BUGS 0.5*Examples
Volume 1 (version i)

David Spiegelhalter **Andrew Thomas** **Nicky Best**
Wally Gilks

MRC Biostatistics Unit, Institute of Public Health,
Robinson Way, Cambridge CB2 2SR

Tel: 44-1223-330300 *Fax: 44-1223-330388*
e-mail: bugs@mrc-bsu.cam.ac.uk *ftp: ftp.mrc-bsu.cam.ac.uk*

August 14, 1996

Introduction and Disclaimer

These worked examples illustrate the use of the BUGS language and sampler in a wide range of problems. They contain a number of useful "tricks", but are certainly not exhaustive of the models that may be analysed.

We emphasise that all the results for these examples have been derived in the most naive way: in general a burn-in of 500 iterations and a single long run of 1000 iterations. This is not recommended as a general technique: no tests of convergence have been carried out, and traces of the estimates have not even been plotted. However, comparisons with published results have been made where possible. Times have been measured on a 60 MHz superSPARC: a 60 MHz Pentium PC appears to be about 4 times slower, and a 30 MHz superSPARC about 2 times slower.

Users are warned to be extremely careful about assuming convergence, especially when using complex models including errors in variables, crossed random effects and intrinsic priors in undirected models.

 The support of the Economic and Social Research Council (UK) is gratefully acknowledged. The work was funded in part by ESRC (UK) Award Number H519 25 5023.

Warning

BUGS version 0.5

Release date: August 14, 1996

BUGS version 0.5 released on August 14, 1996 is a TEST version only.

If you encouter any errors in the program, please notify us by e-mailing `bugs@mrc-bs.cam.ac.uk`. In particular, users are warned that BUGS version 0.5 may crash during sampling with the error

```
Can not locate mode of sampling density
```

or

```
Allowed number of function evaluations exceeded for ARS.
```

Such errors typically occur when estimating models involving a log or logit function of parameters whose values are very close to zero. We are currently working to fix this bug, and will release a revised version 0.5 when this has been sorted out. Please note that the `Cosmos` example in BUGS Examples Volume 2 crashes with this error when running BUGS version 0.5, although the model can be run successfully using BUGS version 0.30.

Contents

1 `Rats`: Normal hierarchical models with missing data

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

$$\begin{aligned} Y_{ij} &\sim \text{Normal}(\alpha_i + \beta_i(x_j - \overline{x}), \tau_c) \\ \alpha_i &\sim \text{Normal}(\alpha_c, \tau_\alpha) \\ \beta_i &\sim \text{Normal}(\beta_c, \tau_\beta) \end{aligned}$$

where $\overline{x} = 22$, 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. Interest particularly focusses on the intercept at zero time (birth), denoted $\alpha_0 = \alpha_c - \beta_c\overline{x}$. The appropriate graphical model is shown in Figure 1.

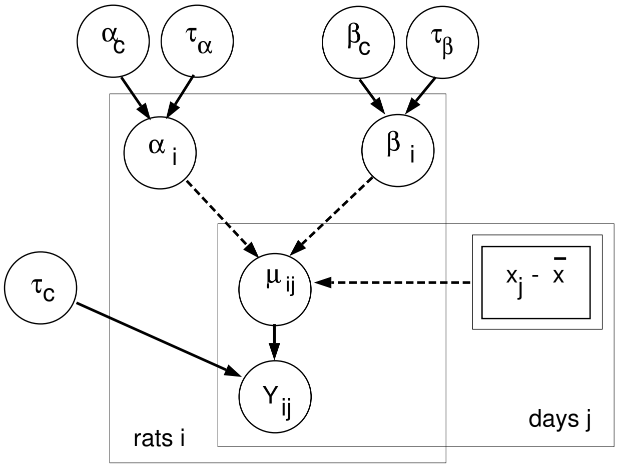

Figure 1: Graphical model for `rats` example

Rats: model specification in BUGS

```
model rats;
const
   N = 30,  # number of rats
   T = 5;   # number of time points
var
   tau.c, alpha0, alpha.c, beta.c, x[T],
   mu[N,T], Y[N,T], alpha[N], beta[N],
   tau.alpha, tau.beta, sigma, x.bar;

data Y in "ratsy.dat", x in "ratsx.dat";
inits in "rats.in";

{
    for (i in 1:N) {
       for (j in 1:T) {
          mu[i,j] <- alpha[i] + beta[i]*(x[j] - x.bar);
          Y[i,j]   ~ dnorm(mu[i,j],tau.c)
       }
       alpha[i] ~ dnorm(alpha.c,tau.alpha);
       beta[i]  ~ dnorm(beta.c,tau.beta);
    }
    alpha.c   ~ dnorm(0,1.0E-4);
    beta.c    ~ dnorm(0,1.0E-4);
    tau.c     ~ dgamma(1.0E-3,1.0E-3);
    tau.alpha ~ dgamma(1.0E-3,1.0E-3);
    tau.beta  ~ dgamma(1.0E-3,1.0E-3);
    sigma    <- 1.0/sqrt(tau.c);
    x.bar    <- mean(x[]);
    alpha0   <- alpha.c - beta.c*x.bar;
}
```

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

If the data are input in rectangular format, 2 files are required. The response data `y` are in file `ratsy.dat`:

```
151 199 246 283 320
145 199 249 293 354
147 214 263 312 328
155 200 237 272 297
.....
.....
157 205 248 289 316
137 180 219 258 291
153 200 244 286 324
```

and the measurements times `x` are in `ratsx.dat`:

```
8.0
15.0
22.0
29.0
36.0
```

Alternatively, the data may be input in S format, as in file `ratsS.dat`. In this case, both `y` and `x` may be included in the same file:

```
list(Y =  c(151.0,199.0,246.0,283.0,320.0,
            145.0,199.0,249.0,293.0,354.0,
            .....
            153.0,200.0,244.0,286.0,324.0),
      x = c(8.0,15.0,22.0,29.0,36.0))
```

and the data statement on line 10 of the `rats.bug` file must be changed to

```
data in "ratsS.dat";
```

Analysis

A naive run, using no diagnostics for convergence, gave the following results for the population intercept α_0 at time 0 and the population gradient β_c.

```
Bugs>update(500)   500    updates took  00:00:02
Bugs>monitor(alpha0)
Bugs>monitor(beta.c)
Bugs>update(1000)   1000   updates took  00:00:04
Bugs>stats(alpha0)
                  mean        sd       2.5% :  97.5%  CI    median      sample
                1.063E+2   3.590E+0   9.968E+1   1.132E+2   1.061E+2     1000
Bugs>stats(beta.c)
                  mean        sd       2.5% :  97.5%  CI    median      sample
                6.183E+0   1.095E-1   5.968E+0   6.393E+0   6.179E+0     1000
```

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 called `ratsmiss.dat`, and is obtained by simply replacing data values by `NA` (see below). The `rats.bug` file only has to change the `data` declaration to `data Y in "ratsmiss.dat";` we note that this is the only change necessary, since the distinction between observed and unobserved quantities is made in the data file and not the model specification.

Data file "ratsmiss.dat"

```
151 199 246 283 320
145 199 249 293 354
.....
153  NA  NA  NA  NA
```

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 is a sample run.

```
Bugs>update(500)   500    updates took  00:00:02
Bugs>monitor(beta.c)
Bugs>monitor(Y[26,])
Bugs>update(1000)   1000   updates took  00:00:04
Bugs>stats(beta.c)
                  mean        sd       2.5% :  97.5%  CI    median      sample
                6.537E+0   1.411E-1   6.260E+0   6.811E+0   6.533E+0     1000
Bugs>stats(Y[26,])
                  mean        sd       2.5% :  97.5%  CI    median      sample
[26,2]          2.044E+2   8.937E+0   1.865E+2   2.212E+2   2.046E+2     1000
[26,3]          2.497E+2   1.076E+1   2.294E+2   2.706E+2   2.493E+2     1000
[26,4]          2.952E+2   1.280E+1   2.700E+2   3.216E+2   2.949E+2     1000
[26,5]          3.413E+2   1.603E+1   3.115E+2   3.742E+2   3.401E+2     1000
```

We note that our estimate 6.54 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.

2 Pump: 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

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

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 is shown below.

Pump	1	2	3	4	5	6	7	8	9	10
t_i	94.3	15.7	62.9	126	5.24	31.4	1.05	1.05	2.1	10.5
x_i	5	1	5	14	3	19	1	1	4	22

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

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

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.

Figure 2 shows the graph corresponding to the above model, and the associated BUGS analysis is given below.

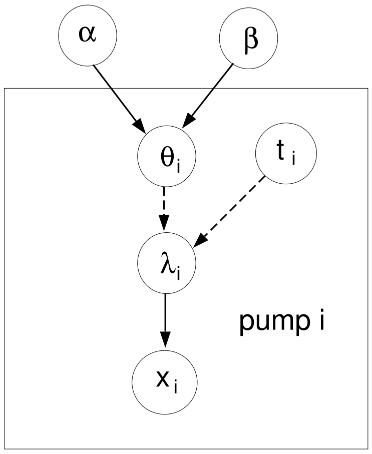

Figure 2: Graphical model for pump example.

Model specification for pump example

```
model pump;
const
   N = 10;  # number of pumps
var
   theta[N],     # failure rate of each pump
   x[N],         # number of failures per pump
   t[N],         # length of operation time
   alpha,beta,   # parameters of gamma prior
   lambda[N];    # theta[]*t[]

data t, x in "pump.dat";
inits in "pump.in";

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

   alpha ~ dexp(1.0);
   beta  ~ dgamma(0.1,1.0);

}
```

Analysis A BUGS run of 1000 iterations took 2 seconds after a 500 iteration burn-in. Posterior mean estimates for selected parameters are listed below, together with the corresponding estimates obtained by George *et al.* (1993) (denoted *GM&S* estimate).

variable	BUGS estimate (95% interval)	*GM&S* estimate
θ_1	0.06 (0.02, 0.12)	0.06
θ_2	0.10 (0.01, 0.30)	0.10
θ_9	1.58 (0.47, 3.39)	1.59
θ_{10}	1.97 (1.24, 2.93)	1.99
α	0.73 (0.28, 1.38)	0.70
β	0.98 (0.24, 2.36)	0.90

3 `Seeds`: random effects 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 2×2 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 ith plate, $i = 1, \ldots, N$. 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	.26	5	6	.83	8	16	.50	3	12	.25
23	62	.37	53	74	.72	10	30	.33	22	41	.54
23	81	.28	55	72	.76	8	28	.29	15	30	.50
26	51	.51	32	51	.63	23	45	.51	32	51	.63
17	39	.44	46	79	.58	0	4	.00	3	7	.43
			10	13	.77						

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

$$\begin{aligned} r_i &\sim \text{Binomial}(p_i, n_i) \\ \text{logit}(p_i) &= \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_{12} x_{1i} x_{2i} + b_i \\ b_i &\sim \text{Normal}(0, \tau). \end{aligned}$$

where x_{1i}, x_{2i} are the seed type and root extract of the ith plate, and an interaction term $\alpha_{12} x_1 x_2$ is included. $\alpha_0, \alpha_1, \alpha_2, \alpha_{12}, \tau$ are given independent "noninformative" priors. The graphical model is shown in Figure 3.

The deviance for this model may be calculated within `BUGS` as a logical node. The log-likelihood for an observation r_i arising from a binomial model with denominator n_i and success probability p_i is

$$\text{llike}_i = r_i \log(p_i) + (n_i - r_i) \log(1 - p_i)$$

The saturated log likelihood for the binomial model is

$$\text{llike.sat}_i = r_i \log(\frac{r_i}{n_i}) + (n_i - r_i) \log(1 - \frac{r_i}{n_i})$$

Hence the deviance is given by $2(\sum_i \text{llike.sat}_i - \sum_i \text{llike}_i)$.

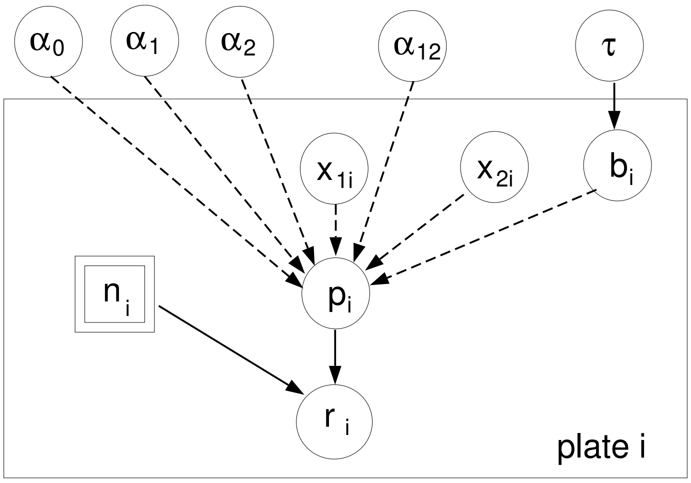

Figure 3: Graphical model for the seeds example

Model specification for the seeds example

```
model seeds;
const
   N = 21;  # number of samples
var
   alpha0, alpha1, alpha2, alpha12, tau, sigma,
   x1[N], x2[N], p[N], r[N], n[N], b[N],
   llike[N], llike.sat[N], deviance;

data r,n,x1,x2 in "seeds.dat";
inits in "seeds.in";
{
   alpha0  ~ dnorm(0.0,1.0E-6);  # intercept
   alpha1  ~ dnorm(0.0,1.0E-6);  # seed coeff
   alpha2  ~ dnorm(0.0,1.0E-6);  # extract coeff
   alpha12 ~ dnorm(0.0,1.0E-6);
   tau     ~ dgamma(1.0E-3,1.0E-3);    # 1/sigma^2
   sigma  <- 1.0/sqrt(tau);
   for (i in 1:N) {
      b[i]         ~ dnorm(0.0,tau);
      logit(p[i]) <- alpha0 + alpha1*x1[i] + alpha2*x2[i] +
                     alpha12*x1[i]*x2[i] + b[i];
      r[i]         ~ dbin(p[i],n[i]);
      # log likelihood for sample i:
      llike[i]     <- r[i]*log(p[i]) + (n[i]-r[i])*log(1-p[i]);
      # log likelihood for saturated model:
      llike.sat[i] <- r[i]*log(r[i]/n[i]) + (n[i]-r[i])*log(1-r[i]/n[i]);
   }
   deviance <- 2 * (sum(llike.sat[]) - sum(llike[]));
}
```

Initial values in S object format

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

Analysis

We may compare simple logistic, maximum likelihood (from EGRET), penalized quasi-likelihood (PQL) (Breslow and Clayton, 1993) and `BUGS` results, using a burn-in of 500 iterations and estimation based on 1000 samples.

variable	Logistic regression $\beta \pm SE$	maximum likelihood $\beta \pm SE$	PQL $\beta \pm SE$	`BUGS` $\beta \pm SE$
constant (α_0)	-.558 ± .126	-.548 ± .167	-.542 ± .190	-.542 ±.178
seed (α_1)	.146 ± .223	.097 ± .278	.077 ± .308	.028 ±.340
extract (α_2)	1.318 ± .177	1.337 ± .237	1.339 ± .270	1.368±.253
interaction (α_{12})	-.778 ± .306	-.811 ± .385	-.825 ± .430	-.792±.426
scale (σ)	—	.236 ± .110	.313 ± .121	.292±.152

`BUGS` produces samples for the deviance just like any other node. Hence we obtain a *distribution* for the deviance as shown in Figure 4 (see also the section "Model criticism and selection" in the `BUGS` manual 0.50).

