

Lab 8: Introduction to WinBUGS

- Goals:**
1. Introduce the concepts of Bayesian data analysis.
 2. Learn the basic syntax of WinBUGS.
 3. Learn the basics of using WinBUGS in a simple example.

Next lab we will look at two case studies: (1) NMMAPS and (2) Hospital ranking data.

PART I WinBUGS and Bayesian Analysis

A. What is WinBUGS

BUGS = **B**ayesian **I**nferece **U**sing **G**ibbs **S**ampling

Not using Windows? Try

OpenBUGS: <http://mathstat.helsinki.fi/openbugs>

JAGS: <http://www-fis.iarc.fr/~martyn/software/jags>

- WinBUGS is a Bayesian analysis software that uses Markov Chain Monte Carlo (MCMC) to fit statistical models.
- WinBUGS can be used in statistical problems as simple as estimating means and variances or as complicated as fitting multilevel models, measurement error models, and missing data models.
- WinBUGS fits fixed-effect and multilevel models using the Bayesian approach. Stata fits fixed effects and limited multilevel models using maximum likelihood or generalized least squares.
- Often results from WinBUGS and Stata are very similar.

B. What is Bayesian Analysis

Consider the following conditional probability statement:

$$P(\theta | data) = \frac{P(data | \theta) P(\theta)}{P(data)},$$

where θ is the unobserved parameter that we want to learn about using the observed data.

There are three important components:

(1) $P(data | \theta)$:

- This represents the “**model**” part in a statistical analysis.
- It describes our assumptions that the observed data were generated based on the parameter θ .
- E.g. θ is the mean of a Normal distribution with variance 1.
- E.g. θ is the coefficients in a linear regression model where data here include both the response (Y) and covariates (X).

(2) $P(\theta)$:

- This represents “**prior**” assumption of θ .
- It describes our prior belief of θ , typically using a distribution.
- E.g. Assume $\theta \sim \text{Normal}(0, 20) \cdot I_{\theta > 0}$, so θ is strictly positive.
- E.g. Assume $\theta \sim \text{Normal}(0, 10^5)$, a pretty non-informative prior.
- Choosing appropriate priors can be tricky and we will see many examples that are typically used in standard statistical analyses.

(3) $P(\theta | \text{data})$:

- This represents the “posterior” distribution of θ .
- It describes all the information on θ after combining prior knowledge on θ and what our data informed us about θ .
- Since $P(\theta | \text{data})$ is a probability distribution, statistical inference is made by examining the different characteristics of this distribution.
- E.g. the posterior mean, median or mode can be our estimate of θ .
- E.g. the variance and middle 95% of the posterior distribution can tell us about the uncertainty in our estimates.

Therefore, Bayesian data analyses typically involve the following ingredients:

- (1) Specify a model that specifies the relation between the unknown parameters and the observed data.
- (2) Specify prior distributions for the unknown parameters.
- (3) Obtain the posterior distributions.
- (4) Make inference using the posterior distributions.

C. Why Do Bayesian Analysis

Here are some advantages of the Bayesian approach:

- All uncertainty in parameter estimation is included in the final inference. (E.g. Bayesian versus empirical Bayes estimates of random effects).
- Estimation (particularly the uncertainty) for any function of the parameters can be easily obtained by examining the corresponding posterior distribution.
- Prior information can be easily integrated.
- Does not rely on large-sample asymptotic theory.

Here are some elements that make Bayesian analysis more complex:

- Need to specify prior distributions.
- Only in very simple models can $P(\theta | \text{data})$ be derived explicitly.
→ Solution: Monte Carlo sampling!!!

Markov Chain Monte Carlo?

- Since we specify $P(\text{data} | \theta)$ and $P(\theta)$, the only element that prevents us from obtaining $P(\theta | \text{data})$ is the marginal distribution $P(\text{data})$.
- However, $P(\text{data})$ is often difficult to evaluate, especially when the number of parameters is large.
- Note that $P(\text{data})$ does NOT depend on the parameter θ . Many methods have been developed to draw samples from $P(\theta | \text{data})$ and WinBUGS does this for us automatically!!!
- Given samples from $P(\theta | \text{data})$, we can calculate the desired statistics such as the mean or variance to make statistical inference. The precision of how our samples resemble the true posterior distribution is only limited by the number of draws we make.

