AFRL-ML-WP-TR-2003-4096

NONDESTRUCTIVE EVALUATION (NDE) TECHNOLOGY INITIATIVES (NTIP)

Delivery Order 0039: Statistical Comparison of Competing Material Models

Charles Annis, P.E.

Statistical Engineering
36 Governors Court
Palm Beach Gardens, FL 33418-7161

JANUARY 2003

Final Report for 22 July 2002 – 20 December 2002

Approved for public release; distribution is unlimited.

MATERIALS AND MANUFACTURING DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7750

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE U.S. GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY AND RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY RELATE TO THEM.

THIS REPORT HAS BEEN REVIEWED BY THE OFFICE OF PUBLIC AFFAIRS (ASC/PA) AND IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Juan D. Calzada

JUAN CALZADA, Project Engineer
Nondestructive Evaluation Branch
Metals, Ceramics & NDE Division

James C. Malas

JAMES C. MALAS, Chief
Nondestructive Evaluation Branch
Metals, Ceramics & NDE Division

Gerald J. Petrak

GERALD J. PETRAK, Assistant Chief
Metals, Ceramics & Nondestructive
Evaluation Division
Materials & Manufacturing Directorate

IF YOUR ADDRESS HAS CHANGED, IF YOU WISH TO BE REMOVED FROM OUR MAILING LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION, PLEASE NOTIFY, AFRL/MLLP, WRIGHT-PATTERSON AFB, OH 45433-7750 TO HELP US MAINTAIN A CURRENT MAILING LIST.

COPIES OF THIS REPORT SHOULD NOT BE RETURNED UNLESS RETURN IS REQUIRED BY SECURITY CONSIDERATIONS, CONTRACTUAL OBLIGATIONS, OR NOTICE ON A SPECIFIC DOCUMENT.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
January 2003	Final	07/22/2002 – 12/20/2002

4. TITLE AND SUBTITLE
NONDESTRUCTIVE EVALUATION (NDE) TECHNOLOGY INITIATIVES (NTIP)
Delivery Order 0039: Statistical Comparison of Competing Material Models

5a. CONTRACT NUMBER
F33615-97-D-5271

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62102F

6. AUTHOR(S)
Charles Annis, P.E.

5d. PROJECT NUMBER
4349

5e. TASK NUMBER
40

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Statistical Engineering
36 Governors Court
Palm Beach Gardens, FL 33418-7161

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Materials and Manufacturing Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7750

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/MLLP

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-ML-WP-TR-2003-4096

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
This report contains color.

14. ABSTRACT ***(Maximum 200 Words)***

The ability to decide objectively between competing material models has always been an important engineering task. For a variety of reasons, simple inspection of the general "fit" of various models to the appropriate data does not always result in an obvious choice for the superior model, and heretofore there has been no statistical measure that can be used to choose between models (except in the simple case of "nested" models). Bayes factors overcome these previous limitations and can provide a valid objective statistical measure for choosing between competing material models. This study explores the use of Bayes factors as an objective measure in choosing between competing material models. The focus is on the use of this method by engineers who may or may not be well versed in statistical methods. In addition, the statistical background necessary to gain an in-depth understanding of the Bayes method is included as appendices. Also discussed in some detail is the Random Fatigue Limit s-N model proposed by Pascal and Meeker, which holds considerable promise for describing HCF behavior, as well as the applicability of applying Bayes factors to this model.

15. SUBJECT TERMS
Bayes Factors, modeling, random fatigue limit

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Juan Calzada
Unclassified	Unclassified	Unclassified	SAR	62	19b. TELEPHONE NUMBER *(Include Area Code)* (937) 255-1605

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Preface
i

Table of Contents

Preface

ii

A Statistical Criterion for Judging Material Models

Using Bayes Factors to Select the Best Model

Executive Summary:

1) The ability to decide objectively between competing material models has always been an important engineering task. For a variety of reasons, simple inspection of the general "fit" of various models to the appropriate data does not always result in an obvious choice for the superior model, and heretofore there has been no statistical measure that can be used to choose between models (except in the simple case of "nested" models). Bayes factors overcome these previous limitations and can provide a valid objective statistical measure for choosing between competing material models.

2) In plain English, the **Bayes factor** is the ratio of the probability that a model is correct to the probability that it is not: It is the **odds** of being correct. A more precise definition is:

> *The Bayes factor is the (posterior) odds favoring one model versus another when the prior oddd of the two models are equal. After mathematical simplification that is:*
>
> $B_{1,2} = P(\text{data} \mid \text{explanatory model}_1) / P(\text{data} \mid \text{explanatory model}_2)$

3) Unlike conventional hypothesis testing which sets up a null hypothesis and then tries to disprove it, Bayes factors provide a mechanism for evaluating evidence *in favor of* one model over another.

4) Also unlike conventional hypothesis testing, Bayes factors do *not* require that competing models be "nested" (i.e.: that one model be a subset of a more complex model, as with a polynomial model seeking to include or exclude a higher order term). Thus Bayes factors can chose among more sophisticated models which might be rather different mathematically.

Acknowledgement:

This work was supported by Independent Contractor Agreement 02-S437-039-C1, Universal Technology Corporation Prime Contract F33615-97-D-5271, Task Order 0027, Delivery Order 0039, under the technical direction of Dr. Ryan Morrissey, AFRL/MLLMN, Program Manager.

Introduction:

Explaining Bayes factors to an engineering readership perhaps unfamiliar with the distinction between say, probability and statistics[1], is not unlike explaining a stress intensity factor to a statistical readership unfamiliar with the distinction between stress and strain[1]. While elucidating the statistical underpinnings is necessary, this should augment, not compete with, the central exposition. To make the requisite material available, but still begin with the central topic – Bayes factors – I have organized the supporting material into appendices, rather than integrating the material into the text, which would make the reading tedious. Thus the reader learns immediately what Bayes factors are and how to calculate them, while still having access to the critical background material.

How This Report Is Organized:

This report is organized into three sections followed by nine appendices, as follows:

1) Introductory material and table of contents,

2) Section 1, Bayes factors,

3) Section 2, How to Calculate Bayes factors,

4) Section 3, Random Fatigue Limit Data and Model,

5) Appendices A and B, WinBUGS (the software necessary to execute Bayesian regression) and the supporting code,

6) Appendices C and D, the R-Project (statistical software for computing Bayes factors from the WinBUGS output) and supporting code

7) Appendix E, Review of Statistical Fundamentals,

8) Appendix F, Overview of Regression analysis,

9) Appendix G Notes on Goodness-of-Fit for Statistical Distributions,

10) Appendix H, A very brief recent chronology of Bayesian methods,

11) Appendix I, Bibliography.

It was my original intension to use the random fatigue limit model as the central comparative example, however, deficiencies in the model's ability to describe the Ti6Al4V data, discussed herein, precluded extensive use of the model for that purpose. None of this detracts from the utility of Bayes factors in comparing material models, but it does suggest that further work on the RFL model itself is necessary before the RFL model can be accepted into more widespread use.

[1] There's a difference?

Section

1

Bayes Factors

Bayes's Theorem and Bayes Factors.

Bayes's Theorem may look complicated but behind the many integral signs is a rather simple statement of joint probability, *viz*. the joint probability of two events (say, the data and the prediction) is the product of their conditional probabilities.

For example, let the experiment be A and the prediction, B. Both have occurred, AB. The probability of both A *and* B together is $P(AB)$. . The law of conditional probability says that this probability can be found as the product of the conditional probability of one, given the other, times the probability of the other. That is

$P(A|B) \times P(B) = P(AB) = P(B|A) \times P(A)$, if both $P(A)$ *and* $P(B)$ are non zero.

Simple algebra shows that:

$$\boldsymbol{P(B|A) = P(A|B) \times P(B) / P(A)} \qquad \text{Equation 1}$$

This is Bayes's Theorem is its simplest form. In words this says that the posterior probability of B (the updated prediction) is proportional to the product of the conditional probability of the experiment, given the influence of the parameters being investigated, times the prior probability of those parameters, with proportionality constant 1/$P(A)$.

It should be immediately obvious that equation 1 has no off-putting integrals and thus may not be recognized as Bayes's theorem. While the simple definition uses only fixed probabilities for events A *and* B, for most real applications the probabilities involved are not single-valued but rather probability densities.

Further, $P(A)$ (the total probability of the data) is seldom known directly and must be computed by summing over (or integrating over) all possible values for the parameter, θ. So $\boldsymbol{y}$ represents the data, and θ indexes the probability density, and can be a *k*-dimension vector. Thus, A, in equation 1 is replaced by $\boldsymbol{y}$, and P($\boldsymbol{y}$) becomes the marginal probability density of the data, found by integrating out the parameter vector θ.

$$P(\theta \mid y) = \frac{P(y \mid \theta)\pi(\theta)}{P(y)} \qquad \text{Equation 2}$$

where

$$P(y) = \iint \ldots \int P(y \mid \theta)\pi(\theta)\, d\theta$$

The object then is not to find a single posterior probability, but a probability density, indexed by θ. For example $\boldsymbol{\theta} = (\mu, \sigma^2)^T$ for a normal distribution, where μ is the mean, and σ is the standard deviation, or for a regression model where $\mu_k = \beta_0 + \beta_1 X_k$, $\boldsymbol{\theta} = (\beta_0, \beta_1, \sigma^2)^T$. We will be interested in the ratio of the posterior probability of the data, resulting from one model of material behavior to the posterior probability under a different material model. If the resulting odds – the Bayes factor - is sufficiently large (see table 1 in the following section) then the data argue for the superiority of one model over another.

