Examples Volume 2

References:
Sorry - an on-line version of the references is currently unavailable.
Please refer to the existing Examples documentation available from
`http://www.mrc-bsu.cam.ac.uk/bugs.`

Dugongs: nonlinear growth curve

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

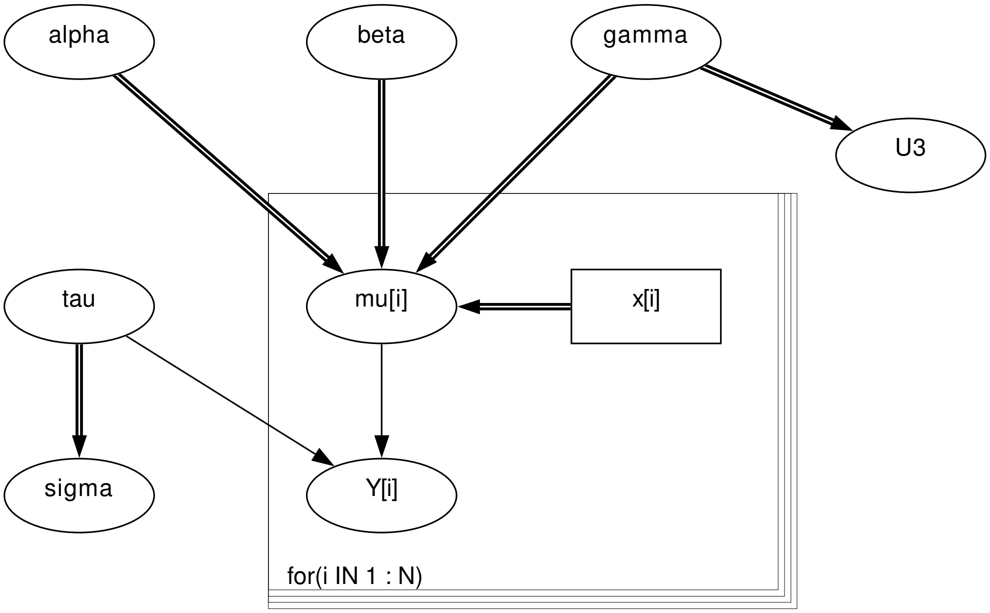
Dugong	1	2	3	4	5	....	26	27
Age (X)	1.0	1.5	1.5	1.5	2.5	....	29.0	31.5
Length (Y)	1.80	1.85	1.87	1.77	2.02	....	2.27	2.57

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

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

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

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



```
model
{
   for( i in 1 : N ) {
      Y[i] ~ dnorm(mu[i], tau)
      mu[i] <- alpha - beta * pow(gamma,x[i])
   }
   alpha ~ dnorm(0.0, 1.0E-6)
   beta ~ dnorm(0.0, 1.0E-6)
   gamma ~ dunif(0.5, 1.0)
   tau ~ dgamma(0.001, 0.001)
   sigma <- 1 / sqrt(tau)
   U3 <- logit(gamma)
}
```

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
U3	1.861	0.2678	0.01189	1.321	1.865	2.37	1001	10000
alpha	2.652	0.07094	0.003378	2.532	2.646	2.808	1001	10000
beta	0.9729	0.07649	0.001806	0.8251	0.9711	1.129	1001	10000
gamma	0.8623	0.03259	0.001393	0.7894	0.8658	0.9145	1001	10000
sigma	0.0992	0.01496	1.831E-4	0.07513	0.09742	0.1339	1001	10000

```
model
   {
      for( i in 1 : N ) {
         Y[i] ~ dnorm(mu[i], tau)
         mu[i] <- alpha - beta * pow(gamma,x[i])
      }
      alpha ~ dnorm(0.0, 1.0E-6)
      beta ~ dnorm(0.0, 1.0E-6)
      logit(gamma) <- U3
      tau ~ dgamma(0.001, 0.001)
      sigma <- 1 / sqrt(tau)
      U3 ~ dnorm(0, 1.0E-4)
   }
```

```
list(alpha = 1, beta = 1, tau = 1, U3 = 0)
```

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample

U3	1.912	0.2609	0.01072	1.415	1.904	2.459	2001	9000
alpha	2.665	0.07564	0.002835	2.544	2.655	2.848	2001	9000
beta	0.9753	0.07757	0.00325	0.8274	0.9752	1.132	2001	9000
gamma	0.8684	0.02941	0.00123	0.8046	0.8704	0.9212	2001	9000
sigma	0.09871	0.01474	2.373E-4	0.07482	0.09716	0.1321	2001	9000

Orange Trees: Non-linear growth curve

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

$$Y_{ij} \sim \text{Normal}(\eta_{ij}, \tau_C)$$

$$\eta_{ij} = \frac{\phi_{i1}}{1 + \phi_{i2} \exp(\phi_{i3} x_j)}$$

$$\theta_{i1} = \log(\phi_{i1})$$
$$\theta_{i2} = \log(\phi_{i2} + 1)$$
$$\theta_{i3} = \log(-\phi_{i3})$$

The BUGS code is as follows

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

Data (click to open)

Inits (click to open)

