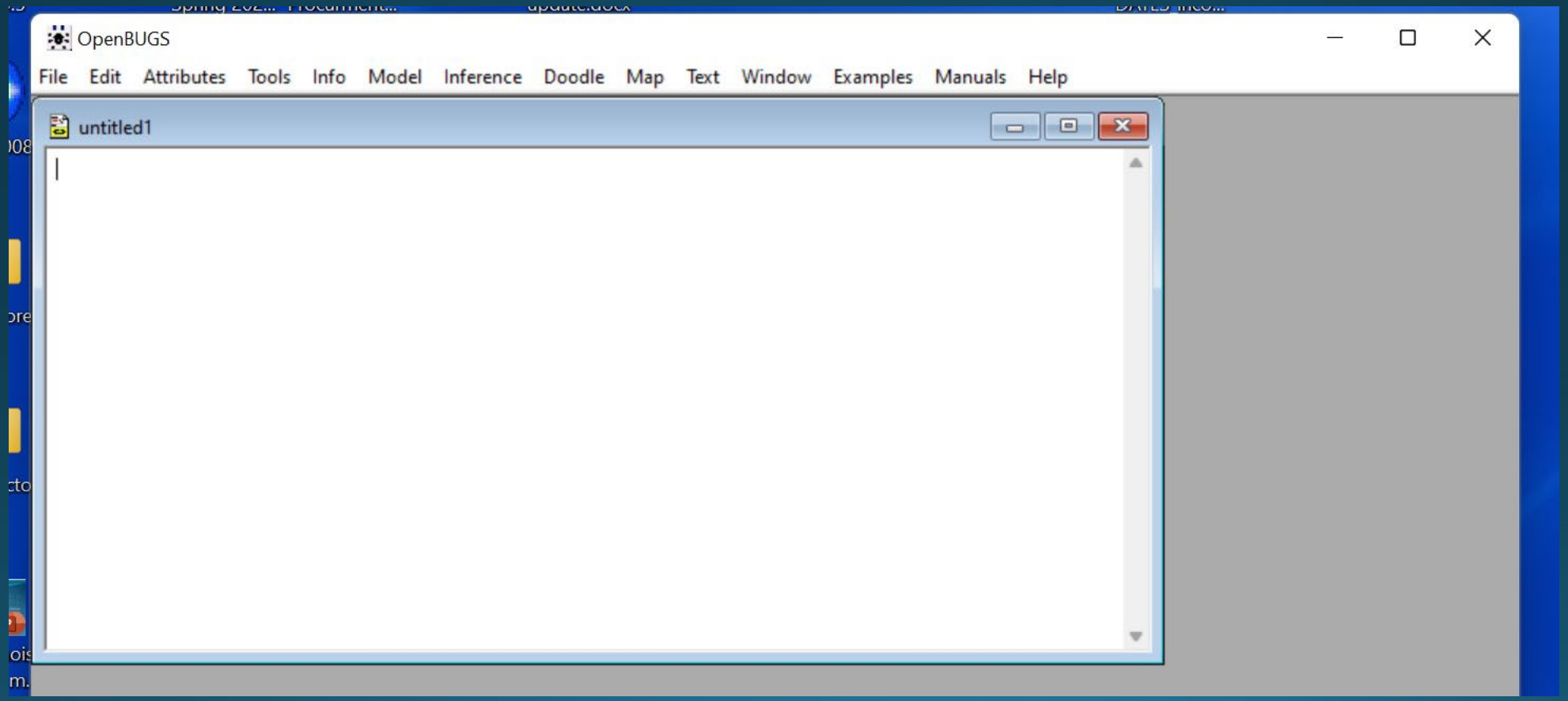
Open/WinBUGS

Overview

- BUGS (Bayesian inference Using Gibbs Sampling) is a packaged language for running MCMC on Bayesian models
- OpenBUGS (and its predecessor WinBUGS)
  - Is a Windows package interface to the BUGS language
  - (JAGS is another related package accessible from R)
  - (STAN is an alternative package again runnable from R)
- OpenBUGS can be installed easily as per usual Windows packages.
- If you use MAC or LINUX then I suggest you use Wine emulator to run OpenBUGS.

