B

Introduction to WinBUGS

B.1 INTRODUCTION

WinBUGS (the MS Windows operating system version of BUGS: Bayesian Analysis Using Gibbs Sampling) is a versatile package that has been designed to carry out Markov chain Monte Carlo (MCMC) computations for a wide variety of Bayesian models. The software is currently distributed electronically from the BUGS Project website. The address is

> http://www.mrc-bsu.cam.ac.uk/bugs/overview/contents.shtml

(click the WinBUGS link). If this address fails, a current link is maintained on the textbook website, or try the search words: "WinBUGS Gibbs" in a search engine. The downloaded software is restricted to fairly small models, but can be made fully functional by acquiring a license, currently for no fee, from the BUGS Project website. Versions of BUGS for other operating systems can be found at the BUGS project website.

The WinBUGS installation contains an extensive user manual (Spiegelhalter et al. 2002) and many completely worked examples. The manual and examples are under the "Help" pull-down menu on the main WinBUGS screen (Fig. B.1). The user manual is a detailed and helpful programming and syntax reference;
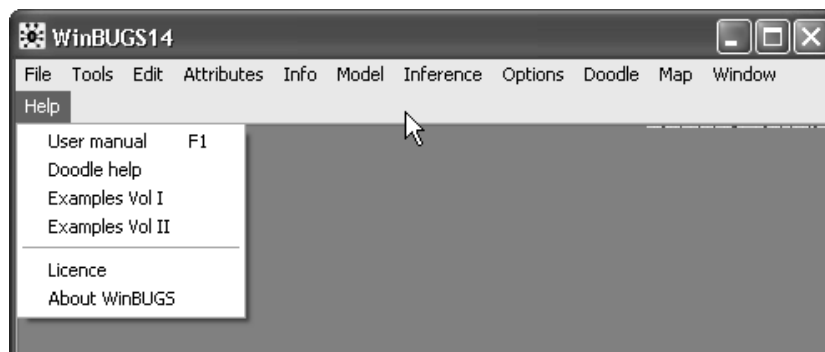

Figure B.1   The main WinBUGS screen, showing pull-down help menu.

however, the quickest way to become familiar with WinBUGS programming and syntax is to work through a few of the examples.

WinBUGS implements various MCMC algorithms to generate simulated observations from the posterior distribution of the unknown quantities (parameters or nodes) in the statistical model. The idea is that with sufficiently many simulated observations, it is possible to get an accurate picture of the distribution; for example, by displaying the simulated observations as a histogram as in Fig. 11.3 on page 232.

A WinBUGS analysis — model specification, data, initial values, and output — is contained in a single *compound document*. Analytic tools are available as pull-down menus and dialog boxes. Data files are entered as lists (or can be embedded as sub documents). Output is listed in a separate window but can be embedded in the compound document to help maintain a paper trail of the analysis. Any part of the compound document can be folded out of sight to make the document easier to work with. Data can be expressed in *list structures* or as rectangular tables in plain text format; however, WinBUGS cannot read data from an external file.

B.2   SPECIFYING THE MODEL — PRIOR AND LIKELIHOOD

To calculate a posterior distribution it is necessary to tell WinBUGS what prior distribution to use and what likelihood distribution to use. Distributions and likelihoods available in WinBUGS are listed in Table I of the WinBUGS user manual, and some of them are described in this section. Notice that all distribution and likelihood names begin with the letter "d" (for "distribution").

dnorm(μ, τ) is the normal distribution with parameters μ and $\tau = 1/\sigma^2$.

It is important to understand that WinBUGS specifies the normal distribution in terms of the mean μ and precision τ, rather than in terms of mean and standard deviation σ. The relationship between standard deviation and precision is $\sigma = 1/\sqrt{\tau}$. An important special case is dnorm(0, 0), which is flat over the entire number line. This distribution is *improper* in the sense that there is infinite area under the curve, and in practice a dnorm(0, ε) is used to represent ignorance, where ε is a small number such as 0.001.

dbin(p, n) is the binomial distribution with parameters n and p.

dbin is the distribution of the number of successes in n observations of a Bernoulli process with parameter p; for example the number of heads in 100 coin tosses has a dbin(0.5,100) distribution, and the number of black marbles in a sample of size n from a box in which the proportion of black marbles is p has a dbin(p, n) distribution.

dbeta(*a*, *b*) is the beta distribution with parameters *a* and *b*.

dbeta is a very flexible distribution family; it applies to an unknown quantity that takes values between 0 and 1 — for example, a success rate. An important special case is dbeta(1, 1), which is the uniform (flat) prior distribution over the interval (0,1). However, the dbeta(0, 0) distribution is more often used to represent complete ignorance about an unknown rate p because it implies that the log odds, $\ln(p/(1-p))$ has a uniform distribution over the entire number line. dbeta(0, 0) is an improper distribution with infinite curve area, and in practice dbeta(ε, ε) is used, with ε a small number such as 0.001.

dgamma(*a*, *s*) is the gamma distribution.

dgamma is a very flexible distribution family. It applies to unknown quantities that take values between 0 and ∞; for example, the unknown precision τ of an unknown quantity. Complete ignorance about a positive-valued unknown quantity is generally represented as a dgamma(0, 0) distribution. Since this distribution is improper, dgamma(ε, ε) is used in practice, with ε a small number such as 0.001.

B.3   INFERENCE ABOUT A SINGLE PROPORTION

The instructor in a statistics class spun a new Lincoln penny $n = 25$ times and observed "heads" $x = 11$ times. I am to obtain the posterior distribution of p, the rate at which a penny spun this way will land heads. I might profess complete ignorant about the unknown quantity p (the rate that the coin lands heads), or I might have some prior knowledge. In any case it is most convenient to represent my prior opinion as a beta distribution. For example, a flat prior is specified this way:

$$p \sim dbeta(1, 1)$$

The tilde ($\sim$) is pronounced "has a ___ distribution." Thus, my prior belief about p has a beta(1,1) distribution.

The second thing that WinBUGS needs to be told is the likelihood of the data *x*. Since *x* (the number of heads) can be modeled as the number of black marbles in a sample of size $n = 25$, the likelihood of *x* successes in *n* observations of a Bernoulli process (such as spinning a coin) is specified this way:

$$x \sim dbin(p,n)$$

Fig. B.2 is the WinBUGS program that makes use of these statements to analyze the coin spinning data ($x=11$ heads in $n =25$ spins). The word "MODEL" is not mandatory; WinBUGS treats everything between the opening and closing braces { } as a description of the statistical model, that is, a description of the prior

```
MODEL {
  p ~ dbeta(a,b)
  x ~ dbin(p,n)
}
DATA list(a=1,b=1,x=11,n=25)
```

Figure B.2   WinBUGS program to compute the posterior distribution
of the success rate p based on 11 successes in n trials.

distribution of p and the likelihood of *x*. Observed data are entered by means of a
list separated by commas. The word "list" and the parentheses are required, but
the word DATA is treated as a comment. The data list contains the inputs to the
analysis: the analyst's prior belief (a=1, b=1) and the observed data (x = 11 heads
in n = 25 spins).

B.3.1   Setting Up the Model

Launch WinBUGS.   The icon, which resembles a spider, is in the directory
where WinBUGS was installed — it is convenient to drag a shortcut to the
desktop.   Read and then close the license agreement window, and open a new
document window (pull down: file/new).