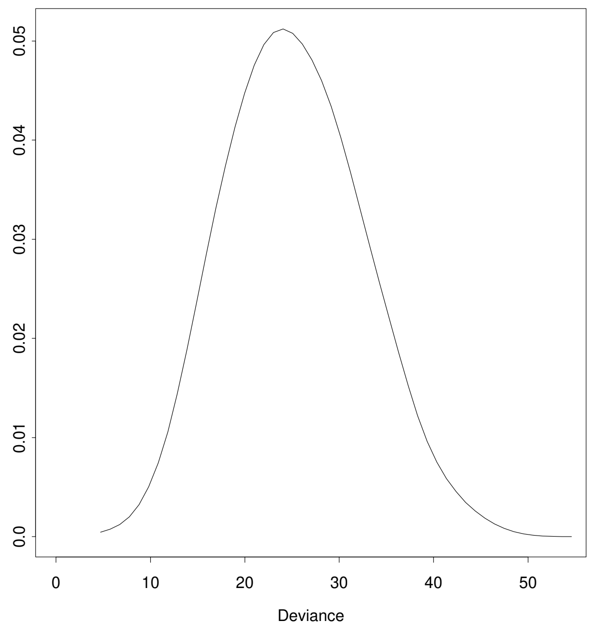

Figure 4: Posterior distribution of the deviance for the `seeds` example

3.1 Constraining random effects to sum to zero

It is possible to impose a constraint that random effects add to zero, which may be useful when, for example, using intrinsic priors in which the prior mean is not specified. Alternatively, one might be interested in estimating fixed effect coefficients for the particular individuals under study, rather than coefficients for the population from which it is assumed the individuals are drawn

The model is achieved by creating additional independent random effects c_i and setting $b_i = c_i - \overline{c}$, so that the marginal prior distributions are $b_i \sim \text{Normal}(0, \frac{N}{N-1}\tau)$ and the b's sum to zero. The model is contained in `seedszro.bug`, and its essentials are shown below.

```
alpha0 ~ dnorm(0.0,1.0E-6);  # intercept
....
tau ~ dgamma(1.0E-3,1.0E-3);    # 1/sigma^2
for (i in 1:N) {
   c[i] ~ dnorm(0.0,tau);
   b[i] <- c[i] - mean(c[]);    #  make sure b's add to zero
   logit(p[i]) <- alpha0 + alpha1*x1[i] + alpha2*x2[i] +
                  alpha12*x1[i]*x2[i] + b[i];
   r[i] ~ dbin(p[i],n[i]);
   ....
}
```

This slows down the sampling somewhat: after a 500 iteration burn-in, 1000 iterations took $2\frac{1}{2}$ minutes, with results $\hat{\alpha_0} = -.565 \pm .177, \hat{\alpha_1} = .093 \pm .302, \hat{\alpha_2} = 1.370 \pm .272, \hat{\alpha_{12}} = -.861 \pm .455, \hat{\sigma} = .282 \pm .158$.

3.2 An alternative parameterisation for the precision of the random effects — the unobserved covariate model

It is possible to parameterise the unknown variability between the random effects directly in the linear predictor, while the random effects have a completely specified distribution. The random effect then appears as an unobserved covariate, with its coefficient σ being the standard deviation of the random effects. In this case, we obtain the model

$$
\begin{aligned}
r_i &\sim \text{Binomial}(p_i, n_i) \\
\text{logit}(p_i) &= \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_{12} x_{1i} x_{2i} + \sigma b_i \\
b_i &\sim \text{Normal}(0, 1).
\end{aligned}
$$

If we gave σ the "non-informative" prior distribution $p(\sigma) \approx 1/\sigma$ (approximately equivalent to a $\Gamma(\epsilon, \epsilon)$ distribution with mean 1 and variance $1/\epsilon$), then we would have essentially the same model as when parameterised in terms of the b's having precision τ, and τ given a non-informative prior $p(\tau) \approx 1/\tau$. However, $p(\sigma) \approx 1/\sigma$ is not log-concave in σ (Gilks and Wild, 1992), and so currently cannot be implemented in BUGS unless there is a conjugate likelihood (note that the prior $p(\tau) \approx 1/\tau$ *is* used extensively in these examples, but always when there are conjugate normal-likelihood terms).

We may however, use the exponential prior $p(\sigma) = e^{-\sigma}$. We would expect, however, some tendency of the estimate of σ to be pulled towards 1 compared with the original analysis in terms of τ. File `seedssig.bug` contains this model, whose essentials are shown below.

```
sigma ~ dexp(1.0);
for (i in 1:N) {
   b[i] ~ dnorm(0.0,1.0);
   logit(p[i]) <- alpha0 + alpha1*x1[i] + alpha2*x2[i] +
                  alpha12*x1[i]*x2[i] + sigma * b[i];
   r[i] ~ dbin(p[i],n[i]);
}
```

Running this model for 1000 iterations after a 500 burn-in produced estimates of $\hat{\alpha_0} = -.526 \pm .213, \hat{\alpha_1} = .016 \pm .356, \hat{\alpha_2} = 1.318 \pm .302, \hat{\alpha_{12}} = -.777 \pm .472, \hat{\sigma} = .340 \pm .151$. We note that the expected inflation of σ towards 1.

3.3 A uniform prior for the standard deviation of the random effects

An alternative non-informative prior distribution for σ is to let $p(\sigma)$ be locally uniform on a range $(0, r)$. This may be achieved in BUGS using 2 different parameterizations: (i) the unobserved covariate model (§3.2) with $\sigma \sim$ Uniform$(0, r)$ instead of $\sigma \sim$ Gamma$(1, 1)$; (ii) the original model parameterized in terms of the b's having precision τ, but giving τ a Pareto$(\frac{1}{2}, r^{-2})$ prior instead of a Gamma$(0.001, 0.001)$ prior (see section on "Non-informative priors" in the BUGS manual 0.50). The BUGS code for the above models (with r=10) may be found in `seedsuni.bug` and `seedspar.bug` respectively. The results of a 5000 iteration BUGS run following a 1000 iteration burn-in are shown below.

variable	Uniform(0,10) on σ $\beta \pm SE$	Pareto($\frac{1}{2}$, 0.01) on τ $\beta \pm SE$
constant (α_0)	-.563 ± .187	-.552 ±.211
seed (α_1)	.114 ± .286	.081 ± .356
extract (α_2)	1.393 ± .289	1.428 ± .319
interaction (α_{12})	-.912 ± .438	-.940 ± .519
scale (σ)	.364 ± .161	.377 ±.146

4 Surgical: institutional ranking

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

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

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

$$r_i \sim \text{Binomial}(p_i, n_i), \qquad \text{i = 1,...,12}$$

4.1 Fixed effects model

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:

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

The BUGS code is given below.

BUGS code for fixed effects model

```
model surg.fix;
const
   N = 12;    # number of hospitals
var
    r[N],     # number of deaths
    n[N],     # total number of operations
    p[N];     # 'true' probability of death

data r, n in "surgical.dat";
inits in "surgical.in";

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

4.2 Random effects model

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:

$$\begin{aligned} \text{logit}(p_i) &= b_i \\ b_i &\sim \text{Normal}(\mu, \tau) \end{aligned}$$

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

Figure 5 shows the graph corresponding to the above model, and the essentials of the `BUGS` code are given below.

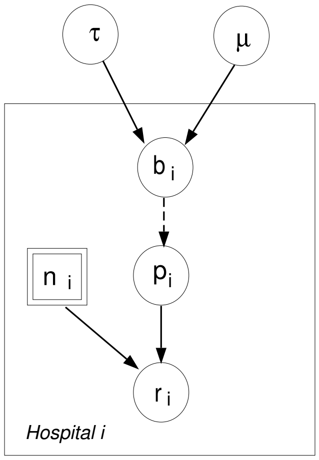

Figure 5: Graphical model for the random effects `surgical` example

`BUGS` code for random effects model

```
for (i in 1:N) {
  r[i] ~ dbin(p[i], n[i]);
  logit(p[i]) <- b[i];
  b[i] ~ dnorm(mu, tau);
}
# Priors:
  mu ~ dnorm(0.0,1.0E-6);
  pop.mean <- exp(mu)/(1+exp(mu));  # population mean on natural scale
  tau ~ dgamma(1.0E-3, 1.0E-3); sigma <- 1.0/sqrt(tau);
```

Analysis & Results

After a 500 iteration burn-in, a `BUGS` run of 1000 iterations took 1 second for the fixed effects model, and 7 seconds for the random effects model. Figure 6 shows the posterior mean and 95% credible interval for the estimated surgical mortality rate in each hospital for both the fixed and random effect models.

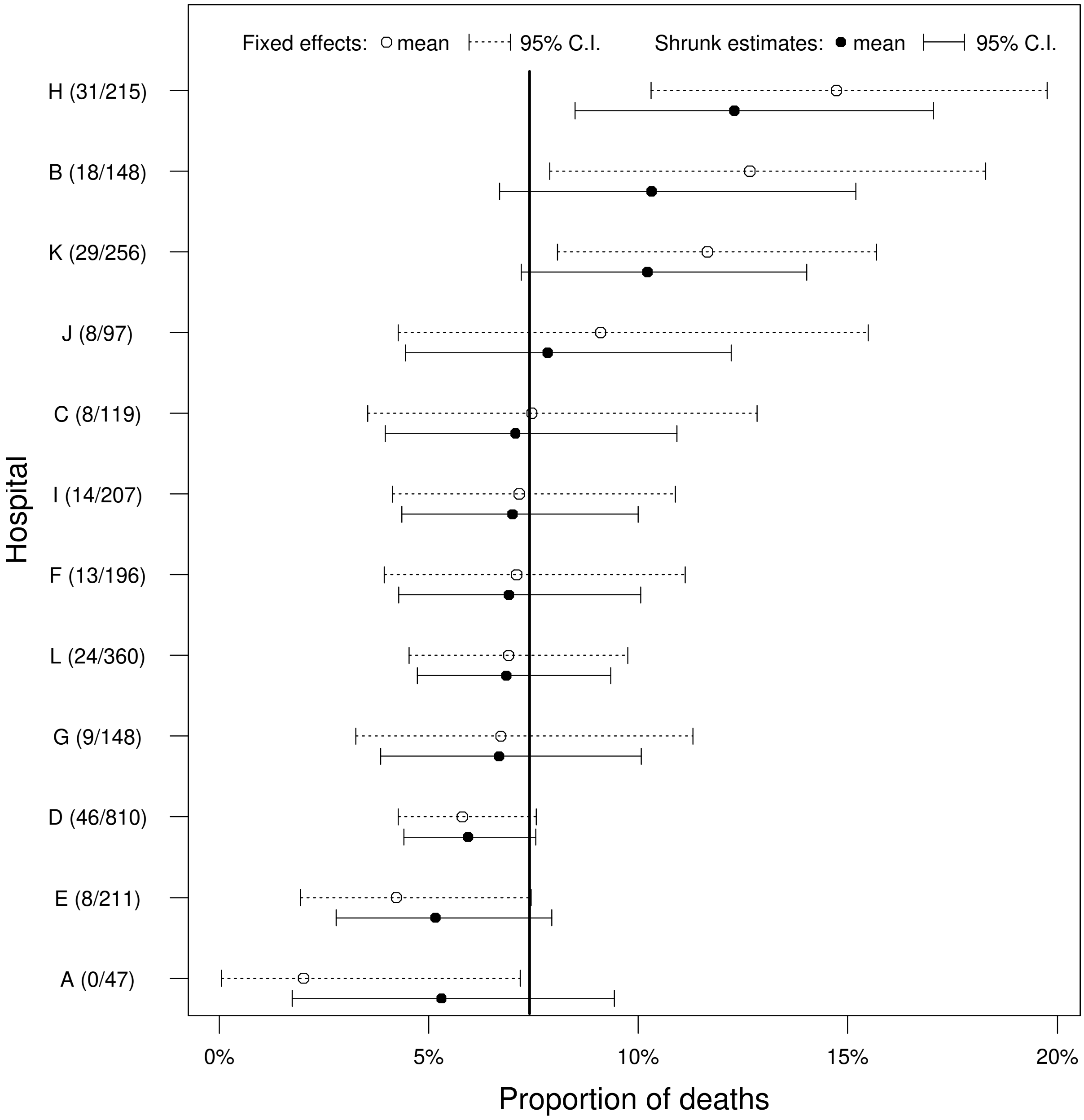

Figure 6: Fixed and shrunk estiamtes of the surgical mortality rates in each hospital. Numbers in brackets show the observed nuber of deaths and the total number of operations. The vertical line at $p = 7.3\%$ indicates the population mean failure rate (`pop.mean`) estimated from the random effects model

4.3 Ranking each hospital

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 which may be summarised to provide an estimate of the mean or median rank for each hospital, plus a 95% credible interval. The latter captures the (typically large) uncertainty associated with the rank position of each hospital.

We compute the ranks in BUGS using the `step` function as follows

