



Idaho National Laboratory

Bayesian Inference in Risk Assessment: Advanced Topics (P-502)

July 2009

**United States
Nuclear Regulatory Commission**



DISCLAIMER NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability of responsibility for any third party's use, or the results of such use, or any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

Bayesian Inference in Risk Assessment: Advanced Topics (P-502)

Course Presented by

***Dana Kelly
Dana.Kelly@inl.gov***

***Curtis Smith
Curtis.Smith@inl.gov***

***July 7 – 10, 2009
NRC Professional Development Center
Bethesda, MD***

Course Overview

- *P-502 is a follow-on to P-102*
 - *Assumes you have had P-102 or equivalent background in Bayesian inference for risk assessment*
- *No test*
- *Major topics (may not cover all topics)*
 - *Bayesian networks introduction*
 - *Models for recovery and repair*
 - *Models for population variability*
 - *Uncertain or fuzzy data*
 - *Time trends*
 - *Mixture priors*
 - *Model checking*

Course Overview

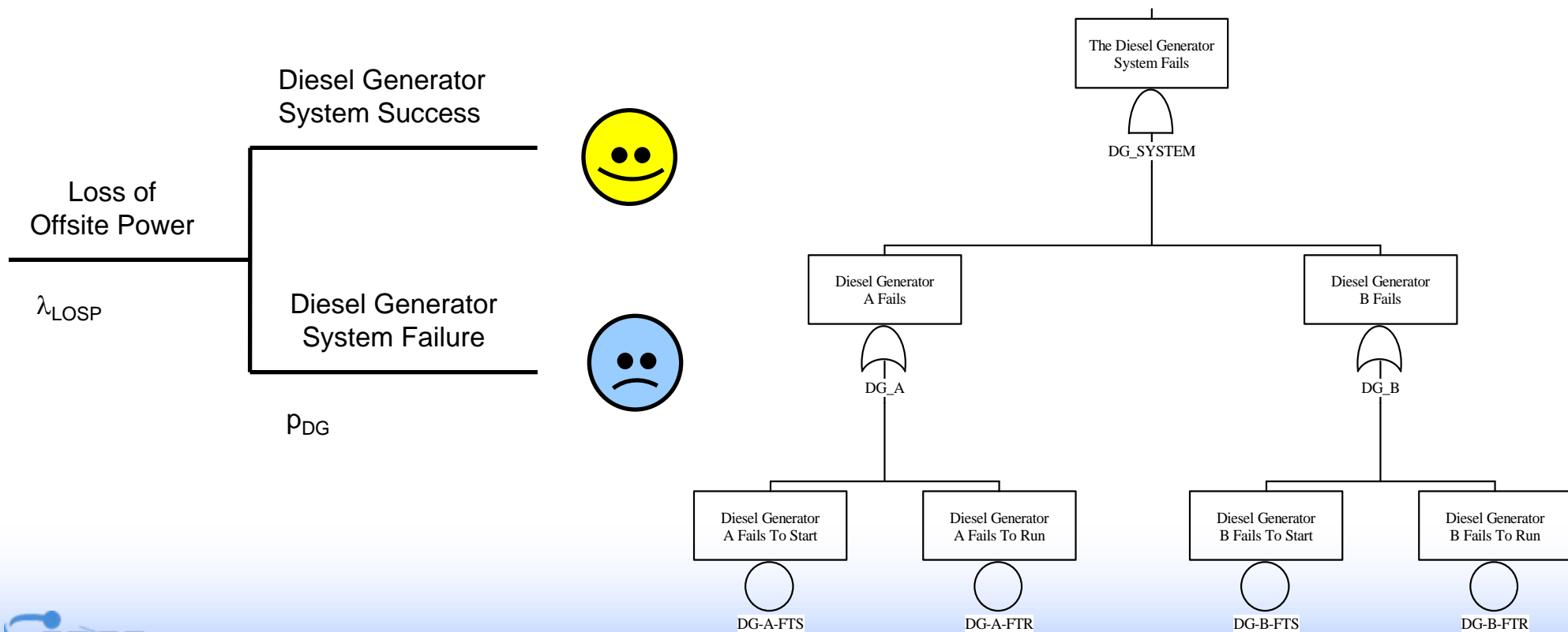
- *Course philosophy (or lack thereof) can be summed up in two ways:*
 1. *“Shut up and calculate.” – David Mermin, describing the Copenhagen interpretation of quantum mechanics.*
 2. *“Stop petting the mule and load the wagon.” – Abraham Lincoln*
- *Primary tool is WinBUGS*
 - *Implements Markov chain Monte Carlo (MCMC) sampling*

Loss of Offsite Power (LOSP) Example

- *The “LOSP example” was used as a central example throughout most of the P-102 course*
 - *We will refer to this example at times in P-502*
- *A system uses offsite power, but has two standby emergency diesel generators (EDGs)*
- *Occasionally offsite power is lost (an “initiating event”)*
 - *When this happens the EDGs are demanded to start and run.*
- *The system*
 - *Succeeds if either EDG starts and runs for six-hour mission time*
 - *Fails otherwise*

LOSP Example

- A PRA will have an event tree representing the scenario
 - Fault tree will represent the diesel generator failures



The Minimal Cut Sets

*LOSP*DG-A-FTS*DG-B-FTS* or

*LOSP*DG-A-FTS*DG-B-FTR* or

*LOSP*DG-A-FTR*DG-B-FTS* or

*LOSP*DG-A-FTR*DG-B-FTR*

Recovery of Offsite Power

- *Core damage can be averted if offsite power is recovered*
- *Assume traditional engineering analysis shows...*
 - *Recovery must occur by six hours to avert core damage*
- *Append nonrecovery event to minimal cut sets*
 - *This represents probability that offsite power is not recovered within six hours*

Recovery of EDG

- *It is possible that a failed EDG can be recovered in time to prevent reaching an undesired end state*
- *Assume recovery must occur within two hours after second EDG fails*
- *Append nonrecovery event to cut sets*

Recovered Cut Sets

*LOSP****DG-A-FTS*******DG-B-FTS*******OSP-NONREC*******EDG-NONREC** or
*LOSP****DG-A-FTS*******DG-B-FTR*******OSP-NONREC*******EDG-NONREC** or
*LOSP****DG-A-FTR*******DG-B-FTS*******OSP-NONREC*******EDG-NONREC** or
*LOSP****DG-A-FTR*******DG-B-FTR*******OSP-NONREC*******EDG-NONREC**

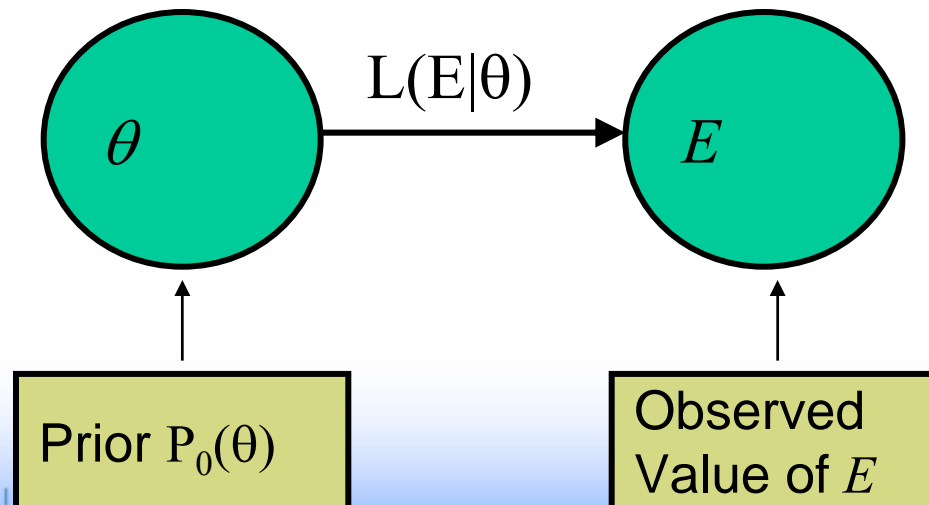
Section 2: Introduction to Bayesian Networks and WinBUGS

- *What are Bayesian networks and why are they useful?*
- *Bayesian networks allow “math-free” analysis of complex problems*
 - *Formulates problem as graphical model*
 - *Lessens mathematical burden on the analyst*
 - *Allows easier communication with non-specialists*
 - *Analogy from physics: two approaches to calculating interaction amplitude in quantum field theory (e.g., electron-electron scattering):*
 1. *Explicit path integral formulation, solved by perturbative expansion of integral*
 2. *Draw Feynman diagram for each term in expansion and apply rules to calculate amplitude*
 - *“Feynman [diagrams] brought quantum field theory to the masses.”
Julian Schwinger*

Introduction to Bayesian Networks and WinBUGS

- Networks are new way of representing old (simple) problems and allow easy analysis of new (complex) problems*
- Network representation of Bayes Theorem*

$$P_1(\theta | E) = \frac{L(E | \theta)P_o(\theta)}{\int L(E | \theta)P_o(\theta)d\theta}$$



Why WinBUGS?

- *Tool tailored for numerical analysis of Bayesian networks*
- *WinBUGS (OpenBUGS) is standard tool for Bayesian inference in wider statistical community*
 - *"WinBUGS...has become the most popular means for numerical investigation of Bayesian inference." Bayesian Statistics, An Introduction, 3rd Edition, by Peter Lee*
- *WinBUGS is free, open-source software*

Why WinBUGS?

- *WinBUGS is flexible, allowing new problems to be solved as they arise*
 - *Some tension between flexibility and ease of use*
 - *Batch mode operation possible for routine problems*

Prior Distributions for LOSP Example (from P-102 Course)

- $\lambda_{LOSP} \sim \text{gamma}(1.58, 43.96 \text{ reactor-critical years})$
 - From “Reevaluation of Station Blackout Risk at Nuclear Power Plants: NUREG/CR-6890, December 2005
 - Above result is composite from several subtypes of LOSP event
- $p_{FTS} \sim \text{beta}(0.957, 190)$
 - From S. A. Eide, “Historical Perspective On Failure Rates for US Commercial Reactor Components,” Reliability Engineering and System Safety, **80** (2003), pp. 123-132
- $\lambda_{FTR} \sim \text{gamma}(1.32, 1137 \text{ hrs})$
 - From Eide (2003)
 - Above result is composite of two rates

LOSP Example Data

- *The observed number of LOSP events over a period of time*
 - *1 initiating event in 9.2 operating years*
- *The observed number of failures out of a number of demands*
 - *1 failure to start in 75 demands*
- *The observed number of failures in an observed total operating time*
 - *0 failures to run in 146 running hours*

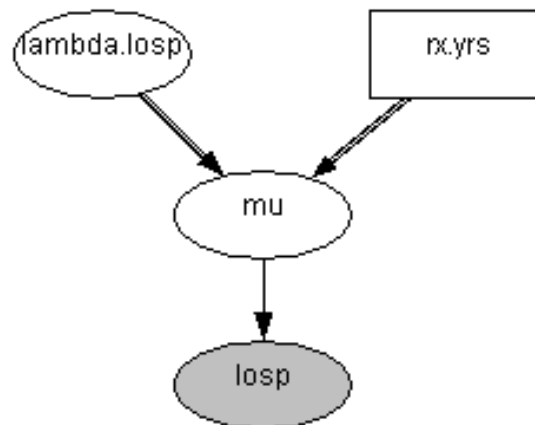
Summary of Bayesian Estimates for LOSP Example

Parameter	Distribution	Point Est. (Mean)	90% Interval
λ_{LOSP}	<i>Industry Prior</i>	$3.6E-2 \text{ yr}^{-1}$	$(4.6E-3, 9.2E-2) \text{ yr}^{-1}$
	<i>Posterior</i>	$4.9E-2 \text{ yr}^{-1}$	$(1.1E-2, 1.1E-1) \text{ yr}^{-1}$
ρ_{FTS}	<i>Industry Prior</i>	$5.0E-3$	$(2.3E-4, 1.5E-2)$
	<i>Posterior</i>	$7.4E-3$	$(1.3E-3, 1.8E-2)$
λ_{FTR}	<i>Industry Prior</i>	$1.2E-3 \text{ hr}^{-1}$	$(1.1E-4, 3.2E-3) \text{ hr}^{-1}$
	<i>Posterior</i>	$1.0E-3 \text{ hr}^{-1}$	$(9.6E-5, 2.8E-3) \text{ hr}^{-1}$

Posterior credible intervals generally shorter than those from data alone (i.e., confidence interval) or prior alone (prior credible interval)

Bayesian Network Model for LOSP Initiating Event

- Also referred to as directed acyclic graph (DAG)
 - DAG shows relationship of nodes (what influences what)
 - Oval represents stochastic node
 - Rectangle represents constant node
 - Arrows illustrate influence (“flow of information”)



Introduction to WinBUGS Package

- BUGS = Bayesian inference Using Gibbs Sampling
- Freely available software, download from
<http://mathstat.helsinki.fi/openbugs/>
- Source code available in Pascal (called OpenBUGS)
- Simulates posterior distribution directly using Markov Chain Monte Carlo (MCMC) sampling (covered later)



Introduction to WinBUGS

- *WinBUGS script (program) has four parts:*
 - *Model section*
 - *Data*
 - *Inits (initial values for model parameters)*
 - *Comments*
 - *Anything preceding model section or set off with #*
- *Script is compiled to process model and associated data*

Example WinBUGS Script (LOSP update.odc in BUGS folder)

- *LOSP frequency from example problem*
 - *Industry prior is $\text{gamma}(1.58, 43.96 \text{ yr})$*
 - *Data: 1 event in 9.2 yr*

Script to update LOSP frequency

```
model    {  
    losp ~ dpois(mu) #Poisson distribution for number of LOSP events  
    mu <- lambda.losp*rx.yrs #Parameter of Poisson distribution  
    lambda.losp ~ dgamma(1.58, 43.96) #Prior distribution for LOSP frequency  
}  
  
data  
list(losp=1, rx.yrs=9.2)
```

Example WinBUGS Script (LOSP update.odc in BUGS folder)

- *Dissect the script*

Model defined here between { }

Script to update LOSP frequency

```
model {  
  losp ~ dpois(mu) #Poisson distribution for number of LOSP events  
  mu <- lambda.losp*rx.yrs #Parameter of Poisson distribution  
  lambda.losp ~ dgamma(1.58, 43.96) #Prior distribution for LOSP frequency  
}
```

Note observable event and model have same name

```
data  
list(losp=1, rx.yrs=9.2)
```

Example WinBUGS Script (LOSP update.odc in BUGS folder)

- *Dissect the script*

Script to update LOSP frequency

```
model {  
  losp ~ dpois(mu) #Poisson distribution for number of LOSP events  
  mu <- lambda.losp*rx.yrs #Parameter of Poisson distribution  
  lambda.losp ~ dgamma(1.58, 43.96) #Prior distribution for LOSP frequency  
}
```

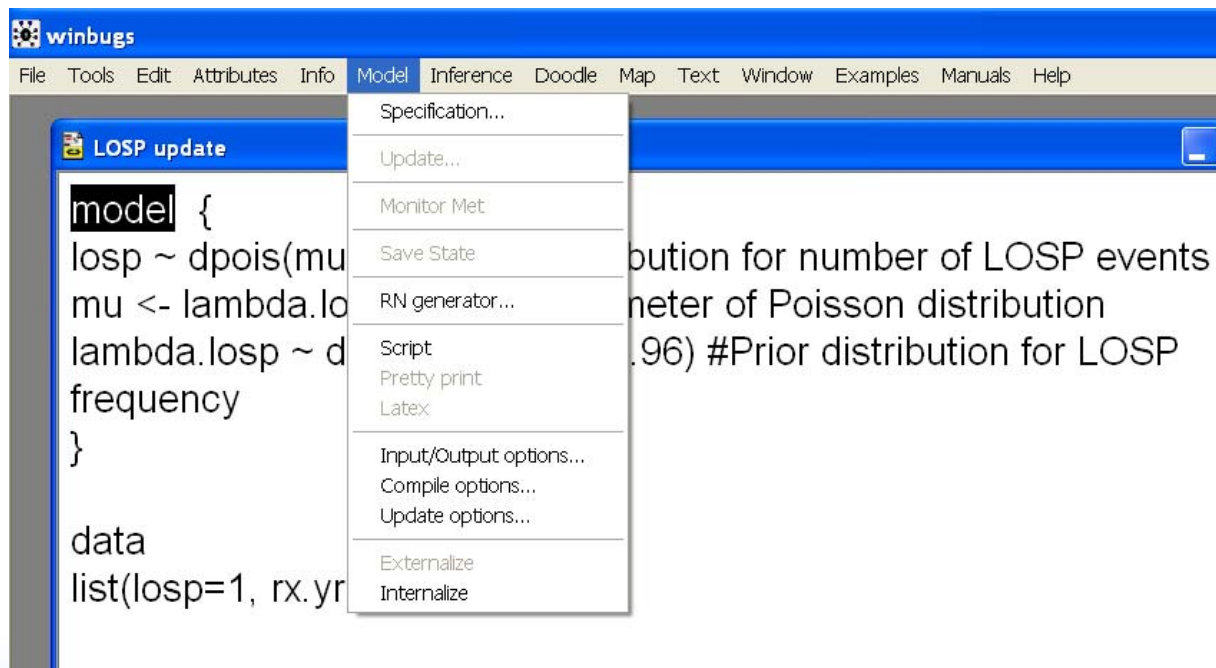
```
data  
list(losp=1, rx.yrs=9.2)
```

WinBUGS “knows” that prior needs to be “updated”

Data defined after list keyword

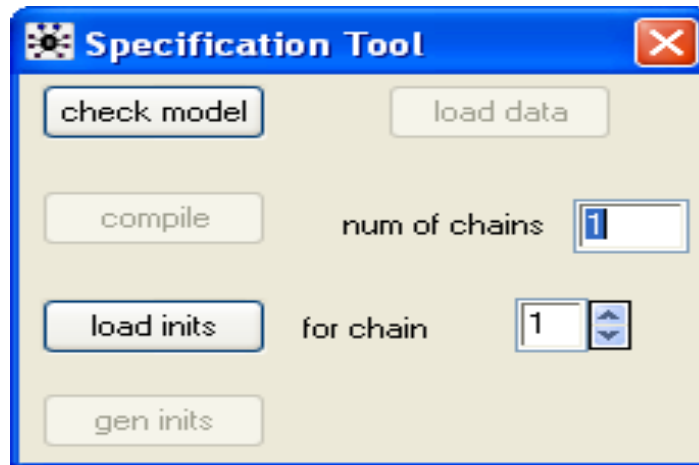
Running the Example Script

- *Double-click on “model”*
- *Select “Model – Specification” from toolbar:*



Running the Example Script

- *Specification toolbox appears on screen:*



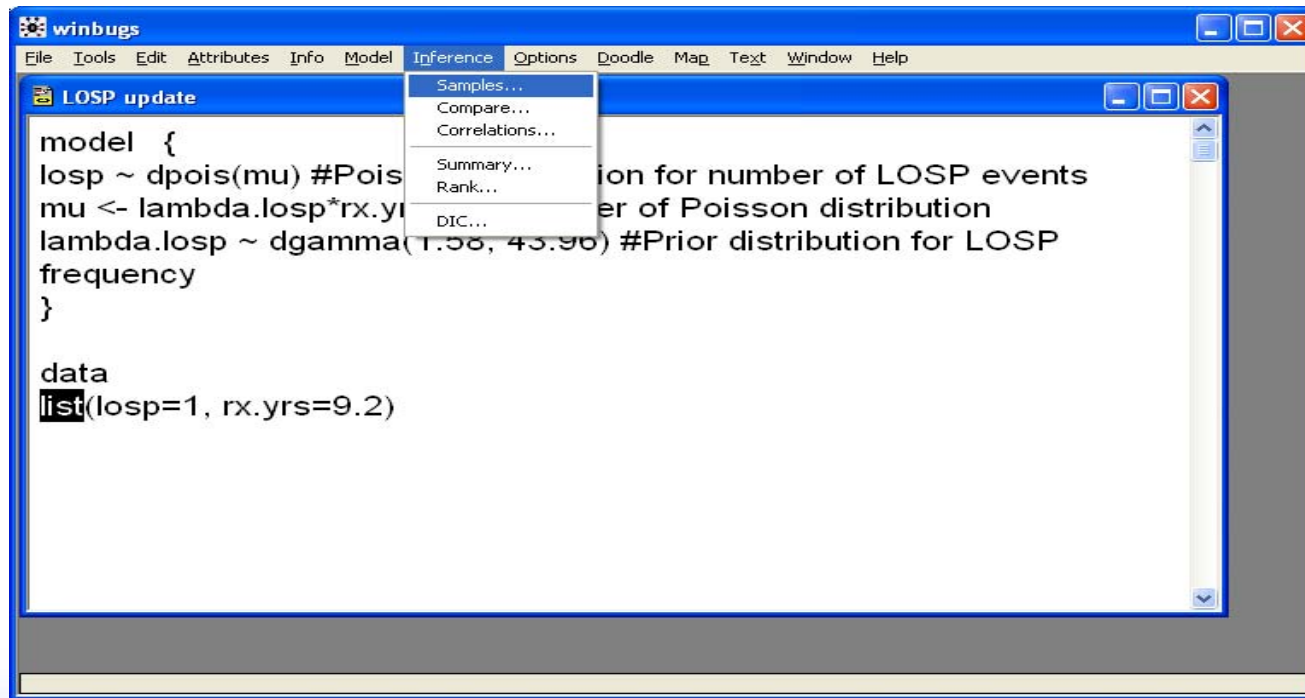
- *Select “Check Model”*
 - *WinBUGS should report “model is syntactically correct” (lower left corner of window)*

Running the Example Script

- *Double-click “list” in the data portion of the script, then select “load data”*
 - *WinBUGS reports “data loaded”*
- *Select “compile”*
 - *WinBUGS reports “model compiled”*
- *For such a simple model, we will let WinBUGS generate initial values*
 - *Select “gen inits”*
 - *WinBUGS reports “initial values generated, model initialized”*

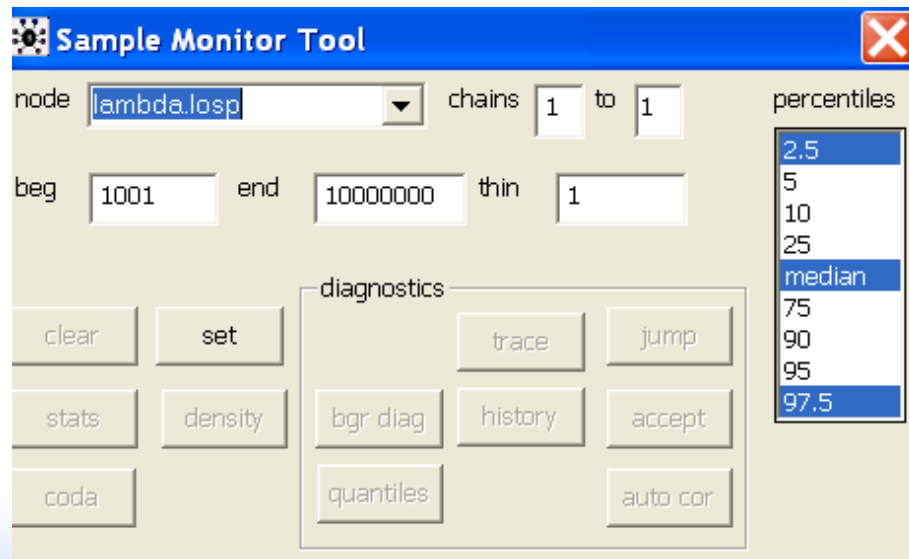
Running the Example Script

- *Close the specification toolbox*
- *Select “Inference – Samples” from the toolbar*



