9.1 Analysis of Variance models

Assume a continuous response variable y with categorical explanatory variable x (i.e. factor). Assume x has an influence on expected values of y, hence there would be different mean values for different categories of x. The variance is assumed to be σ^2. The mean value for category a ($a = 1, \ldots, L$) is

$$\mu_a = \mu_0 + \alpha_a$$

This could be used if the data were reported as some number of measurements per each category (group), hence we could write a loop over the groups (data in tabular format). Alternatively, we could write the model per each individual measurement so that we loop over the individuals (data in individual format). The model for individual measurements $i = 1, \ldots, n$ is

$$y_i \sim \mathrm{N}(\mu_i, \sigma^2) \qquad\qquad \mu_i = \mu_0 + \alpha_{a_i}$$

where the prior is needed for σ^2, μ_0 and $\alpha_1, \ldots, \alpha_L$. To make the model identifiable, either (CR) corner constraints or (STZ) sum-to-zero constraints are needed.

(CR): define $\alpha_1 = 0$
(STZ): define $\sum_{a=1}^{L} \alpha_a = 0$ by setting $\alpha_1 = -\sum_{a=2}^{L} \alpha_a$ so that $\mu_0 = \frac{1}{L} \sum \mu_a$.

The interpretation of these two parameterizations is different. In (CR) μ_0 represents the mean of the 'reference category' (the 'corner') and each $\alpha_a, a \geq 2$ represents the effect with respect to that. In (STZ) μ_0 represents the overall mean effect in all categories.

Note: it would also be possible to parameterize the model without μ_0. Then, we would simply have $\alpha_1, \ldots, \alpha_L$ to be estimated, and we would not need CR or STZ constraints. The model would be identifiable directly. Each parameter would then represent the mean of the corresponding category itself. Depending on application context, any of these parameterizations could be preferred.

9.1.1 Example: school tutors

(BMUW, p. 171). A group of 25 students randomly divided into 4 classes, each with different tutor. Student grades were then recorded. Model and data below. Parameters α_i represent tutor effects with respect to the reference group (CR) or to the overall mean (STZ).

```
model{
 for(i in 1:n){
  mu[i] <- m + alpha[class[i]]
  grade[i]~ dnorm(mu[i],tau)
  }
  # stz constraints:
  alpha[1] <- -sum(alpha[2:tutors])
  # cr constraints:
  # alpha[1] <- 0.0
  # priors:
  m ~ dnorm(0,0.001)
  for(i in 2:tutors){alpha[i]~dnorm(0,0.001)}
  tau ~ dgamma(0.001,0.001)
  s <- sqrt(1/tau)
}
 # inits:
 list(m=1,alpha=c(NA,0,0,0),tau=1)
 # data:
 list( n=25, tutors=4, grade=c(84,58,100,51,28,89,97,50,76,83,45,42,83,64,47,
 83,81,83,34,61,77,69,94,80,55,79),
 class=c(1,1,1,1,1,1,2,2,2,2,2,2,2,3,3,3,3,3,3,3,4,4,4,4,4,4))
```

Note: in setting initial values, `NA` is used for the parameter that is not defined as random in the model. Try computing the model and check the results.

9.1.2 Two-way ANOVA

The previous model is extended to accommodate additional categorical explanatory variables. Assume these are A and B with levels L_A and L_B. Then:

$$\mu_{a,b} = \mu_0 + \alpha_a + \beta_b$$

So that for an individual measurement we have:

$$y_i \sim \mathrm{N}(\mu_i, \sigma^2) \qquad\qquad \mu_i = \mu_0 + \alpha_{a_i} + \beta_{b_i}$$

(CR) Corner constraints: $\alpha_1 = \beta_1 = 0$ so that $\mu_{1,1} = \mu_0$
(STZ) Sum-to-zero constraints: $\sum \alpha_a = \sum \beta_b = 0$ by setting $\alpha_1 = -\sum_{a=2}^{L_A} \alpha_a$ and $\beta_1 = -\sum_{b=2}^{L_B} \beta_b$ so that $\mu_0 = \frac{1}{L_A L_B} \sum \sum \mu_{a,b}$