```
for (i in 1:N) {
  for (j in 1:N) {
   not.less.than[i,j] <- step(p[i] - p[j]);
  }
  rank[i] <- sum(not.less.than[i,]);
}
```

where `step(x)` $= 1$ if `x` ≥ 0 and 0 otherwise. The ith row of the array `not.less.than[]` thus contains a 1 in columns corresponding to hospitals with an equal or lower estimated failure probability than hospital i, and zeros elsewhere. Summing this row yields the total number of hospitals who have a 'better' (lower) failure rate than hospital i, and thus correpsonds to that hospital's rank.

Results

Figure 7 shows the posterior mean and 95% credible interval for the estimated surgical mortality rate in each hospital for both the fixed and random effect models. These interval estimates 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.

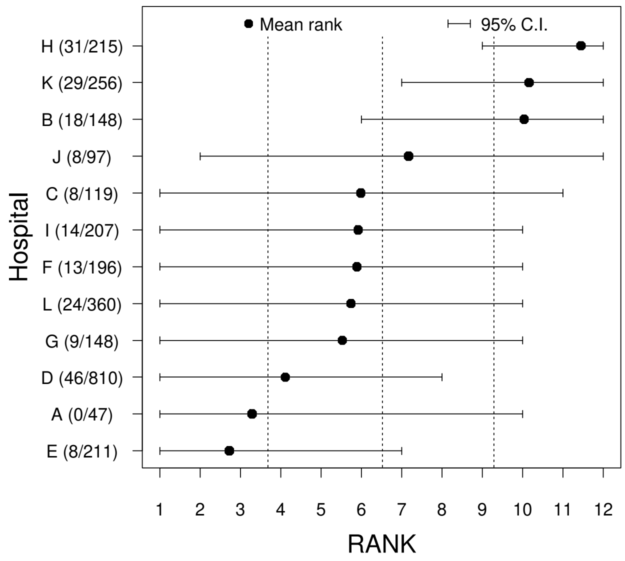

Figure 7: Posterior means and 95% credible intervals for the rank of each hospital. Vertical dashed lines indicate the position of lower and upper quartiles and median rank

5 `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	60	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

$$\begin{aligned} y_{ij} &\sim \text{Poisson}(\mu_{ij}) \\ \log \mu_{ij} &= \alpha + \beta \log(x_i + 10) + \gamma x_i + \lambda_{ij} \\ \lambda_{ij} &\sim \text{Normal}(0, \tau). \end{aligned}$$

$\alpha, \beta, \gamma, \tau$ are given independent "noninformative" priors. The appropriate graph is shown in Figure 8.

As for the `Seeds` example, we may calculate the deviance as a deterministic node in `BUGS`. The log-likelihood for an observation y_{ij} arising from a Poisson model with mean μ_{ij} is

$$\text{llike}_{ij} = y_{ij} \log(\mu_{ij}) - \mu_{ij} + \text{constant}$$

The saturated log-likelihood for the Poisson model is

$$\text{llike.sat}_{ij} = y_{ij} \log(y_{ij}) - y_{ij} + \text{constant}$$

and the deviance is given by $2(\sum_{ij} \text{llike.sat}_{ij} - \sum_{ij} \text{llike}_{ij})$.

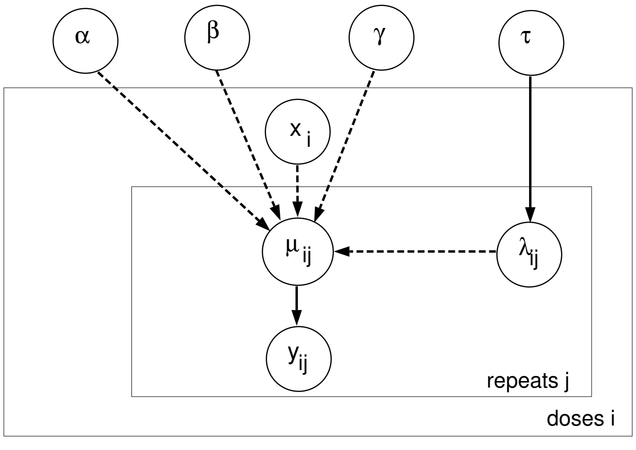

Figure 8: Graphical model for `salm` example

Salm: model specification in BUGS

Note that each covariate has been centred about its mean in the BUGS code. This greatly improves the stability and convergence of the simulations.

```
model salm;
const
   doses = 6, plates = 3;
var
   alpha, alpha.star, beta, gamma, mu[doses,plates], y[doses,plates],
   lambda[doses,plates], sigma, tau, x[doses], logx[doses],
   llike[doses,plates], llike.sat[doses,plates], deviance;
data y, x in "salm.dat";
inits in "salm.in";
{
   alpha.star ~ dnorm(0.0,1.0E-4);     # intercept
   beta ~ dnorm(0.0,1.0E-4);           # mutagenic effect
   gamma ~ dnorm(0.0,1.0E-10);         # toxic effect
   tau ~ dgamma(1.0E-3,1.0E-3);        # Gamma prior on precision
   sigma <- 1.0/sqrt(tau);
   for(i in 1:doses){
      for(j in 1:plates){
         log(mu[i,j])   <- alpha.star + beta*(logx[i]-mean(logx[]))
                             + gamma*(x[i]-mean(x[])) + lambda[i,j];
         y[i,j]          ~ dpois(mu[i,j]);
         lambda[i,j]     ~ dnorm(0.0,tau);
         llike[i,j]     <- y[i,j]*log(mu[i,j]) - mu[i,j];
         llike.sat[i,j] <- y[i,j]*log(y[i,j]) - y[i,j];
     }
     logx[i]        <- log(x[i]+10);
   }
   alpha <- alpha.star - beta*mean(logx[]) - gamma*mean(x[]);
   deviance <- 2 * (sum(llike.sat[,]) - sum(llike[,]));
}
```

Analysis

1000 iterations took 12 seconds after a 500 iteration burn-in. The resulting parameter estimates and standard errors can be compared with those of Breslow (1984) using a quasi-likelihood approach. Also shown below are the results of re-parameterizing the random-effects precision (τ) in terms of a Pareto (0.5, 0.04) prior, which is equivalent to assuming a uniform prior on (0, 5) for σ.

	Quasi-likelihood	BUGS (Gamma prior on τ)	BUGS (Pareto prior on τ)
α	$2.203 \pm .364$	$2.201 \pm .392$	$2.189 \pm .410$
β	$.311 \pm .099$	$.292 \pm .141$	$.311 \pm .108$
γ	$-.000974 \pm .000437$	$-.000911 \pm .000615$	$-.000999 \pm .000450$
σ	$.268$	$.266 \pm .088$	$.286 \pm .083$
Deviance	–	18.89 ± 6.26	18.02 ± 5.90

6 Equiv: bioequivalence and missing data 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 ith subject ($i = 1, ..., 10$) in the kth period ($k = 1, 2$) is assumed to be of the form

$$\begin{aligned} Y_{ik} &\sim \mathrm{N}(m_{ik}, \tau_1) \\ m_{ik} &= \mu + (-1)^{T_{ik}-1}\frac{\phi}{2} + (-1)^{k-1}\frac{\pi}{2} + \delta_i \\ \delta_i &\sim \mathrm{N}(0, \tau_2) \end{aligned}$$

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 is shown in Figure 9.

Two methods of analysis are shown. The first exploits the transformation used by Gelfand *et al.* (1990) which essentially integrates out the random effects:

$$\begin{aligned} \frac{Y_{i1} + Y_{i2}}{2} &\sim \mathrm{N}(\mu, 2\tau_3) \\ \frac{Y_{i1} - Y_{i2}}{2} &\sim \mathrm{N}(\frac{\pi + seq\ \phi}{2}, 2\tau_1) \end{aligned}$$

where $seq = 1$ for sequence AB, -1 for sequence BA, and $\tau_3^{-1} = \tau_1^{-1} + 2\tau_2^{-1}$. The BUGS code is given on the next page. The fixed effects μ, π and ϕ are given vague normal priors, and the precisions are all given gamma(.001,.001) with the constraint that $\tau_3 \leq \tau_1$.

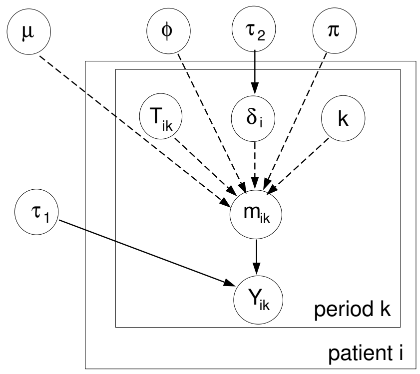

Figure 9: Graphical model for `equiv` example

Model specification for `equiv` **example**

```
model equiv;

const
  N = 10,          # number of patients
  P = 2 ;          # number of periods

 var
  Y[N,P],          # response for patient i in period k
  m[N,P],          # expected response for patient i in period k
  T[N,P],          # treatment for patient i in period k
  seq[N],          # sequence (1=AB; -1=BA)
  Y.mean[N],       # individual means
  Y.diff[N],       # individual differences
  mu.diff[N],      # expectations of individual differences
  mu,              # overall mean
  phi,             # treatment effect (log scale)
  theta,           # treatment effect
  equivalence,     # 1 if effect between 0.8 and 1.2
  pi,              # period effect
  d[N],            # subject random effect
  tau1,tau2,tau3,  # precisions
  sigma1,sigma2,   # s.d.
  sigma3;

  data seq, T, Y in "equiv.dat";
# data seq, T, Y in "equivmiss.dat";
inits in "equiv.in";

{

# Transformed model
#
#  for (i in 1:N) {
#     Y.mean[i] <- mean(Y[i,]);
#     Y.diff[i] <- (Y[i,1] - Y[i,2])/2;
#     Y.diff[i] ~ dnorm(mu.diff[i], tau1);
#     mu.diff[i] <- .5 * pi + seq[i] * phi /2;
#     Y.mean[i] ~ dnorm(mu, tau3);
#  }
#  tau1 ~ dgamma(0.001, 0.001)I(tau3,);
#  sigma1 <- sqrt(2/tau1);
#  tau3 ~ dgamma(0.001, 0.001)I(,tau1);
#  sigma3 <- sqrt(2/tau3);
#  sigma2 <- sqrt(1/tau3 - 1/tau1);
```

```
# Original model
  for (i in 1:N) {
     d[i] ~ dnorm(0,tau2);  # Subject random effect
     for (k in 1:P){
        Y[i,k] ~ dnorm(m[i,k], tau1);
        m[i,k] <- mu + pow(-1, T[i,k]-1)* phi /2 +
                      pow(-1, k-1)* pi /2 + d[i]
     }
  }
  tau1 ~ dgamma(0.001, 0.001);  sigma1 <- sqrt(1/tau1);
  tau2 ~ dgamma(0.001, 0.001);  sigma2 <- sqrt(1/tau2);

  pi ~ dnorm(0, 1.0E-06);
  phi ~ dnorm(0, 1.0E-06);
  mu ~ dnorm(0, 1.0E-06);
  theta <- exp(phi);
# 1 if  0.8 < theta < 1.2
  equivalence <- step(theta - 0.8) - step(theta - 1.2);
}
```

We note the use of initial transformations, the symmetric use of the `I(tau[3],)` construction to specify the inequality between τ_3 and τ_1. We also use the `step` function to indicate whether $\theta = e^{\phi}$ lies between .8 and 1.2, which traditionally determines bioequivalence.

Gelfand *et al.* (1990) also report the effect of removing observations Y_{11}, Y_{32}, Y_{62} from the data. This is easily achieved by substituting `NA` for their values in the data file.

Analysis

5000 iterations took between 3 and 15 seconds after a 500 iteration burn-in. The results are shown below.

Quantity	Transformed	(S.E.)	Original	(S.E.)	Missing	(S.E.)
$\theta = e^{\phi}$	.99	(0.05)	.99	(0.05)	.98	(.07)
$P(.8 \leq \theta \leq 1.2)$	.999	–	.999	–	.989	–
σ_1	.10	(.03)	.11	(.03)	.14	(.04)
σ_2	.15	(.05)	.14	(0.05)	.13	(.06)
Y_{11} (1.40)					1.38	(0.19)
Y_{32} (1.58)					1.57	(0.18)
Y_{62} (1.31)					1.43	(0.20)

We note that our conclusions, that the treatments are extremely similar, are substantially different from those of Gelfand *et al.* (1990) who concluded that the probability of bioequivalence was small. However, we are inclined to believe our results in view of the fact that the average responses under the two treatments are very close (1.432 and 1.440), and our predictions of the missing data points are better than those shown in Gelfand *et al.* (1990).

7 `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.

		Yield (in grams)				
	1	1545	1440	1440	1520	1580
	2	1540	1555	1490	1560	1495
Batch	*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:

$$
\begin{aligned}
y_{ij} &\sim \text{Normal}(\mu_i, \tau_{within}) \\
\mu_i &\sim \text{Normal}(\theta, \tau_{between})
\end{aligned}
$$

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 $\sigma^2_{between}$. 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 $\tau_{between}$.

The graph for this model is shown in Figure 10 and the essentials of the `BUGS` code are given below.

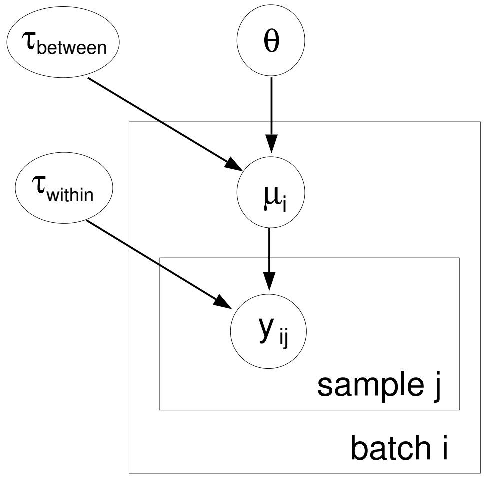

Figure 10: Graphical model for the `dyes` example

Dyes: model specification in BUGS