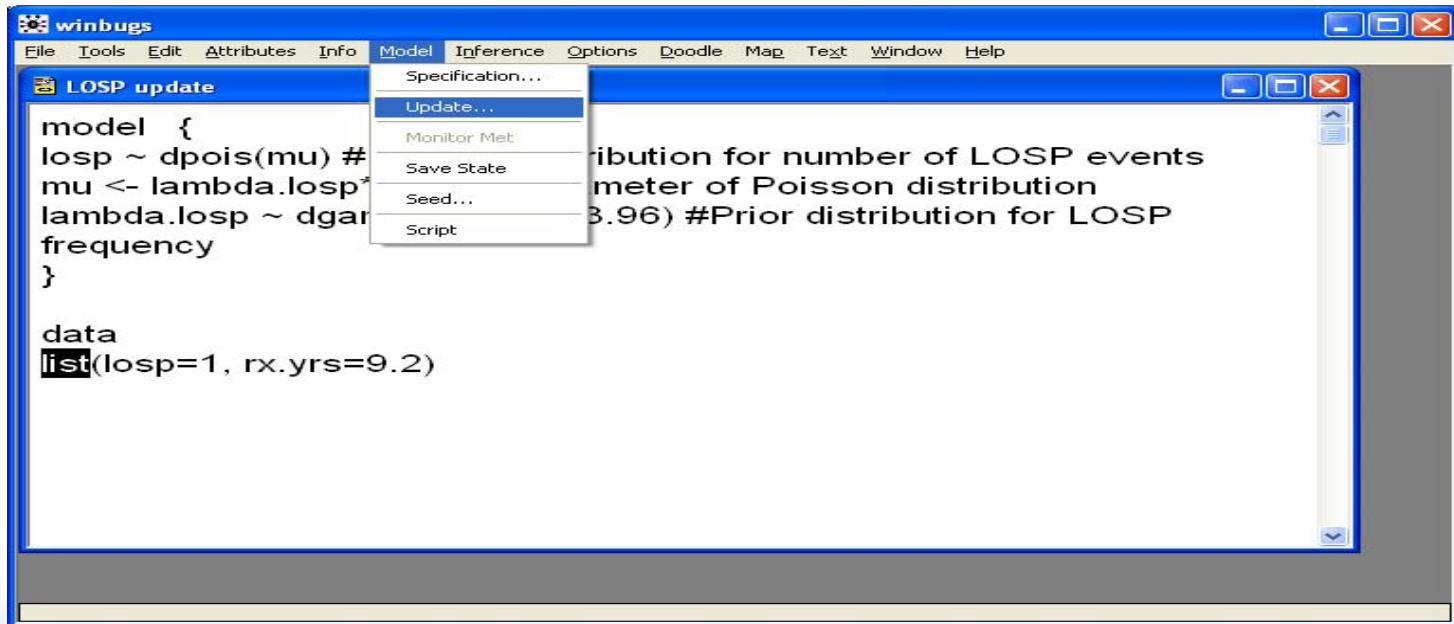
Running the Example Script

- *Type the variable name you want to monitor in the “node” box*
- *Select 1,000 burn-in samples (will discuss later)*
 - *Type 1001 in the “beg” box*



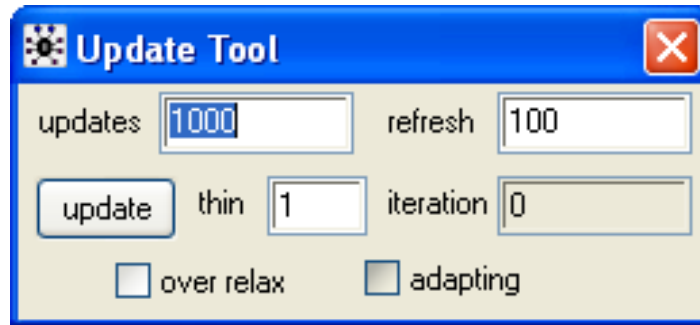
Running the Example Script

- Click “Set” button
- Close Sample Monitor Tool
- Select “Model – Update” from the toolbar



Running the Example Script

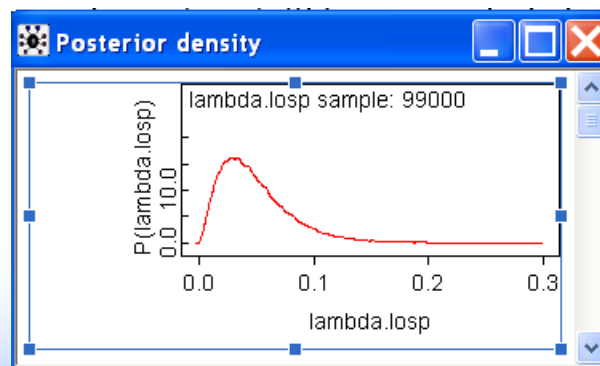
- *Update tool appears on screen*



- *Enter number of samples*
 - *100,000 is typical for simple problems*
- *Increase refresh from 100 to 1000 or 10000 to speed up sampling*
- *Select “Update”*
 - *WinBUGS reports “model is updating” and shows current iteration*

Running the Example Script

- *Close update tool when sampling is complete*
- *Re-open “Inference – Samples” from the toolbar*
- *Select “lambda.iosp” from the list of monitored nodes*
- *Select the desired percentiles from box on right*
 - *Check 5th, 50th, and 95th, by holding down Ctrl key*
- *Select “density” to get graph of posterior density*



Running the Example Script

- Select “stats” to get posterior mean and percentiles



A screenshot of a software window titled "Node statistics". The window contains a table with the following data:

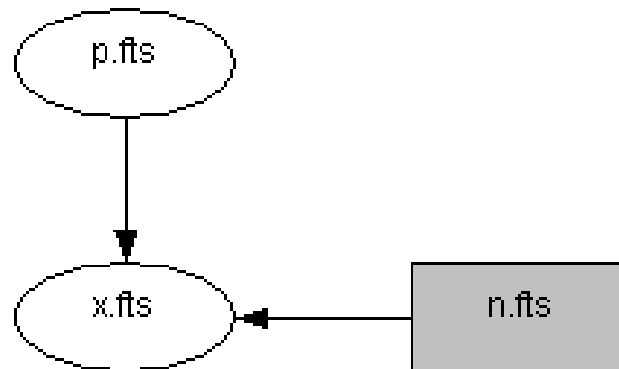
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
lambda.losp	0.04838	0.0302	9.43E-5	0.01146	0.04228	0.1064	1001	99000

- Compare with results from P-102

Parameter	Distribution	Mean	90% Interval
λ_{LOSP}	Posterior	$4.9E-2 \text{ yr}^{-1}$	$(1.1E-2, 1.1E-1) \text{ yr}^{-1}$

Bayesian Model for EDG FTS

- *Recall aleatory model for number of EDG failures is binomial distribution*
 - *Unknown parameter is p_{FTS}*
 - *DAG model*



WinBUGS Script for EDG FTS

```
model {  
  x.fts~ dbin(p.fts, n.fts) #Binomial distribution for number of failures  
  p.fts ~ dbeta(alpha, beta)#Beta prior for p.fts  
}
```

Data

```
list(x.fts=1, n.fts=75 alpha=0.957, beta=190)
```



Beta-binomial update.odc

Conjugate Prior Exercises with BUGS

1. *An analyst proposes a $\text{gamma}(0.1, 100 \text{ Rx-yr})$ prior distribution for small LOCA. Using BUGS, find the posterior mean and 90% interval based on 0 small LOCAs in 2,276 Rx-yrs.*
2. *Assume that failures to start of a diesel can be described by a binomial distribution with probability of failure on demand, p .*
 - a) *What is the constrained noninformative (CNI) prior if we want the mean to be 0.001?*
 - b) *Using the CNI prior from part (a), and 2 failures are observed in 500 demands, use BUGS to find the posterior mean and 95% credible interval for p .*

Noninformative Priors with WinBUGS

- *Recall Jeffreys prior distributions from P-102*
 - *Binomial model: $\text{beta}(0.5, 0.5)$*
 - *Poisson model: $\text{gamma}(0.5, 0)$*
 - *Exponential model: $\text{gamma}(0, 0)$*
- *Last two are not proper distributions*
- *Enter in WinBUGS as $\text{dgamma}(0.5, 0.0001)$ and $\text{dgamma}(0.0001, 0.0001)$*
- *Redo LOSP example problem from P-102 with Jeffreys priors*

Nonconjugate Priors – Lognormal Distribution

- *Generic databases often express uncertainty in terms of lognormal distribution*
- *Experts often provide order-of-magnitude estimates, represented well by lognormal distribution*
- *For these or other reasons, we may prefer a nonconjugate prior*

Lognormal Distribution

- *Definition of a lognormal distribution:*
 - X is **lognormal**(μ, σ^2) if $\ln(X)$ is **normal**(μ, σ^2)
- *Will encounter lognormal distribution in various areas of risk assessment*
 - **Often used** as a prior distribution in PRA, even though it is **not conjugate**
 - Sometimes used as likelihood function (e.g., LOSP recovery time)
 - Often used to model hazard (earthquake frequency) and fragility (probability of seismic failure) in seismic PRA

Facts About the Lognormal Distribution

- Median of X is e^μ
- Mean of X is $\exp(\mu + \sigma^2/2)$
- Variance of X is $(\text{mean})^2[\exp(\sigma^2) - 1]$
- Error factor (EF) is defined as $e^{1.645\sigma}$
- Other ways to write EF (applies only to lognormal)
 - $EF = 95^{\text{th}}/50^{\text{th}} = 50^{\text{th}}/5^{\text{th}} = (95^{\text{th}}/5^{\text{th}})^{1/2}$

$$\Pr(X \leq x) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right)$$

where Φ is tabulated in HOPE, Appendix C

– Can also use =LOGNORMDIST(x, μ, σ) in Excel

Facts About the Lognormal Distribution

- *Lognormal distribution is determined (in general) by*
 - μ and σ^2 (or $\tau = 1/\sigma^2$)
 - Median and mean
 - Median and variance
 - Mean and variance
 - Median and EF
 - Mean and EF
- *SAPHIRE uses mean and EF*

Example Problem with Nonconjugate Prior

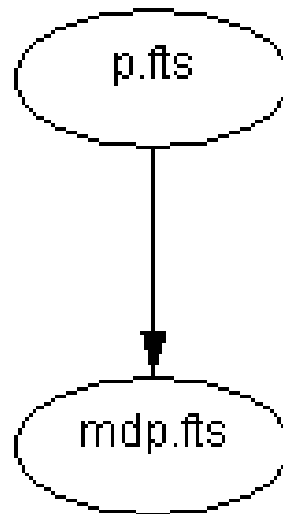
- *Interested in failure on demand for standby pump*
- *Generic database represents uncertainty in p via **lognormal** distribution with*
 - *mean of 0.003*
 - *error factor of 10*
- *Observe 0 failures in 36 demands*
- *What is posterior mean of p ?*

WinBUGS Example with Nonconjugate Prior

- *Generic prior is lognormal with mean = 0.003 and EF = 10*
- *Data is 0 failures in 36 demands*

```
model {  
  mdp.fts ~ dbin(p.fts, n.fts) #Binomial distribution for number of failures  
  p.fts ~ dlnorm(mu, tau) #Lognormal prior for p.fts  
  sigma <- log(EF)/1.645 #Solve for lognormal parameter, given EF  
  tau <- pow(sigma, -2) #Lognormal parameter required by WinBUGS  
  mu <- log(prior.mean) - pow(sigma, 2)/2 #Solve for other lognormal  
  parameter, using prior mean  
}  
  
data  
list(mdp.fts=0, n.fts=36, prior.mean=3.E-3, EF=10)
```

DAG Model



WinBUGS Example with Nonconjugate Prior

- *Run this model with 100,000 samples (1,000-sample burn in)*
 - *Posterior mean of $p.fts = 0.0022$*
 - *Compare with 0.0022 using Gaussian quadrature*
 - *Posterior 90% interval is $(1.1E-4, 7.9E-3)$*

BUGS Exercises with Nonconjugate Priors

1. *A prior for λ is taken as lognormal with a mean of 0.04/yr and an EF of 2.*
 - a) *Find the 5th and 95th percentiles of this prior distribution.*
 - b) *If 2 events are seen over a 7-yr period, update the prior distribution to find the posterior mean of λ .*
 - c) *What is the probability of seeing 2 or more events in the next 7 years? (Instructor-led)*
2. *The prior distribution for component unreliability on demand is log-uniform. That is, the logarithm (base 10) of the failure probability is uniformly distributed, in this case between -4 and -1. If 1 failure is observed in 119 demands, find the posterior mean and 90% interval of the unreliability.*

BUGS Exercises with Nonconjugate Priors

3. *A system has two redundant motor-driven pumps, each of which fails on demand with prior median probability 0.01 and upper bound probability 0.05, and a turbine-driven pump, which fails on demand with prior median probability 0.03, and upper bound probability 0.08. The motor-driven pumps each can supply 50% of the system needs, and the turbine-driven pump can supply 100% of the system needs. Assume that each pump fails independently of the others. Assume that the epistemic uncertainty in each pump's failure probability is lognormally distributed.*
- a) Write the expression for failure of the system, in symbols. Caution: think about state-of-knowledge correlation.*
 - b) What is the mean probability that the system fails on demand if there has been 1 turbine-driven pump failure in 28 demands and 0 motor-driven pump failures in 64 demands?*
 - c) What is the 95th percentile (upper bound) of this probability?*

Miscellaneous BUGS Exercises

1. *Pipes are designed to contain pressurized fluid. Let P_F denote the pressure at which a pipe fails. This failure pressure is treated as random, corresponding to variability in the strength of pipes. Assume that P_F is lognormally distributed, with median = 975 psig and $\sigma = 0.4$. If a pipe experiences a pressure of 1200 psig, what is the probability that the pipe fails? Repeat the above problem, but now model the system pressure (to which the pipe is exposed) as being normally distributed with a mean of 1200 psig and a standard deviation of 100 psig.*

Miscellaneous BUGS Exercises

2. *A diesel generator is required to have a failure probability (on demand) of 0.05 or less. The facility PRA has estimated the failure probability as being beta-distributed with a mean value of 0.027 and $\beta = 54$. If we observe 3 failures in 27 tests of the EDG, what is the probability that this criterion is met?*
3. *A licensee is updating the initiating event frequency for loss of turbine-building cooling water. Their prior distribution is lognormal with a mean of 0.02/yr and an error factor of 10. They have observed no losses in 27.5 Rx-yrs. Carry out the update and find the posterior mean and 90% interval.*

Miscellaneous BUGS Exercises

4. *In 60 demands of an AFW turbine-train, there were two failures to start. The unknown parameter of interest is $p = \text{Pr}(\text{failure to start})$. Assume a beta prior distribution for p , with $\alpha = 0.6$ and $\beta = 10.2$. What is the posterior mean? Find the 5th and 95th percentiles of the posterior distribution.*
5. *In 20 reactor-calendar years (17.3 reactor-critical years), there were 2 initiating events involving total loss of feedwater flow. The unknown parameter of interest is λ , the frequency of events per reactor-critical year. Assume a gamma prior distribution for λ , with $\alpha = 0.8$ and $\beta = 10.6$. Find the posterior mean and 95% credible interval for λ .*

Common-Cause Failure

- *Observe N failures in component group of size k*
 - *n_1 involve one component, n_2 involve 2, etc.*

$$\sum_{i=1}^k n_i = N$$

- *In alpha-factor model, failure counts have multinomial distribution with (vector) parameter α*

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$$

Common-Cause Failure

- *Conjugate prior for α is $\text{Dirichlet}(\theta)$*

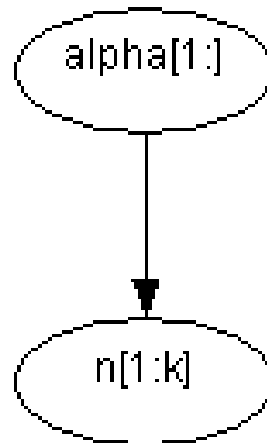
$$\theta = (\theta_1, \theta_2, \dots, \theta_k)$$

- *In equation form:*

$$f(n_1, n_2, \dots, n_k \mid \tilde{\alpha}) = \frac{(\sum_{i=1}^k n_i)!}{\prod_{i=1}^k n_i!} \prod_{i=1}^k \alpha_i^{n_i}$$

$$\pi_0(\tilde{\alpha}) = \frac{\Gamma(\sum_{i=1}^k \theta_i)}{\prod_{i=1}^k \Gamma(\theta_i)} \prod_{i=1}^k \alpha_i^{\theta_i - 1}$$

DAG Model



CCF Example

- *Group of size 3, $n_1 = 200$, $n_2 = 5$, $n_3 = 1$*
- *Usual noninformative prior is Dirichlet(1, 1, 1)*
 - *Multivariate analog of uniform distribution*
 - *Mean of each alpha-factor is 0.5*
- *Find posterior mean and 90% interval for alpha-factors*

CCF Example

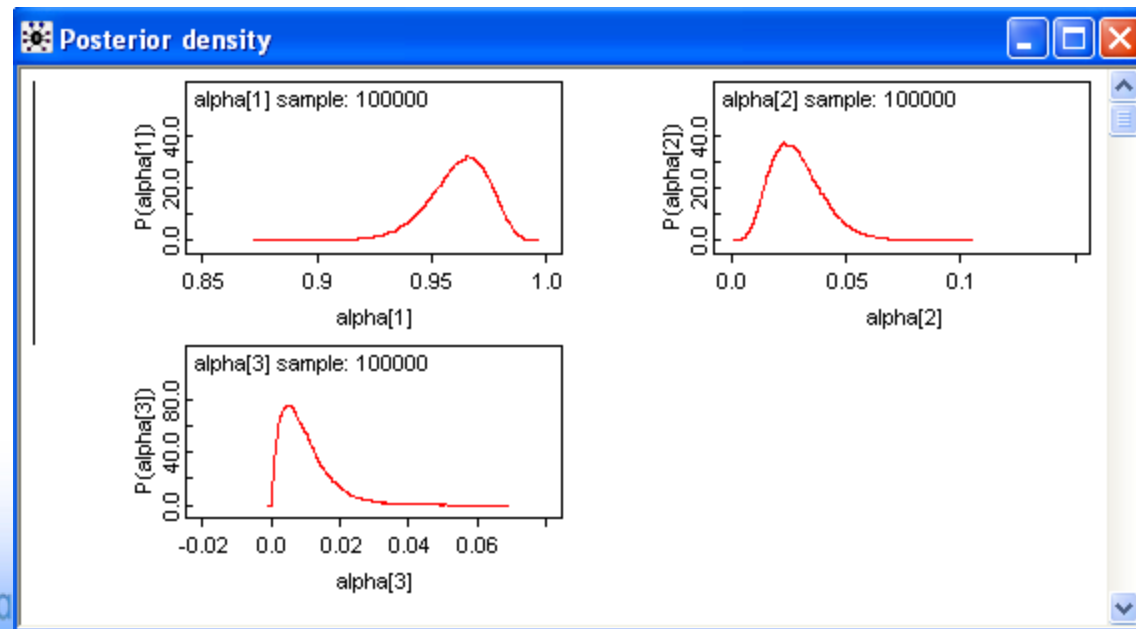
- *BUGS script*

```
model {  
  n[1:groupsize] ~ dmulti(alpha[1:groupsize], N)  
  N <- sum(n[1:groupsize])  
  alpha[1:groupsize] ~ ddirch(theta[])  
  theta[1] <- 1  
  theta[2] <- 1  
  theta[3] <- 1  
}  
  
data  
list(n=c(200, 5, 1), groupsize=3)
```

CCF Example

- Results*

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
alpha[1]	0.9618	0.0132	4.226E-5	0.9379	0.9632	0.9807	1001	100000
alpha[2]	0.02865	0.01148	3.695E-5	0.01267	0.02719	0.04962	1001	100000
alpha[3]	0.009572	0.006721	2.004E-5	0.001711	0.00807	0.02255	1001	100000



Section 3: Introduction to Markov Chain Monte Carlo Sampling (the Gee in BUGS)

- *Biggest problem in Bayesian inference is denominator of Bayes' Theorem*
 - *Multidimensional integral in multiparameter problems*
 - *Very hard to do, BCE**
 - *Very easy (and cheap!) to do now, but can be hazardous to your health!*
- *MCMC sampling works by generating a sample directly from the posterior distribution*
 - *MCMC is not medieval mathematics!*

*
BCE = Before Computer Era

Basic Premise of MCMC

- *Draw samples by running Markov chain constructed so that chain converges to joint posterior distribution*
- *Markov chain*
 - *Sequence of random variables X_0, X_1, X_2, \dots*
 - *Distribution of X_{n+1} depends only on X_n (Markov property)*
 - *After enough “time” (burn-in period), samples converge to stationary distribution (under certain regularity conditions)*
 - *Chain “forgets” its initial state*
 - *Stationary distribution is posterior distribution of interest*
 - *Can use samples taken after convergence (i.e., after “burn-in”) to estimate parameters of interest*
 - *Various methods exist to construct $f(x_{n+1}|x_n)$*
 - *Called “transition kernel”*

Gibbs Sampling

- *We want a sample from joint posterior distribution:
 $g(\theta_1, \theta_2, \dots, \theta_p | \mathbf{x})$*
- *Suppose we can sample from $g(\theta_i | \theta_j, \mathbf{x}; i \neq j)$*
- *Under certain conditions, these samples can be used to construct a sample from $g(\theta_1, \theta_2, \dots, \theta_p | \mathbf{x})$ and from any marginal distribution $g(\theta_i | \mathbf{x})$*

Example of Gibbs Sampling – Two Variables

- *Let $X \sim \text{binomial}(n, p)$, $p \sim \text{beta}(\alpha, \beta)$ (α, β known)*
 - *Familiar conjugate model for FTS*
- *Suppose we want $\Pr(X=x)$, not conditional upon p*
- *$f(x|p)$ is $\text{binomial}(p, n)$ – usual likelihood function*
- *$f(p|x)$ is $\text{beta}(\alpha + x, \beta + n - x)$ – posterior distribution*
- *Start with initial value p_o , sample x_o from $\text{binomial}(p_o, n)$*
- *Sample p_1 from $\text{beta}(\alpha + x_o, \beta + n - x_o)$*
- *Sample x_1 from $\text{binomial}(p_1, n)$*
- *Sample p_2 from $\text{beta}(\alpha + x_1, \beta + n - x_1)$, etc.*
- *Eventually, for large enough k , $x_k \sim f(x) = \Pr(X = x)$*

Example of Gibbs Sampling – Three Variables

- *In previous example, can write $f(x)$ analytically, so do not really need Gibbs sampling*
- *What if number of demands, n , is uncertain?*
 - *Model n with Poisson distribution: $n \sim \text{Poisson}(\mu)$*
- *Now cannot write $f(x)$ analytically*
- *Can write conditional distributions*
 - *$f(x|p, n)$ is binomial(p, n)*
 - *$f(p|x, n)$ is beta($\alpha + x, \beta + n - x$)*
 - *$f(n | x, p) \propto e^{-(1-p)\mu} \frac{[(1-p)\mu]^{n-x}}{(n-x)!}$*

Example of Gibbs Sampling – Three Variables

- *Start with initial values p_o and n_o*
- *Sample x_o from $\text{binomial}(p_o, n_o)$*
- *Sample p_1 from $\text{beta}(\alpha + x_o, \beta + n_o - x_o)$*
- *Sample n_1 from $f(n_1|x_o, p_o)$*
- *Sample x_1 from $\text{binomial}(p_1, n_1)$, etc.*

Extensions to Gibbs Sampling

- *Gibbs sampling requires knowing all conditional distributions*
- *Adaptive rejection sampling is more general form of Gibbs sampling*
 - *Only requires function proportional to conditional distribution*
 - *Works when conditional density is log-concave*
 - *Widely used in WinBUGS*
- *Other types of MCMC*
 - *Metropolis-Hastings sampling*
 - *Slice sampling*
 - *.....*

Metropolis-Hastings Sampling

- *Useful when conditional distributions are not available, so Gibbs sampling won't work*
- *Starts with proposal distribution, which is often normal*
- *Proposal density, $p(x, y)$*
 - *When chain is at point x , a value y is generated from $p(x, y)$*
- *If $p(x, y)$ is “reversible”: $\pi(x)p(x, y) = \pi(y)p(y, x)$, then π is desired stationary distribution and we're done*
 - *Usually, this is not the case*

Metropolis-Hastings Sampling

- *What we usually find is that process moves from x to y too often and from y to x too rarely:*
 - $\pi(x)p(x, y) > \pi(y)p(y, x)$
- *Correct for this by introducing probability $\alpha(x, y)$*
 - *Probability of moving from x to y*

$$\alpha(x, y) = \begin{cases} \min\left[\frac{\pi(y)p(y, x)}{\pi(x)p(x, y)}, 1\right] & \pi(x)p(x, y) > 0 \\ 1, & \text{otherwise} \end{cases}$$

Metropolis-Hastings Sampling

- *Some remarks about $\alpha(x, y)$:*
 - *Don't need to know $\pi(\bullet)$*
 - *Appears in both numerator and denominator*
 - *Depends on proposal density (more in a second)*
 - *If candidate value is rejected, chain stays at current value*
 - *If proposal density is symmetric, probability of move reduces to $\pi(y)/\pi(x)$*

Metropolis-Hastings Sampling

- *Summary of M-H algorithm:*
 1. *Repeat for $j = 1, 2, \dots, n$*
 2. *Generate y from $p(x^{(j)}, \bullet)$ and u from $\text{unif}(0, 1)$*
 3. *If $u \leq \alpha(x^{(j)}, y)$*
 - *Set $x^{(j+1)} = y$*
 - *Else*
 - *Set $x^{(j+1)} = x^{(j)}$*
 - *Return values $\{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$*