Specify the Model.   Type the contents of Fig. B.2 in the document window you
just opened. To save the program, select the window containing the program you
just typed in, pull down the File menu, select Save as, enter a file name (WinBUGS
will add the extension .odc), navigate to an appropriate directory, and save the
file.

Check the Syntax.   Pull down the Model menu, and select "Specification." The
specification tool window will open (Fig. B.3) Single-click anywhere in the model
(between the curly braces), and then click the "check model" button. Look in the
message bar along the bottom of the WinBUGS window. You should see the
phrase, "model is syntactically correct." Syntax errors produce a variety of error
messages. For example, in Fig. B.4 the programmer typed "p = dbeta(1,1)"
instead of "p ~ dbeta"). Note that the cursor | is positioned somewhere after the
symbol that caused the error. The default cursor is hard to see but can be made
more visible by checking the box for "Thick Caret" in the Edit\Preferences\ dialog
box. Correct any syntax errors and repeat the "check model" and "load data"
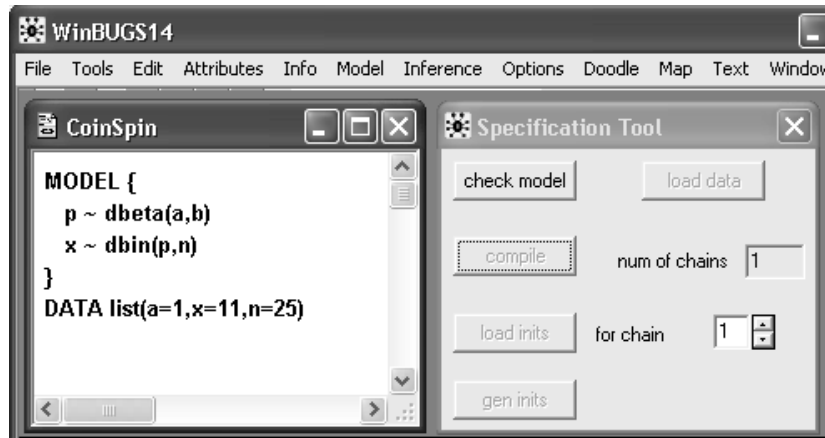steps above until the model is free of syntax errors.

Figure B.3 The specification tool. Position the cursor in the program window, then click "check model" in the specification tool window.

Enter Data.   In the DATA statement, highlight any part of the word "list" and then click "load data" in the specification tool window. If the highlighting extends beyond the word "list," there will be an error message. Correct any data errors, and repeat the "check model," "compile," and "load data" steps.
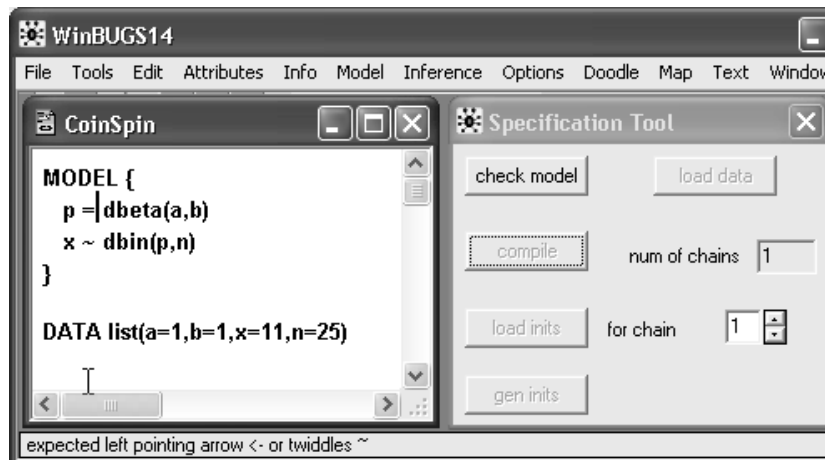

Figure B.4 A syntax error. The programmer typed "p = "instead of "p ~ ". WinBUGS positions the cursor after the character that caused the error.

Compile the Model.   Click "compile" and look for the words "model compiled" in the message bar across the bottom of the WinBUGS window. Fig. B.5 shows a compilation error caused by not providing a value for the parameter b in the data list. Another common compilation error is misspelling a variable name — WinBUGS is case-sensitive, which means that it interprets b and B as different symbols. Inconsistent spelling of the same variable is one of the most common errors in WinBUGS programs.

Notice that WinBUGS used the word "node" in the error message in Fig. B.5. A *node* is any variable or constant that is mentioned in the model. In this case the nodes are a, b, n, p, and x. Node p is an unknown quantity; the other four are known quantities entered via the data statement.

Generate Initial Values.   Click "gen inits" in the Specification Tool. Sometimes WinBUGS will display an error message indicating that it is unable to generate initial values. In such cases it is up to the programmer to provide initial values. We'll learn how to do this later.

B.3.2   Computing the Posterior Distribution

Select the Nodes (Unknown Quantities) to be Monitored.   *Monitoring* a node means asking that WinBUGS keep a file of the simulated values of that node. In this case we must monitor node p, the unknown success rate. To do this pull down the Inference menu and select "Samples." The Sample Monitor Tool will
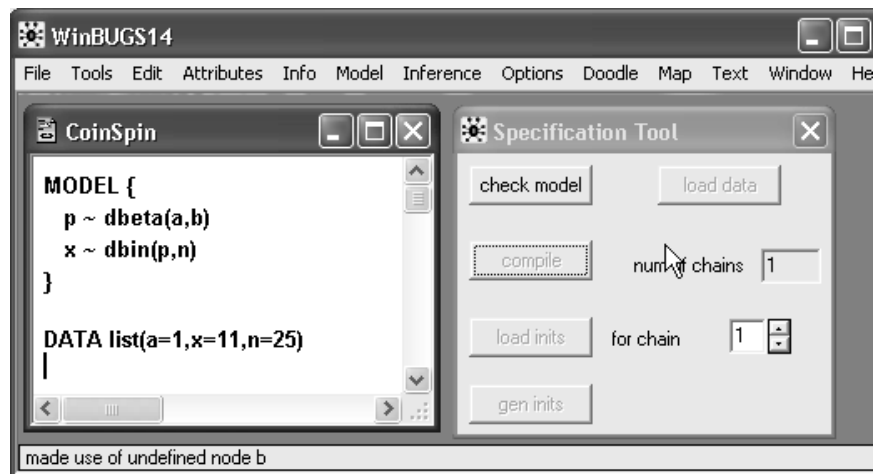

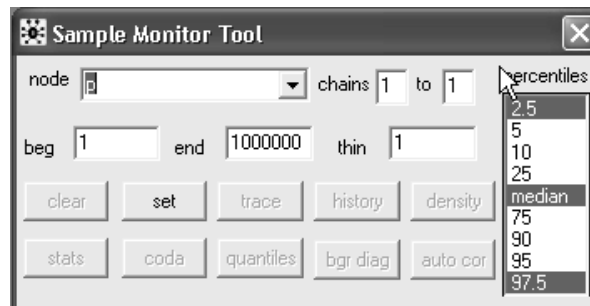**Figure B.5**   A compilation error. The programmer failed to provide a value for node "b".