```
for (i in 1:BATCHES) {
   for (j in 1:SAMPLES) {
      y[i,j] ~ dnorm(mu[i], tau.within);
   }
   mu[i] ~ dnorm(theta, tau.between);
}

theta ~ dnorm(0.0, 1.0E-10);
tau.within ~ dgamma(0.001, 0.001);  sigma2.within <- 1/tau.within
tau.between ~ dgamma(0.001, 0.001);  sigma2.between <- 1/tau.between

sigma2.total <- sigma2.within + sigma2.between;
f.within <- sigma2.within/sigma2.total;
f.between <- sigma2.between/sigma2.total;
```

Note that the above model formulation uses the concept of hierarchical centering (Gelfand *et al.*, 1995) (see Section 'Parameterisation' in the BUGS manual version 0.5). Box and Tiao use a different parameterisation given below

$$\begin{aligned} y_{ij} &= \theta + b_i + w_{ij} \\ \mathrm{Var}(w_{ij}) &= \sigma^2_{within} \\ \mathrm{Var}(b_i) &= \sigma^2_{between} \end{aligned}$$

A literal translation of this parameterisation into the BUGS language would lead to the following declaration

```
y[i,j] <- theta + b[i] + w[i,j];
```

However, it does not make sense to declare `y[i,j]` as a deterministic node in BUGS since the values of `y[i,j]` are already known and are read in from the data file. Instead, we could use the following representation:

```
for (i in 1:BATCHES) {
   for (j in 1:SAMPLES) {
      y[i,j] ~ dnorm(mu[i], tau.within);
    }
   mu[i] <- theta + b[i];
   b[i]   ~ dnorm(0.0, tau.between);
}
```

The above model formulation should yield parameter estimates equivalent to those obtained using the hierarchically centered model, and is generally the form used in our examples.

Analysis

After a 5000 iteration burn-in, a further 50000 iterations took 19 seconds. (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). The posterior means, medians and 95% intervals for selected quantities are shown below. 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. Also shown are the Box and Tiao (B&T) estimates of σ^2_{within} and $\sigma^2_{between}$ obtained 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.

	BUGS			B&T
	mean	**median**	**95% C.I.**	**estimate**
θ	1528	1528	(1483,1572)	–
σ^2_{within}	2991	2760	(1551, 5736)	2451
$\sigma^2_{between}$	2376	1393	(0.013, 10694)	1764
$f_{between}$	0.35	0.35	(0.00, 0.82)	–

We note that there is minimal information in the data concerning $\sigma^2_{between}$, and hence there will be considerable sensitivity to the prior chosen.

8 Stacks: robust and ridge 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

$$\begin{aligned}
\mu_i &= \beta_0 + \beta_1 z_{i1} + \beta_2 z_{i2} + \beta_3 z_{i3} \\
y_i &\sim \text{Normal}(\mu_i, \tau) \\
y_i &\sim \text{Double exp}(\mu_i, \tau) \\
y_i &\sim \text{Logistic}(\mu_i, \tau) \\
y_i &\sim \text{t}(\mu_i, \tau, d)
\end{aligned}$$

where $z_{ij} = (x_{ij} - \overline{x}_{.j})/sd(x_{.j})$ 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

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

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 appropriate graph for the ridge regression model is shown in Figure 11.

The following BUGS code will fit all the necessary models by changing the lines that are commented out: the version shown here fits a ridge regression for logistic errors.

Stacks: model specification in BUGS

```
model stacks;
const
   p = 3,   # number of covariates
   N = 21,  # number of observations
   PI = 3.141593;
var
   x[N,p],        #  raw covariates
   z[N,p] ,       # standardised covariates
   Y[N],mu[N],    # data and expectations
   stres[N],      # standardised residuals
   outlier[N],    # indicator if |stan res|  > 2.5
   beta0,beta[p], # standardised intercept, coefficients
   b0,b[p],       # unstandardised intercept, coefficients
   phi,           # prior precision of standardised coefficeints
   tau,sigma,d;   # precision, sd and degrees of freedom of t distn
data Y,x in  "stacks.dat";
inits in "stacks.in";
{
# 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] ~ dlogis(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,.00001);
  for (j in 1:p) {
#     beta[j] ~ dnorm(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-3,1.0E-3);
# standard deviation of error distribution
# sigma <- sqrt(1/tau);             # normal errors
# sigma <- sqrt(2)/tau;             # double exponential errors
  sigma <- sqrt(pow(PI,2)/3)/tau ;  # logistic errors
# sigma <- sqrt(d/(tau*(d-2)));     # t errors on d degrees of freedom
}
```

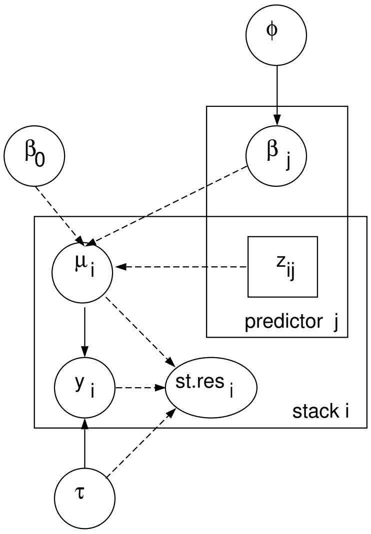

Figure 11: Graphical model for `stacks` example

A BUGS run of 500 burn-in + 1000 iterations took between 4 and 14 secs for each model and gave the following output, including observations with probability greater than 0.1 of being outliers. Included are the results of Birkes and Dodge (1993) (B&D) for least squares, least absolute deviation (LAD) and ridge regression.

	b_0 *s.d.*	b_1 *s.d.*	b_2 *s.d.*	b_3 *s.d.*	σ *s.d.*	outliers
(B&D)						
Least squares	-39.9	.72	1.30	-.15	3.24	
LAD	-39.7	.83	.57	-.06	2.17	
Ridge	-40.6	.69	1.31	-.13		
Independence						
Normal	-39.5	.71	1.32	-.16	3.37	#21
	12.4	*.13*	*.36*	*.16*	*.62*	
D Exp	-38.8	.83	.79	-.12	3.44	#1,3,4,21
	9.3	*.14*	*.35*	*.13*	*.85*	
Logistic	-40.0	.79	1.04	-.14	3.35	#3,4,21
	10.8	*.15*	*.39*	*.15*	*.69*	
t_4	-40.5	.84	.85	-.12	3.47	#4,21
	10.2	*.15*	*.38*	*.13*	*.81*	
Ridge						
Normal	-41.0	.67	1.34	-.12	3.41	#21
	12.6	*.15*	*.38*	*.17*	*.65*	
D Exp	-38.8	.83	.79	-.12	3.49	#1,3,4,21
	9.2	*.14*	*.35*	*.13*	*.88*	
Logistic	-40.0	.79	1.04	-.14	3.34	#1,3,4,21
	11.6	*.15*	*.39*	*.16*	*.76*	
t_4	-40.5	.84	.85	-.12	3.47	#4,21
	9.7	*.14*	*.35*	*.12*	*.81*	

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

9 `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 therpay 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.

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
......							
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 models *II* and *III* of Breslow and Clayton (1993), in which Base is transformed to log(Base/4) and Age to log(Age), and a Treatment by log(Base/4) interaction is included. Also present are random effects for both individual subjects ($b1_j$) and also subject×visit random effects (b_{jk}) to model extra-Poisson variability within subjects. V4 is an indicator variable for the 4th visit. Model *III* is given below: model *II* is the same but without the b_{jk} random effect for each count.

$$
\begin{aligned}
y_{jk} &\sim \text{Poisson}(\mu_{jk}) \\
\log \mu_{jk} &= \alpha_0 + \alpha_{Base} \log(\text{Base}_j/4) + \alpha_{Trt}\text{Trt}_j \\
&\quad \alpha_{BT}\text{Trt}_j \log(\text{Base}_j/4) + \alpha_{Age}\text{Age}_j + \alpha_{V4}\text{V4}_k + b1_j + b_{jk} \\
b1_j &\sim \text{Normal}(0, \tau_{b1}) \\
b_{jk} &\sim \text{Normal}(0, \tau_b).
\end{aligned}
$$

Coefficients and precisions are given independent "noninformative" priors. The appropriate graph is shown in Figure 12

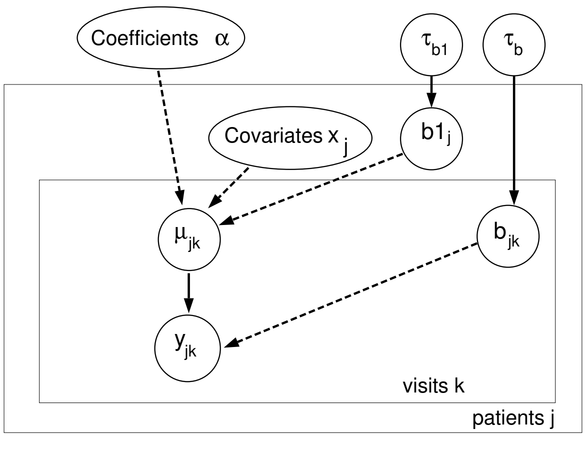

Figure 12: Graphical model for `epil` example, using Model *III* of Breslow and Clayton (1993)

Model specification for `epil` **example**

```
model epil3;
const
   N = 59,   # number of patients
   T = 4;    # number of clinic visits
var
   y[N,T], mu[N,T], b1[N], b[N,T], Base[N], log.Base4[N], Trt[N],
   Age[N], log.Age[N], V4[T], BT[N], alpha.Base, alpha.Trt,
   alpha.Age, alpha.V4, alpha.BT, alpha0, a0, log.Base4.bar, Trt.bar,
   log.Age.bar, V4.bar, BT.bar, tau.b1, tau.b, sigma.b1, sigma.b;
data y in "epily.dat", Trt,Base,Age in "epilcov.dat", V4 in "epilv4.dat";
inits in "epil.in";
{
   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;
}
```

Note that in the `BUGS` code we have standardized each covariate about its mean to ensure approximate prior independence between the regression coefficients. Without this parameterization we have found severe convergence problems.

Analysis

For both models, a burn-in of 3000 iterations was followed by a further 6000 iterations. This took approximately 15 minutes for model *II* and 30 minutes for model *III*, showing the relative slowness of handling models of this sort. The results may be compared with those of Breslow and Clayton (1993).

	PQL	`BUGS`	PQL	`BUGS`
variable	coeff ± SE	coeff ± SE	coeff ± SE	coeff ± SE
Fixed effects				
constant	-1.25 ± 1.2	-1.46 ± 1.29	-1.27 ± 1.2	-1.44 ± 1.25
Base	.87 ± .14	.85 ± .11	.86 ± .13	.91 ± .13
Trt	-.91 ± .41	-1.10 ± .32	-.93 ± .40	-.89 ± .42
Base x Trt	.33 ± .21	.43 ± .17	.34 ± .21	.31 ± .21
Age	.47 ± .36	.54 ± .35	.47 ± .35	.48 ± .37
V4	-.16 ± .05	-.16 ± .06	-.10 ± .09	-.11 ± .09
Subject level random effects				
σ_{b1}	.53 ± .06	.54 ± .07	.48 ± .06	.50 ± .07
Unit level random effects				
σ_b	—	—	.36 ± .04	.37 ± .04

10 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_i^C denote number of events in the control group in trial i, and r_i^T denote events under active treatment in trial i. Our model is:

$$
\begin{aligned}
r_i^C &\sim \text{Binomial}(p_i^C, n_i^C) \\
r_i^T &\sim \text{Binomial}(p_i^T, n_i^T) \\
\text{logit}(p_i^C) &= \mu_i \\
\text{logit}(p_i^T) &= \mu_i + \delta_i \\
\delta_i &\sim \text{Normal}(d, \tau).
\end{aligned}
$$

"Noninformative" priors are given for the μ's. τ and d. The graph for this model is shown in Figure 13. 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 $p(\delta_{new}|\hat{d}, \hat{\tau})$. *Full Bayes* allows for the uncertainty concerning d and τ.

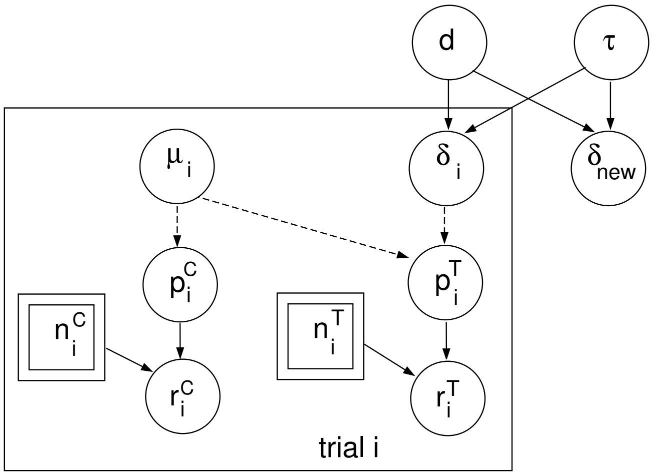

Figure 13: Graphical model for `blocker` example

Model specification for `blocker` example

```
model blocker;

const
   Num=22;    # Number of studies

var
  rt[Num], nt[Num], rc[Num], nc[Num],
  pc[Num], pt[Num], mu[Num], delta[Num],
  d, tau, sigma, delta.new;

data rt, nt, rc, nc in "blocker.dat";
inits in "blocker.in";

{

  for (i in 1:Num) {
       rt[i]          ~ dbin(pt[i],nt[i]);
       rc[i]          ~ dbin(pc[i],nc[i]);
       logit(pc[i]) <- mu[i];
       logit(pt[i]) <- mu[i] + delta[i];
       delta[i]       ~ dnorm(d,tau);
       mu[i]          ~ dnorm(0.0,1.0E-5);
  }

  d          ~ dnorm(0.0,1.0E-6);
  tau        ~ dgamma(1.0E-3,1.0E-3);
  sigma    <- 1/sqrt(tau);
  delta.new ~ dnorm(d,tau);

}
```

Analysis

A simple `BUGS` run of 1000 iterations took 23 seconds and gave the following results which may be compared to those of Carlin (1992).

variable	Carlin coeff ± SE	`BUGS` coeff ± SE
population mean (d)	-.243 ± .071	-.259 ± .056
new study (δ_{new})	-.245 ± .203	-.263 ± .119
scale (σ)	–	.096 ± .057

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 $p(\sigma^2) \propto 1/\sigma^2$ (see his Figure 1).

10.1 A t-distribution as a population prior

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}. The BUGS code, for a t distribution on $v = 4$ degrees of freedom, is given in blockert.bug, and the essential shown below.

