A Short Introduction to Bayesian Modelling Using WinBUGS

BUGS

Ioannis Ntzoufras

Associate Professor in Statistics

E-mail: ntzoufras@aueb.gr

Department of Statistics, Athens University of Economics & Business

Contents

0… Bibliography

WinBUGS and Related Software

1. ***WINBUGS 1.4.3***
   - ✓ available at http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14.exe
   - ✓ Registration page http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/register.shtml [FREE]
2. ***WinBUGS Development Site***
   - ✓ available at http://www.winbugs-development.org.uk/
3. ***WinBUGS Online resources (incl. Addins)***
   - ✓ available at http://www.mrc-bsu.cam.ac.uk/bugs/weblinks/webresource.shtml
4. ***OpenBUGS*** available at http://www.openbugs.info/w/
5. ***Classic BUGS 0.6,***
   - ✓ available at http://www.mrc-bsu.cam.ac.uk/bugs/classic/bugs06/prog06.exe
6. ***CODA and R-CODA (software for convergence diagnostics)***
   - ✓ available at http://www.mrc-bsu.cam.ac.uk/bugs/classic/coda04/cdaprg04.exe
   - ✓ R-coda available at http://www-fis.iarc.fr/coda/
7. ***BOA (Excellent CODA clone for R – software for convergence diagnostics)***
   - ✓ available at http://www.public-health.uiowa.edu/boa/

0… Bibliography

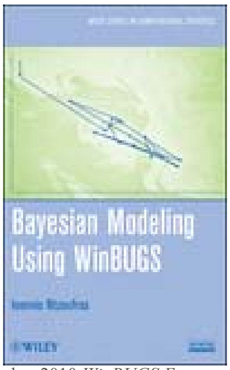
WinBUGS Manuals and Online Resources

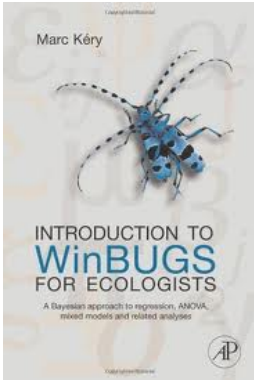
- Spiegelhalter, D., Thomas, A., Best, N. and Lunn, D. (2003). *WinBUGS User Manual, Version 1.4*,
  - available at http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/manual14.pdf
- WinBUGS Examples
  - Vol 1, http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/Vol1.pdf
  - Vol 2, http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/Vol2.pdf
  - Vol 3, http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/Vol3.pdf
  - Additional New Examples http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/examples.shtml
- GEOBUGS Manual http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/geobugs12manual.pdf
- Additional electronic material (tutorial, courses papers) for WINBUGS
  - available at http://www.mrc-bsu.cam.ac.uk/bugs/weblinks/webresource.shtml .
- WinBUGS THE MOVIE – Online tutorial
  - available at http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/winbugsthemovie.html
- Additional documentation and manuals for BUGS/CODA available at http://www.mrc-bsu.cam.ac.uk/bugs/documentation/contents.shtml .

0… Bibliography

WinBUGS Books

1. Ntzoufras, I. (2009). *Bayesian Modelling Using WinBUGS*. Wiley.
2. Kery, M. (2010). *Introduction to WinBUGS for Ecologists: Bayesian approach to regression, ANOVA, mixed models and related analyses*. Academic Press.
3. Surhone L.M., Timpledon, M.T., Marseken S.F. (2010). *WinBUGS*. Betascript Publishing





0… Bibliography

Bayesian Data Analysis Books

- Carlin B. and Louis T. (2008). *Bayes and Empirical Bayes Methods for Data Analysis*. 3rd edition, London: Chapman and Hall.
- Gelman A., Carlin J.B., Sten H.S. and Rubin D.B. (2003). *Bayesian Data Analysis*. 2nd edition. London: Chapman and Hall.
- Bolstad W.M. (2007). *Introduction to Bayesian Statistics*, 2nd Edition, Wiley-Blackwell.
- Jackman, S. (2009). *Bayesian Analysis for the Social Sciences*, Wiley Series in Probability and Statistics, Wiley-Blackwell.
- Marin J.M. and Robert C. (2007). *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*, Springer Texts in Statistics.

0… Bibliography

Bayesian Modeling Books

- Books of P.D. Congdon:
  1. (2010). *Applied Bayesian Hierarchical Methods*. Chapman and Hall/CRC.
  2. (2007). *Bayesian Statistical Modelling*. 2nd Edition. Willey and Sons.
  3. (2003). *Applied Bayesian Modelling*. Wiley-Blackwell
  4. (2005). *Bayesian Models for Categorical Data*. Wiley-Blackwell.
- Gelman A. and Hill J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models,* Analytical Methods for Social Research, Cambridge University Press.
- Dey D., Ghosh S.K. and Mallick B.K. (2000). *Generalized Linear Models: A Bayesian Perspective*, Chapman & Hall/CRC Biostatistics Series, CRC Press.

1… GENERALISED LINEAR MODELS

- **1.1. Data**
- **1.2. Three Main Components**
- **1.3. General Principles of Statistical Modelling**

1… Generalised Linear Models
1.1. Data

1) RESPONSE VARIABLE (Y): also called dependent or endogenous variable
   - Y is a random variable
2) EXPLANATORY VARIABLES (X_j): Independent or Exogenous variables
   - X_j are usually assumed fixed by the experiment

1… Generalised Linear Models
1.2. Three main components

(1) RANDOM COMPONENT
- $Y_i \sim$ DISTRIBUTION $f(\boldsymbol{\theta}_i)$
- $\boldsymbol{\theta}_i$: VECTOR OF PARAMETERS FOR SUBJECT i

(2) SYSTEMATIC COMPONENT
- $\eta_i = \beta_0+\beta_1 X_{1i}+\ldots+ \beta_p X_{pi}$
- η_i : Linear Predictor of the model

1… Generalised Linear Models
1.2. Three main components

(3) LINK FUNCTION

- Connects the random component & the linear predictor
- $g(\boldsymbol{\theta}_i) = \eta_i = \beta_0+\beta_1 X_{1i}+\ldots+ \beta_p X_{pi}$
- Usually $\boldsymbol{\theta}$ is the mean of Y

1… Generalised Linear Models
1.3. General Principles of Modelling

- It is art
- All models are wrong
  - ✓Some of them are more useful than others
  - ✓We seek for models which describe reality
  - ✓We fit and check many different models
- Always use some diagnostics for checking the goodness of fit

2…Introduction to Bayesian Inference

2.1. The Bayesian Paradigm

2.2. Posterior distribution

2.1 The Bayesian Paradigm

The usual classical approach

- is based on the likelihood function $f\,(\boldsymbol{y}|\boldsymbol{\theta})$
- $\boldsymbol{\theta}$ parameter vector => unknown parameters that we wish to estimate
- Estimation of $\boldsymbol{\theta}$ is achieved via some estimators with some good statistical properties such as unbiasness
- Usually we obtain "good" estimators by maximising the likelihood function (maximum likelihood estimators or MLEs)
- EXAMPLE: for $Y_i \sim N(\mu,\sigma^2)$
  we estimate μ using the sample mean given by $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

The Bayesian approach

- Assumes that the parameters are random variables and not fixed unknowns.
- Specifies the prior distribution $f(\theta)$
- Inference is based on the posterior distribution $f(\theta|y)$ which combines information coming from both the prior distribution and the likelihood (i.e. the data)

The Bayesian approach

Advantages

- Pure probability based approach
- Can incorporate information coming from experts or from previous studies (meta-analysis) via the prior.

Disadvantages

- Subjectivity (via the prior)
- Difficulties in computing or interpreting the posterior distribution

The Bayesian approach

Posterior distribution is calculated using
BAYES THEOREM

$$f(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{f(\boldsymbol{\theta}, \mathbf{y})}{f(\mathbf{y})} = \frac{f(\mathbf{y} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(\mathbf{y})}$$

$$\propto f(\mathbf{y} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta})$$

Posterior $\propto$ Likelihood x Prior

[proportional]

A simple example: *Posterior distribution of the mean of the normal distribution*

1. Data/Likelihood: $Y_i \sim N(\mu, \sigma^2)$
   σ^2 here is assumed to be known and constant
2. Prior: $\mu \sim N(\mu_0, \sigma_0^2)$
3. Posterior:

$$f(\boldsymbol{\theta} \mid \mathbf{y}) = N\left(w\bar{y} + (1-w)\mu_0,\ w\frac{\sigma^2}{n} \right)$$

$$w = \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2 / n}$$

2.2. Posterior distribution

Analytical Calculation of the posterior distribution is sometimes difficult

- **1970s**: Conjugate priors resulting in posteriors of the same type (and known form)
- **1980s**: Asymptotic approximations of the posterior
- **1990s**: Obtaining random samples from the posterior using Markov Chain Monte Carlo (MCMC) methods.

3. Markov Chain Monte Carlo (MCMC) Methods

Introduction

3.1. Metropolis-Hastings Algorithm

3.2. Gibbs Sampling

3. Markov Chain Monte Carlo (MCMC) Methods

Existed in the past in physics

- **1954** Metropolis *et al*. (Metropolis Algorithm)
- **1970** Hastings (Metropolis-Hastings Algorithm)
- **1984** Geman and Geman (Gibbs Sampling)
- **1990** Smith *et al*. (Implementation of MCMC methods in Bayesian problems)
- **1995** Green (Reversible Jump MCMC)

3. Markov Chain Monte Carlo (MCMC) Methods

What is the idea:

Since we cannot analytically calculate the posterior distribution then we generate a random sample from this distribution and estimate the posterior

- Describe the posterior using posterior summaries estimated by the generated sample (e.g. posterior mean or variance)
- Plot marginal posteriors
- Estimate posterior dependencies using sample correlations etc.

3. Markov Chain Monte Carlo (MCMC) Methods

The logic:

We construct a Markov chain which has a stationary distribution the posterior distribution of interest

Every iteration (step) of the algorithm depends only on the previous one.

We use this chain to "generate" a sample from the stationary (target) distribution

3. Markov Chain Monte Carlo (MCMC) Methods

The procedure

- We specify some arbitrary initial values $\boldsymbol{\theta}^{(0)}$ for the parameters $\boldsymbol{\theta}$
- For t=1,2, …, T we generate random values $\boldsymbol{\theta}^{(t)}$ according to our algorithm
- When the chain has *converged* then we have values from the stationary distribution
- We eliminate the initial K values to avoid any possible effect due to the arbitrary selection of initial values. (*Burn-in* period)

3. Markov Chain Monte Carlo (MCMC) Methods

Terminology

- **Initial values**: Starting values $\boldsymbol{\theta}^{(0)}$ of the parameter vector $\boldsymbol{\theta}$. They are used to initialize the algorithm.
- **Iteration**: Refers to one iteration of the algorithm => to one observation of the generated sample
- **Burn–in Period**: The period (and the number of iterations) until the algorithm stabilizes and starts to give random values from the posterior distribution

3. Markov Chain Monte Carlo (MCMC) Methods

Terminology (2)

- **Convergence**: When the chain is giving values from the stationary (target) distribution
- **Convergence diagnostics**: Tests to assure convergence
- **MCMC output**: The simulated sample

3. Markov Chain Monte Carlo (MCMC) Methods

Terminology (3)

- MCMC algorithms are based on Markov chains
  => the generated sample is not IID
  => i.e. there is *autocorrelation* between the subsequently generated values (as in time series data)
- We are interested to eliminate this autocorrelation
  1. We monitor autocorrelations using ACF plots
  2. If there are significant ACs of order L
     => we keep 1 iteration every L
- **Thin**: is the number of iterations we eliminate in order to keep one iteration.

  Thinning can be also used to save storing space.

3. Markov Chain Monte Carlo (MCMC) Methods

ALGORITHMS

- METROPOLIS-HASTINGS ALGORITHM
- GIBBS SAMPLING
- MANY OTHERS MORE ADVANCED (too much for this sort course)

3.1. Metropolis–Hastings Algorithm

- If we are in t iteration of the algorithm => set $\boldsymbol{\theta}^{cur}=\boldsymbol{\theta}^{(t-1)}$ i.e. the current values of $\boldsymbol{\theta}$.
- Generate a new proposed (or candidate) values $\boldsymbol{\theta}^{prop}$ from a proposal distribution $q(\boldsymbol{\theta}^{prop}|\boldsymbol{\theta}^{cur})$.
- Calculate $a = \min\left\{1, \dfrac{f(\boldsymbol{\theta}^{prop} \mid \boldsymbol{y})q(\boldsymbol{\theta}^{cur} \mid \boldsymbol{\theta}^{prop})}{f(\boldsymbol{\theta}^{cur} \mid \boldsymbol{y})q(\boldsymbol{\theta}^{prop} \mid \boldsymbol{\theta}^{cur})}\right\}$
- Set $\boldsymbol{\theta}^{(t)}=\boldsymbol{\theta}^{prop}$ with probability α και $\boldsymbol{\theta}^{(t)}=\boldsymbol{\theta}^{cur}$ with probability $(1-\alpha)$

3.1. Metropolis–Hastings Algorithm

- Note that for the calculation of α we do not need to know the normalizing constant since

$$a = \min\left\{1, \frac{f(\boldsymbol{\theta}^{prop} \mid \boldsymbol{y})q(\boldsymbol{\theta}^{cur} \mid \boldsymbol{\theta}^{prop})}{f(\boldsymbol{\theta}^{cur} \mid \boldsymbol{y})q(\boldsymbol{\theta}^{prop} \mid \boldsymbol{\theta}^{cur})}\right\}$$

$$= \min\left\{1, \frac{\left\{f(\boldsymbol{y} \mid \boldsymbol{\theta}^{prop})f(\boldsymbol{\theta}^{prop})/f(\boldsymbol{y})\right\}q(\boldsymbol{\theta}^{cur} \mid \boldsymbol{\theta}^{prop})}{\left\{f(\boldsymbol{y} \mid \boldsymbol{\theta}^{cur})f(\boldsymbol{\theta}^{cur})/f(\boldsymbol{y})\right\}q(\boldsymbol{\theta}^{prop} \mid \boldsymbol{\theta}^{cur})}\right\}$$

$$= \min\left\{1, \frac{f(\boldsymbol{y} \mid \boldsymbol{\theta}^{prop})f(\boldsymbol{\theta}^{prop})q(\boldsymbol{\theta}^{cur} \mid \boldsymbol{\theta}^{prop})}{f(\boldsymbol{y} \mid \boldsymbol{\theta}^{cur})f(\boldsymbol{\theta}^{cur})q(\boldsymbol{\theta}^{prop} \mid \boldsymbol{\theta}^{cur})}\right\}$$

3.1. Metropolis–Hastings Algorithm

- Note that for the calculation of α we do not need to know the normalizing constant since

$$a = \min\left\{1, \frac{f(\boldsymbol{y} \mid \boldsymbol{\theta}^{prop}) f(\boldsymbol{\theta}^{prop}) q(\boldsymbol{\theta}^{cur} \mid \boldsymbol{\theta}^{prop})}{f(\boldsymbol{y} \mid \boldsymbol{\theta}^{cur}) f(\boldsymbol{\theta}^{cur}) q(\boldsymbol{\theta}^{prop} \mid \boldsymbol{\theta}^{cur})}\right\}$$

α depends on

- The likelihood
- The prior
- The proposal

3.1. Metropolis–Hastings Algorithm

Random walk Metropolis

- Usual choice for the proposal:

$$q(\boldsymbol{\theta}^{prop} \mid \boldsymbol{\theta}^{cur}) = N(\boldsymbol{\theta}^{cur}, c^2).$$

- We propose a new value $\boldsymbol{\theta}^{prop}$ with mean equal to the current value of the chain and variance controlled by c^2.
- c^2 is also called ***tuning parameter*** since it affects the convergence of the chain and must be tuned appropriately.
- The acceptance probability is simplified to

$$a = \min\left\{1, \frac{f(\boldsymbol{\theta}^{prop} \mid \boldsymbol{y})}{f(\boldsymbol{\theta}^{cur} \mid \boldsymbol{y})}\right\} = \min\left\{1, \frac{f(\boldsymbol{y} \mid \boldsymbol{\theta}^{prop}) f(\boldsymbol{\theta}^{prop})}{f(\boldsymbol{y} \mid \boldsymbol{\theta}^{cur}) f(\boldsymbol{\theta}^{cur})}\right\}$$

due to the symmetry of the proposal

3.1. Metropolis–Hastings Algorithm

Random walk Metropolis

Tuning of c^2

It affects the convergence of the chain and must be tuned appropriately.

- Small values make the chain to move slowly
  => Propose values very close to the current values
  => accept them with high probability
  => High autocorrelations
- Large values make the chain to move less but with bigger moves
  => Propose values away from the current values
  => reject them with high probability
  => The chain may stack to the same set of values for a long time
  => High autocorrelations

3.1. Metropolis–Hastings Algorithm

Random walk Metropolis

Tuning of c^2 – Optimal acceptance

- Roberts et al. (1997), Neal and Roberts (2008)
  - 23% for multidimensional problems
  - 45% for univariate cases
- Any choice of c^2 from 20–40% should be fine

  "there is little to be gained by fine tuning of acceptance rates"

  (Roberts and Rosental, 2001)

3.2. Gibbs Sampling

- If we are in *t* iteration of the algorithm

  => set $\boldsymbol{\theta}^{cur}=\boldsymbol{\theta}^{(t-1)}$ i.e. the current values of $\boldsymbol{\theta}$.

