Computing in Statistics, 22S:166
Lab 4
Oct. 18, 2006
Introduction to WinBUGS

1 If you want WinBUGS for your home PC

WinBUGS was developed by David Spiegelhalter and colleagues at the Medical Research Council Biostatistics Unit in Cambridge, England. If you want WinBUGS for your own PC, you can download files from

`http://www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml`

Be sure to get the "key" that transforms the student version into the full-function version.

2 Bringing up WinBUGS and Finding Documentation

Select WinBUGS from the "All Programs" menu list.

On-line documentation is available under the "Help" menu, and it can be printed. I will also put a written copy of the documentation on reserve in the Mathematical Sciences library. Be sure to take a look at the examples as well as the manual itself.

3 Finding the example WinBUGS code

Bring up Netscape and go to the course web page. Under "Handouts," click on "winbugs.example." Open this document in a window, as we will want to copy and paste it into a WinBUGS window.

4 To fit a model using WinBUGS

Models may be defined for WinBUGS by using either a simple, R/Splus-like command language or by drawing graphical models. The steps below are for the command language version.

Things to note:

- WinBUGS parameterizes the normal distribution in terms of mean and *precision*.
- WinBUGS does not permit improper priors, except for the `dflat()` prior (uniform on the whole real line), which cannot be used for precisions. The manual recommends very small values of both parameters for a gamma to approximate a gamma(0, 0)

prior. This prior is *not* recommended for certain precisions in hierarchical models, as we will discuss later in the semester.
- If you wish to estimate the posterior distribution of a *function* of model parameters, WinBUGS can compute the function and generate samples of it.
- If you wish to estimate the posterior predictive distribution of potential new data, you can add one or more "NA"s to the data list. This is WinBUGS' notation for an unknown data value. It will then treat that as one more unknown quantity to simulate.

1. Select the "File" menu, and from it select "New."
2. Highlight and copy the code from the course web page and paste it into the WinBUGS new window. You can also type in new code in this type of window, or load in previously-saved programs.

   WinBUGS code must include the following sections (see example):

   - model
   - data (alternatively, the data may be a file that has been read into another window)
   - initial values
3. Use the mouse to highlight the word "model" at the beginning of the model section of your code. Then select the "Model" menu and from it select "Specification" and then "check model." Watch for a message at the bottom of the WinBUGS window either confirming the validity of the model or reporting errors.
4. Highlight the word "list" at the beginning of your data listing. From the "Specification tool" box select "load data." Again check for a message confirming data loading or errors.
5. In the "Specification tool" box, change the number of chains from 1 to 3.
6. From the "Specification tool" box select "Compile."
7. Highlight the word "list" at the beginning of your initial values section. Select the "Model" menu and from it select "load inits." Again check for a message. You will get a message that some nodes are unitialized. Continue to load initial values for each of the other 2 chains.
8. Select the "Model" menu and "Update." You will be prompted for how many iterations you want to run the sampler. For now, just accept the default of 1000.
9. To start saving samples from the posterior distribution of the unknown quantities in your model, select the "Inference" menu and "Samples" from it. Type the name of each parameter whose posterior distribution you want to study in the window in the prompt box (this will be just p in this simple example), and click on "set" after each one.

10. Select the "Model" menu and "Update." You will be prompted for how many iterations you want to run the sampler. For now, enter 2000.

11. Go to the Options pull-down menu and be sure that "Use log" is checked. This will cause all the output we are about to request to go into a single window instead of creating a bzillion small windows cluttering up the screen.

12. Go back to the "Sample monitor" box and select the desired parameter in the node box. Entering an asterisk requests all monitored nodes. Then, one at a time, click "trace," "history," "stats", "density," and "GRdiag." We will discuss the meaning of this output.

13. To get more precision in your posterior estimation, you may return to the "Update" box and request additional samples. Then go back to the previous step to include these samples in the output analysis.

14. To print the content of any window, click on that window and then select the "File" menu and "Print." If you wish, you may copy and paste graphical and tabular output from the "Sample monitor" windows into a single window for compact printing.