Results

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

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
mu[1]	5.257	0.1279	0.002334	5.002	5.256	5.505	5001	10000
mu[2]	2.211	0.1277	0.004119	1.965	2.209	2.469	5001	10000
mu[3]	-5.869	0.1091	0.004242	-6.113	-5.861	-5.676	5001	10000
sigma[1]	0.2332	0.1357	0.00204	0.08448	0.204	0.5494	5001	10000
sigma[2]	0.1383	0.1147	0.003672	0.02607	0.1078	0.4207	5001	10000
sigma[3]	0.1012	0.08341	0.002777	0.02317	0.07675	0.3234	5001	10000
sigmaC	8.065	1.244	0.03079	6.014	7.93	10.92	5001	10000

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

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
mu[1]	5.254	0.1242	0.004513	5.002	5.258	5.488	5001	10000
mu[2]	2.22	0.1252	0.007917	1.994	2.216	2.469	5001	10000
mu[3]	-5.861	0.1143	0.008563	-6.098	-5.86	-5.657	5001	10000
sigma[1]	0.2245	0.1235	0.00357	0.07706	0.1963	0.5306	5001	10000
sigma[2]	0.1342	0.1219	0.005743	0.02447	0.1009	0.4428	5001	10000
sigma[3]	0.1098	0.09349	0.005828	0.02354	0.08214	0.3591	5001	10000
sigmaC	8.025	1.216	0.03895	6.03	7.89	10.77	5001	10000
theta[1,1]	5.079	0.08832	0.007158	4.949	5.066	5.326	5001	10000
theta[1,2]	2.134	0.1542	0.01001	1.823	2.136	2.423	5001	10000
theta[1,3]	-5.851	0.149	0.0126	-6.19	-5.849	-5.583	5001	10000
theta[2,1]	5.395	0.05096	0.003465	5.3	5.393	5.505	5001	10000
theta[2,2]	2.207	0.1245	0.008209	1.962	2.205	2.46	5001	10000
theta[2,3]	-5.825	0.1015	0.007943	-6.028	-5.828	-5.624	5001	10000
theta[3,1]	5.079	0.09932	0.008296	4.945	5.06	5.356	5001	10000
theta[3,2]	2.187	0.1351	0.008393	1.915	2.188	2.447	5001	10000
theta[3,3]	-5.908	0.1494	0.01298	-6.286	-5.89	-5.666	5001	10000
theta[4,1]	5.441	0.04836	0.003287	5.347	5.442	5.543	5001	10000
theta[4,2]	2.269	0.1395	0.009928	2.024	2.256	2.566	5001	10000
theta[4,3]	-5.816	0.1021	0.008087	-6.008	-5.825	-5.591	5001	10000
theta[5,1]	5.291	0.06828	0.005157	5.174	5.284	5.438	5001	10000
theta[5,2]	2.299	0.1351	0.009323	2.05	2.295	2.589	5001	10000
theta[5,3]	-5.907	0.1075	0.008937	-6.125	-5.903	-5.7	5001	10000

Orange Trees: Non-linear growth curve

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

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

Data (click to open)

Inits (click to open)

Results

A 4000 iteration Metropolis adaptive phase plus 1000 update burn in followed by a further 10000 updates gave the parameter estimates:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
mu[1]	5.265	0.1351	0.001577	4.992	5.263	5.537	5001	10000
mu[2]	2.2	0.1656	0.002555	1.874	2.197	2.522	5001	10000
mu[3]	-5.88	0.141	0.002287	-6.171	-5.877	-5.614	5001	10000
sigma[1]	0.2581	0.1145	0.001681	0.1268	0.231	0.558	5001	10000
sigma[2]	0.2679	0.1291	0.002343	0.1191	0.2368	0.5925	5001	10000
sigma[3]	0.2296	0.1101	0.001523	0.1085	0.2036	0.5048	5001	10000
sigmaC	7.853	1.19	0.02499	5.923	7.715	10.53	5001	10000

Biopsies: discrete variable latent class model

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

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

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

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

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

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

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

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

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

Data (click to open)

Inits (click to open)

Results

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

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
error[2,1]	0.5875	0.0663	0.001731	0.4557	0.5884	0.714	1001	10000
error[2,2]	0.4125	0.0663	0.001731	0.286	0.4116	0.5444	1001	10000
error[3,1]	0.342	0.04584	7.001E-4	0.256	0.3403	0.4363	1001	10000
error[3,2]	0.03729	0.01782	2.503E-4	0.009585	0.03488	0.07774	1001	10000
error[3,3]	0.6207	0.04782	7.253E-4	0.5222	0.622	0.7107	1001	10000
error[4,1]	0.09933	0.04218	5.187E-4	0.03382	0.09397	0.1968	1001	10000
error[4,2]	0.02225	0.02302	3.867E-4	5.186E-4	0.01488	0.08594	1001	10000
error[4,3]	0.2037	0.06101	9.381E-4	0.1013	0.1984	0.3374	1001	10000
error[4,4]	0.6747	0.07271	0.001124	0.5228	0.6792	0.8044	1001	10000
p[1]	0.1529	0.04962	0.001503	0.04877	0.1551	0.2459	1001	10000
p[2]	0.3109	0.0549	0.00144	0.216	0.3066	0.4323	1001	10000
p[3]	0.3892	0.0437	6.675E-4	0.3055	0.3879	0.4775	1001	10000
p[4]	0.1471	0.0298	3.433E-4	0.094	0.1448	0.2106	1001	10000

