Assessing population dynamics using WinBUGS

Olivier GIMENEZ

Joint work with Len THOMAS, Simon BONNER, Ruth KING and colleagues

What is the Bayesian Approach?

- The experimenter begins with some prior beliefs about the system (or not…).
- And then updates these beliefs on the basis of observed data.
- This updating procedure is based upon what is known as Bayes' Theorem:

$$\pi(\boldsymbol{\theta} \mid data) \propto L(data \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})$$

Posterior distribution of the parameters given the data

Likelihood of the data given the parameters

Prior distribution of the parameters

Summarising the Bayesian's Beliefs

- The posterior distribution (or the corresponding marginal distributions) are the best summaries of the parameters.
- However, point estimates and uncertainty intervals are often more interpretable.
- It is the process of summarising the posterior that is the source of the computational complexity of the Bayesian approach.
- The Markov chain Monte Carlo algorithm is a powerful tool for estimating these summary statistics of interest.

Bayesian Integration Made Easy

- Bayesian inference is based upon the estimation of posterior summaries such as the mean.
- These require integration of the posterior density e.g.

$$E(g(\theta)) = \int g(\theta)\pi(\theta \mid x)d\theta$$

- Expectations of this sort can be estimated by drawing samples $\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(n)}$ from the posterior distribution, π, and then calculating the sample mean.

$$E(g(\theta)) = \int g(\theta)\pi(\theta \mid x)d\theta \approx \frac{1}{n}\sum_{i=1}^{n} g(\theta^{(i)})$$

- This is known as *Monte Carlo Integration.*
- So, how do we generate a sample from π?

Bayesian modelling using MCMC

- Suppose that we can construct a Markov chain with stationary distribution equal to the posterior distribution of interest.
- Then, once the chain has converged, realisations can be regarded as a dependent sample from this distribution.
- There are now several ways of constructing these chains, where we only know the distribution upto proportionality.
- Metropolis-Hastings, Gibbs sampler, Metropolis-within-Gibbs, …

WinBUGS

- Bayesian analyses using MCMC algorithms typically involve a fair amount of computer programming, but…
- Programming isn't everyone's cup of tea!!.
- However, there are alternatives…
- WinBUGS is the most widely used package which allows a large number of possible models to be fitted.

A brief history

- ***1989***: Unix version called BUGS.
- ***1998***: first Windows version, WinBUGS was born.
- Initially developed by the ***MRC Biostatistics Unit*** in Cambridge and now joint work with ***Imperial College School of Medicine*** at St Mary's, London.
- **Win**dows **B**ayesian inference **U**sing **G**ibbs **S**ampling.
- *Software for the Bayesian analysis of complex statistical models using Markov chain Monte Carlo (MCMC) methods.*

Who?

Nicky Best
Imperial College Faculty of Medicine, London (UK)

Andrew Thomas
University of Helsinki, Helsinki (Finland)

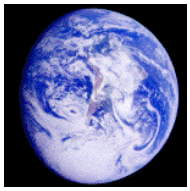
Wally Gilks
MRC Biostatistics Unit
Institute of Public Health
Cambridge (UK)

David Spiegelhalter
MRC Biostatistics Unit
Institute of Public Health
Cambridge (UK)

Freely downloadable from: http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml

A biological example throughout

White stork (Ciconia ciconia)



Demographic components
(fecundity, breeding success, survival, etc…)

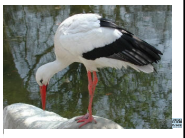
Climate
(Temperature, rainfall, etc…)

WinBUGS & Linear Regression

Y
Number of chicks per pairs

2.55		15.1	67
1.85		13.3	52
2.05		15.3	88
2.88		13.3	61
3.13		14.6	32
2.21		15.6	36
2.43		13.1	72
2.69		13.1	43
2.55		15.0	92
2.84		11.7	32
2.47		15.3	86
2.69		14.4	28
2.52		14.4	57
2.31		12.7	55
2.07		11.7	66
2.35		11.9	26
2.98		15.9	28
1.98		13.4	96
2.53		14.0	48
2.21		13.9	90
2.62		12.9	86
1.78		15.1	78
2.30		13.0	87

T Temp. May (°C)
R Rainf. May (mm)



WinBUGS & Linear Regression

1. Do temperature and rainfall affect the number of chicks?
2. Regression model:

$$Y_i = \alpha + \beta_r R_i + \beta_t T_i + \varepsilon_i , \quad i=1,\dots,23$$
$$\varepsilon_i \text{ i.i.d.} \sim N(0,\sigma^2)$$

$$\mathbf{Y_i \text{ i.i.d.} \sim N(\mu_i,\sigma^2), \quad i=1,\dots,23}$$
$$\mathbf{\mu_i = \alpha + \beta_r R_i + \beta_t T_i}$$

