Partitioning variance with WinBUGS

As you have already guessed, there is a way to partition variance with WinBUGS. One of the advantages of using WinBUGS for variance partitioning is that it is easy to get direct estiamtes of both the observational and the causal components of variance in a quantitative genetics design.[1]

The data has a fairly straightforward structure, although it will look different from other WinBUGS data you've seen:

```
sire[] dam[] weight[]
1 1 99.2789343018915
1 1 92.8290811609914
1 1 92.1977611557311
1 1 96.2203721102781
1 1 97.435349690979
1 1 99.5683756647001
1 1 92.2818423839868
1 1 99.5110665900414
1 2 103.230303160319
.    .    .
.    .    .
.    .    .
7 52 102.100956942292
7 53 117.842216117441
7 53 124.174111459722
7 53 115.02591859104
7 53 111.885901553834
7 53 115.320308219732
```

The first column is the number of the sire involved in the mating, the second column is the number of the dam involved in the mating, and the last column is the weight of an individual offspring.

Recall that our objective is to describe how much of the overall variation is due to weight differences among individuals that share the same mother, how much is due to differences among mothers in the average weight of their offspring, and how much is due to differences among fathers in the average weight of their offspring. One way of approaching this is to imagine that the weight of each individual is determined as follows:

$$\text{weight}_i = \text{mean(weight)} + \alpha(\text{sire contribution}) + \beta(\text{dam contribution}) + \text{error} \quad ,$$

where α, β, and error are normally distribtued random variables with mean 0 and variances of V_s, V_d, and V_w, respectively. We can translate that into WinBUGS code like this:

[1] I'll only describe the observational component here. Getting the causal components is left as an exercise to be completed as part of Problem #4.

```
# sires
for (s in 1:6) {
   alpha[s] ~ dnorm(0.0, tau.s)
}
alpha[7] <- -sum(alpha[1:6])

# dams
for (d in 1:52) {
   beta[d]  ~ dnorm(0.0, tau.d)
}
beta[53] <- -sum(beta[1:52])

# offspring
for (i in 1:300) {
   mu[i] <- nu + alpha[sire[i]] + beta[dam[i]]
   weight[i] ~ dnorm(mu[i], tau.within)
}

# variance components
v.within <- 1.0/tau.within
v.dam    <- 1.0/tau.dam
v.sire   <- 1.0/tau.sire
```

Then all we need to do is to put appropriate priors on `tau`, `tau.d`, and `tau.s`.[2]

```
# Priors
nu ~ dnorm(0.0,0.001)
tau.within ~ dgamma(0.001,0.001)
tau.sire ~ dgamma(0.001,0.001)
tau.dam ~ dgamma(0.001,0.001)
```

If you put this all together in a WinBUGS model and specify priors of `tau.within = 1`, `tau.sire = 1`, and `tau.dam = 1`, then WinBUGS will produces these results:[3]

`v.sire`	5.634(0.4923,20.58)
`v.dam`	4.543(2.2960,8.24)
`v.within`	11.45(9.582,13.63)

[2] For some reason I don't understand, the authors of WinBUGS decided to specify the normal distribution using the reciprocal of the variance instead of the variance itself.

[3] I would recommend using 10,000 iterations at each step of the analysis. This one seems to take a while to settle down.