  $\boldsymbol{\theta}^{cur}=(\theta_1^{cur},\theta_2^{cur}, \dots ,\theta_p^{cur})$
- Generate θ_1^{new} from $f(\theta_1|\theta_2^{cur},\dots, \theta_p^{cur},\mathbf{y})$
- Generate θ_2^{new} from $f(\theta_2|\theta_1^{new},\theta_3^{cur},\dots, \theta_p^{cur},\mathbf{y})$
- … … … … … … … … … … … … … … … …
- Generate θ_j^{new} from $f(\theta_j|\theta_1^{new},\dots,\theta_{j-1}^{new},\theta_{j+1}^{cur},\dots, \theta_p^{cur},\mathbf{y})$
- … … … … … … … … … … … … … … … … …
- Generate θ_p^{new} from $f(\theta_p|\theta_1^{new},\dots, \theta_{p-1}^{new},\mathbf{y})$
- *Set* $\boldsymbol{\theta}^{(t)}=\boldsymbol{\theta}^{new}$

3.2. Gibbs Sampling

$f(\theta_j|\theta_1,\dots,\theta_{j-1},\theta_{j+1},\dots, \theta_p,\mathbf{y})$
- is called the full conditional of the posterior distribution
- it is frequently denoted by $f(\theta_j|\bullet)$ or $f(\theta_j|rest)$

3.2. Gibbs Sampling

Differences with Metropolis-Hastings algorithm

- $\boldsymbol{\theta}^{(t-1)} \neq \boldsymbol{\theta}^{(t)}$ – A new set of values is always generated
- The Gibbs sampler is a special case of MH with proposal $q() = f(\theta_j | \bullet)$
- Every time we update one parameter at a time (or a block of parameters)

3.2. Gibbs Sampling

$f(\theta_j | \bullet)$ may be unknown

- Use adaptive rejection sampling for log-convave distributions (Gilks & Wild, 1992)
- For generalized linear models (GLMs), posterior distributions are log-concave (Dellaportas & Smith, 1993)
- This is the main approach used in WinBUGS
- Metropolis steps for the unknown conditionals can be used

3.2. Gibbs Sampling

Advantages

- Simple to implement
- No tuning – automatic

Disadvantages

- Need to calculate conditional posteriors
- Some conditional posteriors may not be available
- No flexibility if high autocorrelations exist

3.2. Gibbs Sampling

Gibbs sampling for a Normal regression model

$$Y_i \sim N(\mu_i, \sigma^2) \text{ for } i=1,2,\dots,n$$
$$\mu_i = \alpha + \beta X_i$$
$$\boldsymbol{\theta}=(\alpha, \beta, \sigma^2)^T$$

- **PRIORS**:

$$f(\boldsymbol{\theta})=f(\alpha, \beta, \sigma^2)=f(\alpha)\, f(\beta)\, f(\sigma^2)$$

- $f(\alpha) \sim Normal(\mu_\alpha, \sigma_\alpha^2)$
- $f(\beta) \sim Normal(\mu_\beta, \sigma_\beta^2)$
- $f(\sigma^2) \sim Inverse\ Gamma(\gamma, \delta)$

  $\Rightarrow f(\tau)=Gamma(\gamma, \delta)$ *for* $\tau=1/\sigma^2$

Gibbs Sampling for normal regression

Full Conditional Posteriors

- $$\alpha \mid \beta, \sigma^2, \mathbf{y} \sim N\left(w_1(\bar{y} - b\bar{x}) + (1-w_1)\mu_\alpha, \, w_1 \frac{\sigma^2}{n} \right)$$

- $$\beta \mid \alpha, \sigma^2, \mathbf{y} \sim N\left(w_2 \frac{\sum_{i=1}^{n} x_i y_i - an\bar{x}}{\sum_{i=1}^{n} x_i^2} + (1-w_2)\mu_\beta, \; w_2 \frac{\sigma^2}{\sum_{i=1}^{n} x_i^2} \right)$$

$$w_1 = \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma^2 / n} \qquad w_2 = \frac{\sigma_\beta^2}{\sigma_\beta^2 + \sigma^2 / \sum_{i=1}^{n} x_i^2}$$

Gibbs Sampling for normal regression

Full Conditional Posteriors

$$\sigma^2 \mid \alpha, \beta, \mathbf{y} \sim \text{Inverse Gamma}\left(\frac{n}{2} + \gamma, \; \frac{1}{2} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2 + \delta \right)$$

4… WinBUGS Language

4.1. Introduction: What is WinBugs?

4.2. A Simple Example

4… WinBUGS Language

4.1. Introduction: What is WinBUGS

- **BUGS**: **B**ayesian inference **U**sing **G**ibbs **S**ampling
- Computing Language for definition of the Model (likelihood, prior)
- Computes the full Conditional Distributions and generates samples from the posterior distribution of interest

4… WinBUGS Language

4.1. Introduction: What is WinBUGS

- **1989**: The BUGS project was initiated by the MRC Biostatistics Unit (Spiegelhalter, Gilks, Best, Thomas).
- **1996**: BUGS version 0.5 for DOS and Unix
- **1997**: BUGS version 0.6 for DOS and Unix
- **1997**: First experimental version of WinBUGS
- **6th August 2007**: Current (and final) version 1.4.3 of WiNBUGS, developed jointly with the Imperial College School of Medicine at St. Mary's, London.

4… WinBUGS Language

4.1. Introduction: What is WinBUGS

- **Now & Future**: Development of OpenBUGS at the *University of Helsinki* in Finland
  [open source experimental version of WinBUGS but unstable in comparison to WiNBUGS]
  http://www.openbugs.info/w/

- **WinBUGS web-site**:
  http://www.mrc-bsu.cam.ac.uk/bugs
  (includes a wide variety of add-in software, utilities, related papers, and course material)

4… WinBUGS Language
4.1. Introduction: What is WinBUGS

Main Characteristics of WinBUGS

- Continuation of Classic BUGS: Similar + new methodological developments
- Development page http://www.winbugs-development.org.uk/
- Graphical representation of model using **DoodleBUGS**
- Menu Driven control of each model session
- Can be called via other packages such as R, Matlab, Excel
- Can be used to estimate parameters of complicated models

4… WinBUGS Language
4.1. Introduction: What is WinBUGS

Installation of WinBUGS

1. Download WinBUGS14.exe to your computer from http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14.exe
2. Double click on WinBUGS14.exe and follow the instructions
3. Go to c:\Program Files\WinBUGS14 directory and create a shortcut of file the file WinBUGS14.exe
4. Double click on WinBUGS14.exe to run WinBUGS.
5. Download the free key from http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/WinBUGS14_immortality_key.txt
6. Open the key from WinBUGS and follow the instructions
7. *After following the instructions given in the key, check that the **Keys.ocf** file in ..\WinBUGS14\Bugs\Code\ has been updated. (Some people have found they need to re-boot the machine to complete installation of the key.)*
8. Download the 1.4.3 upgrade patch http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/patches.shtml
9. Open the patch from WinBUGS and follow the instructions

4… WinBUGS Language
4.1. Introduction: What is WinBUGS

BEWARE:

GIBBS SAMPLING CAN BE DANGEROUS

4… WinBUGS Language
4.1. Introduction: What is WinBUGS

Why it is dangerous for our health (and our academic status)?

- Wrong Results due to lack of convergence
- Slow convergence
- Bad Starting Values
- Bad Construction of Model
- Over-parameterised Model
- Overflow or Underflow problems (resulting in trap messages in WinBUGS).
- The program may be very slow or stack and never finishes.

4… WinBUGS Language
4.1. Introduction: What is WinBUGS

Structure of WinBUGS Model code

- Model code
  - ✓ likelihood specification,
  - ✓ prior,
  - ✓ other parameters of interest
- Data
  - Simple rectangular form and/or
  - List form
- Initial Values
- Script file (optional, needed for background running of the model)

4… WinBUGS Language
4.1. Introduction: What is WinBUGS

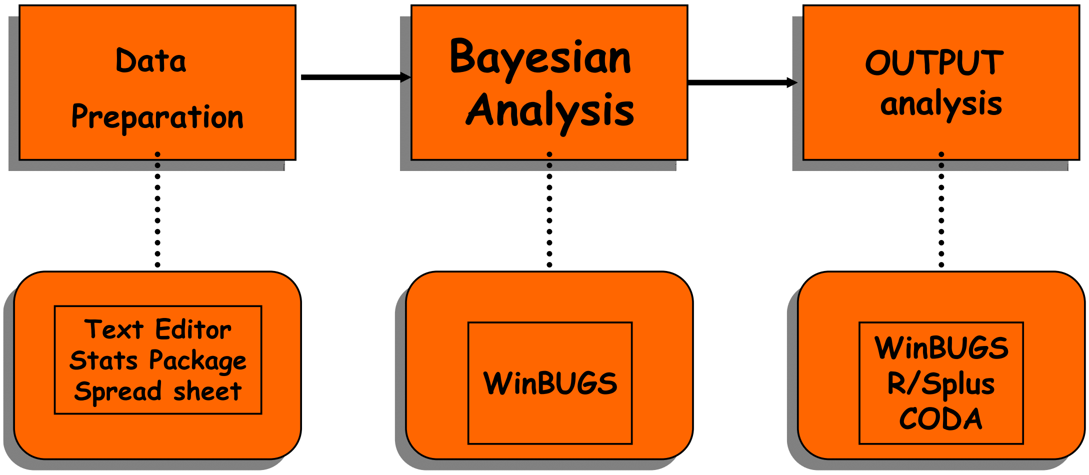
Types of files

- ODC files

  WinBUGS files where we can write the code and save results (incl. graphs)
- Simple text files

4… WinBUGS Language

4.1. Introduction: What is WinBUGS



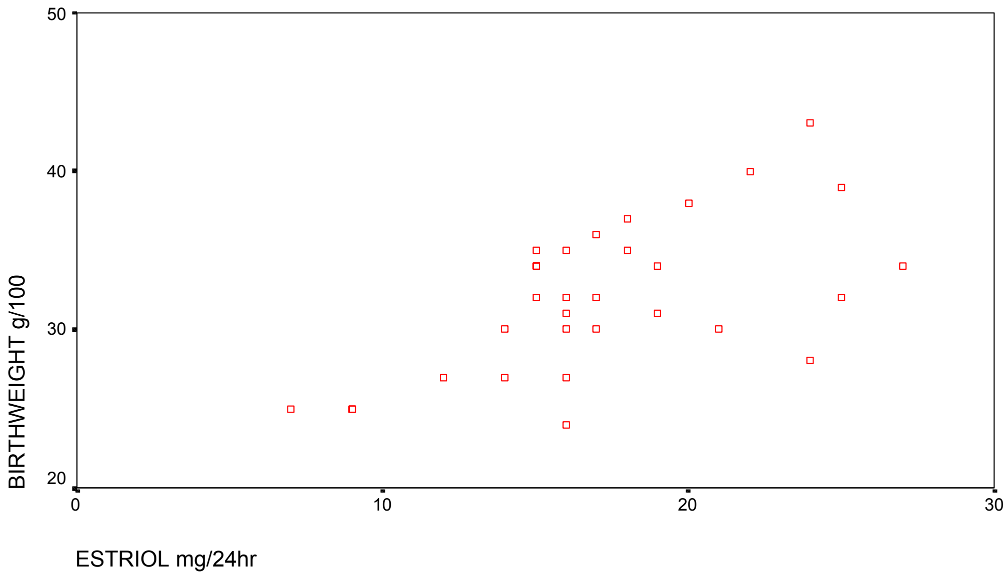
4… WinBUGS Language

4.2. A simple example

- Green & Touchston (1963, *Am.Jour. Of Obsterics & Gynecology*)
- STUDY OF THE RELATIONSHIP
  - ✓ Y : Birthweight
  - ✓ X : Estriol level of women
  - ✓ Sample Size n=31
- The relationship can be examined in the following graph

4… WinBUGS Language

4.2. A simple example



4… WinBUGS Language

4.2.1 Building the Model

- RANDOM COMPONENT: $Birth_i \sim Normal(\mu_i, \sigma^2)$
- SYSTEMATIC COMPONENT: $\eta_i = \alpha + \beta \times Estriol_i$
- LINK FUNCTION: $\mu_i = \eta_i = \alpha + \beta \times Estriol_i$
- for i=1,...,31
- PRIORS (Non-informative)

  $f(\alpha)$=Normal (0, 10^4)

  $f(\beta)$=Normal (0, 10^4)

  $f(\sigma^2)$=Inverse Gamma (10^{-4}, 10^{-4}) or

  $f(\tau=\sigma^{-2})$=Gamma (10^{-4}, 10^{-4})

4… WinBUGS Language
4.2.2 Writing the WinBUGS Model Code

In a file *.ODC we define our model

Distribution commands

- ✓WinBUGS MANUAL p. 56 – 59
- ✓Ntzoufras (2009) p. 90 – 91

Commands for arithmetic functions

- ✓WinBUGS MANUAL p. 13–14
- ✓Ntzoufras (2009) p. 94

4… WinBUGS Language
4.2.2 Writing the WinBUGS Model Code

Structure:

Shell

```
model{
... [model commands] ...
}
```

Main Model

- ✓Likelihood
- ✓Prior distributions
- ✓Additional parameters of interest

4… WinBUGS Language
4.2.2 Writing the WinBUGS Model Code

Model Code

- Starts with the command `model`
- Model code is included in curly brackets `{}`
- `~` : Defines a random variable (*stochastic relationship*) i.e. that a variable "follows" a distribution
  (see manual p. 56-59 and Ntzoufras,2009, p. 90-91 for commands)
- `<-` : Defines an equality (*deterministic relationship*)
  (see manual p. 13-14 and Ntzoufras,2009, p. 94 for commands)

Distributions in WinBUGS (I)

Discrete Univariate
[top | home]

Bernoulli

`r ~ dbern(p)`

$$p^r(1-p)^{1-r}; \quad r=0,1$$

Binomial

`r ~ dbin(p, n)`

$$\frac{n!}{r!(n-r)!}p^r(1-p)^{n-r}; \quad r=0,\ldots,n$$

Categorical

`r ~ dcat(p[])`

$$p[r]; \quad r=1,2,\ldots,\dim(p); \quad \sum_i p[i]=1$$

Negative Binomial

`x ~ dnegbin(p, r)`

$$\frac{(x+r-1)!}{x!(r-1)!}p^r(1-p)^x; \quad x=0,1,2,\ldots$$

Poisson

`r ~ dpois(lambda)`

$$e^{-\lambda}\frac{\lambda^r}{r!}; \quad r=0,1,\ldots$$

Distributions in WinBUGS (II)

Continuous Univariate [top | home]

Beta

p ~ dbeta(a, b)

$$p^{a-1}(1-p)^{b-1}\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}; \quad 0 < p < 1$$

Chi-squared

x ~ dchisqr(k)

$$\frac{2^{-k/2}x^{k/2-1}e^{-x/2}}{\Gamma(\frac{k}{2})}; \quad x > 0$$

Double Exponential

x ~ ddexp(mu, tau)

$$\frac{\tau}{2}\exp(-\tau|x-\mu|); \quad -\infty < x < \infty$$

Exponential

x ~ dexp(lambda)

$$\lambda e^{-\lambda x}; \quad x > 0$$

Gamma

x ~ dgamma(r, mu)

$$\frac{\mu^r x^{r-1}e^{-\mu x}}{\Gamma(r)}; \quad x > 0$$

Generalized Gamma

x ~ gen.gamma(r, mu, beta)

$$\frac{\beta}{\Gamma(r)}\mu^{\beta r}x^{\beta r-1}\exp\left[-(\mu x)^{\beta}\right]; \quad x > 0$$

Distributions in WinBUGS (III)

Log-normal

x ~ dlnorm(mu, tau)

$$\sqrt{\frac{\tau}{2\pi}}\frac{1}{x}\exp\left(-\frac{\tau}{2}(\log x-\mu)^2\right); \quad x > 0$$

Logistic

x ~ dlogis(mu, tau)

$$\frac{\tau\exp(\tau(x-\mu))}{(1+\exp(\tau(x-\mu)))^2}; \quad -\infty < x < \infty$$

Normal

x ~ dnorm(mu, tau)

$$\sqrt{\frac{\tau}{2\pi}}\exp\left(-\frac{\tau}{2}(x-\mu)^2\right); \quad -\infty < x < \infty$$

Pareto

x ~ dpar(alpha, c)

$$\alpha c^{\alpha}x^{-(\alpha+1)}; \quad x > c$$

Student-t

x ~ dt(mu, tau, k)

$$\frac{\Gamma\left(\frac{k+1}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}\sqrt{\frac{\tau}{k\pi}}\left[1+\frac{\tau}{k}(x-\mu)^2\right]^{-(k+1)/2};$$

$$-\infty < x < \infty; \quad k \geq 2$$

Uniform

x ~ dunif(a, b)

$$\frac{1}{b-a}; \quad a < x < b$$

Weibull

x ~ dweib(v, lambda)

$$v\lambda x^{v-1}\exp(-\lambda x^{v}); \quad x > 0$$

Distributions in WinBUGS (IV)

Discrete Multivariate [top | home]

Multinomial

`x[] ~ dmulti(p[], N)`

$$\frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_i^{x_i};$$

$$\sum_i x_i = N; \quad 0 < p_i < 1; \quad \sum_i p_i = 1$$

Continuous Multivariate [top | home]

Dirichlet

`p[] ~ ddirch(alpha[])`

$$\frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i p_i^{\alpha_i - 1};$$

$$0 < p_i < 1; \quad \sum_i p_i = 1$$

Multivariate Normal

`x[] ~ dmnorm(mu[], T[,])`

$$(2\pi)^{-d/2} |T|^{1/2} \exp\left(-\tfrac{1}{2}(x-\mu)' T (x-\mu)\right);$$

$$-\infty < x < \infty$$

Multivariate Student-t

`x[] ~ dmt(mu[], T[,], k)`

$$\frac{\Gamma((k+d)/2)}{\Gamma(k/2)\, k^{d/2} \pi^{d/2}} |T|^{1/2}$$

$$\times \left[1 + \tfrac{1}{k}(x-\mu)' T (x-\mu)\right]^{-(k+d)/2};$$

$$-\infty < x < \infty; \quad k \geq 2$$

Wishart

`x[,] ~ dwish(R[,], k)`

$$|R|^{k/2} |x|^{(k-p-1)/2} \exp\left(-\tfrac{1}{2} \mathrm{Tr}(Rx)\right);$$

x symmetric & positive definite

4… WinBUGS Language

4.2.2 Writing the WinBUGS Model Code

IN OUR EXAMPLE

MAIN MODEL

Normal Distribution: `y~dnorm( mu, tau )`

mu= mean

tau= precision = $1/\sigma^2$

Gamma Distribution : `y~dgamma( a, b )`

mean= a/b

`x[i]` : i element of vector x

`d[i,j]` : element of i row and j column of table d

4… WinBUGS Language

4.2.2 Writing the WinBUGS Model Code

(1) $Birth_i \sim Normal(\mu_i, \sigma^2)$

(2) $\eta_i = \alpha + \beta \times Estriol_i$

(3) $\mu_i = \eta_i = \alpha + \beta \times Estriol_i$

for i=1,...,31

```
for (i in 1:n) {
   Birth[i]~dnorm(mu[i],tau)

   mu[i] <-a+b*estriol[i]
}
```