Figure B.6   The Sample Monitor Tool. Enter each node to be monitored clicking "set" after each entry

appear (Fig. B.6). In the "node" field type "p" (without quotes) and click "set." In more complicated models repeat these steps for each of the unknown quantities of interest. If the "set" button does not darken after you type a node name, check for a spelling error (perhaps you used the wrong case).

Generate Simulated Values of All Unknown Quantities.   Pull   down   the Model menu and select "Update"; the *Update Tool* will appear (Fig. B.7). In the "updates" field enter the desired number of simulations (for example 5000) as shown in Fig. B.7. In a complex model it is a good practice to start with 100 in the "updates" field and 10 in the "refresh" field to get some idea of how fast the simulation runs. Click "update" to start the simulations. The simulation can be stopped and restarted by clicking the "update" button. Several thousand to hundreds of thousands of simulations are required to get reasonably accurate posterior probabilities, moments, and quantiles. The updates field controls how often the display is refreshed – changing it has no effect on the speed of simulations; making it smaller, however, reduces the amount of time that WinBUGS is unresponsive.

Occasionally, a "Trap" display such as Fig. B.8 will appear during simulations. If this happens, try clicking the "update" button twice to restart the
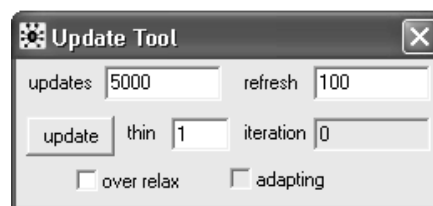

Figure B.7   The Update Tool. Enter the desired number of simulations in the "updates" field. If simulations are generated slowly, enter a smaller number in the "refresh" field.
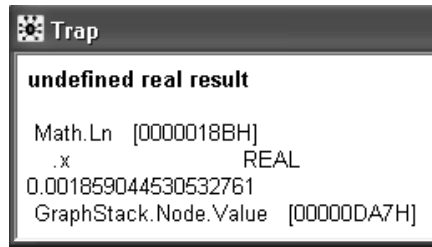
Figure B.8   An error trap may be transitory or may require tightening the prior.

simulations. If the trap continues to reappear, the model will have to be modified, typically by making prior distributions more informative.

Examine the Posterior Distribution.   Return to the Sample Monitor Tool, and enter 1001 in the "beg" field – this instructs WinBUGS to discard the first 1000 simulations to get past any initial transients. In the "node" field enter the name of the unknown quantity that you want to examine and click "density" to see a graph of its posterior density, and then click "stats" to see quantiles and moments of the posterior distribution. The default display is the posterior mean and standard deviation, along with the median and 95% credible interval; however, you can select other percentiles by clicking any number of choices in the "percentile" window of the Sample Monitor Tool. To select a percentile, position the arrow cursor over the desired percentile and ctrl-click the left mouse button. Fig. B.9 shows how to request the 5th and 95th percentiles, which are the endpoints of the 90% credible interval.

A complete WinBUGS session is displayed in Fig. B.10. The *kernel density* graph in Fig. B.10 is a smoothed histogram of the simulations and is an approximation of the posterior distribution. The *node statistics* table lists the mean

Figure B.9   The Sample Monitor Tool. The user has requested five percentiles.The arrow cursor is positioned to select the 10th percentile (ctrl-click the left mouse button).

and standard deviation of the posterior distribution of each monitored quantity as well as selected percentiles. The default display includes the median and the 2.5th and 97.5th percentiles but other percentiles can be requested as explained in the previous paragraph. The columns of the display are labeled as follows:

- **node** The name of the unknown quantity
- **mean** The average of the simulations, an approximation of the μ of the posterior distribution of the unknown quantity
- **sd** The standard deviation of the simulations, an approximation of the σ of the posterior distribution
- **MC error** The computational accuracy of the mean
- **2.5%** the 2.5th percentile of the simulations, an approximation of the lower endpoint of the 95% credible interval
- **median** The median or 50th percentile of the simulations, and
- **97.5%** The 97.5th percentile of the simulations, an approximation of the upper endpoint of the 95% credible interval
- **start** The starting simulation (after discarding the start-up)
- **sample** The number of simulations used to approximate the posterior distribution

The MC error is purely technical, like round-off error, and can be made as small as desired by increasing the number of simulations (reported under "sample" in the node statistics table). On the other hand, the posterior standard deviation, the analog of the standard error in conventional statistical inference, represents genuine uncertainty and cannot be reduced other than by obtaining additional real data. Note also that the number 50,000 in the "samples" column in the Node statistics table is the number of simulations, *not* the sample size of the data n = 25.

The WinBUGS output displayed in Fig. B.10 indicates that the posterior distribution of p, the rate of occurrence of heads in penny spinning, is approximately normal (judging from the graph) with $\mu = 0.4441$ and $\sigma = 0.0939$. These numbers are computationally accurate to about ± 0.0004 (MC error); consequently it would be more appropriate to report $\mu = 0.444$ and $\sigma = 0.094$.

Figure B.10 A complete WinBUGS session. The posterior distribution of p, the rate of occurrence of heads in coin spinning, is approximately normal with $\mu = 0.444$ and $\sigma = 0.094$.
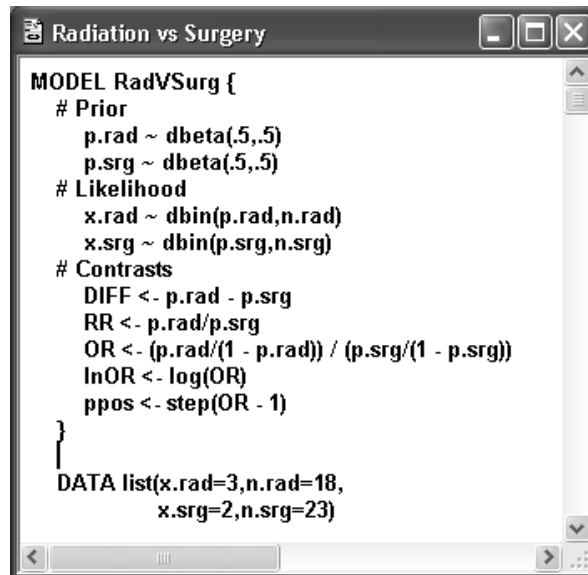
B.4   TWO RATES – DIFFERENCE, RELATIVE RISK, AND ODDS RATIO

In a study comparing radiation therapy vs. surgery, cancer of the larynx remained uncontrolled in 3 of 18 radiation patients and 2 of 23 surgery patients. Fig. B.11 shows the analysis. The unknowns p_{rad} and p_{srg} are the rates of failure of radiation and surgery, respectively, and x_{rad}, n_{rad}, x_{srg}, and n_{srg} are the observed data. The prior distributions of the unknown quantities have been given dbeta(.5, .5), which makes the difference ($\Delta = p_{rad} - p_{srg}$) have a more nearly uniform prior. Note that text following a pound sign, #, is interpreted as a comment.