Following the convention[2] of the Bayesian literature, $\pi(\boldsymbol{\theta})$ is the **prior** density of the model's parameters θ.[3] Thus the **marginal distribution** of the data, $\boldsymbol{y}$, (see Appendix E, figure 8) can be determined by integrating over the model parameter space:

$$m(\mathbf{y} \mid M_1) = \int f(\mathbf{y} \mid \theta_1, M_1)\pi_1(\theta_1)d\theta_1 \qquad \text{Equation 3a}$$

and

$$m(\boldsymbol{y} \mid M_2) = \int f(\boldsymbol{y} \mid \theta_2, M_2)\pi_2(\theta_2)d\theta_2 \qquad \text{Equation 3b}$$

where f is the **conditional probability** density of the data, given the model and its parameters. (Conditional, marginal, and joint probability are reviewed in Appendix F.) Notice that we are *not* maximizing over the parameter space, but integrating over it.

So for both models, the marginal probability of the data is found by summing over both models:

$$m(\boldsymbol{y}) = m(\boldsymbol{y} \mid M_1)\pi(M_1) + m(\boldsymbol{y} \mid M_2)\pi(M_2) \qquad \text{Equation 4}$$
$$= \int f(\boldsymbol{y} \mid \theta_1, M_1)\pi(M_1)\pi_1(\theta_1)d\theta_1 + \int f(\boldsymbol{y} \mid \theta_2, M_2)\pi(M_2)\pi_2(\theta_2)d\theta_2$$

Finally from equation 3 as the part of the numerator and equation 4 as the denominator we can write Bayes's theorem (equation 2) as

$$P(M_1 \mid \boldsymbol{y}) = \frac{m(\boldsymbol{y} \mid M_1) \times \pi_1(M_1)}{m(\boldsymbol{y} \mid M_1)\pi(M_1) + m(\boldsymbol{y} \mid M_2)\pi(M_2)} \qquad \text{Equation 5a}$$

and

---

[2] While this report is written by an engineer for an engineering readership, the ubiquity of Bayesian concepts mandates a familiarity with the notational conventions of that literature, even though it does appear strange to the engineer's eye. Thus the Greek letter $\boldsymbol{\pi}$ will indicate a prior density unless explicitly stated as being the more familiar geometric ratio of circumference to diameter. The use of the Greek letter $\boldsymbol{\pi}$ is to avoid confusion with the Latin $\boldsymbol{p}$ when both are used in the same equation.

[3] In many circumstances the model parameters are constants (e.g. the mean, $\mu = const.$). In other situations the model parameters themselves have **hyperparametric** probability densities (e.g. $\mu = \phi(\eta, \tau^2)$). Of course these hyperparameters too can have hyper-hyperparameters, but as a practical matter such further hyperparameterization beyond one level is of little value, unless there is, or will be, data at those levels to help estimate them.

$$P(M_2 \mid \mathbf{y}) = \frac{m(\mathbf{y} \mid M_2) \times \pi_1(M_2)}{m(\mathbf{y} \mid M_1)\pi(M_1) + m(\mathbf{y} \mid M_2)\pi(M_2)} \qquad \text{Equation 5b}$$

and if the prior probabilities are equal, i.e. neither model is favored before considering the data, $\pi_1(M_1) = \pi_2(M_2) = ½$ and thus the prior odds are ½ / ½ = 1:1 = 1.

We are now ready to calculate the Bayes factor.

The Bayes factor is really an odds ratio that reduces to the posterior odds favoring one model over another when their prior odds are equal (i.e.: where no preference is given to either model in advance of the data).

$$B_{1,2} = \frac{posterior\ odds_{M1:M2}}{prior\ odds_{M1:M2}} =$$

$$= \frac{m(M_1 \mid \mathbf{y}) / m(M_2 \mid \mathbf{y})}{\pi_1(M_1) / \pi_2(M_2)} \qquad \text{Equation 6}$$

$$= \frac{\left[\frac{m(M_1 \mid \mathbf{y}) \times \pi_1(M_1)}{m(\mathbf{y})}\right] / \left[\frac{m(M_2 \mid \mathbf{y}) \times \pi_2(M_2)}{m(\mathbf{y})}\right]}{\pi_1(M_1) / \pi_2(M_2)}$$

$$= \frac{m(\mathbf{y} \mid M_1)}{m(\mathbf{y} \mid M_2)} \qquad \text{Equation 7}$$

So that the *posterior odds = Bayes factor* X *prior odds.* When the prior odds are equal, i.e. 1:1, then the Bayes factor equals the posterior odds. So the Bayes factor favoring model 1 over model 2 would be

$$B_{1,2} = P(\text{data} \mid \text{explanatory model}_1) / P(\text{data} \mid \text{explanatory model}_2),$$

which is read "The Bayes factor favoring Model 1 over Model 2 is the ratio of the probability of the data, given model 1 to the probability of the data, given model 2," in short: the Bayes factor is the ratio of the marginal probabilities of the data (for equal prior odds). Thus to compute the Bayes factor favoring one material model over another one need only compute these two marginal probabilities. This is no more difficult than "belling the cat."[4]

Practical Issues in Computing the Bayes Factor

It is one thing to define a procedure, and quite another thing to carry it out. In fact, the onerous computational difficulties associated with Bayesian statistics kept the discipline estranged from practical application and within the halls of Academia until only recently.

[4] In the children's fairy tale the mice, under continuous threat from the cat, decided that if they had some warning of the cat's proximity they could safely hide. They decided to hang a noisy bell around the cat's neck as this would surely provide ample warning. But, they discovered, the implementation of their simple plan was far from simple. Calculating Bayes factors is simple in theory but less so in practice.

The re-discovery during the 1990s of Markov Chain Monte Carlo[5] removed the computational fetters from Bayesian methods and allowed them to flourish. While Bayesian computations are now feasible, they still are rather involved, and has been observed by Han and Carlin (2001), " ... all methods require significant human and computer effort ...")

Guidelines for Interpreting Bayes Factors[6]

The following table can be used to interpret Bayes Factors.

Table 1 Bayes Factor Criteria

$B_{1,2}$	$2\ ln\ B_{1,2}$	Interpretation
Less than 1:1	Negative	Supports M_2
1:1 to 3:1	0 to 2	Weak support for M_1
3:1 to 20:1	2 – 6	Support for M_1
20:1 to 150:1	6 – 10	Strong evidence favoring M_1
Greater than 150:1	Over 10	Very strong support for M_1

Notice that a Bayes Factor of 20:1 resembles the frequentist "significance level" of 5%.

The Bayes factor comparing two models for compressive strength of *radiata* pine (a literature referee problem discussed in detail in Section 2 and in Appendix D) is **4852**, strong evidence indeed of the superiority of one model over its rival.

A more familiar problem is the choice between using, say, the Smith-Watson-Topper parameter or the Walker equivalent stress parameter in describing s-N behavior. It was originally planned to demonstrate Bayes Factors using the Random Fatigue Limit model with the entire 102-point Ti-6Al-4V dataset, and the RFL model is examined in some detail in Section 3. Unfortunately the statistical assumptions for using the RFL model do not hold for this dataset, and may be problematic for other data as well. Nevertheless, using only those data with cycle counts less than less than $1.2X10^5$ and a model without a mixture of probability densities, does illustrate a Bayes factor comparison.

The Bayes factor is **22.4** in favor of the SWT against the $S_{equivalent}$ parameterization for this limited dataset which, according to Table 1 above, shows moderate to strong support for the SWT parameterization. Please note that this is NOT a universal conclusion and is only presented here to illustrate Bayes factors with real fatigue data. A listing of the calculations is presented in Appendix D.

But how do you actually calculate the Bayes Factor? The next section uses a referee problem from the statistics literature to compare two simple linear regression models describing the compressive strength of *radiata* pine, as a function of its density, x, or its

[5] Markov Chain Monte Carlo (MCMC) should not be confused with the more familiar Monte Carlo sampling which shares a similar name but little else. A comparison of these very different methods can be found in Annis, Charles, "Modeling High Cycle Fatigue with Markov Chain Monte Carlo: A New Look at an Old Idea," AIAA 2002-13800, presented at 43rd AIAA/ASME/ASCE/AHS Structures and Dynamics Conference, Denver, CO, 22-25 April, 2002

[6] Table 1 is adapted from Kass and Rafferty (1995) and Congdon (2001).

density adjusted for resin content, z, since resin contributes to the density but not to the strength of the wood. While wood is not usually a material of interest to flight propulsion, this example is well-suited to demonstrate the effectiveness of Bayes Factors in comparing two similar models. Here, the form of the models is identical but the explanatory variables differ. An analogous situation might be using stress, rather than strain in an s-N model (or competing parameterizations for describing stress). Since the *radiata* dataset has been widely studied, it also provides a demonstration that the methods reported here are valid.

Section

2

How To Calculate Bayes Factors

Overview:

The steps for calculating Bayes Factors are presented here, followed by more detailed instructions for each step.

1) For each model, perform a Bayesian regression to determine the numerical estimates of the model parameters, and their posterior densities.

    a) In most cases the posterior densities of the model parameters will be Normal, providing a posterior *mean* and *standard deviation*.

    b) In most cases the posterior density for the variance will inverse gamma, or equivalently the density for the precision (precision = 1/variance) will be gamma, with parameters for *shape* and *scale*.

2) Create a post-convergence MCMC chain using the converged values for the model parameters in place of their densities, and update so that sampling for the precision comes from the post-convergence gamma density parameters.

    a) Continue the sampling to generate the post-convergence values for the precision density. This is the key to the Chib algorithm.