PRIORS

$f(\alpha)=Normal\ (\ 0,\ 10^4\ )$ `a~dnorm(0.0,1.0E-04)`

$f(\beta)=Normal\ (\ 0,\ 10^4\ )$ `b~dnorm(0.0,1.0E-04)`

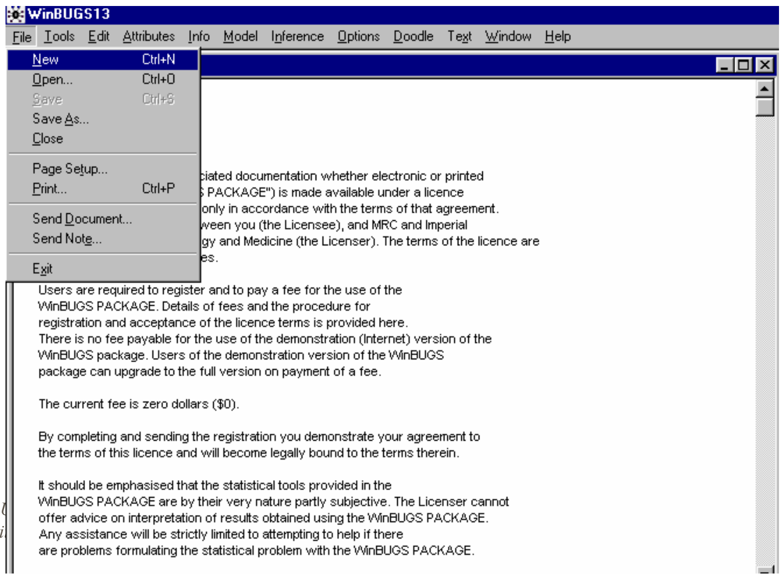
$f(\tau)=Gamma(10^{-4},10^{-4})$ `tau~dgamma(1.0E-04,1.0E-04)`

$\sigma^2=1/\ \tau$ `s2<-1/tau`

4… WinBUGS Language

4.2.2 Writing the WinBUGS Model Code

- Start WinBUGS
- Select "New" from file bar



4… WinBUGS Language

4.2.2 Writing the WinBUGS Model Code

Write the Model Commands

WinBUGS13 - [untitled1]

File Tools Edit Attributes Info Model Inference Options Doodle Text Window Help

```
model estriol;
{
#       definition of likelihood function
#
        for (i in 1:n) {
                birth[i]~dnorm( mu[i], tau ); # random component
                mu[i]<-a.star+b*(estriol[i]-mean(estriol[]));        # systematic component
                                        #       & link function
        }
#       prior distributions
#
        a.star~dnorm( 0, 1.0E-04 ); # normal prior for a
        b~dnorm( 0, 1.0E-04 ); # normal prior for b
        tau~dgamma( 1.0E-04 , 1.0E-04 ); # gamma prior for precision
        s2<-1/tau;
        a<-a.star-b*mean(estriol[]);
}
```


4… WinBUGS Language

4.2.2 Writing the WinBUGS Model Code

- Write the **Model Commands**
- Followed by the **initial values**

```
list(a.star=0.0, b=0.0, tau=1.0)
```

- and the data

```
list(n=31)
estriol[] birth[]
7 25
9 25
… …
25    39
24    43
END
```

← **Empty line**

4… WinBUGS Language

4.2.2 Writing the WinBUGS Model Code

Alternative way of defining data (using list format)

```
list(n=31,
  estriol=c(7, 9, 9, 12, 14, 16, 16,
  14, 16, 16, 17, 19, 21, 24, 15, 16,
  17, 25, 27, 15, 15, 15, 16, 19, 18,
  17, 18, 20, 22, 25, 24),
  birth = c(25, 25, 25, 27, 27, 27, 24,
  30, 30, 31, 30, 31, 30, 28, 32, 32,
  32, 32, 34, 34, 34, 35, 35, 34, 35,
  36, 37, 38, 40, 39, 43)
)
```

5… Running a model in WinBUGS

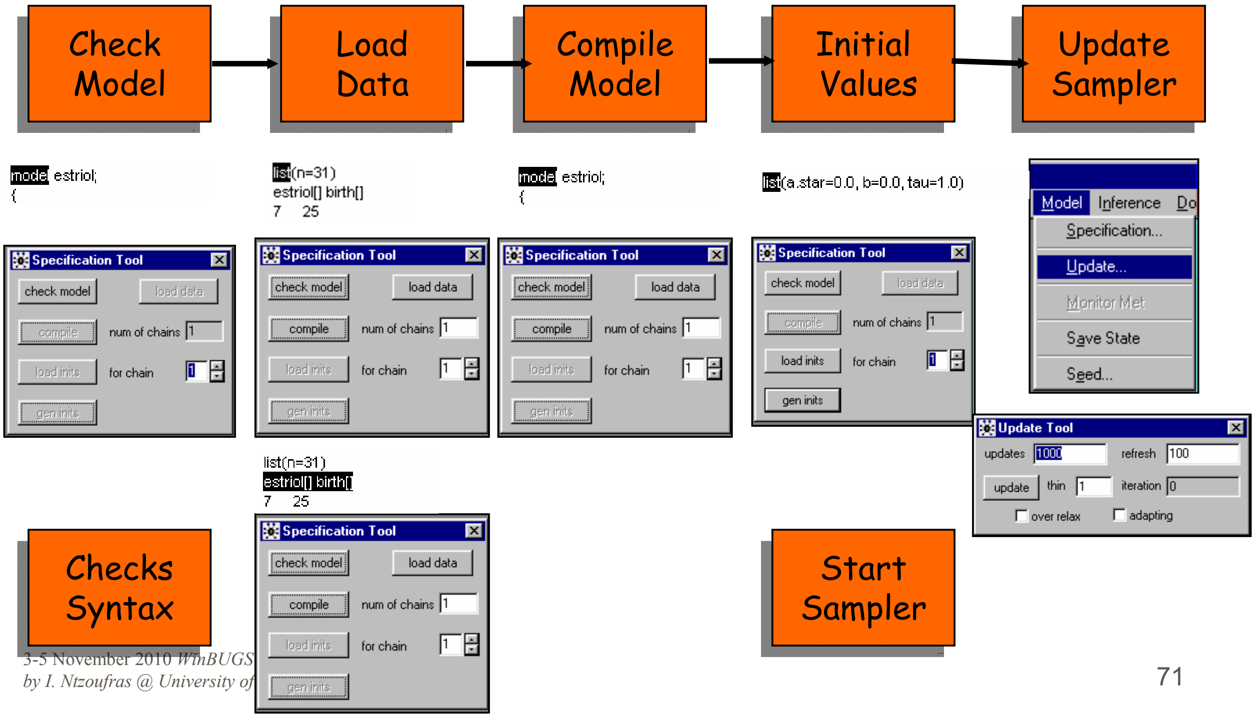
5.1. Generating values from the posterior

5.2. Analysis of the MCMC output

5.3. Running the model in the Background Using Scripts

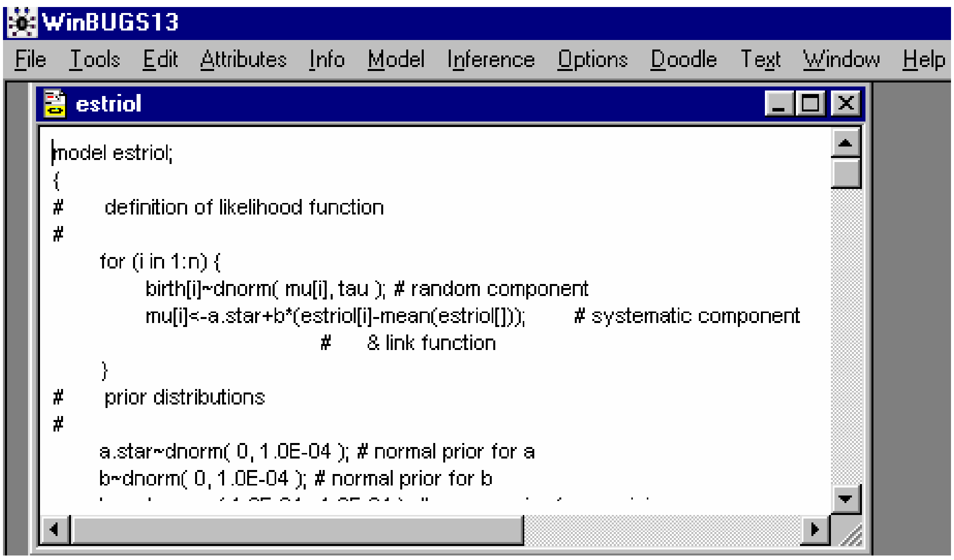
5.4. Deviance Information Criterion (DIC) in WiNBUGS

5... Running a model in WinBUGS
5.1. Generating values from the posterior



5... Running a model in WinBUGS
5.1. Generating values from the posterior

1... Check Model

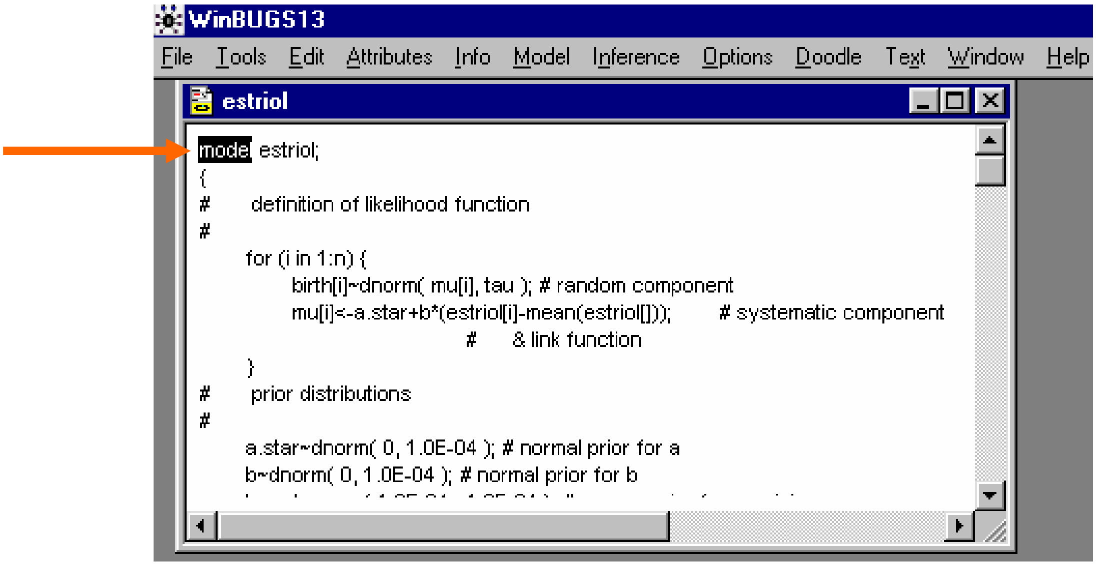


5... Running a model in WinBUGS

5.1. Generating values from the posterior

1... Check Model

HIGHLIGHT THE MODEL COMMAND

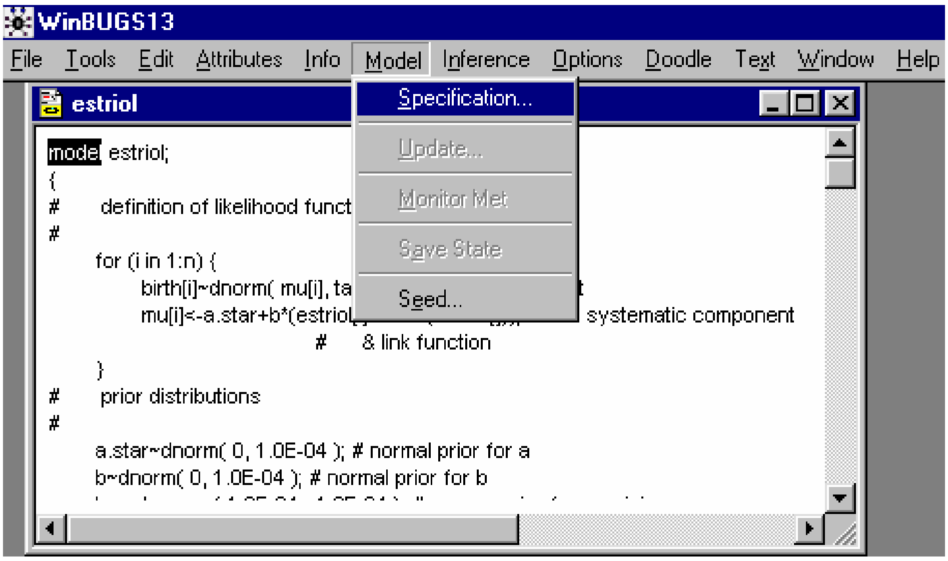


5... Running a model in WinBUGS

5.1. Generating values from the posterior

1... Check Model

SELECT "SPECIFICATION" FROM "MODEL" MENU

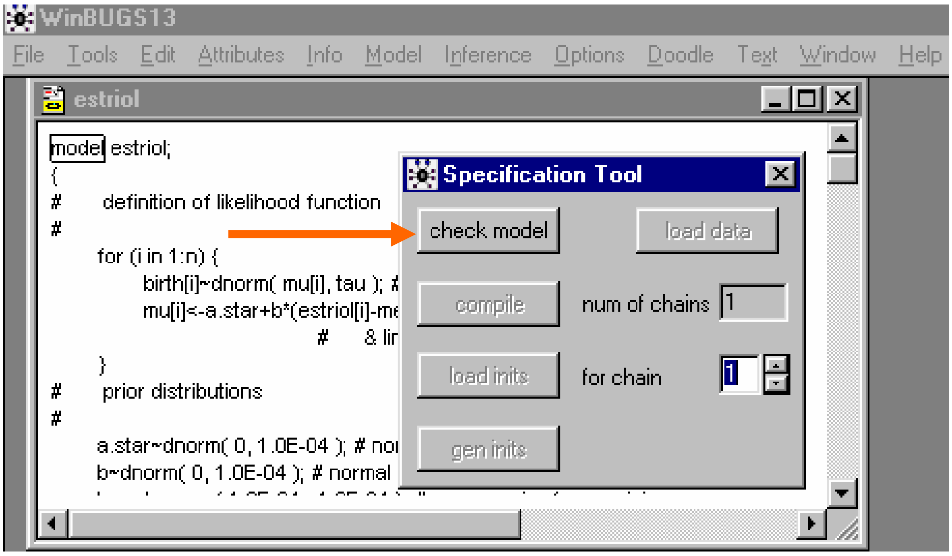


5... Running a model in WinBUGS

5.1. Generating values from the posterior

1... Check Model

SELECT "CHECK MODEL"

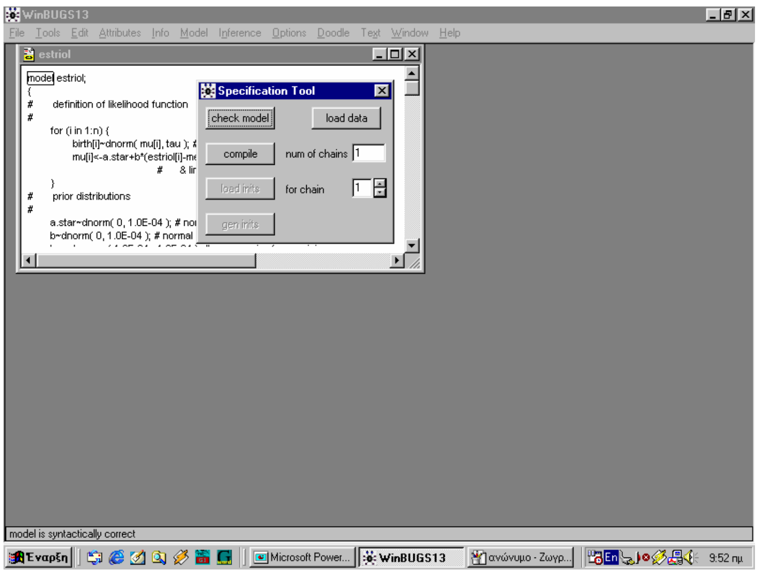


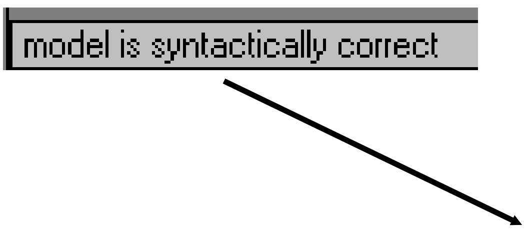
5... Running a model in WinBUGS

5.1. Generating values from the posterior

1... Check Model

IF SYNTAX IS CORRECT THEN



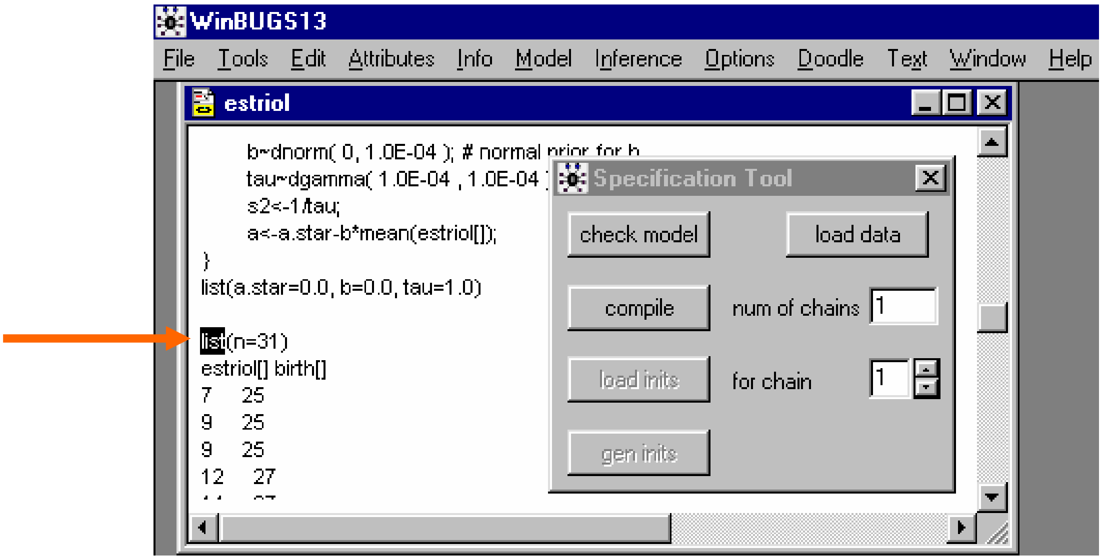


5... Running a model in WinBUGS

5.1. Generating values from the posterior

2... Load Data

HIGHLIGHT "LIST" OR THE 1ST LINE OF THE DATA

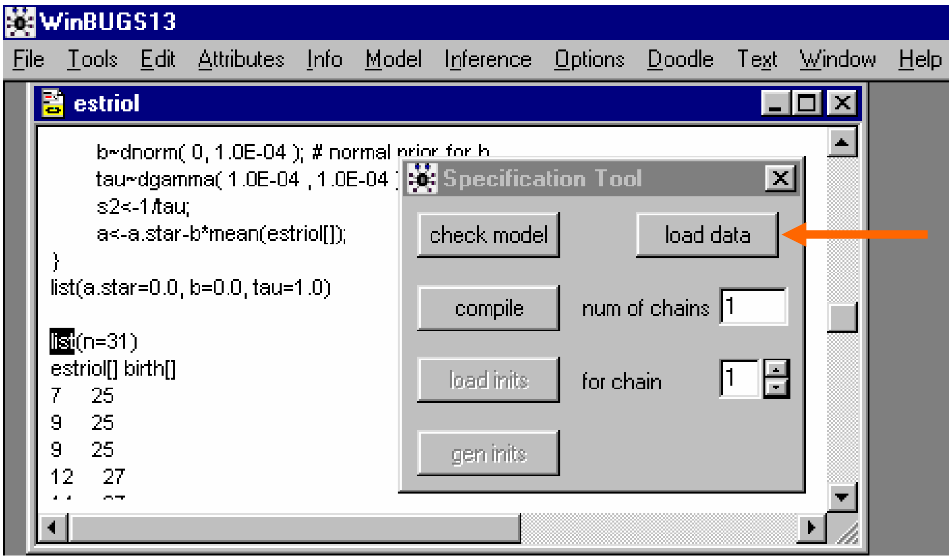


5... Running a model in WinBUGS

5.1. Generating values from the posterior

2... Load Data

CLICK THE BOX "LOAD DATA"