9.1.3 Two-way interaction

$$\mu_{a,b} = \mu_0 + \alpha_a + \beta_b + \gamma_{a,b}$$

CR: $\gamma_{1,b} = \gamma_{a,1} = 0$ for all a and b.
STZ: $\sum_{a=1} \gamma_{a,b} = \sum_{b=1} \gamma_{a,b} = 0$, by setting $\gamma_{a,1} = -\sum_{b=2} \gamma_{a,b}$ for all $a = 1, 2, \ldots, L_A$ and $\gamma_{1,b} = -\sum_{a=2} \gamma_{a,b}$ for all $b = 2, 3, \ldots, L_B$.

Effectively, $L_A + L_B - 1$ parameters are replaced by deterministic assignments.

Corner constraints are often used for somewhat simpler construction.

Example: Schizotypal personality data (BMUW, p. 178). We have two factors: 'Economic status' and 'Gender'. The response is 'SPQ score' from schizotypal personality questionnaire. To ignore the third factor ('city vs. village') we just incorporate it as two repeated measurements K=1,2 below:

	Male		Female	
Econ stat.	K=1	K=2	K=1	K=2
low	9	14	18	29
medium	25	26	22	25
high	23	24	12	13

Hence:

```
list(LA=3,LB=2,K=2,
y=structure(.Data=c(9,14,18,29,25,26,22,25,23,24,12,13),
.Dim=c(3,2,2)))
```

The model code (data in tabular format) has this structure:

```
for (a in 1:LA){
   for (b in 1:LB){
       for (k in 1:K){
           y[a,b,k] ~ dnorm( mu[a,b,k], tau )
           mu[a,b,k] <- mu0 + econ[a] + gender[b] + econ.gender[a,b]
       }}}
       #### CR Constraints
       econ[1] <- 0.0
       gender[1] <- 0.0
       econ.gender[1,1] <-0.0
       for (a in 2:LA){econ.gender[a,1]<-0.0}
       for (b in 2:LB){econ.gender[1,b]<-0.0}
```

We might also have different numbers of observations in each category, but here we just had $K = 2$ in each. Results: both gender and econ.stat effects are far away from zero. Both interaction terms are far away from zero. No evidence to say there's a difference between interaction terms (might be assumed to be equal, to simplify).

Individual data format for the same model:

```
for(i in 1:n){
y[i] ~ dnorm(mu[i],tau)
mu[i] <- mu0 + alpha[a[i]] + beta[b[i]] + gamma[a[i],b[i]]
}
```

Here the loop is over all individuals, for which we need to know the indexing of categories a_i, b_i, to select the appropriate parameter value.

10 Categorical variables in normal models

(ch 6 in BMUW). Use of dummy variables (or: indicator variables) to select which parameter to apply with which measurement. For example, using CR constraints with L_A categories for factor A:

$$\mu_i = \mu_0 + \alpha_2 D^A_{i2} + \alpha_3 D^A_{i3} + \ldots + \alpha_{L_A} D^A_{iL_A}$$

so that the dummy variable $D_{il} = 1$ for the category l that is needed to go with the measurement i, and $D_{ia_i} = 0$ for $a_i \neq l$. Hence, for each measurement we have a dummy vector D containing a single one to mark the correct category, and zeros elsewhere. The first category is the corner, so that we require the constraint $\alpha_1 = 0$, to make μ_0 to be identifiable as the mean of the corner category. Hence, notice that only $L_A - 1$ dummies are needed to be defined and the set of parameters of interest is $(\mu_0, \alpha_2, \ldots, \alpha_{L_A}) = (\beta_0, \beta_1, \ldots, \beta_{L_A})$. In WinBUGS, the dummies can be either given in the data list or defined as part of the model code. Assume that the categorical variables are given as `a[i]` for measurement i, so we can use this:

```
for(i in 1:n){
D.A2 <- equals(a[i],2)
...
D.ALA <- equals(a[i],LA)
}
```

or using matrix of size $n \times L_A$:

```
for(i in 1:n){
  for(l in 1:LA){
  D.A[i,l] <- equals(a[i],l)
  }
}
```

In this way, we have defined L_A dummy variables, but only $L_A - 1$ are needed for the linear expression of μ_i. Hence, we only use $L_A - 1$ columns of matrix `D.A[]`, excluding its first column.