We can now calculate the marginal density of the data, given this model, $m(\mathbf{y} \mid Model)$. From equation 10 (page 15), we will need the log of the likelihood, the log of the prior, and the log of the posterior.

3) Compute $m(\mathbf{y} \mid Model)$ = exp(log.likelihood + log.prior – log.posterior)

4) Compute the Bayes Factor, BF = $m(\mathbf{y} \mid Model\ 1) / m(\mathbf{y} \mid Model\ 2)$

We will now discuss each of these steps in detail. In practice, steps 1 and 2 are performed using **WinBUGS**, and steps 3 and 4 using **R**. An example, using a referee case from the Bayesian literature (cf.: Han and Carlin, 2001) is provided in Appendix B (WinBUGS) and Appendix D, (R).

How to Perform a Bayesian Regression:

WinBUGS is the most popular commercially available software for Bayesian analysis (Kass, Carlin, Gelman, and Neal, 1998). The current (September, 2002) license fee is zero dollars ($0.00). Instructions for downloading the WinBUGS software package are provided in Appendix A. .

Building the Graphical Regression Model, Step-by-Step

While it is possible to write BUGS code directly, the code itself is not executed sequentially as it might appear from the code. The Gibbs algorithm evaluates each conditional density in tern and thus may take no heed of the order of statements in the code. Nevertheless, it is easy to avoid possible difficulties by doing all the coding graphically.

In WinBUGS a directed graph is called a Doodle. To create a Doodle like figure 7 in Appendix B, click Doodle, New, and OK to create a blank Doodle sheet. Click on the new sheet where you want to place the first node, say, the model's intercept, *alpha*. (If you create something you didn't intend, make sure it is highlighted, hold the Crtl key, and press the delete key.) The particulars of the new node appear at the top of the sheet. To activate a field, click on its name in blue, and the cursor will be positioned ready for your input. Type *alpha* in the blank for name. The default node type is stochastic, and the default density is normal, so no changes there are necessary. While the parameters for this node could be read in before executing the complied code, here we will enter them directly. For the mean, type *3000.*, and for the precision type *1.E-6*. Remember the precision is $1/\sigma^2$. This is the prior density for the parameter *alpha*. It is centered near its MLE (maximum likelihood estimate, as determined from either an earlier Bayesian regression or using a conventional estimate from **R**) and has a very large standard deviation of 1000. Such a prior density is called *vague*, since it allows the data to determine the final result.

Repeat the process to create the *beta* node. The density here is also normal with mean *185.*, and precision *1.E-4*, corresponding to a standard deviation of 100.

Next create the *y.calc[i]* node below the intercept (*alpha*) and slope (*beta*) nodes. To do that position the cursor where you want the node and click. Name the node *y.calc[i]* including the square brackets for the index, *i*. Now, while the node is highlighted, hold down the Ctrl key and change it to be logical, rather than stochastic, by clicking on type, and choosing logical from the drop-down menu. All logical nodes must be defined when you create them. In the value location type *alpha + beta * x[i]* . Now, while still editing the *y.calc[i]* node, hold down the Ctrl key and click the *alpha* node to draw the arrow. This arrow is only cosmetic, and is the only cosmetic feature of a Doodle, since only logical functions must be defined when they are created. Next, while still holding down the Ctrl key, click on the *beta* node to draw the second arrow.

Now create the precision node, *tau.* Click where you want the node to be located and enter its name. This is a stochastic node, with a gamma density. Click on "density" and choose dgamma from the drop-down menu. (Notice that all probability densities begin with "d.") Enter *3* and *1.8E5* for the *shape* and *scale* parameters, respectively. A plot of the gamma density for different values of *shape* and *scale* can be found in Appendix F.

Next, create the node for standard deviation, and label it *sig*. To create the node, click where you want it located. Change the type to logical, and for value, enter *1./sqrt(tau)*. Notice the decimal point. It is good practice to include the decimal when the value is non-integer.

Finally create the node for the observed value for y, *y.obs[i]*. and type in its name, including the bracketed counter, *[i]*. This is a stochastic node, with a normal density. Define its mean and precision graphically by first depressing the Ctrl key and clicking on the *y.calc[i]* node, and then the *tau* node. If you get the sequence confused, with the Ctrl key pressed, re-click on the offending node and the arrow will disappear, along with the associated arrant definition.

All that remains is to tell WinBUGS that this variable must be incremented. To create the required plate, position the mouse when you want the upper left corner to be, depress the Ctrl key, and right-click the mouse. (Remember, if you create something you didn't intend, make sure it is highlighted, hold the Ctrl key, and press the delete key. To highlight a plate, click on its lower right corner.) Enter the index, *i*, with no brackets, and its starting and ending values, from *1*, to *Npts*. *Npts* will be read-in during compilation.

Your directed graph is completed. If you're sure there are no errors, click Doodle, Write Code, to create WinBUGS code representing the Doodle. Although it's easy to write the code directly, Spiegelhalter and colleagues (2000) recommend using the Doodle pad because a correct Doodle will produce correct code. All of us are familiar with code that compiles but doesn't produce the intended results.

Helpful Hint: When building up a directed graph from an existing model by replacing constants with hyperparameters, don't forget to remove the previously defined constants from the input list for your old model in providing an input list for your new one. WinBUGS will use unintended input values to override DAG nodes having the same name.

Executing the BUGS Code: Bayesian Regression Step-by-Step

After the Doodle is complete, the next step is to generate the code.

Click Doodle, then Write Code. A new sheet containing the code defined by the directed acyclic graph will appear. If there are logical inconsistencies, variable misspellings, or other errors, a notice to that effect will appear in the status window at the bottom of the page on the left. Correct the error and Write Code again, until syntactically correct code is generated.

Next, click on Model, then Specification, to open the specification tool. Using the mouse, highlight the word "model" at the top of the code sheet, and click check model. If all is well, a notice to that effect will appear in the status window at the bottom of the page. The model will require input for defining constants or providing data. The easiest way to do this is with a list, using "**S/R**"[7] syntax, e.g. :

```
list(
Npts=42,
y.obs=c(3040, 2470, 3610, 3480, 3810, 2330, 1800, 3110, 3160, 2310,
4360, 1880, 3670, 1740, 2250, 2650, ...
```

(See Appendix A for a complete example.)

Notice that numbers defining a vector input must be concatenated using the "c( )" syntax.

Highlight the word "list" and click Load Data, and check the status window to see that the data loaded properly. If an unexpected variable name or a missing variable is encountered, it will be noted in the status window.

[7] **S** is a Statistics computing language developed by Bell Labs. The commercial rights are currently held by Insightful Corp and marketed under the name of **S-Plus**®. S-Plus is widely used in statistical research because its algorithms can be modified by the user, unlike most commercial products. But S-Plus is expensive, and there is available a “freeware” version that is maintained by the academic community. While **R** does not have many of the refinements of S-Plus, it does have all the serious statistical computing capabilities. Instructions for downloading the R software package are provide in Appendix X.

Click on Compile. Check the status window for confirmation.

WinBUGS needs starting values to begin its iterations. These can be supplied using another list, or in many cases WinBUGS can provide guesses. To generate initial values click Gen Inits, and check the status window. (It's a good idea to supply your own initial values, if you know them, to avoid a perhaps silly choice from a very broad prior. It isn't all or nothing: you can supply some initial values, then Gen Inits to generate those remaining.) Here we'll supply the initial values.

```
Inits
list(alpha= 3000., beta=185., tau=1.1111E-5)
```

The model is now defined, compiled, initiated, and ready to run.

Close the Specification Tool, and click on Inference, Samples. Choose which percentiles you wish to monitor. 2.5 and 97.5 are the defaults. Choose which node you wish to monitor by entering the node name in the window, and clicking set. Enter as many nodes as you wish. Here we will enter in turn, *alpha*, *beta*, *tau* and *sig*. To see real time updating of the nodes during simulation, enter the node name, or enter an asterisk * to see all the nodes you've entered, and click Trace.

The model is now ready to generate samples. Before leaving the Sample Monitor Tool, enter 10,001 as the beginning observation to use. Markov Chain Monte Carlo chains need to run for a while so that the influence of the starting position is lost. Leave the Sample Monitor Tool open and click Model, Update, to see the Update Tool. Choose a sample size, say 30,000, which means we will record a sample of 20,000. This may be excessive, but a good place to begin. Arrange the windows so that the Dynamic trace is visible and unencumbered.

Click Update to begin simulation. Current sampled values from the nodes selected will be displayed dynamically.