3. Estimation of parameters: α, β_r, β_t, σ
4. Frequentist inference uses t-tests.

Linear Regression using Frequentist approach

Y
Number of chicks per pairs

2.55		15.1	67
1.85		13.3	52
2.05		15.3	88
2.88		13.3	61
3.13		14.6	32
2.21		15.6	36
…		…	…
2.30		13.0	87

T Temp. May (°C)
R Rainf. May (mm)

$$Y = 2.451 + 0.031\ T - 0.007\ R$$

	Estimate	Std. Error	t value	Pr(>|t|)
temperature	0.031069	0.054690	0.568	0.57629
rainfall	-0.007316	0.002897	-2.525	0.02011 *

Linear Regression using Frequentist approach

Y
Number
of chicks
per pairs

2.55
1.85
2.05
2.88
3.13
2.21
...
2.30

15.1	67
13.3	52
15.3	88
13.3	61
14.6	32
15.6	36
...	...
13.0	87

T Temp. May (°C)
R Rainf. May (mm)

$$Y = 2.451 + 0.031\ T - 0.007\ R$$

	Estimate	Std. Error	t value	Pr(>|t|)
temperature	0.031069	0.054690	0.568	0.57629
rainfall	-0.007316	0.002897	-2.525	0.02011 *

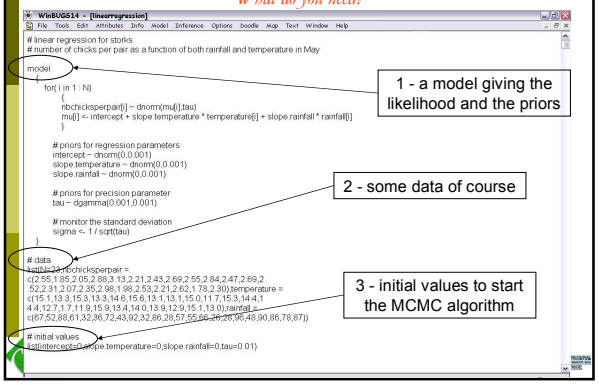
⇒ ***Influence of Rainfall only***

Analysis using WinBUGS

- Three basic parts of a WinBUGS file:
  - 1. Model specification:
    - BUGS language: Similar to S-Plus/R.
    - Graphical language: Using DoodleBUGS
      - Illustrate the structure of the problem
      - But, some feature of BUGS language cannot be represented by DoodleBUGS.
      - See manual for further details
  - 2. Data
  - 3. Initial values for unknowns θ

Running WinBUGS

What do you need?



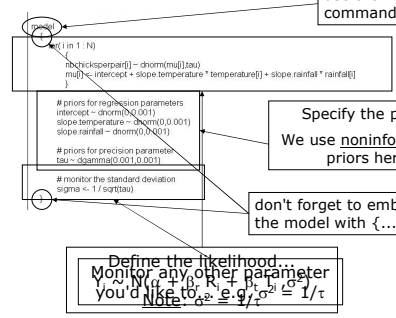
1 - a model giving the likelihood and the priors

2 - some data of course

3 - initial values to start the MCMC algorithm

Running WinBUGS

The model



use the WinBUGS command 'model'

Specify the priors

We use noninformative priors here

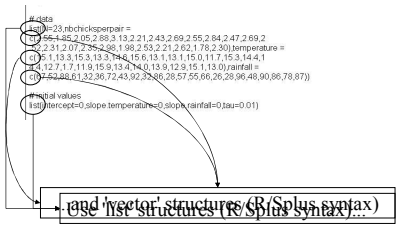
don't forget to embrace the model with {...}

Define the likelihood...
$Y_i \sim N(\alpha + \beta_T R_i + \beta_T t_{bi}, \sigma^2)$
You'd like to... e.g. $\sigma^2 = 1/\tau$
Note: $\sigma = 1/\tau$

Monitor any other parameter

Running WinBUGS

Data and initial values

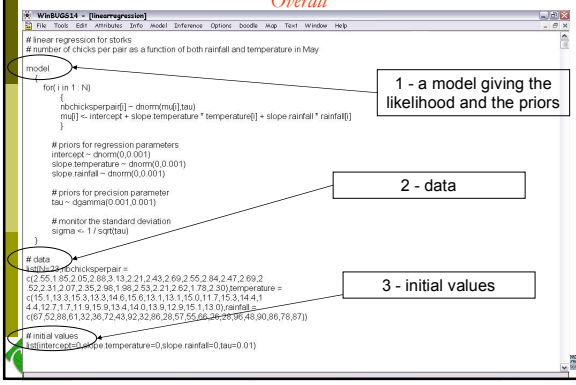


→ Use 'list' structures (R/Splus syntax)...

and 'vector' structures (R/Splus syntax)