The dummy variable definition is a bit more complicated if we choose to use STZ constraints: $D^A_{il} = 1$ if $a_i = l$ to pick the parameter for category l, and $D^A_{il} = 0$ otherwise, except set $D^A_{il} = -1$ if $a_i = 1$. Then we have $\mu_i = \mu_0 - \alpha_2 - \ldots - \alpha_{L_A}$ for the 1st category. In WinBUGS code, the STZ dummies can be derived by:

```
for(i in 1:n){
  for(l in 1:LA){
  DSTZ[i,l] <- equals(a[i],l) - equals(a[i],1)
  }
}
```

Or by making use of the already defined CR dummies:

```
for(i in 1:n){
  for(l in 1:LA){
  DSTZ[i,l] <- D[i,l] - D[i,1]
  }
}
```

where we exploit `D[i,1]` that is otherwise redundant for the CR dummies.

The following table clarifies the use of dummy variables (BMUW, p. 192), with three categorical variables, with 2, 3 and 2 levels, respectively. See e.g. how the dummies for economic status, 2nd category, are coded. The response variable y_i in each of the $i = 1, \ldots, 10$ categories represents the median SPQ score in each category. (BMUW, Example 5.3, p. 178).

Categories			Dummies with CR				Dummies with STZ			
Gender	E.status	City	D_2^{gen}	$D_2^{\text{e.stat}}$	$D_3^{\text{e.stat}}$	D_2^{city}	D_2^{gen}	$D_2^{\text{e.stat}}$	$D_3^{\text{e.stat}}$	D_2^{city}
1	**1**	1	0	**0**	0	0	-1	**-1**	-1	-1
1	**2**	1	0	**1**	0	0	-1	**1**	0	-1
1	**3**	1	0	**0**	1	0	-1	**0**	1	-1
2	**1**	1	1	**0**	0	0	1	**-1**	-1	-1
2	**2**	1	1	**1**	0	0	1	**1**	0	-1
2	**3**	1	1	**0**	1	0	1	**0**	1	-1
1	**1**	2	0	**0**	0	1	-1	**-1**	-1	1
1	**2**	2	0	**1**	0	1	-1	**1**	0	1
2	**1**	2	1	**0**	0	1	1	**-1**	-1	1
2	**2**	2	1	**1**	0	1	1	**1**	0	1

Notice that in the above table, dummies of the type D_1 are not shown because they are not involved (not needed) in the expression for μ_i.

Dummy variables can be useful for defining **interaction terms** too. With CR dummies the interactions clearly correspond to products:

$$D_{i,a,b}^{AB} = D_{ia}^{A} D_{ib}^{B}$$

since the product is zero whenever either one of the terms is zero. To check how the same product works with STZ dummies, consider what the interaction terms would be for two 3-level factors. We have D_{i2}^{A} which is 1 if $i \in A_2$ but 0 otherwise, **except** it is -1 when $i \in A_1$. Likewise D_{i3}^{A} and D_{i2}^{B} and D_{i3}^{B} are defined as STZ dummies.

Consider interactions expressed in terms of products of these STZ dummies ('(2,2)', '(2,3)', '(3,2)' and '(3,3)') which can be collected as terms in the sum:

$$\gamma_{22} D_{i2}^{A} D_{i2}^{B} + \gamma_{32} D_{i3}^{A} D_{i2}^{B} + \gamma_{23} D_{i2}^{A} D_{i3}^{B} + \gamma_{33} D_{i3}^{A} D_{i3}^{B}$$

Now when we are in category (2,2), we have $D_{i2}^{A} = 1$ and $D_{i2}^{B} = 1$ while all the rest are zero. Hence we get the term γ_{22} only. Likewise γ_{23} and γ_{32} and γ_{33} are obtained if we are in category (2,3), (3,2) or (3,3).