When the simulation is complete, select the Sample Monitor Tool window, and click History to see all values plotted. Click Density to see the posterior probability densities (figure 1). Click Stats to see the summary statistics including the mean and standard deviation, and the selected quantiles, table 2

Figure 1 WinBUGS Plots of Posterior Densities

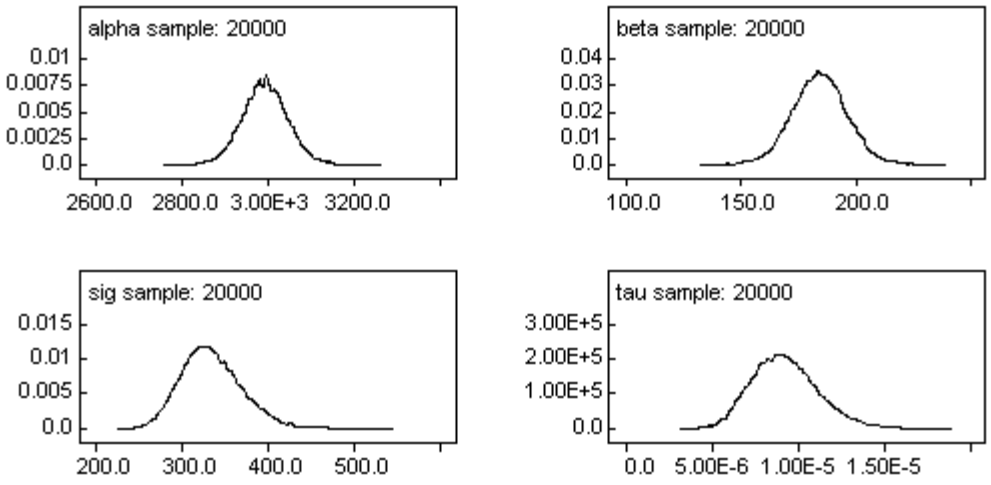


Table 2 WinBUGS Statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	2992.0	51.64	0.3663	2891.0	2992.0	3094.0	10001	20000
beta	184.4	11.65	0.08135	161.3	184.4	207.4	10001	20000
sig	334.1	35.56	0.2816	272.5	331.0	410.4	10001	20000
tau	9.261E-6	1.929E-6	1.533E-8	5.938E-6	9.129E-6	1.346E-5	10001	20000

The Bayesian regression is complete, and we are ready to compute the required marginal likelihoods that are precursors to Bayes factors.

How to Build the Post-convergence Gibbs Sampler:

The Bayesian regression has provided converged samples from the posterior densities for the model regression parameters (Table 2).

We now have the mean and standard deviation for the model parameters alpha (intercept) and beta (slope) as well as the standard deviation, and the precision. We need to sample form the post convergence density of the precision (tau, above) to compute it's posterior mean. To do that we need the shape and scale parameters for the gamma density; what we have however are the mean and standard deviation.

The mean of the gamma density is *shape/scale*, and the variance is *shape/(scale)*^{2} The shape and scale parameters can then be calculted directly from the estimates of the mean and standard deviation (stdev=variance^2) variance in table 2.

```
shape.tau.star <- (tau.star/sd.tau.star)^2
scale.tau.star <-  tau.star/(sd.tau.star^2)
```

We now change the WinBUGS model to fix the values of *alpha* and *beta*, and continue sampling from the posterior density for *tau*. The result is shown here.

Table 3 Post-Convergence Values for Precision and Standard Deviation

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
sig	332.8	25.48	0.1862	287.6	331.2	387.7	10001	20000
tau	9.188E-6	1.387E-6	1.005E-8	6.652E-6	9.117E-6	1.209E-5	10001	20000

The key to Chib's algorithm is estimating the posterior density for *tau*, not as the density of the posterior mean of *tau*, which would be skewed by infrequent but very large values for *tau*, but rather by the average of the sampled *posterior densities*. In other words the average of the ordinates of each value drawn from the post-convergence Gibbs sample:

$$tau.posterior.denity \approx \frac{1}{G}\sum_{g}^{G} tau.density_{g} \qquad \text{Equation 8}$$

Thus it is not the result shown in table 3 that we need but the 20,000 post-convergence draws from the gamma density for *tau*. In practice these values are easily obtained from the WinBUGS sample using the Coda feature, and saving them for further computation in **R**. The resulting samples are stored as `x.tau` (for model 1) and `z.tau` (for model 2)/

How to Calculate the Marginal Likelihood from the WinBUGS Results:

The posterior marginal density for the data under the given model is, from equation 4:

$$m(\boldsymbol{y}) = \frac{f(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \boldsymbol{y})} \qquad \text{Equation 9}$$

or in computationally more convenient logarithmic form:

$$\log \hat{m}(\boldsymbol{y}) = \log f(\boldsymbol{y} \mid \boldsymbol{\theta}^*) + \log \pi(\boldsymbol{\theta}^*) - \log \hat{\pi}(\boldsymbol{\theta}^* \mid \boldsymbol{y}) \qquad \text{Equation 10}$$

where the caret represents a statistical parameter estimate (since the true value is not available) and the superscript star (*) represents the parameter value at it's posterior mean.

We have all the values needed for $\boldsymbol{\theta}^*$ from table 3 and the saved post-convergence sample for *tau*. All that remains is the calculations.

The first term is the (log of the) *likelihood*. It is the product of the ordinates of a normal density for each observed value of y, centered at the model's estimate, $\hat{y}_i = \alpha^* + \beta^* x_i$ and a standard deviation of σ^*. The log of this product is the sum of the individual logs, so the first term is

$$\log f(\boldsymbol{y} \mid \boldsymbol{\theta}^*) = \sum_{i}^{Npts} \log f(y_i) \qquad \text{Equation 11}$$

and $f(\cdot)$ is the normal density.

```
# likelihood
log.likelihood <- 0.
for(i in 1:length(CC.df$y)){
log.likelihood <- log.likelihood +
log(dnorm(x=CC.df$y[i], mean=(C1.star + C2.star*CC.df$x.m.xbar[i]),
   sd=sig.star))}
```

Programming Aside: Loops, like the one above, are inefficient in **S/R** an interpreted, object-oriented language that treats arrays as a single object.. The entire `log.likelihood` loop above, including its initial zeroing, could be carried out faster using this single line of **R**-code:

```
# likelihood
log.likelihood <- sum(log(dnorm(x=CC.df$y, mean=(C1.star +
   C2.star*CC.df$x.m.xbar), sd=sig.star)))
```

The loop syntax is used here to make the calculation more understandable to those less familiar with **S/R**.

The second term in equation 10 is the prior density, i.e. the assumed "probability" of the model parameters before observing any data. It is the value of the normal density evaluated at the post-convergence values for alpha, beta, and tau, using the prior density parameters. Examination of the **R** code will clarify this.

$$log\,\pi(\boldsymbol{\theta}^{*}) = log\,\pi(\alpha^{*}) + log\,\pi(\beta^{*}) + log(\tau^{*}) \qquad \text{Equation 12}$$

where $\pi(\cdot)$ is normal.

```
# prior
log.prior.C1 <- log(dnorm(C1.star, mean=mu.C1.prior, sd=sd.C1.prior))
log.prior.C2 <- log(dnorm(C2.star, mean=mu.C2.prior, sd=sd.C2.prior))
log.prior.tau <- log(dgamma(tau.star, shape=shape.tau.prior,
rate=scale.tau.prior))
log.prior <- log.prior.C1 + log.prior.C2 + log.prior.tau
```

It is important to note that the parameters C1, and C2 are assumed to be independent, since the data was centered ($x_{centered} = x - mean(x)$). This assumption is *not* always valid and must be verified. (For the example in Appendix D the correlation between intercept and slope in model 1 is -0.003438777, and 0.0004354387 for model 2, thus not meaningfully different from zero). If the correlation is not negligible, then the prior *joint* density for C1 and C2 must be used. For example:

```
# prior (correlated C1 and C2)
log.prior.tau <- log(dgamma(tau.star, shape=shape.tau.prior,
rate=scale.tau.prior))
log.prior.C1.C2 <- log(dmvnorm(x=c(C1.star, C2.star), mean=c(mu.C1.prior,
mu.C2.prior), sd=c(sd.C1.prior, sd.C2.prior), rho=rho.C1.C2.prior) )
log.prior <- log.prior.C1.C2 + log.prior.tau
```

Of course this result is numerically identical to the case of independent parameters when $\rho_{C1,C2} = 0$. In both cases the justification for normal behavior of the model parameters is the Central Limit Theorem (see Appendix E).

The final term in the marginal density is the posterior density of the parameter estimates.

$$log\,\hat{\pi}(\boldsymbol{\theta}^{*} \mid \boldsymbol{y}) = log\,\pi(\alpha^{*} \mid \boldsymbol{y}) + log\,\pi(\beta^{*} \mid \boldsymbol{y}) + log(\tau^{*} \mid \boldsymbol{y}) \qquad \text{Equation 13}$$

where the first two terms are straight forward: They are the ordinates of the normal densities of the converged parameter estimates evaluated at their respective posterior means. The estimate for the posterior of *tau* is determined from equation 8, above.

```
# posterior
log.posterior.C1 <- log(dnorm(C1.star, mean=C1.star, sd=sd.C1.star))
log.posterior.C2 <- log(dnorm(C2.star, mean=C2.star, sd=sd.C2.star))
log.posterior <- log.posterior.C1 + log.posterior.C2 + log(density.tau.star)
```

Finally we can calculate the marginal density of the data for this model.

```
log.marginal.density.x <- log.likelihood + log.prior - log.posterior
```

This process is repeated for the second model. Create and execute a Bayesian regression model; Generate 20,000 (say) post-convergence values for the precision, tau; and used these results to calculate the model's likelihood, prior and posterior densities.

```
log.marginal.density.z <- log.likelihood + log.prior - log.posterior
```

The Bayes factor is then calculated from the ratio of these marginal densities. Since we have worked with logarithms, the Bayes factor requies an exponentiatino. The Bayes Factor in favor of Model 2 over Model 1 isthe inverse of the Bayes Factor favoring Model 1 over Model 2:

```
1./exp(log.marginal.density.x - log.marginal.density.z)
```

In the referee example the Bayes factor is "about 4862" (Han and Carlin, 2001). Since the result depends on the random behavior of the Gibbs sampler, some variation will be observed. Han and Carlin also report a range of (4835.1 – 4940.7) using different computational methods.

Since the Bayes Factor is the odds favoring one model over another if their prior odds are equal, then a number as large a 4000 is convincing evidence indeed of Model 2's superiority in describing the data. It is often convenient to compare twice the *log* of the Bayes Factor rather than the Bayes Factor itself. In this example that quantity is about 17.

