Lab 1: Introduction to WINBUGS in the CLICC lab

- PLEASE, OPEN A WORD FILE WHERE YOU CAN COPY PASTE THE OUTPUT OF THIS SESSION.
- YOU MUST TURN IN ANSWER TO THE QUESTIONS GIVEN AT THE END OF THE HANDOUT TODAY
- IF YOU READ THE HANDOUT CAREFULLY AND SLOWLY, YOU PROBABLY WILL NOT ONLY HAVE TIME TO DO THE LAB QUESTIONS BUT ALSO THE HOMEWORK QUESTION.

Objectives:

1. Intro to WinBUGS
2. WinBUGS documentation
3. Entering model
4. Entering data
5. Compiling
6. Looking at results
7. Checking convergence

About WinBUGS

WinBugs (**Bayesian inference Using Gibbs Sampling**) is a statistical package that can fit general statistical models from a Bayesian perspective using a computational technique called **Markov chain Monte Carlo** abbreviated MCMC. Gibbs Sampling is a special type of MCMC. It is incredibly flexible in the models that can be fit. Complex models and models with large amounts of data can take a long time (hours) to run. Simple models are fit quickly. Various extensions are available (spatial, PK). OpenBugs is a current development version.

WINBUGS requires that you specify a complete JOINT distribution for the data and unknown parameters. You may enter the data, and WINBUGS will choose a computational method for fitting the model and giving you inferences. It supplies many basic Bayesian summaries of your inferences, but if we need more complex output, or in a nicer format, we need to move the output to another statistical package like R for additional calculations.

WINBUGS is like SAS in that the language is a bit weird at first. One of the easiest ways to get started is to find an example that is vaguely similar to the model you need to fit and adapt it to your data set and model. WINBUGS comes with a manual and an extensive set of examples. The manual is short, fairly complete and it is definitely WORTH READING. The **examples** are quite good and there are a lot of them. You should wander through a sampling of at least 5, looking at the variety of models and context.

For your home computer, you can download WINBUGS from the WINBUGS website at http://www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml . Click on Winbugs. Download the main program, and download the update to 1.4.3. **Be sure to register which will cause Winbugs people to email you a key which allows you to use the full power of WINBUGS.**

Open WINBUGS

Actually do the following commands on your computer while reading this.

After you log in to the computer and Windows comes on, click the **Start** menu and then **Programs**. Click on **Social Science Applications.** Go down the menu and you'll find **WinBUGS14**, almost at the bottom of the list. Click to start. Once you have WINBUGS started, inside WinBugs, a window with a license agreement opens; you can close the license agreement. To start a new file, click on the **File** menu and then hit the **New** option. Or Cntrl-N. This will create a new window where you can type in your program. You may resize the window to make things easier to see.

First go to help menu and open up "About WinBUGS" It should say version 1.4.3.

Documentation

The manual is under the **help menu**. You should read the manual, chapters listed at the top of the page at home. You may skip 4 (Doodlebugs), we will not be using it in class. (This doesn't mean it's not important!) We will cover much of this material in labs also, so it is ok if you don't understand it all at first. Read the manual at home, not in lab. Two reference tables are important. Table 1 is under "Model Specification" and "Distributions" has its own chapter now.

Examples are also under the help menu, organized into two sets of examples.

TODO

Do the following.

A. Click on the help menu and open the manual. Page down the manual, inspect the table of contents. Go find table 1. The former table 2 is now the chapter titled "distributions".
B. What is in table 1 and the distributions chapter? Write it on a separate piece of paper.
C. Open **Examples Vol 1** under the help menu.
D. Find the first example, the famous **Rats** data set. Click on the name to open the example.
E. Page down the example.
   a. Page down to just below the diagram, where you will see the words "BUGS *language for rats example:*". See the word "model" directly below? Click on it, notice what happens. This is the start of the model specification.
   b. Below the model code you will see a bold **BLUE** word **DATA** and two bold arrows with the words "click on one of the arrows to …" between them. Click on the arrow a few times to see what happens.
   c. When you click on the dark arrow to open up the DATA or INITS the arrow becomes hollow and is not bolded. Click again to close up.
   d. This illustrates the three parts to a WINBUGS code:
      i. model specification,
      ii. data
      iii. initial values
   e. Note: there can be more than one data set entered into WINBUGS for a single model,
F. Find the **Stacks** Example, click on the name to open it in another window.

G. What model(s) does the Stacks example illustrate?

Some Notation

WINBUGS uses a non-standard notation for the normal density. Rather than specifying the mean and variance, WINBUGS models specify the mean and precision, which is one over the variance.

$$\text{Precision} = (\text{Variance})^{-1}$$

To denote this, we will write β_0 ~ No (-5.682, .05464), or x ~ No(a,b) meaning that β_0 is normally distributed with mean –5.682 and precision .05464 or x is distributed normally with mean a and precision b (or variance b^{-1}). Using N instead of No, for example x ~ N(a,b^{-1}}) means the same as x ~ No(a,b).