```
for (i in 1:Num) {
     ......
     delta[i] ~ dt(d,tau,4);
     mu[i]    ~ dnorm(0.0,1.0E-5);
}
d         ~ dnorm(0.0,1.0E-6);
delta.new ~ dt(d,tau,4);
```

A 1000 iteration run produced an estimate for d of -.241 (SD .072), for δ_{new} of -.230 (SD .180) and for *sigma* of 0.102 (SD .056), showing little influence in allowing the different shaped distribution, since there are no outlying studies.

10.2 A hierarchical t-distribution with unknown degrees of freedom

It is possible to treat the degrees of freedom parameter v as an additional unknown random variable in the model. This is most easily accomplished by specifying v to be a discrete variable. For example, let v take values 2,4,6,8,10,12,15,20,30 or 50. We then assume a uniform prior over the possible categories. That is, each value of v is given equal prior probability = 1/number of categories. The BUGS code is given in blockht.bug, and is shown on the next page.

Analysis

2000 iterations took 67 seconds, after a burn-in of 2000 iterations (note that we have used a longer burn-in than usual, since the degrees of freedom parameter typically converges quite slowly). This produced the following estimates: d = -.256 (SD .062); δ_{new} = -.260 (SD .149); σ = .095 (SD .054), and v = 10.300 (SD 4.746). A t distribution on v = 10.3 degrees of freedom has a slightly heavier tail than Normal. However, the estimates of d and δ_{new} are very similar to those obtained by assuming a Normal population prior for the true treatment effects in each study, suggesting that allowing the different shaped distribution has little influence.

The number of categories and choice of values for v is somewhat arbitary. This can influence the resulting estimates since there is generally little information in the data concerning the value of v. We have found that a prior which places greater weight on low degrees of freedom, but also includes values large enough to give an approximately Normal distribution (*e.g.* v= 30, 50) works best. Some fine-tuning may be necessary to ensure that 'jumps' between successive values of v are small enough to ensure that the sampler does not get 'stuck' on a single value for hundreds of iterations. In addition, it is not necessary to assume equal prior probabilities for each category of v (see Verdinelli and Wassweman (1991) for example). Note that assuming a *continuous* distribution for v would lead to a non log-concave full conditional distribution which BUGS is currently unable to sample from — see the Dugongs example for further details of this type of problem.

Model specification for `blockht` **example**

```
model blockht;

const
   Num=22,    # Number of studies
   Nbins=10;   # Number of categories for v

var
  rt[Num], nt[Num], rc[Num], nc[Num], pc[Num],
  pt[Num], mu[Num], delta[Num], d, tau, sigma,
  delta.new, v, eta[Nbins], k, prior[Nbins];

data rt, nt, rc, nc in "blocker.dat";
inits in "blocker.in";

{

  for (i in 1:Num) {
       rt[i]          ~ dbin(pt[i],nt[i]);
       rc[i]          ~ dbin(pc[i],nc[i]);
       logit(pc[i]) <- mu[i];
       logit(pt[i]) <- mu[i] + delta[i];
       delta[i]       ~ dt(d,tau,v);
       mu[i]          ~ dnorm(0.0, 1.0E-5);
  }
  delta.new      ~ dt(d,tau,v);
  d       ~ dnorm(0.0,1.0E-6);
  tau     ~ dgamma(1.0E-3,1.0E-3);
  sigma <- 1/sqrt(tau);

  for (n in 1:Nbins) {
       prior[n] <- 1/Nbins;   # Uniform prior on v
  }
  k ~ dcat(prior[]);
  v <- eta[k];            # degrees of freedom for t

  # Specify values taken by v: note that these
  # could be included in the data file instead

  eta[1] <- 2;   eta[2] <- 4;   eta[3] <- 6;
  eta[4] <- 8;   eta[5] <- 10   eta[6] <- 12;
  eta[7] <- 15;  eta[8] <- 20;  eta[9] <- 30;
  eta[10] <- 50;

}
```

11 `Oxford`: smooth fits to log-odds ratios in case-control studies

Breslow and Clayton (1993) re-analyse 2×2 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-rays/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

$$\begin{aligned} \log \psi_i &= \alpha + \beta_1 \text{year}_i + \beta_2(\text{year}_i^2 - 22) + b_i \\ b_i &\sim \text{Normal}(0, \tau). \end{aligned}$$

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

We let r_i^0 denote number of exposures among the n_i^0 controls in stratum i, and r_i^1 denote number of exposures for the n_i^1 cases. The we assume

$$\begin{aligned} r_i^0 &\sim \text{Binomial}(p_i^0, n_i^0) \\ r_i^1 &\sim \text{Binomial}(p_i^1, n_i^1) \\ \text{logit}(p_i^0) &= \mu_i \\ \text{logit}(p_i^1) &= \mu_i + \log \psi_i \end{aligned}$$

Assuming this model with independent vague priors for the μ_i's provides the correct conditional likelihood. The appropriate graph is shown in Figure 14.

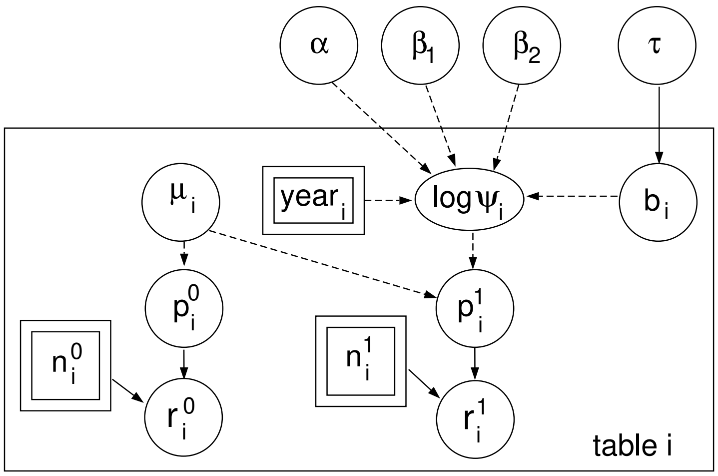

Figure 14: Graphical model for `oxford` example

`Oxford`: model specification

```
model oxford;

const
   K=120;    # Number of strata

var
  r1[K], n1[K], r0[K],  n0[K],year[K],
  p1[K], p0[K], mu[K], logPsi[K], b[K],
  alpha, beta1, beta2, sigma, tau;

data r1,n1,r0,n0,year in "oxford.dat";
inits in "oxford.in"

{

  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*(pow(year[i],2)-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);

}
```

Analysis

A simple `BUGS` run took 3 minutes for 1000 iterations after a 500 iteration burn-in. The comparison with the PQL fit of Breslow and Clayton (1993) is as follows.

	PQL	`BUGS`
variable	coeff ± SE	coeff ± SE
constant (α)	.566 ± .070	.580 ± .061
year (β_1)	-.0469 ± .0167	-.0471 ± .016
$\text{year}^2 - 22(\beta_2)$	.0071 ± .0033	-.0070 ± .0030
scale (σ)	.15 ± .10	.13 ± .06

12 LSAT: latent variable models for item-response data

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
Total		1000

12.1 Rasch model

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:

$$\begin{aligned} \text{logit}(p_{jk}) &= \theta_j - \alpha_k, \qquad j = 1, ..., 1000; k = 1, ..., 5 \\ \theta_j &\sim \text{Normal}(0, \tau) \end{aligned}$$

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

$$\begin{aligned} \text{logit}(p_{jk}) &= \beta\theta_j - \alpha_k, \qquad j = 1, ..., 1000; k = 1, ..., 5 \\ \theta_j &\sim \text{Normal}(0, 1) \end{aligned}$$

where β corresponds to the scale parameter ($\sqrt{\frac{1}{\tau}}$) 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 $a_k = \alpha_k - \overline{\alpha}$ to enable comparision of the BUGS posterior parameter estimates with the Boch and Aitkin marginal maximum likelihood estimates.

Boch and Aitkin compute the following likelihood ratio chi-square statistic (deviance) for testing the assumed model against a general multinomial alternative

$$G^2 = 2\left(\sum_{i=1}^{R} m_i \log \frac{m_i}{NP_i}\right)$$

on $R-1-q$ degrees of freedom, where q is the number of paramters to be estimated, m_i is the observed number of students scoring response pattern i = 1,...,R=32 and P_i is the (unconditional) probability of a randomly selected student responding with pattern i. *Conditional* on ability level θ, this probabilty is

$$P_{i|\theta} = \prod_{k=1}^{5} p_{k|\theta}^{r_{ik}}(1-p_{k|\theta})^{(1-r_{ik})}$$

where $p_{k|\theta} = \beta\theta - \alpha_k$ and $r_{ik} = 0$ or 1 according to the value of the kth item in the ith pattern.

To obtain the *unconditional* probability P_i, we integrate over θ as follows

$$P_i = \int P_{i|\theta} f(\theta) d\theta$$

where $f(\theta)$ is the posterior ability distribution. Using Monte Carlo integration and a sample of simulated values for $P_{i|\theta}$, we may approximate this integral by

$$P_i \approx \frac{1}{n}\sum_n P_{i|\theta}^{(n)}$$

where n indexes iteration. That is, we may estimate P_i by the posterior mean of $P_{i|\theta}$. To generate $P_{i|\theta}$ within BUGS, we just sample a random ability paramter θ_{new} from a standard normal at each iteration, calculate $p_{k|\theta}$ and substitute this into the equation for $P_{i|\theta}$ above. At the end of the BUGS run, we calculate the posterior mean of the $P_{i|\theta}$'s and use this as our estimate of P_i in the formula for G^2.

The BUGS code for this model is given on the next page, and the graph is shown in Figure 15.

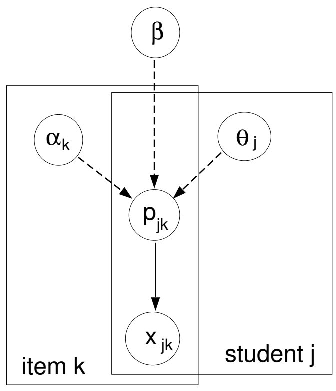

Figure 15: Graphical model for LSAT example.

```
model LSAT;
const
   N = 1000,    # number students
   R = 32,      # number of possible test results
   T = 5;       # number of tests
var
   response[R,T], m[R], culm[R], alpha[T], a[T], theta[N], r[N,T],
   p[N,T], beta, theta.new, p.theta[T], p.item[R,T], P.theta[R];
data response,m,culm in "lsat.dat";
inits in "lsat.in";
{
# 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,);

# Compute probability of response pattern i, for later use in computing G^2
  theta.new ~ dnorm(0,1);          # ability parameter for random student
  for(k in 1:T) {
       logit(p.theta[k]) <- beta*theta.new - alpha[k];
       for(i in 1:R) {
          p.item[i,k] <- pow(p.theta[k],response[i,k])
                          * pow((1-p.theta[k]),(1-response[i,k]));
       }
  }
  for(i in 1:R) {
     # P_i|theta = PROD_k p_k|theta
     P.theta[i] <- p.item[i,1]*p.item[i,2]*p.item[i,3]*p.item[i,4]*p.item[i,5];
  }
}
```

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

Analysis

A 2000 iteration `BUGS` run took 55 minutes after a 1000 iteration burn-in and produced the following results:

		† **B&A estimate**	**`BUGS` mean (95% C.I.)**		**`BUGS` normalized mean (95% C.I.)**	
	Item		α		a	
Threshold	1	−1.255	−2.738	(−2.991, −2.490)	−1.258	(−1.459, −1.056)
	2	0.476	−1.004	(−1.163, −0.846)	0.476	(0.337, 0.608)
	3	1.235	−0.240	(−0.388, −0.096)	1.242	(1.111, 1.376)
	4	0.168	−1.310	(−1.477, −1.142)	0.169	(0.029, 0.310)
	5	−0.625	−2.109	(−2.323, −1.905)	−0.629	(−0.805, −0.471)
β		–	0.762	(0.623, 0.904)	–	–
G^2 (on 25 d.f.)		21.80	22.08	–	–	–

† B&A =Boch & Aitkin (1981) Marginal maximum likelihood estimate

12.2 2-parameter probit model

The Rasch model for the LSAT data may be extended to a 2-parameter model, and a normal cumulative density assumed for the item response function, as follows:

$$\begin{aligned} \text{probit}(p_{jk}) &= \delta_k(\theta_j - \gamma_k), \qquad j = 1, ..., 1000; k = 1, ..., 5 \\ \theta_j &\sim \text{Normal}(0, 1) \end{aligned}$$

Here, δ_k, the slope parameter, provides a measure of how well item k discriminates between individuals of different abilities. The threshold parameter γ_k is again a measure of item difficulty. For computational purposes, it is convenient to re-parameterize the above model as

$$\text{probit}(p_{jk}) = \delta_k \theta_j - \eta_k, \qquad j = 1, ..., 1000; k = 1, ..., 5$$

where $\eta_k = \delta_k \gamma_k$.

Note that, as for the 1-parameter Rasch model, we calculate $e_k = \eta_k - \overline{\eta}$, which corresponds to a sum-to-zero constraint on the item-intercept parameters. In addition, we must impose a constraint on the slope parameters δ_k to ensure identifiabililty in the 2-parameter Rasch model. In `BUGS` this is achieved by specifying a half-normal prior with scale = 1 for the δ_k's; this fixes the population variance and constrains the slopes to be positive. Alternatively, Boch and Aitkin (1981) apply the restriction that $\prod_{k=1}^{5} \delta_k = 1$. Hence we also calculate $d_k = \delta_k \ / \ (\prod_{k=1}^{5} \delta_k)^{\frac{1}{5}}$ to enable comparison of the `BUGS` and Boch and Aitkin slope estimates. Finally we compute $g_k = e_k \ / \ d_k$ in `BUGS`; these correspond to Boch and Aitkin's 'sum-to-zero' constrained threshold estimates.

The essentials of the BUGS code for the 2 parameter model are given below.