Running WinBUGS

Overall



1 - a model giving the likelihood and the priors

2 - data

3 - initial values

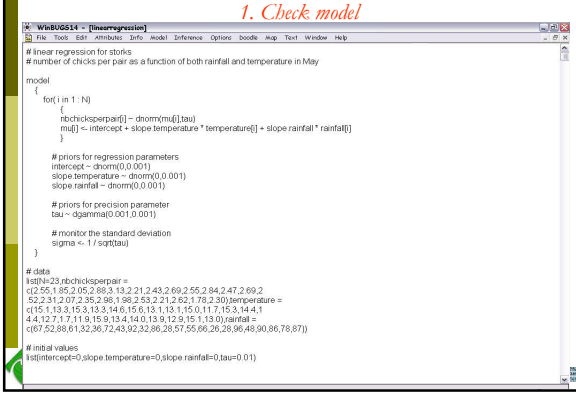
Running WinBUGS

At last!!

- **Initialization**
  - 1 - check model
  - 2 - load data
  - 3 - compile model
  - 4 - load initial values
- **Run MCMC**
  - 5 - generate burn-in values
  - 6 - parameters to be monitored
  - 7 - perform the sampling to generate posteriors
- **Posterior summary**
  - 8 - check convergence
  - 9 - display results

Running WinBUGS

1. Check model

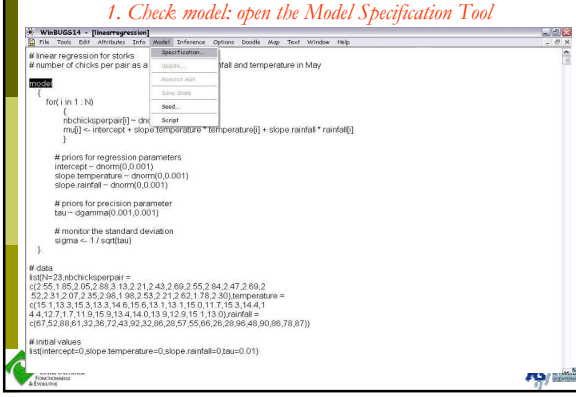


Running WinBUGS

1. Check model: highlight 'model'

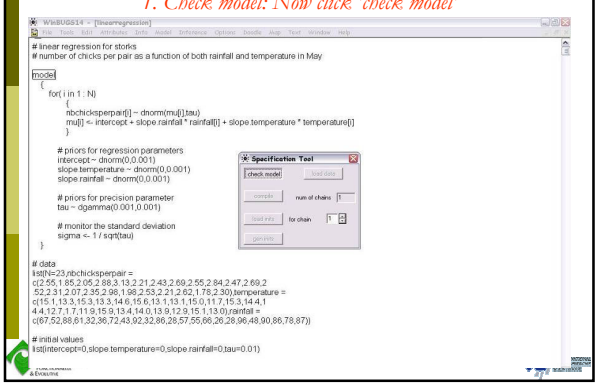
Running WinBUGS

1. Check model: open the Model Specification Tool



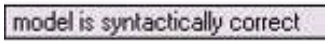
Running WinBUGS

1. Check model: Now click 'check model'



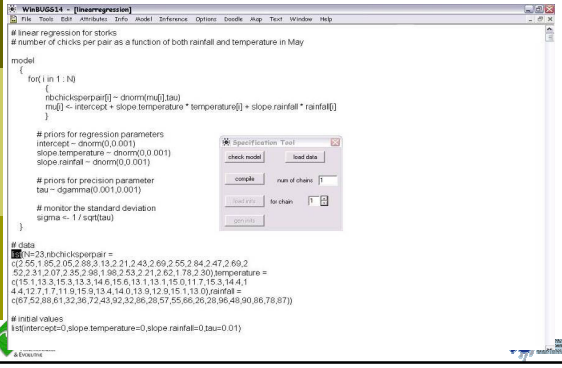
Running WinBUGS

1. Check model: Watch out for the confirmation at the foot of the screen



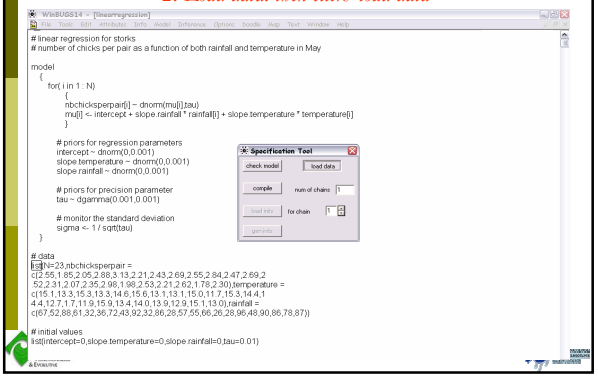
Running WinBUGS

2. Load data: Now highlight the 'list' in the data window



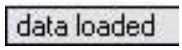
Running WinBUGS

2. Load data: then click 'load data'



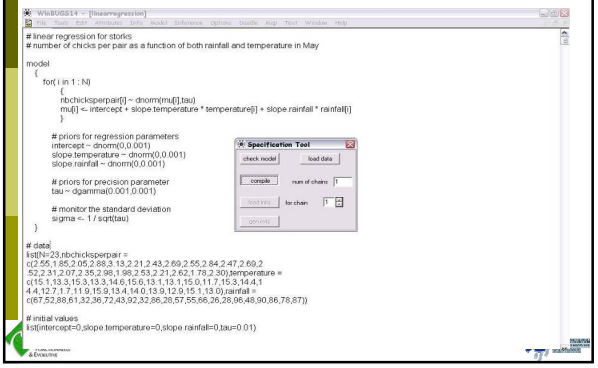
Running WinBUGS