Eyes: Normal Mixture Model

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

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

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

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

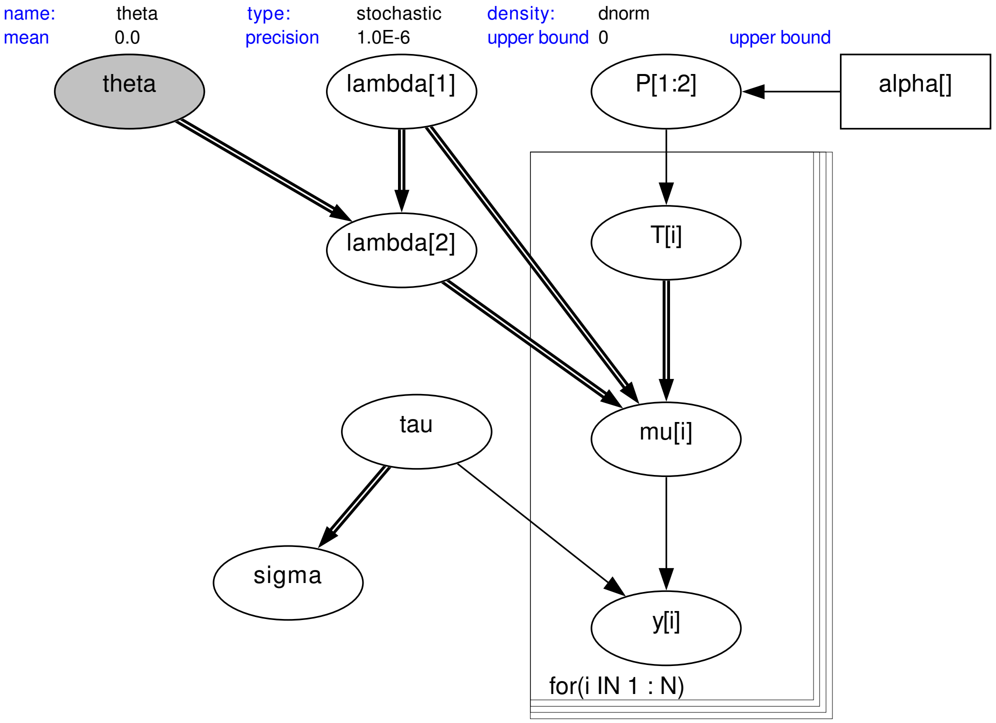
$$T_i \sim \text{Categorical}(P).$$

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

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

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

λ_1, θ, τ, P, are given independent ``noninformative" priors, including a uniform prior for P on (0,1). The appropriate graph and the BUGS code are given below.



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

Data (click to open)

Inits (click to open)

Results

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

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
P[1]	0.6014	0.08981	0.002305	0.4267	0.602	0.7701	1001	10000
P[2]	0.3986	0.08981	0.002305	0.2299	0.398	0.5733	1001	10000
lambda[1]	536.8	1.023	0.03708	535.0	536.7	539.0	1001	10000
lambda[2]	548.9	1.388	0.03856	546.0	548.9	551.3	1001	10000
sigma	3.805	0.726	0.03322	2.932	3.652	6.014	1001	10000

Hearts: a mixture model for count data

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

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

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

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

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

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

The BUGS code for this model is given below:

```
model
{
   for (i in 1 : N) {
      y[i] ~ dbin(P[state1[i]], t[i])
      state[i] ~ dbern(theta)
```

```
            state1[i] <- state[i] + 1
            t[i] <- x[i] + y[i]
            prop[i] <- P[state1[i]]
         }
         P[1] <- p
         P[2] <- 0
         logit(p) <- alpha
         alpha ~ dnorm(0,1.0E-4)
         beta <- exp(alpha)
         logit(theta) <- delta
         delta ~ dnorm(0, 1.0E-4)
      }
```

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha	-0.4809	0.2795	0.002701	-1.044	-0.4767	0.0652	1001	10000
beta	0.6427	0.1812	0.001765	0.3521	0.6208	1.067	1001	10000
delta	0.3144	0.6177	0.006344	-0.8919	0.3124	1.553	1001	10000
theta	0.5717	0.1391	0.001417	0.2907	0.5775	0.8253	1001	10000

Air: Berkson measurement error

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

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

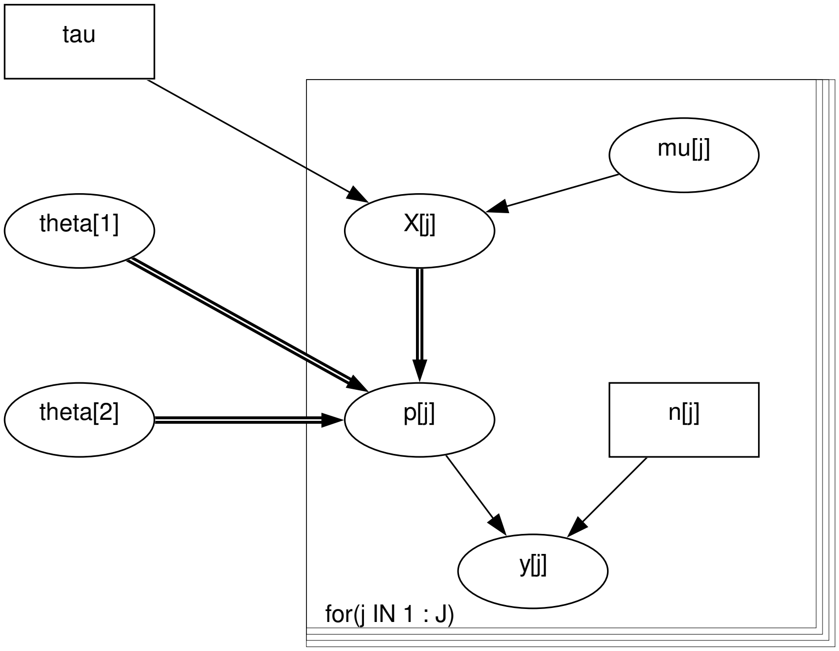
A discrete covariate z_j (j = 1,2,3) representing NO_2 concentration in the child's bedroom classified into 3 categories is used as a surrogate for true exposure. The nature of the measurement error relationship associated with this covariate is known precisely via a calibration study, and is given by