```
for (j in 1:N) {
   for (k in 1:T) {
      probit(p[j,k]) <- delta[k]*theta[j] - eta[k];
      r[j,k] ~ dbern(p[j,k]);
      ......
   }
   theta[j] ~ dnorm(0,1);
}
for (k in 1:T) {
   eta[k] ~ dnorm(0,0.0001);
   e[k] <- eta[k] - mean(eta[]);  # sum-to-zero constraint

   delta[k] ~ dnorm(0,1) I(0,);   # scale = 1, slope +ve
   d[k] <- delta[k]/pow(delta[1]*delta[2]*delta[3]
           *delta[4]*delta[5], 0.2);  # PRODUCT_k (d_k) = 1

   g[k] <- e[k]/d[k];   # equivalent to B&A's threshold parameters
}
```

Analysis

A 5000 iteration BUGS run took approxiamtely 2 hours after a 5000 iteration burn-in and produced the following results:

		† **B&A estimate**	**BUGS mean (95% CI)**		**BUGS normalized mean (95% CI)**	
	Item			η		g
	1	−0.679	−1.558	(−1.789, −1.395)	−0.699	(−0.890, −0.556)
	2	0.316	−0.601	(−0.708, −0.504)	0.256	(0.159, 0.352)
Threshold	3	0.788	−0.156	(−0.255, −0.064)	0.704	(0.620, 0.795)
	4	0.092	−0.777	(−0.901, −0.671)	0.080	(0.029, 0.178)
	5	−0.517	−1.204	(−1.358, −1.075)	−0.346	(−0.474, −0.230)
	Item			δ		d
	1	0.979	0.410	(0.163, 0.716)	0.995	(0.456, 1.696)
	2	1.015	0.428	(0.222, 0.679)	1.047	(0.593, 1.709)
Slope	3	1.265	0.569	(0.317, 0.972)	1.414	(0.778, 2.646)
	4	0.948	0.405	(0.204, 0.657)	0.987	(0.541, 1.583)
	5	0.840	0.361	(0.143, 0.619)	0.874	(0.407, 1.475)
G^2 (on 21 d.f.)		21.29	21.66	–	–	–

† B&A =Boch & Aitkin (1981) Marginal maximum likelihood estimate

13 Bones: latent trait model for multiple ordered categorical 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 the `BUGS` `calibrat.dat` file, which contains this data as a rectangular array, 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.

```
 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 1–34**
1	0.6	1 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 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 1 3 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 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

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

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 code for this model is given below, and the graph is shown in Figure 16. 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.

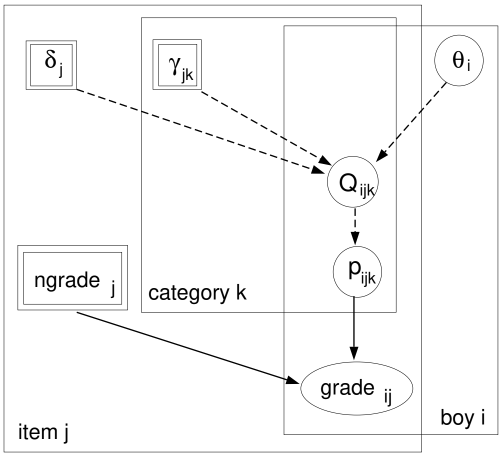

Figure 16: Graphical model for `bones` example.

```
const
   nChild = 13,   # number of children
   nInd = 34,     # total number of indicators
   nGrade = 5;    # maximum number of grades
var
   grade[nChild,nInd],        # grade of each indicator for each child
   p[nChild,nInd,nGrade],     # probability of response
   Q[nChild,nInd,nGrade - 1], # cumulative prob of > response
   delta[nInd],               # discrimination parameter
   gamma[nInd,nGrade - 1],    # threshold parameters
   ncat[nInd],                # number of categories for each indicator
   theta[nChild];             # latent variable = skeletal age

data gamma, delta,  ncat in "calibrat.dat", grade in "bones.dat";
inits in "bones.in";

{
   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]]);

      }
   }
}
```

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

```
for (k in 2:(ngrade[j]-1)) { .... }
```

This loop need only be evaluated for items with 3 or more grades. It will be skipped whenever `j` corresponds to binary items because the BUGS compiler will not attempt to generate any code within a loop where the second index < first index. In the present example, whenever `j` corresponds to a binary item, `ngrade[j]` $= 2$, and so the loop indices become

```
for (k in 2:1) { .... }
```

Analysis

A 500 iteration burn-in (25 seconds) followed by 1000 updates (52 seconds) gave the following estimates of SA (θ_i) for each boy

Boy ID	**CA**	BUGS		**Multilog 5 (Thissen)**	
1	0.6	0.306	±0.208	0.341	±0.181
2	1.0	1.351	±0.254	1.316	±0.250
3	2.0	2.350	±0.292	2.346	±0.266
4	3.0	2.902	±0.290	2.909	±0.288
5	5.0	5.517	±0.514	5.510	±0.492
6	6.0	6.744	±0.613	6.711	±0.606
7	7.0	6.455	±0.584	6.411	±0.591
8	8.0	8.920	±0.720	8.928	±0.709
9	9.0	8.977	±0.640	9.001	±0.680
10	12.0	11.93	±0.694	11.911	±0.697
11	14.0	11.58	±0.923	11.408	±0.859
12	16.0	15.78	±0.548	15.758	±0.553
13	18.0	16.98	±0.738	16.887	±0.712

14 `Inhaler`: random effects model for ordinal responses from a cross-over trial

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	**2**	**3**	**4**	**TOTAL**
		Easy	**Only clear after re-reading**	**Not very clear**	**Confusing**	
Response	**1**	59 (63)	35 (13)	3 (0)	2 (0)	99 (76)
in	**2**	11 (40)	27 (15)	2 (0)	1 (0)	41 (55)
period 1	**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 $R_{i,t}$ from the ith subject ($i = 1, ..., 286$) in the tth period ($t = 1, 2$) thus assumes integer values between 1 and 4. It may be expressed in terms of a continuous latent variable $Y_{i,t}$ taking values on $(-\infty, \infty)$ as follows:

$$R_{i,t} = j \qquad \text{if} \qquad Y_{i,t} \in [a_{j-1}, a_j), \qquad j = 1, .., 4 \tag{1}$$

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

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

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

$$\begin{aligned}
\mu_{1,1} &= \frac{\beta}{2} + \frac{\pi}{2} \\
\mu_{1,2} &= -\frac{\beta}{2} - \frac{\pi}{2} - \kappa \\
\mu_{2,1} &= -\frac{\beta}{2} + \frac{\pi}{2} \\
\mu_{2,2} &= \frac{\beta}{2} - \frac{\pi}{2} + \kappa
\end{aligned}$$

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 $p_{i,t,j} = Q_{i,t,j-1} - Q_{i,t,j}$, where $Q_{i,t,0} = 1$ and $Q_{i,t,4} = 0$ (see also the `Bones` example).

The graph of this model is shown in Figure 17, and the BUGS code is given on the next page. 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 $a_1 < a_2 < a_3$.

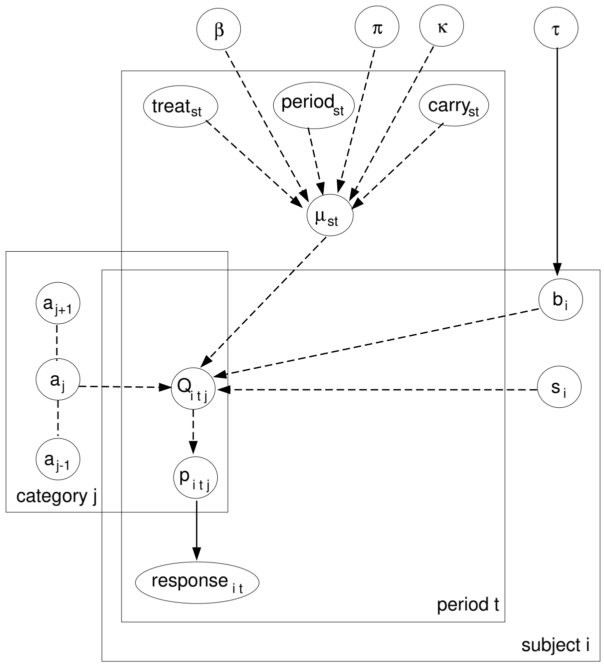

Figure 17: Graphical model for `inhaler` example

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. The second data file, `xover.dat` contains the three $2 \times T$ matrices `treat`, `period` and `carry` which indicate whether to add or subtract the treatment, period and carryover effects respectively when modelling the mean for each period and group.

Analysis

1500 iterations took 25 mins after a 1500 iteration burn-in. The results are shown below, and are compared with those of Ezzet and Whitehead, who used the Newton-Raphson method and numerical integration to obtain maximum-likelihood estimates of the parameters.

parameter	**ML estimate**	**(S.E.)**	BUGS **mean**	**(S.E.)**
β (treatment)	1.17	(0.75)	1.06	(0.32)
π (period)	−0.23	(0.20)	−0.24	(0.19)
κ (carryover)	0.21	(0.49)	0.25	(0.25)
σ	–	–	1.26	(0.26)
log σ (log S.D.)	0.17	(0.23)	0.21	(0.21)
a_1	0.68	–	0.72	(0.14)
a_2	3.85	–	3.95	(0.35)
a_3	5.10	–	5.31	(0.48)

Model specification for `inhaler` example

```
model inhaler;

const
  N = 286,       # number of patients
  T = 2,         # number of periods
  S = 2,         # number of sequences (AB & BA)
  Npattern = 16, # number of possible response patterns
  Ncut = 3;      # number of cut points (= number of response categories - 1)

var
  pattern[Npattern,T], # response pattern e.g. (1,1) or (2,4) etc.
  Ncum[Npattern,T],    # cumulative total
  response[N,T],       # response for patient i in period t
  p[N,T,(Ncut+1)],     # prob of response j for patient i in period t
  Q[N,T,Ncut],         # cumulative prob of response worse than j
                       # for patient i in period t
  seq[N],              # treatment sequence (1=AB; 2=BA)
  mu[S,T],             # logistic mean for group g & period t
  treat[S,T], beta,    # treatment effect
  period[S,T], pi,     # period effect
  carry[S,T], kappa,   # carryover effect
  a[Ncut],             # cut points for latent response variable
  b[N],                # subject random effect
  tau,                 # precision of subject effects
  sigma, log.sigma;
data pattern, Ncum in "inhaler.dat", treat, period, carry in "xover.dat";
inits in "inhaler.in";
{
#
# Construct individual response data from contingency table
#
  for (i in 1:Ncum[1,1]) {
     seq[i] <- 1; for (t in 1:T) { response[i,t] <- pattern[1,t] }
  }
  for (i in (Ncum[1,1]+1):Ncum[1,2]) {
     seq[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]) {
        seq[i] <- 1; for (t in 1:T) { response[i,t] <- pattern[k,t] }
     }
     for(i in (Ncum[k,1]+1):Ncum[k,2]) {
        seq[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[seq[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 (s in 1:S) {
     for(t in 1:T) {
        # logistic mean for sequence s in period t
        mu[s,t] <- beta*treat[s,t]/2 + pi*period[s,t]/2 + kappa*carry[s,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);
}
```

15 `Litters`: beta-binomial for clustered response data

The table below shows the data of Williams (1975) on the mortality in 2 sets of 16 litters of pigs.

Litter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Proportion (set 1) Surviving	$\frac{13}{13}$	$\frac{12}{12}$	$\frac{9}{9}$	$\frac{9}{9}$	$\frac{8}{8}$	$\frac{8}{8}$	$\frac{12}{13}$	$\frac{11}{12}$	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{8}{9}$	$\frac{11}{13}$	$\frac{4}{5}$	$\frac{5}{7}$	$\frac{7}{10}$	$\frac{7}{10}$
Proportion (set 2) Surviving	$\frac{12}{12}$	$\frac{11}{11}$	$\frac{10}{10}$	$\frac{9}{9}$	$\frac{10}{11}$	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{8}{9}$	$\frac{8}{9}$	$\frac{4}{5}$	$\frac{7}{9}$	$\frac{4}{7}$	$\frac{5}{10}$	$\frac{3}{6}$	$\frac{3}{10}$	$\frac{0}{7}$

We would like to assume that the survival rates in the litters within each set are similar, but not identical. The simplest conjugate model is to assume the observed number of deaths r_{ij} in litter i of group j is binomial with sample size n_{ij} and true rate p_{ij}, and then assume the true rates are drawn from a beta distribution with unknown parameters. This model is also considered by George *et al.* (1993).

$$
\begin{aligned}
r_{ij} &\sim \text{Binomial}(p_{ij}, n_{ij}) \\
p_{ij} &\sim \text{Beta}(a_j, b_j)
\end{aligned}
$$

The beta parameters are given gamma priors, which must have a parameter $\alpha \geq 1$ in order to be log-concave: we have chosen gamma(1,.001) priors.