2. Load data: watch out for the confirmation at the foot of the screen



Running WinBUGS

3. Compile model: Next, click 'compile'

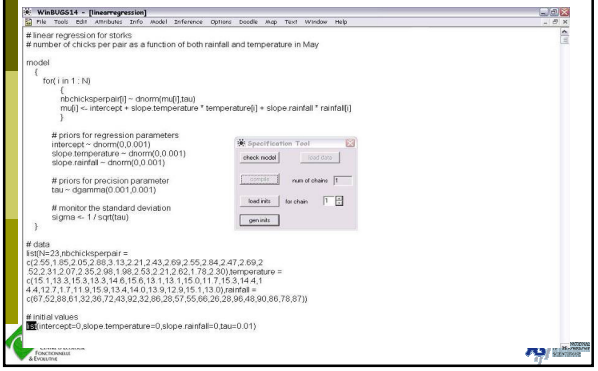


Running WinBUGS

3. Compile model: watch out for the confirmation at the foot of the screen

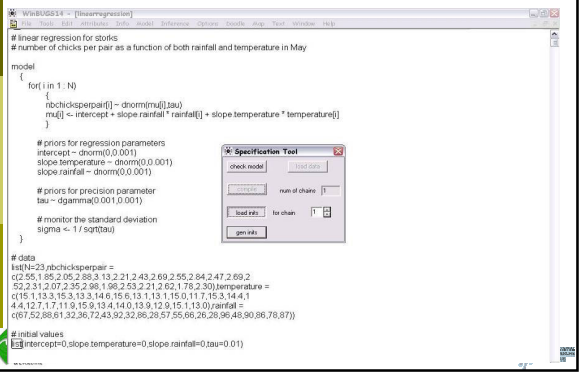
Running WinBUGS

4. Load initial values: highlight the 'list' in the data window



Running WinBUGS

4. Load initial values: click 'load inits'

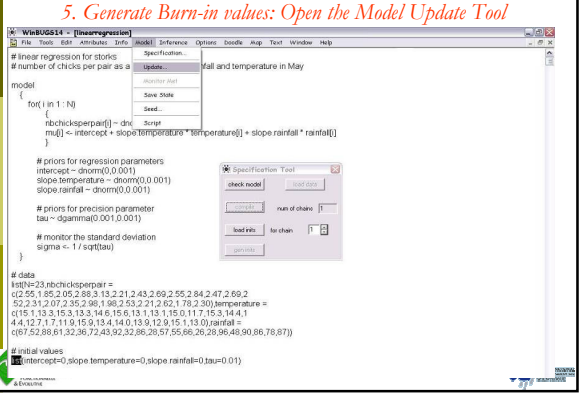


Running WinBUGS

4. Load initial values: watch out for the confirmation at the foot of the screen

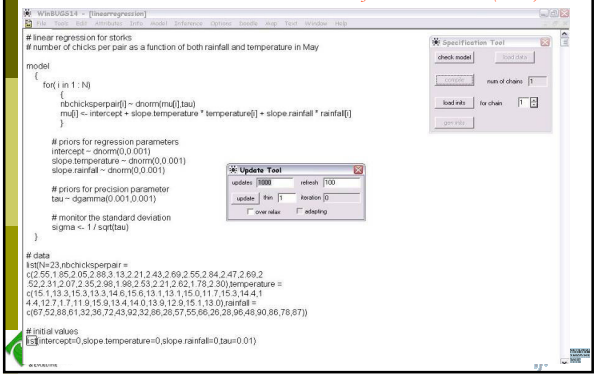
Running WinBUGS

5. Generate Burn-in values: Open the Model Update Tool



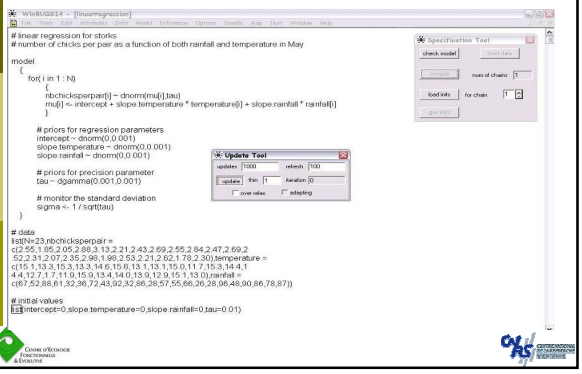
Running WinBUGS

5. Generate Burn-in values: Give the number of burn-in iterations (1000)



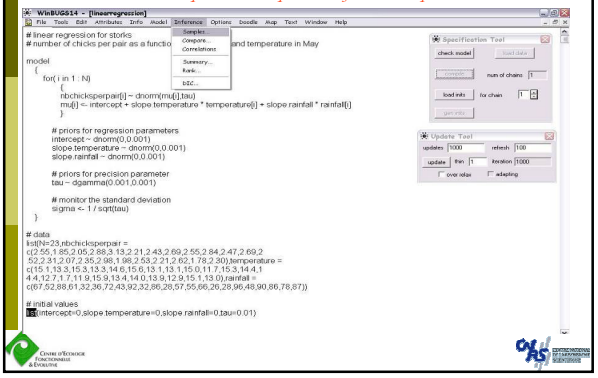
Running WinBUGS

5. Generate Burn-in values: click 'update' to do the sampling



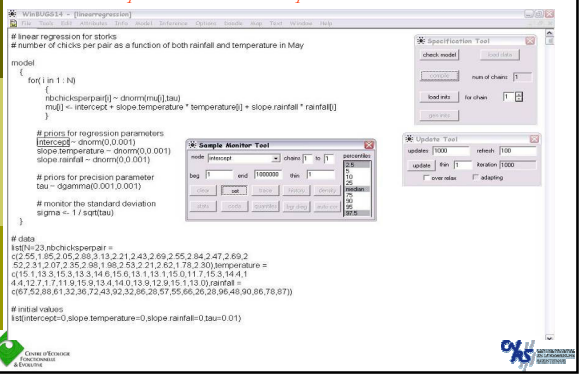
Running WinBUGS

6. Monitor parameters: open the Inference Samples Tool



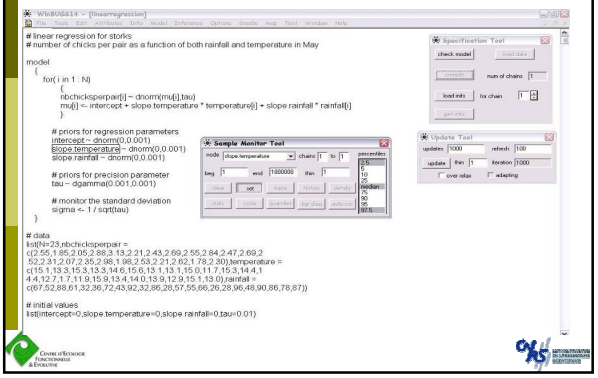
Running WinBUGS

6. Monitor parameters: Enter 'intercept' in the node box and click 'set'



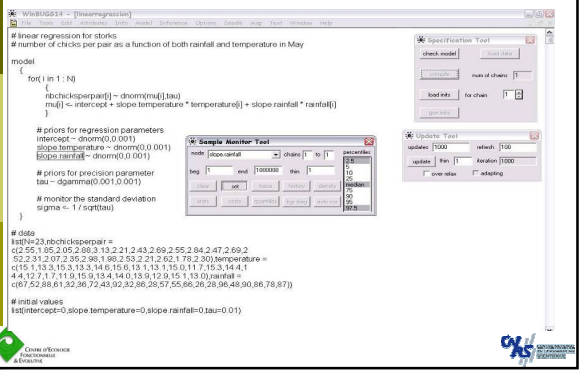
Running WinBUGS