But if we are in category (1,2), we have $D^A_{i2} = -1$ and $D^B_{i2} = 1$ which gives $-\gamma_{22}$. Also $D^A_{i3} = -1$ and $D^B_{i2} = 1$ which gives $-\gamma_{32}$. But $D^B_{i3} = 0$ because we are in category 2 for factor B. Therefore we get $-\gamma_{22} - \gamma_{32}$ for category (1,2). In this way, the following interaction terms are picked for all 3×3 categories:

$\gamma_{22} + \gamma_{32} + \gamma_{23} + \gamma_{33}$	$-\gamma_{22} - \gamma_{32}$	$-\gamma_{23} - \gamma_{33}$
$-\gamma_{22} - \gamma_{23}$	γ_{22}	γ_{23}
$-\gamma_{32} - \gamma_{33}$	γ_{32}	γ_{33}

And they have the required sum to zero constraints as they should with STZ parameterization.

Furthermore, these dummies could be collected in a design matrix X so that we can then use vector notations (and `inprod()`). Constraints are then automatically implemented for higher order interactions too.

Below is WinBUGS code for comparison with those given in BMUW, tables 6.2 and 6.3, p. 193-194. Try running this model with different explanatory factors (main effects) and their interactions.

```
 model{  # alternative code for those in tables 6.2, 6.3 in BMUW
 for(i in 1:n){
   for(k in 1:2){
     Dcr.gen[i,k] <- equals(g[i],k);
     Dstz.gen[i,k] <- Dcr.gen[i,k]-Dcr.gen[i,1]
   }  # Dcr.gen[i,1], Dstz.gen[i,1] are not used
   for(k in 1:3){
     Dcr.eco[i,k] <- equals(e[i],k)
     Dstz.eco[i,k] <-  Dcr.eco[i,k]-Dcr.eco[i,1]
   }  # Dcr.eco[i,1], Dstz.eco[i,1] are not used
   for(k in 1:2){
     Dcr.city[i,k] <- equals(ci[i],k)
     Dstz.city[i,k] <- Dcr.city[i,k] - Dcr.city[i,1]
   } # Dcr.city[i,1], Dstz.city[i,1] are not used
         # Design matrix construction with interactions:
         X[i,1] <- 1  # constant one for the 1st column
         X[i,2] <- equals(g[i],2) # gender (CR constraints)
         X[i,3] <- equals(e[i],2) # econ2 (CR constraints)
         X[i,4] <- equals(e[i],3) # econ3 (CR constraints)
         X[i,5] <- equals(ci[i],2) # city (CR constraints)
         X[i,6] <- X[i,2]*X[i,3]   # gender*econ2
         X[i,7] <- X[i,2]*X[i,4]   # gender*econ3
         X[i,8] <- X[i,2]*X[i,5]   # gender*city
         X[i,9] <- X[i,3]*X[i,5]   # econ2*city
         X[i,10] <- X[i,4]*X[i,5]  # econ3*city  (X[i,3] in BMUW? )
         X[i,11] <- X[i,2]*X[i,3]*X[i,5] # gender*econ2*city
         X[i,12] <- X[i,2]*X[i,4]*X[i,5] # gender*econ3*city
} for(i in 1:n){
 y[i] ~ dnorm(mu[i],tau)
 mu[i] <-mu0+alpha[2]*Dcr.gen[i,2]+alpha[3]*Dcr.eco[i,2]+
             alpha[4]*Dcr.eco[i,3]+alpha[5]*Dcr.city[i,2]
 yx[i] ~ dnorm(mux[i],taux)
 yx[i] <- y[i]
 mux[i] <- inprod(X[i,],beta[])
}
 mu0 ~ dnorm(0.0, 0.0001)
 tau ~ dgamma(0.01,0.01); taux ~ dgamma(0.01,0.01)
 s <- sqrt(1/tau); sx <- sqrt(1/taux)
 for(k in 1:5){ alpha[k] ~  dnorm(0,0.0001)} # alpha[1] is not used
 for(k in 1:12){ beta[k] ~ dnorm(0,0.0001)} # parameters for design matrix
 }
 list(n=10,g=c(1,1,1,2,2,2,1,1,2,2),e=c(1,2,3,1,2,3,1,2,1,2),
 ci=c(1,1,1,1,1,1,2,2,2,2),y=c(9,25,23,18,22,12,14,26,29,25))
 list(alpha=c(0,0,0,0,0),mu0=0,tau=1,beta=c(0,0,0,0,0,0,0,0,0,0,0,0),taux=1)
```

10.1 qualitative and quantitative variables as explanatory variables

ANCOVA models: analysis of covariance. Models with qualitative explanatory variables (factors) are extended by including also quantitative explanatory variables (continuous covariates). Possibly of most interest could be the interaction term between categorical and numerical variables: the slopes of regression lines for different groups.