OpenBUGS screen shot

- 

OpenBUGS demo.....

OpenBUGS

OpenBUGS

A demonstration with reference to disease mapping

Outline

- Introduction
- BUGS and OpenBUGS
- Graphical Models
- DoodleBUGS
- Example - Disease Mapping
- Summary

Introduction

- Bayesian Inference Using Gibbs Sampling
  - BUGS
- Analysis of Complex Models
- Bayesian Methods
- Markov Chain Monte Carlo Integration
  - Useful when no closed form exists

Classic BUGS

- Declarative Language
  - Similar to Splus
- Complex Statistical Models
  - Missing data
  - Measurement Error
  - No closed form for Likelihood
- Graphical Modelling
- Flexible compared to approximations

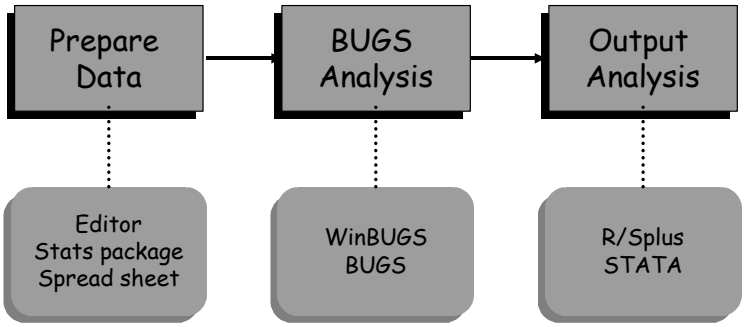
OpenBUGS

- Similar to Classic BUGS
  - Plus new methodological developments
- Graphical representation of model
  - DoodleBUGS
- Menu Control of session
- Cut and paste to other packages

BUGS and OpenBUGS

- No data management facility
  - Why reinvent the wheel?
- "Easy" interface with other packages
  - R and Splus
  - Stata (S. Bashir)
- Simple analysis of output

Working with BUGS



Graphical Models

- Complex multivariate probability models
  - Representation
  - Visualisation
- Graphs...
  - simplify complex models
  - communicate structure of the problem
  - provide basis for computation

OpenBUGS

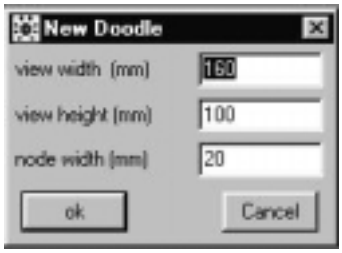
- BUGS language
- DoodleBUGS
- OpenBUGS
  https://www.mrc-bsu.cam.ac.uk/software/bugs/openbugs/

DoodleBUGS

- Start OpenBUGS
- Select "Doodle" from menu bar

DoodleBUGS - Basics

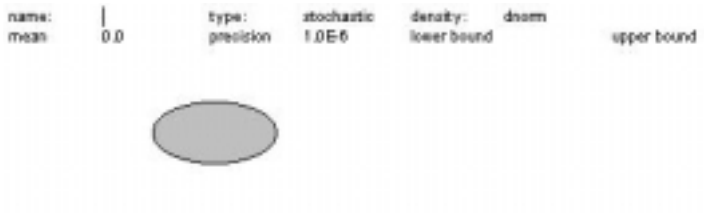
- Select "New…"



- Press "ok"
- You have a window to "Doodle" in.

Nodes

- Creating a node
  - Mouse click in Doodle Window



- Deleting a node: CTRL + Del

Node Types

- Nodes can be
  - Stochastic

  - Logical

  - Constant (rectangle)

Example - Simulation

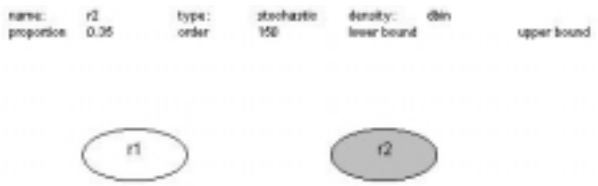
- Let
  - r1 ~ Bin (0.25, 250)
  - r2 ~ Bin (0.35, 150)
- Calculate p: common proportion for r1 & r2
- p = (r1+r2)/400
- Classical p = 0.2875