6. Monitor parameters: Enter 'slope_temperature' in the node box and click 'set'



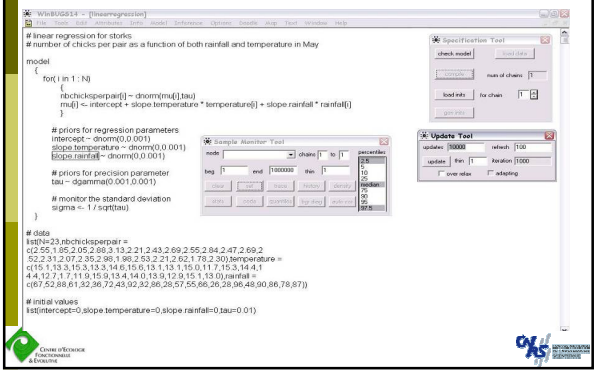
Running WinBUGS

6. Monitor parameters: Enter 'slope_rainfall' in the node box and click 'set'



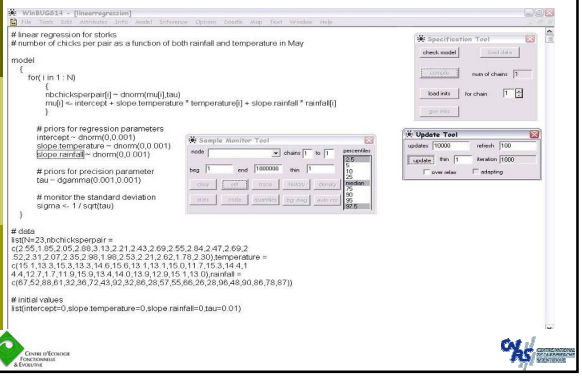
Running WinBUGS

7. Generate posterior values: enter the number of samples you want to take (10000)



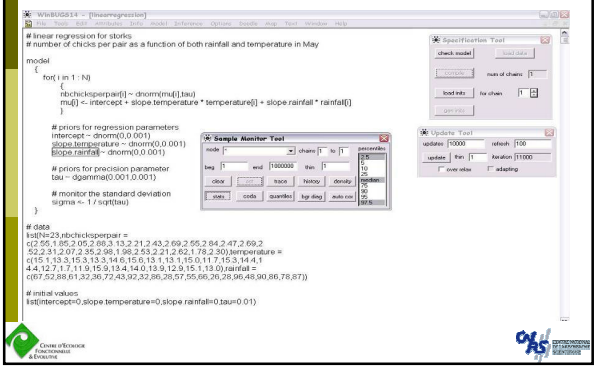
Running WinBUGS

7. Generate posterior values: click 'update' to do the sampling



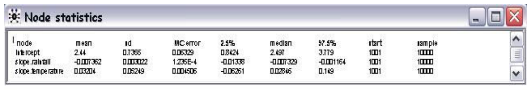
Running WinBUGS

8. Summarize posteriors: Enter '' in the node box and click 'stats'*



Running WinBUGS

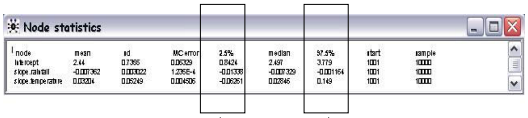
8. Summarize posteriors: mean, median and credible intervals



node	mean	sd	MC error	2.5%	median	97.5%	start	sample
intercept	2.44	0.7365	0.06329	0.8424	2.497	3.779	1001	10000
slope.rainfall	-0.007362	0.002022	1.23E-4	-0.01308	-0.007329	-0.001164	1001	10000
slope.temperature	0.03204	0.05249	0.004505	-0.06351	0.02846	0.149	1001	10000

Running WinBUGS

8. Summarize posteriors: 95% Credible intervals



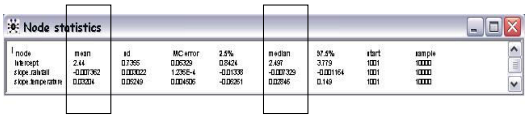
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
intercept	2.44	0.7365	0.06329	0.8424	2.497	3.779	1001	10000
slope.rainfall	-0.007362	0.002022	1.23E-4	-0.01308	-0.007329	-0.001164	1001	10000