Using WINBUGS.

Model and Data Entry.

Actually do the following commands on your computer while reading this.

Log in to the computer and let Windows come on. There may be a PROGRAM folder on the desktop which has a link to start WinBUGS14. Else click the **Start** menu and then **Programs**. Go down the menu and you'll find **WinBUGS14**, click to start.

Once you have WINBUGS started, a window with a license agreement opens whose window you may close. To start a new file, click on the **File** menu and then hit the **New** option. This will create a new window where you can type in your program. You may resize the window to make things easier to see.

We use the following example from the WINBUGS manual to illustrate how to run a WINBUGS program. Consider a set of 5 observed (x, *y*) pairs (1, 1), (2, 3), (3, 3), (4, 3), (5, 5), with x as covariate and y as response. We shall fit a simple linear regression of *y* on *x*, using the notation

$$y_i \sim \text{No}(mu_i, tau) \qquad (1)$$

$$mu_i = \alpha + \beta(x_i - x.bar) \qquad (2)$$

IMPORTANT: Before you continue: look at this data and guess the slope and intercept estimates yourself without doing any calculations.

A WINBUGS program has three parts: **model** specification, **data** input and **initialization**. The model is the joint distribution of the data and parameters. Suppose that we know that the slope is near 1 and the intercept is rather uncertain and could be positive or negative. The variance tau^{-1} is thought to be near 1. We use the prior distributions

$$\alpha \sim \text{No}(0, .0001)$$

$$\beta \sim \text{No}(1,1),$$

and

$$\text{tau} \sim \text{Gamma}(.25,.25).$$

This says that we think α has mean zero, and standard deviation $100 = .0001^{-1/2}$, that is, we have no clue as to the value of the intercept, and values in the range (-200,200) are all fairly reasonable, although those nearer to zero are more plausible than those near –200. In the parameterization above, by subtracting off x.bar from x, α is the average response at the average of the x's. The slope β has prior mean 1, standard deviation 1. Thus a priori we think β is in the range (-1, 3) with probability around .95. And we think it highly unlikely that β is outside the range (-2,4), for example. For tau, we think an approximate guess at it's value is 1 = .25/.25, and its variance is $4 = .25/.25^2$. This uses the WINBUGS parameterization of the gamma density.

The WINBUGS language allows a concise expression of this model. In the new window (control N), **type in**

```
model
{
    for(i in 1:N){
        y[i] ~ dnorm(mu[i], tau)
        mu[i] <- alpha + beta * (x[i] - x.bar)
        }
    sigma <- 1/sqrt(tau)
    alpha ~ dnorm(0, .0001)
    beta ~ dnorm(1,1)
    tau ~ dgamma(.25,.25)
}
```

Notice the use of the two character string "<-" for assignment, not an equals sign =.

Check the following:

- **(i) spelling,**
- **(ii) that left and right parentheses match**
- **(iii) spacing is correct**.

Sigma is the sampling standard deviation. It is more interpretable than **tau**.

Now enter the data. Data may be represented in the form of an R or Splus object.

```
list(x = c(1, 2, 3, 4, 5), y = c(1, 3, 3, 3, 5), N = 5, x.bar = 3)
```

Type this in. Again check all parentheses, formatting and commas. It can be in the same window as your program, or in a different window, it does not matter. This defines the covariate x, the response y, the sample size N, and the average of the x's, x.bar.

Save your files.

Now we identify our model and data to Winbugs. There are several standard steps for each model / data set combination. We interact with WINBUGS through the menu options. Menu and tool options are **greyed out** until they can be executed.

Go to the Winbugs menus at the top of the window, and click on "model", and go down to **Specification** and select it. This will make the **Specification Tool** appear. Go back to your model, and double click on the word "model", highlighting it. Then click on the **check model** command from the **Model Menu** – 'model is syntactically correct' should appear in the status line which can be found at the bottom left of your window. If it does, go on to the next step, else go back, and attempt to fix the problem with your model. If possible, look for where the cursor is lightly blinking in your program, this may give a hint to where the error is.

If you click "load model" a second time after a valid model has been loaded, you will get a message "**The new model will replace the old model**". Click "OK".

To check and load the data set highlight the key word **list** in your file. Execute the **load data** command from the **Specification Tool**; 'data loaded' should appear in the status line. If it does not, you need to reedit the data. When you have lots of data, you may have several lists of data, and you may click **load data** multiple times. We have very little data, and we only need to load data one time.