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

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

$$y_j \sim \text{Binomial}(p_j, n_j)$$
$$\text{logit}(p_j) = \theta_1 + \theta_2 x_j$$

where p_j is the probability of respiratory illness for children in the jth exposure group. The regression coefficients θ_1 and θ_2 are given vague independent normal priors. The graphical model is shown below:



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

Data (click to open)

Inits (click to open)

Results

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

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
X[1]	12.92	7.877	0.4227	-3.775	13.3	26.96	1001	10000
X[2]	27.21	7.473	0.1946	13.05	27.01	42.63	1001	10000
X[3]	40.85	8.721	0.3502	24.18	40.84	58.37	1001	10000
theta[1]	-0.9628	1.0	0.08808	-4.233	-0.7183	0.2104	1001	10000

Re-parameterised model with centred covariates:

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

Inits (click to open)

Results

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

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
X[1]	13.27	8.04	0.4047	-3.199	13.57	28.24	1001	10000
X[2]	27.28	7.455	0.1798	12.69	27.2	42.06	1001	10000
X[3]	41.03	8.468	0.2267	25.39	40.83	58.25	1001	10000
theta[1]	-0.9269	0.7985	0.05205	-3.068	-0.7581	0.206	1001	10000
theta[2]	0.04033	0.02995	0.001308	0.008708	0.04105	0.1133	1001	10000

Cervix: case - control study with errors in covariates

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

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

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

$$d_i \sim \text{Bernoulli}(p_i) \quad i = 1,...,2044$$
$$\text{logit}(p_i) = \beta_{0C} + \beta x_i \quad i = 1,...,2044$$

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

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

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

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

The BUGS code is given below. The role of the variables `x1` and `d1` is to pick the appropriate value of ϕ (the incidence of w) for any given true exposure status x and disease status d. Since x and d take the values 0 or 1, and the subscripts for ϕ take values 1 or 2, we must first add 1 to each `x[i]` and `d[i]` in the BUGS code before using them as index values for ϕ. BUGS does not allow subscripts to be functions of variable quantities --- hence the need to create `x1`and `d1` for use as subscripts. In addition, note that γ_1 and γ_2 were not simulated directly in BUGS, but were calculated as functions of other parameters. This is because the dependence of γ_1 and γ_2 on d would have led to a cycle in the graphical model which would no longer define a probability distribution.

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

Data (click to open)

Inits (click to open)

Results

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

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
beta0C	-0.9265	0.207	0.01102	-1.357	-0.9144	-0.5501	1001	10000
gamma1	0.4371	0.05431	0.002994	0.3286	0.4372	0.5451	1001	10000
gamma2	0.5969	0.06438	0.003332	0.4727	0.5948	0.731	1001	10000
phi[1,1]	0.3177	0.05363	0.002669	0.2142	0.3187	0.4168	1001	10000
phi[1,2]	0.2138	0.08148	0.004308	0.07201	0.209	0.3849	1001	10000
phi[2,1]	0.5694	0.06434	0.002928	0.4461	0.5696	0.691	1001	10000
phi[2,2]	0.7623	0.06328	0.003054	0.6371	0.7643	0.8789	1001	10000
q	0.4943	0.04017	0.002071	0.4135	0.494	0.5728	1001	10000

Re-parameterised model with centred covariates:

```
model
{
   for (i in 1 : N) {
      x[i]   ~ dbern(q)        # incidence of HSV
      logit(p[i]) <- beta0 + beta * (x[i] - mean(w[])) # logistic model
      d[i]  ~ dbern(p[i])       # incidence of cancer
      x1[i] <- x[i] + 1
      d1[i] <- d[i] + 1
      w[i]  ~ dbern(phi[x1[i], d1[i]])     # incidence of w
   }
   q      ~ dunif(0.0, 1.0)          # prior distributions
```

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

Inits (click to open)

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
beta0C	-0.921	0.2036	0.0114	-1.327	-0.9178	-0.5276	1001	10000
gamma1	0.4389	0.05766	0.003321	0.3274	0.4396	0.5513	1001	10000
gamma2	0.5964	0.0635	0.003451	0.4721	0.5967	0.719	1001	10000
phi[1,1]	0.318	0.05831	0.003108	0.2003	0.319	0.4263	1001	10000
phi[1,2]	0.221	0.08396	0.004839	0.0738	0.2146	0.3988	1001	10000
phi[2,1]	0.5664	0.0666	0.003198	0.4325	0.5682	0.6918	1001	10000
phi[2,2]	0.7585	0.06472	0.003465	0.6282	0.7603	0.8797	1001	10000
q	0.4953	0.04198	0.002285	0.4138	0.4953	0.5773	1001	10000

