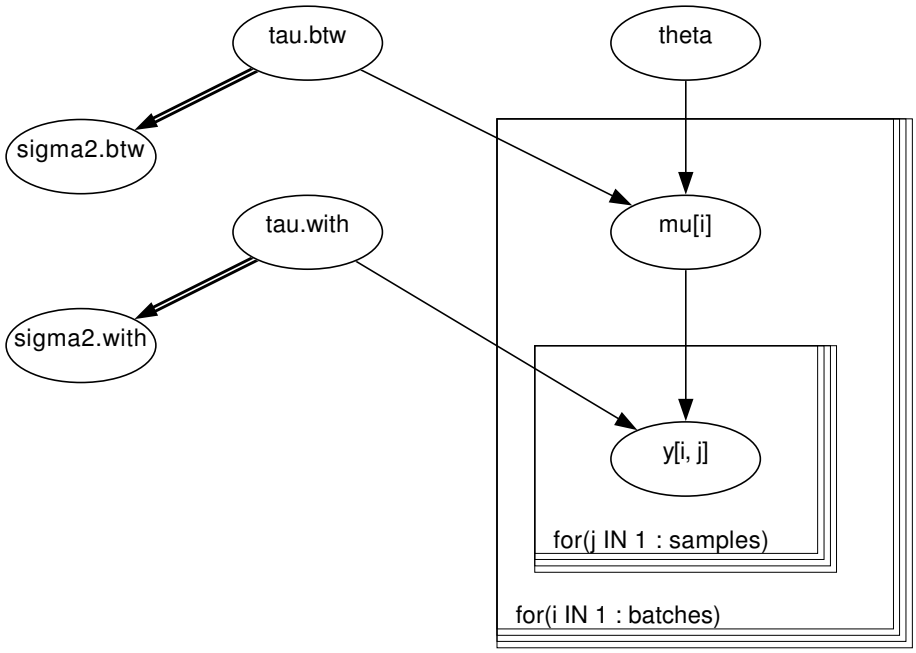
The object of the study was to determine the relative importance of between batch variation versus variation due to sampling and analytic errors. On the assumption that the batches and samples vary independently, and contribute additively to the total error variance, we may assume the following model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

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

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

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 $\tau_{between}$ is the inverse of the between-batch variance $s^2_{between}$. The total variation in product yield is thus $\sigma^2_{total} = \sigma^2_{within} + \sigma^2_{between}$ and the relative contributions of each component to the total variance are $f_{within} = \sigma^2_{within} / \sigma^2_{total}$ and $f_{between} = \sigma^2_{between} / \sigma^2_{total}$. We assume standard non-informative priors for θ, τ_{within} and $\tau_{between}$.

Graphical model for dyes example



Bugs language for dyes example

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

Data (click to open)

Inits (click to open)

Results

A 25000 update burn in followed by a further 100000 updates gave the parameter estimates

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
sigma2.btw	2207.0	4289.0	40.94	0.008042	1290.0	10210.0	25001	100000
sigma2.with	3034.0	1102.0	19.89	1561.0	2807.0	5752.0	25001	100000
theta	1528.0	21.5	0.1712	1484.0	1527.0	1571.0	25001	100000

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 $\sigma^2_{between}$ has a very long upper tail: hence the posterior mean is considerably larger than the median. Box and Tiao estimate σ^2_{within} = 2451 and $\sigma^2_{between}$ = 1764 by classical analysis of variance. Here, $\sigma^2_{between}$ 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 $\sigma^2_{between}$ 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 $z_{ij} = (x_{ij} - \text{xbar}_j) / \text{sd}(x_j)$ 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 expontential (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

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

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 (click to open)

Inits (click to open)

Results

a) Normal error

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b[1]	0.7135	0.1402	0.002709	0.436	0.7129	0.9943	1001	10000
b[2]	1.301	0.3829	0.007076	0.5377	1.3	2.053	1001	10000
b[3]	-0.1511	0.1666	0.00215	-0.4792	-0.1511	0.1836	1001	10000
b0	-40.0	12.64	0.136	-65.26	-40.02	-14.64	1001	10000
outlier[3]	0.01	0.0995	8.876E-4	0.0	0.0	0.0	1001	10000
outlier[4]	0.0494	0.2167	0.002287	0.0	0.0	1.0	1001	10000
outlier[21]	0.3118	0.4632	0.005947	0.0	0.0	1.0	1001	10000
sigma	3.393	0.6217	0.007985	2.435	3.308	4.859	1001	10000

