Practical 3: Introduction to Bayesian statistics using WinBUGS

In this practical you will get your first exposure to the WinBUGS computer package. For the purposes of the practical we will not explain exactly what WinBUGS is doing as this will be covered in detail in the next lecture. However you should think of it as a package that performs simulation based methods to get Bayesian estimates for the models described in lecture 2.

We will here give step-by-step instructions to perform some Bayesian analyses for the girl's height and weight data covered in the lecture. The data can be found in exercises in Sanford Weisberg's Applied Linear Regression book. The data are taken from the Berkeley Guidance Study that enrolled children born in Berkeley, California between January 1928 and June 1929. We will look at two models: Firstly a model for the mean height of the sample of 10 18 year old girls and secondly a model for a regression of weight on height for the same sample.

Firstly you will need to load up the WinBUGS 1.4 software. Where the icon/option for this will be on your machines varies so we will tell you exactly where the software is at the start of the practical. When you select the icon WinBUGS will start up in a window on your machine. We now need to Open the file associated with the first model that we will run in WinBUGS.

The WinBUGS code for the first model should appear as follows:

```
#----MODEL Definition----------------
model
{
        for(i in 1:N)
        {
                height[i] ~ dnorm(mu,tau)
        }
        mu ~ dnorm(priormean, priorprec)
        priorprec <- 1/priorvar
        tau <- 1/50
#----prior 1
        priormean <- 165
        priorvar <- 4
#----prior 2
#priormean <- 170
#priorvar <- 9
}
#----Initial values file----------------------------
list(mu = 150)
#----Data File-----------------------------------
list(N= 10, height=c(169.6,166.8,157.1,181.1,158.4,165.6,166.7,156.5,168.1,165.3))
```

You will see that WinBUGS files are split into three parts: a model description, an intial values file and a data file. Historically the BUGS software package which was a predecessor to WinBUGS would talk three files as input but in WinBUGS we are able to put all three parts into one file. As this lab is just a familiarization exercise of the package we will not go into great detail of what each line means. We will return to WinBUGS later in practicals and give more details. For now all we need to realize is that a WinBUGS file contains these 3 parts.

The model definition always begins with the word *model* and is contained within a pair of {} parentheses. Both the data and initial values files start with the word *list* and are contained in () parentheses. WinBUGS borrows much of its style from the R package that you will have looked at this morning in particular the use <- for equals and *c* to represent an array. The model definition consists of two types of statement; distributional relationships represented by a ~ and deterministic relationships represented by a <- symbol. The # symbol is a special comment character meaning that everything after it in the line is ignored by the program. The final unusual feature in WinBUGS is that the normal distribution (*dnorm*) is defined in terms of its mean and precision (rather than variance).

This would be a good point to revisit the lecture slides and match up the information in this WinBUGS file with the normal distribution with unknown mean example we looked at there.

Running the model in WinBUGS

To set up this model in WinBUGS we need to perform the following steps:

Select **Specification** from the **Model** menu. On the **Specification** window click on the **check model** button. This should give the message '*model is syntactically correct*' at the bottom of the screen. Next we need to load in the data for the model. Highlight the *list* identifier at the start of the data list and click on the **load data** button in the **specification** window. If this is successful the message '*data loaded'* will appear at the bottom of the screen. Next we have to combine the data and model definition by clicking on the **compile** button. Again if this operation is successful a message appears at the bottom of the screen, this time stating that '*model compiled*'. As WinBUGS works through simulation techniques all unknown parameters (in this case simply *mu*) need starting values. These are stored in the initial values file and so the final step to model set up is to highlight the *list* identifier at the start of the initial values file and click on the **load inits** button on the **specification** window. This will then give the final message '*initial values loaded; model initialized*'.

We next need to specify the parameters we are interested in monitoring before starting the program simulating values. From the **Inference** menu select the **Samples** options and a window will appear that allows the user to specify which parameters to monitor. In the

node box type *mu* and click on the **set** button.

We are now ready to start generating sample values for *mu*. Under the **Model** menu select the **Update** window. If we now click on the **Update** button the software will generate 1000 values from the distribution for *mu*. This will happen very quickly in less than 1 second. If we now wish to look at the 1000 values we must return to the **Samples** window. We need to again type *mu* in the **node** box. This time all the buttons will be highlighted and if we click on the **history** button we get the following:

EMBED Unknown

Here we see the chain of 1000 values for *mu*. We can get more information by clicking on the **stats** button which gives the following:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	165.3	1.508	0.04947	162.4	165.3	168.2	1	1000

We can here compare with the theoretical results in the lecture of a point estimate of 165.23 and 95% credible interval of (162.31,168.15). If we run for more iterations we will get more accurate estimates. To do this change the 1,000 to 4,000 and click on the **update** button. We will now get the following estimates based on 5,000 updates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	165.2	1.497	0.01981	162.3	165.2	168.1	1	5000

Prior 2

In the lecture we looked at a second prior for the mean and this is commented out in the WinBUGS file. If we modify the file as follows we can try out the second prior:

```
#----MODEL Definition----------------
model
{
        for(i in 1:N)
        {
                height[i] ~ dnorm(mu,tau)
        }
        mu ~ dnorm(priormean, priorprec)
        priorprec <- 1/priorvar
        tau <- 1/50
#----prior 1
#       priormean <- 165
#       priorvar <- 4
#----prior 2
    priormean <- 170
    priorvar <- 9
}
#----Initial values file----------------------------
list(mu = 150)
#----Data File---------------------------------
list(N= 10, height=c(169.6,166.8,157.1,181.1,158.4,165.6,166.7,156.5,168.1,165.3))
```

You will have to perform the same set of model specification steps to run this model as you did for the first prior. If you run with this prior for 5,000 iterations you should get the

following results:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	167.1	1.801	0.02383	163.6	167.1	170.6	1	5000

Note these compare favourably with the results in the lecture where we had a point estimate of 167.12 and 95% credible interval of (163.61,170.63).