DoodleBUGS

- Start with r1 ~ Bin(0.25, 250) (stochastic node)

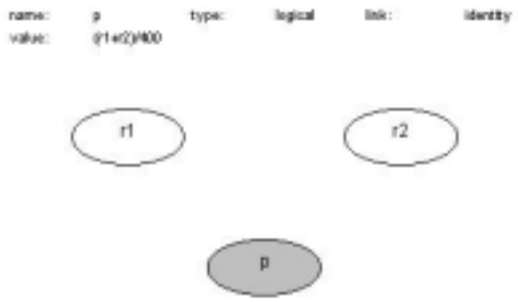
DoodleBUGS

- Add r2 ~ Bin(0.35, 150) (stochastic node)



Logical Nodes

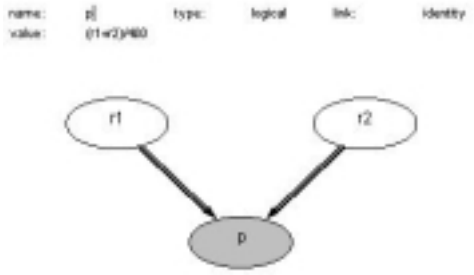
- Add p as a logical node



- To define a logical node click on "type" for choices.

Logical Functions

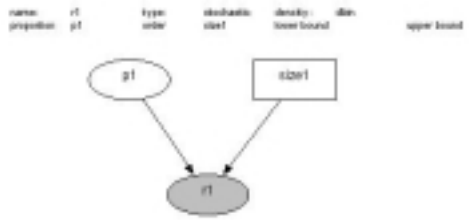
- Add "edges" for the logical relationship



- Whilst node p is highlighted, CTRL + click in "parent nodes" r1 and r2 (hollow arrows ✂ logical function)

Stochastic Nodes

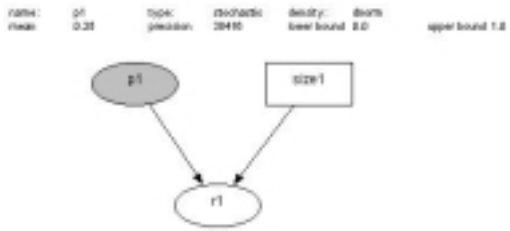
- Stochastic dependence
  - p1 ~ N(0.25, 0.000026) (i.e., p1 ~ [0.24, 0.26])
  - size1 = 250 (constant)



  - Single arrows for stochastic dependencies

Normal Distribution

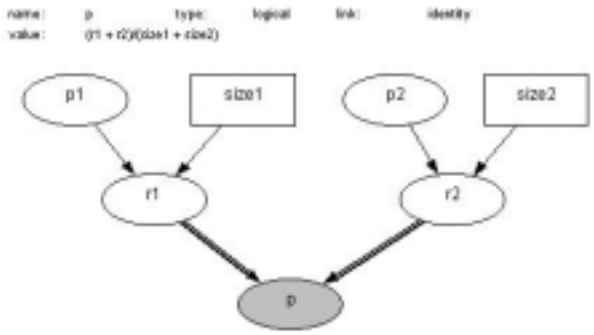
- Note the Normal distribution in BUGS is defined as N (mean, precision) where precision = 1/variance



  - Note that we can define upper and lower bounds so that the proportion is between 0 and 1.

DoodleBUGS Model

- Let us add these stochastic dependencies to our "logical" model



DoodleBUGS Model

- What is our model?
  - r1 ~ Bin (p1, size1)
  - p1 ~ N (0.25, 0.000026)
  - size1 = 250

  - r2 ~ Bin (p2, size2)
  - p2 ~ N (0.35, 0.000026)
  - size2 = 150

OpenBUGS Modelling

- Running our model in OpenBUGS
  - Create a New document
    - Menu bar - File - New