b) Double exponential error

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b[1]	0.831	0.131	0.003069	0.5563	0.8341	1.092	1001	10000
b[2]	0.7545	0.3405	0.007192	0.1691	0.7245	1.521	1001	10000
b[3]	-0.1152	0.1199	0.001538	-0.3629	-0.1106	0.1157	1001	10000
b0	-38.78	8.788	0.08773	-56.6	-38.76	-21.08	1001	10000
outlier[1]	0.0453	0.208	0.002081	0.0	0.0	1.0	1001	10000
outlier[3]	0.0578	0.2334	0.002381	0.0	0.0	1.0	1001	10000
outlier[4]	0.2929	0.4551	0.00534	0.0	0.0	1.0	1001	10000
outlier[21]	0.59	0.4918	0.007638	0.0	1.0	1.0	1001	10000
sigma	3.492	0.8657	0.01174	2.169	3.362	5.528	1001	10000

c) t4 error

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b[1]	0.8361	0.1412	0.004002	0.5501	0.8401	1.114	1001	10000
b[2]	0.8565	0.3744	0.01016	0.1622	0.8302	1.67	1001	10000
b[3]	-0.1254	0.1289	0.001917	-0.3816	-0.1251	0.1311	1001	10000
b0	-40.22	9.839	0.1218	-59.98	-40.16	-20.85	1001	10000
outlier[3]	0.0334	0.1797	0.002467	0.0	0.0	1.0	1001	10000
outlier[4]	0.2343	0.4236	0.007791	0.0	0.0	1.0	1001	10000
outlier[21]	0.5904	0.4918	0.01125	0.0	1.0	1.0	1001	10000
sigma	3.478	0.8485	0.01899	2.146	3.368	5.464	1001	10000

d) Normal eror ridge regression

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b[1]	0.6816	0.1345	0.002706	0.4144	0.6828	0.9451	1001	10000
b[2]	1.317	0.3652	0.006841	0.5829	1.318	2.042	1001	10000
b[3]	-0.1266	0.1644	0.002001	-0.4488	-0.1267	0.2017	1001	10000
b0	-40.53	12.46	0.1255	-65.46	-40.51	-15.94	1001	10000
outlier[3]	0.0189	0.1362	0.001325	0.0	0.0	0.0	1001	10000
outlier[4]	0.0471	0.2119	0.002388	0.0	0.0	1.0	1001	10000
outlier[21]	0.2795	0.4488	0.006227	0.0	0.0	1.0	1001	10000
sigma	3.395	0.6197	0.007177	2.425	3.312	4.828	1001	10000

e) Double exponential error ridge regression

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

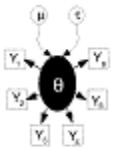
	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b[1]	0.796	0.1328	0.00319	0.5135	0.8005	1.058	1001	10000
b[2]	0.7883	0.3334	0.007516	0.2109	0.7546	1.53	1001	10000
b[3]	-0.09911	0.1169	0.001705	-0.3386	-0.09686	0.1279	1001	10000
b0	-38.82	8.665	0.09608	-56.67	-38.77	-21.62	1001	10000
outlier[1]	0.0603	0.238	0.00301	0.0	0.0	1.0	1001	10000
outlier[3]	0.0735	0.261	0.00317	0.0	0.0	1.0	1001	10000
outlier[4]	0.2875	0.4526	0.005526	0.0	0.0	1.0	1001	10000
outlier[21]	0.5463	0.4979	0.008318	0.0	1.0	1.0	1001	10000
sigma	3.499	0.8798	0.01195	2.17	3.359	5.636	1001	10000

f) t4 error ridge regression

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b[1]	0.7949	0.1399	0.003333	0.5108	0.7975	1.063	1001	10000
b[2]	0.9094	0.3624	0.008192	0.2511	0.8879	1.701	1001	10000
b[3]	-0.1087	0.1304	0.001715	-0.3691	-0.1071	0.1468	1001	10000
b0	-40.34	9.901	0.1074	-60.01	-40.33	-20.52	1001	10000
outlier[1]	0.0327	0.1779	0.0023	0.0	0.0	1.0	1001	10000
outlier[3]	0.0466	0.2108	0.002868	0.0	0.0	1.0	1001	10000
outlier[4]	0.2127	0.4092	0.007202	0.0	0.0	1.0	1001	10000
outlier[21]	0.5218	0.4995	0.01065	0.0	1.0	1.0	1001	10000
sigma	3.521	0.8663	0.01802	2.157	3.41	5.487	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.