Metropolis-Hastings Sampling

- *Conditions for convergence (regularity conditions)*
 - *Markov chain must be irreducible and aperiodic*
 - *Must be able to move from x to y in finite number of steps with nonzero probability*
 - *Number of moves required is not a multiple of some integer*
 - *Conditions satisfied if proposal density is nonzero on same support as $\pi(\bullet)$*
 - *Usually also satisfied by density with restricted support*
 - *Uniform distribution around current point*

Metropolis-Hastings Sampling

- *Choosing proposal density*
 - *Often use normal distribution*
 - *Many other choices exist*
 - *Care must be taken in “tuning” the proposal density location and spread as the sampling progresses*
- *Note that Gibbs sampling is Metropolis-Hastings sampling with an acceptance rate of 1.0*
 - *Gibbs sampling is more efficient when all the necessary conditional distributions are available*
- *See paper by Chib and Greenberg in References folder for more details*

Slice Sampling

- *Metropolis-Hastings sampling can be very inefficient if the rejection rate for moves is too high*
- *Various alternatives have been proposed*
- *WinBUGS makes much use of “slice sampling”*
 - *See paper by Radford Neal in References folder for all the details*

Slice Sampling

- *Basic idea*
 1. *Find function $f(x)$ proportional to desired density*
 2. *For current point, $x^{(0)}$, generate uniform sample, y , from $(0, f(x^{(0)}))$*
 3. *Define horizontal slice through this sample*
 - *Can use f^{-1} to do this*
 4. *Draw new point, $x^{(1)}$, uniformly from this horizontal slice*

Slice Sampling

- *Example: slice sampling from normal distribution*
 1. *Choose initial x , say $x = 0$*
 2. *Choose y uniformly from $(0, \Phi(x))$*
 3. *Choose new x uniformly from $\left[-2\ln(y\sqrt{2\pi}), 2\ln(y\sqrt{2\pi})\right]$*
 4. *Repeat*

Practical Issues for MCMC

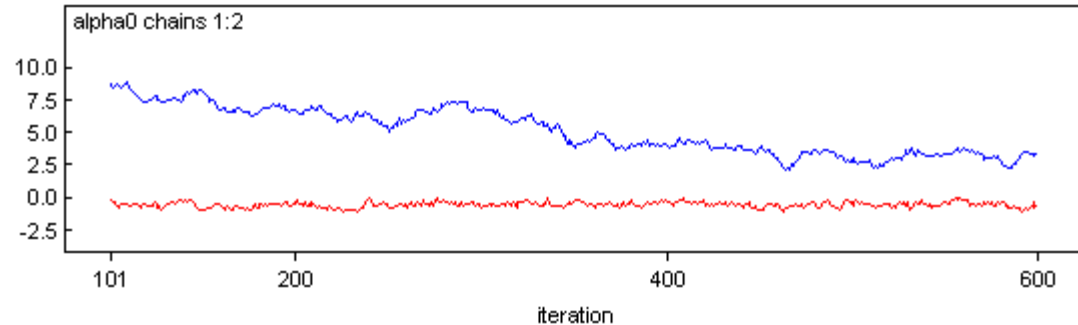
- *MCMC is both art and science*
- *Must choose initial values for sampling*
 - *Best to pick value where posterior density is large*
 - *MLEs often a good choice, if available*
 - *Usually not an issue except for very complicated models*
 - *Less complicated models will usually converge quickly even with poor estimates*
- *Samples values are dependent, measured by autocorrelation*
 - *Dependence should decrease as distance between values (lag) increases*
 - *High lag autocorrelation can mean chains not exploring full range of posterior distribution*

Practical Issues for MCMC

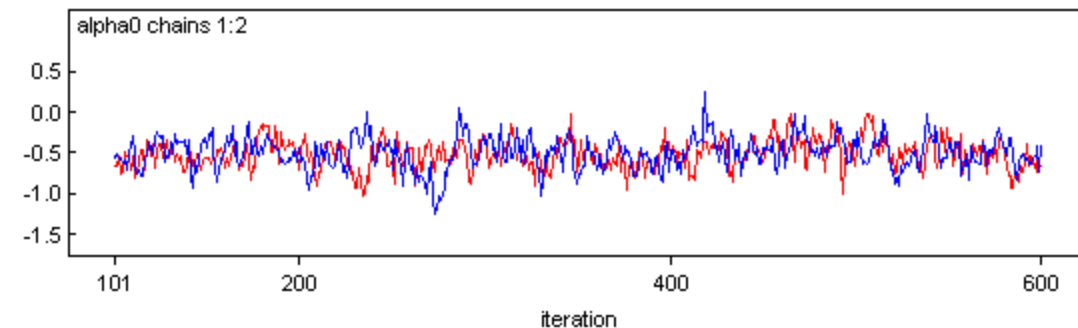
- *How to tell when Markov chain has converged*
 - *How many samples needed for burn in?*
 - *Has sampler obtained good coverage of posterior?*
 - *How many samples needed for desired precision?*

How Many Samples for Burn In?

- *Run multiple chains, starting at different points*
- *Look for good mixing of chains*
 - *Poor mixing*



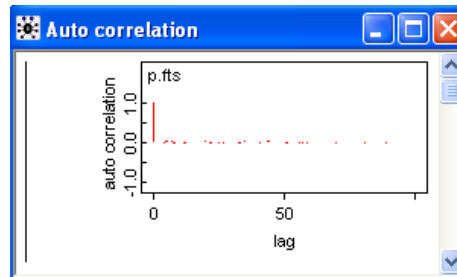
- *Good mixing*



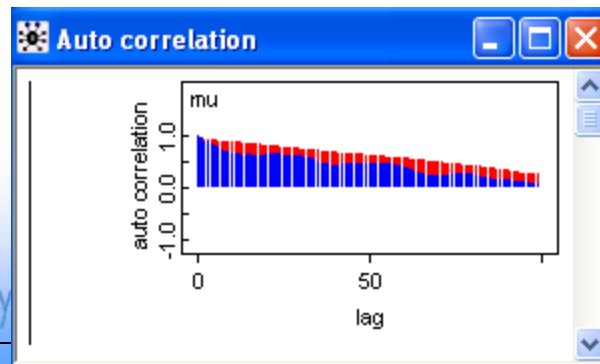
Good Coverage of Posterior

- *Run multiple chains, starting at widely dispersed points in the parameter space*
- *Look for good mixing of chains*
- *Can check to see if plot of autocorrelation vs. lag approaches 0 as lag increases*

– *Good:*



– *Less good:*



How Many Samples After Convergence?

- *WinBUGS manual suggests running until Monte Carlo error is 5% or less of sample standard deviation for each parameter*
- *None of the three aspects of convergence tends to be a problem for typical risk assessment models*
 - *But do not forget about it!*

Convergence Diagnostics - BGR

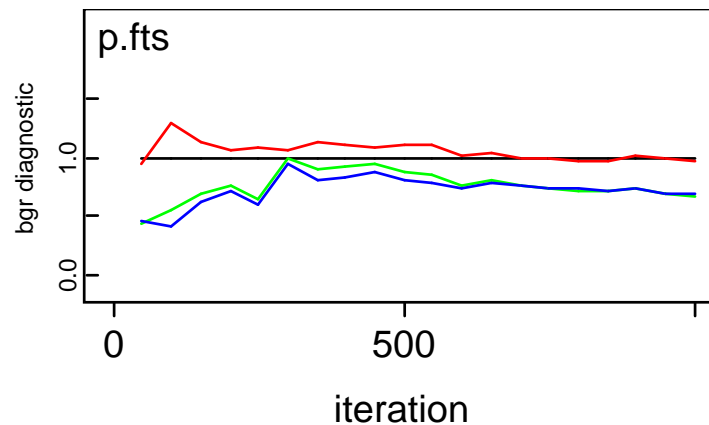
- *Brooks-Gelman-Rubin (BGR) statistic is implemented in WinBUGS*
 - *Overview*
 - *Compares between-chain variance to within-chain variance*
 - *If chains have converged, all should have same within-chain variance*
 - *BGR ≈ 1 indicates convergence*
 - *Must run at least 2 chains to calculate BGR statistic*

Convergence Diagnostics - BGR

- *Details*
 - *Between-chain estimate is green*
 - *Within-chain estimate is blue*
 - *Ratio (between/within) is red*
 - *Ratio expected to start out > 1 and converge to 1*
 - *Rule of thumb: $R < 1.2$ for convergence*
 - *Also want between-chain and within-chain estimates to be stable*
 - *Double-click on BGR graph, then <Ctrl>-left click gives values of R*

Convergence Diagnostics - BGR

- *Example*

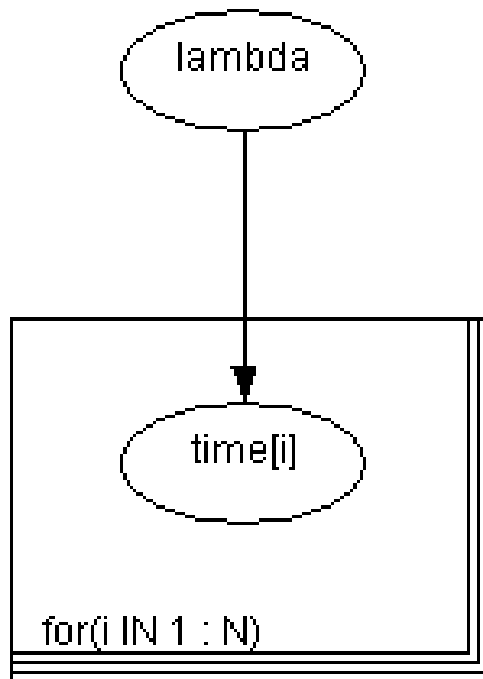


- *Indicates convergence*

Section 4: Models for Recovery and Repair

- *Recall from P-102 that simplest aleatory model for time as a random variable is the exponential distribution*
 $T \sim \exp(\lambda)$
- *Unknown parameter is λ , which is recovery rate in present context*
 - $MTTR = 1/\lambda$
- *Data is series of times for recovery or repair, assumed to be random sample from $\exp(\lambda)$ distribution*
- *For Bayesian inference, need prior distribution for λ*
 - *Conjugate prior is gamma distribution*
 - *Jeffreys prior is (improper) $\text{gamma}(0, 0)$*
 - *Nonconjugate prior*

DAG Representation

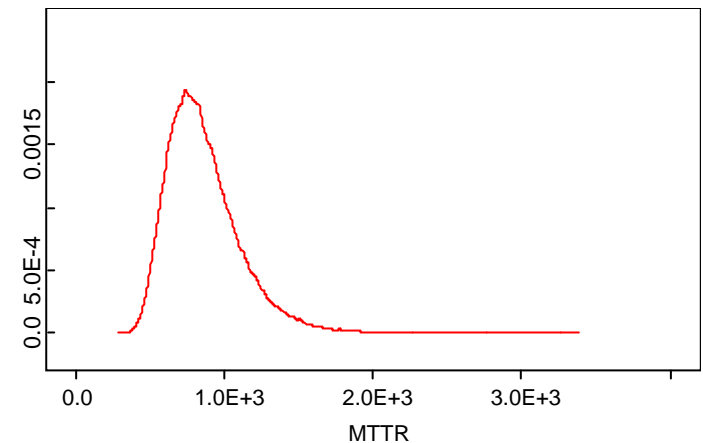
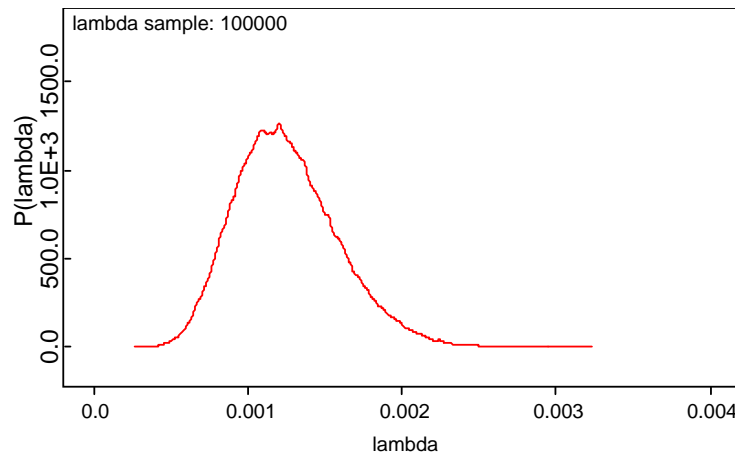


WinBUGS Script

```
model {  
  for(i in 1:N) {  
    time[i] ~ dexp(lambda) #Exponential distribution for each time  
  }  
  lambda ~ dgamma(0.0001, 0.0001) #Jeffreys prior for lambda  
  MTTR <- 1/lambda  
}  
data  
list(time=c(105, 1, 1263, 72, 37, 814, 1.5, 211, 330, 7929, 296, 1, 120, 1), N = 14)  
  
inits  
list(lambda=1.E-3)  
list(lambda=5.E-3)
```

Results

Node statistics								
I	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
MTTR	860.5	247.9	0.7589	539.3	819.7	1320.0	1001	100000
lambda	0.001252	3.357E-4	1.018E-6	7.574E-4	0.00122	0.001854	1001	100000



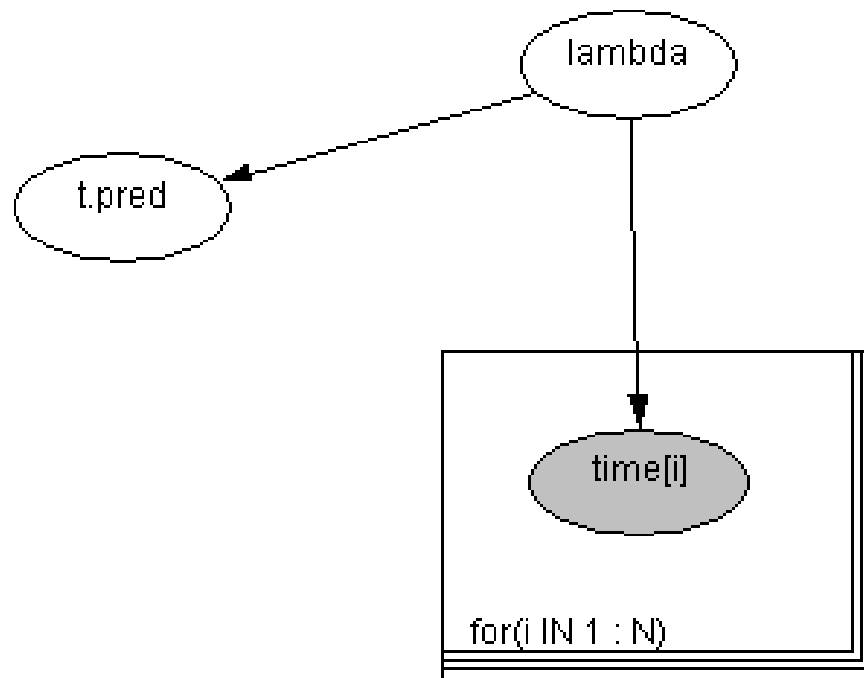
What is Distribution of New Time, Given Observed Times?

- *The distribution for MTTR found previously describes the uncertainty in the mean time to recovery/repair*
- *Distribution of new time is obtained by averaging over the posterior distribution for λ :*

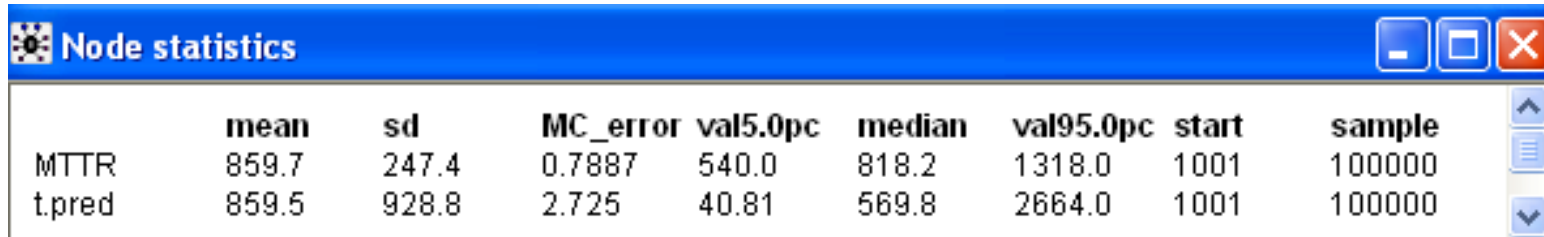
$$\begin{aligned} f(t_{\text{new}} | t_1, \dots, t_N) &= \int_0^{\infty} f(t_{\text{new}} | \lambda) g(t_1, \dots, t_N) d\lambda \\ &= \int_0^{\infty} \lambda e^{-\lambda t_{\text{new}}} g(\lambda | t_1, \dots, t_N) d\lambda \end{aligned}$$

- *Called posterior predictive distribution*
 - *Will use this extensively for model-checking*

DAG Model for Predicted Time



Results for Predicted Time

A screenshot of a software window titled "Node statistics". The window has a blue header bar with the title and standard window controls (minimize, maximize, close). Below the header is a table with two rows of data. The first row is for "MTTR" and the second row is for "t.pred". The columns are: mean, sd, MC_error, val5.0pc, median, val95.0pc, start, and sample. The "t.pred" row shows a significantly wider uncertainty interval (val5.0pc to val95.0pc) compared to the "MTTR" row.

	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
MTTR	859.7	247.4	0.7887	540.0	818.2	1318.0	1001	100000
t.pred	859.5	928.8	2.725	40.81	569.8	2664.0	1001	100000

- *Note wider uncertainty interval for t.pred*

Specifying a Model

- *From experience, recovery time does not tend to be exponential*
 - *Recovery rate $[h(t)]$ often decreases with time after the initiator*
 - $h(t) = f(t)/[1 - F(t)]$
 - *Common models for non-constant recovery rate*
 - *Weibull*
 - *Gamma*
 - *Lognormal*

Alternatives to Exponential Model

- $T \sim \text{Weibull}(\alpha, \lambda)$
 - $f(t) = \alpha \lambda t^{\alpha-1} \exp(-\lambda t^\alpha)$
 - $F(t) = 1 - \exp(-\lambda t^\alpha)$
 - $E(T) = \lambda^{-1/\alpha} \Gamma(1 + 1/\alpha)$
 - For $\alpha = 1$, reduces to exponential(λ)
 - Recovery rate is function of time
 - $h(t) = \alpha \lambda t^{\alpha-1}$
 - $\alpha < 1 \Rightarrow h(t)$ decreasing with time
 - $\alpha > 1 \Rightarrow h(t)$ increasing with time

Alternative Weibull Parameters

- *Previous slide uses parameters defined for WinBUGS, which were convenient for Bayesian analysis in the days before MCMC*
- *Alternative parameterization uses $\beta = \lambda^{-1/\alpha}$, giving*
 - $f(t) = (\alpha/\beta)(t/\beta)^{\alpha-1}\exp[-(t/\beta)^\alpha]$
 - $F(t) = 1 - \exp [-(t/\beta)^\alpha]$
 - $E(T) = \beta\Gamma(1 + 1/\alpha)$
 - $h(t) = (\alpha/\beta)(t/\beta)^{\alpha-1}$

Alternatives to Exponential Model

- $T \sim \text{gamma}(\alpha, \beta)$
- $$f(t) = \frac{\beta^\alpha t^{\alpha-1} e^{-\beta t}}{\Gamma(\alpha)}$$
- $E(T) = \alpha/\beta$
- For $\alpha = 1$, reduces to exponential(β)
- Recovery rate is function of time
 - Cannot be written in closed form
 - $\alpha < 1 \Rightarrow h(t)$ decreasing with time
 - $\alpha > 1 \Rightarrow h(t)$ increasing with time

Alternatives to Exponential Model

- $T \sim \text{lognormal}(\mu, \sigma^2)$
- $f(t) = \frac{1}{\sqrt{2\pi\sigma t}} \exp\left[-\frac{(\ln t - \mu)^2}{2\sigma^2}\right]$
- $F(t) = \Phi[(\ln t - \mu)/\sigma]$
- $E(T) = \exp(\mu + \sigma^2/2)$
- *Recovery rate is function of time*
 - *Increases from 0 initially (often very fast), then decreases, approaching 0 asymptotically*
 - *Approach to 0 can be very slow*
 - *Useful when early recoveries dominate*

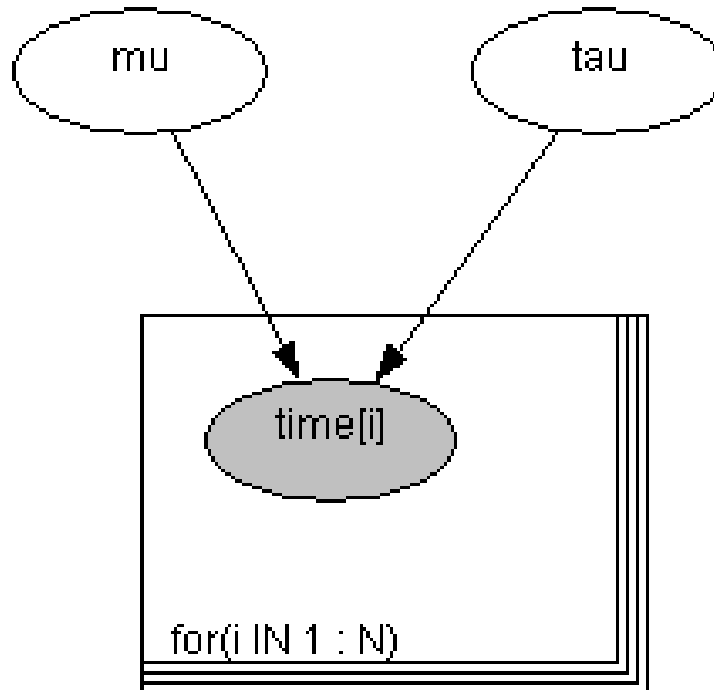
Alternatives to Exponential Model

- *Note that all alternatives considered have two unknown parameters*
 - *Weibull(α, λ)*
 - *Gamma(α, β)*
 - *Lognormal(μ, σ) or Lognormal(μ, τ)*
- *Need to specify joint prior distribution, e.g., $g(\mu, \tau)$*

Specifying Joint Prior via Bayesian Network

- *Much effort was spent in the past developing complicated joint prior distributions that reflected correlation between two parameters*
- *With Bayesian network, we can specify independent priors and not worry about this*
 - *In Bayesian network, parameters are independent until data is observed*
 - *Posterior distribution will reflect proper dependence between parameters*

Example DAG for Lognormal Model



Choosing Priors for μ and τ

- *Will use independent priors*
- *Range of μ is $-\infty$ to ∞*
- *Range of τ is 0 to ∞*
- *If prior information is available, can use this*
- *Noninformative priors are often used*
 - *Do not use Jeffreys prior in this case*
 - *Use Laplace (infinite uniform) prior or very diffuse normal prior for μ*
 - *Use finite uniform prior for σ*

Choosing Priors for μ and τ

- *Easier to think in terms of $\sigma = \tau^{-0.5}$*
 - *Recall that $EF = e^{1.645\sigma}$ for the lognormal distribution*
 - *$\sigma = 10$ corresponds to $EF > 10^7$*