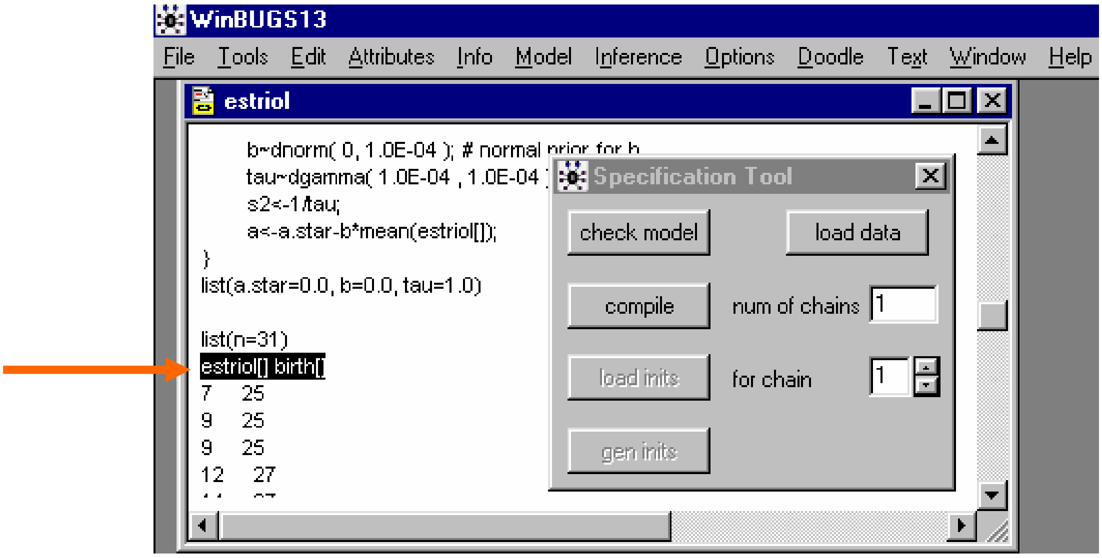


5... Running a model in WinBUGS

5.1. Generating values from the posterior

2... Load Data

HIGHLIGHT ADDITIONAL DATA

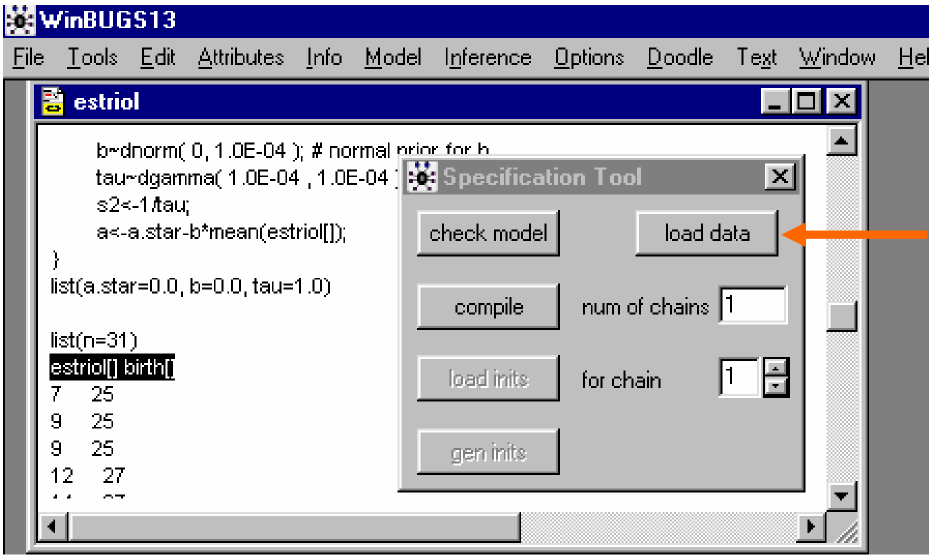


5... Running a model in WinBUGS

5.1. Generating values from the posterior

2... Load Data

CLICK (AGAIN) THE BOX "LOAD DATA"

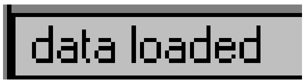


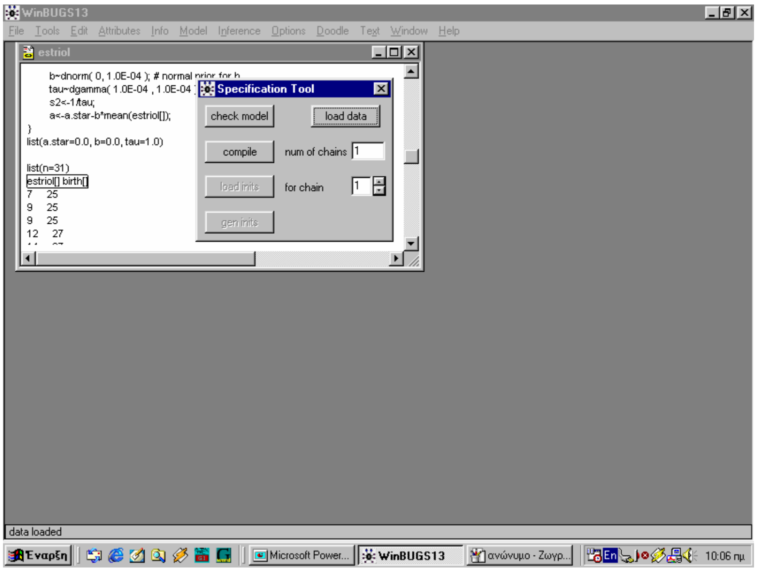
5... Running a model in WinBUGS

5.1. Generating values from the posterior

2... Load Data

IF DATA ARE LOADED THEN



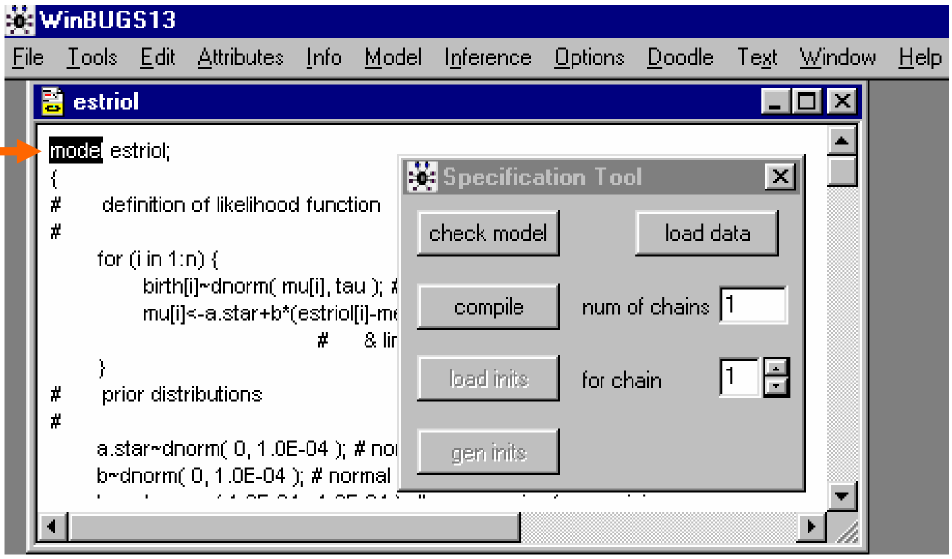


5... Running a model in WinBUGS

5.1. Generating values from the posterior

3... Compile Model

HIGHLIGHT "MODEL" COMMAND (AGAIN)

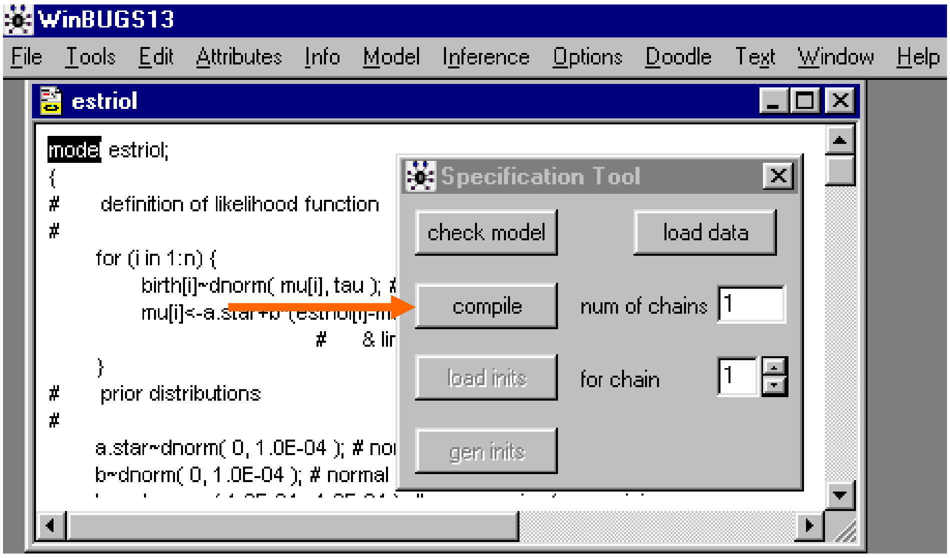


5... Running a model in WinBUGS

5.1. Generating values from the posterior

3... Compile Model

CLICK ON BOX "COMPILE"

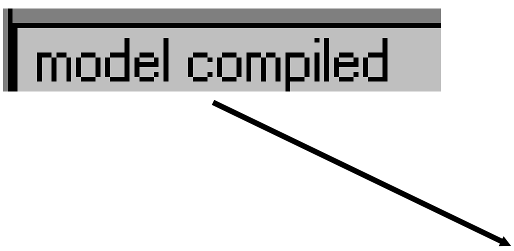


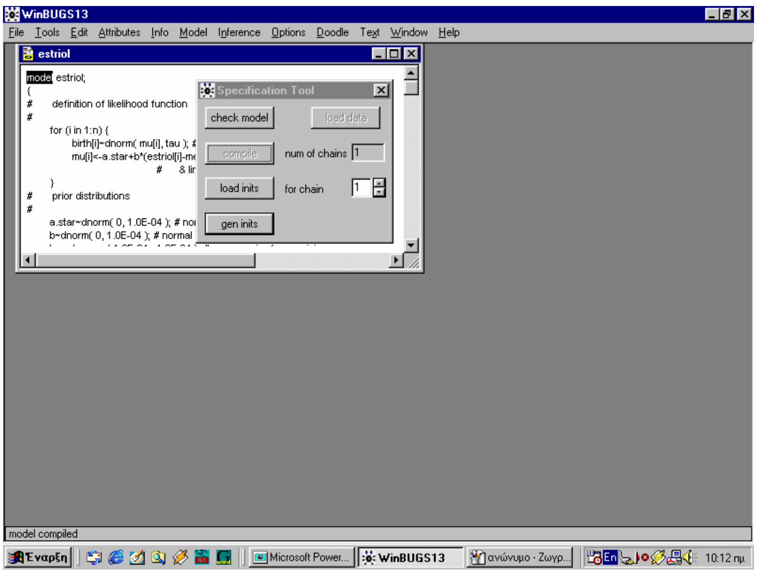
5... Running a model in WinBUGS

5.1. Generating values from the posterior

3... Compile Model

IF MODEL COMPILED THEN



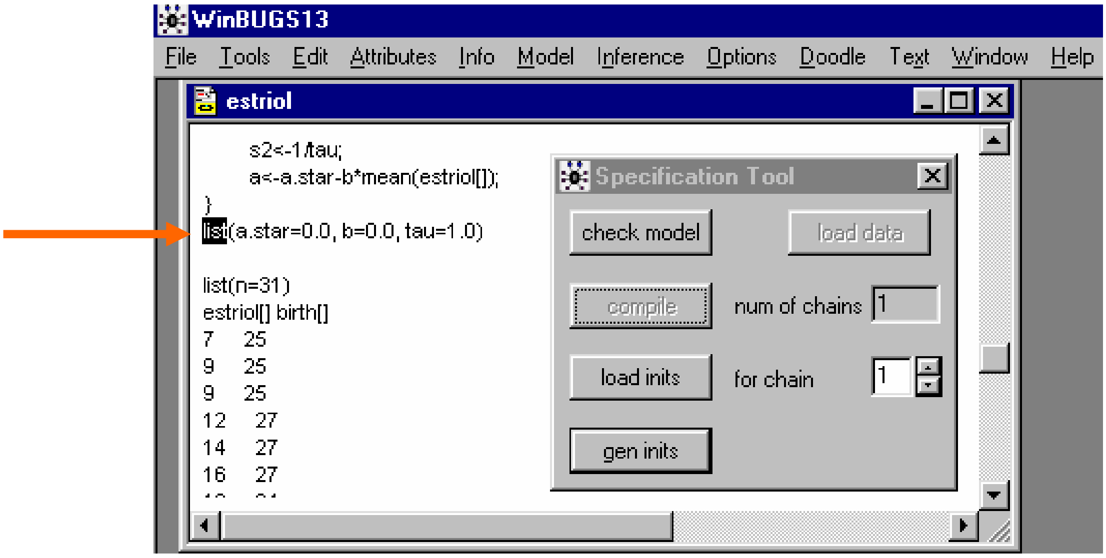


5... **Running a model in WinBUGS**

5.1. Generating values from the posterior

4... Load or Generate Initial Values

HIGHLIGHT "LIST" OF INITIAL VALUES

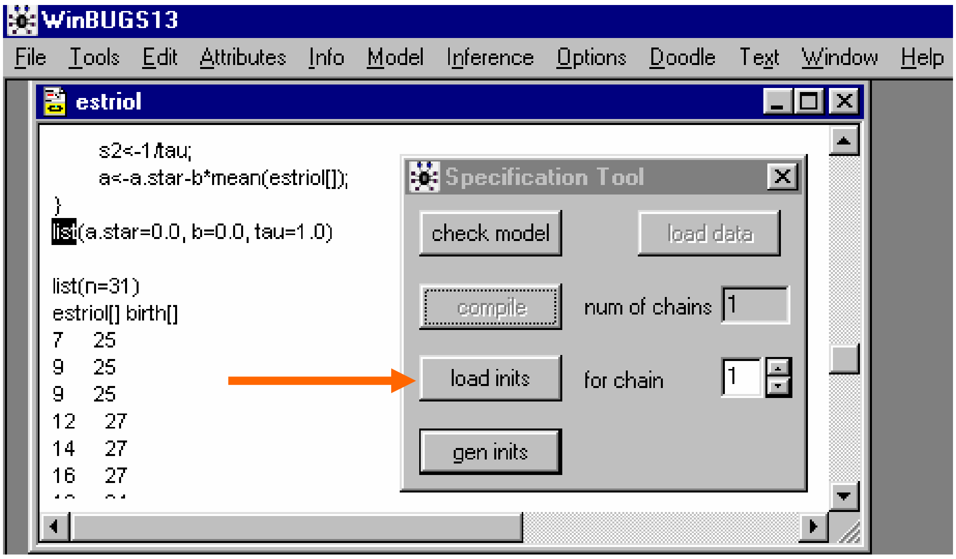


5... **Running a model in WinBUGS**

5.1. Generating values from the posterior

4... Load or Generate Initial Values

CLICK ON BOX "LOAD INITS"



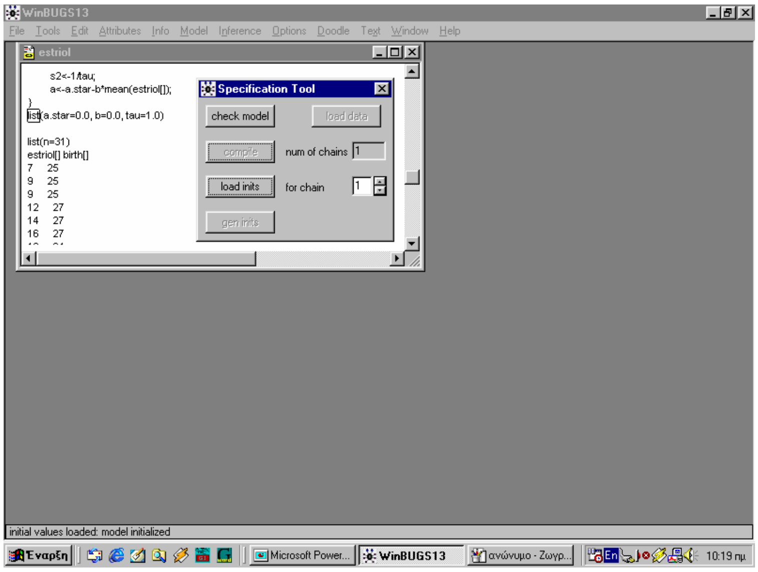
5... Running a model in WinBUGS

5.1. Generating values from the posterior

4... Load or Generate Initial Values

IF INITIAL VALUES ARE LOADED THEN

initial values loaded: model initialized



3-5 November 2010 *WinBUGS Erasmus Tutorial*
by I. Ntzoufras @ University of Pavia