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:

$$r^C_i \sim \text{Binomial}(p^C_i, n^C_i)$$

$$r^T_i \sim \text{Binomial}(p^T_i, n^T_i)$$

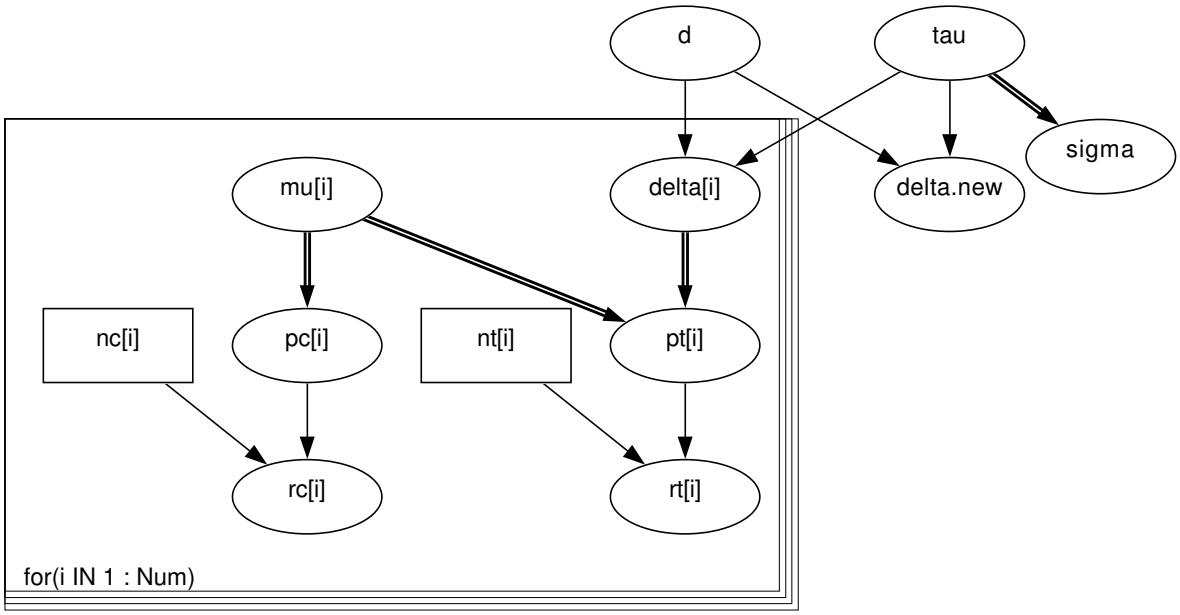
$$\text{logit}(p^C_i) = \mu_i$$

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

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

``Noninformative" priors are given for the μ_i's. τ and d. The graph for this model is shown in 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 $p(\delta_{new} \mid d_{hat}, \tau_{hat})$. *Full Bayes* allows for the uncertainty concerning d and τ.

Graphical model for blocker example:



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] ~ dt(d, tau, 4)
   }
   d ~ dnorm(0.0,1.0E-6)
   tau ~ dgamma(0.001,0.001)
   delta.new ~ dt(d, tau, 4)
   sigma <- 1 / sqrt(tau)
}
```

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
d	-0.2492	0.06422	0.002004	-0.3727	-0.2502	-0.1194	1001	10000
delta.new	-0.2499	0.1509	0.002389	-0.5592	-0.2536	0.07169	1001	10000
sigma	0.1189	0.07	0.003521	0.02428	0.1067	0.2825	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 (approximate) use of $p(\sigma^2) \sim 1 / \sigma^2$ (see his Figure 1).

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

$$\log\psi_i = \alpha + \beta_1 year_i + \beta_2(year_i^2 - 22) + b_i$$

$$b_i \sim Normal(0, \tau)$$

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

We let r^0_i denote number of exposures among the n^0_i controls in stratum *i*, and r^1_i denote number of exposures for the n^1_i cases. The we assume

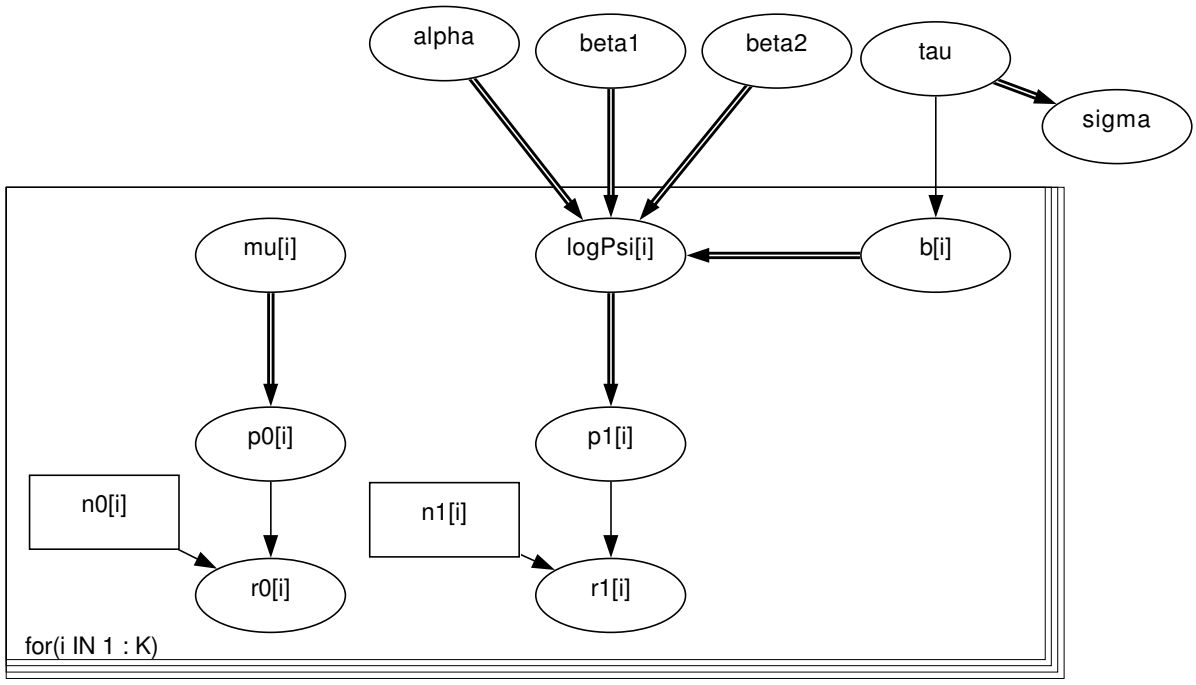
$$r^0_i \sim Binomial(p^0_i, n^0_i)$$

$$r^1_i \sim Binomial(p^1_i, n^1_i)$$

$$logit(p^0_i) = \mu_i$$

$$logit(p^1_i) = \mu_i + \log\psi_i$$

Assuming this model with independent vague priors for the μ_i's provides the correct conditional likelihood. The appropriate graph 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)
   tau    ~ dgamma(1.0E-3, 1.0E-3)
   sigma <- 1 / sqrt(tau)
}
```

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha	0.579	0.062	0.001545	0.4587	0.5793	0.7037	1001	10000
beta1	-0.04557	0.01553	3.929E-4	-0.07688	-0.0457	-0.01586	1001	10000
beta2	0.007041	0.003084	8.953E-5	0.001018	0.007004	0.01314	1001	10000
sigma	0.09697	0.06011	0.005036	0.02419	0.08059	0.2457	1001	10000

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

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:

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

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

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

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

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

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 $a_k = \alpha_k - \alpha_{bar}$ to enable comparision of the *BUGS* posterior parameter estimates with the Boch and Aitkin marginal maximum likelihood estimates.