slope.temperature	0.03204	0.05249	0.004505	-0.06351	0.02846	0.149	1001	10000

tell us the same story

	Estimate	Std. Error	t value	**Pr(>|t|)**
temperature	0.031069	0.054690	0.568	**0.57629**
rainfall	-0.007316	0.002897	-2.525	**0.02011 ***

Running WinBUGS

8. Summarize posteriors: Mean and Median



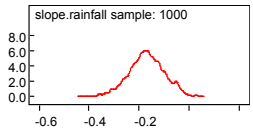
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
intercept	2.44	0.7365	0.06329	0.8424	2.497	3.779	1001	10000
slope.rainfall	-0.007362	0.002022	1.23E-4	-0.01308	-0.007329	-0.001164	1001	10000
slope.temperature	0.03204	0.05249	0.004505	-0.06351	0.02846	0.149	1001	10000

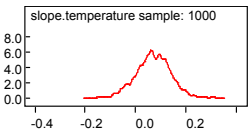
tell us the same story

	Estimate	Std. Error	t value	Pr(>|t|)
temperature	**0.031069**	0.054690	0.568	0.57629
rainfall	**-0.007316**	0.002897	-2.525	0.02011 *

Re-running WinBUGS

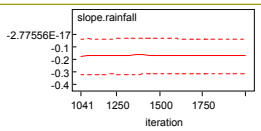
8. Summarize posteriors: click 'density' in Sample Monitor Tool pop-up window

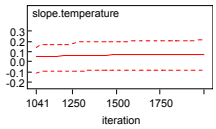




Re-running WinBUGS

8. Summarize posteriors: click 'quantiles' in Sample Monitor Tool pop-up window





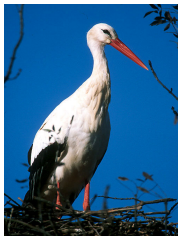
Re-running WinBUGS

1,2,...7, and 8. Summarize posteriors: further options...