This example illustrates the "arrow" symbol for assigning values to logical nodes; that is, unknown quantities, such as the odds ratio, that are computed from more basic unknown quantities p_{rad} and p_{srg} . Here the logical nodes are the difference of the two failure probabilities (DIFF), the relative risk (RR), the odds ratio (OR), and a Boolean variable (ppos, explained below) that counts the number of times the research question "Is radiation less effective?" is true:

$$\text{Research question: Is it true that } p_{srg} > p_{rad} ? \tag{B.1}$$

The fundamental unknown quantities are the success rates for the two different treatment modes; however, answering the research question requires contrasting the two rates. In Chapter 7 we learned three ways of contrasting two rates: the difference (DIFF or Δ), the odds ratio (OR), and the relative risk (RR).


```
Radiation vs Surgery

MODEL RadVSurg {
    # Prior
        p.rad ~ dbeta(.5,.5)
        p.srg ~ dbeta(.5,.5)
    # Likelihood
        x.rad ~ dbin(p.rad,n.rad)
        x.srg ~ dbin(p.srg,n.srg)
    # Contrasts
        DIFF <- p.rad - p.srg
        RR <- p.rad/p.srg
        OR <- (p.rad/(1 - p.rad)) / (p.srg/(1 - p.srg))
        lnOR <- log(OR)
        ppos <- step(OR - 1)
}

DATA list(x.rad=3,n.rad=18,
          x.srg=2,n.srg=23)
```

Figure B.11   Comparing two rates – illustrating computed (logical) nodes.

The research question can be stated in terms of any one of the contrasts:

$$\text{Is } \Delta > 0?$$
$$\text{Is OR} > 1?$$
$$\text{Is RR} > 1?$$

(B.2)

The contrasts D, OR, and RR are functions of the two success rates, p_{rad} and p_{srg}. In algebra, functional relationships are written with an = sign, however, in many computer languages a distinction is made between two variables occupying the same location in memory (=) and a value being computed and assigned to a variable. In WinBUGS, the identity symbol (=) is used only in data lists, whereas the assignment symbol "<−" (often pronounced "gets") is used in the model to indicated that one node gets its value from other nodes. For example, RR gets its value by dividing the surgical cure rate by the radiation cure rate: RR <− p_{srg} / p_{rad}.

The twiddle symbol indicates that a node has a particular distribution. For example, $x \sim dbin(p,n)$ means that "x is distributed like the number of successes in n observations of a Bernoulli process." Inadvertent use of an "=" sign instead of a twiddle or an arrow is one of the most common reasons for a compilation error message. The equal sign is never used in a WinBUGS model, although it is used in data lists.

The WinBUGS program in Fig. B.11 uses the step( ) function to create a Boolean variable that counts the number of simulations in which the sentence "$p_{rad} \geq p_{srg}$" is true. Here's how it works: if V is any node, then step(V) equals 1 if $V \geq 0$ and equals 0 if $V < 0$. Consequently, step(a − U) equals 1 if a − U $\geq$ 0; that is, if a $\geq$ U. The step( ) function can be used to compute left- or right-tail areas:

$$
\begin{array}{lll}
P(U \leq a): & \text{new\_node\_1} <- \text{step}(a - U) \\
P(U < a): & \text{new\_node\_2} <- 1 - \text{step}(U - a) \\
P(U \geq b): & \text{new\_node\_3} <- \text{step}(U - b) \\
P(U > b): & \text{new\_node\_4} <- 1 - \text{step}(b - U)
\end{array}
$$

(B.3)

The word "new_node" is generic; each tail request must have a unique node name such as "ppos" in Fig. B.11. The mean value of a Boolean node such as those in Equation (B.3) is a probability; for example, the mean of new_node_1 is the Monte-Carlo estimate of P(U$\leq$a).

Table B.1 is a list of some of the other functions available in WinBUGS; a complete list is found in Table II of the WinBUGS user manual (Spiegelhalter et al. 2002). Note that the natural logarithm, ln( ), is called log( ) in WinBUGS, and the pow function is used to raise a number to a power

$$U^c: \qquad \text{new\_node} <- \text{pow}(U, c)$$

Name	Action
step(x)	1 if x $\geq$ 0, otherwise 0
log(x)	ln(x)
logit(p)	ln(p/(1-p))
exp(x)	exp(x)
abs(x)	$|x|$
pow(x,c)	x^c
sqrt(x)	$\sqrt{x}$

.

To compute the posterior distributions of DIFF, RR, and OR, follow the steps in Section B.3 with this change: under the heading "Select the Nodes ... to be Monitored" on page 310 it is necessary to enter four node names, one at a time. First type DIFF in the "node" field and click "set," then do the same for OR, RR, and ppos. Be careful about capitalization, since WinBUGS is case-sensitive. After entering the node names, continue with the instructions.

A second change is required at the last step, "Examine the Posterior Distribution," page 312. To graph the posterior distributions and compute moments, quantiles, and credible intervals, enter an asterisk in the node field of the Sample Monitor Tool (Fig. B.12); then click "stats" and "density."

The raw output, displayed in Fig. B.12, has been edited and annotated for greater clarity in Fig. B.13. The edits involved clarifying the meaning of the nodes, and not reporting unreliable digits. (For example, the statistics for DIFF are unreliable beyond the 4th decimal place, because the MC error is about 0.0005). Although each of the contrasts suggests that surgery has the lower failure rate, none of them rules out equality.

Figure B.12   How to request posterior distribution statistics for all four monitored nodes. Enter an asterisk in the node field then click stats and density.

Node statistics

node	mean	sd	MC error	2.5%	median	97.5%
DIFF	0.07993	0.1064	4.893E-4	-0.1232	0.07542	0.2997
OR	3.444	5.489	0.02497	0.3227	2.049	14.77
RR	2.815	4.013	0.01825	0.3726	1.849	10.93
lnOR	0.736	0.9696	0.004443	-1.131	0.7173	2.693
ppos	0.7804	0.4139	0.001837	0.0	1.0	1.0

Figure B.13   Node statistics for comparing surgical and radiological failure rates based on 50,000 simulated values.

The 95% credible intervals for the odds ratio and relative risk include 1 (meaning equal rates), and the 95% credible intervals for the difference and log odds ratio include 0 (also meaning equal rates). On the other hand, odds ratios as high as 15 and relative risks as high as 11 cannot be ruled out. The posterior probability that radiation has the higher failure rate is about 78%.

Posterior distributions are graphed in Fig. B.14. As expected the odds ratio and relative risk have heavily skewed distributions, but the difference and log odds ratio appear to have nearly normal distributions. The distribution of the node, ppos, requires some explanation. Recall that ppos was produced by the step( ) function, which means that it is a Boolean (0 or 1) variable. The value ppos = 1 identifies simulations in which surgery is better than radiation. The histogram of ppos has only two bars — at 0 and 1— and the height of the bar at 1 is the proportion of simulations for which the ppos = 1, that is, the proportion of times the sentence "Rad > Srg" was true, which approximates the posterior probability that it is true. That proportion is also equal to the mean and therefore the mean is the only meaningful descriptive statistic for a Boolean variable. The mean (i.e. the proportion of 1's) completely describes the data, and for that reason the histogram and statistics other than the mean are confusing and have been suppressed in Table B.2.

Table B.2   Surgery vs. radiation: posterior moments and quantiles.