BUGS *language for LSAT model*

```
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.

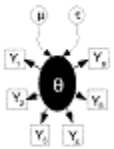
Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a[1]	-1.261	0.1048	0.001383	-1.468	-1.258	-1.06	1001	10000
a[2]	0.4788	0.07014	7.807E-4	0.3419	0.4787	0.62	1001	10000
a[3]	1.238	0.07017	9.431E-4	1.102	1.237	1.376	1001	10000
a[4]	0.1697	0.0731	8.377E-4	0.02005	0.1718	0.3104	1001	10000
a[5]	-0.6257	0.08552	0.001034	-0.7943	-0.6242	-0.4612	1001	10000
beta	0.7581	0.06914	0.001899	0.6223	0.7576	0.8957	1001	10000



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 boundarys 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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 2 1 1
2	1.0	2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 3 1 1 2 1 1
.	.	. . . . .
.	.	. . . . .
12	16.0	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 1 NA 2 1 3 2 5 5 5 5 5 5
13	18.0	2 2 2 2 2 2 2 2 2 2 NA 2 2 2 2 2 2 2 2 2 3 3 3 NA 2 NA 2 3 4 5 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.

Thissen models the above data using 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 $p_{j,K_j} = Q_{j,K_j-1}$. For items with 3 or more categories, the probability of observing an itermediate grade is $p_{j,k} = Q_{j,k-1} - Q_{j,k}$ (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 theta[i] ~ dnorm(0.0, 0.001). That is, each boy is treated as a separate problem with is no `learning' or `borrowing strength' across individuals, and hence no hierachical 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 : nInd) {
# 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 : nInd) {
            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 (click to open)

Inits (click to open)

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

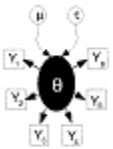
```
grade[i, j] ~ dcat(p[i, j, 1  :ngrade[j]])
```

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
theta[1]	0.3244	0.2085	0.002304	-0.0999	0.3317	0.7238	1001	10000
theta[2]	1.366	0.256	0.002403	0.8998	1.352	1.899	1001	10000
theta[3]	2.357	0.2726	0.002822	1.823	2.355	2.903	1001	10000
theta[4]	2.902	0.2959	0.002816	2.317	2.901	3.476	1001	10000
theta[5]	5.535	0.4996	0.004758	4.599	5.527	6.542	1001	10000
theta[6]	6.751	0.6046	0.006357	5.597	6.741	7.951	1001	10000
theta[7]	6.451	0.5857	0.005726	5.358	6.431	7.638	1001	10000
theta[8]	8.93	0.6971	0.006642	7.546	8.938	10.3	1001	10000
theta[9]	8.981	0.6719	0.007133	7.629	8.993	10.27	1001	10000
theta[10]	11.94	0.6871	0.00698	10.63	11.93	13.28	1001	10000
theta[11]	11.58	0.9078	0.009463	9.957	11.53	13.48	1001	10000
theta[12]	15.79	0.5624	0.005932	14.72	15.79	16.92	1001	10000
theta[13]	16.96	0.7477	0.007337	15.56	16.93	18.52	1001	10000



Inhaler: ordered catagorical 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				
		1 Easy	**2 Only clear after re-reading**	**3 Not very clear**	**4 Confusing**	**TOTAL**
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 ($i = 1,...,286$) 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 (-inf, inf) as follows:

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

where a_0 = -inf and a_4 = inf. 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. Prob(Y_{it} >= a_j) is given by

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

where b_i represents the random effect for subject *i*. Here, $\mu_{s_i t}$ 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 $p_{itj} = Q_{itj - 1} - Q_{itj}$, 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. 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) {
    for(t in 1 : T) {
# logistic mean for group i in period t
        mu[g, t] <- beta * treat[g, t] / 2 + pi * period[g, t] / 2 + kappa * carry[g, t]
    }
  }
  beta ~ dnorm(0, 1.0E-06)
  pi ~ dnorm(0, 1.0E-06)
  kappa ~ dnorm(0, 1.0E-06)

# ordered cut points for underlying continuous latent variable
  a[1] ~ dnorm(0, 1.0E-06)I(, a[2])
  a[2] ~ dnorm(0, 1.0E-06)I(a[1], a[3])
  a[3] ~ dnorm(0, 1.0E-06)I(a[2], )

  tau ~ dgamma(0.001, 0.001)
  sigma <- sqrt(1 / tau)
  log.sigma <- log(sigma)

}
```

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

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a[1]	0.712	0.1382	0.004156	0.4566	0.7069	0.9981	1001	10000
a[2]	3.936	0.3298	0.01597	3.326	3.924	4.614	1001	10000
a[3]	5.28	0.4699	0.01893	4.412	5.262	6.239	1001	10000
beta	1.067	0.3199	0.008454	0.4575	1.057	1.714	1001	10000
kappa	0.2463	0.2503	0.005605	-0.2394	0.2456	0.7475	1001	10000
log.sigma	0.195	0.203	0.01356	-0.2494	0.2145	0.5424	1001	10000
pi	-0.2367	0.1976	0.002313	-0.6214	-0.238	0.1509	1001	10000
sigma	1.24	0.2412	0.01562	0.7793	1.239	1.72	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 β = 1.17 +/- 0.75, π = -0.23 +/- 0.20, κ = 0.21 +/- 0.49, logσ = 0.17 +/- 0.23, a1 = 0.68, a2 = 3.85, a3 = 5.10

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 portion of the data, giving survival times in weeks, are shown below. 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

$$f(t_i, \mathbf{z}_i) = re^{\boldsymbol{\beta}\mathbf{z}_i} t_i^{r-1} \exp(-e^{\boldsymbol{\beta}\mathbf{z}_i} t_i^r)$$

where t_i is the failure time of an individual with covariate vector $\mathbf{z}_i$ and $\boldsymbol{\beta}$ is a vector of unknown regression coefficients. This leads to a baseline hazard function of the form