Birats: a bivariate normal hierarchical model

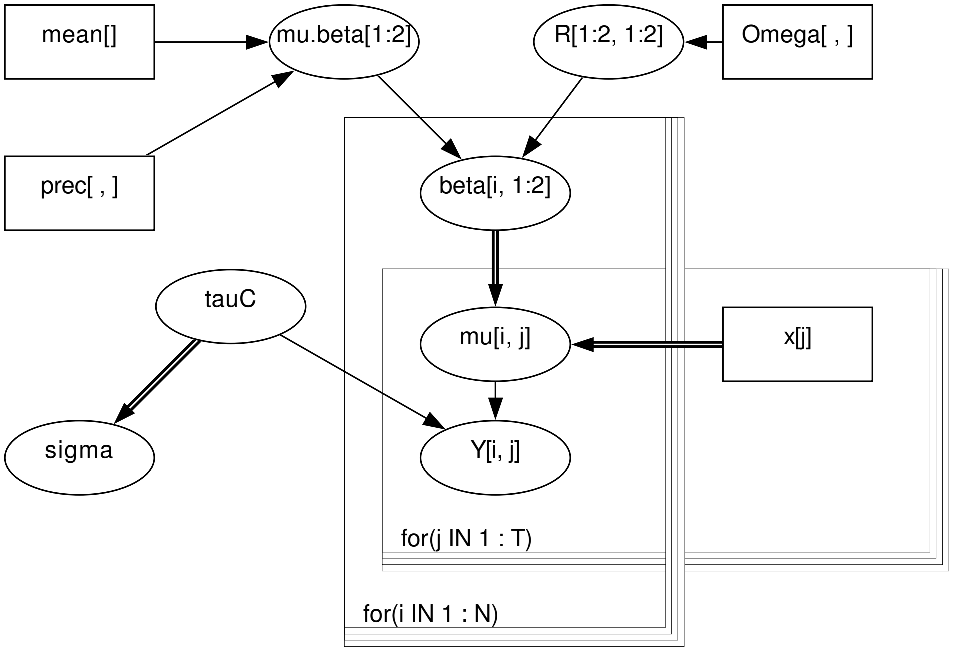
We return to the Rats example, and illustrate the use of a multivariate Normal (MVN) population distribution for the regression coefficients of the growth curve for each rat. This is the model adopted by Gelfand etal (1990) for these data, and assumes *a priori* that the intercept and slope parameters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

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

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

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

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



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

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

Data (click to open)

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
mu.beta[1]	106.6	2.35	0.0335	101.8	106.6	111.2	1001	10000
mu.beta[2]	6.185	0.1062	0.001351	5.981	6.185	6.397	1001	10000
sigma	6.136	0.4781	0.009095	5.283	6.1	7.137	1001	10000

Schools: ranking school examination results using multivariate hierarcical models

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

Data

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

Model

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

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

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

Prior distributions

The fixed effects β_k (k=1,...,8), θ and ϕ were assumed to follow vague independent Normal distributions with zero mean and low precision = 0.0001. The random school-level coefficients

α_{kj} (k = 1,2,3) were assumed to arise from a multivariate normal population distribution with unknown mean $\boldsymbol{\gamma}$ and covariance matrix $\boldsymbol{\Sigma}$. A non-informative multivariate normal prior was then specified for the population mean $\boldsymbol{\gamma}$, whilst the inverse covariance matrix $\mathbf{T} = \boldsymbol{\Sigma}^{-1}$ was assumed to follow a Wishart distribution. To represent vague prior knowledge, we chose the the degrees of freedom for this distribution to be as small as possible (i.e. 3, the rank of $\mathbf{T}$). The scale matrix **R** was specified as

$$\begin{vmatrix} 0.1 & 0.005 & 0.005 \\ 0.005 & 0.01 & 0.005 \\ 0.005 & 0.005 & 0.01 \end{vmatrix}$$

which represents our prior guess at the order of magnitude of $\boldsymbol{\Sigma}$.

The BUGS code is given below:

```
model
{
   for(p in 1 : N) {
      Y[p] ~ dnorm(mu[p], tau[p])
      mu[p] <- alpha[school[p], 1] + alpha[school[p], 2] * LRT[p]
         + alpha[school[p], 3] * VR[p, 1] + beta[1] * LRT2[p]
         + beta[2] * VR[p, 2] + beta[3] * Gender[p]
         + beta[4] * School.gender[p, 1] + beta[5] * School.gender[p, 2]
         + beta[6] * School.denom[p, 1] + beta[7] * School.denom[p, 2]
         + beta[8] * School.denom[p, 3]
      log(tau[p]) <- theta + phi * LRT[p]
      sigma2[p] <- 1 /  tau[p]
      LRT2[p] <- LRT[p] * LRT[p]
    }
    min.var <- exp(-(theta + phi * (-34.6193))) # lowest LRT score = -34.6193
    max.var <- exp(-(theta + phi * (37.3807)))  # highest LRT score = 37.3807

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

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

# Hyper-priors:
```

```
         gamma[1 : 3] ~ dmnorm(mn[1:3 ], prec[1:3 ,1:3 ]);
         T[1 : 3, 1 : 3 ] ~ dwish(R[1:3 ,1:3 ], 3)
      }
```

Data (click to open)

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

Inits (click to open)

