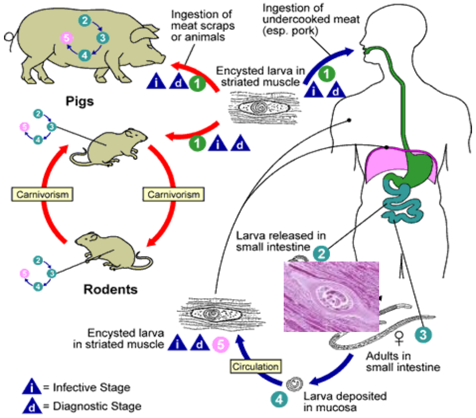
Introduction to Bayesian

EXERCISES WITH WINBUGS

María J. Vilar
Dep. of Food Hygiene and Environmental Health



- Software used to generate a random sample
- BUGS "Bayesian inference Using Gibbs Sampling"
- Available from the BUGS Project website
- Can be run from other software packages (e.g., R, Matlab)
- Free of charge

Based in the simulation technique MCMC (Markow Chain Monte Carlo Methods)

Menus

- Basic operations: File, Window, Help
- Editing operations: Tools, Edit, Text
- MCMC functions: Info, Model, Inference, Options
- Specialized operations: Doodle, Maps

Code structure

Enclose all commands ⇒

```
Model {
……………………
}
```

Model parameters or nodes

Stochastic nodes ⇒ Variable ~ distribution (parameter1, parameter2,…)

X ~ dbin(p, n)

Deterministic nodes ⇒ Y <- parameter.function (x)   Y <- log(x)

Vector v of length n and elements v_i (for i=1, 2, …,n) ⇒ v[ ] and elements v[i]

Multidimensional nodes ⇒ for (i in I1 : I2) { ..   for (i in 1:n) { ..

```
Model
for (i in 1:n) {
X[i] ~ distribution ( parameter1[i] , parameter2 )
}
```

Distribution name	WinBUGS syntax	Probability or density function $f(x)$	Mean	Variance
		Discrete distributions		
(1) Bernoulli	`x ~ dbern(p)`	$p^x(1-p)^{1-x}$	p	$p(1-p)$
(2) Binomial	`x ~ dbin(p, n)`	$n!p^x(1-p)^{n-x}/[x!(n-x)!]$	np	$np(1-p)$
(3) Categorical	`x ~ dcat(p[])`	p_x	$\sum_{x=1}^{K} xp_x$	$\sum_{x=1}^{K} [x-E(x)]^2 p_x$
(4) Negative binomial	`x ~ dnegbin(p, r)`	$(x+r-1)!p^r(1-p)^x/[x!(r-1)!]$	$r(1-p)/p$	$r(1-p)/p^2$
(5) Poisson	`x ~ dpois(lambda)`	$\exp(-\lambda)\lambda^x/x!$	λ	λ
		Continuous distributions		
(6) Beta	`x ~ dbeta(a, b)`	$\Gamma(a+b)x^{a-1}(1-x)^{b-1}/[\Gamma(a)\Gamma(b)]$	$a/(a+b)$	$ab/[(a+b)^2(a+b+1)]$
(7) Chi-squared	`x ~ dchisqr(k)`	See gamma$(k/2, \frac{1}{2})$	k	$2k$
(8) Double exponential	`x ~ ddexp(mu, tau)`	$\frac{1}{2}\tau\exp(-\tau\lvert x-\mu\rvert)$	μ	$\sqrt{2}/\tau$
(9) Exponential	`x ~ dexp(lambda)`	$\lambda e^{-\lambda x}$	$1/\lambda$	$1/\lambda^2$
(10) Gamma	`x ~ dgamma(a, b)`	$b^a x^{a-1}e^{-bx}/\Gamma(a)$	a/b	a/b^2
(11) Generalized gamma	`x ~ gen.gamma(a, b, r)`	$rb(bx)^{ra-1}\exp\left[-(bx)^r\right]/\Gamma(a)$	$\Gamma(a+1/r)/[b\Gamma(a)]$	$[\Gamma(a+2/r^2)\Gamma(a)-\Gamma(a+1/r)^2]/[\lambda\Gamma(a)]^2$
(12) Log-normal	`x ~ dlnorm(mu, tau)`	$\sqrt{\tau/(2\pi)}x^{-1}\exp\left[-\tau/2(\log x-\mu)^2\right]$	$e^{\mu+1/(2\tau)}$	$(e^{1/\tau}-1)e^{2\mu+1/\tau}$
(13) Logistic	`x ~ dlogis(mu, tau)`	$\tau e^{\tau(x-\mu)}\left[1+e^{\tau(x-\mu)}\right]^{-2}$	μ	$\pi^2/[3\tau^2]$
(14) Normal	`x ~ dnorm(mu, tau)`	$\sqrt{\tau/(2\pi)}\exp[-\tau(x-\mu)^2/2]$	μ	$1/\tau$
(15) Pareto	`x ~ dpar(a, c)`	$ac^a x^{-a-1}$	$ab/(a-1)$	$ab^2/[(a-1)^2(a-2)]$
(16) Student's t	`x ~ dt(mu, tau, v)`	$\Gamma[(v+1)/2]\sqrt{\tau/(2\pi)}[\Gamma(v/2)]^{-1} \times \left[1+\tau v^{-1}(x-\mu)^2\right]^{-(v+1)/2}$	μ	$v\tau^{-1}/(v-2)$
(17) Uniform	`x ~ dunif(a, b)`	$1/(b-a)$	$a/(a+b)$	$\frac{1}{12}(b-a)^2$
(18) Weibull	`x ~ dweib(v, lambda)`	$v\lambda x^{v-1}\exp(-\lambda x^v)$	$\lambda^{-1/v}\Gamma\left(1+v^{-1}\right)$	$\left[\Gamma\left(1+2v^{-1}\right)-\Gamma\left(1+v^{-1}\right)^2\right]\lambda^{-2/v}$