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

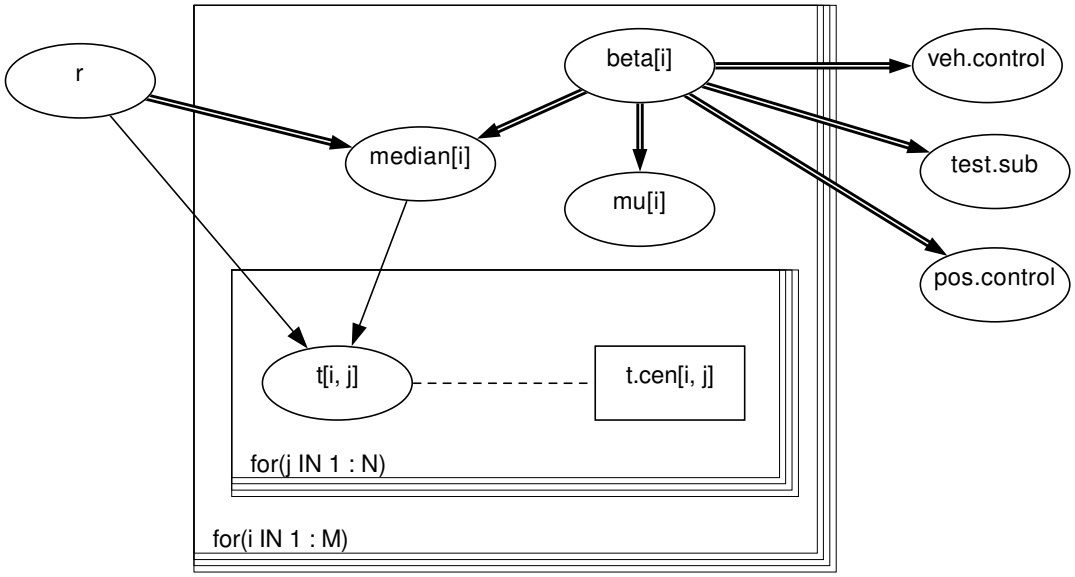
Setting $\mu_i = e^{\boldsymbol{\beta}\mathbf{z}_i}$ gives the parameterisation

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

For censored observations, the survival distribution is a truncated Weibull, with lower bound corresponding to the censoring time. The regression $\boldsymbol{\beta}$ 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 $\mathbf{z}_i$ is given by $m_i = (\log 2e^{-\boldsymbol{\beta}\mathbf{z}_i})^{1/r}$