5... Running a model in WinBUGS

5.1. Generating values from the posterior

4... Load or Generate Initial Values

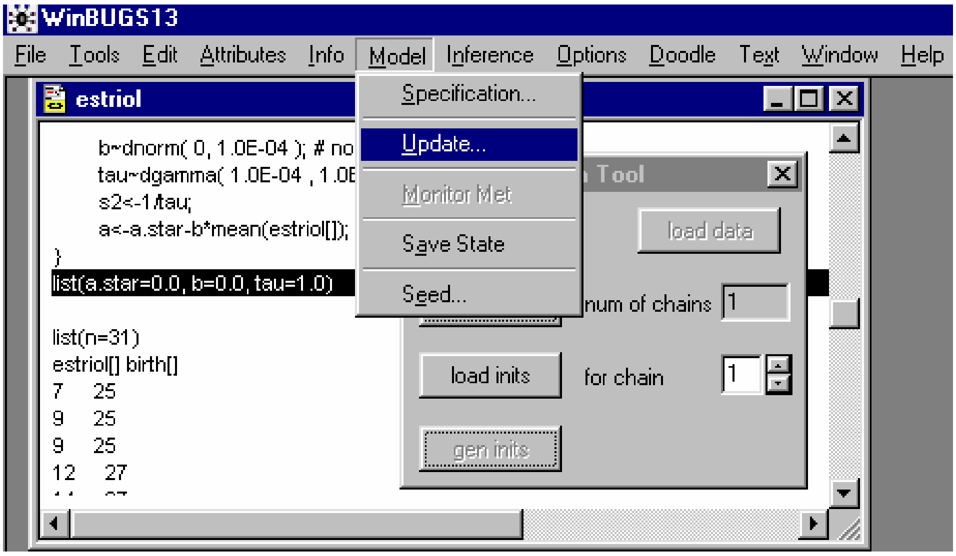
NOW WINBUGS IS READY TO GENERATE SAMPLES USING GIBBS SAMPLING

5... **Running a model in WinBUGS**

5.1. Generating values from the posterior

5... GENERATING BURN-IN VALUES

SELECT "UPDATE" IN "MODEL" MENU

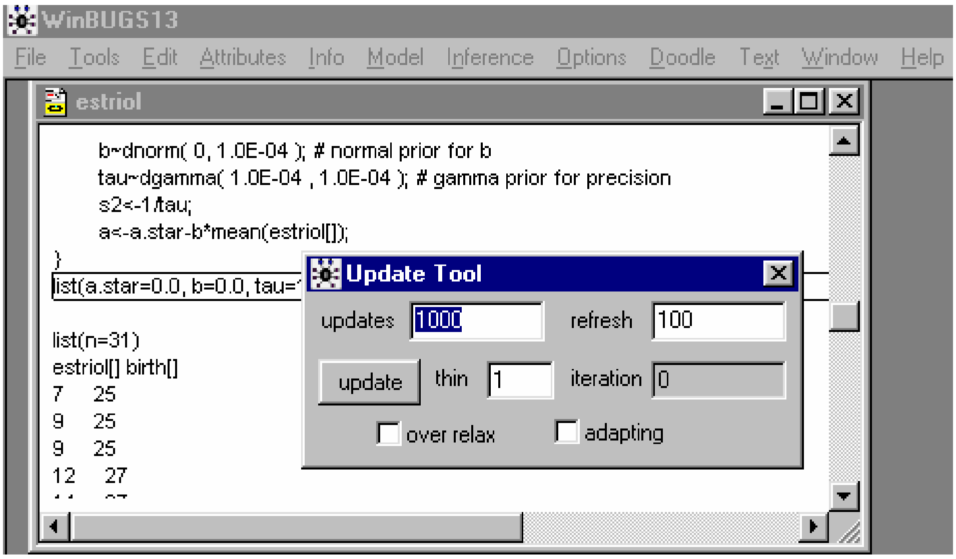


5... **Running a model in WinBUGS**

5.1. Generating values from the posterior

5... GENERATING BURN-IN VALUES

GIVE THE NUMBER OF BURN-IN ITERARATIONS

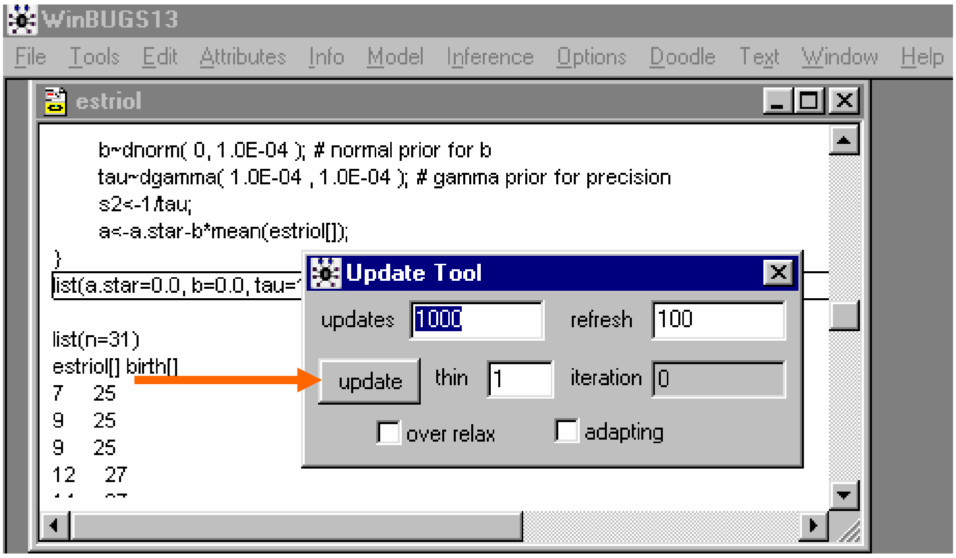


5... **Running a model in WinBUGS**

5.1. Generating values from the posterior

5... GENERATING BURN-IN VALUES

CLICK ON UPDATE TO GENERATE VALUES

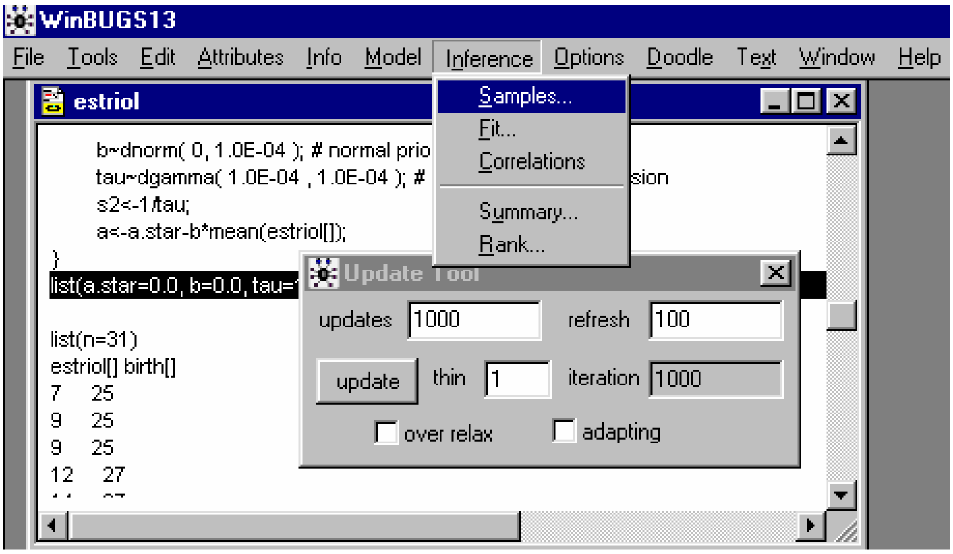


5... **Running a model in WinBUGS**

5.1. Generating values from the posterior

6... MONITORING PARAMETERS

SELECT "SAMPLES" IN "INFERENCE" MENU

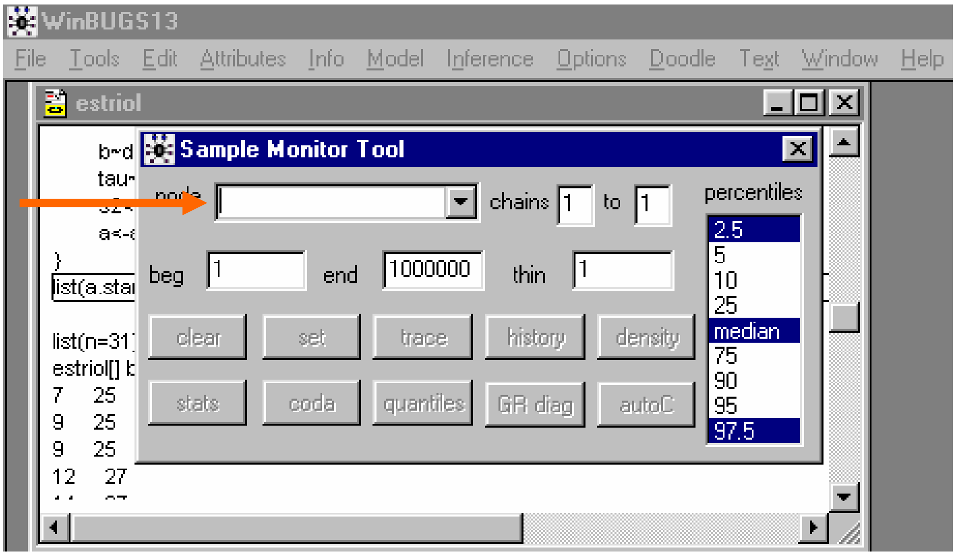


5... Running a model in WinBUGS

5.1. Generating values from the posterior

6... MONITORING PARAMETERS

WRITE NAME OF MONITORED PARAMETER

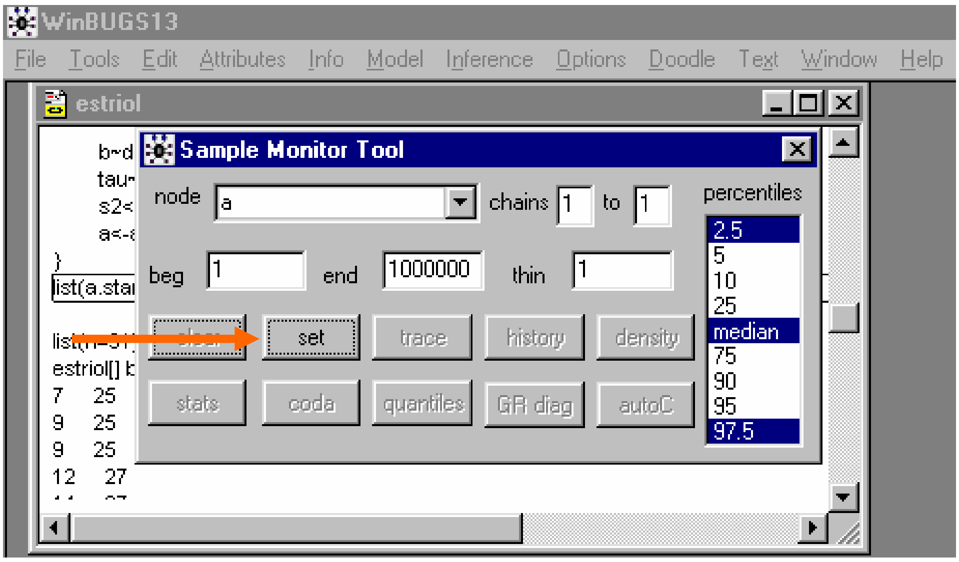


5... Running a model in WinBUGS

5.1. Generating values from the posterior

6... MONITORING PARAMETERS

CLICK ON "SET" BOX



5... **Running a model in WinBUGS**

5.1. Generating values from the posterior

6... MONITORING PARAMETERS

- WRITE “a” AND CLICK “SET” BOX
- WRITE “b” AND CLICK “SET” BOX
- WRITE “s2” AND CLICK “SET” BOX

5... **Running a model in WinBUGS**

5.1. Generating values from the posterior

7... GENERATING POSTERIOR VALUES

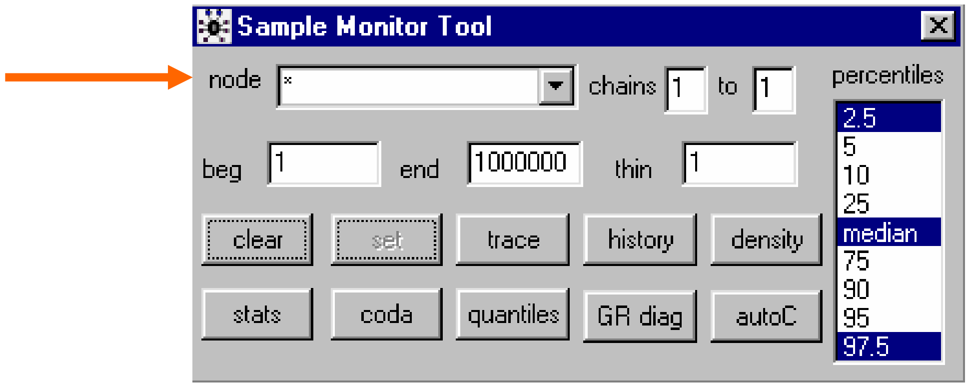
SELECT “UPDATE TOOL” UPDATE 1000 ITERATIONS

5... Running a model in WinBUGS

5.2. Analysis of the MCMC output

8a... Obtaining posterior summaries

- **SELECT "SAMPLE MONITOR TOOL" AND WRITE NAME OF DESIRED PARAMETER [*=ALL MONITORED PARAMETERS]**

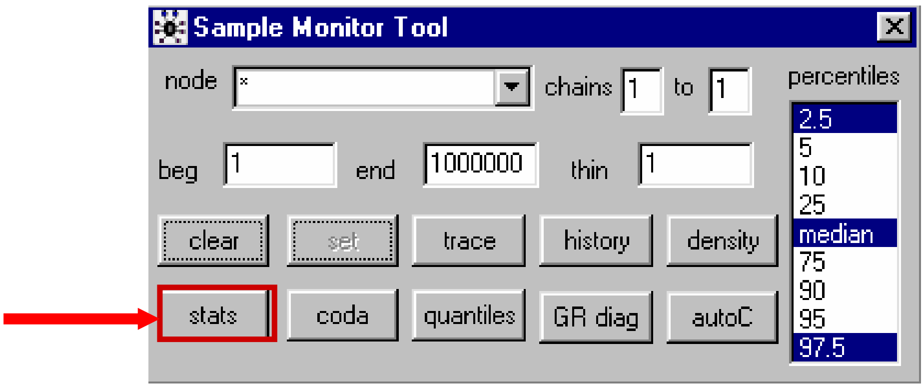


5... Running a model in WinBUGS

5.2. Analysis of the MCMC output

8a... Obtaining posterior summaries

- **CLICK ON "STATS" BOX**



5... Running a model in WinBUGS

5.2. Analysis of the MCMC output

8a... Obtaining posterior summaries

- **A Table with the posterior summaries will appear**

Node statistics

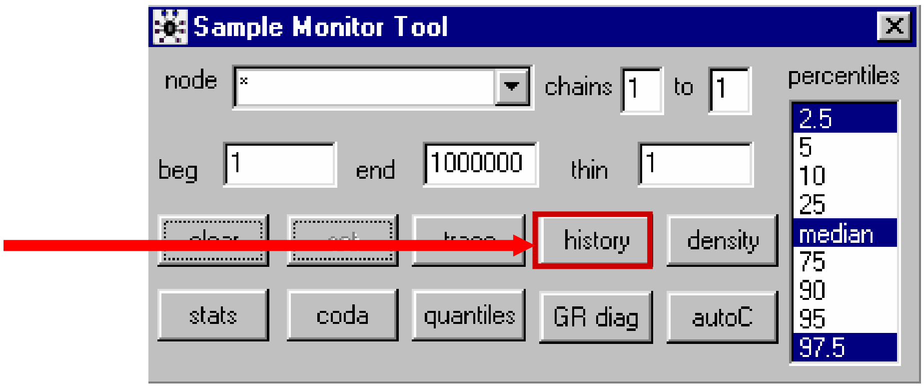
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
a	21.44	2.696	0.08362	16.06	21.48	26.68	1001	1000
b	0.6135	0.1481	0.004041	0.3255	0.6101	0.8979	1001	1000
s2	15.62	4.499	0.1344	9.01	14.9	25.5	1001	1000

5... Running a model in WinBUGS

5.2. Analysis of the MCMC output

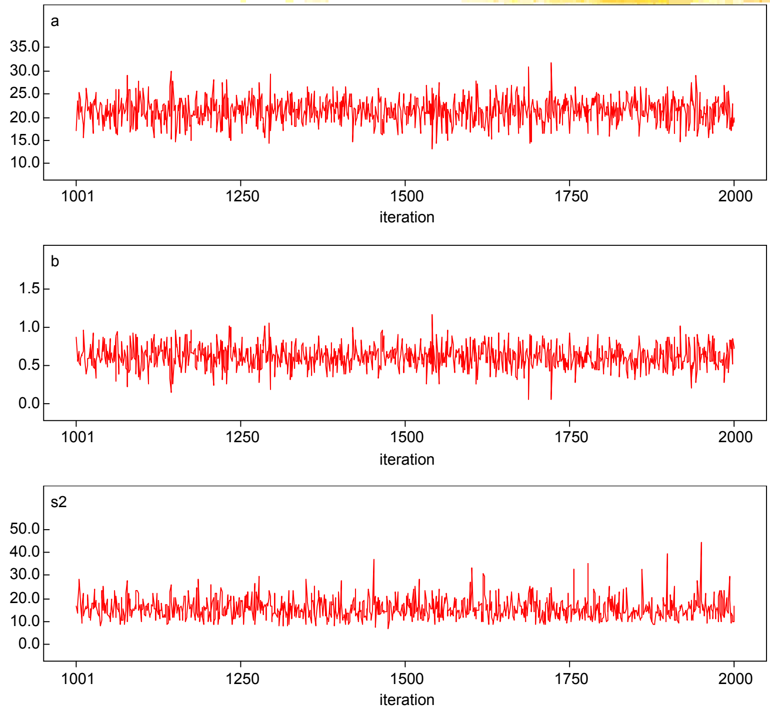
8b... Obtaining trace plots for all generated values

- **CLICK ON "HISTORY" BOX PRODUCES TRACE PLOTS**



5... Running a model in WinBUGS

5.2. Analysis of the MCMC output



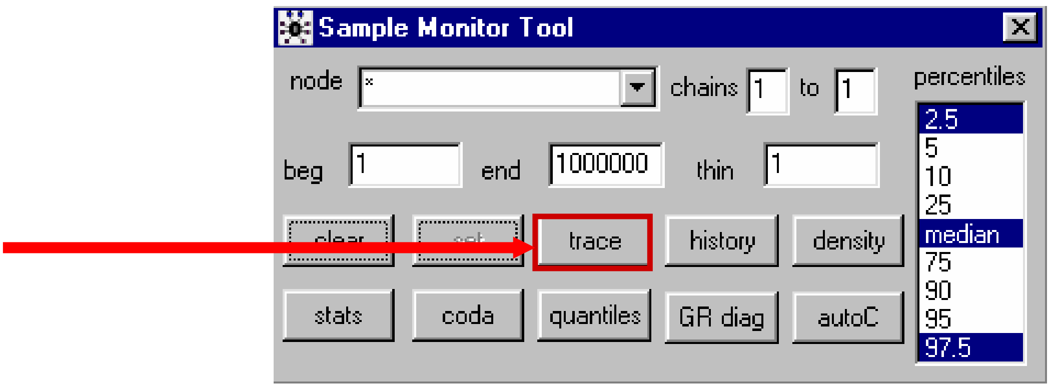
- Trace plots will appear
- All values must be stabilised within a zone of values.
- Ups and downs indicate autocorrelation or non-convergence