Results

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

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
beta[1]	2.64E-4	9.842E-5	2.429E-6	7.499E-5	2.638E-4	4.558E-4	1001	10000
beta[2]	0.4219	0.06225	0.00344	0.301	0.4206	0.5466	1001	10000
beta[3]	0.1725	0.04834	0.001427	0.07689	0.1722	0.2682	1001	10000
beta[4]	0.125	0.1377	0.006001	-0.1558	0.1276	0.3939	1001	10000
beta[5]	0.06201	0.1038	0.004941	-0.1475	0.06323	0.2624	1001	10000
beta[6]	-0.2769	0.1875	0.007158	-0.6584	-0.2728	0.08729	1001	10000
beta[7]	0.1441	0.1061	0.004271	-0.05912	0.1428	0.36	1001	10000
beta[8]	-0.1667	0.1733	0.006393	-0.4943	-0.1675	0.1846	1001	10000
gamma[1]	-0.6778	0.09568	0.005593	-0.8668	-0.6783	-0.4862	1001	10000
gamma[2]	0.03135	0.01019	1.396E-4	0.01139	0.03139	0.05137	1001	10000
gamma[3]	0.9597	0.08626	0.004849	0.7947	0.959	1.129	1001	10000
phi	-0.002605	0.002829	3.159E-5	-0.008146	-0.00259	0.002927	1001	10000
theta	0.5801	0.03205	3.518E-4	0.5163	0.5803	0.6414	1001	10000

Estimating the ranks

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

BUGS

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

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

i	age_i	$year_i$	$cases_i$	$person\text{-}years_i$
1	1	6	2	41380
2	1	7	0	43650
...	...			
77	13	5	31	13600

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

$$\begin{aligned} cases_i &\sim \text{Poisson}(\mu_i) \\ \log \mu_i &= \log\ person\text{-}years_i + \alpha_{age_i} + \beta_{year_i} \\ \beta_k &\sim \text{Normal}(0, \tau) \end{aligned}$$

Autoregressive smoothing of relative risks

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

$$\begin{aligned} cases_i &\sim \text{Poisson}(\mu_i) \\ \log \mu_i &= \log\ person\text{-}years_i + \alpha_{age_i} + \beta_{year_i} \\ \beta_1 &\sim \text{Normal}(0, 0.000001\tau) \\ \beta_2 \mid \beta_1 &\sim \text{Normal}(0, 0.000001\tau) \\ \beta_k \mid \beta_{1,\ldots,k-1} &\sim \text{Normal}(2\,\beta_{k-1} - \beta_{k-2}, \tau) \qquad k > 2 \end{aligned}$$

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

For computational reasons Breslow and Clayton impose constraints on their random effects β_k in order that their mean and linear trend are zero, and counter these constraints by introducing a linear term b x year$_i$ and allowing unrestrained estimation of α_j. Since we allow free movement of the β's we dispense with the linear term, and impose a "corner" constraint α_1 =0 .

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

Data (click to open)

Inits (click to open)

Results

A 1000 update burn in followed by a further 100000 updates gave the parameter estimates

```
            mean      sd        MC_error  val2.5pc  median    val97.5pc start     sample
logRR[1]    -1.075    0.2503    0.008578  -1.619    -1.043    -0.6951   1001      100000
logRR[2]    -0.7717   0.1584    0.005506  -1.107    -0.755    -0.5203   1001      100000
logRR[3]    -0.4721   0.08179   0.002555  -0.651    -0.463    -0.338    1001      100000
logRR[4]    -0.2016   0.03908   6.68E-4   -0.278    -0.2018   -0.1166   1001      100000
logRR[6]    0.1588    0.04625   0.001162  0.04592   0.1683    0.2269    1001      100000
logRR[7]    0.319     0.06949   0.002112  0.164     0.3282    0.4369    1001      100000
logRR[8]    0.4829    0.08673   0.002982  0.3022    0.4896    0.6469    1001      100000
logRR[9]    0.6512    0.1066    0.003936  0.4366    0.6566    0.8613    1001      100000
logRR[10]   0.8466    0.1281    0.00484   0.5911    0.8513    1.094     1001      100000
logRR[11]   1.059     0.1811    0.006206  0.7041    1.06      1.415     1001      100000
sigma       0.05286   0.04374   0.001335  0.006732  0.04159   0.1625    1001      100000
```

Beetles: choice of link function

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

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

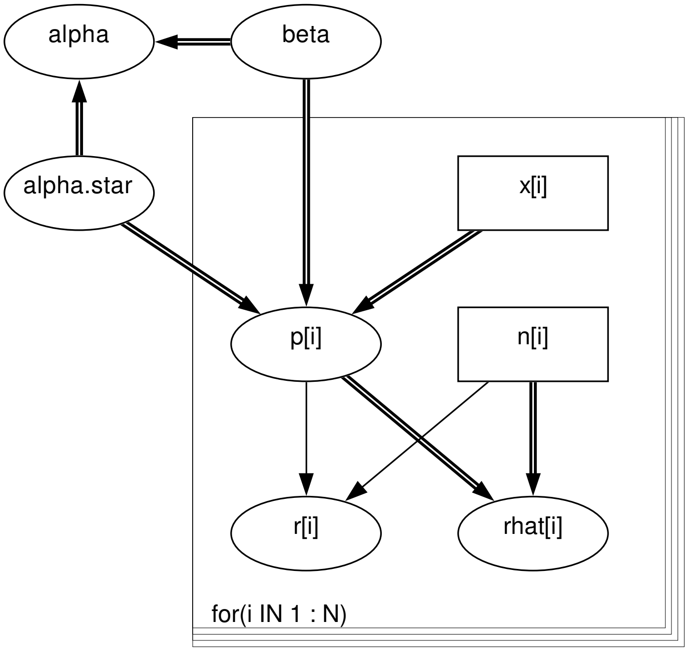
We assume that the observed number of deaths r_i at each concentration x_i is binomial with sample size n_i and true rate p_i. Plausible models for pi include the logistic, probit and extreme value (complimentary log-log) models, as follows