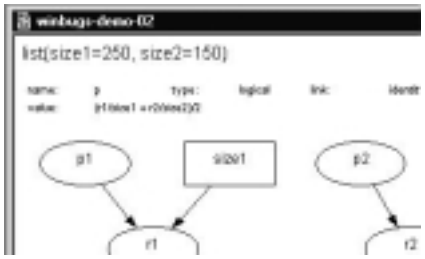
- A New document window will appear

OpenBUGS Document

- Select your Doodle from your Doodle Window
  - Menu bar - Edit - Select Document
- Copy your Doodle
  - Menu bar - Edit - Copy
- Paste it into your New Document
  - Menu bar - Edit - Paste

Model Data

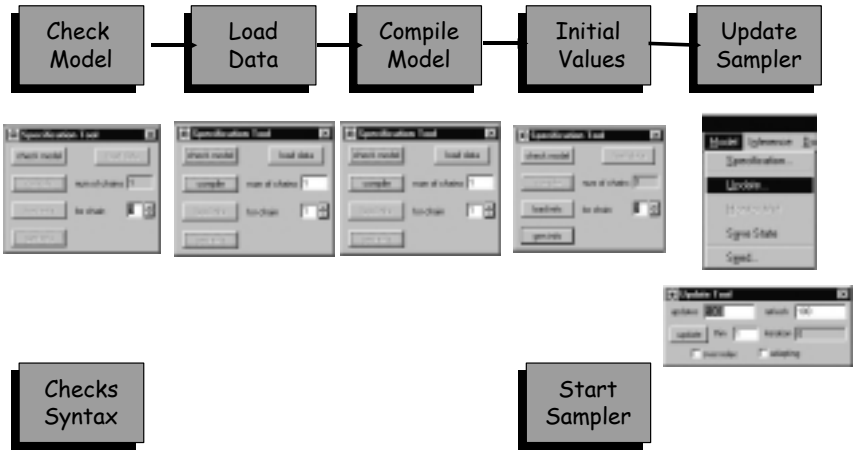
- Before running we need to give BUGS some data
  - Type **`list(size1=250, size2=150)`** at the top (or the bottom) of your new document.



Running BUGS

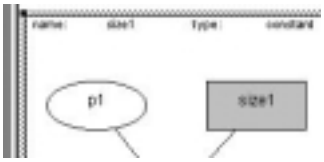
- Use "Specification..." from the "Model" option on Menu Bar to run BUGS

Running BUGS



Check Model

- Select the Doodle (note the hairy boarder)

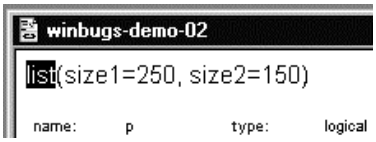


- Menu bar - Model - Check model
- Note the message in bottom left hand corner

model is syntactically correct

Load Data

- Highlight the word "list"



- Menu bar - Model - Data
- Bottom left hand corner

Compiling the Model

- Menu bar - Model - Compile
- Bottom left hand corner

Load Initial Values

- Menu bar - Model - Gen inits
- Bottom left hand side

Update the Model

- Menu bar - Model - Update

- 1000 MCMC updates to be carried out.

Burn In

- Model has been updated
- MCMC run did not store any data.
  - Used for the "burn in"
- Store values by "monitoring" them to
  - Draw inferences
  - Monitor MCMC run

Monitoring Nodes

- Monitoring p our parameter of interest
- Menu bar - Inference - Samples...

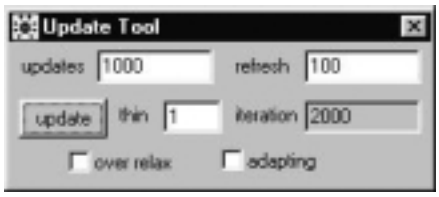
- Sample Monitor Tool

Monitoring Nodes

- Type name of node "p" to monitor
- Press "set"

Update & Monitor

- Update model again



- 1000 values "monitored" of the MCMC run for p

Summary Statistics

- Summary statistics
- Select "p" from the Sample Monitor Tool
- Press "stats" (Sample Monitor Tool)

Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p	0.2873	0.02296	6.687E-4	0.245	0.285	0.335	1001	1000