WinBUGS Syntax	Function	Description
1. `abs(x)`	$\|x\|$	Absolute value
2. `cloglog(x)`	$\log(-\log(1-x))$	Complementary log–log function
3. `cos(x)`	$\cos(x)$	Cosine function
4. `cut(x)`		Posterior of x is not updated by the likelihood
5. `equals(x1, x2)`	$f(x_1, x_2) = 1$ when $x_1 = x_2$; $= 0$ otherwise	Binary indicator function for equal nodes
6. `exp(x)`	e^x	Exponent value
7. `inprod(v1[], v2[])`	$\sum_i v_{1i} v_{2i}$	Inner product of two vectors
8. `interp.lin(x, v1[], v2[])`	$v_{2i} + (v_{2,i+1} - v_{2i}) \times (x - v_{1i})/(v_{1,i+1} - v_{1i})$	Interpolation line
8. `inverse(M[,])`	$\boldsymbol{A}^{-1}$	Inverse of a symmetric positive-definite matrix
9. `log(x)`	$\log(x)$	Logarithm (ln)
10. `logdet(M[,])`	$\log\|\boldsymbol{A}\|$	Logarithm of the determinant of a symmetric positive-definite matrix
11. `logfact(k)`	$\log(k!)$	Log factorial function of an integer
12. `loggam(x)`	$\log(\Gamma(x))$	Log gamma function
13. `logit(x)`	$\log \frac{x}{1-x}$	Logit function
14. `max(x1, x2)`	$\max(x_1, x_2)$	Maximum of two values
15. `mean(v[])`	$\overline{v} = \sum_{i=1}^{n} v_i/n$, where n is the length of vector v	Sample mean
16. `min(x1, x2)`	$\min(x_1, x_2)$	Minimum of two values
17. `phi(x)`	$P(X \leq x),\ X \sim N(0,1)$	CDF of standardized normal
18. `pow(x, z)`	x^z	Power function
19. `sin(x)`	$\sin(x)$	Sine function
20. `sqrt(x)`	$\sqrt{x}$	Square root
21. `rank(v[], k)`	$\sum_i I(v_i \leq v_k)$, where $I(z) = 1$ if z true and 0 otherwise	Rank of s component of a vector
22. `ranked(v[], k)`	$v_i : \sum_s I(v_s \leq v_i) = k$	Element of a vector with rank s
23. `round(x)`		Round to the closest integer
24. `sd(v[])`	$\sqrt{\sum_{i=1}^{n}(v_i - \overline{v})^2/(n-1)}$	Sample standard deviation
25. `step(x)`	$f(x) = 1$ when $x \geq 0$; 0 otherwise	Binary indicator function of positive nodes
26. `sum(v[])`	$\sum_i v_i$	Sum of a vector's components
27. `trunc(x)`		Truncation to the closest smaller than x integer