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

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

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

The corresponding graph is shown below:



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

Data (click to open)

Inits (click to open)

Results

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

Logit model

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
alpha	-60.79	5.147	0.05624	-71.29	-60.67	-51.17	1001	10000
beta	34.31	2.893	0.03171	28.91	34.24	40.23	1001	10000
rhat[1]	3.56	0.9488	0.009435	1.997	3.463	5.634	1001	10000
rhat[2]	9.932	1.677	0.01549	6.909	9.851	13.43	1001	10000
rhat[3]	22.47	2.091	0.01736	18.36	22.46	26.63	1001	10000
rhat[4]	33.87	1.751	0.0152	30.32	33.89	37.25	1001	10000
rhat[5]	50.03	1.646	0.01661	46.66	50.06	53.12	1001	10000
rhat[6]	53.21	1.102	0.01191	50.86	53.28	55.17	1001	10000
rhat[7]	59.14	0.7338	0.008143	57.52	59.22	60.38	1001	10000
rhat[8]	58.68	0.4241	0.004761	57.72	58.74	59.36	1001	10000

Probit model

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
alpha	-35.04	2.646	0.02934	-40.46	-34.98	-29.93	1001	10000
beta	19.79	1.488	0.01657	16.9	19.75	22.84	1001	10000
rhat[1]	3.442	1.014	0.0106	1.743	3.348	5.693	1001	10000
rhat[2]	10.76	1.692	0.01684	7.632	10.7	14.23	1001	10000
rhat[3]	23.48	1.916	0.01865	19.79	23.47	27.24	1001	10000
rhat[4]	33.81	1.626	0.01706	30.58	33.83	36.96	1001	10000
rhat[5]	49.6	1.648	0.01865	46.27	49.63	52.73	1001	10000
rhat[6]	53.27	1.17	0.01353	50.76	53.33	55.38	1001	10000
rhat[7]	59.6	0.7542	0.008725	57.88	59.67	60.84	1001	10000
rhat[8]	59.17	0.3729	0.004308	58.28	59.23	59.72	1001	10000

Extreme value (cloglog) model

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
alpha	-39.77	3.221	0.02839	-46.41	-39.68	-33.74	1001	10000
beta	22.15	1.788	0.01573	18.81	22.1	25.85	1001	10000
rhat[1]	5.623	1.119	0.01006	3.63	5.551	8.055	1001	10000
rhat[2]	11.28	1.581	0.01461	8.351	11.21	14.52	1001	10000
rhat[3]	20.91	1.891	0.0183	17.29	20.89	24.66	1001	10000
rhat[4]	30.32	1.666	0.01688	26.98	30.33	33.56	1001	10000
rhat[5]	47.74	1.74	0.01713	44.21	47.77	51.01	1001	10000
rhat[6]	54.08	1.231	0.01134	51.48	54.15	56.25	1001	10000
rhat[7]	61.02	0.5304	0.004795	59.75	61.12	61.77	1001	10000
rhat[8]	59.92	0.09563	9.349E-4	59.66	59.95	60.0	1001	10000

Endo: conditional inference in case-control studies

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

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

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

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

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

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

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

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

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

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

All these techniques are illustrated in the code given below, which includes a transformation of the original summary statistics into full data. In this example, all but the second conditional-likelihood approach are commented out.

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

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

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

Data (cklick to open)

Inits (cklick to open)

Results

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

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
beta	1.871	0.4123	0.009414	1.111	1.844	2.761	5001	10000

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

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

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

Note the repeated x's.

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

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

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

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

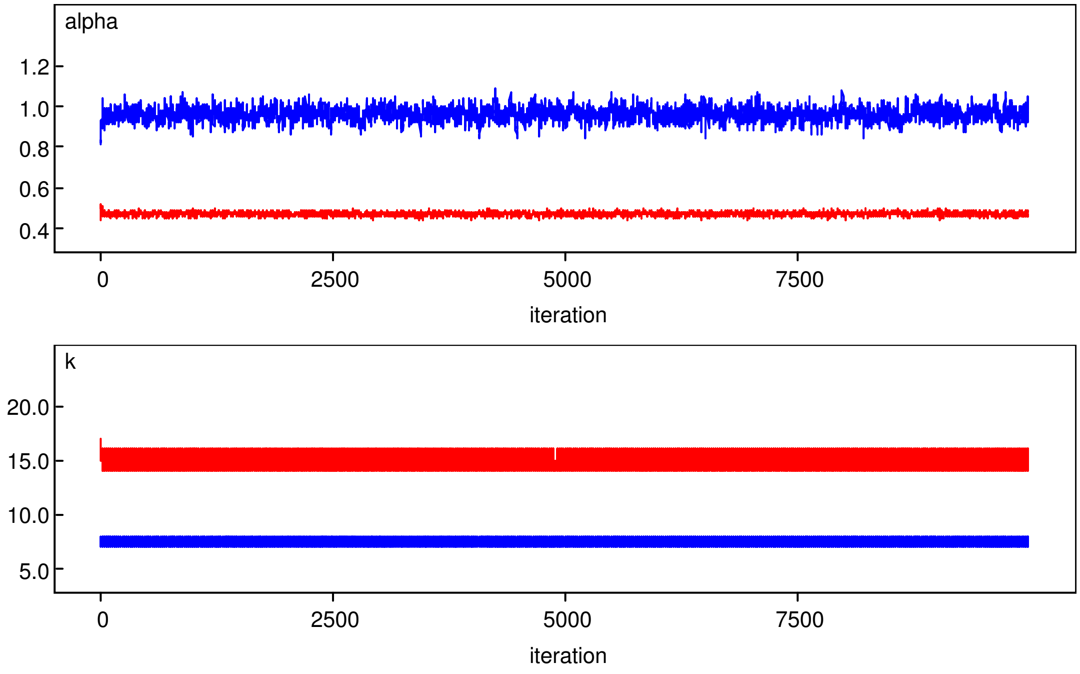
Note way of constructing a uniform prior on the integer k, and making the regression parameter depend on a random changepoint.