- Node statistics window

Summary Statistics

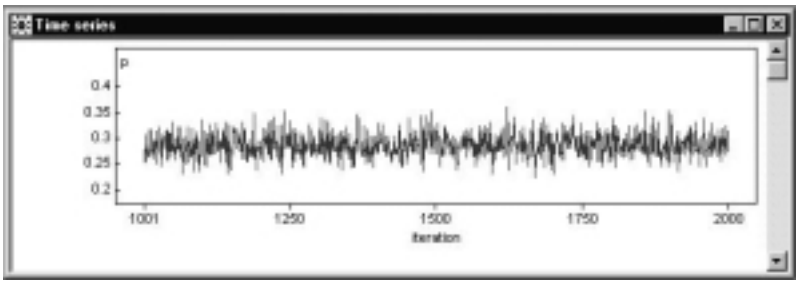
Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p	0.2873	0.02296	6.687E-4	0.245	0.285	0.335	1001	1000

- Mean = 0.2873
- Median = 0.285 (usually more stable)
- 95% credible interval (0.245, 0.335)
- MCMC run size 1000

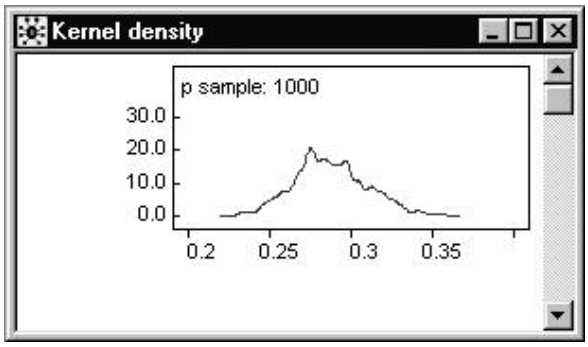
MCMC Time Series

- Press "History" in Sample Monitor Tool



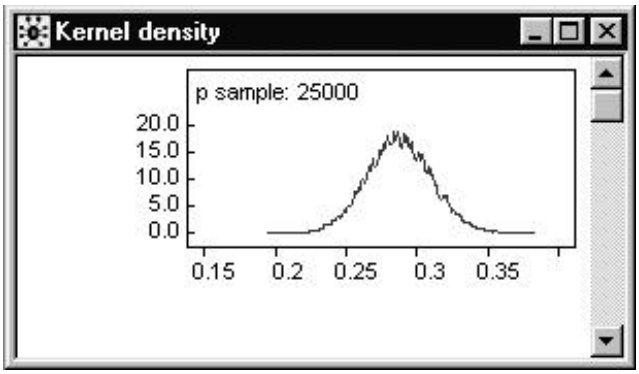
Kernel Density

- Press "Density" in the Sample Monitor Tool



Kernel Density

- Increase monitored values to 25,000



Plates

- Creating a plate
  - CTRL + mouse click in Doodle Window

  - Deleting a plate: CTRL + Del

Lips: spatial smoothing of cancer rates

The rates of lip cancer in 56 counties in Scotland have been analysed by Clayton and Kaldor (1987) and Breslow and Clayton (1993). The form of the data includes the observed and expected cases (expected numbers based on the population and its age and sex distribution in the county), a covariate measuring the percentage of the population engaged in agriculture, fishing, or forestry, and the "position" of each county expressed as a list of adjacent counties.

County	Observed cases	Expected cases	x (% in agric.)	SMR	Adjacent counties
1	9	1.4	16	652.2	5,9,11,19
2	39	8.7	16	450.3	7,10
...	...	...	...	...	...
56	0	1.8	10	0.0	18,24,30,33,45,55

We note that the extreme SMRs (Standardised Mortality Ratios) are based on very few cases. Breslow and Clayton initially consider a random-effects Poisson model allowing for over-dispersion, where O_i, E_i are the observed and expected cancer incidence in the ith county.

$$
\begin{aligned}
O_i &\sim \text{Poisson}(\mu_i) \\
\log \mu_i &= \log E_i + \alpha_1 x_i / 10 + b_i \\
b_i &\sim \text{Normal}(\alpha_0, \tau) \\
SMR_i &= 100\, \mu_i / E_i
\end{aligned}
$$

α_0, α_1 and τ are given independent "noninformative" priors. We note that the prior distribution for the b's can be easily shown to be equivalent to a model with an "intrinsic" prior

$$b_i \mid b_{j,\, j \neq i} \sim \text{Normal}(\bar{b}_{\backslash i}, N-1/N\, \tau)$$

where N is the number of counties, and $\bar{b}_{\backslash i} = 1/(N-1)\, \Sigma_{j \neq i}\, b_j$ is the average in all counties except i.