Next execute the **compile** command from the **Specification Tool** to set up the data structures used by WinBUGS to do the computations – '**model compiled**' should appear in the status line at the bottom left. If it does not, there is something wrong, and you need to go back and fix your model.

Finally the MCMC sampler must be given some **initial values**. **Type:**

```
list(alpha = 0, beta = 1, tau = 1)
```

and highlight the key word **list** and then execute the **load inits** command from the **Specification Tool** – '**initial values loaded: model initialized**' should appear in the status line. You have been led through the various steps needed to check the model, load the data, compile the model, and load initial values. Initial values are part of the numerical algorithm and have NOTHING to do with the science or the statistics of your data analysis.

The location of the data in the list command can be anywhere in your window or in another WINBUGS window altogether. The main thing is that it can't be in the middle of the model commands. You may also have multiple versions of the data, the model and the initial values in the same file.

Note: You may want to save your files now. WINBUGS has its own file format `.odc`, it also knows about text files. When you load a file, you will need to know the type, and tell WINBUGS correctly what type that is. I usually use a combination of .odc and .txt files with WINBUGS.

If you wish, you may now close the Specification Tool, by clicking it's upper right hand x box. To get it back again, go to the model menu and selection "specification".

Computation and Estimation.

WinBUGS is now ready to do estimation. Recall that the inference in Bayesian inference is a probability distribution. WINBUGS does estimation by sampling from the posterior of the parameters. These samples can be summarized by means and standard deviations or histograms or kernel density estimates may be plotted.

First we select the "**Update**" command from the **model** menu. This command opens a dialog box with two numerical entry fields: ***updates*** for the number of MCMC cycles or updates to be carried out and ***refresh*** for the number of updates between redrawing the screen. The dialog contains one button: ***update*** to start updating the model. This button becomes active when the model has been successfully compiled and given initial values. Click the "update" button. Notice how the iteration counter increments. The dialog box also shows the total number of updates carried out. Before we do anything further, we need another box.

Select the **Samples…** command from the **Inference** menu. This command opens a **Sample monitor tool** dialog box to tell WINBUGS what parameters you are interested in. The variables of interest must be typed in **the *node*** text field. Do the following

Type **alpha** in the node box. Notice that "**set**" is now dark. Click on **set**.
Type **beta** in the node box. Click on **set**.
Type **sigma** in the node box. Click on **set**.
(Type **tau** in the node box. Click on **set**.)

We aren't interested in **tau**, so we might not enter it the node box. If you misspell the name of a random variable, set will not highlight. If we do not enter a parameter in this dialog box, WINBUGS will not be able to give us inference about that parameter at the end. To check that you entered everything successfully, click on the drop down window where you were typing, and see the list of parameters that you entered. You should see alpha, beta, and sigma. If you go ahead and enter tau, you will see that too.

Now go to the **updates tool**. It is found under the model menu. Enter 10000 in the **updates** window. It should have 1000 already; you can append a single "0" if you wish. This model is very small, our computers are fast enough, and this should run very quickly. Click **updates** to start the software going. As the package iterates, you will see the numbers in the iteration box cycling through. When WINBUGS is done, you will see a comment at the lower left stating how much time was used. For larger models, users often do a much smaller run, so they can estimate how long 1000 or 10000 cycles will take. Some people start with 1 or 2 iterations; then move on to 10 or 20. This also helps debug problems when they occur.

The ***beg*** and ***end*** numerical fields in the **Sample Monitor tool** are used to select a subset of the stored iterations (or Gibbs Samples or MCMC samples) for analysis. Typically we throw out the first few samples, then use the remaining samples for inference. Let us use 1000 samples for burn-in. Enter the number 1001 in the "beg" box. We will use the 9000 remaining of the 10000 samples for estimation. If you run more (for example by clicking the "update" button again), then you will use more than 9000 samples for your inference. More is better, as long as the program runs and does not blow up or is not too slow. I often try for 100000 after burn in, and have used as many as 1000000 (for complex models).

To get the summary inferences, go to the node in **Sample Monitor Tool** and using the dropbox, select beta. All of the boxes should darken in the Sample monitor tool now. Click on **density**. Another window should open, which you may wish to widen to view properly. Also, it may be hidden so you may need to move it around. If you lose a window, go to the **Window** menu to find it again. How do we interpret the resulting density? Where the density (y-value on plot) is high, those corresponding x-axis values are values that beta is likely to be, while where the density is low are values that beta is unlikely to be. You can click on the density and resize it to get a better look at it.