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]
}
```

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 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.

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
median[1]	23.65	2.002	0.05203	20.06	23.54	27.93	1001	10000
median[2]	35.18	3.54	0.05764	29.22	34.88	43.21	1001	10000
median[3]	26.68	2.437	0.05828	22.36	26.5	31.99	1001	10000
median[4]	21.28	1.849	0.03371	18.01	21.16	25.36	1001	10000
pos.control	0.3088	0.3416	0.005912	-0.3644	0.3115	0.9685	1001	10000
r	2.902	0.2781	0.02332	2.367	2.904	3.444	1001	10000
test.sub	-0.3475	0.3435	0.004663	-1.022	-0.3478	0.3185	1001	10000
veh.control	-1.143	0.365	0.006778	-1.87	-1.141	-0.4315	1001	10000

Kidney: Weibull regression with random efects

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

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

$$\log\mu_{ij} = \alpha + \beta_{age}AGE_{ij} + \beta_{sex}SEX_i + \beta_{disease1}DISEASE_{i1} + \beta_{disease2}DISEASE_{i2} + \beta_{disease3}DISEASE_{i3} + b_i$$

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

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 the survival distribution is a truncated Weibull for censored observations as discussed in the mice example. The regression coefficients and the precision of the random effects τ are given independent ``non-informative" priors, namely

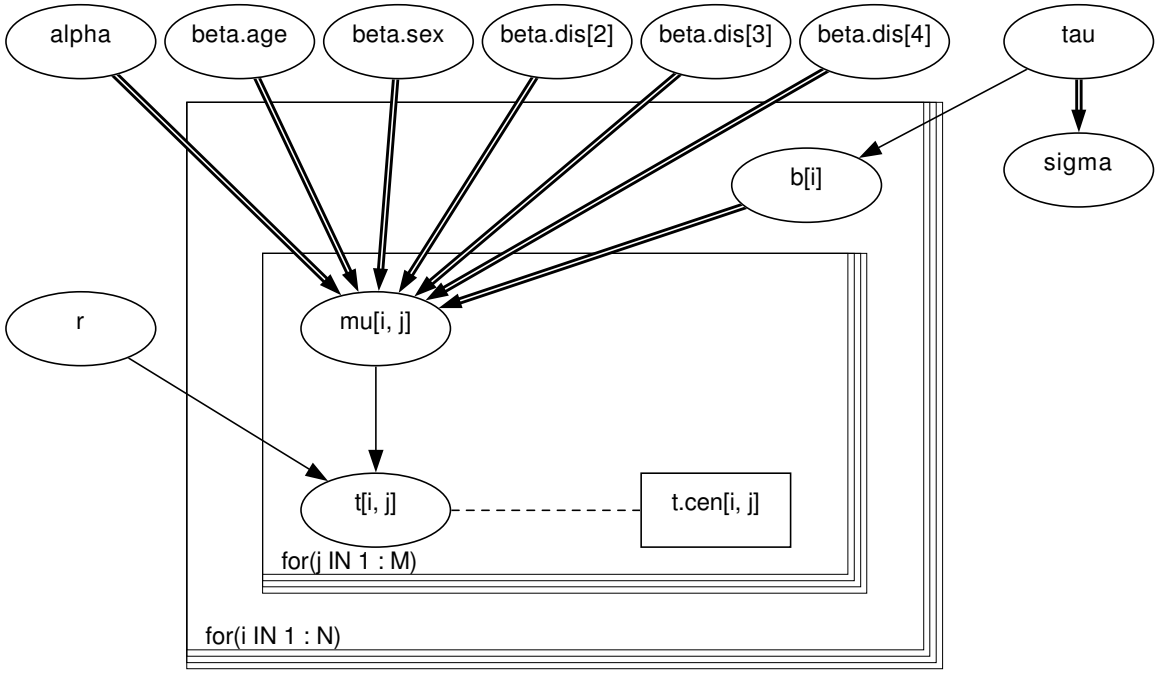
$$b_k \sim \text{Normal}(0, 0.0001)$$

$$\tau \sim \text{Gamma}(0.0001, 0.0001)$$

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 and *BUGS* language are given below.

Graphical model for kidney example:



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

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha	-4.529	0.9036	0.06244	-6.348	-4.473	-2.932	1001	10000
beta.dis[2]	0.1265	0.5679	0.01859	-0.9922	0.1201	1.3	1001	10000
beta.dis[3]	0.5995	0.5781	0.02205	-0.5284	0.5863	1.815	1001	10000
beta.dis[4]	-1.198	0.8483	0.03147	-2.805	-1.206	0.5525	1001	10000
beta.sex	-1.945	0.5019	0.028	-3.054	-1.906	-1.042	1001	10000
r	1.205	0.1711	0.01523	0.9005	1.2	1.541	1001	10000
sigma	0.6367	0.3802	0.03159	0.04092	0.6494	1.366	1001	10000

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,...,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

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

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 $E(dN_i(t) \mid F_{t-})$ corresponds to the probability of subject *i* failing in the interval [t, t+dt). As dt -> 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

$$I_i(t) = Y_i(t)\lambda_0(t)\exp(\boldsymbol{\beta}\mathbf{z}_i)$$

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 $\lambda_0(t)\exp(\boldsymbol{\beta}\mathbf{z}_i)$ is the familiar Cox regression model. Thus we have observed data D = $N_i(t)$, $Y_i(t)$, $\mathbf{z}_i$; i = 1,..n and unknown parameters $\boldsymbol{\beta}$ and $\Lambda_0(t)$ = 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

$$P(\boldsymbol{\beta}, \Lambda_0() \mid D) \sim P(D \mid \boldsymbol{\beta}, \Lambda_0()) P(\boldsymbol{\beta}) P(\Lambda_0())$$

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

n

$$\Pi[\Pi\ I_i(t)^{dN_i(t)}]\ \exp(-\ I_i(t)dt)$$
$$i = 1\ \ t >= 0$$

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 $I_i(t)dt$:

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

We may write

$$I_i(t)dt\ = Y_i(t)\exp(\boldsymbol{\beta z}_i)d\Lambda_0(t)$$

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 $\boldsymbol{\beta}$ 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(Idt[i, j])            # Likelihood
        Idt[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(0.0,0.000001)
}
```

