1 Oring Example in BUGS

Recall the logistic regression model for the oring data with

\[ Y_i \sim \text{Bernoulli}(\pi_i) \]
\[ \eta_i = \beta_0 + \beta_1 \text{temp}_i \]
\[ \log(\pi_i/(1 - \pi_i)) = \eta_i. \]

The S-Plus summary for this model was

```r
> summary(glm(Y ~ temp, family=binomial, data=orings))

Coefficients:
                     Value Std. Error  t value
(Intercept)  10.8753321  5.69793801  1.908643
temp          -0.1713202  0.08336339 -2.055102

(Dispersion Parameter for Binomial family taken to be 1 )

Null Deviance: 28.97459 on 23 degrees of freedom
Residual Deviance: 23.03045 on 22 degrees of freedom

Number of Fisher Scoring Iterations: 4
```

Correlation of Coefficients:

(Intercept)
temp -0.9958713

We will consider fitting this model in WinBugs. Models in WinBugs may be specified either graphically, using DoodleBUGS, or in the Bugs language. The model declaration requires specification of the likelihood and prior distributions. Start up WinBUGS on a PC. Under the file menu, open a new file, and with the Bugs editor enter the model specification. Note the S-plus like syntax!

```r
model {
  for (i in 1:24) {
    Y[i] ~ dbern(pi[i])
    logit(pi[i]) <- alpha0 + alpha1*temp[i]
  }
  alpha0~dnorm(0,E-04)
  alpha1~dnorm(0,E-04)
}
```

In the “Model” menu, one can select the specification tool, which provides model checking, loading of data, compiling and specification of initial values. Once the model is typed in, click the “check model” button. If (when) there are errors, they will appear at the bottom of the screen.

If the model syntax is correct, then the next step is to load in the data. I entered the values in a file with the following structure

```r
temp[] Y[]
53 1
56 1
...
To load the data file, open the file (using the file menu) and while that window is active click on the “load data” button in the specification tool. One can also use S-Plus and the function `dpars()` to create input files. Care does have to be taken with matrices because S-Plus and BUGS store matrices in differently!

The next step is to compile the code and model; click on “compile” in the specification tool. If all is well we just need to give some starting values and then we will be ready to generate values from the posterior distribution!

The starting values below are specified using the S-Plus data format

```r
list(alpha0=-11, alpha1=-.2)
```

Highlight “list” and then click on the “load inits” button in the Specification Tool.

To run the Gibbs sampler, bring up the “Samples” tool from under the Inference menu. Using the “set” button, we can specify which “nodes” that we would like to monitor. Enter “alpha0” and click on set, then enter “alpha1” and click on set. To observe trace plots, select a node, and click on “trace”. (repeat for other nodes) To start sampling, bring up the “Update” tool from the “Model” menu. Click Update to update values by sampling from the Markov chain. The trace plots will show the generated values as the chain is updated (see next page)
To change the model to reduce correlation between the intercept and slope, we can center the X’s.

```r
> summary(glm(Y ~ I(temp - mean(temp)), family=binomial, data=orings))
Call: glm(formula = Y ~ I(temp - mean(temp)), family = binomial, data = orings)

Deviance Residuals:
     Min       1Q   Median       3Q      Max
-1.212493 -0.8252676 0.470546 0.5907502 2.051237

Coefficients: Value Std. Error t value
(Intercept) -1.1028082 0.53970012 -2.043372
I(temp - mean(temp)) -0.1713202 0.08336339 -2.055102

(Dispersion Parameter for Binomial family taken to be 1 )

Null Deviance: 28.97459 on 23 degrees of freedom
Residual Deviance: 23.03045 on 22 degrees of freedom

Number of Fisher Scoring Iterations: 4

Correlation of Coefficients:
   (Intercept)
I(temp - mean(temp)) 0.2854867

BUGs code:

model {
 for ( i in 1:24) {
   Y[i] ~ dbern(pi[i])
   logit(pi[i]) <- alpha0 + alpha1*(temp[i] - mean(temp[]))
 }
 alpha0~dnorm(0, 1.0E-6)
 alpha1~dnorm(0, 1.0E-6)
}