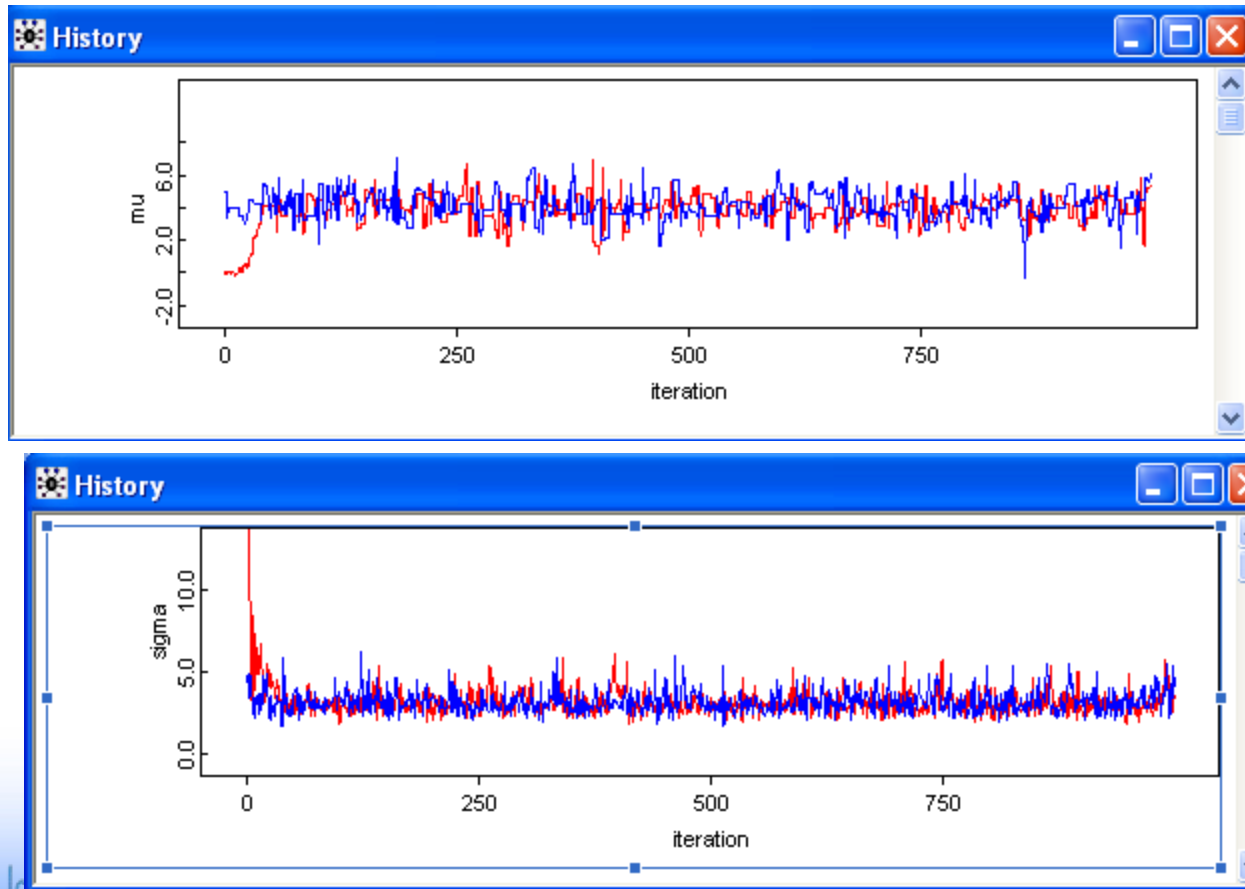
```
model
{
    for (i in 1 : N) {
time.rec[i] ~ dlnorm(mu, tau) #Lognormal model for recovery time
    }
mu ~ dflat() #Diffuse prior for mu
sigma ~ dgamma(0.0001, 0.0001)
tau <- pow(sigma, -2)
}
```

Running the Script

- *We are interested in inference for two parameters, so convergence is a concern*
 - *Will run two chains starting at different points*
- *Check history plots for mu and sigma*
- *Check BGR statistic*

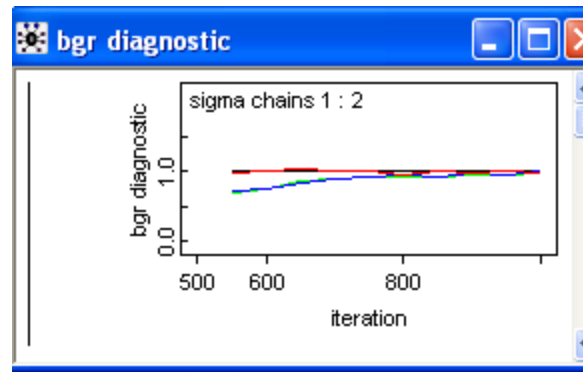
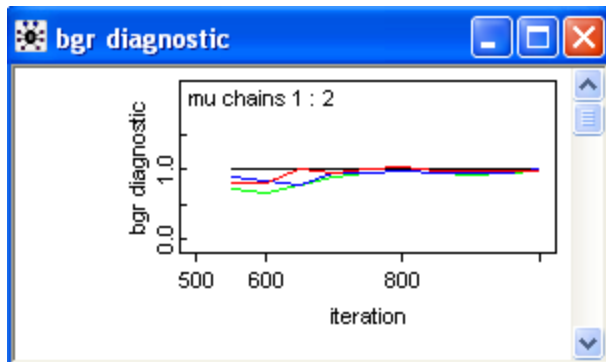
Results

- *History plots after 1,000 samples*



Results

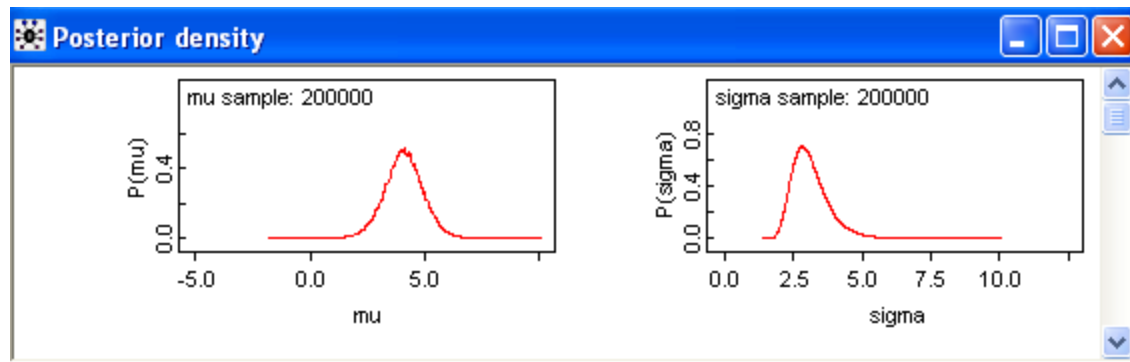
- *BGR plots after 1,000 samples*



Results

- *Discard first 1,000 samples for burn-in, run additional 100,000 samples to estimate μ and σ :*

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
mu	4.097	0.8525	0.004145	2.704	4.098	5.483	1001	200000
sigma	3.118	0.6688	0.002175	2.237	3.013	4.356	1001	200000



Exercises

Use the data in “times-1.txt” in the BUGS folder for the following questions.

- 1. Assuming an exponential aleatory model, and a lognormal prior for λ with a mean of 1 and error factor of 3, find the posterior mean and 90% interval for λ .*
- 2. Assume a lognormal aleatory model with independent, diffuse priors on the lognormal parameters. Find the posterior mean and 90% interval for λ .*

Exercises

3. *The file “EDG repair times.txt” in the BUGS folder contains 18 repair times recorded when a single EDG failed.*
- a) *Find the posterior means of the parameters of a Weibull aleatory model for these repair times. Use independent, diffuse priors for the Weibull parameters.*
 - b) *Generate two, independent predicted repair times from the posterior distribution. What is the mean and 90% credible interval for each of these times?*
 - c) *Define a new variable that is the minimum of each of the two times generated in part (b). What is the mean and 90% credible interval for this minimum time?*

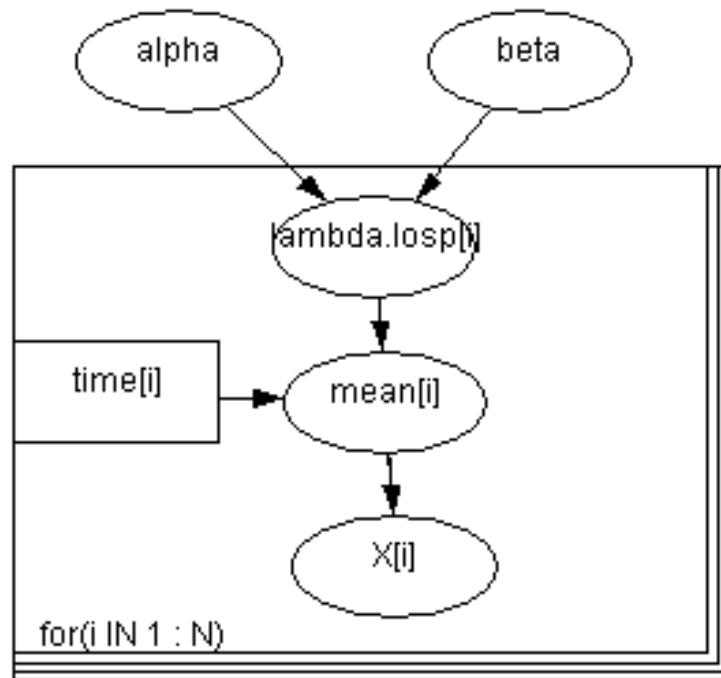
Section 5: Modeling Population Variability

- *Problem: what to do when information cannot be pooled*
 - *Across components*
 - *Across plants*
 - *Across time*
 - *Across experts*
- *Inappropriately pooling information leads to underestimation of uncertainty*

Example: Variability in λ_{LOSP}

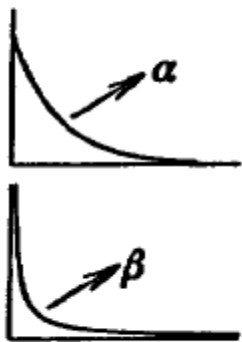
- X_i is number of LOSP events at Plant i in time t_i
- Assume variability in λ_{LOSP} from plant to plant can be modeled with a gamma distribution
 - $\lambda_{\text{LOSP}, i} \sim \text{gamma}(\alpha, \beta)$, $i = 1, 2, \dots, n$
- There are $(n + 2)$ unknown parameters: $\lambda_{\text{LOSP}, i}$, α , β
- In past problems we specified α and β , and there was only one λ_{LOSP}
 - α and β are now **uncertain**

DAG Model

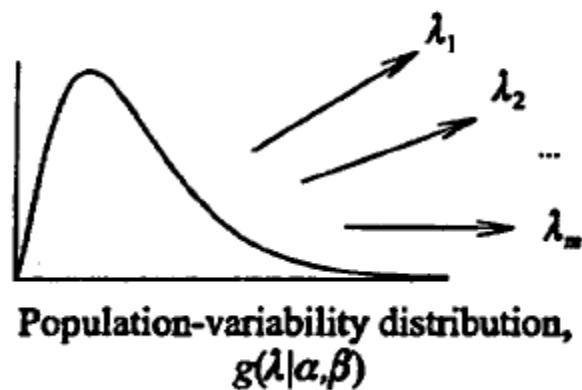


Another Representation of the Problem

Level 0: Hyperprior



Level 1: Population variability



Level 2: Plant-specific data

Plant 1, $\lambda_1, t_1 \longrightarrow X_1 \sim \text{Poisson}(\lambda_1 t_1)$
Plant 2, $\lambda_2, t_2 \longrightarrow X_2 \sim \text{Poisson}(\lambda_2 t_2)$
...
Plant m , $\lambda_m, t_m \longrightarrow X_m \sim \text{Poisson}(\lambda_m t_m)$

Hierarchical Bayes

- *Previous approach is called “hierarchical Bayes”*
 - *Bayesian approach is to specify prior in stages (hierarchies)*
 - *First stage is gamma(α, β) prior for $\lambda_{\text{LOSP}, i}$*
 - *Second stage is joint prior $\pi_o(\alpha, \beta)$*
 - *Called hyperprior*
 - *α, β called hyperparameters*
 - *Often use vague (noninformative) prior for hyperparameters*
 - *Two stages typical, but can model three or more*

$$\pi_o(\lambda) = \iint \pi_o(\lambda | \alpha, \beta) \pi_o(\alpha, \beta) d\alpha d\beta$$

Overall prior

First-stage
prior

Second-stage
prior
(hyperprior)

Example: Variability in LOSP Frequency

- Data*

<i>Plant</i>	<i>Failures</i>	<i>Time (yrs)</i>
<i>Plant 1</i>	2	15.986
<i>Plant 2</i>	1	16.878
<i>Plant 3</i>	1	18.146
<i>Plant 4</i>	1	18.636
<i>Plant 5</i>	2	18.792
<i>Plant 6</i>	0	18.976
<i>Plant 7</i>	12	18.522
<i>Plant 8</i>	5	19.04
<i>Plant 9</i>	0	18.784
<i>Plant 10</i>	3	18.868
<i>Plant 11</i>	0	19.232

WinBUGS Model

```
model {  
  for (i in 1 : N) {  
    lambda.losp[i] ~ dgamma(alpha, beta) #Model variability in LOSP frequency  
    mean[i] <- lambda.losp[i] * time[i] #Poisson parameter for each plant  
    x[i] ~ dpois(mean[i]) #Poisson dist. for events at each plant  
  }  
  lambda.ind ~ dgamma(alpha, beta)  
  alpha ~ dgamma(0.0001, 0.0001) #Vague hyperprior for alpha  
  beta ~ dgamma(0.0001, 0.0001) #Vague hyperprior for beta  
}  
data  
Open "LOSP frequencies.txt"  
  
inits  
list(alpha=1, beta=1)  
list(alpha=0.5, beta=5)
```

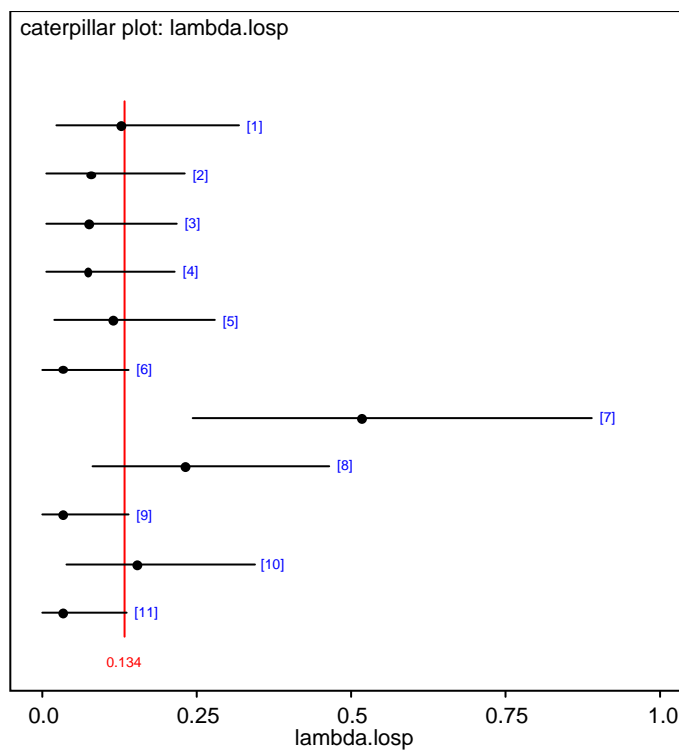
Results

Node statistics								
I	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
alpha	0.9864	0.8921	0.01191	0.2699	0.761	2.398	1001	200000

Node statistics								
I	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
beta	7.644	7.492	0.09477	1.301	5.733	19.88	1001	200000

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
lambda.ind	0.1744	4.909	0.01105	9.519E-4	0.08422	0.537	1001	200000

Results



Exercises

1. *Analyze the pooled data using the Jeffreys prior and compare the results to those from the hierarchical model*
2. *Replace the gamma first-stage prior with a lognormal distribution. Use a `dflat()` hyperprior for the first parameter. For the second parameter, reparameterize in terms of σ and place a `dunif(0, 10)` hyperprior on σ .*
 - a) *How do the mean and median of `lambda.ind` compare to the previous results? Explain.*
 - b) *How do the 90% intervals compare.*
 - c) *Any conclusions about choosing a first-stage prior?*

Example: Variability in EDG FTS Probability

- Assume $X_i \sim \text{binomial}(p_{\text{FTS}, i}, n_i)$ for each EDG
- Assume $p_{\text{FTS}, i} \sim \text{beta}(\alpha, \beta)$
- Assume α and β have independent, vague hyperpriors
 - $\pi_o(\alpha, \beta) = \pi_o(\alpha)\pi_o(\beta)$
 - Vague hyperprior is gamma with both parameters $\ll 1$

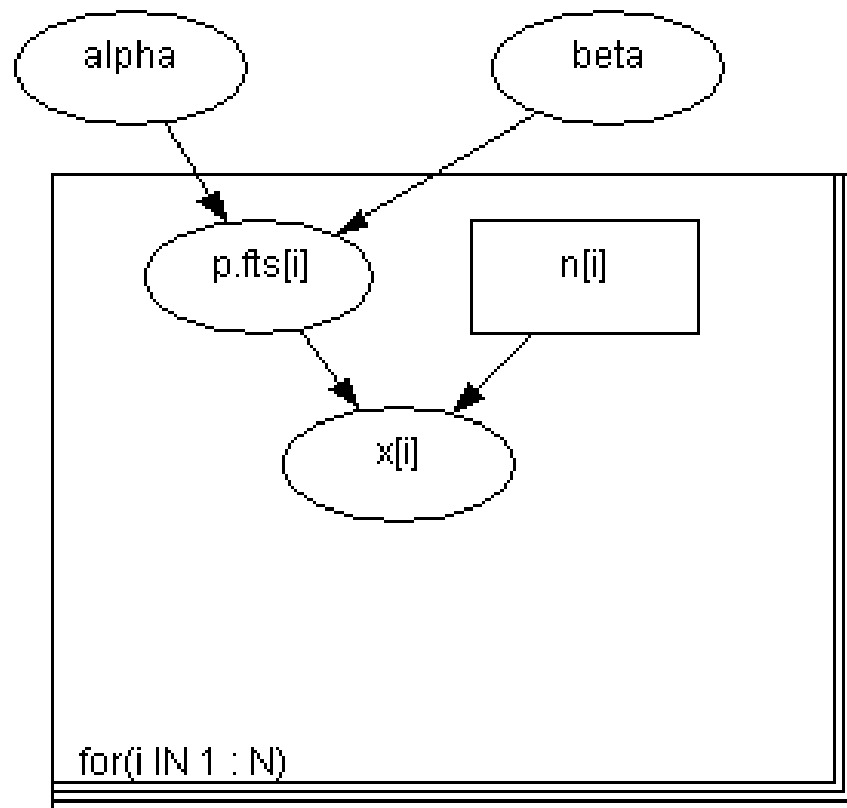
Variability in EDG FTS Probability (cont.)

- *Data*

Data taken from
(Siu and Kelly, 1998)

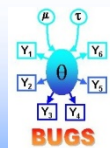
	<i>Failures</i>	<i>Demands</i>
<i>Plant 1</i>	<i>0</i>	<i>140</i>
<i>Plant 2</i>	<i>0</i>	<i>130</i>
<i>Plant 3</i>	<i>0</i>	<i>130</i>
<i>Plant 4</i>	<i>1</i>	<i>130</i>
<i>Plant 5</i>	<i>2</i>	<i>100</i>
<i>Plant 6</i>	<i>3</i>	<i>185</i>
<i>Plant 7</i>	<i>3</i>	<i>175</i>
<i>Plant 8</i>	<i>4</i>	<i>167</i>
<i>Plant 9</i>	<i>5</i>	<i>151</i>
<i>Plant 10</i>	<i>10</i>	<i>150</i>

DAG Model



WinBUGS Model

```
model {  
  for (i in 1 : N) {  
    x[i] ~ dbin(p.fts[i], n[i]) #Binomial dist. for EDG failures  
    p.fts[i] ~ dbeta(alpha, beta) #Beta prior for FTS probability  
  }  
  p.ind ~ dbeta(alpha, beta)  
  alpha ~ dgamma(0.0001, 0.0001) #Vague hyperprior for alpha  
  beta ~ dgamma(0.0001, 0.0001) #Vague hyperprior for beta  
}  
  
inits  
list(alpha=1, beta=25)  
list(alpha=0.5, beta=75)
```



hierarchical EDG FTS probability.odc

Results

- Results for 100,000 samples (10,000 burn in)*

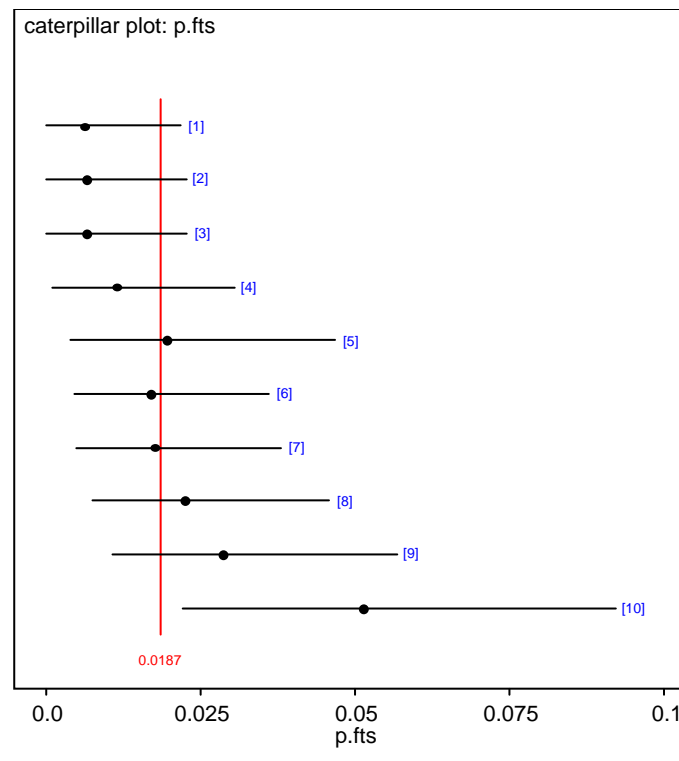
Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
alpha	2.099	4.69	0.1102	0.3235	1.087	6.059	10001	200000

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
beta	112.1	239.2	5.537	12.19	58.24	332.6	10001	200000

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p.ind	0.02098	0.02851	1.19E-4	3.724E-4	0.01397	0.06395	10001	200000

Results

- Results for 100,000 samples (10,000 burn in)*



Empirical Bayes

- *Older approach but still widely used*
- *Data are used to estimate parameters of first-stage prior*
 - *Prior should not depend on observed data, so this is a criticism*
 - *Does not directly include uncertainty in parameters*
 - *Approximations can be used for this*
 - *Prior then updated, with same data used to estimate prior*
- *Can be viewed as approximation to hierarchical Bayes*
- *Widely used in analyses for NRC (e.g., NUREG/CR-6928)*
- *Can be used to estimate initial values for hierarchical Bayes*

Empirical Bayes

- *How it works:*
 1. *Specify first stage prior*
 - a) *Gamma(α , β) for Poisson data*
 - b) *Beta(α , β) for binomial data*
 2. *Write down marginal likelihood*
 3. *Find values of α , β that maximize marginal likelihood*
 4. *Update prior with observed data (yes, data are used twice)*

Empirical Bayes

- *Can use Excel to do empirical Bayes*
 - *Need to get Excel to calculate natural logarithm of marginal likelihood*
 - *Done via Visual Basic macro*
 - *Use Solver to find alpha and beta that maximize natural logarithm of marginal likelihood*
 - *Can also find so-called Kass-Steffey adjustments to address parameter uncertainty*



Microsoft Excel

LOSP variability.xls
Log-like.xla

Potential Issues with Hierarchical Bayes

- *Convergence can be a problem*
 - *Especially when variability is not large*
 - *May need to reparameterize model to speed up convergence*
- *Some results can be sensitive to choice of first-stage prior*
 - *Especially when variability is very large*
- *See (Kelly and Atwood, 2008) for more details*

Exercises

1. *This example uses WinBUGS to analyze a fairly large data set. The file “edg_data.txt” in the BUGS folder contains data for failure on demand for 195 EDGs. Recalling that the MLE of p is given by x/n , you should find that the MLE is > 0.05 for EDGs 183, 184, and 191-195. There is a desire to demonstrate that $\Pr(p > 0.05) < 0.05$. In English, we want to show that we are 95% sure that EDG reliability on demand is at least 95%. If we analyze each EDG separately, using a Jeffreys prior, we will find quite a few that do not meet the criterion (i.e., too many false positives). Pooling the data would also be inappropriate, giving a very narrow credible interval for p ; all of the EDGs would meet the criterion by a wide margin. We would like to get a better answer than either of these simple approaches gives by developing a hierarchical Bayes model that describes the variation in p across the 195 EDGs in the dataset.*
 - a) *Use WinBUGS to analyze a hierarchical Bayes model for this data. Treat the number of failures for each EDG as binomial, with $p_i \sim \text{beta}(\alpha, \beta)$. Use independent diffuse hyperpriors for α and β . We want to find which EDGs have $\Pr(p > 0.05) > 0.05$.*
 - b) *Re-analyze this model, using a $\text{uniform}(0, 10)$ hyperprior for α , and an independent $\text{uniform}(0, 100)$ hyperprior for β . Does this change in priors affect your conclusions about which EDGs have $\Pr(p > 0.05) > 0.05$?*

Section 6: Uncertain Data

- *Problem: observed data are uncertain or “fuzzy”*
 - *Interval-valued observations and other types of censoring*
 - *Unclear whether failure occurred*
 - *Number of demands not known with certainty*
- *Will illustrate Bayesian approach to these problems*