- Click 'history' to get trace plots and 'autocor' to get autocorrelation diagnostics
- Run several chains with over-dispersed initial values to use the Brooks-Gelman-Rubin statistic to check convergence (should be close to 1).
- Click 'coda' to produce lists of data suitable for external treatment via the Coda R package.

Analysing mark-recapture data using WinBUGS

- We focus on the Cormack-Jolly-Seber model.
- We consider data relating to population of White storks breeding in Baden Württemberg (Germany).
- Interest lies in estimating *adult survival rates*.
- Mark-recapture data from 1956-71 are available.



Analysing mark-recapture data using WinBUGS

- Data: m-array

Year of release (19-)	Number released	Year of first recapture (19-)											
		57	58	59	60	61	62	63	64	65	66	67	68
56	26	19	2	0	0	0	0	0	0	0	0	0	0
57	50	0	33	3	0	0	0	0	0	0	0	0	0
58	53	0	0	35	4	0	0	0	0	0	0	0	0
59	69	0	0	0	42	1	0	0	0	0	0	0	0
60	73	0	0	0	0	42	1	0	0	0	0	0	0

…

⋮

Analysing mark-recapture data using WinBUGS

- The parameters are:
  - probability ϕ_i that an individual survives to occasion *i+1* given that it is alive at time *i*;
  - probability p_j that an individual is recaptured at time *j*.

Analysing mark-recapture data using WinBUGS

- Conditioning on the releases and assuming independence among cohorts, the likelihood is a product of multinomial probability distributions corresponding to each row of the m-array.
- The probability of the m-array cells are complex nonlinear functions of the survival and detection probabilities.
- For example:

$$\Pr(m_{35}|R_3) = \phi_3(1-p_4)\phi_4 p_5$$

Analysing mark-recapture data using WinBUGS

- CJS likelihood coded in WinBUGS

```
# Define the product-multinomial likelihood
# R[i] is the number of birds released in year i
# matrix q[,] contains the m-array cell probabilities

for (i in 1:ni) {
m[i,1:(nj+1)] ~ dmulti(q[i,],R[i])
}
```

- Cell probabilities coded in a complex way…

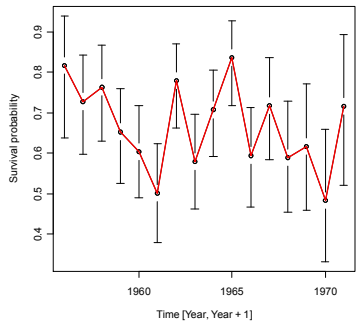
Analysing mark-recapture data using WinBUGS

- Priors

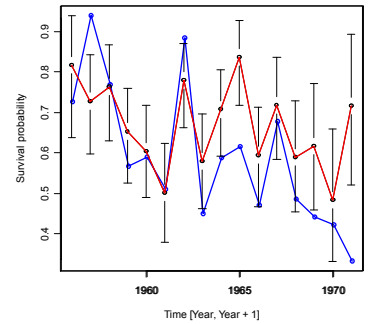
```
# Define the priors for phi
# Time-dependent parameters
# Recall that Beta(1,1) is equivalent to U(0,1)
for (i in 1:ni) {
phi[i] ~ dbeta(1,1)
}

# Define priors for p
# Constant parameters
p ~ dbeta(1,1)
```

Time-dependent survival probabilities

Posterior medians and 95% credible intervals



Time-dependent survival probabilities
Rainfall at Kita station (Sahel)



Incorporating covariates

- Use the logistic link function:

$$\text{logit}(\phi) = \beta_0 + \beta_1 x$$

- Minor modifications in the WinBUGS code

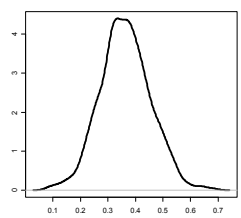
```
# Define the survival probability as a function of the covariate
# with random effects
for (i in 1:ni) {
logit(phi[i]) <- beta[1] + beta[2] * cov[i]
}

# prior for the regression parameters
for (j in 1:2){beta[j] ~ dnorm(0,1.0E-6)
```

Incorporating covariates: Results

Numerical summaries

	mean	sd	2.5%	25%	50%	75%	97.5%
beta[1]	0.68	0.07	0.54	0.63	0.68	0.73	0.82
beta[2]	0.37	0.09	**0.20**	0.30	0.36	0.42	**0.55**

Posterior distribution



Incorporating covariates: Issues

- Use the logistic link function:

$$\text{logit}(\phi) = \beta_0 + \beta_1 x$$

  - +ve: survival estimates in [0,1]
  - -ve 1: variation completely determined by covariate
  - -ve 2: no missing values
  - -ve 3: linear or quadratic relationship
  - -ve 4: variable selection?