5... Running a model in WinBUGS

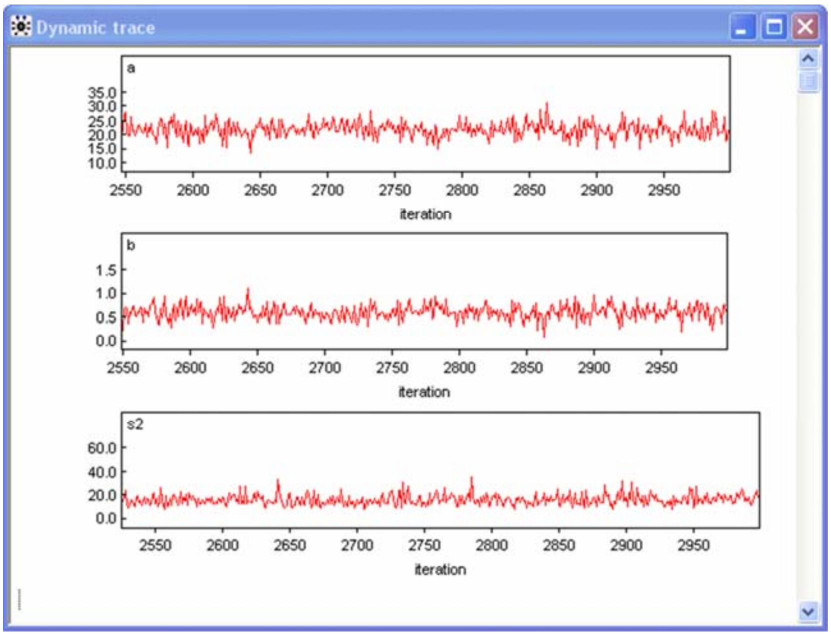
5.2. Analysis of the MCMC output

8b... Obtaining online trace plots

- **CLICK ON "TRACE" BOX PRODUCES TRACE PLOTS**
- *This will produce animated trace plots updated online*



5... **Running a model in WinBUGS**
5.2. Analysis of the MCMC output

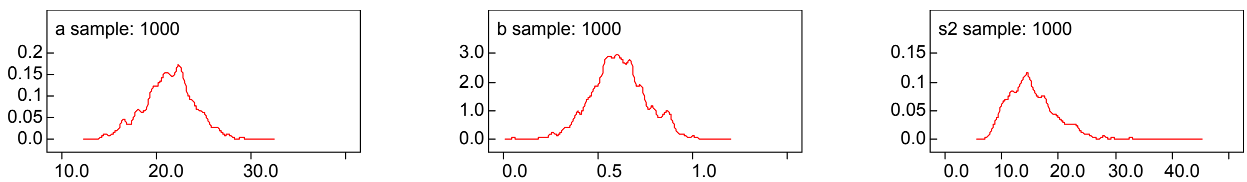


- Online updated trace plots will appear
- This is for 1000 iterations updated every 100

5... **Running a model in WinBUGS**
5.2. Analysis of the MCMC output

8... SUMMARIZING THE POSTERIOR

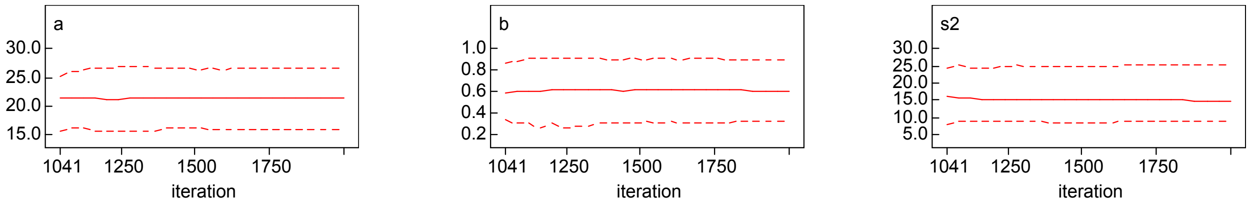
CLICK ON "DENSITY" BOX PRODUCES PLOTS OF THE ESTIMATED DENSITY FOR THE POSTERIOR DISTRIBUTION



5... Running a model in WinBUGS
5.2. Analysis of the MCMC output

8... SUMMARIZING THE POSTERIOR

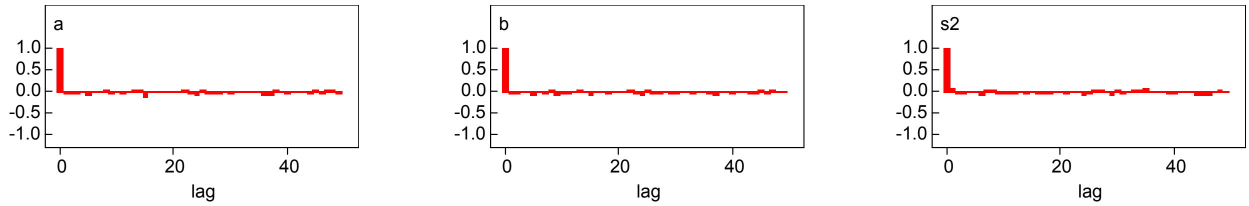
CLICK ON "QUANTILES" BOX PRODUCES THE FOLLOWING GRAPHS



5... Running a model in WinBUGS
5.2. Analysis of the MCMC output

8... SUMMARIZING THE POSTERIOR

CLICK ON "AUTOC" BOX PRODUCES AUTO-CORRELATIONS PLOTS



5... Running a model in WinBUGS

5.2. Analysis of the MCMC output

8... SUMMARIZING THE POSTERIOR

- "CODA" BOX PRODUCES LIST OF DATA IN FORM THAT CAN BE LOADED BY CODA SOFTWARE
- "GR DIAG" BOX PRODUCES CONVERGENCE DIAGNOSTIC (DEMANDS MULTIPLE CHAINS)

5... Running a model in WinBUGS

5.3. Running the model in the Background Using Scripts

5.3.1. Introduction

BATCH MODE METHOD: SCRIPTING

only available in WINBUGS 1.4 and later versions

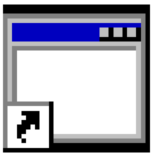
- Alternative way to generate random variables without "clicking" around and having to wait for the results to ask for specific output analysis
- We need at least 4 files in WinBUGS (odc or txt format)
  1. Script code (with commands for generation and output analysis)
  2. Model code
  3. Data files (it can be more than one)
  4. Initial values files (1 for each chain)

5... Running a model in WinBUGS

5.3. Running the model in the Background Using Scripts

Example `script.odc`

Click on



BackBugs14.lnk

Within the WINBUGS14 directory

Usually in `c:\Program Files\Winbugs14\`

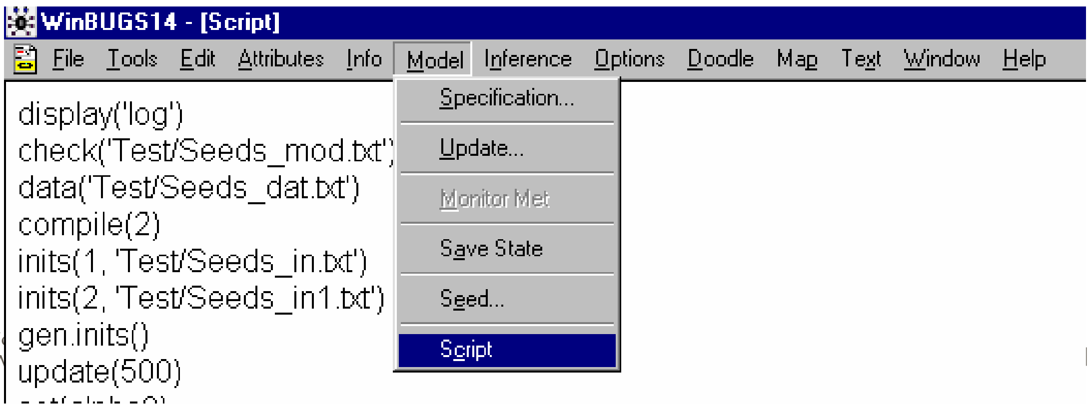
5... Running a model in WinBUGS

5.3. Running the model in the Background Using Scripts

Open the script file

`c:\Program Files\Winbugs14\script.odc`

We can run this script (and the corresponding model) in the background by selecting **MODEL>SCRIPT**



5... Running a model in WinBUGS

5.3. Running the model in the Background Using Scripts

5.3.2. Some script commands

- `display('log')`: Opens a log file where it stores all results
- `check('Test/Seeds_mod.txt')`: Check the syntax of the model code in file `Seeds_mod.txt` placed in the sub-directory `Test`.
- `data('Test/Seeds_dat.txt')`: Loads the data from file `Seeds_dat.txt` placed in the sub-directory `Test`.
- `compile(2)` : Compilation and initialization of 2 chains.
- `inits(1, 'Test/Seeds_in.txt')`: Loading the initial values of the first chain from file `Seeds_in.txt` placed in the sub-directory `Test`.

5... Running a model in WinBUGS

5.3. Running the model in the Background Using Scripts

5.3.2. Some script commands (2)

- `gen.inits()`: Generation of initial values
- `update(500)`: Generation of 500 values (iterations) (Burn-in period).
- `set(alpha0)`: We start monitoring (i.e. saving) the generated values for parameter/node `alpha0`.
- `update(1000)`: Generation of additional 1000 values/iteratons.
- `stats(*)`: Descriptive statistics from the simulated sample of all monitored parameters
- `history(*)`: Trace plots for all monitored values
- `trace(*)`: Dynamic (on-line) Trace plot for all monitored values

5... Running a model in WinBUGS

5.3. Running the model in the Background Using Scripts

5.3.2. Some script commands (3)

- `density(*)`: Kernel density plot of the posterior distribution of all monitored parameters
- `autoC(*)`: Autocorrelation plots of all monitored parameters
- `quantiles(*)`: Quantiles plots of all monitored parameters
- `coda(*,output)`: Saving all values of the monitored parameters in the file **output** in a CODA format. If the namefile is empty then two WiNBUGS windows are opened with the corresponding files.
- `save('seedsLog')`: Saves all results of the log window in file **seedLog.odc** (WINBUGS format including diagrams). If the file has **txt** suffix then all results are saved in text format file without any figures and graphs.
- `quit()`: Exits WinBUGS

5... Running a model in WinBUGS

5.3. Running the model in the Background Using Scripts

5.3.2. Background running of Estriol example

Step 1: Create 5 files

1. script.odc
2. model.odc
3. data.odc
4. data2.odc
5. inits.odc

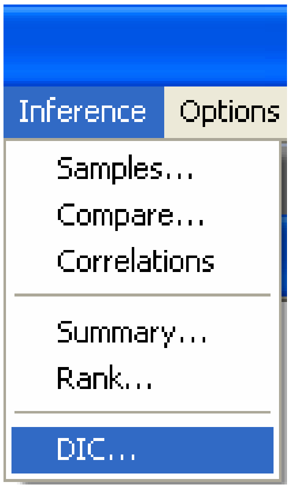
Step 2: Open SCRIPT.ODC and run it from MODEL>SCRIPT.

5... **Running a model in WinBUGS**

5.4. Deviance Information Criterion (DIC) in WiNBUGS

DIC

- was introduced by Spiegelhalter et al. (2002)
- measure of model comparison and adequacy.
- Equivalent to AIC for simple models
- Smaller DIC values indicate better-fitting models.
- must be used with caution. It assumes that the posterior mean can be used as a "good'' summary of central location for description of the posterior distribution.



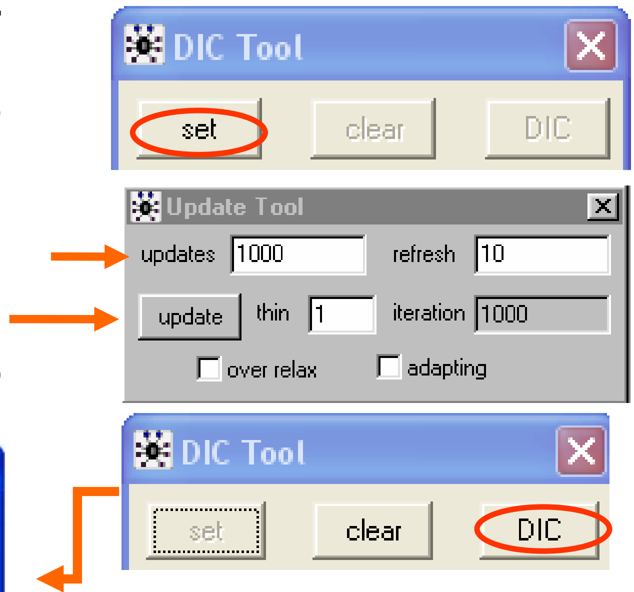
- Problems when posterior distributions are not symmetric or unimodal.

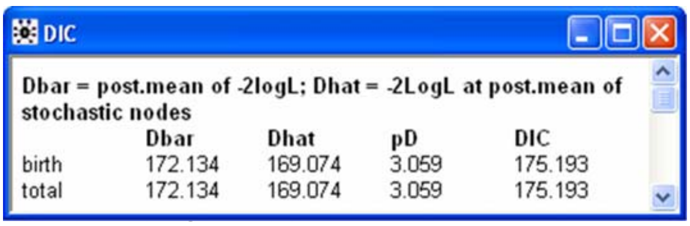
5... **Running a model in WinBUGS**

5.4. Deviance Information Criterion (DIC) in WiNBUGS

The procedure for calculating DIC is the following:

1. Press the set button to start storing DIC values.
2. Return to the Update Tool and generate additional values (e.g. 1000).
3. Press the DIC button to obtain the results.





Dbar = post.mean of -2logL; Dhat = -2LogL at post.mean of stochastic nodes

	Dbar	Dhat	pD	DIC
birth	172.134	169.074	3.059	175.193
total	172.134	169.074	3.059	175.193

6… Model Code Details

6… Model Code Details

6.1. General details

Types of nodes (parameters) in WinBUGS

1. **Constants**: Fixed values or data (usually specified in the data section)
2. **Random**: Random variables of the model that are characterized by a distribution.
3. **Deterministic**: simple functions or transformations of other model parameters.

   Can be either random of constants.

6… Model Code Details

6.1. General details

Node Specification

Random nodes are specified using the syntax

```
Variable ~ Distribution(parameter1, parameter2,…)
```

e.g. `X~dnorm( mu, tau )` for X~Normal(μ, 1/τ)

Deterministic nodes are defined using the assignment sign `<-`

e.g. `s2 <- 1/tau` for setting $\sigma^2=1/\tau$

6… Model Code Details

6.1. General details

Node names, dimensions and elements:

1. **Unidimensional**: just write a name e.g. x
2. **Vector**: name followed by brackets e.g. v[]
   - v[]: all elements of vector v
   - v[i]: the i^{th} element of v
   - v[n:m]: elemements n,…,m of vector v.
3. **Matrix:** name followed by [,] (the comma denotes the 2 dimensions – rows and columns of the matrix) e.g. m[,]
4. **Array:** name followed by "[", a number of commas and "]" (to denote the number of dimensions – 2 commas denote a 3 dimensional array) e.g. a[,,]

Node names, dimensions and elements (2)

Matrix:

- M[,]: all elements of matrix M
- M[i,j]: element of i^{th} row and j^{th} column of matrix M
- M[i,]: elements of i^{th} row of matrix M
- M[,j]: elements of j^{th} column of matrix M
- M[n:m,j]: elements of n^{th},…,m^{th} rows of j^{th} column of matrix M
- M[i, n:m]: elements of n^{th},…,m^{th} columns of i^{th} row of matrix M
- M[n:m,]: elements of n^{th},…,m^{th} rows of matrix M
- M[,n:m]: elements of n^{th},…,m^{th} columns of matrix M
- M[n:m,k:l]: elements of n^{th},…,m^{th} rows of k^{th},…,l^{th} column of matrix

Node names, dimensions and elements (3)

Array

- A[, ,]: all elements of array **A**
- A[i, j, k]: the A_{ijk} element of the array **A**
- A[i, ,]: elements with first dimension equal to i of the array **A**
- A[, j ,]: elements with second dimension equal to j of the array **A**
- A[, , k]: elements with third dimension equal to k of the array **A**
- A[i, j,]: elements with first dimension equal to i and the second equal to j of the array **A**. Similarly we define A[i, , k] and A[, j, k].
- A[n:m, ,]: elements with first dimension equal to n, …,m of the array **A**.
- Similarly we define A[, n:m,] and A[, n:m,].
- A[n_1 :m_1 , n_2 :m_2 , n_3 :m_3]: elements A_{ijk} with $i=n_1,\ldots,m_1$, $j=n_2,\ldots,m_2$ and $k=n_3,\ldots,m_3$.

6… **Model Code Details**

6.1. General details

Node names, dimensions and elements (4)

- Within brackets, calculations using basic operations (+, -, * and /) are allowed.
- Nested indexing of the type x[y[i]] can be used.

6… **Model Code Details**

6.2. Functions Description

Simple arithmetic functions

Various simple arithmetic functions are available in WinBUGS, including

- The absolute value (abs)
- The sine and cosine functions (sin, cos)
- The exponent and the natural logarithm (exp, log)
- The logarithm of the factorial of an integer number (logfact)
- The logarithm of the gamma function (loggam)
- The square root value (sqrt)
- round and trunc => obtain the closest and the lower closest integer values, respectively.

All these functions require as an argument a single scalar node.

Simple arithmetic functions (2)

All these functions require as an argument a single scalar node.

In order to set y equal to a function of x, we write

y <- one.parameter.function(x)

e.g., if y=|x|, then in WinBUGS we write

y <- abs(x)

- y <- max(x1, x2): compare two values x_1, x_2 and keep (in y) the maximum one.
- Similar also for y<-min(x1,x2)
- For $y=x^z$ we write y <- pow(x, z)

Statistical functions

Within \winbugs, simple statistical functions can be calculated.

- mean(v[]) : Sample mean of vector ***v***
- sd(v[]) : Sample standard deviation of ***v***
- sum(v[]) : Sum of all values of v
- y <- rank(v[], k) : rank of the k^{th} element of vector ***v***
- y <- ranked(v[], k) : we obtain the element of ***v*** with rank equal to k.
  - ✓ Minimum => miny <- ranked(v[], 1)
  - ✓ Maximum => maxy <- ranked(v[], n)
  - ✓ Median
    - ✓ => mediany <- ranked(v[], (n+1)/2) if n is odd
    - ✓ => mediany <- 0.5*(ranked(v[], n/2)+ranked(v[], n/2+1)) if n is even.
- y <- phi(x): CDF for x coming from a N(0,1)

6… Model Code Details

6.2. Functions Description

Link functions

- cloglog(p) : log(-log(1-p))
- logit(p) : log{p/(1-p)}
- probit(p) : Φ-1(p); inverse function of the CDF of N(0,1).
  i.e. probit(y) <- x is equivalent to y <- phi(x)
- log(x) : logarithm of x.