Data

Rectangular format

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

List format

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

Example

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

```
list( n=4, p=5,
      datamatrix=structure(
                 .Data=c( 12, 2, 0.3, 1, 20, 23,  5, 0.2, 2, 21,
                          54, 9, 0.9, 1, 23, 32, 11, 2.1, 2, 20),
                 .Dim=c(4,5) )
```

Data

Initial values

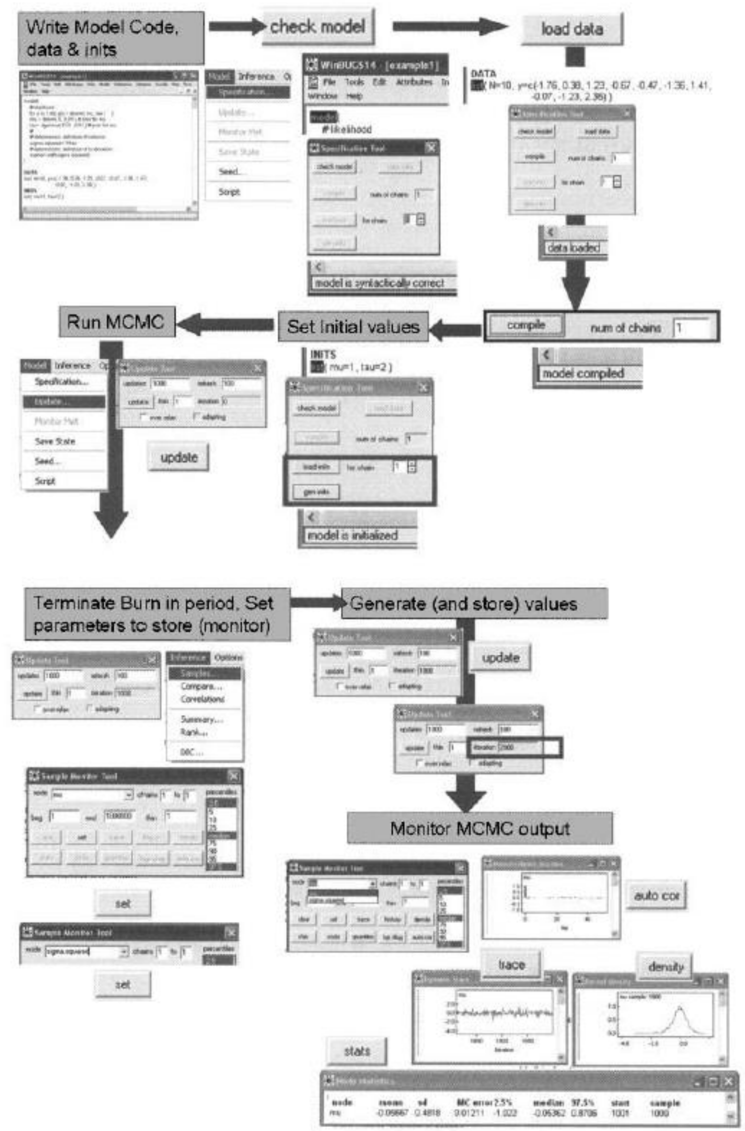
- Initiate MCMC
- Same format that data
- For all stochastic nodes

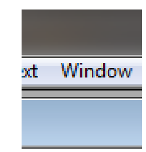
Missing values : NA

Compiling and simulating values

1. Open model specification tool
2. Check the model's syntax
3. Load data
4. Compile model
5. Set initial values
6. Run the MCMC algorithm