When Durations are Uncertain

- *Consider following data for fire suppression time*

<i>Event</i>	<i>Duration (mins.)</i>
1	< 5
2	< 5
3	< 15
4	14
5	15
6	15
7	10 - 30
8	15
9	150
10	100 - 300

When Durations are Uncertain

- *Use simple model for T (fire duration)*
 - $T_{supp} \sim \exp(\lambda)$
 - λ is suppression rate, units of min^{-1}
- *Data are interval-censored, meaning we only have an interval for some points*
 - *Know duration was $>$ lower bound time and $<$ upper bound time*
- *Handled in WinBUGS by specifying*
 $\text{time.supp}[i] \sim \text{dexp}(\text{lambda})\text{C}(\text{lower}[i], \text{upper}[i])$
- *Specifying data*
 - *If specific duration observed, set $\text{lower}[i] = \text{upper}[i] = \text{duration}$ for that point*
 - *No fixed time observed, set $\text{time.supp}[i] = \text{NA}$, $\text{lower}[i]$ and $\text{upper}[i]$ set to bounds of interval*
 - *WinBUGS imputes missing data using the likelihood function*

When Durations are Uncertain

- *WinBUGS program*

Suppression data, uncertain times

```
model
{
  for (i in 1 : N) {
    time.supp[i] ~ dexp(lambda)C(lower[i], upper[i]) #Exponential distribution for suppression times
  }
  lambda ~ dgamma(0.0001,0.0001) #Diffuse prior for exponential parameter
}

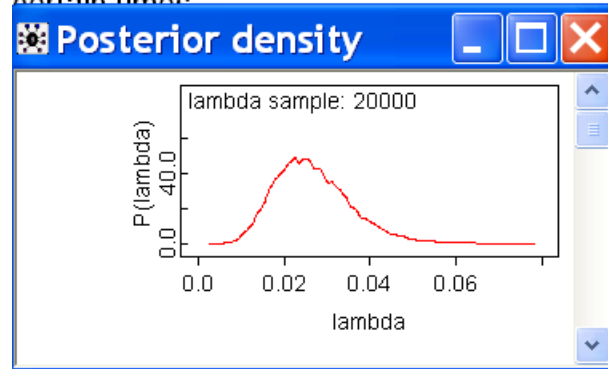
data
list(time.supp=c(NA, NA, NA, 14, 15, 15, NA, 15, 150, NA), lower=c(0,0,0,14,15,15,10,15,150,100), upper=c(5,
5, 15, 14, 15, 15, 30, 15, 150, 300), N = 10)

inits
list(lambda = 0.1) #initial values
list(lambda = 0.01)
```

When Durations are Uncertain

- *Results*

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
lambda	0.02668	0.008713	7.163E-5	0.01423	0.02569	0.04244	1001	20000



When Durations are Uncertain

- *This is sometimes referred to as likelihood-based approach to uncertain data*
 - *Observed time in interval $[a, b]$ contributes $e^{-\lambda a} - e^{-\lambda b}$ to the likelihood function in Bayes' Theorem*
- *Caution: cannot have too much censoring and still get sensible estimates*
 - *If all durations are censored, cannot get meaningful estimate*
- *Frequentist estimate is 0.0265/min, very close to posterior mean*
 - *Expect values to be close with diffuse prior on λ*
 - *Frequentists cannot easily find uncertainty in estimate*

When Demands are Uncertain

- *Often a problem with components such as MOVs, where each demand may not be recorded*
- *Handle in Bayesian approach by assigning a distribution to number of demands, n*
 - *Uniform(a, b)*
 - *Poisson(μ)*
- *Uniform(a, b)*
 - *N is discrete variable, so really want discrete uniform distribution*
 - *Can program this into WinBUGS*
 - *Easier to sample from continuous uniform distribution and round sample to nearest integer*
 - *Endpoints are under-represented, but this is not usually a problem*

When Demands are Uncertain

- *Example: Assume prior distribution for MOV FTO is lognormal with mean = 0.003 and EF = 10*
 - *Assume 1 failure is seen, and demands are not recorded, but are known to lie between 12 and 40*
 - *WinBUGS model*

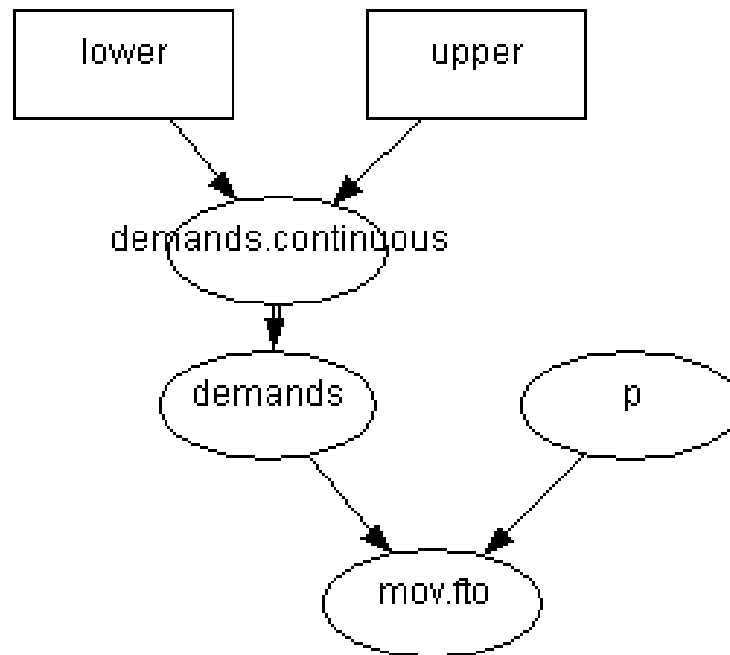
Modeling demands with uniform distribution

```
model {  
  mov.fto ~ dbin(p, demands) #binomial model for MOV FTO  
  p ~ dlnorm(-6.79, 0.5102) #prior for p (mean = 0.003, EF = 10)  
  demands.continuous ~ dunif(lower, upper) #Models uncertainty in demands  
  demands <- round(demands.continuous) #Rounds to integer demands  
}
```

```
data  
list(lower=12, upper=40, mov.fto=1)
```

When Demands are Uncertain

- *DAG model*



When Demands are Uncertain

- *Results*

A screenshot of a software window titled "Node statistics". The window has a blue header bar with the title and standard window controls (minimize, maximize, close). Below the header is a table with statistical data. The table has two rows: a header row and a data row. The header row contains the labels: "mean", "sd", "MC_error", "val5.0pc", "median", "val95.0pc", "start", and "sample". The data row contains the values: "0.009556", "0.0132", "2.924E-4", "6.713E-4", "0.00539", "0.0314", "1001", and "9000". The table is enclosed in a white border with a blue header bar.

	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p	0.009556	0.0132	2.924E-4	6.713E-4	0.00539	0.0314	1001	9000

- *Comparable to results with n set to mean value (26)*
 - *Posterior mean somewhat sensitive to n , but uncertainty in p relatively insensitive to uncertainty in n*

When Demands are Uncertain

- *Try same example, but with Poisson distribution for number of demands*
 - *Use mean of 26 (same as previous example)*
 - *Recall variance = mean for Poisson distribution \Rightarrow standard deviation = 5.1*

Modeling demands with Poisson distribution

```
model {  
  mov.fto ~ dbin(p, demands) #binomial model for MOV FTO  
  p ~ dlnorm(-6.79, 0.5102) #prior for p (mean = 0.003, EF = 10)  
  demands ~ dpois(mu) # Poisson distribution for number of demands  
}
```

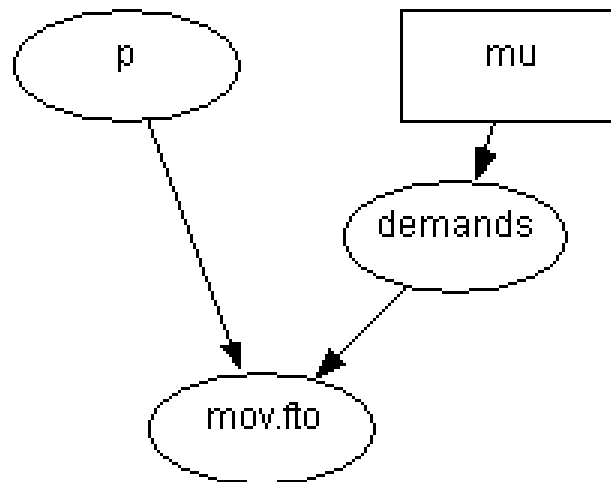
data

```
list(mu = 26, mov.fto=1)
```



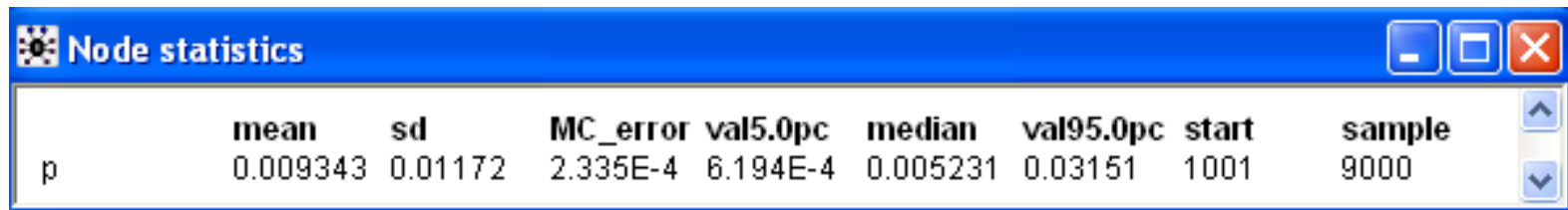
When Demands are Uncertain

- *DAG model*



When Demands are Uncertain

- *Results*

A screenshot of a software window titled "Node statistics". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is a table with 9 columns: "p", "mean", "sd", "MC_error", "val5.0pc", "median", "val95.0pc", "start", and "sample". The table contains one row of data for parameter "p".

p	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p	0.009343	0.01172	2.335E-4	6.194E-4	0.005231	0.03151	1001	9000

– *Essentially the same as with uniform distribution for n*

When Failures Are Uncertain

- *Information from LERs and other information sources may not be clear enough to ascertain exact number of failures*
- *Sloppy record-keeping may result in imprecise estimates*
 - *Analogous to rounding of value*
- *Two approaches*
 - *Posterior-averaging*
 - *Likelihood-based*

Posterior-Averaging Approach

- *Analyst must develop subjective prior distribution for number of observed events*
- *Example 1: plugging of service water strainers*
 - *Inspection Report (IR) describes 7 plugging events over time period of interest*
 - *3 of these may not have been complete plugging events from perspective of PRA model*
 - *Therefore, actual number of events is 4, 5, 6, or 7*

Posterior-Averaging Approach

- *Analyst, by poring over the IR, and applying expert knowledge, has come up with following distribution for actual number of complete plugging events:*

$$Pr(x = 4) = 0.75$$

$$Pr(x = 5) = 0.15$$

$$Pr(x = 6) = 0.075$$

$$Pr(x = 7) = 0.025$$

- *Note that probabilities must sum to unity!*

Posterior-Averaging Approach

- *WinBUGS program for Example 1*

Number of Poisson events is uncertain

```
model {  
  #Use following if data are observed with subjective probability specified by p[] vector  
  for(i in 1:N) {  
    x[i] ~ dpois(mu[i])  
    mu[i] <- lambda[i]*time.exp  
    lambda[i] ~ dgamma(0.5, 0.0001) #Jeffreys prior for lambda  
  }  
  lambda.avg <- lambda[r] #Overall composite lambda, monitor this node  
  r ~ dcat(p[])  
}
```

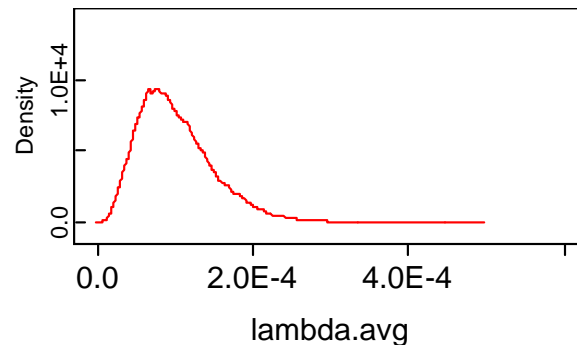
Data

```
list(x=c(4,5,6,7), p=c(0.75, 0.15, 0.075, 0.025), N=4)  
list(time.exp=48180)
```

Posterior-Averaging Approach

- *Results for Example 1*

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
lambda.avg	1.012E-4	4.858E-5	2.17E-7	3.678E-5	9.312E-5	1.924E-4	1001	49000



Posterior-Averaging Approach

- *Example 2: MOV failures in 381 demands*
 - *Because of vagueness of problem reports, can't tell if actual number of failures was 3, 4, 5, or 6*
 - *Analyst develops following distribution:*
 - $Pr(x = 3) = 0.1$
 - $Pr(x = 4) = 0.7$
 - $Pr(x = 5) = 0.15$
 - $Pr(x = 6) = 0.05$

Posterior-Averaging Approach

- *WinBUGS program for Example 2*

Number of binomial failures is uncertain

```
model {  
  #Use following if data are observed with subjective probability specified by q[] vector  
  for (i in 1:N) {  
    x[i] ~ dbin(p[i], D)  
    p[i] ~ dbeta(0.5, 0.5) #Jeffreys prior  
  }  
  p.avg <- p[r] #Composite posterior, monitor this node  
  r ~ dcat(q[])  
}
```

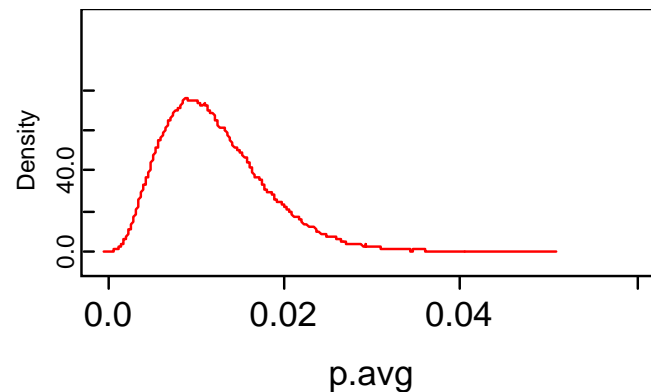
Data

```
list(x=c(3,4,5,6), q=c(0.1, 0.70, 0.15, 0.05), N=4)  
list(D=381)
```

Posterior-Averaging Approach

- *Results for Example 2*

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p.avg	0.01222	0.005879	2.418E-5	0.004338	0.01129	0.0232	1001	49000



Likelihood-Based Approach

- *Revisit same examples, but treat number of failures as “missing”, but known to be in the interval [4, 7] for Example 1 and [3, 6] for Example 2.*
- *WinBUGS program for Example 1*

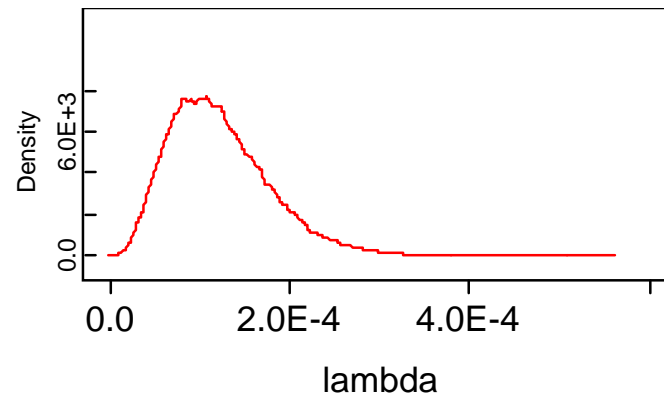
```
Number of Poisson events is uncertain
model {
  #Use following for likelihood-based approach
  x ~ dpois(mu)|(lower, upper)
  mu <- lambda*time.exp
  lambda ~ dgamma(0.5, 0.0001) #Jeffreys prior
}

data
#Load following for likelihood-based approach
list(x=NA, lower=4, upper=7)
list(time.exp=48180)
```

Likelihood-Based Approach

- *Results for Example 1*

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
lambda	1.217E-4	5.518E-5	3.153E-7	4.591E-5	1.141E-4	2.234E-4	1001	49000



Likelihood-Based Approach

- *WinBUGS program for Example 2*

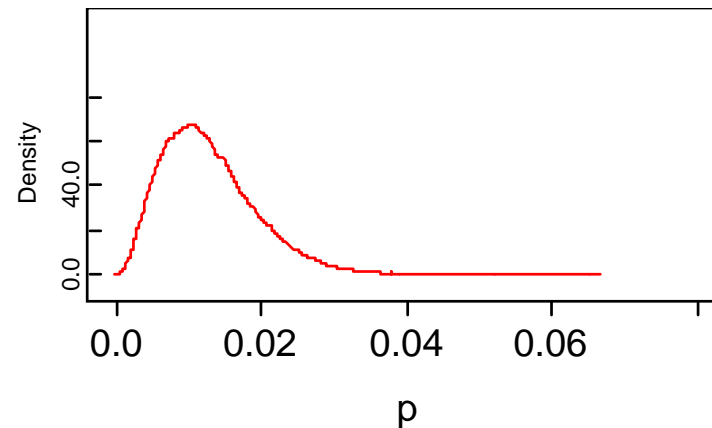
```
Number of binomial failures is uncertain
model {
  #Use following for likelihood-based approach
  x ~ dbin(p, D)I(lower, upper)
  p ~ dbeta(0.5, 0.5) #Jeffreys prior
}

data
#Load following for likelihood-based approach
list(x=NA, lower=3, upper=6)
list(D=381)
```

Likelihood-Based Approach

- *Results for Example 2*

Node statistics								
i	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p	0.01274	0.00641	2.722E-5	0.00409	0.01179	0.0247	10001	90000



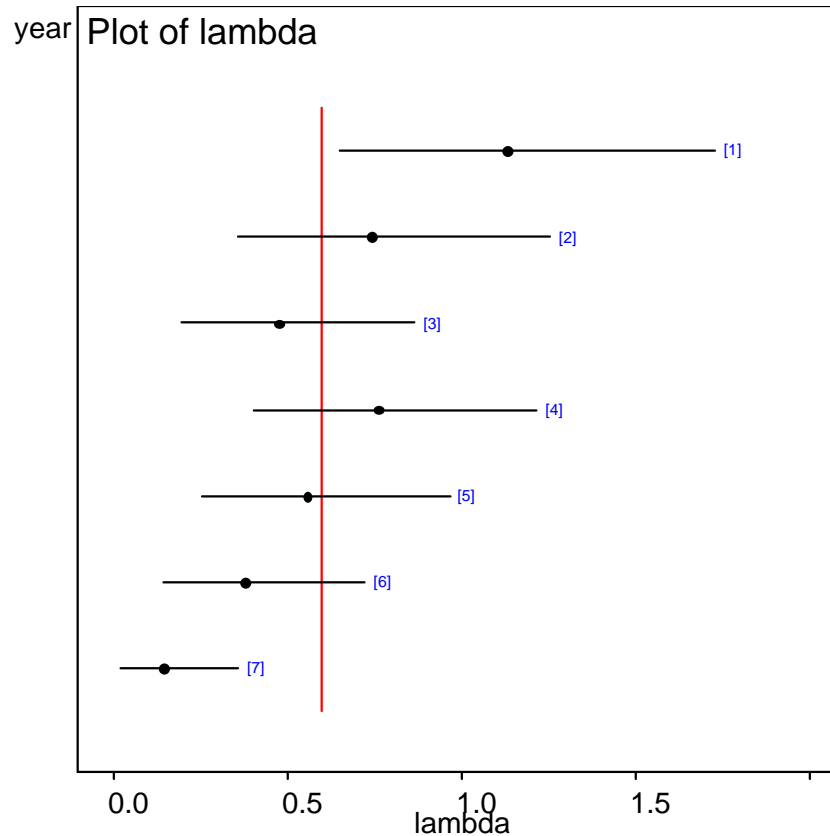
Section 7: Bayesian Modeling of Time Trends

- *Problem*
 - *What if there is a monotonic trend over time for p or λ ?*
- *Will illustrate*
 - *Graphical methods*
 - *Implementation of quantitative trend models in WinBUGS*

Example Data

<i>Year</i>	<i>Number of Events</i>	<i>Exposure Time</i>
1	16	14.63
2	10	14.15
3	7	15.75
4	13	17.77
5	9	17.11
6	6	17.19
7	2	17.34

Is There a Time Trend in λ ?

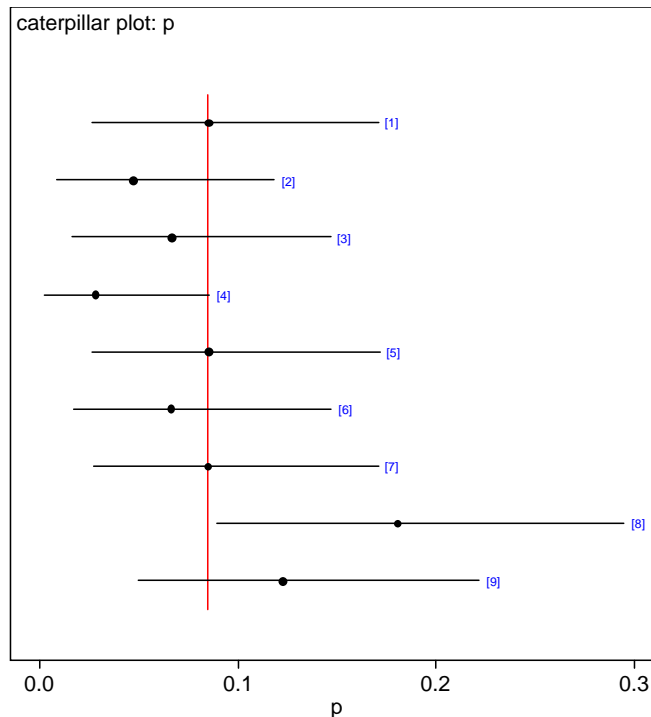


- *Graph indicates apparent decreasing trend over time*

Example Valve Leakage Data

<i>Year</i>	<i>Number of Failures</i>	<i>Demands</i>
1	4	52
2	2	52
3	3	52
4	1	52
5	4	52
6	3	52
7	4	52
8	9	52
9	6	52

Is There a Time Trend in p?