The WinBUGS models, and the **R** code for computing Bayes Factors is presented in Appendices A, B, C, and D.

Section

3

Random Fatigue Limit Data and Model

The Ti-6Al-4V data are presented in the following table.

Table 4 RFL data

	spec	cycles	SWT	C			spec	cycles	SWT	C			spec	cycles	SWT	C
1	11	4,802	114.35	1		35	71	152,993	65.00	1		69	27	1,834,570	55.00	1
2	6	5,210	115.55	1		36	44	153,000	63.73	1		70	133	2,400,000	60.00	1
3	1	5,319	115.02	1		37	45	153,918	63.73	1		71	79	2,684,170	52.50	1
4	7	5,805	107.09	1		38	102	160,971	60.00	1		72	62	2,840,000	57.02	1
5	2	6,737	108.07	1		39	24	183,276	62.50	1		73	63	2,844,620	57.02	1
6	8	9,290	96.62	1		40	46	192,463	60.37	1		74	28	3,247,816	60.00	1
7	3	9,700	102.01	1		41	25	205,774	60.00	1		75	80	3,569,869	51.50	1
8	12	13,790	97.31	1		42	47	209,277	60.37	1		76	64	4,526,292	55.90	1
9	9	16,046	93.38	1		43	48	227,000	59.03	1		77	65	4,760,000	59.03	1
10	4	20,199	84.29	1		44	49	257,988	58.36	1		78	81	5,281,133	55.00	1
11	13	26,336	86.14	1		45	50	290,896	59.03	1		79	82	5,563,469	50.00	1
12	14	34,064	79.51	1		46	51	297,754	59.03	1		80	83	6,086,142	50.00	1
13	19	37,767	77.50	1		47	52	344,000	60.37	1		81	29	6,553,514	60.00	1
14	10	38,841	79.69	1		48	103	363,445	57.00	1		82	66	6,793,930	53.67	1
15	32	42,300	73.79	1		49	104	406,873	61.00	1		83	107	7,049,526	61.00	1
16	5	48,537	78.27	1		50	72	408,178	60.00	1		84	67	7,268,673	57.02	1
17	101	49,629	61.00	1		51	26	474,975	55.00	1		85	108	8,746,159	60.00	1
18	15	50,265	75.12	1		52	53	491,430	63.73	1		86	137	29,714,022	57.02	1
19	33	57,273	73.79	1		53	105	591,316	61.00	1		87	109	44,904,195	57.00	1
20	34	67,900	57.02	1		54	54	633,000	60.37	1		88	84	10,000,000	47.50	2
21	20	68,227	72.50	1		55	55	633,168	60.37	1		89	85	10,000,000	48.00	2
22	35	69,800	67.08	1		56	56	889,000	57.02	1		90	68	10,000,000	50.31	2
23	36	69,866	67.08	1		57	57	953,156	58.14	1		91	69	10,000,000	55.16	2
24	37	73,000	67.08	1		58	73	958,757	60.00	1		92	30	10,000,000	58.00	2
25	38	85,023	69.09	1		59	74	1,015,716	55.00	1		93	31	10,000,000	60.00	2
26	21	88,303	67.50	1		60	106	1,098,728	58.00	1		94	112	100,000,000	57.00	2
27	39	91,557	65.74	1		61	75	1,340,436	57.00	1		95	111	100,000,000	59.00	2
28	40	93,200	63.73	1		62	58	1,370,000	53.67	1		96	115	100,000,000	59.00	2
29	41	98,046	63.73	1		63	76	1,453,661	53.00	1		97	110	100,000,000	60.00	2
30	22	103,346	70.00	1		64	59	1,493,080	57.02	1		98	116	100,000,000	60.00	2
31	42	109,880	67.08	1		65	60	1,646,300	57.02	1		99	113	100,000,000	60.50	2
32	23	112,506	65.00	1		66	61	1,650,000	60.37	1		100	114	100,000,000	60.50	2
33	70	119,054	65.00	1		67	77	1,687,437	55.00	1		101	134	1,000,000,000	50.00	2

Comments on Transforming the Data

Because regression parameters are correlated[8].cycles were counted in units of ten-million to mitigate numerical difficulties. Since using natural logarithms (rather than base 10 logs) produces a proportionality constant of unity between the CDF and the PDF, the WinBUGS model uses natural logs of the transformed cycles, i.e.: $\texttt{lnNobs} = ln(N / 10^{7})$. The original units are in table 4.

[8] *All* regression model parameters are correlated (cf.: Fisher, 1925). Under some circumstances, for example when the data are centered at $\overline{X}, \overline{Y}$ some of the model covariances are zero. (See also Annis, 2002, for an example related to crack propagation and strucural llife prediction.)

Figure 2 Random Fatigue Limit Model on Semi-log Axes Showing the Probability Densities for the Error-in-Cycles and for the RFL.

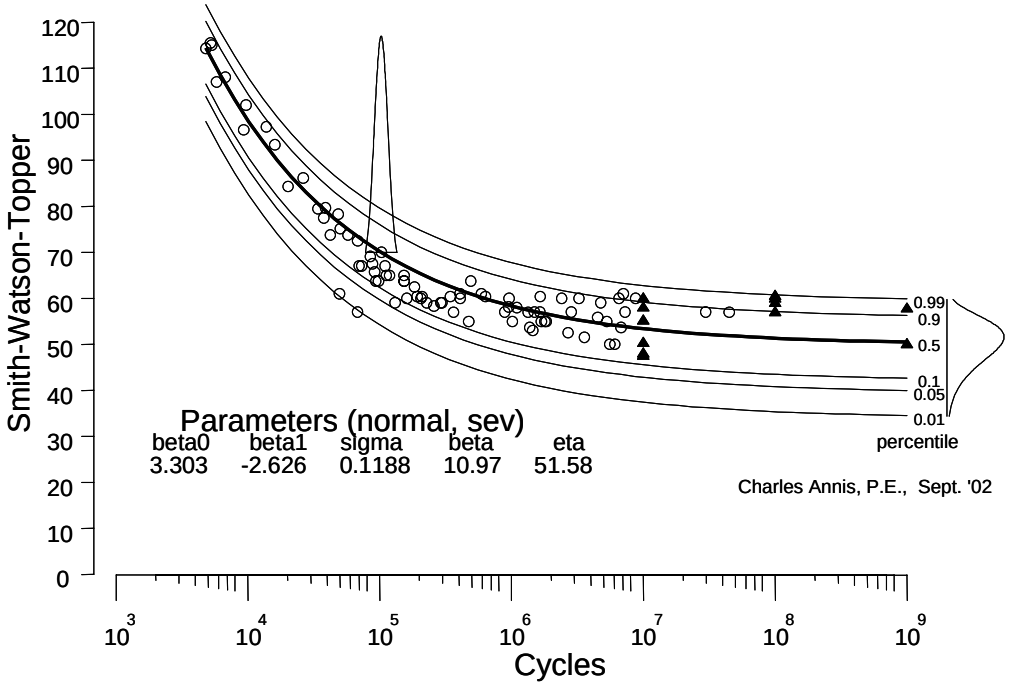


Figure 2 also shows, the probability density for the random fatigue limits (on the right) and the density of errors-in-cycles. Note that a large fraction of the observed variability in cycles is explained by the variability in fatigue limits.

The Random Fatigue Limit (RFL) Model.

Appendix F describes the workings of ordinary least squares (OLS), which is adequate for describing low cycle fatigue data without runouts. OLS, however, can not deal with censored observations, and was superseded by the method of maximum likelihood to account for runouts correctly. However, with the acquisition of very long-life data, approaching 10^9 cycles, standard mathematical descriptions of HCF behavior in the lower right corner of the s-N curve were seen to be inadequate, even using censored data techniques. The scatter was clearly not constant over the entire s-N curve, violating one of the assumptions of the current mathematical models. Modifying conventional s-N models to include a fixed fatigue limit did not have the hoped-for result: These models produced lower bound curves more closely bunched near runout, which is opposite to what the data themselves say.

Actually, there are at least three practical problems with the traditional, large numbers of s-N tests, approach to estimating fatigue limits:

1) At the stress levels of interest, s-N curves are relatively flat. To get failures below the median fatigue limit requires tests that are potentially orders of magnitude longer than the desired cyclic life. The primary interest, of course, is in a stress level at which one percent or fewer of the structures would not survive, say, 10^7 cycles. While

extrapolation of a median *s-N* curve to the long life of interest *may* be acceptable, estimation of the first percentile of the fatigue limit distribution most likely is not.

2) The variability of fatigue lives increases greatly as test stresses decrease and a general model for this changing scatter has not been accepted. Thus, extrapolation of the percentiles of the fatigue strength to longer lives also requires somehow connecting the standard deviation of fatigue lives with stress levels.

3) Current HCF problems indicate the necessity to define fatigue limits at lives longer than 10^7. The economic burden of testing runout lives to 10^8,or longer, only compounds the first two problems.

Because of the enormous scatter in lives at the stress levels of prime interest to HCF, and the expense of testing to very long cycle counts, a better method for estimating and validating fatigue limits is needed.

An innovative approach to describing fatigue limit behavior was proposed recently by Pascual and Meeker (1999). They postulated a random fatigue limit (RFL) model in which each specimen has its own fatigue limit in much the same way that each specimen has its own fatigue lifetime if tested at a sufficiently high stress. This random fatigue limit is explicitly included in the *s-N* model. Maximum likelihood methods (described later) are then used to estimate the parameters of the *s-N* equation as well as the parameters of the fatigue limit distribution. The percentiles of the fatigue limit distribution are easily calculated from the estimated parameters. The random fatigue limit model produces the proper shape of the median *s-N* function and the type of scatter typically seen in fatigue tests at HCF stress levels, as is illustrated in figure 2, for Ti-6Al-4V. Notice the behavior of the RFL, shown at the right. The distribution is skewed downward and so the median (50%) necessarily is below the mode (maximum ordinate value).