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

Data (click to open)

Inits for chain 1 Inits for chain 2(click to open)

Traces of two chains shows complete dependence on starting values



Results are hopeless - no mixing at all.

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

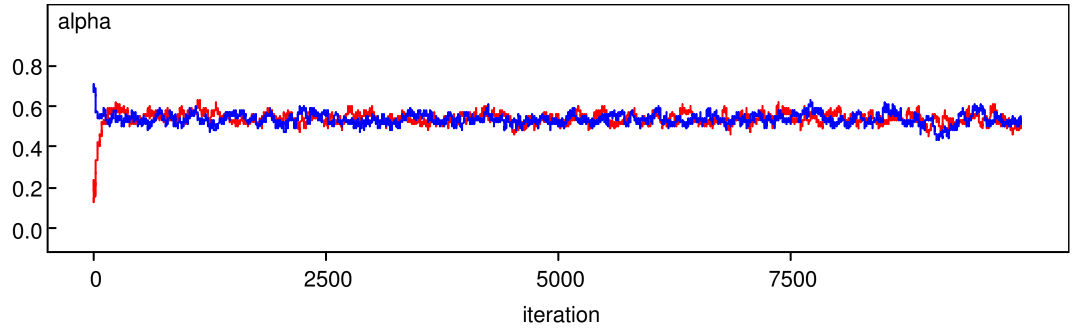
TRY USING CONTINUOUS PARAMETERISATION

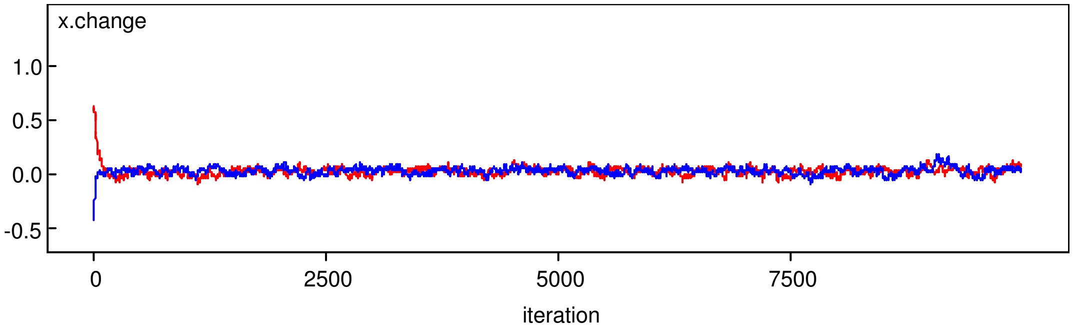
```
model
{
   for(i in 1 : N) {
    Y[i] ~ dnorm(mu[i], tau)
      mu[i] <- alpha + beta[J[i]] * (x[i] - x.change)
      J[i] <- 1 + step(x[i] - x.change)
   }
   tau ~ dgamma(0.001, 0.001)
   alpha ~ dnorm(0.0,1.0E-6)
   for(j in 1 : 2) {
   beta[j] ~ dnorm(0.0,1.0E-6)
   }
  sigma <- 1 / sqrt(tau)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

Data (click to open)

Inits for chain 1 Inits for chain 2(click to open)

Results





	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha	0.537	0.02569	0.001316	0.4895	0.535	0.5881	1001	20000
beta[1]	-0.4184	0.01511	6.303E-4	-0.4468	-0.419	-0.3876	1001	20000
beta[2]	-1.014	0.01747	5.38E-4	-1.049	-1.013	-0.9799	1001	20000
sigma	0.0221	0.003271	3.919E-5	0.0168	0.02171	0.02952	1001	20000
x.change	0.02597	0.03245	0.001668	-0.03754	0.02868	0.0839	1001	20000

Not wonderful mixing, but reasonable

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

Strong correlation of alpha and changepoint

alpha	x.change	-0.932941

Asia: expert system

Evidence propagation

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

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

Data (click to open)

Inits (click to open)

Results

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
bronchitis	1.811	0.3918	0.001409	1.0	2.0	2.0	10001	100000
either	1.185	0.3885	0.001287	1.0	1.0	2.0	10001	100000
lung.cancer	1.101	0.3011	0.001006	1.0	1.0	2.0	10001	100000
smoking	1.628	0.4833	0.001764	1.0	2.0	2.0	10001	100000
tuberculosis	1.089	0.2854	9.782E-4	1.0	1.0	2.0	10001	100000
xray	1.223	0.4161	0.00135	1.0	1.0	2.0	10001	100000