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
S.placebo[1]	0.9282	0.04863	5.004E-4	0.8094	0.9387	0.9907	1001	10000
S.placebo[2]	0.8538	0.06843	7.467E-4	0.6926	0.8639	0.9571	1001	10000
S.placebo[3]	0.8161	0.07561	7.661E-4	0.6422	0.8244	0.9362	1001	10000
S.placebo[4]	0.7432	0.08534	8.86E-4	0.5586	0.7503	0.8892	1001	10000
S.placebo[5]	0.6703	0.09256	9.762E-4	0.4749	0.6755	0.835	1001	10000
S.placebo[6]	0.5633	0.09747	9.302E-4	0.3666	0.5661	0.7477	1001	10000
S.placebo[7]	0.5304	0.09778	9.097E-4	0.338	0.5336	0.7148	1001	10000
S.placebo[8]	0.4142	0.09387	8.073E-4	0.2374	0.4119	0.6037	1001	10000
S.placebo[9]	0.3812	0.09325	8.172E-4	0.2086	0.3779	0.5701	1001	10000
S.placebo[10]	0.32	0.08945	8.307E-4	0.1583	0.315	0.509	1001	10000
S.placebo[11]	0.2583	0.0845	7.771E-4	0.111	0.2511	0.4395	1001	10000
S.placebo[12]	0.2257	0.08105	7.359E-4	0.08703	0.2181	0.402	1001	10000
S.placebo[13]	0.1956	0.07723	7.293E-4	0.06867	0.1873	0.3668	1001	10000
S.placebo[14]	0.1656	0.07326	6.788E-4	0.04889	0.1567	0.3298	1001	10000
S.placebo[15]	0.1398	0.06788	6.183E-4	0.03602	0.1305	0.2953	1001	10000
S.placebo[16]	0.0867	0.05455	5.259E-4	0.01301	0.07663	0.22	1001	10000
S.placebo[17]	0.04445	0.03913	4.092E-4	0.002506	0.03349	0.1484	1001	10000
S.treat[1]	0.983	0.01372	1.541E-4	0.9473	0.9866	0.9982	1001	10000
S.treat[2]	0.9643	0.02175	2.58E-4	0.9115	0.9692	0.9922	1001	10000
S.treat[3]	0.9544	0.02538	3.003E-4	0.8918	0.9598	0.9884	1001	10000
S.treat[4]	0.9343	0.03217	4.071E-4	0.8573	0.9398	0.9797	1001	10000
S.treat[5]	0.9125	0.03913	5.007E-4	0.821	0.9185	0.9701	1001	10000
S.treat[6]	0.8772	0.04896	6.526E-4	0.7654	0.8838	0.9521	1001	10000
S.treat[7]	0.8652	0.05234	6.984E-4	0.745	0.8717	0.947	1001	10000
S.treat[8]	0.8178	0.06456	8.45E-4	0.6736	0.8246	0.9229	1001	10000
S.treat[9]	0.8024	0.06872	9.057E-4	0.6528	0.8099	0.9151	1001	10000
S.treat[10]	0.771	0.07613	9.916E-4	0.6064	0.7786	0.8976	1001	10000
S.treat[11]	0.7339	0.08462	0.001154	0.5522	0.7409	0.8774	1001	10000
S.treat[12]	0.7114	0.08897	0.001204	0.5224	0.7174	0.8659	1001	10000
S.treat[13]	0.6882	0.0932	0.001268	0.4913	0.6937	0.8528	1001	10000
S.treat[14]	0.6619	0.097	0.001318	0.4641	0.6669	0.8355	1001	10000
S.treat[15]	0.636	0.1007	0.00137	0.4318	0.6406	0.8191	1001	10000
S.treat[16]	0.5662	0.111	0.001493	0.3453	0.5688	0.773	1001	10000
S.treat[17]	0.4761	0.1189	0.001548	0.2502	0.4747	0.7085	1001	10000
beta	1.538	0.4176	0.005644	0.7718	1.521	2.384	1001	10000

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

$$I_i(t)\,dt = Y_i(t)\exp(\beta' z_i + b_{pair_i})\,d\Lambda_0(t) \quad i = 1,...,42; \quad pair_i = 1,...,21$$
$$b_{pair_i} \sim \text{Normal}(0, \tau)$$

A non-informative Gamma prior is assumed for τ, the precision of the frailty parameters. Note that the above 'additive' formualtion of the frailty model is equivalent to assuming multiplicative frailties with a log-Normal population distibution. 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(Idt[i, j])
         Idt[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);
      }
      tau ~ dgamma(0.001, 0.001)
      sigma <- sqrt(1 / tau)
      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 (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
beta	-1.607	0.4399	0.009042	-2.507	-1.592	-0.7798	1001	10000
sigma	0.2415	0.2255	0.01543	0.02854	0.1625	0.8548	1001	10000