Mathematical Formulation of the RFL Model:

Earlier attempts at modeling the stress-life *(s-N)* behavior of cyclic fatigue in the long life regime used a linear equation relating log(cycles) and log(stress), modified with a constant runout stress, or fatigue limit:

$$\log N_i = \beta_0 + \beta_1 \log (S_i - \beta_2) + \varepsilon_i \qquad \text{Equation 14}$$

where, for specimen *i*, N_i represents cycles to failure, S_i is the applied stress parameter, and β_2 is a *constant* fatigue limit $(S_i > \beta_2)$, and ε_i is a random variable representing the scatter in cycles to failure about the predicted life. Typically, the life random variable, ε, would be represented by a lognormal distribution with zero mean. For this assumption, ε_i is the difference between the log life of specimen *i* and the log median life at the test stress S_i. The parameters of the median life prediction, β_0, β_1, and β_2 are estimated from test data and β_2 is interpreted as the fatigue limit stress condition. Since β_2 is an asymptote, the *s-N* curve flattens as S approaches the fatigue limit. This model is only marginally adequate for the median behavior in the long life regime but it is not consistent with the commonly observed increase in the standard deviation of lives as S approaches the constant fatigue limit (see figure 2). But the main shortcoming of a constant fatigue limit is that it doesn't work. Since it is a single-valued constant, the fatigue limit, β_2, must be less than the lowest stress tested (so that the logarithm of $(S_i - \beta_2)$ is defined) whether the specimen failed at that stress or not. This causes the β_2 asymptote to be so low as to

produce an unrealistic material model that had to be continually revised downward to accommodate newer, low stress data.

The random fatigue limit model is a generalization of equation 14 in which the fatigue limit term is modeled as a random variable that can be considered to result from inherent, but unknown, quality characteristics of each specimen in the population. Thus the value of the fatigue limit is *not* a single constant, but rather an individual characteristic of each specimen (or component). The RFL model for test specimen *i* is given by:

$$\log N_i = \beta_0 + \beta_1 \log (S_i - \gamma_i) + \varepsilon_i \qquad \text{Equation 15}$$

where γ_i is the random fatigue limit for specimen i $(S_i > \gamma_i)$ and is expressed in units of the stress parameter. In this model, ε is the random life variable associated with scatter from specimens that have the same fatigue limit.

The RFL model produces probabilistic s-N curves that have the characteristics commonly seen in HCF data. This is illustrated in figure 2 which presents the 1st, 5th, 10th, 50th, 90th 95th and 99th percentile s-N curves as would be determined from the distribution of fatigue limits. The percentile s-N curves display the commonly observed shape in the HCF regime. Further, it is easily seen in figure 2 that a difference in test lives from two specimens with slightly different fatigue limits could be quite large. The increased scatter in fatigue lives is explained by different specimens having different fatigue limits and this is true regardless of the scatter in life at higher stresses. Thus, the RFL model accommodates not only the flattening of the s-N curve but also the increased scatter that is typical of HCF lives. Experience to date indicates that the fatigue limit scatter dominates in the HCF regime when S is close to γ_i while the scatter in life is more significant when S is large compared to γ_i.

There are two random variables in the RFL model for which probability distributions are needed. Experience again suggests the conventional lognormal distribution is appropriate for ε_i, the scatter in cyclic lives. Thus, the conditional distribution of cycles to failure given γ will be a lognormal distribution with mean equal to $\beta_0 + \beta_1 \log (S - \gamma)$ and standard deviation equal to σ_ε. Then ε is lognormal $(0,\sigma_\varepsilon)$. The Weibull distribution does well describing the skewed downward behavior of the random fatigue limit, γ_i. The Weibull parameters, $\eta, \beta,$ represent the 63.2th percentile runout stress, and shape parameter, respectively. Thus η has the same units as the stress metric. Specimens will have inherent, but unknown, quality differences that result in distinct fatigue resistance in the long life regime. The upper limit of runout stress is observed to be more restrictive than the lower limit. That is, while a very low quality specimen is sometimes observed, albeit infrequently, extremely high runout stresses are never observed. (And it is these infrequent lower performers that are at the crux of the HCF problem.) The RFL model provides a means to measure the propensity for this life-limiting behavior.

Figure 2 illustrates another important lesson: Computing the covariance between life and stress parameter would be woefully inadequate to describe HCF behavior. This is because the covariance only accounts for the uniform, linear, relationship between variables, and in HCF as with many other engineering situations, relationships are not linear, nor do they exhibit constant data scatter.

Deficiencies in the RFL Model for the Ti-6Al-4V Data

While the Random Fatigue Limit model is the first major improvement in modeling fatigue behavior in decades, it is not without its shortcomings. These are undeniable when the model is used to describe the 102 point Ti-6Al-4V dataset. These shortcomings are likely to be resolved, but at present they present a distraction to more widespread use of the model.

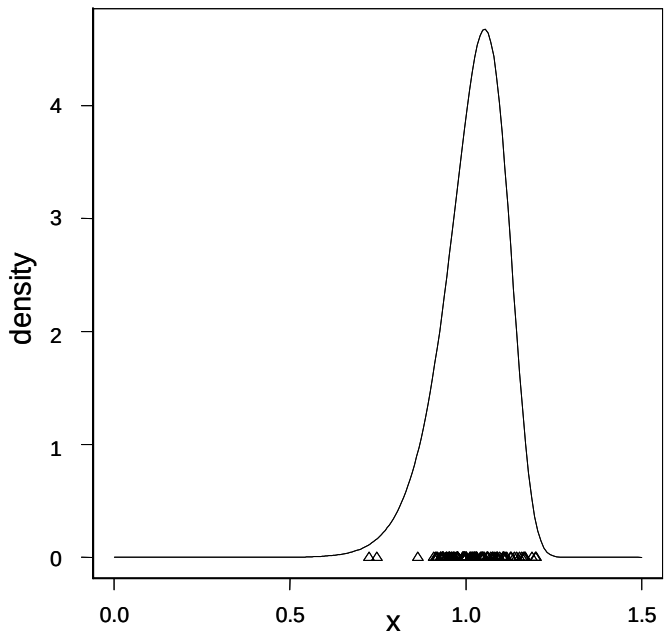
There are two serious difficulties with the model in its present form:

1) Both the Maximum Likelihood, and Bayesian Maximum Posterior Density algorithms for estimating model parameters cannot allocate total model deviation into reasonable fractions of errors-in-cycles and random fatigue limits, and tend toward a vanishingly small standard deviation of r the errors-in-cycles density.

2) The formulation of the probability density for the individual, imputed, random fatigue limits does not have the proper shape, and correcting this is not as simple as it might appear.

Figure 3 shows the probability density for the individual RFL values fit by the WinBUGS model.

Figure 3 Weibull Density used for the Random Fatigue Limit



These difficulties are discussed briefly here.

Simulation studies (not reported here) where the contributions of both the error-in-cycles density and the RFL density are known, show that while both MLE and MPD methods will produce a realistic model, both methods tend to allocate almost all the error to the RFL density. In fact, Harry Martz, (Johnson, Valen E., Mark Fitzgerald, Harry F. Martz, 1999) one of the reviewers of Pascual and Meeker's original paper (Pascual and Meeker, 1999)

observed this tendency. Furthermore, it is well known in the statistics community that maximum likelihood parameter estimation of mixtures of distributions is problematical. The classic example being the observed data from a mixture of two densities, $N_1(\mu_1 , \sigma_1^2)$ and $N_2(\mu_2 , \sigma_2^2)$ with mixing fraction f. The MLE assigns all the probability to one observation with zero variance. Thus the ordinate of the resulting density at that point is infinite, overwhelming any other allocation of probability. The RFL model shares this tendency.

The Weibull density is parameterized in WinBUGS as equation 16-a, below

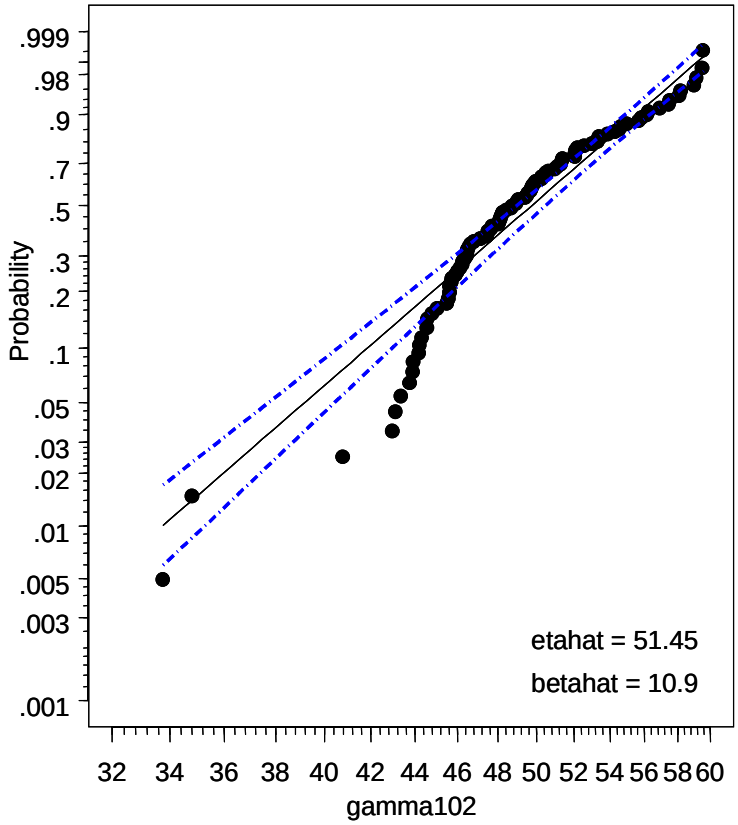
$$f(x) = r\lambda x^{r-1} e^{-\lambda x^r} \qquad \text{Equation 16-a}$$

This parameterization is different from the more common one, in equation 16-b, familiar to engineers. While the engineering parameterization is in terms of the cumulative distribution function (cdf) and WinBUGS uses the probability density function (pdf), the largest difference is that the scaling by η in the engineering parameterization occurs *before* the exponentiation by β.

$$F(x) = 1 - e^{-(x/\eta)^{\beta}} \qquad \text{Equation 16-b}$$

Figure 3 shows that at least qualitatively the Weibull density is adequate for describing the behavior of the Random Fatigue Limit Model for these 102 observations. Actually, however, a closer look shows that the fit is not a good as one might hope.