Spatial smoothing using an intrinsic prior

Breslow and Clayton consider a random-effects Poisson model allowing for over-dispersion and spatial correlation, using the conditional autoregressive (CAR) model of Besag (1974), which may be written

$$
\begin{aligned}
O_i &\sim \text{Poisson}(\mu_i) \\
\log \mu_i &= \log E_i + \alpha_0 + \alpha_1 x_i / 10 + \sigma b_i \\
b_i &\sim \text{Normal}(\bar{b}_i, n_i) \\
n_i &= \text{Number of neighbours of county } i \\
\bar{b}_i &= 1/n_i\ \Sigma_{j = \text{neighbour}(i)}\ b_j
\end{aligned}
$$

As with the exchangeable model, introducing the intrinsic prior means that a level term α_0 is not necessary in this model, although Breslow and Clayton (1993) retain this term due to their imposition of the constraint that $\Sigma_i\, b_i = 0$. Note that a standard noninformative prior for σ, such as a Gamma(0.001, 0.001) gives a full conditional which is not log concave; hence WinBUGS will use the slice sampling algorthim for this parameter. An exponential prior (or a Gamma with shape parameter > 1) will yield a log-concave full conditional however, allowing the use of adaptive rejection sampling.

Spatial model with intrinsic prior and hyperparameter

As in the `ice` example, we can introduce a precision parameter as a hyperparameter for the random effects.

$$
\begin{aligned}
O_i &\sim \text{Poisson}(\mu_i) \\
\log \mu_i &= \log E_i + \alpha_0 + \alpha_1 x_i / 10 + b_i \\
b_i &\sim \text{Normal}(\bar{b}_i, \tau_i) \\
n_i &= \text{Number of neighbours of county } i \\
\bar{b}_i &= 1/n_i \ \Sigma_{j = \text{neighbour}(i)} \ b_j \\
\tau_i &= n_i / \tau
\end{aligned}
$$

It can be shown that this model is equivalent to the improper prior

$p(b_1,...,b_I \mid \tau)$ proportional to $\tau^{I/2} \exp(-\tau / 2 \ \Sigma \ n_i \ b_i \ (b_i - \bar{b}_i))$

which provides the correct likelihood term for τ. Breslow and Clayton mention that this prior can also be expressed as

$p(b_1,...,b_I \mid \tau)$ proportional to $\tau^{I/2} \exp(-\tau / 4 \ \Sigma_{i \sim j} \ n_i \ b_i \ (b_i - b_j)^2)$

where "~" here represents "is a neighbour of".

```
model
{
   b[1:regions] ~ car.normal(adj[], weights[], num[], tau)
   b.mean <- mean(b[])
  for (i in 1 : regions) {
    O[i]       ~ dpois(mu[i])
    log(mu[i]) <- log(E[i]) + alpha0 + alpha1 * x[i] / 10 + b[i]
    SMRhat[i]  <- 100 * mu[i] / E[i]
  }
  alpha1  ~ dnorm(0.0, 1.0E-5)
  alpha0  ~ dflat()
   tau  ~ dgamma(rstar, dstar) sigma <- 1 / sqrt(tau)
}
```

Data ➔ click on one of the arrows to open data ←

Inits ➔ click on one of the arrows to open initial values ←

Results

A 1000 update burn in followed by a further 10000 updates gave the parameter estimates

node	mean	sd	MC error	2.5%	median	97.5%	sample
alpha1	0.3501	0.1338	0.006131	0.07618	0.3556	0.597	10000
sigma	0.756	0.1213	0.002739	0.5448	0.7471	1.021	10000

Summary

- BUGS is a power tool
- Bayesian Analysis
- Simulation Tool
- Graphical Models
- Doodle BUGS
- Simple representation of model
- Advanced disease mapping models can be fitted

(see e.g. Lawson and Clark (2002) *Stats in Med*, 21,359-370)

- Fairly easy to use!