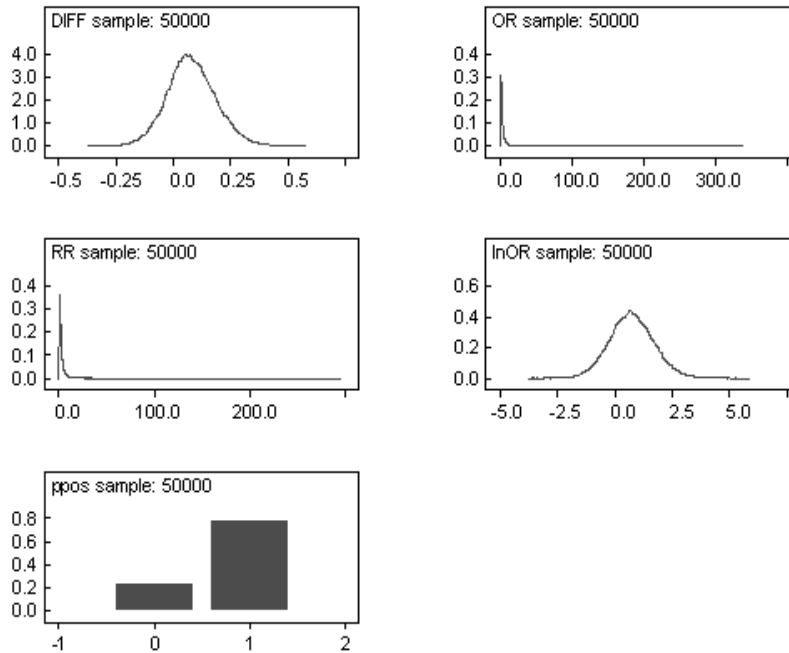| Failure Rate Comparisons	μ	σ	median	95% Credible Interval	
Difference (Rad - Srg)	0.0799	0.1064	0.0754	-0.1232	0.2997
Odds Ratio (Rad/Srg)	3.44	5.49	2.05	0.32	14.77
Relative Risk (Rad/Srg)	2.82	4.01	1.85	0.37	10.93
ln(Odds Ratio)	0.736	0.97	0.717	-1.131	2.693
P(Rad > Srg $\mid$ Data)	0.78				

Figure B.14   Posterior distributions of five nodes. Probability nodes are entirely described by the mean.

.

B.5   "FOR" LOOPS

The purpose of WinBUGS model specification language is to specify the prior distributions of the unknown parameters and the likelihood function of the observed data. It is not a programming language. It does not specify a series of commands to be executed in sequence. In fact, model specification statements can be written in almost any order without changing the meaning of the model. Repetitive model components, as in a hierarchical model, can be specified using "for" loops but conditional branching structures such as "if ... then ... else" are not available and, indeed, have no meaning in model specification.

Using the "For" Structure: The Pediatric Mortality Study .   This example was described in Section 11.7. The data are numbers of patients and numbers of deaths in 12 hospitals. The WinBUGS program in Fig. B.15 uses the "for" programming structure to specify the model more compactly. The following segment illustrates how the "for" structure makes the model specification much more compact:

```
for (i in 1:k) {
      #Prior distribution of Hospital i's True Rate
      p[i] ~ dbeta(a,b)
      #Likelihood of Hospital i's Data              (B.4)
      x[i] ~ dbin(p[i],n[i])
   }
```

The variables (nodes) in this fragment are the true mortality rate in the *i*th hospital, p[i]; and the observed number of patients, n[i], and deaths, x[i], in that hospital. Without the "for" structure, it would have taken 24 lines to specify the prior and likelihood:

```
p[1] ~ dbeta(a,b)
x[1] ~ dbin(p[1],n[1])
p[2] ~ dbeta(a,b)
x[2] ~ dbin(p[2],n[2])
p[3] ~ dbeta(a,b)
x[3] ~ dbin(p[3],n[3])
... 16 lines omitted ...
p[12] ~ dbeta(a,b)
x[12] ~ dbin(p[12],n[12])
```

WinBUGS uses square brackets to denote subscripts. Thus p[i] in program fragment (B.4) is what we would ordinarily write as p_i, the unknown true long-term morality rate in the *i*th hospital, and n[i] and x[i] are what we would write as n_i and x_i. Subscript *i* is the loop *index* and the expression 1:k is its *range*. The range must include only positive integers such as 1:12 or 3:7. A range can be specified in terms of numbers or integer-valued variables.

```
MODEL Hospital {
    #Hyperprior for the Box of Rates
      a~dgamma(.001,.001)
      b~dgamma(.001,.001)
    #Prior Distribution of the True Rates
    for (i in 1:k) {
      #Prior distribution of Hospital i's True Rate
      p[i] ~ dbeta(a,b)
      #Likelihood of Hospital i's Data
       x[i] ~ dbin(p[i],n[i])
    }
}
DATA list(k=12,
    n = c(47,148,119,810,211,196,148,215,207,97,256,360),
    x = c( 0,  18,  8, 46,  8, 13,  9, 31, 14, 8, 29, 24))
INITIAL VALUES list(a=1,b=1)
```

Figure B.15 WinBUGS program for the hospital study.

B.6   DATA ENTRY

Data for a subscripted variable can be entered as a list or in a table. For example, list input is used in Figure B.15,

```
DATA list(k=12,
n = c(47,148,119,810,211,196,148,215,207,97,256,360),
x = c( 0, 18,  8, 46,  8, 13,  9, 31, 14, 8, 29, 24))
```

Individual numbers such as *k* are entered as k=12, for example. Data for the subscripted variables *n* and *x* are specified as *collectives*, indicated by the letter "c" followed by a parenthetical, comma-separated list, for example, n = c(47,..., 360). List input is convenient if there are only a few data items; however, for large data sets it can be more convenient to enter the data in the form of an *embedded table*.