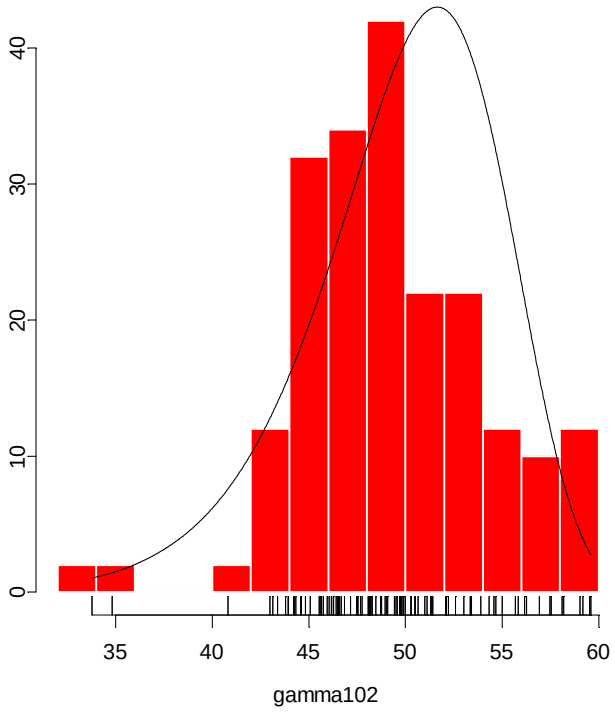
Figure 4 CDF plot of RFL showing poor quality of Weibull model.
Censored observations are shown at the top.



Note: The ML (maximum likelihood) parameter estimates printed on figure 4, (β =51.48, , η = 10.9) differ slightly the WinBUGS estimates (β =51.58, , η = 10.97). As is evident in figure 4, neither provides a good description of the imputed random fatigue limits because the density shape (tail to the left) is contrary to the data (tail to the right). This is the unfortunate consequence of the Weibull not being a location-scale density, so the shape necessary to achieve an accurate variance results in a negative skew.

The histogram of these 102 values is presented in figure 5.

Figure 5 Histogram for the 102 Imputed Random Fatigue Limits Showing Poor Fit of the Weibull Density



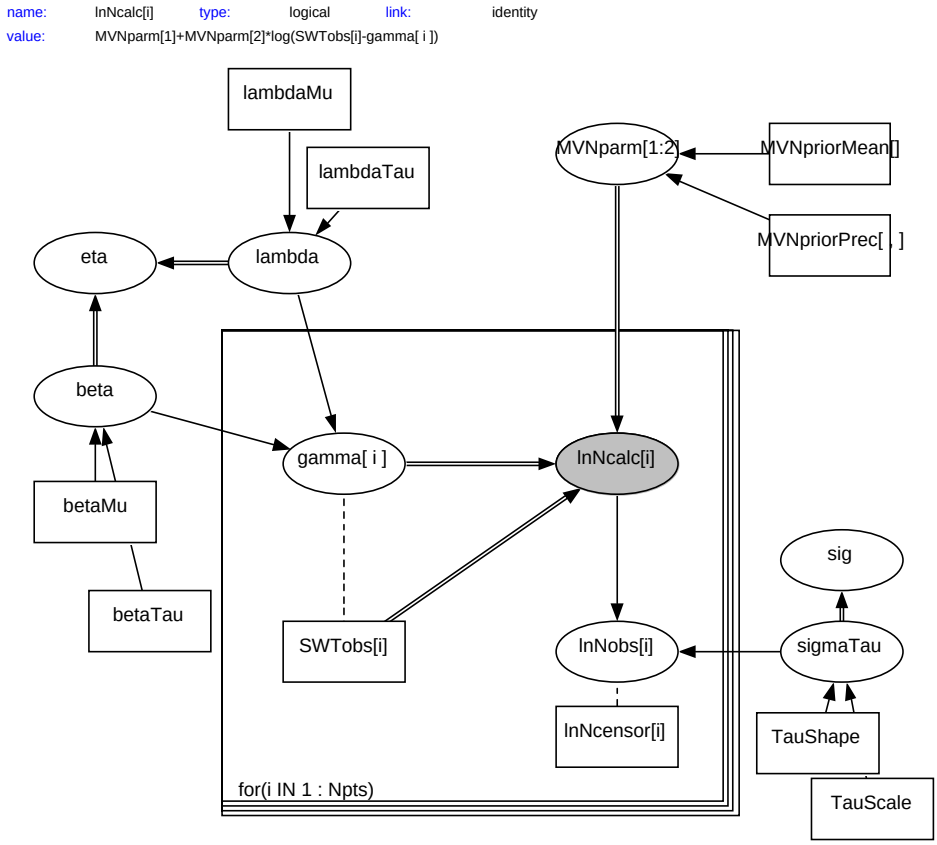
Result: Since Bayes factors rely on the quality of the statistical descriptions of the data, and since the RFL model *for this dataset* has poor statistical qualities, further use of Bayes factors, without correcting the underlying deficiencies in the RFL model itself, would be inappropriate.

Valuable Statistics Lesson: *The quality of your result is only as good as the reality of your statistical assumptions.* In this case the distribution of the imputed individual Random Fatigue Limits is not Weibull. (Further study suggests that it does not appear to follow any conventional statistical density either, although transformation may prove useful.)

WinBUGS Code for Estimating RFL Model Parameters

The WinBUGS graphical implementation for estimating values for the Random Fatigue Limit model. is presented in figure 6

Figure 6 WinBUGS Directed Acyclic Graph (DAD) for RFL Parameter Estimation



The corresponding WinBUGS code is presented here.

```
model;
{
   beta ~ dnorm(betaMu,betaTau)
   lambda ~ dlnorm(lambdaMu,lambdaTau)
   for( i in 1 : Npts ) {
      lnNobs[i] ~ dnorm(lnNcalc[i],sigmaTau)I(lnNcensor[i],)
   }
   for( i in 1 : Npts ) {
      lnNcalc[i] <- MVNparm[1] + MVNparm[2] * log(SWTobs[i] - gamma[i])
   }
   eta <- pow(lambda,( - 1.0) / beta)
   for( i in 1 : Npts ) {
      gamma[i] ~ dweib(beta,lambda)I(,SWTobs[i])
```

```
   }
   sigmaTau ~ dgamma(TauShape,TauScale)
   sig <-  1.0 / sqrt(sigmaTau)
   MVNparm[1:2] ~ dmnorm(MVNpriorMean[],MVNpriorPrec[ , ])
}
```

The resulting parameter estimates is shown in Table 5

Table 5 WinBUGS RFL Parameter Estimates

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
MVNparm[1]	3.303	0.5188	0.04199	2.495	3.214	4.458	10001	20000
MVNparm[2]	-2.626	0.1346	0.01071	-2.921	-2.606	-2.413	10001	20000
beta	10.97	0.454	0.03603	10.07	11.03	11.63	10001	20000
eta	51.58	0.6873	0.03848	50.16	51.61	52.88	10001	20000
sig	0.1188	0.03828	0.001764	0.06317	0.1123	0.2112	10001	20000

For completeness the 102 imputed values for the random fatigue limits are presented below.

```
gamma102<-c(49.77, 52.08, 52.11, 47.47, 51, 46.84, 52.09, 53.41, 52.22,
47.15, 52.13, 48.95, 48.15, 50.51, 45.75, 51.41, 34.82, 48.76, 48.71,
33.77, 49.02, 43.93, 43.95, 44.33, 47.53, 46.26, 44.82, 42.99, 43.38,
49.91, 47.52, 45.65, 46.05, 40.81, 47.75, 46.49, 46.52, 43.13, 46.4,
44.59, 44.6, 45.07, 44.2, 44.24, 45.53, 45.65, 47.68, 44.59, 49.08, 48.1,
43.79, 52.6, 50.65, 50.28, 50.29, 48.17, 49.51, 51.38, 46.59, 49.82,
49.42, 46.16, 45.66, 49.74, 50, 53.34, 48.06, 48.22, 48.28, 53.9, 46.68,
51.31, 51.31, 54.57, 46.28, 51.12, 54.33, 50.49, 45.59, 45.74, 55.84,
49.58, 56.94, 53.02, 56.26, 54.68, 55, 45.94, 46.5, 48.47, 53.37, 55.69,
57.54, 56.17, 58.19, 58.1, 59.18, 59.03, 59.53, 59.59, 49.69, 57.48,
49.77, 52.08, 52.11, 47.47, 51, 46.84, 52.09, 53.41, 52.22, 47.15, 52.13,
48.95, 48.15, 50.51, 45.75, 51.41, 34.82, 48.76, 48.71, 33.77, 49.02,
43.93, 43.95, 44.33, 47.53, 46.26, 44.82, 42.99, 43.38, 49.91, 47.52,
45.65, 46.05, 40.81, 47.75, 46.49, 46.52, 43.13, 46.4, 44.59, 44.6,
45.07, 44.2, 44.24, 45.53, 45.65, 47.68, 44.59, 49.08, 48.1, 43.79, 52.6,
50.65, 50.28, 50.29, 48.17, 49.51, 51.38, 46.59, 49.82, 49.42, 46.16,
45.66, 49.74, 50, 53.34, 48.06, 48.22, 48.28, 53.9, 46.68, 51.31, 51.31,
54.57, 46.28, 51.12, 54.33, 50.49, 45.59, 45.74, 55.84, 49.58, 56.94,
53.02, 56.26, 54.68, 55, 45.94, 46.5, 48.47, 53.37, 55.69, 57.54, 56.17,
58.19, 58.1, 59.18, 59.03, 59.53, 59.59, 49.69, 57.48)
```