PART II WinBUGS in Action

A. How to install WinBUGS

- (1) Download the .exe. file from: <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>
- (2) Find the file WinBUGS14.exe where WinBUGS is installed. You may want to create a shortcut.
- (3) Follow the instructions in the patch file and install the patch in WinBUGS
- (4) Fill out a registration form to obtain a key through email.
- (5) Update the key in WinBUGS.

B. A Simple Demonstration: Inference for Two Normal Means.

Data: two samples of size 20 from two independent Normal distributions with unknown variances.

Data: $X = (x_1, x_2, \dots, x_{20})$ and $Y = (y_1, y_2, \dots, y_{20})$

(1) Statistical model:

$$\begin{aligned} x_i &\sim \text{Normal}(\mu_1, \sigma_1^2) & i = 1, 2, \dots, 20 \\ y_i &\sim \text{Normal}(\mu_2, \sigma_2^2) & i = 1, 2, \dots, 20 \end{aligned}$$

(2) Prior distributions:

In our model, we have four unknowns so four prior distributions are needed.

$$\begin{aligned} \mu_1 &\sim \text{Normal}(0, 100^2) & \mu_2 &\sim \text{Normal}(0, 100^2) \\ 1/\sigma_1^2 &\sim \text{Gamma}(0.001, 0.001) & 1/\sigma_2^2 &\sim \text{Gamma}(0.001, 0.001) \end{aligned}$$

- The Normal distributions for μ 's are flat and cover a large range of values.

- We set the inverse of the variance to have a gamma prior distribution since gamma distribution only takes positive values. Gamma (0.001,.001) has **extremely** large standard deviation.
- We pick the above prior distributions such that they are non-informative in that the data will easily dominate the posterior distributions.

A typical WinBUGS program include three sections: Model, Data, and Initial Values.

Model: Translating our statistical model into a WinBUGS program:

```

model{
  for (i in 1:20){
    x[i] ~ dnorm (mu[1], prec[1])
    y[i] ~ dnorm (mu[2], prec[2])
  }

  mu[1] ~ dnorm (0, 0.0001)
  mu[2] ~ dnorm (0, 0.0001)
  prec[1] ~ dgamma (0.001, 0.001)
  prec[2] ~ dgamma (0.001, 0.001)

  s2[1] <- 1/prec[1]
  s2[2] <- 1/prec[2]
}

```

} Model, P(θ | data)
 } Priors, P(θ)
 } Convert variances to precision

- “model { ... }” specifies the statistical model we are fitting.
- “for (i in 1:20) { ... }” is short-hand for writing the two statements in {...} 20 times.
- The square brackets allow us to index a vector of values. They are equivalent to the subscripts in our previous model.
- WinBUGS uses “**precision**” as a parameter in specifying a Normal distribution instead of variance!!!
 - precision = 1/variance
 - dnorm (0, 0.0001) is the same as a Normal distribution with mean 0 and variance $1/0.0001 = 100^2$.
- The last two lines tell WinBUGS to also keep track of the variances.

Our data in WinBUGS format:

```

list(x = c(6.62, 6.71, 5.07, 4.39, 5.68, 3.94, 5.83, 2.31, 3.60, 4.64,
1.79, 3.12, 3.46, 8.25, 5.49, 6.49, 2.65, 9.14, 5.31, 6.58), y =
c(9.06, 7.00, 8.59, 8.70, 8.64, 8.03, 9.27, 6.01, 7.92, 6.20, 6.39,
9.10, 7.63, 6.75, 8.88, 8.44, 8.95, 5.66, 9.78, 8.09))

```

Often we also need to give initial values for our parameters:

```

list( mu=c(0,0), prec=c(1,1))

```

Run the Analysis in WinBUGS

A. Load model, data, and initial values:

1. Open a new document in WinBUGS and paste all three parts (model, data, initial values) on it.
2. Save the file as an .odc.
3. From the top panel: open Model → Specification. A dialogue box will open.
4. Double-click (highlight) the word “model” in your file and click “check model” on the dialogue box.
5. Look at the bottom left-hand corner for “model is syntactically correct.”
6. Highlight the word “list” for the data section and click “load data” in the dialogue box. Look at the bottom left-hand corner for “data loaded.”
7. Click “compile” and look for “model compiled.”
8. Highlight the word “list” for the initial value section and click “load inits” in the dialogue. Look for “model is initialized.”
9. [Optional] If you did no initialize all parameters, click “gen inits” in the dialogue box.