Available data assumed: (X_i, A_i, Y_i).

Some possible models:

1. Constant: $\mu_i = \beta_0$
2. Common line for all groups: $\mu_i = \beta_0 + \beta_1 X_i$
3. Constant mean within each group: $\mu_i = \beta_0 + \alpha_{a_i}$
4. Parallel line (common slope): $\mu_i = \beta_0 + \beta_1 X_i + \alpha_{a_i} = \beta^*_{0,a_i} + \beta_1 X_i$
5. Common intercept: $\mu_i = \beta_0 + \beta_1 X_i + \gamma_{a_i} X_i = \beta_0 + \beta^*_{1,a_i} X_i$
6. Separate regression lines for each group: $\mu_i = \beta_0 + \beta_1 X_i + \alpha_{a_i} + \gamma_{a_i} X_i = \beta_0 + \alpha_{a_i} + \beta^*_{1,a_i} X_i$.

Models 4-6 are ANCOVA models.

Parallel lines

In model 4, we have $\beta^*_{0,a_i} = \beta_0 + \alpha_{a_i}$ where $a_i \in \{1, \dots, L_A\}$ and we need CR or STZ constraints. The remaining model parameters are $\beta_0, \beta_1, \alpha_2, \dots, \alpha_{L_A}$ so that α_1 for the reference category is omitted. Therefore, for the reference category: $\mu_i = \beta_0 + \beta_1 X_i$.

With CR constraints, the interpretation of $\alpha_l (l \geq 2)$: the expected difference of Y between a subject of group l and a subject in reference group having the same X. $E(Y \mid X, A = l) - E(Y \mid X, A = 1) = \alpha_l$.

With CR constraints, the interpretation of β_1: the expected difference of Y when comparing two subjects differing by one unit in X and belonging to the same group. $E(Y \mid X = x+1, A = l) - E(Y \mid X = x, A = l) = \beta_1$.

Using dummy variables (either CR dummies, or STZ dummies), the model can be coded:

```
mu[i] <- beta0 + beta1*x[i] + inprod(alpha[2:LA],D[i,2:LA])
```

If using the simplified parameters with $\mu_i = \beta^*_{0,a_i} + \beta_1 X_i$ we can avoid constraints because we then have only $(\beta_{0,1}, \ldots, \beta_{0,L_A}, \beta_1)$ as parameters:

```
mu[i] <- beta.star[a[i]] + beta1 * x[i]
```

Separate lines

In model 6, we have $\beta^*_{0,a_i} = \beta_0 + \alpha_{a_i}$ and $\beta^*_{1,a_i} = \beta_1 + \gamma_{a_i}$. Thus the set of parameters is $(\beta_0, \beta_1, \alpha_2, \ldots, \alpha_{L_A}, \gamma_2, \ldots$ With CR constraints we have $\alpha_1 = \gamma_1 = 0$ which leaves β_0 and β_1 to be interpreted as parameters of the reference category. The WinBUGS code is of this form:

```
mu[i] <- beta0 + beta1*x[i] + alpha[a[i]] + gamma[a[i]]*x[i]
```

and either CR or STZ constraints for α and γ are needed. Alternatively, use the simplified parameters to avoid constraints:

```
mu[i] <- beta0.star[a[i]] + beta1.star[a[i]]*x[i]
```

With dummy variables, the model is:

$$\mu_i = \beta_0 + \beta_1 X_i + \sum_{l=2}^{L_A} \alpha_l D^A_{i,l} + \sum_{l=2}^{L_A} \gamma_l D^A_{i,l} X_i$$

Using inprod-command this is coded:

```
mu[i] <- beta0 + beta1*x[i] + inprod(alpha[2:LA],DA[i,2:LA]) +
inprod(gamma[2:LA],DAX[i,2:LA])
```

for which we need to define the product of D and X as:

```
for(l in 1:LA){DAX[i,l] <- DA[i,l]*x[i] }
```

within the loop over $i = 1, \ldots, n$. This code could be made still more compact using the design matrix formulation.