- *Graph appears to indicate increasing trend in time*

Nonhomogeneous Poisson Process

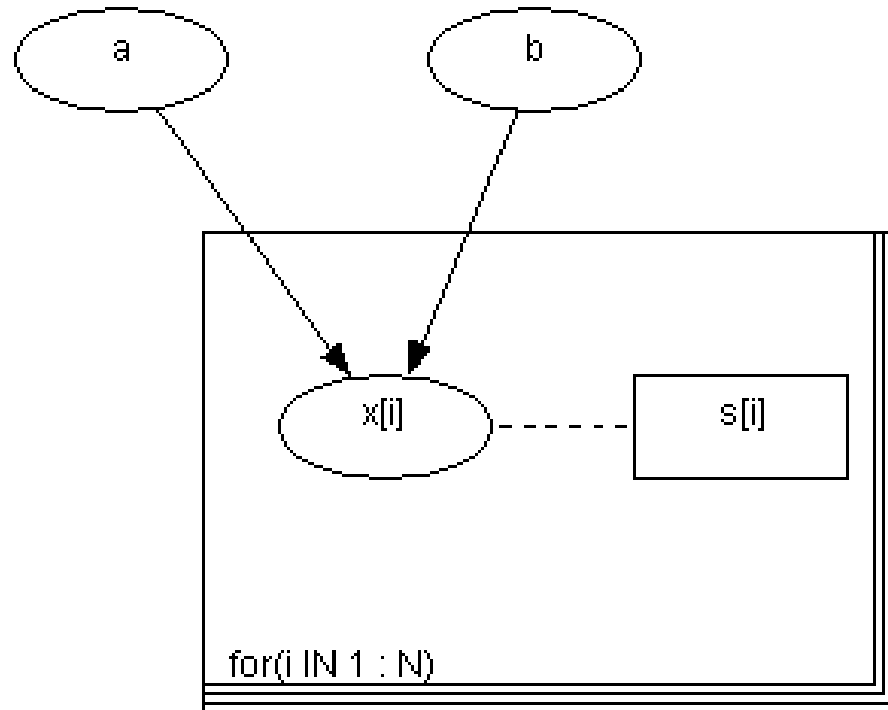
- *Recall that one of the assumptions leading to the Poisson distribution was that λ is constant*
- *Can relax this assumption, and allow λ to vary with time*
- *Leads to what is called a nonhomogeneous Poisson process*
 - *Like a Poisson distribution, but with parameter*

$$\mu(t) = \int_0^t \lambda(s) ds$$

Need Model for $\lambda(t)$

- *Many possibilities:*
 - *Linear: $\lambda(t) = \lambda_o + at$*
 - *Loglinear: $\ln[\lambda(t)] = a + bt$*
 - *Power law: $\lambda(t) = \alpha\lambda_o t^{\alpha-1}$*
 - *Extended power law: $\lambda(t) = \alpha\lambda_o t^{\alpha-1} + \lambda_{\text{ext}}$*
- *No theoretical justification for any of these*
- *Will illustrate loglinear model in WinBUGS*

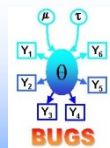
DAG for Loglinear Trend Model



Loglinear Model: $\ln[\lambda(t)] = a + bt$

- *WinBUGS model*

```
model {  
  for (i in 1:N) {  
    x[i] ~ dpois(mu[i])  
    mu[i] <- lambda[i]*s[i]  
    log(lambda[i]) <- a + b*i  
  }  
  
  a~dflat()  
  b~dflat()  
}
```

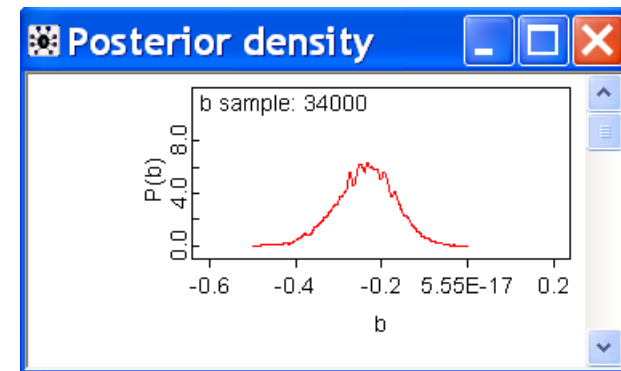


Loglinear model for time trend.odc

Loglinear Model: $\ln[\lambda(t)] = a + bt$

- Results*

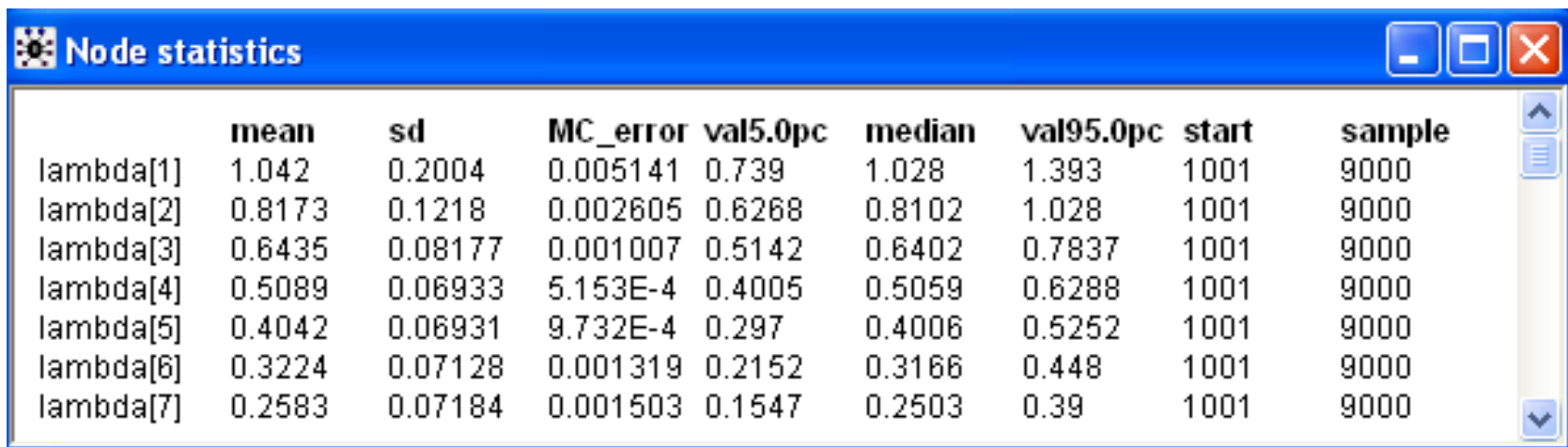
Node statistics								
i	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
b	-0.2375	0.06717	0.002073	-0.3528	-0.235	-0.1312	4001	34000



- $Pr(b < 0)$ near unity (negative slope)*
- Posterior distribution of b is approximately normal*

Loglinear Model: $\ln[\lambda(t)] = a + bt$

- Results for lambda in each year:*

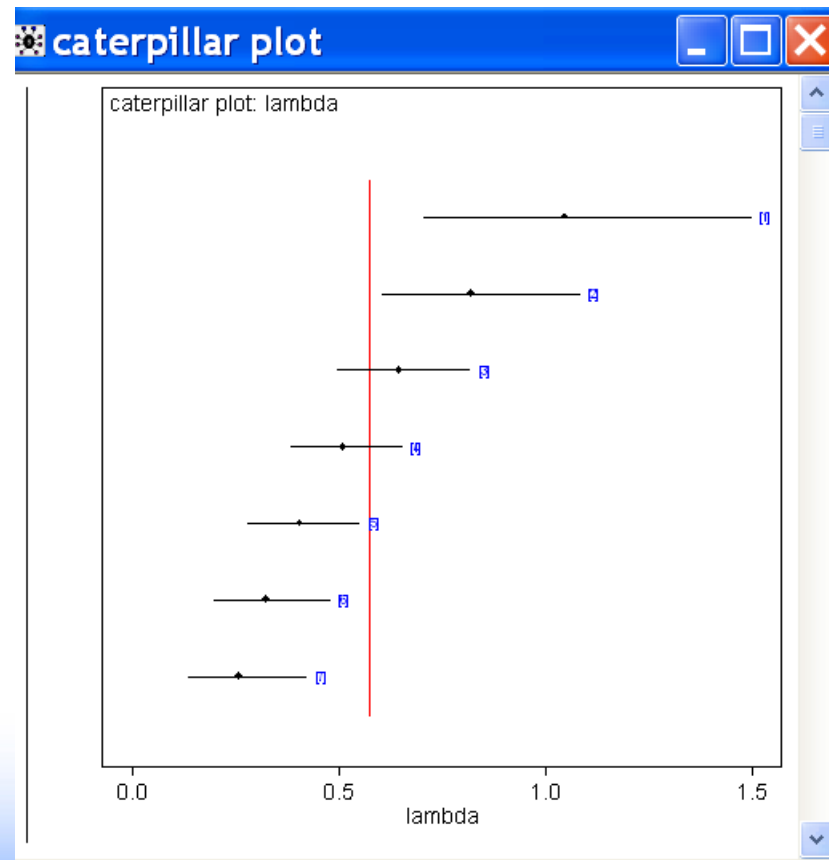


A screenshot of a software window titled "Node statistics". The window contains a table with 9 columns: parameter name, mean, sd, MC_error, val5.0pc, median, val95.0pc, start, and sample. The table lists results for parameters lambda[1] through lambda[7]. The values generally decrease from lambda[1] to lambda[7].

	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
lambda[1]	1.042	0.2004	0.005141	0.739	1.028	1.393	1001	9000
lambda[2]	0.8173	0.1218	0.002605	0.6268	0.8102	1.028	1001	9000
lambda[3]	0.6435	0.08177	0.001007	0.5142	0.6402	0.7837	1001	9000
lambda[4]	0.5089	0.06933	5.153E-4	0.4005	0.5059	0.6288	1001	9000
lambda[5]	0.4042	0.06931	9.732E-4	0.297	0.4006	0.5252	1001	9000
lambda[6]	0.3224	0.07128	0.001319	0.2152	0.3166	0.448	1001	9000
lambda[7]	0.2583	0.07184	0.001503	0.1547	0.2503	0.39	1001	9000

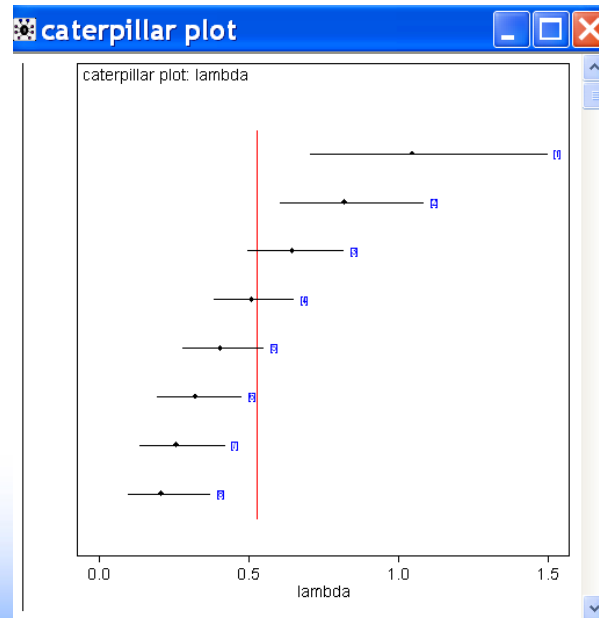
Loglinear Model: $\ln[\lambda(t)] = a + bt$

- *Uncertainty in lambda for each year*



Predicted λ In Year 8

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
lambda[1]	1.048	0.2062	0.006232	0.744	1.027	1.42	4001	32000
lambda[2]	0.8188	0.1241	0.003161	0.6289	0.8123	1.035	4001	32000
lambda[3]	0.6436	0.08268	0.001249	0.5159	0.6405	0.7865	4001	32000
lambda[4]	0.5082	0.06994	6.443E-4	0.3986	0.5048	0.6284	4001	32000
lambda[5]	0.403	0.06982	0.001148	0.2938	0.3996	0.5242	4001	32000
lambda[6]	0.321	0.07156	0.001541	0.2121	0.3157	0.4472	4001	32000
lambda[7]	0.2567	0.07177	0.001747	0.1513	0.25	0.3871	4001	32000
lambda[8]	0.2062	0.07014	0.001819	0.1073	0.1976	0.3362	4001	32000



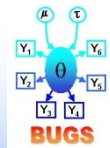
Possible Models for p

- *Constant p*
- *Logistic*
 - $\text{Logit}(p) = \ln(p/1 - p) = a + bt$
- *Probit*
 - $\text{Probit}(p) = \Phi^{-1}(p) = a + bt$
- *Complementary log-log*
 - $\ln(-\ln(1-p)) = a + bt$

Logistic Model: $\text{Logit}(p) = a + bt$

- *WinBUGS model*

```
model
{
  for (i in 1:N) {
    x[i] ~ dbin(p[i], n[i]) #Binomial distribution for failures in each year
    logit(p[i]) <- a + b*i #Use of logit() link function to simplify model
  }
  a~dflat() #Diffuse prior for a
  b~dflat() #Diffuse prior for b
}
```

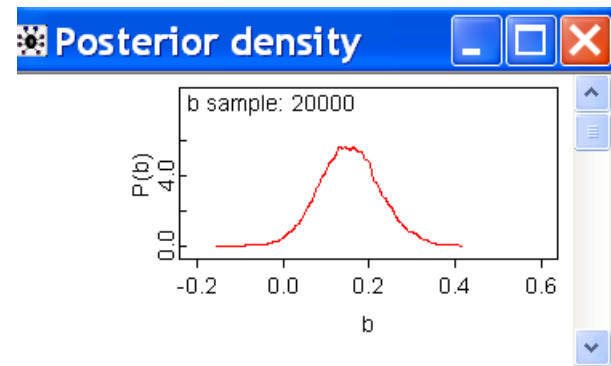


Logistic model for time trend.odc

Logistic Model: $\text{Logit}(p) = a + bt$

- Results

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
b	0.1553	0.07038	0.001592	0.04212	0.1538	0.2742	1001	20000



- $\Pr(b > 0)$ is > 0.95
 - Strong evidence of increasing trend in p over time

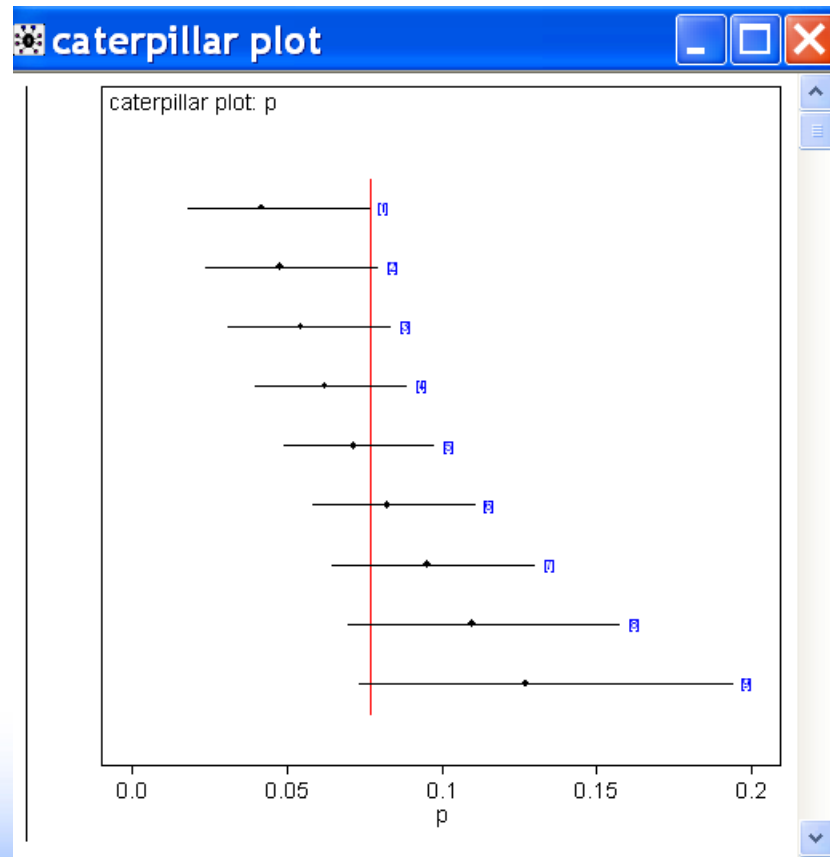
Logistic Model: $\text{Logit}(p) = a + bt$

- Results for p in each year:*

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p[1]	0.04171	0.01501	3.214E-4	0.02066	0.03981	0.06913	1001	20000
p[2]	0.04749	0.01429	3.008E-4	0.02662	0.04611	0.07308	1001	20000
p[3]	0.05427	0.01344	2.691E-4	0.03405	0.05326	0.07787	1001	20000
p[4]	0.06221	0.01262	2.232E-4	0.04285	0.06147	0.08414	1001	20000
p[5]	0.07151	0.01232	1.609E-4	0.05256	0.07086	0.09283	1001	20000
p[6]	0.0824	0.01336	9.171E-5	0.06146	0.08171	0.1057	1001	20000
p[7]	0.09513	0.01665	1.138E-4	0.06919	0.09442	0.1238	1001	20000
p[8]	0.1099	0.02255	2.592E-4	0.07524	0.1087	0.1491	1001	20000
p[9]	0.127	0.03108	4.641E-4	0.08027	0.1247	0.1819	1001	20000

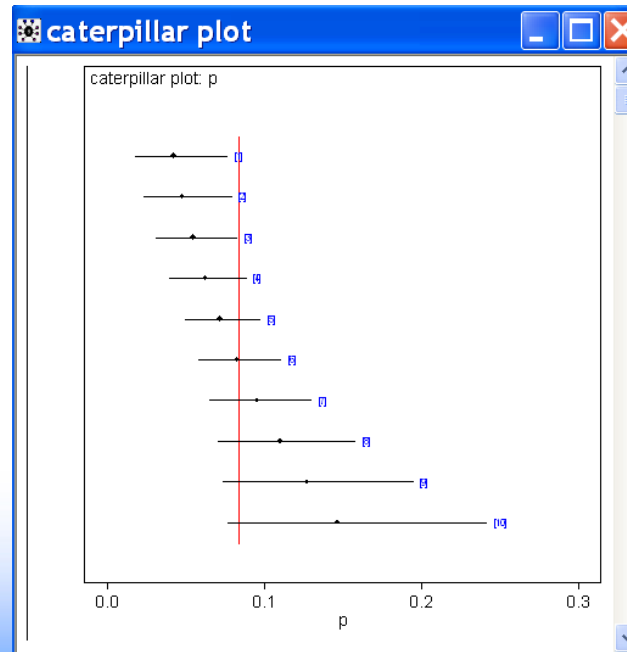
Logistic Model: $\text{Logit}(p) = a + bt$

- *Uncertainty in p for each year*



Predicted Value of p In Year 10

Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p[1]	0.04191	0.01513	3.805E-4	0.02035	0.04016	0.06972	1001	20000
p[2]	0.04767	0.01441	3.514E-4	0.02632	0.0464	0.07349	1001	20000
p[3]	0.05442	0.01354	3.087E-4	0.03365	0.05352	0.07804	1001	20000
p[4]	0.06233	0.01269	2.485E-4	0.04248	0.06167	0.08411	1001	20000
p[5]	0.0716	0.01235	1.684E-4	0.05209	0.071	0.09281	1001	20000
p[6]	0.08245	0.01338	8.42E-5	0.06163	0.0818	0.1057	1001	20000
p[7]	0.09513	0.0167	1.456E-4	0.06925	0.0944	0.1239	1001	20000
p[8]	0.1099	0.0227	3.328E-4	0.07515	0.1084	0.1492	1001	20000
p[9]	0.1269	0.03135	5.819E-4	0.08029	0.1243	0.183	1001	20000
p[10]	0.1465	0.0426	8.91E-4	0.08498	0.1419	0.2235	1001	20000



Section 8: Mixture Priors

- *Problem*
 - *What if a single distribution is inadequate to represent prior knowledge?*
- *Will illustrate*
 - *Implementation of mixture prior distributions in WinBUGS*

What is a Mixture Prior?

- *Can specify prior distribution as a weighted average of distributions*

$$\pi_o(\theta) = \sum_{i=1}^n k_i \pi_{o,i}(\theta)$$

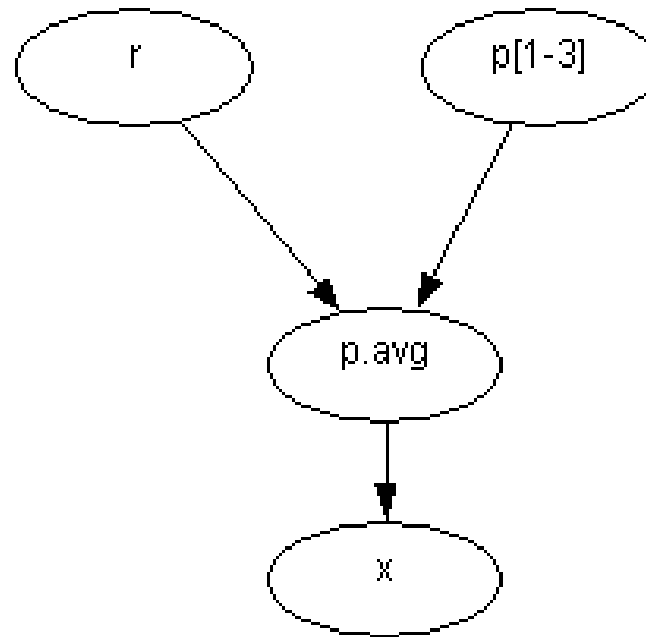
where k_i s are weighting factors, which sum to 1

- *Can lead to multimodal prior (and posterior)*

Example Use of Mixture Prior

- *Consider case where three sets of prior information about p are available*
- *Each gives different lognormal distribution for p*
 - *Source 1: median = 0.03, EF = 5*
 - *Source 2: median = 0.001, EF = 10*
 - *Source 3: median = 0.003, EF = 8.1*
- *Our subjective weights for the three sources are 0.6, 0.2, and 0.2*
- *Use this information as prior, with observed data of 1 failure in 1,200 demands*

DAG Model



Example Use of Mixture Prior

- *Can model this in WinBUGS*

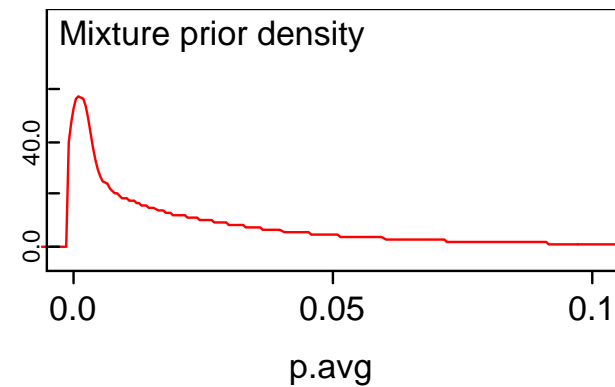
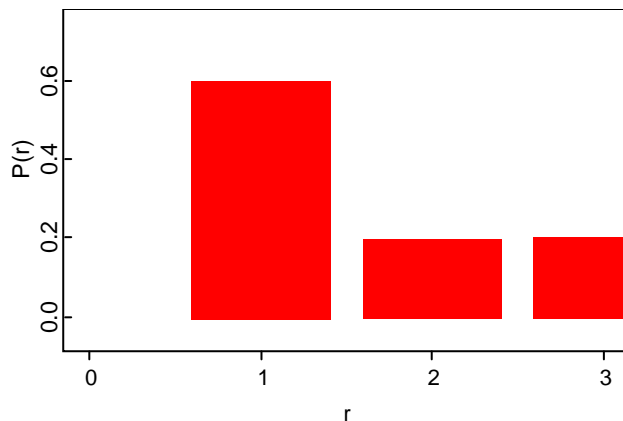
Mixture prior for p in binomial distribution

```
model {  
  for(i in 1:3) {  
    p[i] ~ dlnorm(mu[i], tau[i]) #Individual lognormal priors  
    mu[i] <- log(median[i])  
    sigma[i] <- log(EF[i])/1.645  
    tau[i] <- pow(sigma[i], -2)  
  }  
  p.avg <- p[r] #Mixture prior  
  r ~ dcat(k[]) #Categorical distribution  
  x ~ dbin(p.avg, n) #Binomial likelihood for observed data  
}  
  
data  
list(median=c(3.E-2, 1.E-3, 3.E-3), EF=c(5, 10, 8.1), k=c(0.6, 0.2, 0.2))  
list(x=1, n=1200)
```

Example Use of Mixture Prior

- *Prior*

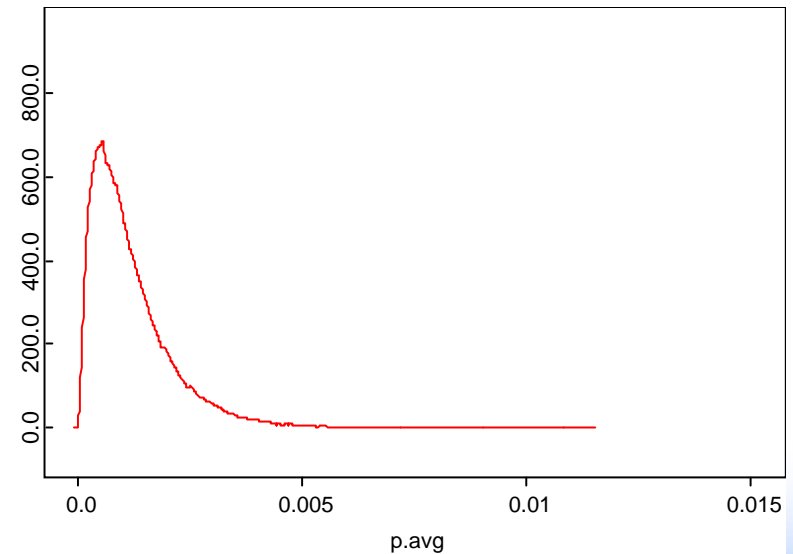
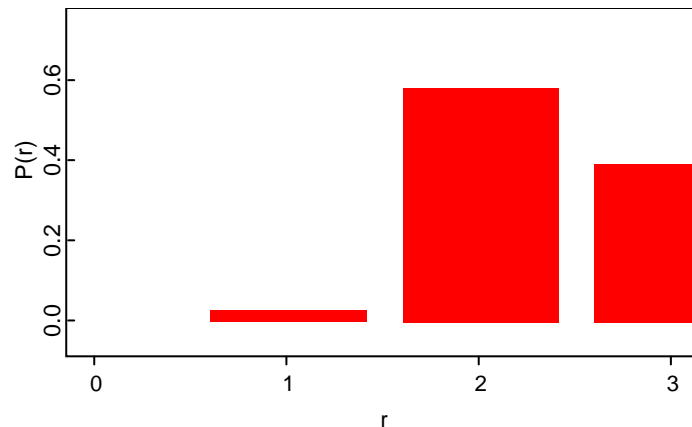
Node statistics								
	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p.avg	0.03081	0.05157	1.686E-4	3.163E-4	0.01381	0.1162	1001	99000