Now we will run a second example, model 3. The steps will be essentially the same, but we will load data from two different sources.

1. Use the mouse to highlight the word "model" at the beginning of the model section for model 3. Then select the "Model" menu and from it select "Specification" and then "check model." WinBUGS will warn you that this new model will replace the model we were working with previously. This is fine.

2. The data for this problem is in two parts – one in list format and one in table format. First highlight the word "list" at the beginning of your data listing. From the "Specification tool" box select "load data." Again check for a message confirming data loading or errors.

3. We are loading additional data from an external data file in table format. Highlight the row of column headings, and again click "load data."

4. In the "Specification tool" box, change the number of chains from 1 to 3.

5. From the "Specification tool" box select "Compile."

6. Highlight the word "list" at the beginning of your initial values section. Select the "Model" menu and from it select "load inits." Again check for a message. You will get a message that some nodes are unitialized. Continue to load initial values for each of the other 2 chains. Even after you load the initial values for the 3rd chain, you will get a message saying there are unitialized nodes. Click "gen inits" to get WinBUGS to generate them automatically.

7. Select the "Model" menu and "Update." You will be prompted for how many iterations you want to run the sampler. For now, just accept the default of 1000.

8. To start saving samples from the posterior distribution of the unknown quantities in your model, select the "Inference" menu and "Samples" from it. Type the name of each parameter whose posterior distribution you want to study in the window in the prompt box (this will be just *mu, sigmasq, and y[19]*), and click on "set" after each one.

9. Select the "Model" menu and "Update." You will be prompted for how many iterations you want to run the sampler. For now, enter 2000.

10. Go back to the "Sample monitor" box and enter an asterisk to request all monitored nodes. Then, one at a time, click "trace," "history," "stats", "density," and "GRdiag."

11. Things to note:
    - WinBUGS fit the model using the mean μ and the precision τ^2. However, we asked WinBUGS also to compute the variance σ^2 so that we could examine the posterior marginal distribution of a quantity we understand better.
    - We monitored $y[19]$ in order to examine the posterior predictive distribution for a new observation

```
# Binomial sampling distribution for data
# unknown parameter is population proportion, p

# model 1

model
{
    y ~ dbin(p, n)
    p ~ dbeta( alpha, beta)
}

data
list(y = 7, n = 50, alpha = 0.5, beta = 0.5)

inits
list(p = 0.1)
list(p = 0.5)
list(p = 0.9)

The following are two versions of a model for the same data.
We assume a normal sampling distribution for the data values.
```

In model 2, we assume that the population precision is known, and the mean mu is the only unknown parameter.

In model 3, we realistically assume that both population parameters are unknown.

```
# Normal sampling distribution for data
# Pretending we know precision
# Population mean mu is unknown parameter
# We can also estimate the posterior predictive distribution by monitoring y[19]


# model 2
model
{
  # likelihood
  for (i in 1:N) {
    y[i] ~ dnorm( mu, tausq )
  }
  # priors
  mu ~ dnorm(0, 0.0001)
}


data
list(y = c(29, 27, 34, 40, 22, 28, 14, 35, 26, 35, 12, 30, 23, 17, 11, 22,
23, 33, NA), N = 19, tausq = .02 )

inits for model 2
list(mu = 0)
list(mu = 20)
list(mu = 40)
```



```
# Normal sampling density for data
# Both mu and tausq unknown

# model 3
model
{
  # likelihood
  for (i in 1:N) {
    y[i] ~ dnorm( mu, tausq )
  }
  # priors
  mu ~ dnorm(0, 0.0001)
  tausq ~ dgamma( 0.0001, 0.0001)
  sigmasq <- 1/tausq
}
```

Here is a different way to give data to WinBUGS.

```
data
list(N = 19 )

additional data
y[]
29
27
34
40
22
28
14
35
26
35
12
30
23
17
11
22
23
33
NA
END

inits for model 3
```

```
list(mu = 0, tausq = 1)
list(mu = 20, tausq = 100)
list(mu = 40, tausq = 1000)
```