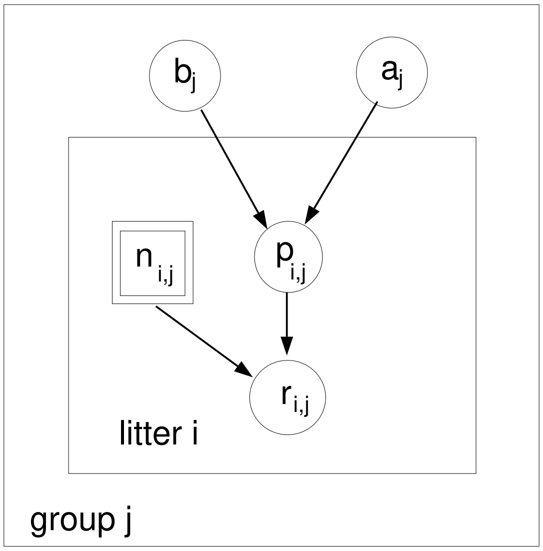

Figure 18: Graphical model for `litter` example

Litters: model specification in BUGS

```
model litters;
const
     N = 16,    # number of litters in each group
     G = 2;     # number of groups
var
   n[N,G],    # denominators
   r[N,G],    # survivors
   p[N,G],    # true survival rpob
     a[G],    # beta parameter
     b[G],    # beta parameter
    mu[G],    # beta mean
 theta[G];    # 1/ beta precision
data n, r in "litters.dat";
inits in "litters.in";
{
  for (j in 1:G) {
     for (i in 1:N) {
        r[i,j] ~ dbin(p[i,j],n[i,j]);
        p[i,j] ~ dbeta(a[j],b[j]);
     }
     mu[j] <- a[j]/(a[j] + b[j]);   theta[j] <- 1.0/(a[j] + b[j]);
     a[i] ~ dgamma(1,.001);   b[i] ~ dgamma(1,.001);
  }
}
```

Note the use of the gamma(1,.001) priors for the a's and b's, equivalent to exponential distributions with mean 1000. This provides a log-concave prior which gently penalizes larger values.

A simple BUGS run took 13 seconds for 5000 iterations after a 500 iteration burn-in, and gave the following results.

parameter	BUGS	(S.E.)	Maximum likelihood	(S.E.)
a_1	1482	886		
b_1	182	116		
μ_1	.892	0.021	0.898	0.026
θ_1	.0016	0.0004	0.021	0.048
a_2	3.6	2.8		
b_2	1.1	.8		
μ_2	.752	0.060	0.740	0.069
θ_2	.307	0.185	0.465	0.234

Warning

The above estimates, particularly a_1, a_2 suffer from extremely poor convergence, limited agreement with m.l.e.'s, and considerable prior sensitivity. This appears to be primarily due to the parameterisation in terms of the highly related a_i and b_i, whereas direct sampling of μ_i and θ_i would be strongly preferable. We suggest that random effects logistic models (see `seeds` example) may be more appropriate.

16 `Mice`: Weibull regression in censored survival analysis

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

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

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

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

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

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

For censored observations, the survival distribution is a *truncated* Weibull, with lower bound corresponding to the censoring time. The regression coefficients β 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_j is given by $m_j = (\log 2 e^{-\beta' z_j})^{1/r}$.

The appropriate graph is shown in Figure 19, using an undirected dashed line to represent a logical range constraint. The `BUGS` code is given on the next page.

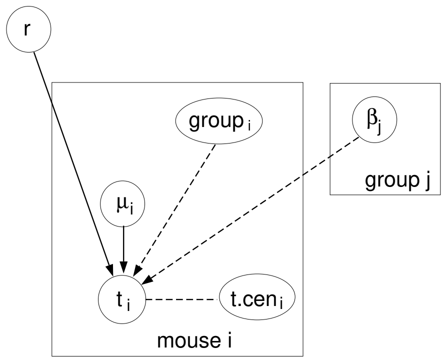

Figure 19: Graphical model for `mice` example

Model specification for `mice` example

```
model mice;

const
  N = 80,    # number of individuals
  M = 4;     # number of treatment groups
var
  t[N],                     # failure time for each mouse
  t.cen[N],                 # censoring time for each mouse
  group[N],                 # treatment group for each mouse
  mu[N], r,                 # Weibull parameters
  beta[M],                  # log relative risk parameters
  median[M],                # median survival for each group
  irr.control,veh.control, # treatment contrasts
  test.sub,pos.control;

data t, t.cen, group in "mice.dat";
inits in "mice.in";
{
  for(i in 1:N) {                            # t.cen[i] = 0
        t[i] ~ dweib(r,mu[i])I(t.cen[i],); # if mouse i fails

        mu[i] <- exp(beta[group[i]]);      # relative risk model
  }
  for(j in 1:M) {
        beta[j] ~ dnorm(0.0, 0.0001);      # prior
        median[j] <- pow(log(2) *          # median survival
                       exp(-beta[j]), 1/r);
  }
  r ~ dgamma(1.0,0.0001);      # slowly decreasing on +ve reals

  irr.control <- beta[1];                  # change
  veh.control <- beta[2]-beta[1];          # parameterisation
  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, and create a single covariate `group` taking values 1, 2, 3 or 4 in the data file. Nested subscripts i.e. `beta[group[i]]` are used to select the required `beta[j]` to appear in the linear predictor according to the value of `group[i]` for the ith individual. 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 in rectangular format

```
12  0 1
17  0 1
21  0 1
25  0 1
11  0 1
26  0 1
.. .. .
.. .. .
35  0 1
NA 40 1
31  0 1
36  0 1
32  0 2
27  0 2
23  0 2
12  0 2
18  0 2
NA 40 2
.. .. .
.. .. .
```

Column 1 refers to the failure times `t` for each mouse, column 2 represents the corresponding censoring times `t.cen`, and column 3 indicates which treatment group the mouse belongs to.

Analysis

A simple `BUGS` run took 22 seconds for 1000 iterations after a 500 iteration burn-in. The output was as follows:

variable	estimate	95% interval
`veh.control`	-1.18	-1.92, -0.43
`test.sub`	-0.34	-1.00, 0.34
`pos.control`	0.38	-0.35, 1.01
r	3.15	2.23, 3.59
`median[1]` (irr)	24.2	21.1, 28.1
`median[2]` (veh)	35.4	39.7, 42.5
`median[3]` (test)	27.0	22.9, 31.7
`median[4]` (pos)	21.5	18.4, 25.3

These should be compared to the plots shown by Dellaportas and Smith (1993)

17 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=censored)	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

$$\begin{aligned}
t_{ij} &\sim \text{Weibull}(r, \mu_{ij}) \qquad i = 1, ..., 38; j = 1, 2 \\
\log \mu_i &= \alpha + \beta_{age}\text{AGE}_{ij} + \beta_{sex}\text{SEX}_i + \beta_{disease_1}\text{DISEASE}_{i,1} \\
&\quad + \beta_{disease_2}\text{DISEASE}_{i,2} + \beta_{disease_3}\text{DISEASE}_{i,3} + b_i \\
b_i &\sim \text{Normal}(0, \tau)
\end{aligned}$$

where $\text{AGE}_{i,j}$ is a continuous covariate, SEX_i is a 2-level factor and $\text{DISEASE}_{i,k}$ $(k = 1, 2, 3)$ 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

$$\begin{aligned}
\beta_k &\sim \text{Normal}(0, 0.0001) \\
\tau &\sim \text{Gamma}(0.0001, 0.0001)
\end{aligned}$$

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 is shown in Figure 20, and the BUGS code is given below. The structure of the data file is similar to that used in the `mice` example.

`Kidney`: **model specification in** `BUGS`

```
model kidney;

const
  N = 38,    # number of patients
  M = 2;     # number of observations per patient
var
  t[N,M],                         # failure time
  t.cen[N,M],                     # censoring time
  mu[N,M], r,                     # Weibull parameters
  b[N],                           # random effects for patients
  tau,                            # precision of random effects
  sigma,                          # 1/sqrt(tau)
  age[N,M],sex[N],disease[N],     # covariates
  beta.age,beta.sex,              # regression coefficients
  beta.disease[4],alpha;          # regressioncoefficients

data t, t.cen, age, sex, disease in "kidney.dat";
inits in "kidney.in";
{
  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.disease[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.disease[1] <- 0;  # corner-point constraint
  for(k in 2:4) {
    beta.disease[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
}
```

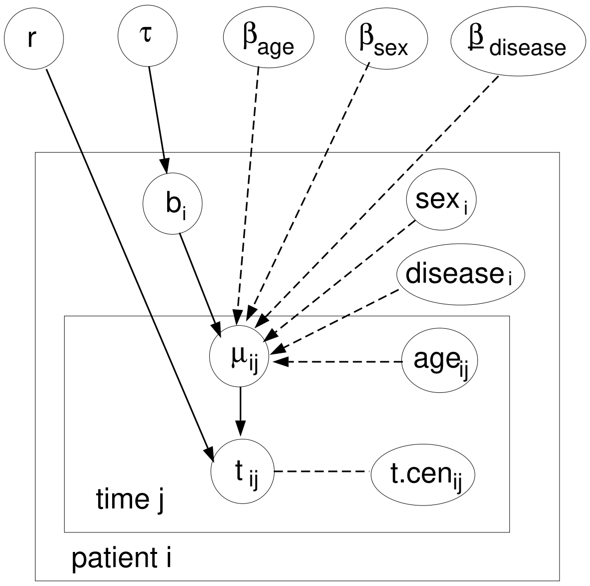

Figure 20: Graphical model for `kidney` example

Analysis

A BUGS run took 4 minutes for 2500 iterations after a 2500 iteration burn-in. The output is summarized in the table below, and the results of McGilchrist and Aisbett (1991)'s Cox analysis using an iterative Newton-Raphson estimation procedure are also shown for comparison.

variable	BUGS estimate (S.E.)	*McG&A* estimate (S.E.)
β_{age}	0.003 (0.015)	0.006 (0.013)
β_{sex} (female)	-1.866 (0.496)	-1.795 (0.434)
$\beta_{disease_1}$ (GN)	-0.055 (0.592)	0.206 (0.484)
$\beta_{disease_2}$ (AN)	0.586 (0.582)	0.410 (0.494)
$\beta_{disease_3}$ (PKD)	-1.269 (0.822)	-1.296 (0.712)
σ	0.496(0.378)	0.382

18 `Leuk`: survival analysis using 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

$$I_i(t)\mathrm{d}t \quad = \quad E(\mathrm{d}N_i(t) \mid \mathcal{F}_{t-})$$

where $\mathrm{d}N_i(t)$ is the increment of N_i over the small time interval $[t, t + \mathrm{d}t)$, and $\mathcal{F}_{t-}$ represents the available data just before time t. If subject i is observed to fail during this time interval, $\mathrm{d}N_i(t)$ will take the value 1; otherwise $\mathrm{d}N_i(t) = 0$. Hence $E(\mathrm{d}N_i(t) \mid \mathcal{F}_{t-})$ corresponds to the probability of subject i failing in the interval $[t, t + \mathrm{d}t)$. As $dt \to 0$ (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

$$I_i(t) \quad = \quad Y_i(t)\lambda_0(t)\exp(\beta' z_i)$$

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 $D = \{N_i(t), Y_i(t), z_i; i = 1, ..., n\}$ and unknown parameters β and $\Lambda_0(t) = \int_0^t \lambda_0(u)\ \mathrm{d}u$, the latter to be estimated non-parametrically.

The joint posterior distribution for the above model is defined by

$$P(\beta, \Lambda_0() \mid D) \quad \propto \quad P(D \mid \beta, \Lambda_0())P(\beta)P(\Lambda_0())$$

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

$$\prod_{i=1}^{n}\left[\prod_{t\geq 0} I_i(t)^{\mathrm{d}N_i(t)}\right]\exp(-\int_{t\geq 0} I_i(t)\mathrm{d}t)$$

This is essentially as if the counting process increments $\mathrm{d}N_i(t)$ in the time interval $[t, t + \mathrm{d}t)$ are independent Poisson random variables with means $I_i(t)\ \mathrm{d}t$:

$$\mathrm{d}N_i(t) \quad \sim \quad \mathrm{Poisson}(I_i(t)\mathrm{d}t)$$

We may write

$$I_i(t)\mathrm{d}t \quad = \quad Y_i(t)\exp(\beta' z_i)\mathrm{d}\Lambda_0(t)$$

where $\mathrm{d}\Lambda_0(t) = \lambda_0(t)\ \mathrm{d}t$ is the increment or jump in the integrated baseline hazard function occurring during the time interval $[t, t + \mathrm{d}t)$. 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 hazard function as failure times, whose logarithms are given 'non-informative' priors. This formulation is also shown below.

The fixed effect regression coefficients β are assigned a vague prior

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

The graph for the Cox model is shown in Figure 21.

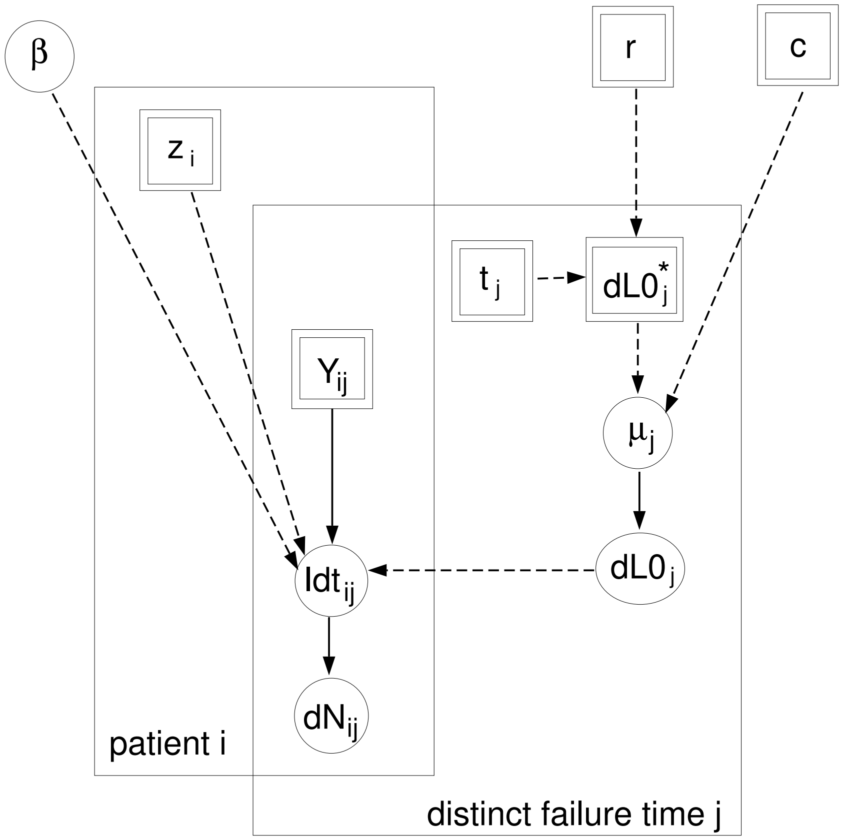

Figure 21: Graphical model for `leuk` example

A widely used example in survival analysis is Frierich *et al.* (1963)'s data comparing time to remission of leukaemia in patients receiving a new drug (6-MP) with control patients. The data are reproduced below.

Treatment	Survival time in weeks						
Placebo	1	1	2	2	3	4	4
	5	5	8	8	8	8	11
	11	12	12	15	17	22	23
6-MP	6*	6	6	6	7	9	10*
	10	11*	13	16	17*	19*	20*
	22	23	25*	32*	32*	34*	35*

* indicates censoring

The BUGS code to analyse this data using the Cox proportional hazards model is as follows

Model specification