B.6.1   Embedding a Data Table in WinBUGS

WinBUGS allows documents to be embedded in a *compound document,* thus providing convenient way to save the program, data, and output in a single computer file. The first step is to prepare a plain text file of data arranged (in this case) as a 12 by 2 matrix. First type or paste the data matrix in a new WinBUGS document window (pull down: file/new to create a new document window), then follow these instructions:
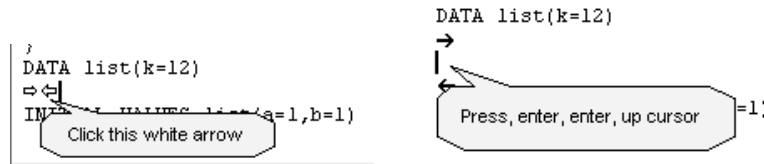
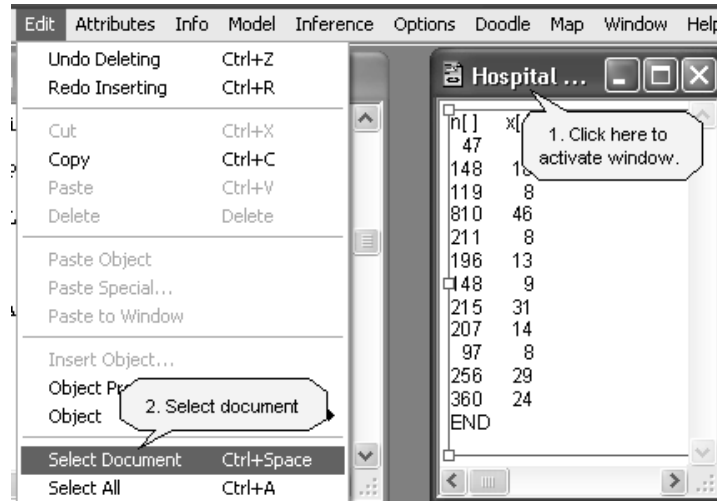Step 1: Create the program and data files in two windows:
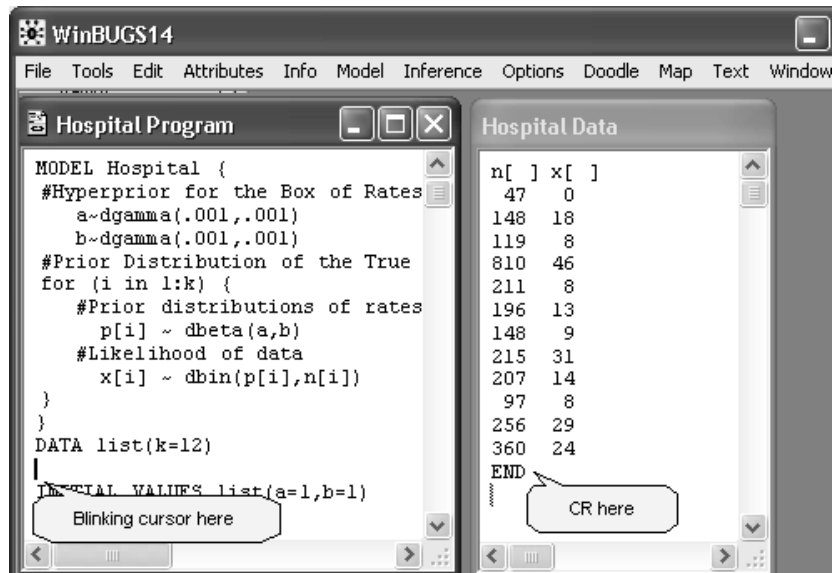
Step 2: Create a fold at the bottom of the program file:
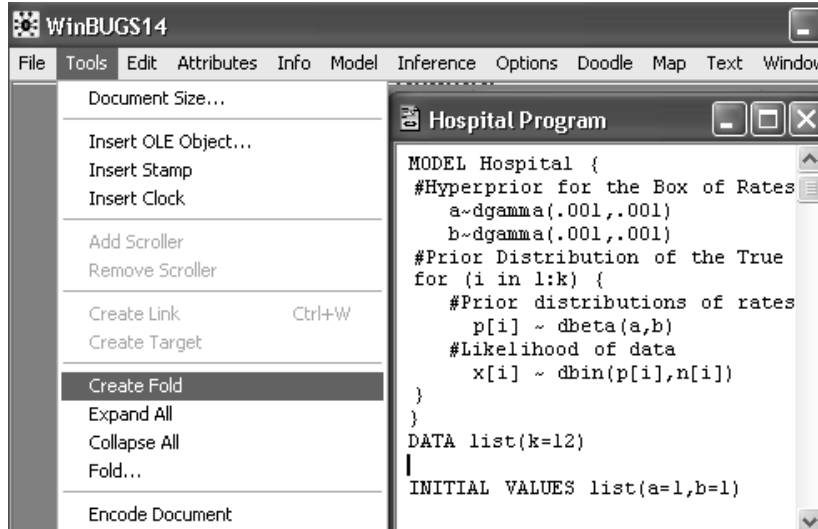
Step 3: Open the fold and enter a blank line:

Step 4: Copy the data document:

Step 5: Copy, and paste the data into the fold, resize the "hairy border" around the data table, and close and label the fold:

### B.6.2   Loading Data from an Embedded Table

After checking the model, if there is a data list as well as a data table, load the list in the usual way, and then open the fold containing the data table. Click anywhere in the data table, but do not highlight any text, as that will produce an error message. The data table should be surrounded by a "hairy border." Click "load data," close the fold and proceed with the compilation. Note that the last line in the data table must be the word END on a separate line followed by a carriage return. If this is missing, WinBUGS will report that there is an incomplete data line.

B.7   PLACING OUTPUT IN A FOLD

It is a good idea to paste output tables and graphs into the compound document containing the model specification and data. This is easy to do and creates a complete record of the analysis in a single document that can be saved and, if desired, re-opened for modification or additional analyses. For example, the node statistics table is initially reported in a separate document. Click anywhere in that document, and copy it (pull down Edit/select all, then Edit/copy). Note that you must choose "select all", not "select document" to copy the node statistics table. Insert and label a fold in the main document. Open the fold, and paste in the node statistics table.

B.8   ADDITIONAL RESOURCES

Readers are urged to look at the user manual (under the Help menu), which has a tutorial chapter and provides much more detail on setting up WinBUGS analyses. The examples "Vol I" and "Vol II" (under the Help menu) are also worth a look. A good way to learn to use WinBUGS with your own data is to imitate an example similar to the analysis that you want to do.   WinBUGS also has the option to set up the model (prior and likelihood) in graphical form. See "Doodle help" under the Help menu as well as the excellent introduction to graphical models in Fryback et al. (2001).
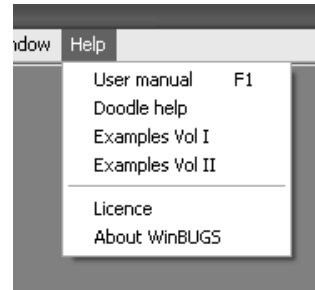
B.9   WEIBULL PROPORTIONAL HAZARDS REGRESSION

This section is significantly more difficult and can be skipped without losing continuity.

Multiple myeloma survival data.   The data to be analyzed is a subset of the multiple myeloma survival data in Table 2 of Krall et al. (1975). The response variable is survival time (time from diagnosis to death). Some patients were alive at the end of the observation period and their survival times are therefore truncated (known only to be longer than the observation period). The regression model uses ln(BUN) as a continuous explanatory variable and has separate slopes and intercepts for men and women. The data in raw form and they must be arranged for WinBUGS are shown in Table B.3.

Notice that the survival time variable must be split into two variables t.obs, corresponding to subjects who died during the observation period, and t.cen, corresponding to censored cases that were still alive at the end of the observation period. For censored cases, t.obs is recorded as "NA," which is WinBUGS representation of an unknown data value. The censoring time variable t.cen is recorded as 0 for uncensored cases. Note also that the explanatory variables, BUN and Sex, have been converted into *doubly subscripted* design variables:

$$x[,1] \quad \text{1 if female,   0 if male}$$
$$x[,2] \quad \ln(\text{BUN}) \text{ if female,   0 if male}$$
$$x[,3] \quad \text{1 if Male,   0 if Female}$$
$$x[,4] \quad \ln(\text{BUN}) \text{ if male,   0 if female}$$

Thus $x[,1]$ and $x[,2]$ are design variables for the female intercept and slope, and $x[,3]$ and $x[,4]$ are design variables for the male intercept and slope. The raw data are not entered in the WinBUGS data table.

