

Introduction to WinBUGS

WinBUGS is a free program available from the Biostatistics Unit of the Medical Research Council in the UK (see link on course web page). Installing WinBUGS is straightforward, one downloads the single file required, typically called winbugs14.exe, and runs that file to install in typical windows fashion. However, this allows only a limited use of WinBUGS, and one also need to install a key (again, this is free) for unlimited use. Instructions are on the WinBUGS home page, and please let me know if you experience any difficulty installing the program or the key. All computers in the basement of Purvis Hall have WinBUGS installed on them for your use.

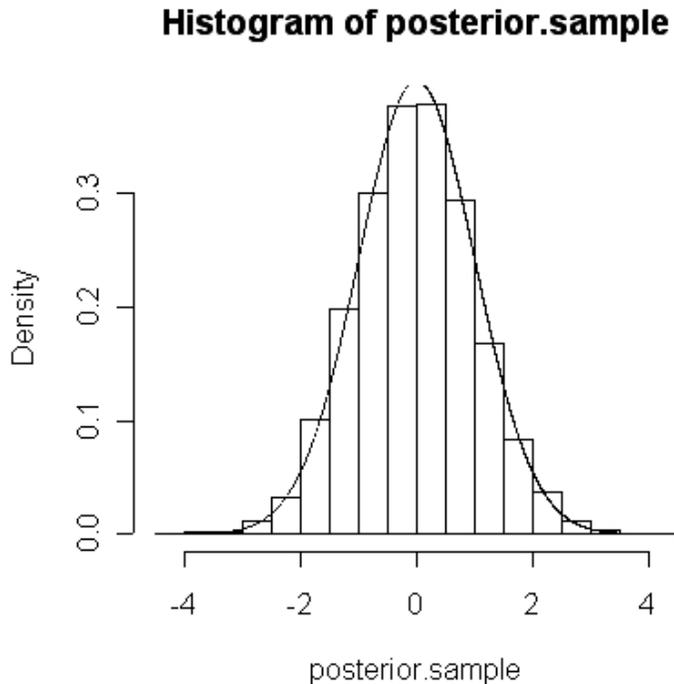
Once installed, a WinBUGS program consists of three parts, all of which can be placed into a single file, or as three separate files (or two files):

Main program: A set of lines that let WinBUGS know what the prior and likelihood of the model are.

Data set: The data that will be used, either formatted as an R like list, or in rectangular format. Note that you can have as many data sets as you wish, and can mix formatting types when entering data.

Initial Values: These are optional (but generally a good idea to include), and are used by WinBUGS to start its algorithm.

WinBUGS solves problems in Bayesian analysis by multiplying the prior by the likelihood, and then taking samples from the posterior distributions via an iterative algorithm called the Gibbs sampler. Thus, rather than get an exact formula for the posterior distribution, WinBUGS returns samples from it, as shown below:



WinBUGS can be used to solve a *very* wide class of problems, much wider than standard software packages. It must be used carefully, as the Gibbs sampler algorithm does not always converge. Good practice dictates to run WinBUGS several times (say, three times) to ensure that the answer is always the same (or extremely close). For problems in this course, convergence will seldom, if ever, be a problem.

Running WinBUGS

Follow these steps to produce analyses in WinBUGS:

1. Open WinBUGS by clicking on the WinBUGS icon on desktop (after installation).
2. Either open an existing WinBUGS file (typical extension is *.odc) if continuing from an already created file, or open a new window from the file menu.

3. Type in the program (or cut and paste if a program already exists, typically will be the case in this course).
4. Below the program (or if you prefer, in another window inside of WinBUGS), type in or cut and paste in the data set, remembering that the format must be either rectangular or list-like (see examples on course web page).
5. In list format, type in initial values (see examples on web page).

For our normal problem with unknown variance, the WinBUGS window will typically look like this:

```

model
{
    # Program must start with model
    # statement and open {
    for (i in 1:27)      # Loop over number of data points
    {
        x[i] ~ dnorm(mu,tau) # Enter the 27 likelihood terms
    }
    mu ~ dnorm(70,0.04) # Prior for the normal mean, mu, 0.04=1/25
    sigma ~ dunif(0,50) # Prior for the normal standard deviation, sigma
    tau<-1/(sigma*sigma) # WinBUGS normal form needs 1/sigma^2
    }
    # End of program part

# Data                # Data section

list(x=c(76, 71, 82, 63, 76, 64, 64, 74, 70, 64, 75, 81, 75, 78, 66,
62, 79, 82, 78, 62, 72, 83, 79, 41, 80, 77, 67))

# Initial Vlaues      # Initial Values section

list(mu=60, sigma = 20)

```

6. Once all input is ready, WinBUGS must be run to produce the output, following these steps:
 - (a) Open the Model → Specification menu item, highlight the word “model” in the program, and then click on “check model”. This checks the model for syntax errors. Watch for (often quite cryptic) error messages on bottom status line, if any errors occur.

- (b) Highlight the word “list” in the data section of the program, and click on “load data”. This loads in your data. Repeat this step as many times as needed to load in all your data. If in rectangular array style, highlight the first line (variable names) rather than the word “list”.
- (c) Click on compile (assuming all is fine so far...if not, find error(s) and repeat first two steps).
- (d) Highlight the word “list” in the initial values section of the program, and click on “load inits”. This loads in your initial values. If you did not provide all initial values, click on “gen inits” to have WinBUGS do this automatically for you (not always a good idea).
- (e) WinBUGS can now be run to produce posterior distributions. First, run some “burn-in” or “throw away” values, to ensure convergence. Typically, 2000 should be (more than) sufficient for models in this course. To create the burn-in, open the Model → Update menu item, change the default 1000 value to 2000, and click on the “update” button.
- (f) To run further iterations “to keep”, open the Inference → Samples menu item. Enter all unknown variables you wish inferences on into the window (in this case, mu and sigma), clicking “set” after each one. Then go back to the update box, which should still be open, and run the number of iterations you want. We will use 10,000, so enter 10000 and click again on the update button.
- (g) Finally, we can look at summaries of the posterior distributions. Going back to the Samples dialogue box, type a “ * ” in the window (meaning look at all variables tracked), and then click on the various items to get the desired summaries. Most useful for us will be “Stats” and “density”, to get basic posterior inferences such as means, medians and 95% credible intervals, and density plots (somewhat crude). The results are:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	71.63	1.727	0.01698	68.24	71.62	74.97	2001	10000
sigma	9.69	1.43	0.01567	7.358	9.532	12.92	2001	10000

For the mean, note that our posterior has mean of about 71.6 (compare to 71.69 found previously), and the standard deviation for this parameter is 1.73, compared to 1.63 found previously. As we have an extra parameter to estimate, this decrease in the accuracy in

estimating μ is expected. Note that we have a new parameter to estimate, σ , which was assumed known before.