B. Run Sampler:

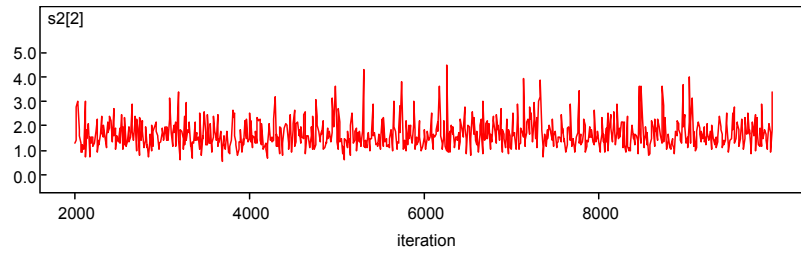
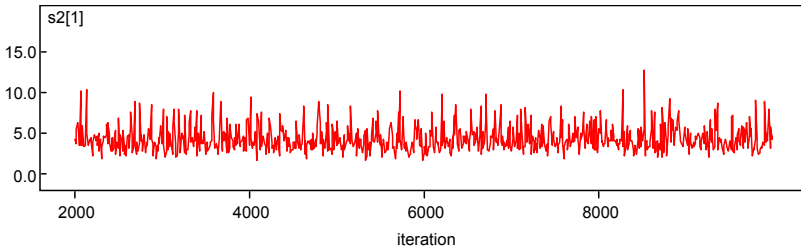
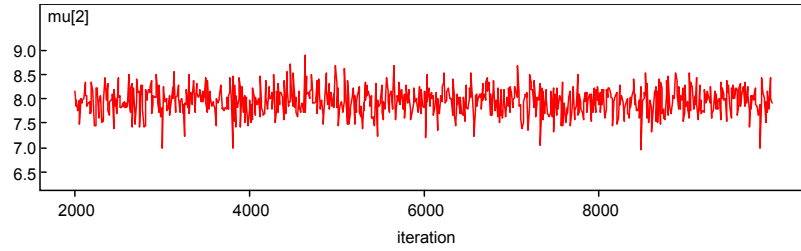
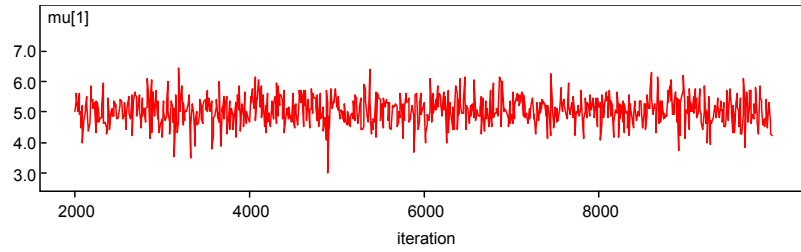
1. Open the “Sample Monitor Tool” window: Menu → Inference → Sample,
2. Type the parameters we are interested in the “node” box and click “set.” In this example we will track both “mu” and “s2.”
3. Open the “Update Tool” window: Menu → Model → Update.
4. In the Update Too box, type in the number of posterior samples you want in the “updates” box. E.g. 20000.
5. Click “Update” and watch it runs in the iteration box!