Now click on **stats**. You will get a numerical summary of the posterior in another window. WINBUGS reports the posterior mean and standard deviation. An estimate (which I ignore, it's usually way too small when it matters) of the error in the calculation of the mean is also reported. Also given are the lower 2.5%, median and upper 97.5% points of the distribution. A 95% posterior interval (ie confidence interval) is given by the 2.5% and 97.5% points. You can select other quantiles to be reported by selecting them in the box at the right of the sample monitor tool.

You can repeat these last two paragraphs for alpha and sigma. More efficient: instead of using the dropbox, type "*" (no quotes, just the asterisk) in the node box. Now repeat the last two paragraphs. This gives you inferences for all parameters in the model that you entered into the node box.

Recall that MCMC is iterative. You can get a picture of what happened over time in the MCMC routine by clicking the **history** and **trace** buttons. History gives a complete listing of the values for that parameter over time, while trace gives a much smaller set. If you click trace and leave it open and then click on the **update** button in the update tool, you will see the trace window update. This particular model's computation is very fast, so the changing pictures may be difficult to decipher, but in more complicated, slower models, there is something of interest to see in this window.

This simple linear regression with a normal prior for alpha and beta and a gamma prior for tau is a mathematically well behaved model. The algorithm used by WINBUGS to get estimates is well behaved here. In particular, it converges quickly, mixes well, and the summary measures that we get from WINBUGS are of good quality. In some complex models, the algorithm is not well behaved, and you need to run the algorithm for a long time (large number of cycles). We inspect the **history** and the **autoC** buttons to help with diagnosing *poor convergence*. The current model is an example of nearly perfect convergence. Later we will have an example of decent but not perfect convergence. Poor convergence occurs when the history plot looks more like ocean waves than a porcupine back; in the autocorrelation plot from the autoC button, if the bar chart is high even out at lag 50 (far right of the plot), that is a bad sign. More on this later.

A summary of the steps:

Step 1: Run Burn-in period. Use **Update** to run a sample of size L, say 1000, iterations. This is a simple model, and this shouldn't take too long to run; also, WINBUGS is wickedly fast.
Step 2: Use **Samples** to choose the parameters of interest and tell WINBUGS to save output for those parameters. You should type alpha beta, and sigma and maybe tau and click on the "set" button each time.
Step 3: Use **Update** command to run more iterations. Let us do 9000 more iterations. If this is suitably fast on your computer, please do more. I usually shoot for 100000 if possible.
Step 4: Go back to **Samples** to get the results. Put 1001 in the **beg** box and 10000 (or higher) in the **end** box. You will have to adjust these numbers depending on your model, and the number of samples that you would like.
Step 5: You can use **stats** button to get the basic statistics for each parameters and can also use the **density** button to get plots of the posterior densities of each parameter.

Inspect the results from the analysis for this simple data set. Is beta estimated to be close to 1 as you'd expect? Is the estimate of sigma sensible? Note the standard error for sigma. Check the posteriors. What are their shapes? Have you ever seen a confidence interval for a variance before? All parameters and unknowns are treated equally by Bayesian inference.

To turn in at the end of the lab

Questions:

1. What information is in table 1of the manual?
2. What information is in the distributions chapter of the manual?
3. What model(s) does the Stacks example use?
4. Turn in properly formatted output from our regression model in this lab (posterior distributions, posterior summary statistics, trace and autocorrelation –all of this for all the parameters monitored). Please, use screen shots of Bugs Ouput copy pasted into a Word file. Interpret your plots and summaries.

Homework to turn in on Monday

5. Change the prior precision for beta to 10000, 100, 10, 1, and .01. The prior precision is the number 10000 in the statement *beta ~ dnorm(1,10000)* in your model program.

   Run the model for each of these values. You will have to enter beta in the Sample Monitor Tool each time, as it does not store the variable names after each run. What happens to the estimate of beta as the prior precision changes? To answer this, (a) Report a table of the estimates as a function of the prior precision. Also report the posterior standard deviation of beta. (b) As the prior precision goes to +infinity, what do you suppose the limit of the values of the estimate and sd are? (c) The least squares estimate of beta is .8. From your empirical study, what is the limit of the Bayesian estimate as the prior precision goes to zero?

6.-This is a home reading assignment:
WINBUGS 1.4.1 manual chapters

1 Introduction:	Skim
2 Compound Documents:	Read
3 Model Specification	Read (skim anything about doodlebugs) See table 1.
5 The Model Menu	Read
6 Inference Menu	Read
7 Info Menu	Read
8 Options Menu	Read
12 Tips and Troubleshooting	Read
13 Tutorial	Read & follow in the program
17 Distributions	Read thoroughly! Worth re-reading.

Other chapters are definitely worth reading.
Read Tutorial AFTER lab.

Inspect and run a sample of 3 examples.

7.- Bring a small sample of your data, properly formatted as a list, to play with it in the lab on Monday.