Example 2: Linear Regression

Our second example involves fitting a linear regression model to the heights and weights dataset. This example can be found in the file *regression.odc* and loaded from the File menu. The WinBUGS code is as follows:

```
#----MODEL Definition----------------

model
{

for(i in 1:N) {
height[i] ~ dnorm(mu[i],tau)
mu[i] <- beta0 + beta1* weight[i]
}
beta0 ~ dnorm(0,1.0E-6)
beta1 ~ dnorm(0,1.0E-6)
#---------prior 1
tau ~ dgamma(1.0E-3,1.0E-3)
sigma2 <- 1.0/tau
#---------prior 2
#tau <- 1.0/(sigma*sigma)
#sigma ~ dunif(0,1000)
#sigma2 <- sigma*sigma
}

#----Initial values file----------------------------
list(beta0 = 0, beta1 = 0, tau = 1)
#list(beta0 = 0, beta1 = 0, sigma = 1)

#----Data File---------------------------------

list(N= 10, height=c(169.6,166.8,157.1,181.1,158.4,165.6,166.7,156.5,168.1,165.3),
weight =c(71.2,58.2,56.0,64.5,53.0,52.4,56.8,49.2,55.6,77.8))
```

This model is very similar to that described at the end of the lecture, although we have got slightly different priors defined. These priors (normal priors with big variances for the two βs and a Gamma(ε,ε) prior for the precision parameter which is close to a Uniform prior for $\log(\sigma^2)$) are commonly used in such models and are easier to specify in WinBUGS. In this example we have three unknown parameters, *beta0*, *beta1*, and *sigma2*. You should from your experience with the first model now be able to use WinBUGS to get estimates for these parameters. Note that when specifying the parameters to be sampled you need to input each of three parameters in turn and click on the **set** button.

If you run the model for 5,000 iterations following the instructions for the last model you will get the following output. Note here that if you type * in to the node box WinBUGS

will give you output of all parameters currently set. Firstly the chains look as follows:

EMBED Unknown
EMBED Unknown
EMBED Unknown
and the statistics are as follows:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta0	143.3	17.33	0.2626	108.8	143.7	177.3	1	5000
beta1	0.3724	0.2873	0.004408	-0.1955	0.3674	0.9471	1	5000
sigma2	60.71	40.6	0.6742	21.33	50.16	169.1	1	5000

Note that you can compare these answers from those in the frequentist methods used in lecture 1.

A second prior for the variance

As discussed briefly in the lecture the choice of a 'default' or 'non-informative' prior for the variance parameter is difficult and here we have chosen a proper prior that is close to a uniform prior on $\log(\sigma^2)$. As an exercise we have included the lines of code (commented out) for a uniform prior on σ. You can now test out this prior instead. You will not to uncomment 4 lines and comment out 3 lines (including changing initial values lines)

You should get the following estimates if you run for 5,000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta0	142.7	18.85	0.2564	105.3	142.6	179.6	501	4500
beta1	0.3825	0.3135	0.004241	-0.222	0.381	1.014	501	4500
sigma2	73.24	55.48	1.245	23.43	57.85	219.6	501	4500

Note that there is quite a difference in the mean estimate for *sigma2* which means the prior is quite influential. Remember of course that our regression is only based on 10 observations and so there is perhaps not a great deal of information in the dataset and hence the choice of prior can be important (see exercise 3)
As this prior is not conjugate WinBUGS uses more complex methods and you will notice that although we have run for 5,000 iterations. WinBUGS has culled off the first 500 iterations to allow the simulations to settle down (check out the chains to see why this is necessary) this is known as a 'burn-in' and will be described in the next lecture.

Further exercises

1. You might like to try changing the priors for *beta0* and *beta1*. These are currently Normal with a variance of 10^6 as a proper approximation to the Uniform distribution on the full real line. WinBUGS allows such a Uniform which it calls a *dflat()* distribution. If you increase the precision from 10^{-6} to say 10^{-4} the prior becomes informative and you may like to try and see what happens when we change priors here.

2. You might like to see what happens if you change the values in the initial values file for this regression model.

3. Try increasing the importance of the data by increasing the size of the dataset fourfold. To do this change N to 40 and repeat the height and weight vectors 4 times. Do the two priors have as much influence now?

4. If you have your own data you might like to try and take a small part of it and fit a linear regression to it, or even just a variant of the first model.

PAGE

P3-PAGE 1

Select **Open** from the **File** menu
Change directory to the summer school directory
Select '*height1*' from the list that appears