- Link functions can be used on the left side of assignment.
  e.g. logit(p) <- a + b*x
- Specify link functions in generalized linear models

6… Model Code Details

6.2. Functions Description

Matrix functions

- M2[1:K,1:K] <- inverse(M1[,]): M2 = Inverse of matrix M1 of dimension K × K

  M2 must be written including its dimension indices i.e., the matrix name must be followed by [1:K,1:K]
- y <- logdet(M1[,]): y is the log of the determinant of matrix M1

Vector functions

- sum(v[]): sum of the elements of vector ***v***.
- inprod(v1[], v2[]): inner product of the elements of vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$

6… **Model Code Details**
6.2. Functions Description

Binary indicator Functions

- y<-equals(x, z) : compares x and z. y=1 if x=z and or y=0 if $x \neq z$
- y<-step(x) : y=0 if x<0 or y=1 if $x \geq 0$

These are used as if statements.

The cut function

This command is used when we do not wish the posterior of each parameter to be updated from the likelihood.

6… **Model Code Details**
6.3. Using for loops

- Functions for multidimensional nodes are limited
- Most of their calculations are completed separately for each element using the *for loop*.
- For example to calculate the sum of two vectors of length n we write:

```
for ( i in 1:n){
    y[i] <- x[i] + z[i]
}
```

It means: calculate $z_i = x_i + z_i$ *for all* $i=1,2,\ldots,n$

- Similarly to add two (n x p) matrices we write:

```
for ( i in 1:n){
   for (j in 1:p) {
            A[i,j] <- X[i,j] + Z[i,j]
}}
```

6… Model Code Details
6.3. Using for loops (2)

To multiply two matrices of dimensions (n x p) and (p x k) we use the *inprod* function (inner product):

```
for ( i in 1:n){
  for (j in 1:k) {
        A[i,j] <- inprod(X[i,], Z[,j])
}}
```

To define that all the elements of a random vector **x**, with dimension n, follow independent normal distributions with mean μ_i and common precision τ we write

```
for ( i in 1:n){
   x[i] ~ dnorm(mu[i], tau)
}
```


6… Model Code Details
6.4. Brackets in WinBUGS

1. Parentheses ( ) are used
   - in mathematical expressions and computations.
   - in functions surrounding their arguments.
   - in *for* loops to declare the values of the index.
2. Square brackets [ ] are used to specify the elements of a vector or array.
3. Curly brackets { } are used to declare the beginning and the end of the model and the *for* loop.

6… Model Code Details
6.5. Model Specification

Likelihood specification *(using for syntax)*

Example 1: Simple normal model

$X_i \sim \text{Normal}(\mu, \sigma^2 = 1/\tau)$

```
for ( i in 1:n){
   x[i] ~ dnorm(mu, tau)
}
```

Example 2: Simple Linear Regression Model

$Y_i \sim \text{Normal}(\mu_i, \sigma^2 = 1/\tau)$ with $\mu_i = \alpha+\beta\, x_i$

```
for ( i in 1:n){
        y[i] ~ dnorm(mu[i], tau)
        mu[i] <- alpha+beta*x[i]
}
tau<-1/sigma2
```

6… Model Code Details
6.5. Model Specification

Prior specification

Example 1: Simple normal model

$X_i \sim \text{Normal}(\mu, \sigma^2 = 1/\tau)$ => parameters $\theta=(\mu,\tau)$

Priors: $\mu \sim \text{Normal}(0, 100)$ → `mu ~ dnorm(0, 0.01)`

$\tau \sim \text{Gamma}(0.01, 0.01)$ → `tau ~ dgamma (0.01, 0.01)`

Example 2: Simple Linear Regression Model

$Y_i \sim \text{Normal}(\mu_i, \sigma^2 = 1/\tau)$ with $\mu_i = \alpha+\beta\, x_i$ => parameters $\theta=(\alpha,\beta,\tau)$

Priors: $\alpha \sim \text{Normal}(0, 100)$

$\beta \sim \text{Normal}(0, 100)$

$\tau \sim \text{Gamma}(0.01, 0.01)$

→

```
alpha ~ dnorm(0, 0.01)
beta ~  dnorm(0, 0.01)
tau ~ dgamma (0.01, 0.01)
```

6… Model Code Details

6.5. Model Specification

Data transformations

For transformations of the response data (which are *random nodes*) you can

- insert directly transformed data (simplest way).
- specify transformation with WinBUGS code using its commands

This is the only case that a node can be specified **twice**:

1) to define the transformation of the original data and

2) then to assign the random distribution.

Example

```
for ( i in 1:n){
    z[i] <-log(abs(y[i]))
    z[i] ~dnorm(mu, tau)
}
```

6… Model Code Details

6.5. Model Specification

Data Specification

1. Rectangular data format.

- Simple and similar to the text files used by most statistical packages.
- It can be used to specify a series of variables-vectors of the same length or a matrix (or array).
- Simple vectors (e.g. vectors y, x1, x2, x3),
  - ✓ we specify the names followed by square brackets in the first line
  - ✓ the values of each observation in each line, separated by empty spaces.
  - ✓ Be careful: The data must conclude with the command END followed by at least one blank line.

```
y[] x1[] x2[] x3[]
10  20   23   12
11  23   11   97
……………………………..
44  25   33   12
END
```

<- BLANK LINE

6… **Model Code Details**

6.5. Model Specification

Data Specification

2. List data format.

- Similar to R/Splus list objects.
- This format can be used to specify all type of data (single constant numbers, i.e. scalar nodes, vectors, matrices, and arrays).
- Syntax: *list(…)*
- Within the parentheses we specify each variable/node separated by commas.

6… **Model Code Details**

6.5. Model Specification

List data format (cont.)

We can define

1. Scalars => scalar.name = scalar.value
2. Vectors => vector.name = c(value1,..., valuen)
3. Matrices => matrix.name = structure(
   .Data = c(value1,...,valuen)
   .Dim = c(row.number, column.number))
4. Arrays => same syntax as matrices but *.Dim* argument will have at least three values specifying the length of each corresponding dimension.

List data format (cont.)

Example:

```
A      = structure(
              .Data = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
              .Dim = c(3, 4) )
```

Reading elements by rows →

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

List data format (cont.)

A simple example of data specification.

Assume we have the following dataset with n=4 observations and p=5 variables.

y	x1	x2	gender	age
12	2	0.3	1	20
23	5	0.2	2	21
54	9	0.9	1	23
32	11	2.1	2	20

WinBUGS Syntax

```
list(n=4, p=5, y=c(12, 23, 54, 32), x1=c(2, 5, 9, 11), x2=c(0.3, 0.2,
    0.9, 2.1), gender=c(1,2,1,2), age=c(20,21,23,20) )
```

6… **Model Code Details**

6.5. Model Specification

Initial Values

- They are used to initiate the MCMC sampler.
- Their format is the same as the list data format.
- Initial values must be provided for all random nodes except for the response data/variables.
- The user does not need to specify all initial values but can generate all or portion of them (using gen inits option).
- Users must be careful when using randomly generated initial values

  => problems when certain parameters are initialized using inappropriate values

  => numerical problems (trap messages in WinBUGS) or slow convergence of the algorithm.

7… **Additional examples**

7.1. Example 2: Binomial Data

7.2. Example 3: Bernoulli Data

7.3. Example 4: Models For 2x2 Contingency Tables

7.4. Example 5: Model For 3-way Tables

7.5. Zero-ones trick

7… Additional examples

7.1. Example 2: Models for Binomial Data

The beetles dataset

- WinBUGS examples vol 2, page 32 – 34

  (or classic BUGS examples, vol 2. p. 43)
- Bliss (1935)
- 8 groups of insects were exposed on different levels of Carbon dislphide
- We record
  - ✓ Concentration (X_i)
  - ✓ Total number of insects in group (n_i)
  - ✓ Total number of insects died (r_i)

7… Additional examples

7.1. Example 2: Models for Binomial Data

(1) $r_i \sim \text{Binomial}(p_i, n_i)$

(2) $\eta_i = \alpha^* + \beta(x_i - \bar{x})$

(3) $\text{logit}(p_i) = \eta_i$

for $i=1,\ldots,8$

```
for (i in 1:n) {
    r[i]~dbin(p[i],n[i])
    z[i]<-x[i]-mean(x[])
    logit(p[i])<-a.star+b*z[i]
}
```

Prior distributions

$f(\alpha^*) = \text{Normal}(0, 10^4)$ `a.star~dnorm(0.0,1.0E-04)`

$f(\beta) = \text{Normal}(0, 10^4)$ `b~dnorm(0.0,1.0E-04)`

$\alpha = \alpha^* - \beta\bar{x}$ `a<-a.star-b*mean(x[])`

Link functions

- $\text{logit}(p) = \log\{p/(1-p)\}$ — `logit(p[i])<-a.star+b*z[i]`
- $\text{probit}(p) = \Phi^{-1}(p)$ — `probit(p[i])<-a.star+b*z[i]`
- $\text{cloglog}(p) = \log(-\log(1-p))$ — `cloglog(p[i])<-a.star+b*z[i]`

7… Additional examples

7.1. Example 2: Models for Binomial Data

FITTED VALUES CAN BE COMPUTED BY

```
r.hat[i]<-n[i]*p[i]
```

ODDS RATIO FOR LOGIT MODELS

```
odds.ratio<-exp(b)
```


7… Additional examples

7.2. Example 3: Models for Bernoulli Data

WAIS Dataset

From Agresti (1990) p. 122–123,

A sample of elderly people were examined for the existence of senility symptoms (1=yes, 0=no).

Explanatory variable: WAIS = score in a sub-scale of Wecshler Adult Intelligence Scale

AIM

- Estimate which WAIS score corresponds to 50% probability of having senility symptoms
- What is the probability of senility symptoms for a typical subject (i.e. with WAIS equal to the sample mean of WAIS)

7… Additional examples

7.2. Example 3: Models for Bernoulli Data

THE MODEL

```
for (i in 1:n) {
     symptom[i]~dbern( p[i] )
     logit( p[i] ) <- a+b*wais[i]
}
```

The WAIS score for p=1/2 is given by

```
x.half<- -a/b;
```

The probability of the person with WAIS equal to the sample mean is given by

```
p.mean<-
  exp(a+b*mean(x[]))/(1+exp(a+b*mean(x[])))
```

7… Additional examples

7.3. Example 4: Poisson models for 2x2 Contingency Tables

Breast cancer & age at first birth

Mahon *et.al.* (1970). *Bulletin of the world health organisation*

Study of the possible relationship between age at 1st birth and breast cancer

Cases: Selected hospitals in USA, Greece, Yugoslavia, Brazil, Taiwan and Japan.

Controls: women with comparable age from the same hospitals

7… **Additional examples**

7.3. Example 4: Poisson models for 2x2 Contingency Tables

	AGE AT FIRST BIRTH	
STATUS	Age>29 (1)	Age<30 (0)
Case (1)	683	2537
Control (0)	1498	8747

7… **Additional examples**

7.3. Example 4: Poisson models for 2x2 Contingency Tables

Status	**Age**	**Counts**
1	1	683
1	0	2537
0	1	1498
0	0	8747

7… Additional examples

7.3. Example 4: Poisson models for 2x2 Contingency Tables

THE MODEL

$Counts_i \sim Poisson(\lambda_i)$

$Log(\lambda_i) = \mu + a \times status_i + b \times age_i + ab \times status_i \times age_i$

for i=1,2,3,4

```
for (i in 1:4) {
   counts[i]~dpois(lambda[i])
   log (lambda[i])<-mu+
     a*status[i]+b*age[i]+ab*status[i]*age[i]
   }
```

ODDS RATIO: `odds.ratio<-exp(ab)`

7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

Lung cancer, smoking and passive smoking

- Sandler, Everson & Wilcox (1985) *Amer.Journal of Epidemiology*
- Study of 518 patients with lung cancer aged 15-59 and 518 controls matched for age and gender.
- AIM: Estimate the effect of passive smoking on the risk of cancer. Passive cancer was defined positively when the spouse was smoking at least one cigarette for the last 6 months
- Confounder: the smoking status of the subject
- For 2x2xJ contingency tables => common odds ratios are estimated using the Maentel-Haenzel $OR_{MH}=(\Sigma a_i d_i/n_i)/(\Sigma b_i c_i/n_i)$

7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

	Non Smokers (0)		Smokers (1)	
	Passive Smoker (1)	Non Passive Smoker(0)	Passive Smoker (1)	Non Passive Smoker (0)
Case(1)	120	111	161	117
Control (0)	80	155	130	124

7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

Two Analyses/Models

1st analysis/model

- We fit a model in each 2x2 table and estimate the odds ratios as in example 4
- Compare the posterior distributions of two odds ratios

2nd analysis/model

- We fit a model in each 2x2 table
- Estimate a common odds ratio for each table

7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

1st analysis/model

```
model{
    #model for 1st table (nonsmokers)
    for (i in 1:4) {
          counts[i]~dpois( lambda[i] );
          log(lambda[i])<-b[1,1]+b[1,2]*status[i]
                   +b[1,3]*passive[i]
                   +b[1,4]*status[i]*passive[i];
    }
```


7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

```
    #model for 2nd table (smokers)
    for (i in 5:8) {
          counts[i]~dpois( lambda[i] );
          log(lambda[i])<-b[2,1]+b[2,2]*status[i]
                   +b[2,3]*passive[i]
                   +b[2,4]*status[i]*passive[i];
    }
    #priors
    for (i in 1:2){
         for (j in 1:p){
              b[i,j]~dnorm(0.0, 1.0E-04)
         }
    }
```

7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

1st analysis/model

MODEL

```
#model for 1st table (nonsmokers)
for (i in 1:4) {
 counts[i]~dpois( lambda[i] );
 log(lambda[i])<-b[1,1]+b[1,2]*status[i]
              +b[1,3]*passive[i]
              +b[1,4]*status[i]*passive[i];}
#model for 2nd table (smokers)
for (i in 5:8) {
 counts[i]~dpois( lambda[i] );
 log(lambda[i])<-b[2,1]+b[2,2]*status[i]
              +b[2,3]*passive[i]
              +b[2,4]*status[i]*passive[i];  }
```


7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

2nd analysis/model

MODEL

```
#model for 1st table (nonsmokers)
for (i in 1:4) {
 counts[i]~dpois( lambda[i] );
 log(lambda[i])<-b[1,1]+b[1,2]*status[i]
              +b[1,3]*passive[i]
              +  ab  *status[i]*passive[i];}
#model for 2nd table (smokers)
for (i in 5:8) {
 counts[i]~dpois( lambda[i] );
 log(lambda[i])<-b[2,1]+b[2,2]*status[i]
              +b[2,3]*passive[i]
              +  ab  *status[i]*passive[i];  }
```

7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

RESULTS

	MLE	**Posterior Mean**	**95% posterior credible interval**
ANALYSIS 1			
OR_0	2.09	2.07±0.036	1.47 - 3.09
OR_1	1.31	1.33±0.022	0.97 - 1.88
ANALYSIS 2			
Common			
OR_{MH}	1.63	1.61 ± 0.0087	1.27 - 2.06

FIGURE 12:
ESTIMATED POSTERIOR DISTRIBUTIONS OF ODDS RATIOS

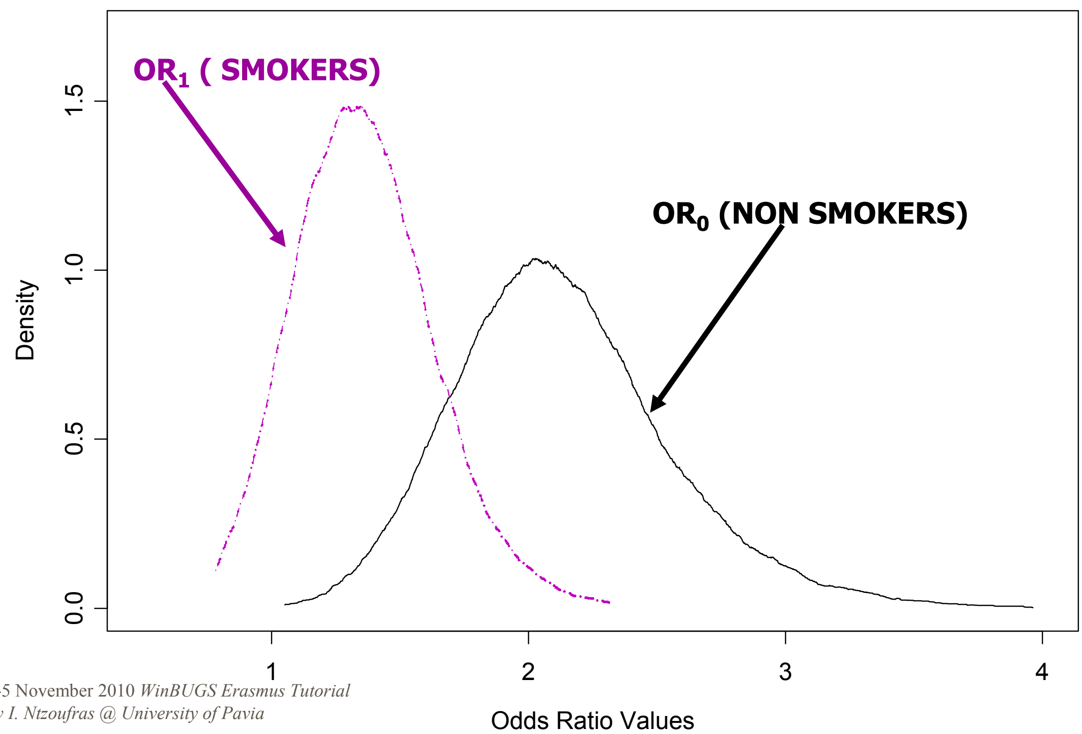
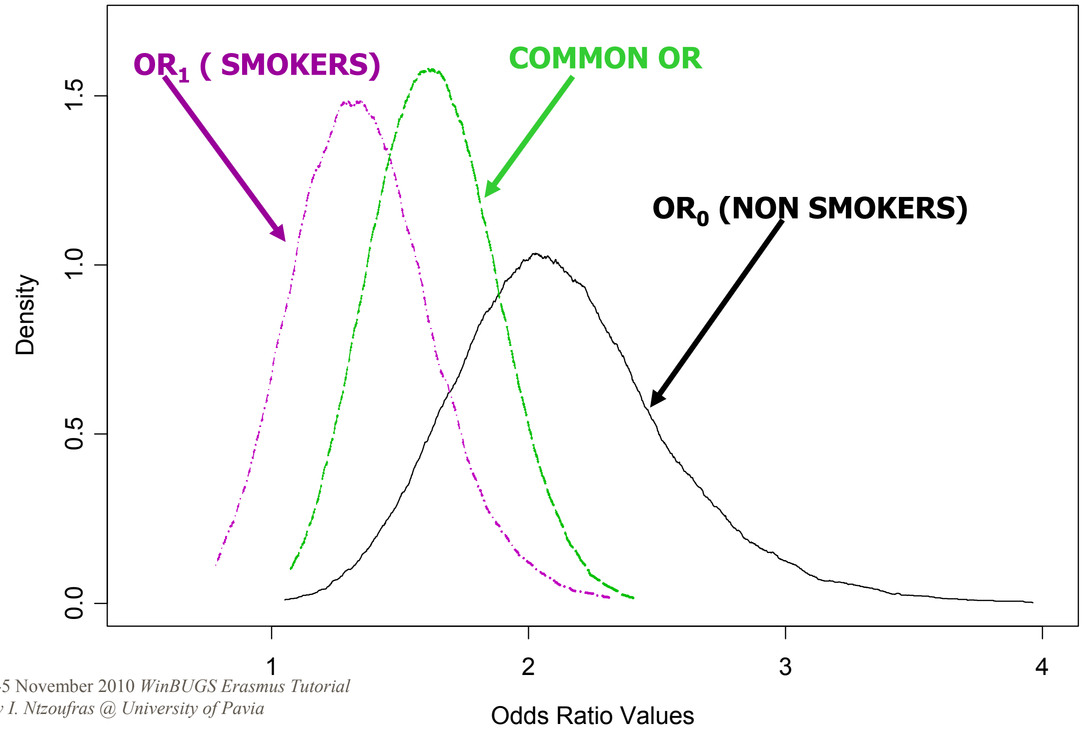