Common intercept

In the above model, remove main effects α_l.

10.1.1 Bioassay example

Fictitious data for blood clotting time (in seconds) under the standard drug and a new test drug, under three different doses (four measurements per dose). See data in table 6.6., p. 204 in BMUW.

Task 1: to estimate the potency ρ of new drug from the parallel lines model with

$$E(Y) = \mu = \begin{cases} \beta_0 + \beta_1 \log(\text{dose}) & \text{Standard} \\ \beta_0 + \beta_1 \log(\rho \text{ dose}) & \text{New drug} \end{cases}$$

By writing $\beta_0' = \beta_0 + \beta_1 \log(\rho)$ we can express the model for the new drug as $\beta_0' + \beta_1 \log(\text{dose})$. The two models have the same slope but different intercepts (i.e. parallel lines). The relative potency can thus be written as $\rho = \exp(\frac{\beta_0' - \beta_0}{\beta_1})$. In WinBUGS, it is thus easy to calculate the posterior distribution of ρ either as having it as a parameter directly, or by calculating it from the parameters $\beta_0', \beta_0, \beta_1$ if these are used as parameters of the model.

Remember that in WinBUGS `beta[1]` corresponds to β_0, etc. Hence, the mean $E(Y)$ for the standard drug is coded

```
beta[1] + beta[2]*log(dose)
```

and for the test drug

```
beta[1] + beta[2]*log(dose) + beta[3]
```

where `beta[3]` equals `beta[2]*log(rho)`. In this parametrization under CR constraints, the difference between the two means is exactly `beta[3]`, and `rho = exp(beta[3]/beta[2])`. Under STZ constraints, the means are

```
beta[1] + beta[2]*log(dose) - beta[3]
```

and

```
beta[1] + beta[2]*log(dose) + beta[3]
```

so that the difference between these two is `2*beta[3] = 2*beta[2]*log(rho)`, hence `rho = exp(2* beta[3] /beta[2])`.

The whole model code in table 6.8, p.207 of BMUW with details (both CR and STZ versions). See also the alternative code in table 6.7 which avoids constraints by having one parameter less.

Since the explanatory variable was log(dose), the interpretation of slope parameter β_1 (and intercept β_0) may not be directly useful. A rescaling was suggested by replacing log(dose) by $\log_2(40 \times \text{dose})$ so that (with the doses applied in data: 0.025, 0.05, 0.1) we have values 0, 1 and 2 for the explanatory variable. Now the intercept is directly the expected clotting time with the lowest dose, and the slope corresponds to the effect when doubling the dose. (With log(dose), the increase of 1 corresponds to multiplication of the old dose by e. With $\log_2(\text{dose})$ the increase of 1 corresponds to multiplication by 2). The new explanatory variable is coded `x[i]<-log(40*dose)/log(2)`. Also the coding of relative potency is changed in this rescaling.

Checking parallel lines assumption: define a model with separate lines and then study if the slope parameters are reasonably different or not:

```
mu[i] <- beta0[drug[i]] + beta1[drug[i]]*log(dose[i])
slope.diff <- beta1[2]-beta1[1]
prob <- 1-step(slope.diff)
```

Task 2: to estimate the potency ρ of new drug from the model with common intercept and different slope (slope ratio analysis)

$$E(Y) = \mu = \begin{cases} \beta_0 + \beta_1 \text{dose} & \text{Standard} \\ \beta_0 + \beta_1 \rho \text{ dose} & \text{New drug} \end{cases}$$