Table B.3   Multiple myeloma survival data.[a]

				Data in WinBUGS format					
Raw Data				Time (months)		Design Variables			
Time	Dead	BUN	Sex	t.obs[ ]	t.cen[ ]	x[,1]	x[,3]	x[,2]	x[,4]
3	1	35	F	3	0	1	3.56	0	0
4	0	84	F	NA	4	1	4.43	0	0
5	1	172	F	5	0	1	5.15	0	0
6	1	130	F	6	0	1	4.87	0	0
6	1	26	F	6	0	1	3.26	0	0
7	1	11	F	7	0	1	2.40	0	0
7	1	15	F	7	0	1	2.71	0	0
7	0	13	F	NA	7	1	2.56	0	0
8	0	12	F	NA	8	1	2.48	0	0
11	1	12	F	11	0	1	2.48	0	0
12	0	14	F	NA	12	1	2.64	0	0
12	0	25	F	NA	12	1	3.22	0	0
13	1	6	F	13	0	1	1.79	0	0
13	0	46	F	NA	13	1	3.83	0	0
16	1	21	F	16	0	1	3.04	0	0
18	1	28	F	18	0	1	3.33	0	0
19	1	18	F	19	0	1	2.89	0	0
19	0	21	F	NA	19	1	3.04	0	0
24	1	20	F	24	0	1	3.00	0	0
26	1	17	F	26	0	1	2.83	0	0
28	0	17	F	NA	28	1	2.83	0	0
41	1	14	F	41	0	1	2.64	0	0
41	0	57	F	NA	41	1	4.04	0	0
52	1	10	F	52	0	1	2.30	0	0
58	1	16	F	58	0	1	2.77	0	0
88	1	15	F	88	0	1	2.71	0	0
92	1	27	F	92	0	1	3.30	0	0
1.25	1	165	M	1.25	0	0	0	1	5.11
1.25	1	87	M	1.25	0	0	0	1	4.47
2	1	33	M	2	0	0	0	1	3.50
2	1	56	M	2	0	0	0	1	4.03
2	1	20	M	2	0	0	0	1	3.00
4	0	90	M	NA	4	0	0	1	4.50
5	1	48	M	5	0	0	0	1	3.87
6	1	23	M	6	0	0	0	1	3.14
6	1	13	M	6	0	0	0	1	2.56
7	1	95	M	7	0	0	0	1	4.55
7	0	34	M	NA	7	0	0	1	3.53
9	1	53	M	9	0	0	0	1	3.97
11	1	13	M	11	0	0	0	1	2.56
11	1	17	M	11	0	0	0	1	2.83
11	1	20	M	11	0	0	0	1	3.00
11	1	37	M	11	0	0	0	1	3.61
11	0	41	M	NA	11	0	0	1	3.71
14	1	25	M	14	0	0	0	1	3.22
15	1	40	M	15	0	0	0	1	3.69
16	1	22	M	16	0	0	0	1	3.09
16	0	14	M	NA	16	0	0	1	2.64
17	1	17	M	17	0	0	0	1	2.83
17	1	39	M	17	0	0	0	1	3.66
19	1	12	M	19	0	0	0	1	2.48
19	0	21	M	NA	19	0	0	1	3.04
25	1	10	M	25	0	0	0	1	2.30
32	1	21	M	32	0	0	0	1	3.04
35	1	13	M	35	0	0	0	1	2.56
37	1	40	M	37	0	0	0	1	3.69
41	1	10	M	41	0	0	0	1	2.30
51	1	37	M	51	0	0	0	1	3.61
53	0	13	M	NA	53	0	0	1	2.56
54	1	18	M	54	0	0	0	1	2.89
57	0	18	M	NA	57	0	0	1	2.89
66	1	28	M	66	0	0	0	1	3.33
67	1	21	M	67	0	0	0	1	3.04
77	0	12	M	NA	77	0	0	1	2.48
89	1	21	M	89	0	0	0	1	3.04

a. Source of raw data: Krall, et al. (1975), Table 2.

Weibull Proportional Hazards Regression.   The basic WinBUGS commands to do any Weibull proportional hazards regression analysis are listed in Figure B.16. The output in panel B of the figure shows that the female intercept is greater than the male intercept and therefore that females with low BUN (blood urea nitrogen) are at greater risk of death than males. However, the male slope is substantially greater than the female slope; consequently, at some BUN value, the males will "catch up" and have greater risk of death. Beyond that general observation, it is difficult to interpret the output without estimating survival rates for males and females at various BUN values. This requires adding some specialized instructions to the generic instructions in Fig. B.16.

Computing Survival Rates and Female/Male Contrasts.   Instructions directing WinBUGS to compute survival rates and make male-female contrasts are listed in Fig. B.16 with output in Fig. B.5; however, the first step is to create an

A. Basic WinBUGS commands for Weibull regression

```
MODEL Weibull PHR {
   # Prior distribution of baseline hazard function
      shape ~ dgamma(1,.001)

   # Prior distribution of the regression coefficients
      for (i in 1:k) { beta[i] ~ dnorm(0,.001) }

   # Likelihood of the survival time data
      for (j in 1:n) {
      HRx[j] <- exp(inprod(x[j,],beta[]))
      Scale[j] <- HRx[j]
      t.obs[j] ~ dweib(shape,Scale[j])I(t.cen[j],)
      }
   # Insert any additional commands starting here.
   }
```

B. Output[a]: Posterior Moments and Quantiles of Regression Coefficients.

Description	Node	Mean	sd	MC error	2.5%	Median	97.5%
Female Intercept	beta[1]	−4.84	1.44	0.05	−7.64	−4.84	−1.97
Female Slope	beta[2]	0.21	0.43	0.015	−0.69	0.24	0.98
Male Intercept	beta[3]	−8.00	1.42	0.06	−10.92	−7.96	−5.56
Male Slope	beta[4]	1.26	0.36	0.015	0.57	1.26	1.99
Weibull shape	shape	1.16	0.13	0.004	0.93	1.15	1.42

a. Based on 60,000 simulations.

Figure B.16   Weibull regression analysis of multiple myeloma survival data.

WEIBULL PROPORTIONAL HAZARDS REGRESSION **327**

Table B.4 "Requests" file[a]

Name	Description: survival rate at 24 months for	Raw Data			WinBUGS formatted data				
		Month	BUN	SEX	t.r[]	x.r[,1]	x.r[,3]	x.r[,2]	x.r[,4]
Sr[1]	Males with BUN = 6	24	6	M	24	0	0	1	1.79
Sr[2]	Females with BUN = 6	24	6	F	24	1	1.79	0	0
Sr[3]	Males with BUN = 14	24	14	M	24	0	0	1	2.64
Sr[4]	Females with BUN = 14	24	14	F	24	1	2.64	0	0
Sr[5]	Males with BUN = 21	24	21	M	24	0	0	1	3.04
Sr[6]	Females with BUN = 21	24	21	F	24	1	3.04	0	0
Sr[7]	Males with BUN = 37	24	37	M	24	0	0	1	3.61
Sr[8]	Females with BUN = 37	24	37	F	24	1	3.61	0	0
Sr[9]	Males with BUN = 172	24	172	M	24	0	0	1	5.15
Sr[10]	Females with BUN = 172	24	172	F	24	1	5.15	0	0

a. This file, in conjunction with the commands in Fig. B.17, instructs WinBUGS to compute the indicated survival rates. Only the WinBUGS formatted data are submitted; the name, description and raw data are not included in the file.

additional data file (Table B.4) that specifies the time point and design variable values at which the survival function is to be calculated. The design variable values in this instance are the minimum, maximum, median, and quartile BUN values actually observed in the data set.