Example Use of Mixture Prior

- *Posterior*

Node statistics								
I	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p.avg	0.001171	9.097E-4	4.158E-6	2.054E-4	9.326E-4	0.002957	1001	100000



Exercises

1. *In its normal state, a component has a CNI prior with a mean failure rate of $4.6\text{E-}5/\text{hr}$. In its degraded state, the prior distribution for the failure rate is a CNI prior with a mean that is a factor of 10 higher. Assume there have been 3 tests with an average run time of 23 hours per test, and that 1 failure to run has been observed. Assume the prior probability that the component is in the degraded state is 0.05.*
 - a) *What is the posterior mean failure rate?*
 - b) *Which is the more likely state of the component based on this information?*
 - c) *Repeat (a) and (b) assuming 2 failures to run have been observed. Note the shape of the posterior density in this case.*

Section 9: Model Validation and Selection

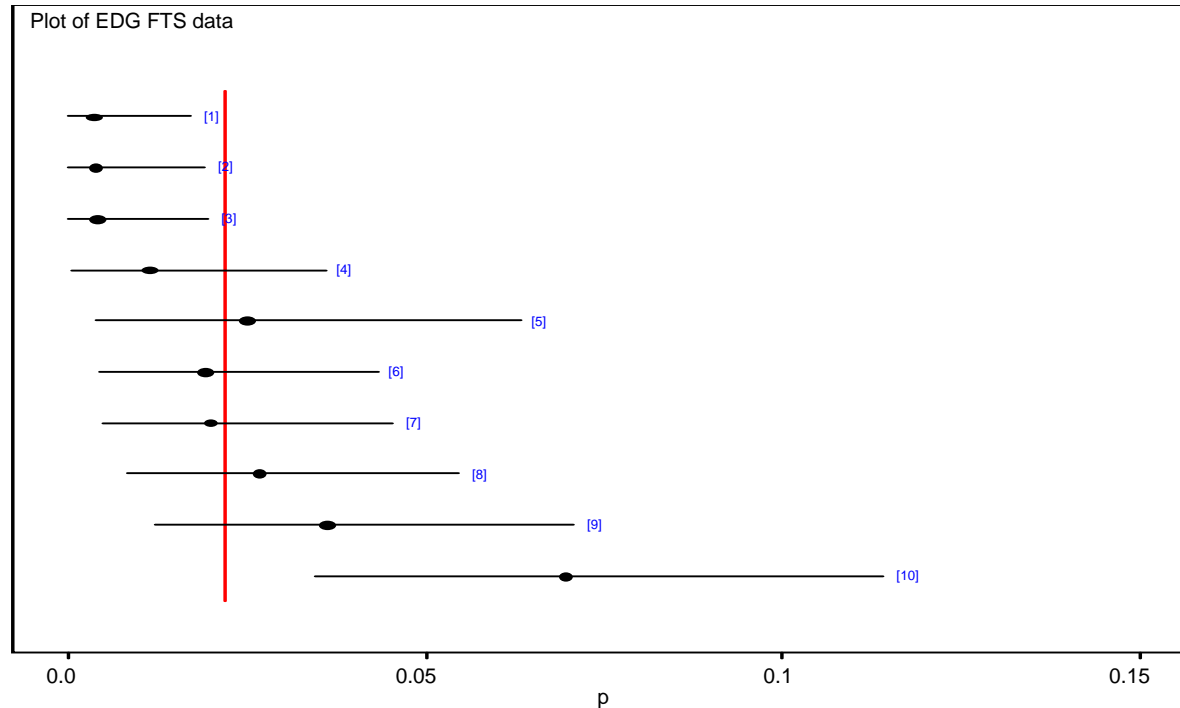
- *Problem*
 - *How to decide if model (prior + likelihood) is a “good fit”*
 - *How to decide among competing models*
- *Will illustrate*
 - *Graphical methods*
 - *Use of predictive distributions and summary statistics*
 - *Information criteria for model selection*
 - *Implementation of quantitative methods in WinBUGS*

Graphical Methods

- *Example: EDG FTS data*
- *Is p_{FTS} varying significantly among the plants?*
 - *Side-by-side plots of Bayes credible intervals*
 - *Use Jeffreys prior*
 - *Plot posterior intervals*

	<i>Failures</i>	<i>Demands</i>
<i>Plant 1</i>	0	140
<i>Plant 2</i>	0	130
<i>Plant 3</i>	0	130
<i>Plant 4</i>	1	130
<i>Plant 5</i>	2	100
<i>Plant 6</i>	3	185
<i>Plant 7</i>	3	175
<i>Plant 8</i>	4	167
<i>Plant 9</i>	5	151
<i>Plant 10</i>	10	150

Graphical Methods

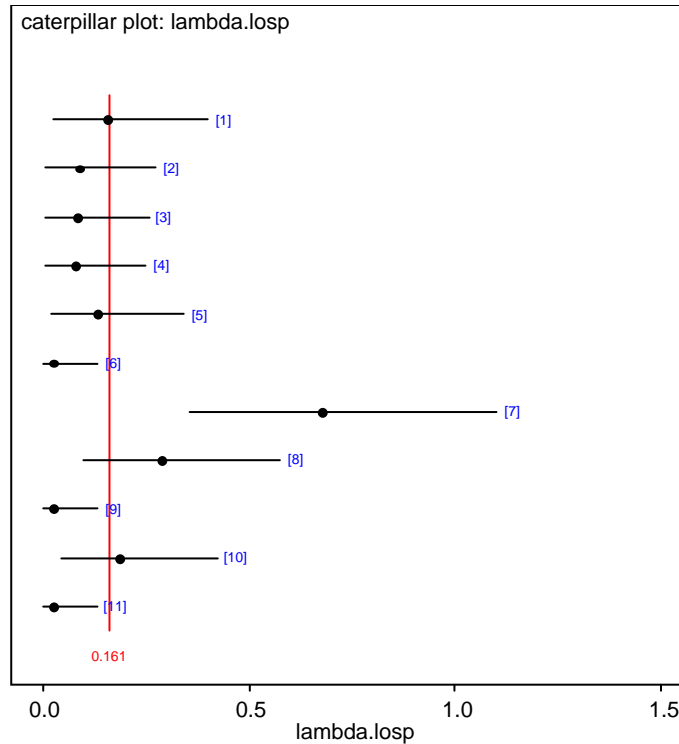


- Graph suggests plant-to-plant variability
 - Pooling of data across plants may not be justified

Example: LOSP frequency data

<i>Plant</i>	<i>Failures</i>	<i>Time (yrs)</i>
<i>Plant 1</i>	<i>2</i>	<i>15.986</i>
<i>Plant 2</i>	<i>1</i>	<i>16.878</i>
<i>Plant 3</i>	<i>1</i>	<i>18.146</i>
<i>Plant 4</i>	<i>1</i>	<i>18.636</i>
<i>Plant 5</i>	<i>2</i>	<i>18.792</i>
<i>Plant 6</i>	<i>0</i>	<i>18.976</i>
<i>Plant 7</i>	<i>12</i>	<i>18.522</i>
<i>Plant 8</i>	<i>5</i>	<i>19.04</i>
<i>Plant 9</i>	<i>0</i>	<i>18.784</i>
<i>Plant 10</i>	<i>3</i>	<i>18.868</i>
<i>Plant 11</i>	<i>0</i>	<i>19.232</i>

LOSP frequency data



Illustrates significant variability

Are Observed Times Exponential?

- Can use cumulative hazard plot
 - For exponential data, hazard function is constant, equal to λ
 - Cumulative hazard function is integral of hazard function from 0 to t , giving

$$H(t) = \int_0^t \lambda(t') dt'$$

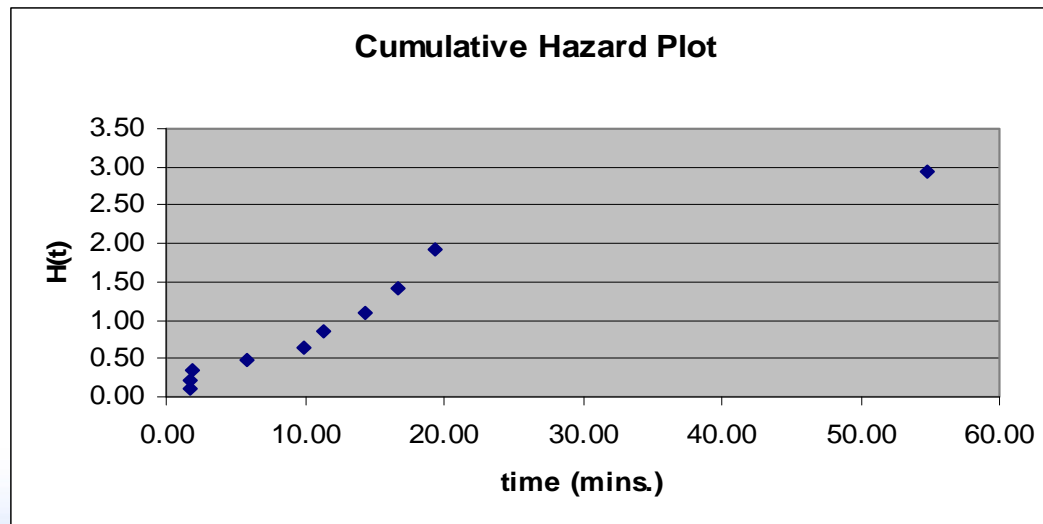
- For exponential data, we get a straight line with slope λ :

$$H(t) = \lambda t$$

- If plot of estimate of $H(t)$ is curved, this suggests λ **not** constant with time
 - Data not exponential
 - Decreasing slope (concave) \Rightarrow decreasing hazard
 - Increasing slope (convex) \Rightarrow increasing hazard

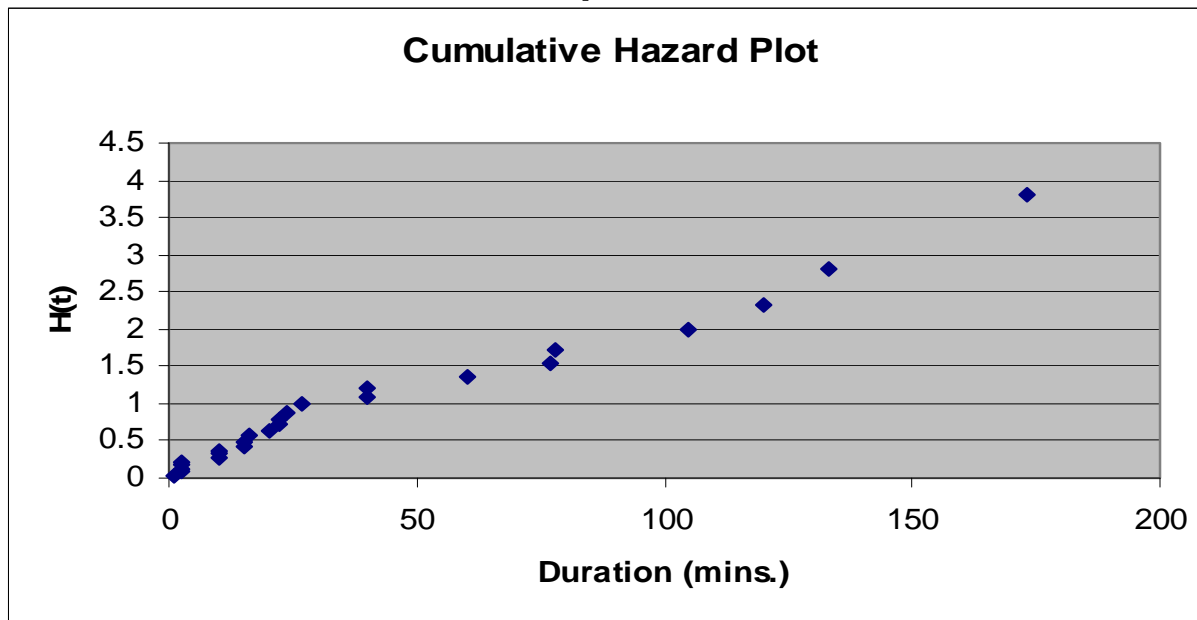
Estimating $H(t)$

- Use a step function, which jumps by $1/n_t$, where n_t is the number of items that have not failed by time t
- Example: observed following fire suppression times, in minutes: 1.7, 1.8, 1.9, 5.8, 10.0, 11.3, 14.3, 16.6, 19.4, 54.8, $n = 10$
 - $H(t)$ jumps from 0 to 0.1 at 1.7 mins., 0.1 to 0.21 at 1.8 mins., etc.



Estimating $H(t)$

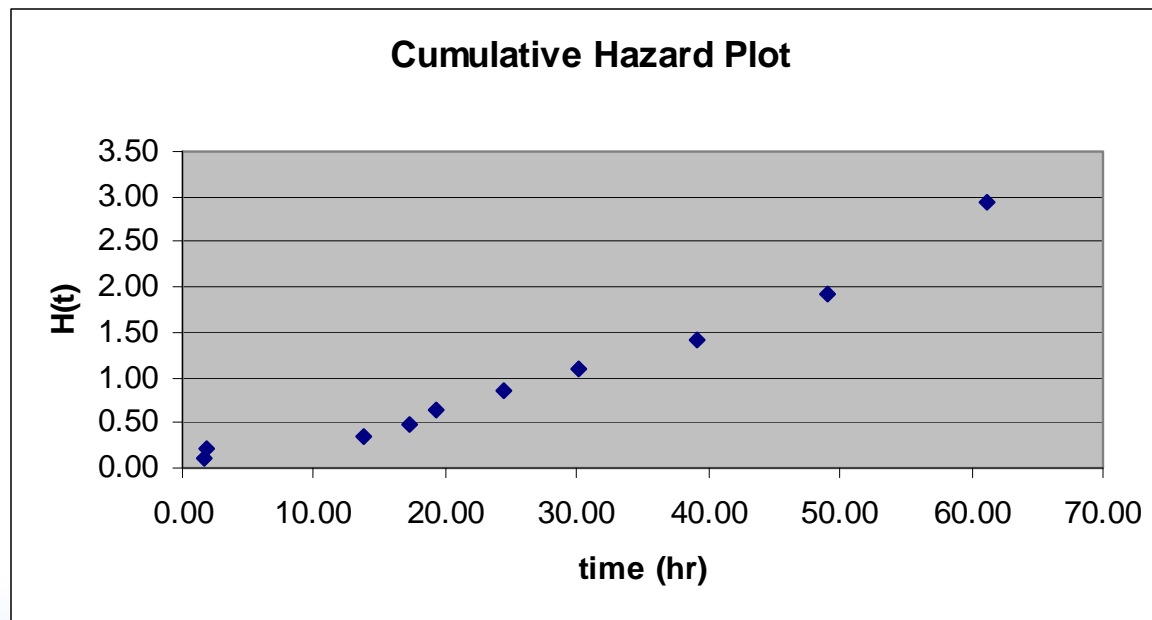
- *With only 10 points, plot is somewhat ragged*
- *Another chart, with $n = 25$ points:*



– *These data look exponential*

Another Example

- Observed the following times (in hours) at which offsite power was recovered: 1.6, 1.8, 13.8, 17.2, 19.2, 24.4, 30.2, 39.1, 49.1, 61.2



Another Example

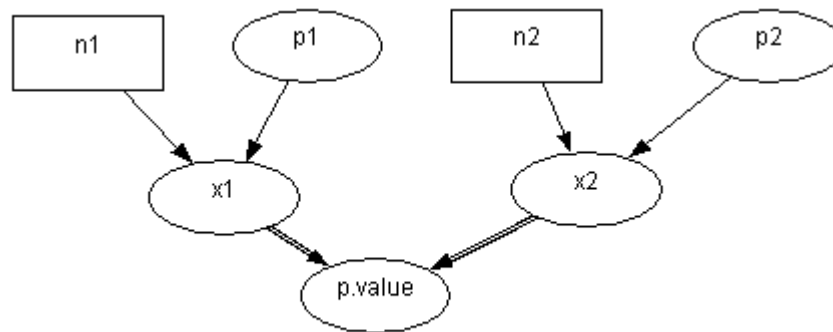
- *This plot looks fairly straight*
- *Are the data really exponential?*
- *Need quantitative tools to help answer this question*
 - *Graphs are useful with fairly large amounts of data, less useful with small amounts*
 - *Data on previous slide are actually from a Weibull distribution with $\lambda(t)$ decreasing as t increases*

Use of Posterior Distribution

- *Example: can EDG data from two trains be pooled?*
 - *Train 1: 3 failures in 75 demands*
 - *Train 2: 5 failures in 69 demands*
- *Use Jeffreys prior and find posterior distribution for each train*
- *Calculate probability that $p_2 > p_1$*
 - *$Pr(p_2 > p_1) = ?$*

Use of Posterior Distribution

- *WinBUGS model*



```
model{  
  p1 ~ dbeta(0.5,0.5)    #Jeffreys prior for p1  
  x1 ~ dbin(p1,n1)       #Binomial distribution for x1  
  p2 ~ dbeta(0.5,0.5)    #Jeffreys prior for p2  
  x2 ~ dbin(p2,n2)       #Binomial distribution for x2  
  p.value <- step(p2 - p1) #Gives Pr(p2 > p1)  
}  
data  
list(x1 = 3, n1 = 75, x2 = 5, n2 = 69)
```

Use of Posterior Distribution

- *Results*

A screenshot of a software window titled "Node statistics". It contains a table with statistical data for a parameter labeled "p.value".

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
p.value	0.7982	0.4013	0.004582	0.0	0.0	1.0	1.0	1.0	1001	9000

- $Pr(p_2 > p_1) = 0.7982$
- *Weak evidence that $p_2 > p_1$*
- *Could probably pool data from both EDGs*

Use of Predictive Distribution

- *In Bayesian framework, “model” comprises*
 - *Likelihood function*
 - *How data were generated*
 - *Prior distribution*
 - *Uncertainty about parameters*
- *Bayesian inference sometimes criticized for sensitivity to prior*
 - *In practice, likelihood function can also be in question*
- *Need to check both parts of our “model”*

Prior Predictive Distribution

- The prior **predictive** distribution is the denominator of Bayes' Theorem

$$f(x) = \int f(x | \theta) \pi_0(\theta) d\theta$$

- Gives probability of observing $X = x$, unconditional upon any particular value of the parameter(s), θ
 - Also called the marginal distribution of X
- Before observing data, can check reasonableness of prior by calculating probabilities for data we expect to see
 - Small probabilities \Rightarrow prior not consistent with expected data
 - Sometimes called “preposterior analysis”
- Not defined for improper priors (e.g., Jeffreys prior for Poisson or exponential data)

Binomial Likelihood – Beta Prior

- $f(x)$ can be found ***analytically***
 - Called *beta-binomial distribution*

$$f(x) = \binom{n}{x} \frac{\Gamma(\alpha + x)}{\Gamma(\alpha)} \frac{\Gamma(\beta + n - x)}{\Gamma(\beta)} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha + \beta + n)}.$$

- α and β are parameters of beta prior distribution for p
- $\Gamma(\bullet)$ is the gamma function, defined for $\theta > 0$ as

$$\Gamma(\theta) = \int_0^{\infty} t^{\theta-1} e^{-t} dt$$

- Excel has $\ln[\Gamma(\bullet)]$ built in (“single parameter estimation.xls” in Excel folder)

Poisson Likelihood – Gamma Prior

- $f(x)$ can be found ***analytically***
 - Called *gamma-Poisson distribution*

$$f(x) = \frac{\Gamma(\alpha + x)}{x! \Gamma(\alpha)} \left(\frac{t}{\beta} \right)^x \left(1 + \frac{t}{\beta} \right)^{-(\alpha+x)}.$$

- α and β are parameters of gamma prior distribution for λ



single parameter estimation.xls

Exponential Likelihood – Gamma Prior

- *It can be shown that $f(\Sigma t_i | \lambda)$ is $\text{gamma}(n, \lambda)$*
- *Can then show that, for a $\text{gamma}(\alpha, \beta)$ prior on λ*

$$\frac{\sum T_i}{\sum T_i + \beta} \sim \text{beta}(n, \alpha)$$

- *If $X \sim \text{beta}(\alpha, \beta)$, can show that $(1 - X) \sim \text{beta}(\beta, \alpha)$, therefore*

$$\frac{\beta}{\sum T_i + \beta} \sim \text{beta}(\alpha, n)$$

- *Can use these in Excel*

Posterior Predictive Distribution

- Gives conditional probability of seeing a **new** set of data, x_{rep} , given the old set, x
- In symbols

$$f(x_{rep} | x) = \int f(x_{rep} | \theta) \pi_1(\theta | x) d\theta$$

- For beta-binomial and gamma-Poisson cases
 - Use earlier formulae, but use **posterior** α and β
- Posterior predictive distribution is primary tool for Bayesian model validation
 - Focuses on predictive validity of model (prior + data)

Using Posterior Predictive Distribution to Check Model

- *Is x_{rep} in tail of $f(x_{rep}|x)$?*
 - *No \Rightarrow model OK*
 - *Yes \Rightarrow problem with prior and/or likelihood*
 - *Check prior sensitivity*
 - *Prior-dominated: sharp prior, sparse data, likelihood function centered away from mode of prior*
 - *Check appropriateness of likelihood*
 - *For example, are failures independent?*

Using WinBUGS to Simulate Predictive Distributions

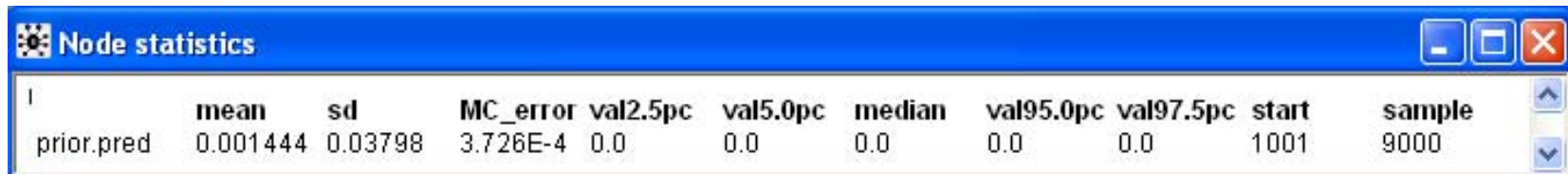
- *Can handle both conjugate and nonconjugate priors*
- *To simulate from prior predictive distribution, compile program **without** loading data*
- *Example: using industry prior for EDG FTS of $\text{beta}(0.957, 190)$*
 - *Will 3 failures in 24 demands be an extreme value in the prior predictive distribution for X ?*
 - *Can use Excel (see “single parameter estimation.xls” in Excel folder)*
 - *$\Pr(X \geq 3) = 0.001$, so 3 failures in 24 demands is **not consistent** with the industry prior*

Using WinBUGS to Simulate Predictive Distributions

- *Use WinBUGS program below*

```
EDG FTS
model {
  edg.fail ~ dbin(p, 24) #Binomial model for EDG FTS in 24
  demands
  p ~ dbeta(0.957, 190) #Beta prior for p
  prior.pred <- step(edg.fail - 3) # Prior pred. Pr(edg.fail >=3)
}
```

- *Compile this program, monitor prior.pred*

A screenshot of the 'Node statistics' window in WinBUGS. The window has a blue title bar with the text 'Node statistics' and standard window controls (minimize, maximize, close). Below the title bar is a table with 11 columns: 'I', 'mean', 'sd', 'MC_error', 'val2.5pc', 'val5.0pc', 'median', 'val95.0pc', 'val97.5pc', 'start', and 'sample'. The table contains one row of data for the node 'prior.pred'.