C. Posterior Inference:

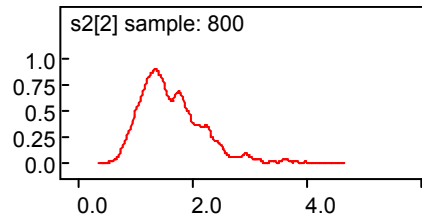
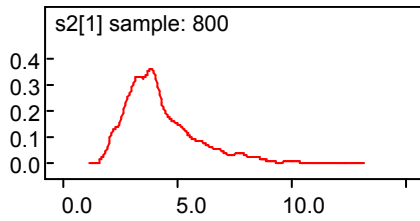
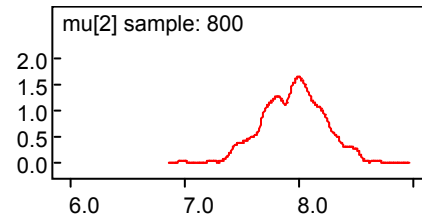
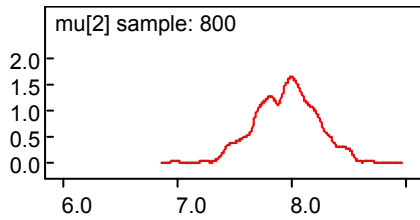
1. In the “Sample Monitor Tool” select “mu” from the drop-down box in “node.”
2. Select the number of initial samples that we want to drop in the “beg” box. This is also known as “burn-in.” Let’s choose 2000 here.
3. Select the “jth” number of iteration you want to keep in the “thin” box. For example, if you pick 10, then only the every 10th sample from the 2000th~10000th iterations will be used for posterior inference. Let’s choose 10 here.

Some Posterior Inference:

History Show sampled value at each iteration. Look for “chains” that show no particular patten and low auto-correlation.



Density: Plot the density estimates of the parameters.



Stats: Show statistics of the posterior distribution based on the samples.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[1]	5.051	0.4654	0.01625	4.134	5.051	5.99	2000	800
mu[2]	7.957	0.2765	0.01001	7.429	7.972	8.498	2000	800
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
s2[1]	4.312	1.605	0.04095	2.148	3.944	8.394	2000	800
s2[2]	1.662	0.5761	0.01988	0.8416	1.549	3.122	2000	800

Parameter	Truth	Posterior mean	95% Posterior Interval	MLE	95% Confidence Interval
μ_1	5	5.05	(4.13, 5.99)	5.05	(4.14, 5.96)
μ_2	8	7.96	(7.43, 8.50)	7.95	(7.38, 8.52)
σ_1^2	4	4.32	(2.15, 8.39)	3.79	(2.19, 8.08)
σ_2^2	1	1.66	(0.84, 3.12)	1.48	(0.85, 3.15)

- The point estimates and 95% posterior interval for the 2 means are very similar to the MLE estimates and its large sample 95% confidence interval.
- The point estimates for the variances are a bit different. Why?

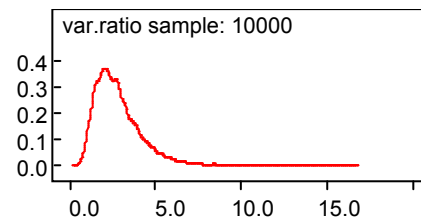
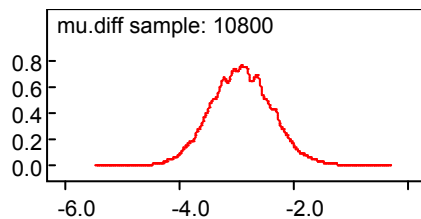
Some Additional Analyses

Perhaps we are also interested in:

- 1) The difference of the population means.
- 2) The ratio of the two variance components.

In standard non-Bayesian analysis, the confidence intervals for these estimates can be quite tricky. However, in a Bayesian analysis we simply add the following two lines in the “model” section of our WinBUGS code.

```
mu.diff <- mu[1] - mu[2]
var.ratio <- s2[1]/s2[2]
```



node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu.diff	-2.909	0.5467	0.004712	-3.965	-2.913	-1.811	9201	10800
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
var.ratio	2.847	1.442	0.01527	1.013	2.555	6.47	10001	10000

We conclude the difference between μ_1 and μ_2 is -2.91 with a 95% posterior interval of (-3.96 ~ -1.81) and the ratio between σ_1^2 and σ_2^2 has a posterior median of 2.55 (95% PI: 1.01~6.47). Therefore we found evidence that μ_1 less than μ_2 and σ_1^2 is greater than σ_2^2 .

Stata has a package that allows you to run WinBUGS in Stata (link below). It still requires you to write WinBUGS program but you can analyze the posterior samples in Stata. R has similar libraries (BRugs and R2WinBUGS) that call WinBUGS or OpenBUGS. JAGS is another MCMC software that you can call from R and is not based on BUGS.

Run WinBUGS in Stata:

<http://www2.le.ac.uk/departments/health-sciences/extranet/BGE/geneticepidemiology/gedownload/information/>