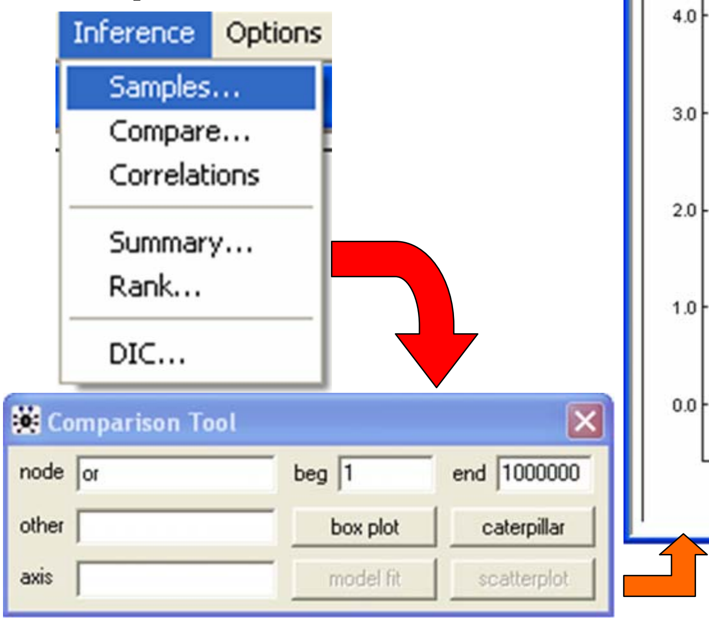
FIGURE 13:
ESTIMATED POSTERIOR DISTRIBUTIONS OF ODDS RATIOS

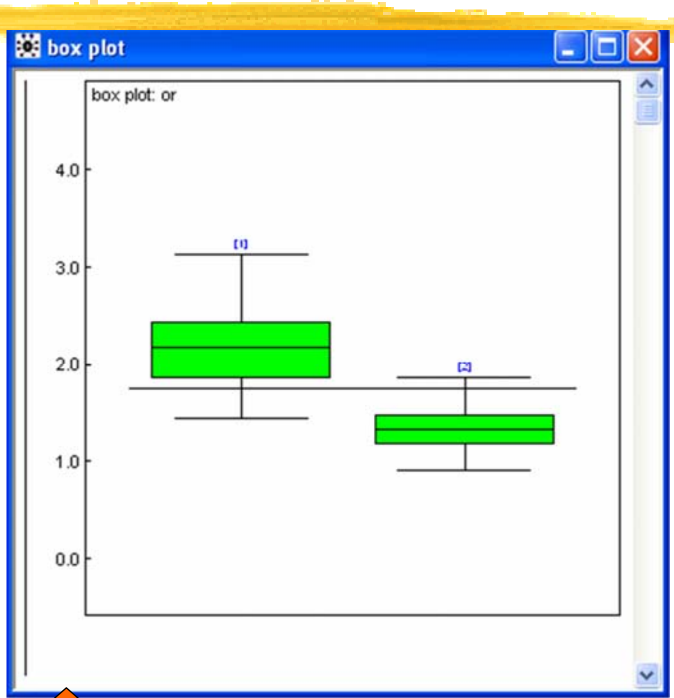


7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

Comparison of the posterior boxplots for Ors in model 1



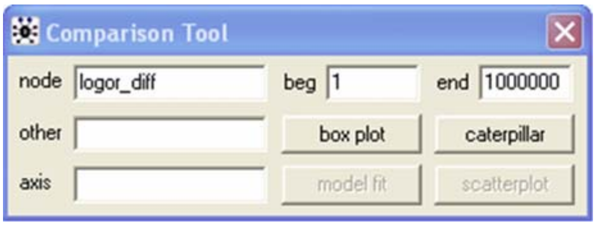


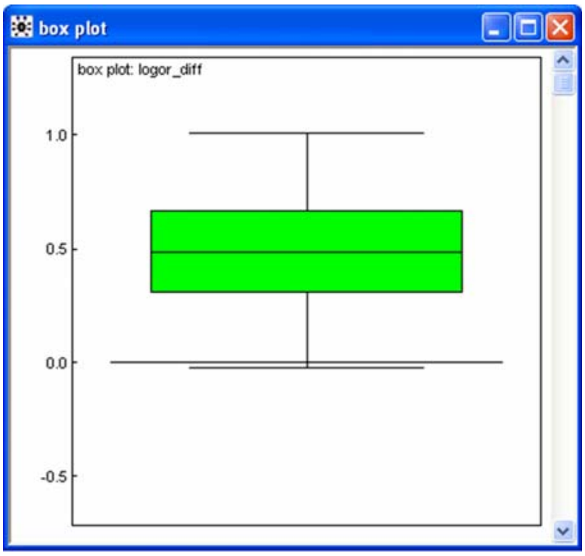
7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

Due to autocorrelations and crosscorrelations, it is better to consider the difference of the log ORs

```
logor_diff <-b[1,4]-b[2,4]
```





7… Additional examples

7.4. Example 5: Estimation of common OR in 2x2x2 Contingency Tables

DIC

```
Dbar = post.mean of -2logL;
Dhat = -2LogL at post.mean of stochastic nodes
```

	Dbar	Dhat	pD	DIC
Model 1	61.309	53.197	8.112	69.420
Model 2	63.416	56.462	6.954	70.371

7… **Additional examples**

7.5. Specification of non-standard distributions – The Zero-Ones Trick

- What happens if we wish to use a distribution which is not directly available ir WinBUGS?
- We can use the zeros or ones trick

7… **Additional examples**

7.5. Specification of non-standard distributions – The Zero-Ones Trick

New prior using the zeros trick

Let us assume that we wish to define a new prior f(θ) for θ

1. Set the prior of θ to be flat over the set of possible values using the commands **`dflat()`** or **`dunif()`**
2. Set a new node/variable (e.g. **`zero`**) equal to 0
3. Define that this node (e.g. **`zero`**) is stochastic and follows the **`Poisson`** distribution with mean equal to **λ**
4. Set the mean **`λ=-logf(θ)`**

7… Additional examples

7.5. Specification of non-standard distributions – The Zero-Ones Trick

New prior using the zeros trick (2)

Example

$f(\theta) = z\,(z + \omega\theta)^{\theta-1}\, e^{-(z + \omega\,\theta)}/(\theta!)$.

Generalized/Lagrangian Poisson with mean $z/(1-\omega)$, variance= $z/(1-\omega)^3$, dispersion index DI= $1/(1-\omega)^2$

```
theta ~ dflat()
zero <- 0
zero ~ dpois(lambda)
lambda <- -( log(zeta)+(theta-1)*
  log(zeta+omega*theta)-(zeta+omega*theta)-
  logfact(theta) )
```

7… Additional examples

7.5. Specification of non-standard distributions – The Zero-Ones Trick

Why this trick works?

f(zero|θ) is the Poisson with equal to minus log-likelihood

φ(θ) is the flat pseudo-prior used to indirectly specif the actual prior

f(θ|zero) is the actual prior

$$f(\theta|zero) \propto f(zero=0|\theta)\times\varphi(\theta) = e^{-\lambda}\times 1 = e^{-(-\log f(\theta))}\times 1 = f(\theta)$$

New prior using the zeros trick (3)

See WinBUGS manual in the new-prior Section for an example with the normal distribution

BE CAREFUL: This method generates samples with

1. Large auto-correlations
2. Slow convergence
3. Large Monte–Carlo error

i.e. it is slow computationally and we usually need to leave WinBUGS to run for a large number of iterations.

New likelihood using the zeros trick

Let us assume that we wish to specify a model with likelihood $y_i \sim f(y_i|\theta)$

1. Set a constant C equal to a large number to ensure that the mean of the Poisson-zeros pseudo-data will be positive
2. Set a vector of pseudo-data **zero** (with length equal to the size of the actual data) equal to zero.
3. Specify that each z_i (i.e. zero[i]) follows the `Poisson` distribution with mean λ_i
4. Set λ_i`=-logf(`y_i`|θ)`

New likelihood using the zeros trick

Example

$f(y_i| z, \omega) = z (z + \omega y_i)^{y_i-1} e^{-(z + \omega y_i)}/(y_i!)$.

Generalized/Lagrangian Poisson with parameters

$\boldsymbol{\theta}=(z,\omega)$, mean=$z/(1-\omega)$ and variance= $z/(1-\omega)^3$,

```
C <- 10000
for (i in 1:N) {
      zeros[i] <- 0
      zeros[i] ~ dpois(lambda[i])
      lambda[i] <- -L[i] + C
      L[i] <-  -( log(zeta)+(y[i]-1)*
  log(zeta+omega*y[i])-(zeta+omega*y[i])-
  logfact(y[i]) )                  }
```

7… Additional examples
7.5. Specification of non-standard distributions – The Zero-Ones Trick

Why this trick works?

$$\prod_{i=1}^{n} f(zero_i \mid y_i, \theta) = \prod_{i=1}^{n} e^{-\lambda_i} = \prod_{i=1}^{n} e^{-(-L_i + C)}$$

$$= \prod_{i=1}^{n} e^{\log f(y_i|\theta) - C} \propto \prod_{i=1}^{n} f(y_i \mid \theta)$$

7… Additional examples

7.5. Specification of non-standard distributions – The Zero-Ones Trick

Ones Trick

Similar to the previous approach but

1. We use the Bernoulli(p[i]) distribution
2. Pseudo-data ones[i] <- 1
3. p[i] <- exp(log.likelihood[i] - C)
4. C is such that p[i] < 1

7… Additional examples

7.5. Specification of non-standard distributions – The Zero-Ones Trick

Example of Generalized Poisson Likelihood

(Rosner 1994, page 94)

Number of deaths fro polio for 1968-76

1968	1969	1970	1971	1972	1973	1974	1975	1976
24	13	7	18	2	10	3	9	16

7… Additional examples

7.5. Specification of non-standard distributions – The Zero-Ones Trick

Example of Generalized Poisson Likelihood

```
model {
C<-10000
for (i in 1:9) {
        zeros[i]<-0
        zeros[i]~dpois( lambda[i] )
        lambda[i]<- C - loglike[i]
        loglike[i] <- log(zeta)+(y[i]-1)* log(zeta+omega*y[i])-
        (zeta+omega*y[i])-logfact(y[i])
   }
   zeta~dgamma(0.001, 0.001)
   omega~dbeta(1,1)
   mean<-zeta/(1-omega)
   var<-zeta/pow(1-omega,3)
   DI<-1/((1-omega)*(1-omega))
}
DATA : list( y=c(24, 13, 7, 18, 2, 10, 3, 9, 16) )
INITS: list( zeta=1, omega=0.5 )
```

8… A SIMPLE HYPOTHESIS TEST

8.1. Introduction: Posterior Odds

LET US CONSIDER THE ESTRIOL EXAMPLE

THEN H_0: $\beta=0$ vs. H_1 $\beta \neq 0$

EQUIVALENT TO THE COMPARISON OF MODELS

m_0 : $Y \sim N(\alpha, \sigma^2)$ and

m_1 : $Y \sim N(\alpha+\beta X, \sigma^2)$

8… A SIMPLE HYPOTHESIS TEST

8.1. Introduction: Posterior Odds

Posterior Model Odds of model m_0 vs. model m_1:

$$PO_{01} = \frac{f(m_0 \mid \mathbf{y})}{f(m_1 \mid \mathbf{y})} = \frac{f(\mathbf{y} \mid m_0)}{f(\mathbf{y} \mid m_1)} \times \frac{f(m_0)}{f(m_1)}$$

Bayes Factor **Prior Model Odds**

8… A SIMPLE HYPOTHESIS TEST

8.1. Introduction: Posterior Odds

$f(m)$: Prior model probability

$f(m|\mathbf{y})$: Posterior model probability

8... A SIMPLE HYPOTHESIS TEST

8.2. Posterior Model Probabilities in BUGS

In BUGS we can estimate the $f(m|\mathbf{y})$ by inserting a latent binary indicator γ:

$$\mathbf{Y \sim Normal(\, \alpha + \gamma \times \beta X \,, \sigma^2 \,).}$$

For details see

- Dellaportas, Forster and Ntzoufras (2002). *Statistics and Computing*, **12**, 27–36.
- Dellaportas, Forster and Ntzoufras (2000). In *Generalized Linear Models: A Bayesian Perspective*, 271–288.
- Ntzoufras (2002). Gibbs Variable Selection Using BUGS. *Journal of Statistical Software*. Volume **7**, Issue 7

8... A SIMPLE HYPOTHESIS TEST

8.2. Posterior Model Probabilities in BUGS

Also in

- Ntzoufras (2009). Bayesian Modeling Using WinBUGS. Wiley (chapter 11)
- Ntzoufras (2009).

  http://stat-athens.aueb.gr/~jbn/courses/2009_varsel_herriot/variable_selection_tutorial_handouts.pdf
- Ntzoufras (2002). Tutorial on Bayesian Model Selection (Msc Hand outs)

  http://stat-athens.aueb.gr/~jbn/courses/bugs2/handouts/modelsel/4_1_tutorial_handouts.pdf

8… A SIMPLE HYPOTHESIS TEST

8.2. Posterior Model Probabilities in BUGS

The distribution $f(\beta|\ \gamma=0)$ is also called pseudoprior or proposal distribution

- Does not Influence the Posterior
- Does Influence the convergence (needs to be close to the posterior $f(\beta|\ \mathbf{y}, \gamma=1)$)

For simplicity we will assume

$$f(\beta|\ \gamma=0)= f(\beta|\ \gamma=1)$$

[it is ok for this simple example]

8… A SIMPLE HYPOTHESIS TEST

8.2. Posterior Model Probabilities in BUGS

(1) $Birth_i \sim Normal(\mu_i, \sigma^2)$

(2) $\eta_i = \alpha + \beta \times Estriol_i$

(3) $\mu_i=\eta_i=\alpha+\beta\times Estriol_i$

for i=1,…,31

```
for (i in 1:n) {
  Birth[i]~dnorm(mu[i],tau)
  x[i]<-estriol[i]-mean(estriol[])
  mu[i] <-a+        b*x[i]
}
```

Prior distributions

$f(\alpha)$=Normal (0, 10^4)

$f(\beta)$=Normal (0, 10^4)

$f(\tau^2)$=Gamma(10^{-4},10^{-4})

$\sigma^2=1/\ \tau$

```
a~dnorm(0.0,1.0E-04)
b~dnorm(0.0,1.0E-04)
tau~dgamma(1.0E-04,1.0E-04)

s2<-1/tau
```

8… A SIMPLE HYPOTHESIS TEST

8.2. Posterior Model Probabilities in BUGS

(1) $Birth_i \sim Normal(\mu_i, \sigma^2)$

(2) $\eta_i = \alpha + \gamma \times \beta \times Estriol_i$

(3) $\mu_i = \eta_i = \alpha + \beta \times Estriol_i$

for i=1,...,31

```
for (i in 1:n) {
   Birth[i]~dnorm(mu[i],tau)
   x[i]<-estriol[i]-mean(estriol[])
   mu[i] <-a+ gamma*b*x[i]
}
```

Prior distributions

$f(\alpha)$=Normal (0, 10^4)

$f(\beta)$=Normal (0, 10^4)

$f(\tau^2)$=Gamma(10^{-4},10^{-4})

$\sigma^2 = 1/\tau$

$\gamma \sim$ Bernoulli (0.5)

```
a~dnorm(0.0,1.0E-04)
b~dnorm(0.0,1.0E-04)
tau~dgamma(1.0E-04,1.0E-04)
s2<-1/tau
gamma~dbern(0.5)
```


8… A SIMPLE HYPOTHESIS TEST

8.2. Posterior Model Probabilities in BUGS

After Burn-in of 1000 iterations and 20,000 iterations

$$f(\gamma=1|\mathbf{y}) = 0.6268$$

$$PO_{10}=BF_{10} = 1.68$$

8… A SIMPLE HYPOTHESIS TEST

8.2. Posterior Model Probabilities in BUGS

- We have not discussed about priors for model selection [too large for this tutorial].
- Large Values of the prior variance of β will activate Bartlett - Lindley's Paradox => $f(\gamma=1|\mathbf{y}) \rightarrow 0.0$
- Unit information prior (BIC):

Variance of β =

sample size X posterior variance with flat prior

8… A SIMPLE HYPOTHESIS TEST

8.2. Posterior Model Probabilities in BUGS

Prior Variance of β = 31 X $(0.1431)^2$ = 0.6348

Prior Precision of β = 1/0.6348 = 1.575

After Burn-in of 1000 iterations and 20,000 iterations of generated values

$$f(\gamma=1|\mathbf{y}) = 0.9922$$

$$PO_{10}=BF_{10} = 127.20$$

8... A SIMPLE HYPOTHESIS TEST

8.3. Summary of Hypothesis Test Example

This example was only for illustration

Don't try directly Bayesian model selection at home

Be very careful in the construction of priors and pseudo-priors

A Short Introduction to Bayesian Modeling Using WinBUGS

END OF TUTORIAL

E-mail: ntzoufras@aueb.gr

ΑΣΟΕΕ 1920

Department of Statistics,
Athens University of
Economics & Business