In other words, the mean for the new drug can be rewritten $\beta_0 + \beta_1' \text{dose}$ where $\beta_1' = \beta_1 \rho$. With these notations, the relative potency is written $\rho = \beta_1'/\beta_1$ and we could use these parameters in the model and monitor ρ as their function. With CR or STZ constraints, the model has $\mu_i = \beta_0 + \beta_1 \text{dose}_i + \gamma_{a_i} \text{dose}_i$. Under CR we have $\gamma_1 = 0$ and γ_2 as our parameters. Under STZ we have $\gamma_1 = -\gamma_2$ and γ_2 as our parameters. Notice that with CR the slopes are equal to β_1 and $\beta_1 + \gamma_2$, so we get $\rho = (\beta_1 + \gamma_2)/\beta_1$. With STZ the slopes are $\beta_1 - \gamma_2$ and $\beta_1 + \gamma_2$, so we get $\rho = (\beta_1 + \gamma_2)/(\beta_1 - \gamma_2)$.

Rescaling of dose: instead of absolute dose as explanatory variable, use dose/ min(dose). Then the slope parameter describes the expected change in response when the dosage is increased by a quantity equal to the minimum dose (in the experiment). In BUGS, the minimum is calculated as `ranked(dose[],1)`.

10.2 Variable selection using indicators

When comparing different models, we need to write the corresponding codes in BUGS. Usually, this means the same model structure but with different collection of explanatory variables. This can be done slightly easier by using binary vector $\gamma = (\gamma_1, \ldots, \gamma_p)$ in the linear predictor:

$$\mu_i = \beta_0 + \sum_{j=1}^{p} \gamma_j \beta_j X_{ij}$$

In BUGS:

```
mu[i] <- beta0 + gamma[1]*beta[1]*x[i,j] + ... + gamma[p]*beta[p]*x[i,p]
```

where X_{ij} is the jth covariate or dummy variable. The indicators γ can be specified in the data list so that the model code remains the same.

More compact presentation uses the data matrix X:

```
for(i in 1:P){ betag[j] <- gamma[j]*beta[j]}
for(i in 1:N){ mu[i] <- inprod(betag[],X[i,])}
```

⇒ Fit all 2^p models and compute DIC for each! Select the model with lowest DIC. (This can be a lot to compute if p is large).

⇒ A suboptimal, but often sufficiently good strategy is to use a stepwise method. First, a starting model is specified. Either the null model (with no covariates) or the full model (with all covariates). In the latter case, we need to have $p+1 < N$ where N is the data sample size. The starting model has indicator vector γ^0. We specify p candidate models so that their γ vectors are identical to γ^0 except for one element. In BUGS, this is constructed:

```
for(k in 1:p){
 for(j in 1:p){
   gamma.can[j,k] <- gamma[j]*(1-equals(k,j))+(1-gamma[j])*equals(k,j)
 }
}
```

To run all the models in a single BUGS simulation, the response variables need to be assigned for each model:

```
for(i in 1:N){
y1[i] <- y[i]
y2[i] <- y[i]
...
yp[i] <- y[i]
y1[i] ~ dnorm(mu.can[i,1],tau.can[1])
y2[i] ~ dnorm(mu.can[i,2],tau.can[2])
...
yp[i] ~ dnorm(mu.can[i,p],tau.can[p])
}
```

The mean values for each candidate model is given by:

```
for(k in 1:p){ for(i in 1:n){
mu.can[i,k]<-beta0.can[k]
+beta.can[1,k]*gamma.can[1,k]*x[i,1]
+...
+beta.can[p,k]*gamma.can[p,k]*x[i,p]
}}
for(k in 1:p){
beta0.can[k] ~ dnorm(0,0.001)
tau.can[k] ~ dgamma(0.001,0.001)
for(j in 1:p){
beta.can[j,k] ~ dnorm(0,0.001)
}}
```

And of course, we need to include the initial model itself:

```
for(i in 1:n){
y[i]~ dnorm(mu[i],tau)
mu[i] <- beta0 + beta[1]*gamma[1]*x[i,1]
               + ...
               + beta[p]*gamma[p]*x[i,p]
}
```

...and the corresponding priors for the initial model. After running all these simultaneously, we can monitor DIC value and select the model with lowest. In the next step, this model will be the initial model, and the procedure is repeated until the best model is the current initial model. The same procedure can be repeated starting from the null or the full model, to see if it leads to the same model choice.