```
model leuk;

const
   N = 42,    # number of patients
   T = 17,    # number of unique failure times
   eps = 0.000001;  # used to guard against numerical
                    # imprecision in step function

var
    obs.t[N],  # observed failure or censoring time for each patient
      t[T+1],  # unique failure times + maximum censoring time
     dN[N,T],  # counting process increment
      Y[N,T],  # 1=subject observed; 0=not observed
    Idt[N,T],  # intensity process
        Z[N],  # covariate
        beta,  # regression coefficient
      dL0[T],  # increment in unknown hazard function
    beta0[T],  # log(increment in unknown hazard function)
 dL0.star[T],  # prior guess at hazard function
           c,  # degree of confidence in prior guess for dL0
       mu[T],  # location parameter for Gamma (= c * dL0.star)
           r,  # prior guess at failure rate
     fail[N],  # failure = 1; censored = 0
  S.treat[T],  # survivor function for treatment group
S.placebo[T];  # survivor function for placebo group

data obs.t, fail, Z in "leuk.dat", t in "failtime.dat";
inits in "leuk.in";
```

```
{
# 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) {

#   beta0[j] ~ dnorm(0,0.001); # include this when using Poisson trick

  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

#  Try Poisson trick - independent log-normal hazard increments
#                    - enables dL0, c, r, mu to be dropped from model
#  Idt[i,j] <- Y[i,j]*exp(beta0[j]+beta*Z[i]); # 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);
}
```

Here, `obs.t[i]` is the follow-up time for patient i ($i = 1, ..., 42$), with `fail[i]` indicating whether this corresponds to a failure or a censored observation.

Data

Part of the data file `leuk.dat` is shown below: column 1 refers to `obs.t`, column 2 is `fail` and column 3 is `Z`

```
 1 1  0.5
 1 1  0.5
 2 1  0.5
 . .   .
 . .   .
17 1  0.5
22 1  0.5
23 1  0.5
 6 1 -0.5
 6 1 -0.5
 6 1 -0.5
 6 0 -0.5
 7 1 -0.5
 . .   .
 . .   .
32 0 -0.5
34 0 -0.5
35 0 -0.5
```

A separate data file (`failtime.dat`) contains the 17 distinct failure times `t[j]` ($j = 1, ..., 17$) plus `t[18]` $= t_{max}$, the maximum follow-up time. These values define the time intervals $[t, t + \mathrm{d}t)$ used for calculating the counting process increments $\mathrm{d}N_i(t)$. Note that the upper end of the interval is defined to be strictly $< t + \mathrm{d}t$, so the counting process for a subject who fails at exactly time $t + \mathrm{d}t$ will not be incremented until the following time interval. Consequently, if t_{max} represents a *failure* time rather than a censoring time, an arbitrary value $> t_{max}$ must be chosen for the final value of the vector `t[]`. This ensures that the counting process for this subject is incremented during the final time interval.

Time-dependent covariates could be included by making `Z[i,j]` the value for the ith individual at the jth failure time.

We note the use of the `step` function to create the counting process increments `dN[i,j]` $= \mathrm{d}N_i(t_j)$ and the risk set process `Y[i,j]` $= Y_i(t_j)$. This function takes the value 1 if its argument is ≥ 0, and 0 otherwise. Thus `step(time[i] - t[j] + eps)` returns the value 1 for `Y[i,j]` if the follow-up time `time[i]` for patient $i \geq$ the current failure time `t[j]`, and 0 thereafter. A similar trick is used in the construction of `dN[i,j]`.

Note that the variable `dN[i,j]` appears *twice* on the left-hand side of a statement. Under the section labelled `# TRANSFORMATIONS`, `dN[i,j]` is a deterministic node, whilst under the `# MODEL` section, it features as a random node. This construction is permitted in `BUGS` to allow creation of data nodes within the program session, rather than having to carry out all data transformations before setting up the `.dat` file (see Section 4.9 of the BUGS manual for more details).

We also calculate the conditional survivor function given covariates z. This is given by

$$S(t;z) = exp(-\int_0^t \lambda_0(u)\mathrm{d}u)^{\exp(\beta' z)}$$

where $\int_0^{t_J} \lambda_0(u)\ \mathrm{d}u = \Lambda_0(t_J) = \sum_{j=0}^{J} \mathrm{d}\Lambda_0(t_j)$.

Analysis

1000 iterations took 41 seconds after a 500 iteration burn-in. The posterior mean (standard error) of the regression coefficient β was 1.55 (0.43). This compares with the standard partial likelihood estimate (obtained using the SAS PHREG procedure) of 1.59 (0.43). Whitehead (1980) estimated β to be 1.51 (0.42) using the Poisson model formulation in GLIM: using the analogous Poisson trick in BUGS gave an estimate of 1.54 (.41). The estimated survival probabilities are shown in Figure 22.

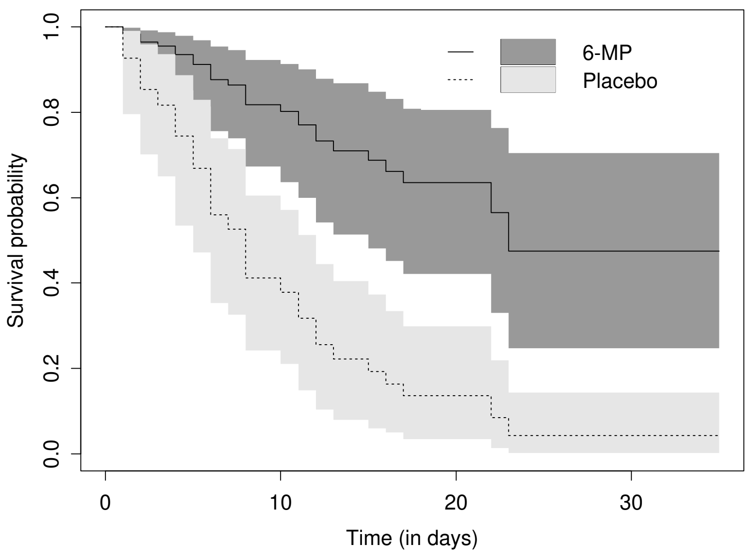

Figure 22: Estimated mean survivor function and 95% credible interval (shaded region) for treatment and placebo groups in the `leuk` example

Tied failure times

The Poisson likelihood formulation of the Cox model outlined above assumes that time is truly continuous, in the sense that no two failures can occur simultaneously. Hence the analysis described here is not strictly correct, since we have allowed multiple failures to occur at any one time. This analysis actually corresponds to Peto (1972)'s treatment of ties. An alternative approach is to randomly perturb the failure times prior to analysis to ensure distinct failure times for each subject. However, this is somewhat conservative since it assumes the null hypothesis to be true. A better method is to *simulate* new perturbations of the failure times at each iteration of the Gibbs sampler, conditional upon the data and the current values of other model parameters. Unfortunately, the declarative structure of the BUGS language does not permit this model at present, although future versions of the program may include it.

18.1 Cox regression with frailties

Frierich *et al.* (1963)'s data actually arise via a *paired* design, although this information has been ignored in most published analyses. 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 a fourth column (called `pair`) in the `BUGS` data file (`leukfr.dat`) to indicate each of the 21 pairs of patients:

```
 1 1  0.5  1
 1 1  0.5  2
 2 1  0.5  3
 . .   .   .
 . .   .   .
17 1  0.5 19
22 1  0.5 20
23 1  0.5 21
 6 1 -0.5 19
 6 1 -0.5 18
 6 1 -0.5  8
 6 0 -0.5  1
 . .   .   .
 . .   .   .
35 0 -0.5 21
```

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

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

A non-informative Gamma prior is assumed for τ, the precision of the frailty parameters. Note that the above 'additive' formulation of the frailty model is equivalent to assuming multiplicative frailties with a log-Normal population distibution. Alternatively Clayton (1991) discusses the Cox proportional hazards model with multiplicative frailties, but assumes a Gamma population distribution. However, this formulation does not lead conclusively to log-concave full conditional distributions, and so cannot currently be implemented in `BUGS`.

The modifications to the `BUGS` code needed to include a frailty term in the `leuk` example are shown below, and may be found in the file `leukfr.bug`.

```
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]+b[pair[i]])*dL0[j]; # Intensity
  }
  .........
}
for(k in 1:Npairs) {
    b[k] ~ dnorm(0.0, tau);
}
tau ~ dgamma(0.001, 0.001); sigma <- sqrt(1/tau);
```

Analysis

2000 iterations took 3 minutes after a 1000 iteration burn-in. The posterior mean (standard error) of the regression coefficient β was 1.58 (0.43). This is slightly larger than the estimate of β when the clustering was ignored. However, the posterior mean of σ was only 0.18 (95% credible interval 0.02–0.59). This suggests that population variation in frailty was small, and hence there is relatively little clustering or association of failure times within matched pairs.

References

Andersen, E. B. (1980). *Discrete statistical models with social science applications.* North-Holland, Amsterdam.

Andersen, P. K. and Gill, R. D. (1982). Cox's regression model for counting process: a large sample study. *Annals of Statistics*, **10**, 1100–20.

Birkes, D. and Dodge, Y. (1993). *Alternative Methods of Regression.* John Wiley and Sons, New York.

Boch, R. D. and Aitkin, M. (1981). Marginal mximum likelihood estimation of item parameters: application of an EM algorithm. *Psychometrika*, **46**, 443–59.

Boch, R. D. and Lieberman, M. (1970). Fitting a response model for n dichotomously scored items. *Psychometrika*, **35**, 179–97.

Box, G. E. P. and Tiao, G. C. (1973). *Bayesian inference in statistical models.* Addison-Wesley, Reading, Massachusetts.

Breslow, N. E. (1984). Extra-poisson variation in log-linear models. *Applied Statistics*, **33**, 38–44.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal American Statistical Association*, **88**, 9–25.

Brownlee, K. A. (1965). *Statistical Theory and Methodology in Science and Engineering.* John Wiley and Sons, New York.

Carlin, J. B. (1992). Meta-analysis for 2 x 2 tables: a Bayesian approach. *Statistics in Medicine*, **11**, 141–59.

Clayton, D. (1994). Bayesian analysis of frailty models. Technical report, Medical Research Council Biostatistics Unit, Cambridge.

Clayton, D. G. (1991). A Monte Carlo method for Bayesian inference in frailty models. *Biometrics*, **47**, 467–85.

Crowder, M. J. (1978). Beta-binomial Anova for proportons. *Applied Statistics*, **27**, 34–7.

Davies, O. L. (1967). *Statistical methods in research and production.* Oliver and Boyd, London.

Dellaportas, P. and Smith, A. F. M. (1993). Bayesian inference for generalized linear and proportional hazards model via Gibbs sampling. *Applied Statistics*, **42**, 443–60.

Ezzet, F. and Whitehead, J. (1993). A random effects model for ordinal responses from a crossover trial. *Statistics in Medicine*, **10**, 901–7.

Frierich, E., Gehan, E., *et al.* (1963). The effect of 6-mercaptopurine on the duration of steroid-induced remissions in acute leukaemia: a model for evaluation of other potentially useful therapy. *Blood*, **21**, 699–716.

Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. M. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association*, **85**, 972–85.

Gelfand, A. E., Sahu, S. K., and Carlin, B. P. (1995). Efficient parameterizations for normal linear models. *Biometrika*, **82**, 479–88.

George, E., Makov, U., and Smith, A. (1993). Conjugate likelihood distributions. *Scandinavian Journal of Statistics*, **20**, 147–56.

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for gibbs sampling. *Applied Statistics*, **41**, 337–48.

Grieve, A. P. (1987). Applications of Bayesian software: two examples. *Statistician*, **36**, 283–8.

Kalbfleisch, J. (1978). Non-parametric Bayesian analysis of survival time data. *Journal of the Royal Statistical Society, Series B*, **40**, 214–21.

Kalbfleisch, J. and Prentice, R. (1980). *The statistical analysis of failure time data*. Wiley, New York.

Lindley, D. V. and Smith, A. F. M. (1972). Bayes estimates for the linear model (with discussion). *Journal of the Royal Statistical Society, Series B*, **34**, 1–44.

McGilchrist, C. and Aisbett, C. (1991). Regression with frailty in survival analysis. *Biometrics*, **47**, 461–6.

Peto, R. (1972). Contribution to the discussion of 'regression models and life tables'. *Journal of the Royal Statistical Society, Series B*, **34**, 187–203.

Roche, A. F., Wainer, H., and Thissen, D. (1975). *Skeletal maturity: The knee joint as a biological indicator*. Plenum, New York.

Thall, P. F. and Vail, S. C. (1990). Some covariance models for longitudinal count data with overdispersion. *Biometrics*, **46**, 657–71.

Thissen, D. (1986). *MULITLOG Version 5: User's Guide*. Scientific Software, Inc, Mooresville, IN, (5 edn).

Verdinelli, I. and Wassweman, L. (1991). Bayesian analysis of outlier problems using the gibbs sampler. *Statistics in Computing*, **1**, 105–40.

Whitehead, J. (1980). Fitting Cox's regression model to survival data using GLIM. *Appl Statist*, **29**, 268–75.

Williams, D. (1975). The analysis of binary responses from toxicological experiments involving reproduction and teratogenicity. *Biometrics*, **31**, 949–52.