Example of a complete model specification

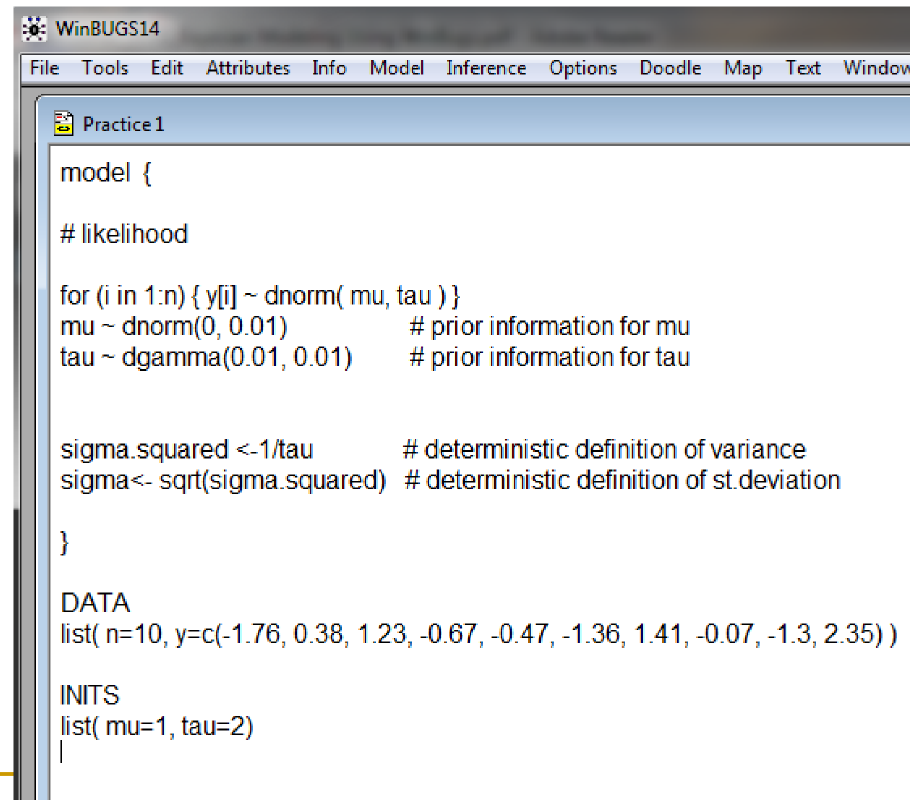




Compiling and simulating values

1. Open model specification tool
2. Check the model's syntax
3. Load data
4. Compile model
5. Set initial values
6. Run the MCMC algorithm

Example of a complete model specification



<u>Example:</u> Model prevalence (frequentist vs. stochastic)

Frequentist approach

Assume that from a large herd of dairy cows, n=200 animals are randomly sampled and x=26 animals test positive for pathogen X.

The apparent prevalence estimate is P = 26 / 200 = 0.13

And the 95% CI is 0.083-0.177 (the adjusted Wald 95% CI = 0.090-0.184)

Bayesian approach

Assume that we have no prior information on disease: beta (1, 1)

Sample size (n) / herd size (N) is ≤ 0.1, we can assume a binomial distribution: X ~ bin (n, p)

For simplicity, we also assume that Se = Sp = 1

Posterior median and 95% PI for P is 0.132 (0.09-0.184)

Prior information (expert opinion): beta(5.62,42.57)… mode=0.1, and 95% sure that P<0.2

Posterior median and 95%PI for P is 0.126 (0.089-0.171)

Practice: Normal Regression Model

We are interested in estimation of the antibody levels against pathogen X in pigs at the slaughterhouse.

The response variable is the antibody level of each pig, and two variables that can affect the levels are the number of pigs that were in the same pen and the distance from the farm to the slaughterhouse.

A dataset of 25 observations was finally collected.

$Y_i \sim N(\mu, \sigma^2)$

$\mu_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in}$ for i = 1, 2, …, n