WinBUGS commands to compute the requested survival rates and male/female contrasts are shown in Fig. B.17; these commands are to be inserted in the program in Panel A of Fig. B.16 at the place indicated. Output is listed in Table B.5, symbols used in the program are explained in the output table. Thus, HRfm[1] is the female-vs.-male hazard ratio for patients at the minimum BUN value, 6. At this BUN level, females are at five time the risk of death of males; at the median, BUN=21, males and females have nearly the same risk, and at

```
# Requested survival rates for this analysis:
   for (j in 1:m) {
     HRr[j] <- exp(inprod(x.r[j,],beta[]))
       Sr[j] <- exp(-HRr[j]*pow(t.r[j],shape))
   }

# Contrasts of interest in this analysis:
   #Female/Male RR's & HR's
    for (j in 1:5) {
       RRfm[j] <- (1-Sr[2*j])/(1-Sr[2*j-1])
       HRfm[j] <- HRr[2*j]/HRr[2*j-1]
   }
```

Figure B.17 Special requests specific to this analysis: survival rates at specific BUN levels for males and females, and female/male contrasts.

Table B.5   Output of Special Requests

Explanatory Variables	Node[a]	Mean	sd	MC error[b]	2.5%	Median	97.5%
Sex	BUN			Female/Male Hazard Ratios			
	6 HRfm[1]	5.099	5.187	0.1644	0.7541	3.545	18.56
	14 HRfm[2]	1.619	0.7098	0.01883	0.6511	1.481	3.359
	21 HRfm[3]	1.025	0.3307	0.004344	0.5213	0.9776	1.807
	37 HRfm[4]	0.5829	0.2383	0.004737	0.2279	0.5461	1.146
	172 HRfm[5]	0.2007	0.2736	0.007766	0.009167	0.1135	0.918
				Female/Male 24-Month Relative Risks			
	6 RRfm[1]	3.852	3.119	0.104	0.7788	2.972	12.09
	14 RRfm[2]	1.438	0.4951	0.01366	0.7125	1.358	2.617
	21 RRfm[3]	1.007	0.2334	0.003134	0.615	0.9838	1.531
	37 RRfm[4]	0.7067	0.1815	0.003861	0.37	0.7009	1.078
	172 RRfm[5]	0.6222	0.2656	0.008312	0.1163	0.6473	0.999
				Male and Female 24-Month Survival			
F	6 Sr[1]	0.8631	0.07329	0.002712	0.6823	0.8779	0.9633
M	6 Sr[2]	0.6067	0.1669	0.005439	0.2328	0.6285	0.8665
F	14 Sr[3]	0.6805	0.07974	0.002664	0.5141	0.6848	0.8248
M	14 Sr[4]	0.5684	0.09397	0.002243	0.3799	0.5700	0.7455
F	21 Sr[5]	0.5357	0.07127	0.001667	0.3956	0.5359	0.6751
M	21 Sr[6]	0.543	0.08071	8.823E-4	0.3823	0.5433	0.6977
F	37 Sr[7]	0.2831	0.07449	0.001148	0.1525	0.2785	0.4411
M	37 Sr[8]	0.4992	0.1146	0.002497	0.2848	0.4965	0.7279
F	172 Sr[9]	0.01022	0.03385	9.349E-4	5.162E-15	1.336E-4	0.1008
M	172 Sr[10]	0.3851	0.2605	0.008167	0.008338	0.3586	0.8853

a. WinBUGS uses the word "node" to mean an unknown quantity in the model.
b. Based on 50,000 simulations.

BUN=172, the maximum level in the data set, females have about one-fifth the risk of males. Cumulative relative risk at 24 months, defined as $(1-S_{female}(24))/(1-S_{male}(24))$ shows a similar pattern. The rather large MC error is caused by "ill conditioning" of the design matrix. This could be cured by reexpressing log BUN as a deviation from its average.

Executing the WinBUGS Program.   All that remains is to assemble the program and data in a WinBUGS document and follow steps 1 through 24 in panels A through H on the following pages. Panel A shows how to assemble the model specification, data list, and data tables in a compound document. The data are in one list and two tables. The data list that specifies the number of cases (n=65), the number of requests (n=10) and the number of design variables (k=4). The Data table comprises the WinBUGS columns of Table B.3, and the requests table the WinBUGS columns of Table B.4. Instructions for pasting data tables into folds are in Section B.6.1. Remember to type the word END, followed by a carriage return, at the bottom of each data table.

The procedure documented in panels A through H shows every mouse-click of the process and looks quite time-consuming; however, in practice it takes less than a minute to get to the simulation step (step 22). The simulation step itself can take several minutes depending on the number of simulations requested and the speed of the analyst's computer. Output from the program is shown in Fig. B.16 and Table B.5. For a discussion of the meaning of the output, see Section 12.4.3.

A. Check the model

B. Load the data list and open the requests table

C. Load the requests table and close it

D. Load the Data Table, Close It, and Compile the Model

E. Verify compilation, load and generate initial values

F. Verify initialization
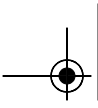
G. Enter the node names (repeat 19 and 20 for shape, Sr, HRfm, and RRfm)
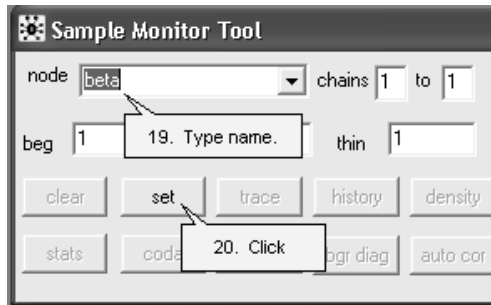
H. Generate the simulations and compute posterior moments and quantiles
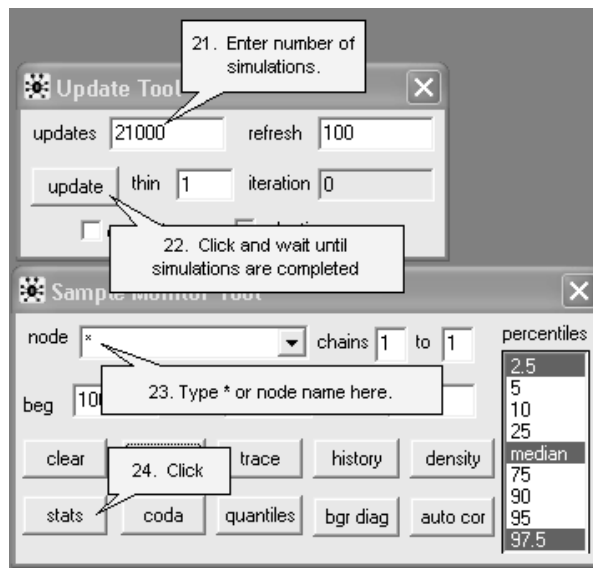
B.10   REFERENCES

Fryback D.G., Stout, N.K.,and Rosenberg, M.A., "An elementary introduction to Bayesian computing using WinBUGS," *International Journal of Technology Assessment in Health Care*, Vol 17, No. 1 (Winter 2001), pp. 98–113.

Krall, John M., Uthoff, Vincent A, and Harley, John B., "A Step-up Procedure for Selecting Variables Associated with Survival," *Biometrics*, Vol. 31, No. 1 (1975), pp. 49–57.

Spiegelhalter, D., Thomas, A., Best, N., and Lunn, D., *WinBUGS User Manual Version 1.4*, Cambridge, UK: MRC Biostatistics Unit (2002).