I	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
prior.pred	0.001444	0.03798	3.726E-4	0.0	0.0	0.0	0.0	0.0	1001	9000

Using WinBUGS to Simulate Predictive Distributions

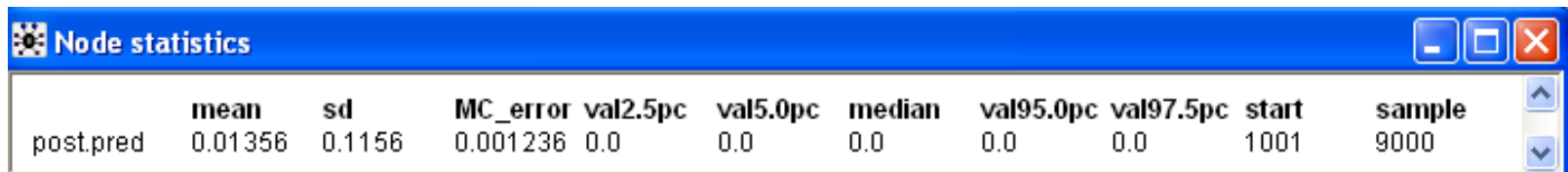
- *To simulate from posterior predictive distribution, define new variable and monitor it*

```
EDG FTS
model {
  edg.fail ~ dbin(p,24) #Binomial model for EDG FTS in 24 demands
  p ~ dbeta(0.957, 190) #Beta prior for p
  fail.new ~ dbin(p,24) #Replicated failures from posterior predictive distribution
  post.pred <- step(fail.new - 3) #Posterior pred. Pr(fail.new >= 3)
}
data
list(edg.fail=3)
```

- ***Fail.new** is node for posterior predictive distribution*
- *Observed data is 3 failures in 24 demands*

Using WinBUGS to Simulate Predictive Distributions

- *Posterior distribution of p is $\text{beta}(3.957, 211)$, giving posterior mean of 0.02*
- *Posterior predictive distribution for X shown in Excel file “single parameter estimation.xls”*
- *WinBUGS results (remember to load data this time!)*

A screenshot of the WinBUGS 'Node statistics' window. The window has a blue title bar with the text 'Node statistics' and standard Windows window controls (minimize, maximize, close). Below the title bar is a table with 11 columns: 'mean', 'sd', 'MC_error', 'val2.5pc', 'val5.0pc', 'median', 'val95.0pc', 'val97.5pc', 'start', and 'sample'. The first column is labeled 'post.pred'. The data row shows values: mean 0.01356, sd 0.1156, MC_error 0.001236, val2.5pc 0.0, val5.0pc 0.0, median 0.0, val95.0pc 0.0, val97.5pc 0.0, start 1001, and sample 9000. There are up and down arrow buttons on the right side of the table.

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
post.pred	0.01356	0.1156	0.001236	0.0	0.0	0.0	0.0	0.0	1001	9000

- *3 failures in 24 demands is quite **unlikely***
 - *Prior doesn't describe our diesel?*
 - *Already know that prior is inconsistent with this data*
 - *Problem with binomial model for failure, also?*
 - *Are failures really independent?*

Use of Summary Statistics from Posterior Predictive Distribution

- *Is data exponential (i.e., is likelihood correct)?*
 - *Use $\sum t_i$ and examine posterior predictive distribution for $\sum T_{rep, i}$*
 - *Where does observed $\sum t_i$ fall in posterior predictive distribution for $\sum T_{rep, i}$?*
 - *Use diffuse prior for λ to eliminate effect of prior*
 - *Observed $\sum t_i$ will always be at median, so we need another approach*

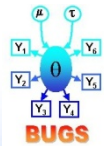
Are Times Exponential?

- *Cramer-Von Mises test, Bayesian version*
- *Uses fact that $F(t) \sim \text{uniform}(0, 1)$*
- *Bayesian version in following script compares observed and replicated values*
- *Example: fire suppression times in minutes:
1.7, 1.8, 1.9, 5.8, 10.0, 11.3, 14.3, 16.6, 19.4, 54.8 (n = 10)*

Are Times Exponential?

```
model
{
  for (i in 1 : N) {
    time.sup[i] ~ dexp(lambda) #Exponential distribution for suppression times
    time.sup.ranked[i] <- ranked(time.sup[,i])
    time.rep[i] ~ dexp(lambda)
    time.rep.ranked[i] <- ranked(time.rep[, i])
    F.obs[i] <- 1 - exp(-lambda*time.sup.ranked[i])
    F.rep[i] <- 1 - exp(-lambda*time.rep.ranked[i])
    diff.obs[i] <- pow(F.obs[i] - (2*i-1)/(2*N), 2)
    diff.rep[i] <- pow(F.rep[i] - (2*i-1)/(2*N), 2)
  }
  lambda ~ dgamma(0.0001,0.0001) #Diffuse prior for exponential parameter
  CVM.obs <- sum(diff.obs[])
  CVM.rep <- sum(diff.rep[])
  p.value <- step(CVM.rep - CVM.obs) #Small value indicates problem with exponential likelihood
}

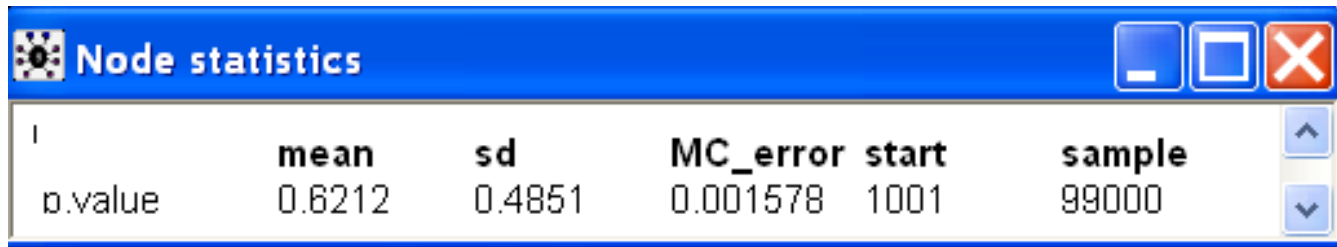
data
list(time.sup=c(1.7 ,1.8, 1.9, 5.8, 10.0, 11.3, 14.3, 16.6, 19.4, 54.8), N=10)
```



suppression rate data.odc

Are Times Exponential?

- *Run 100,000 samples with 1,000 burn-in iterations and monitor mean of p.value node*

A screenshot of a software window titled "Node statistics". The window has a blue header bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. Below the header is a table with two rows. The first row contains the headers: "p.value", "mean", "sd", "MC_error", "start", and "sample". The second row contains the corresponding values: "0.6212", "0.4851", "0.001578", "1001", and "99000". There are up and down arrow buttons on the right side of the table.

p.value	mean	sd	MC_error	start	sample
0.6212	0.4851	0.001578	1001	99000	

- *Value of 0.62 indicates no significant deviation from exponential distribution*

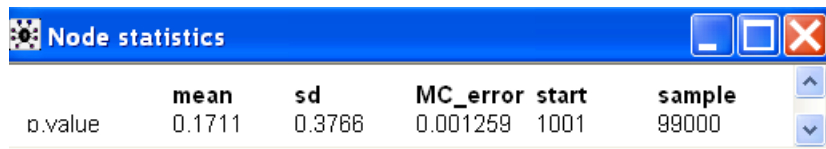
Are Times Exponential?

- *Example: LOSP recovery times in hrs.*

0.2, 37.3, 0.5, 4.3, 80.1, 13.3, 2.1, 3.7, 8.2, 2.8

$n = 10$

- *Run WinBUGS again with these data*

A screenshot of the 'Node statistics' window from the WinBUGS software. The window has a blue title bar with the text 'Node statistics' and standard window control buttons (minimize, maximize, close). Below the title bar is a table with six columns: 'p.value', 'mean', 'sd', 'MC_error', 'start', and 'sample'. The table contains one row of data.

p.value	mean	sd	MC_error	start	sample
0.1711	0.3786	0.001259	1001		99000

- *Smaller p.value suggests exponential distribution may not be adequate aleatory model*

Testing for Population Variability in p

- *Recall EDG FTS data from earlier*
- *Side-by-side plot of credible intervals suggested there is variability in p*
 - *Simple binomial model then not appropriate*
- *Can we quantify this with a test?*
- *Yes!*

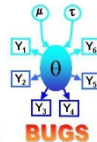
	<i>Failures</i>	<i>Demands</i>
<i>Plant 1</i>	<i>0</i>	<i>140</i>
<i>Plant 2</i>	<i>0</i>	<i>130</i>
<i>Plant 3</i>	<i>0</i>	<i>130</i>
<i>Plant 4</i>	<i>1</i>	<i>130</i>
<i>Plant 5</i>	<i>2</i>	<i>100</i>
<i>Plant 6</i>	<i>3</i>	<i>185</i>
<i>Plant 7</i>	<i>3</i>	<i>175</i>
<i>Plant 8</i>	<i>4</i>	<i>167</i>
<i>Plant 9</i>	<i>5</i>	<i>151</i>
<i>Plant 10</i>	<i>10</i>	<i>150</i>

Testing for Population Variability in p

- *Will use Bayesian analog of chi-square statistic, along with posterior predictive distribution for X , via WinBUGS*
 1. *Specify binomial model for X*
 2. *Diffuse prior for p to eliminate prior sensitivity*
 3. *Generate replicate values from posterior predictive distribution for X , with estimate of p based on pooled data*
 - *What we would expect to see if all p 's are equal*
 4. *Calculate observed and replicate chi-square statistics*
 - *Compares what we see to what we would see, on average, if all p 's are equal*
 5. *Is difference significantly different from 0?*
 - *Yes \Rightarrow variability*

Testing for Population Variability in p

```
model {  
  for (i in 1 : N) {  
    x[i] ~ dbin(p[i], n[i]) #Binomial model for X  
    p[i] <- p.constant #All sources have same p  
    x.rep[i] ~ dbin(p[i], n[i]) #Replicate from posterior predictive distribution  
    diff.obs[i] <- pow(x[i] - p[i]*n[i], 2)/(n[i]*p[i]*(1-p[i])) #Difference between observed  
    and expected x  
    diff.rep[i] <- pow(x.rep[i] - p[i]*n[i], 2)/(n[i]*p[i]*(1-p[i])) #Difference between  
    replicated and expected x  
  }  
  chisq.obs <- sum(diff.obs[])  
  chisq.rep <- sum(diff.rep[])  
  p.value <- step(chisq.rep - chisq.obs) #Value should be near 0.5 for homogeneous data  
  p.constant ~ dbeta(0.5, 0.5) #Jeffreys prior for p  
}
```



Bayesian chi-square test for EDG FTS.odc

Testing for Population Variability in p

- *Run program with EDG FTS data*
 - *Let WinBUGS generate initial values*
- *Monitor mean of p.value node*

A screenshot of the 'Node statistics' window from the WinBUGS software. The window has a blue title bar with the text 'Node statistics' and standard Windows window controls (minimize, maximize, close). Below the title bar is a table with six columns: 'p.value', 'mean', 'sd', 'MC_error', 'start', and 'sample'. The table contains one row of data.

	mean	sd	MC_error	start	sample
p.value	0.006051	0.07755	2.506E-4	1001	99000

- *Small value indicates data nonhomogeneous population*

Testing for Population Variability in p

- *p .value is small*
 - *Analogous with frequentist p -value*
- *Strong evidence of variability in p*
 - *Simple binomial model inadequate*
 - *p not the same for all EDGs*
 - *Need more complex model describing variation in p*
 - *More complex model covered in Section 5*

Testing for Population Variability in λ

- *Another example: LOSP frequency*

<i>Plant</i>	<i>Failures</i>	<i>Time (yrs)</i>
<i>Plant 1</i>	2	15.986
<i>Plant 2</i>	1	16.878
<i>Plant 3</i>	1	18.146
<i>Plant 4</i>	1	18.636
<i>Plant 5</i>	2	18.792
<i>Plant 6</i>	0	18.976
<i>Plant 7</i>	12	18.522
<i>Plant 8</i>	5	19.04
<i>Plant 9</i>	0	18.784
<i>Plant 10</i>	3	18.868
<i>Plant 11</i>	0	19.232

Testing for Population Variability in λ

- *Earlier plot of credible intervals suggested variability*
- *Again, use Bayesian version of chi-square statistic based on replicates from posterior predictive distribution*

Testing for Population Variability in λ

```
model {  
  for (i in 1 : N) {  
    mu[i] <- lambda.losp[i] * time[i] #Poisson parameter for each plant  
    x[i] ~ dpois(mu[i]) #Poisson dist. for events at each plant  
    lambda.losp[i] <- lambda.const #Constant value for lambda  
    x.rep[i] ~ dpois(mu[i])  
    diff.obs[i] <- pow(x[i] - mu[i], 2)/mu[i]  
    diff.rep[i] <- pow(x.rep[i] - mu[i], 2)/mu[i]  
  }  
  chisq.obs <- sum(diff.obs[])  
  chisq.rep <- sum(diff.rep[])  
  p.value <- step(chisq.rep - chisq.obs) #Value should be near 0.5 for homogeneous data  
  lambda.const ~ dgamma(0.5, 0.0001) #Jeffreys prior  
}
```



Bayesian chi-square test for losp frequency.odc

Testing for Population Variability in λ

- Results*

A screenshot of a software window titled "Node statistics". The window has a blue header bar with the title and standard window controls (minimize, maximize, close). Below the header is a table with statistical data. The table has two rows: a header row and a data row. The header row contains the following columns: "mean", "sd", "MC_error", "val5.0pc", "median", "val95.0pc", "start", and "sample". The data row contains the following values: "2.3E-4", "0.01516", "4.835E-5", "0.0", "0.0", "0.0", "1001", and "100000". There is also a column labeled "p.value" in the data row, but it is empty. The table is enclosed in a blue border.

	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
p.value	2.3E-4	0.01516	4.835E-5	0.0	0.0	0.0	1001	100000

- Strong evidence of plant-to-plant variability*

Model Selection via BIC and DIC

- *Previous portions of this section examined goodness-of-fit in absolute sense, using posterior predictive distribution, and summary statistics derived from it*
- *Can also examine relative fit of candidate models*
- *Various information criteria have been proposed to help with this task*
 1. *Bayesian Information Criterion (BIC)*
 2. *Deviance Information Criterion (DIC)*
- *Both of these are based on Akaike Information Criterion (AIC)*

Basic Form of Information Criteria

- BIC, DIC, and AIC all have same general form:*

$$IC = -2\log[f(\vec{t} \mid \vec{\theta})] + g(k)$$

Deviance

Penalty function

- Choose model with smallest IC*

Bayesian Information Criterion (BIC)

- *Uses $g(k) = k \log(n)$, where k is number of unknown parameters and n is number of observations (data points)*
- *Recommended by some for comparing nonhierarchical models*
- *Not calculated by WinBUGS directly, but easy to include in script*

BIC Example

- *Example: LOSP recovery times in hrs.*
0.2, 37.3, 0.5, 4.3, 80.1, 13.3, 2.1, 3.7, 8.2, 2.8
 $n = 10$
- *Earlier Bayesian goodness-of-fit test gave Bayesian p-value of 0.17 for exponential model*
- *Use BIC to compare exponential and Weibull model*
 - *Use diffuse priors on all parameters so that focus is on aleatory model rather than prior distributions*

BIC Example

```
model {  
  for(i in 1:N) {  
    #   time[i] ~ dexp(lambda) #Exponential aleatory model  
    #Exponential log-likelihood components  
    #   log.like[i] <- log(lambda) - lambda*time[i]  
    time[i] ~ dweib(beta, lambda) #Weibull aleatory model  
    #Weibull log-likelihood components  
    log.like[i] <- log(lambda) + log(beta) + (beta-1)*log(time[i]) - lambda*pow(time[i], beta)  
  }  
  lambda ~ dgamma(0.0001, 0.0001)  
  beta ~ dgamma(0.0001, 0.0001)  
  log.like.tot <- sum(log.like[])  
  #Exponential model  
  #BIC <- -2*log.like.tot + log(N)  
  #Weibull model  
  BIC <- -2*log.like.tot + log(N)*2  
}
```



BIC example.odc

BIC Example

- *Results*
 - *Exponential BIC = 77.8*
 - *Weibull BIC = 76.5*
- *Slight preference for Weibull model over exponential*
- *Caveat: BIC measures relative model fit*
 - *All models being considered could be poor in absolute sense*

Deviance Information Criterion (DIC)

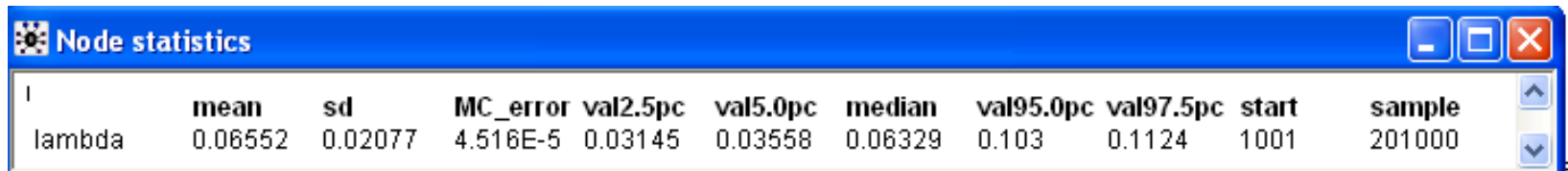
- *Take average deviance over posterior:*
$$Dbar = -2 \int \log[f(\vec{t} | \vec{\theta}) \pi_1(\vec{\theta} | \vec{t})] d\vec{\theta}$$
 - *Dbar is automatically monitored by WinBUGS node called “deviance”*
- *DIC = Dbar + pD*
 - *pD is effective number of parameters*
 - *pD = Dbar – Dhat*
 - *Dhat is deviance evaluated at posterior mean of parameter(s)*
- *Recommended by some for comparing hierarchical models*
 - *In our experience, can use DIC with simple models, too*

Deviance Information Criterion (DIC)

- *DIC (and even pD) can be negative in some cases*
 - *DIC negative when density function is > 1*
 - *Smallest DIC still indicates best fitting model*
 - *Example: three models with DICs of 10, -3, -9*
 - *Third model, with $DIC = -9$, is best fit*
 - *If pD is negative, cannot use DIC*
- *DIC is measure of relative goodness of fit*
 - *Model with smallest DIC can still be poor fit*

Using DIC to Explore Model Fit

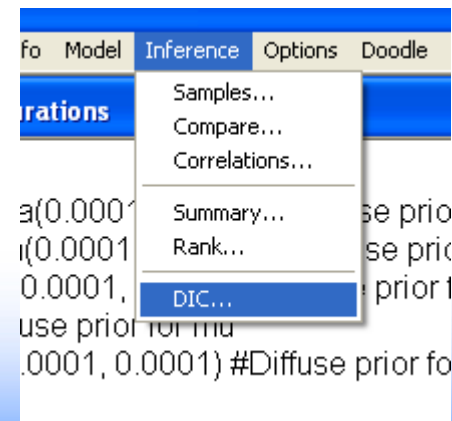
- *Run exponential model first*
- *Do run of 100,000 samples to estimate λ (burn in of 1,000)*
- *Results*



A screenshot of a software window titled "Node statistics". It contains a table with statistical data for a parameter named "lambda".

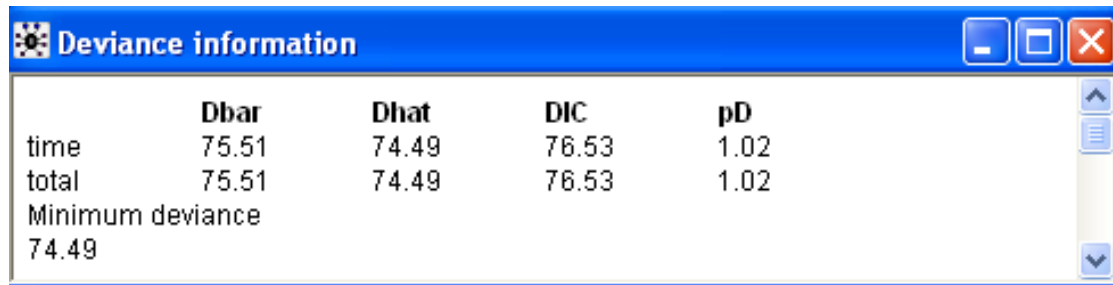
	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
lambda	0.06552	0.02077	4.516E-5	0.03145	0.03558	0.06329	0.103	0.1124	1001	201000

- *Now select “DIC” from “Inference” menu*



Using DIC to Explore Model Fit

- *Set DIC, then run 100,000 more updates*
- *Results for exponential model:*



A screenshot of a software window titled "Deviance information". The window contains a table with five columns: "time", "Dbar", "Dhat", "DIC", and "pD". The table has three rows of data. Below the table, it states "Minimum deviance" followed by the value "74.49". The window has standard Windows-style controls (minimize, maximize, close) in the top right corner.

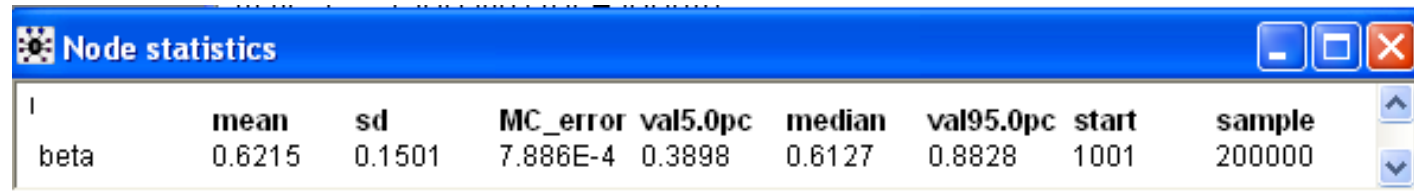
	Dbar	Dhat	DIC	pD
time	75.51	74.49	76.53	1.02
total	75.51	74.49	76.53	1.02

Minimum deviance
74.49

- *Repeat for other possible models*

Using DIC to Explore Model Fit

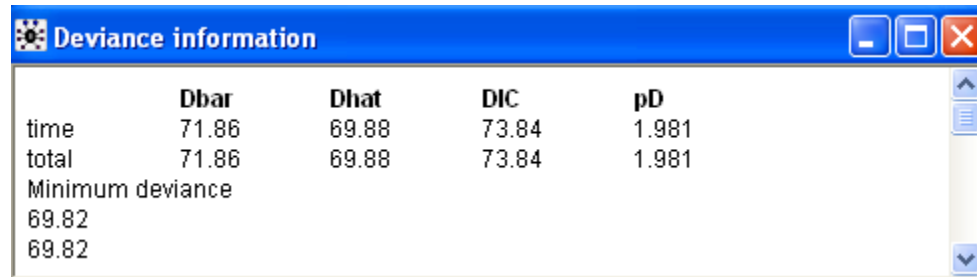
- *Weibull*



A screenshot of a software window titled "Node statistics". It contains a table with the following data:

	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
beta	0.6215	0.1501	7.886E-4	0.3898	0.6127	0.8828	1001	200000

– $\beta < 1 \Rightarrow$ decreasing recovery rate



A screenshot of a software window titled "Deviance information". It contains a table with the following data:

	Dbar	Dhat	DIC	pD
time	71.86	69.88	73.84	1.981
total	71.86	69.88	73.84	1.981
Minimum deviance	69.82			
	69.82			

- $DIC < DIC$ for exponential model
- *Weibull is better fit*
 - *Consistent with results from BIC*

Exercises

1. *The table below shows successful launches/launch attempts for a series of launch vehicles developed after 1980.*

Vehicle	Outcome
Pegasus	9/10
Percheron	0/1
AMROC	0/1
Conestoga	0/1
Ariane-1	9/11
India SLV-3	3/4
India ASLV	2/4
India PSLV	6/7
Shavit	2/4
Taepodong	0/1
Brazil VLS	0/2

Exercises

1. Continued----

- a) *Using a binomial aleatory model for the number of successes for each vehicle, is a common success probability (i.e., no vehicle-to-vehicle variability) a reasonable choice? Use Bayesian p-value to answer this question.*
- b) *Let the success probability vary from vehicle to vehicle according to a beta distribution with parameters $K\delta$ and $K(1 - \delta)$, where δ is the expected success probability before any data are observed, and K controls the dispersion of the population variability distribution. Model this in BUGS using a $\text{beta}(0.5, 0.5)$ hyperprior for δ and a $\text{gamma}(5, 1)$ hyperprior for K . Find the posterior mean and 90% credible interval for success probability of a future launch. How does the Bayesian p-value for this model compare with the constant model above? Has the marginal posterior distribution for K been affected much by the observed data? What might this suggest about sensitivity studies for the hyperprior on K ?*

Exercises

2. *The following projector lamp failure times (in hours) have been collected: 387, 182, 244, 600, 627, 332, 418, 300, 798, 584, 660, 39, 274, 174, 50, 34, 1895, 158, 974, 345, 1755, 1752, 473, 81, 954, 1407, 230, 464, 380, 131, 1205.*
- a) *The vendor has provided an estimate of the mean time to failure (MTTF) for the lamp. This estimate is 1000 hours. Use this value to develop a prior distribution for the lamp failure rate, assuming the time to failure is exponentially distributed.*
 - b) *Compare the posterior distribution for the failure rate with this prior to what would have been obtained using the Jeffreys noninformative prior for the failure rate.*

Exercises

- c) *Use a Weibull aleatory model for the failure time, with diffuse priors on the Weibull parameters. What is the posterior probability that the Weibull shape parameter exceeds 1? What does this suggest about the viability of the Weibull model compared with the exponential model?*
- d) *Use BIC and DIC to compare exponential, Weibull, and lognormal aleatory failure time models for the lamp.*