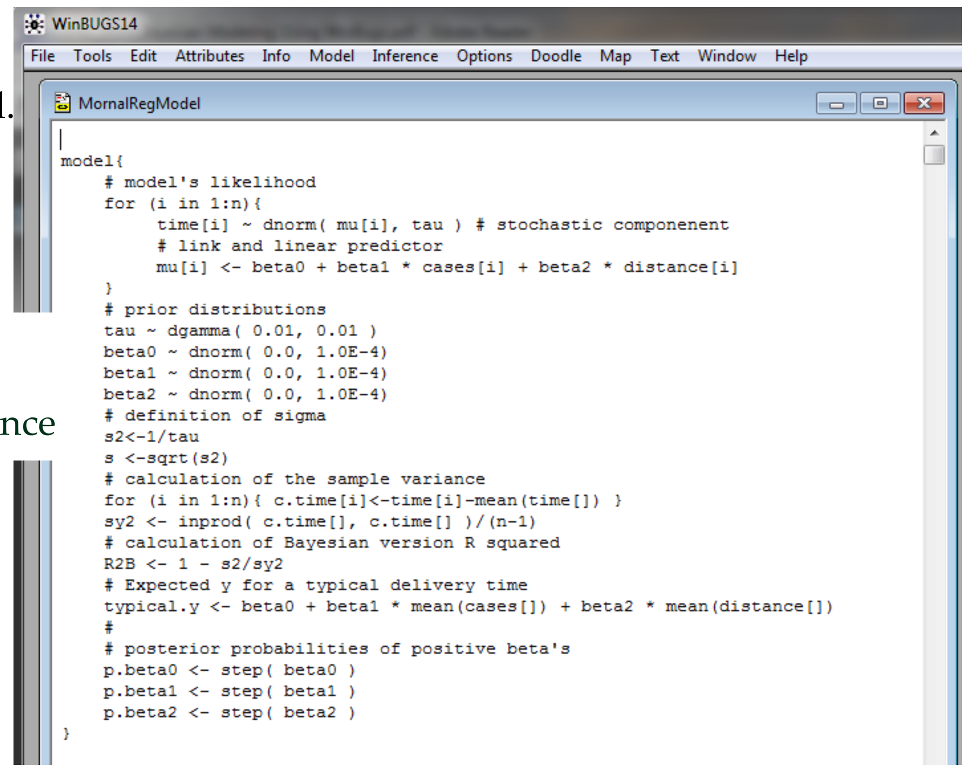
AB[]	pig[]	distance[]
16.68	7	560
11.5	3	220
12.03	3	340
14.88	4	80
13.75	6	150
18.11	7	330
8	2	110
17.83	7	210
79.24	30	1460
21.5	5	605
40.33	16	688
21	10	215
13.5	4	255
19.75	6	462
24	9	448
29	10	776
15.35	6	200
19	7	132
9.5	3	36
35.1	17	770
17.9	10	140
52.32	26	810
18.75	9	450
19.83	8	635
10.75	4	150

Practice: Normal Regression Model

We are interested in estimation of the antibody levels against pathogen X in pigs at the slaughterhouse.

The response variable is the antibody level of each pig, and two variables that can affect the levels are the number of pigs that were in the same pen and the distance from the farm to the slaughterhouse.

A dataset of 25 observations was finally collected.




```
model{
    # model's likelihood
    for (i in 1:n){
        time[i] ~ dnorm( mu[i], tau ) # stochastic componenent
        # link and linear predictor
        mu[i] <- beta0 + beta1 * cases[i] + beta2 * distance[i]
    }
    # prior distributions
    tau ~ dgamma( 0.01, 0.01 )
    beta0 ~ dnorm( 0.0, 1.0E-4)
    beta1 ~ dnorm( 0.0, 1.0E-4)
    beta2 ~ dnorm( 0.0, 1.0E-4)
    # definition of sigma
    s2<-1/tau
    s <-sqrt(s2)
    # calculation of the sample variance
    for (i in 1:n){ c.time[i]<-time[i]-mean(time[]) }
    sy2 <- inprod( c.time[], c.time[] )/(n-1)
    # calculation of Bayesian version R squared
    R2B <- 1 - s2/sy2
    # Expected y for a typical delivery time
    typical.y <- beta0 + beta1 * mean(cases[]) + beta2 * mean(distance[])
    #
    # posterior probabilities of positive beta's
    p.beta0 <- step( beta0 )
    p.beta1 <- step( beta1 )
    p.beta2 <- step( beta2 )
}
```

Model

Expected AB level = 2.36 + 1.6 pig + 0.015 distance

Practice: Binomial logistic regression 1

348 burger have been tested for contamination with E.coli.

We want to estimate the effect of three potential causes: X, Z, and Y, which are all of them dichotomous (i.e. presence or absence of the factor)

Practice: Binomial logistic regression 2

54 lactating cows from a small dairy cattle farm were tested by analysing their milk for somatic cell count (SCC). The aim of the study was to identify cows with mastitis (binary variable) using the SCC score. Moreover, we were interested in calculating the threshold value of X for which $\pi > 0.5$ to enable us to identify possible cows directly using X.

$Yi \sim \text{binomial}\ (\pi_i, N_i)$

$$log \frac{\pi_i}{1-\pi_i} = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} = X_{(i)}\beta$$

$Yi \sim \text{binomial}\ \left(\frac{odds_i}{1+odds_i}, N_i\right)$

$$\log(odds_i) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} = X_{(i)}\beta$$