1. Variation completely determined by covariate

- Incorporate random effects:

$$\text{logit}(\phi_i) = \beta_0 + \beta_1 x_i + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

- A random effects model has σ^2 as the parameter to be estimated, with some associated prior $p(\sigma^2) \sim \Gamma^{-1}(0.01,0.01)$.
- They are imputed within the MCMC algorithm and integrated out.

1. Random effects in WinBUGS

```
# Define the survival probability as a function of the covariate
# with random effects
for (i in 1:ni) {
logit(phi[i]) <- beta[1] + beta[2] * cov[i] + epsilon[i]
epsilon[i] ~ dnorm(0,taueps)
}

# prior for the precision
taueps ~ dgamma(0.01,0.01)

# prior for the regression parameters
for (j in 1:2){beta[j] ~ dnorm(0,1.0E-6)}

# monitor the variance (= 1/precision)
vareps <- 1/taueps
```

Further issues

- 2. Missing values
- 3. Nonparametric modelling
- 4. Variable selection using Reversible Jump MCMC

See paper for details (including the code)

2. Missing values

- In practice, we view the missing covariates as random variables to which we can assign a probability distribution.
- We assume that the change in the covariate between adjacent years is normally distributed with the same mean and variance for all years:

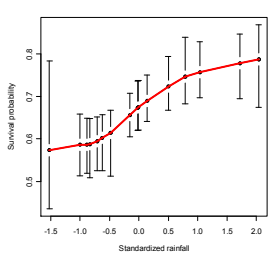
$$x_i \sim N(x_{i-1} + \mu, \sigma_x^2)$$
$$\mu \sim N(0, 10^6)$$
$$\sigma_x^2 \sim \Gamma^{-1}(0.01, 0.01)$$

2. Missing values: results

- We considered 1 and 5 missing values.
- Little impact on survival probability estimates.
- But, significant change in the inference for the regression coefficients.
- Most importantly, with 5 missing values, the 95% credible interval contains 0 which brings the effect of pluviometry on survival into doubt.

3. Nonparametric modelling

- Positive relationship survival/rainfall

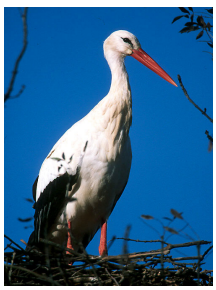


- More in the Bayesian short course & in the Population Dynamics session

4. Recent developments in WinBUGS

Model selection using RJMCMC

- Impact of climate variation (*rainfall*) in the wintering area on population dynamics (*adult survival rates*).
- Amount of rainfall each year from 10 weather stations located around the Sahel region.
- Identifying the given rainfall locations that explain the adult survival rates.



4. White stork survival vs. rainfalls

- Survival vs. multiple covariates on a logit scale:

$$\text{logit}(\phi_i) = \beta_0 + \beta_1 x_i^1 + \ldots + \beta_{10} x_i^{10} + \varepsilon_i$$

- Which rainfalls explain survival variation?
- *Method*: what β's are non-zero?

4. Model Selection

- In the classical framework, likelihood ratio tests or information criterion (e.g. AIC) are often used.
- There is a "similar" Bayesian statistic – the DIC.
- This is programmed within WinBUGS: *Jump* adds-on, http://www.winbugs-development.org.uk/rjmcmc.html
- There are 10 possible covariates, hence a total of 2^{10} possible models (1024!).
- Reversible Jump MCMC is another way of dealing with the issue of model discrimination.

4. Results: posterior model prob.

Model structure	Posterior model probability
0001000000	0.518
0000000000	0.0763
0001010000	0.0474
0001100000	0.0381
0001000001	0.0309
1001000000	0.0255

Pluviometry at Kita station

Conclusions

- Bayesian inference based on posterior distribution, derived from prior x likelihood via Bayes theorem.
- Bayesian inference through Monte Carlo integration: samples of posterior distribution obtained from MCMC simulation.
- WinBUGS allows MCMC algorithms to be implemented.
- Can be called from other programs (SAS, R, MATLAB).
- Great potential in population dynamics studies:
  - Random effects, spline smoothing, missing values, …
  - Further examples in our paper: Distance Sampling and State-Space modelling
- Bayesian session!!

C'est la fin…

Beware of sunburn