Appendix

A

WinBUGS: Software for Bayesian Regression

BUGS is almost an acronym for **B**ayesian inference **U**sing **G**ibbs **S**ampling, and is software for the Bayesian analysis of complex statistical models using **Markov Chain Monte Carlo** (**MCMC**) methods, especially the **Gibbs sampler**. With the inclusion of a GUI for developing directed acyclic graphs, **DAG**s, it has become the method of choice for many Bayesian practitioners and researchers (c.f.: Congdon, 2001).

How to Download the WinBUGS Software Package

BUGS was originally a statistical research project at the Medical Research Council Biostatistics Unit, Cambridge, UK, and it is now developed jointly with the Imperial College School of Medicine at St Mary's, London. The Windows version, **WinBUGS**, with a GUI, and ability to create graphs that produce code, can be downloaded from the BUGS website: **http://www.mrc-bsu.cam.ac.uk/bugs/**

There is no fee for the use of the demonstration (Internet) version of the WinBUGS Package which has some modest size restrictions; however users are required to register and to pay a fee to use the full unrestricted version. The current fee (September, 2002) is zero dollars ($0.00). The current version is 1.3.

The program is easy to download and install and includes instructions for obtaining the full-version licence (*sic*). The entire process is straightforward and even the licence (*sic*) key appears to be handled automatically by e-mail.

Appendix

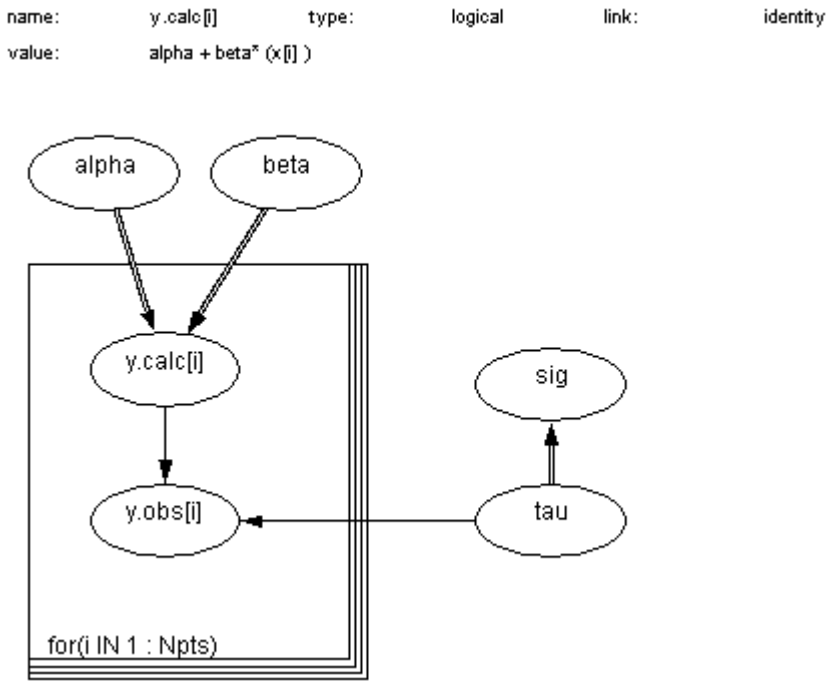
B

WinBUGS Graph and Code

Directed Acyclic Graph

The Directed Acyclic Graph, DAG, for Model 1 is provided in figure 7...It differs from model 2 only in that Model 2 uses z, rather than x as the independent variable. Step-by-step instructions for building a DAG, generating, and then executing the WinBUGS code, are provide in the text, Section 2.

Figure 7 Directed Acyclic Graph for Model 1



The BUGS code

```
model;
{
   for( i in 1 : Npts ) {
      y.calc[i] <- alpha + beta * x[i]
   }
   for( i in 1 : Npts ) {
      y.obs[i] ~ dnorm(y.calc[i],tau)
   }
```

```
   tau ~ dgamma( 3.0,1.8E+5)
   sig <-  1.0 / sqrt(tau)
   alpha ~ dnorm(3.0E+3,1.0E-6)
   beta ~ dnorm(185.0,1.0E-4)
}
```

Data

```
list(
Npts=42,
y.obs=c(3040, 2470, 3610, 3480, 3810, 2330, 1800, 3110, 3160, 2310, 4360,
1880, 3670, 1740, 2250, 2650, 4970, 2620, 2900, 1670, 2540, 3840, 3800,
4600, 1900, 2530, 2920, 4990, 1670, 3310, 3450, 3600, 2850, 1590, 3770,
3850, 2480, 3570, 2620, 1890, 3030, 3030) ,
x=c(1.34047619047618130, -3.15952380952381870, 4.44047619047617910,
3.44047619047618270, 3.64047619047618200, -3.35952380952381800, -
7.95952380952381940, -0.55952380952381731, -0.75952380952381660, -
3.85952380952381800, 5.94047619047617910, -6.35952380952381800,
4.34047619047618480, -5.35952380952381800, -0.35952380952381802, -
2.25952380952381660, 6.64047619047618200, -1.65952380952381870, -
1.15952380952381870, -6.75952380952381660, -3.75952380952381660,
2.84047619047618130, 4.84047619047618480, 4.74047619047618340, -
5.75952380952381660, -2.55952380952381730, 2.94047619047618270,
11.04047619047618100, -5.75952380952381660, 1.34047619047618130,
2.24047619047618340, 3.54047619047618060, -1.15952380952381870, -
5.75952380952381660, 2.44047619047618270, 4.14047619047618200, -
4.65952380952381870, 2.44047619047618270, 2.04047619047618060, -
7.05952380952381730, 5.34047619047618480, 0.34047619047618127))
```

Inits

```
list(alpha= 3000., beta=185., tau=1.1111E-5)
```

Note: The values for *x* above, and *z*, below, have been centered on their respective means to minimize the correlation[9] between the regression parameters *alpha* and *beta*.

```
z=c(-1.38809523809524290, -4.58809523809524220, 5.41190476190476130,
4.21190476190475850, 4.11190476190475710, -2.88809523809524290, -
7.58809523809524220, 0.41190476190475778, -0.48809523809524080, -
2.88809523809524290, 6.41190476190476130, -5.78809523809524150,
2.21190476190475850, -4.78809523809524150, -2.98809523809524080, -
1.48809523809524080, 7.41190476190476130, -1.08809523809524220, -
0.38809523809524293, -6.78809523809524150, -2.88809523809524290,
3.91190476190475780, 5.81190476190475990, 5.71190476190475850, -
5.98809523809524080, -3.68809523809524010, 3.01190476190475920,
11.31190476190476000, -5.48809523809524080, 1.71190476190475850,
2.41190476190475780, 4.61190476190475710, -0.88809523809524293, -
5.38809523809524290, 3.01190476190475920, 3.81190476190475990, -
4.18809523809524010, 3.51190476190475920, -2.98809523809524080, -
8.38809523809524290, 2.61190476190475710, 1.41190476190475780))
```

[9] For at least 75 years it has been well known in the applied statistics community that regression model parameters are correlated (cf.: Fisher, 1925), yet that fact is almost universally unknown to us engineers. Under some circumstances, for example when the data are centered at $\overline{x}, \overline{y}$ some of the model covariances are zero.

Appendix

C

The R Project for Statistical Computing

R is "GNU S" - a language and environment for statistical computing and graphics, similar to the award-winning **S** system, which was developed at Bell Laboratories by a group lead by John Chambers. **R** is designed as a true computer language with control-flow constructions for iteration and alternation, and it allows users to add additional functionality by defining new functions. For computationally intensive tasks, C, C++ and FORTRAN code can be linked and called at run time.

How to Download the R Statistics Language

R can be downloaded from the R Project website, **http://www.r-project.org/**

The GNU license has no fee. While WinBUGS is only available for the MS Windows environment (there are non-graphical versions that run on unix), **R** is available for many operating systems, including Windows.

Like WinBUGS, the **R** language is easy to download and install. **R** is also easy to use but does require some familiarity with statistics since it's primarily user interface is the command line.

Appendix

D

R Code for Computing Bayes Factors

The **R** Code

The **R** code for computing the marginal densities and then the Bayes factors is presented here. Instructions for downloading the **R** statistics language are provided in Appendix C.

To use the code, first the data must be entered:

```
CC.df<-read.table(file="C:\\Documents and Settings\\user name\\My Documents\\S-Plus
Projects v6.1\\Project 3\\R export\\CC.df.txt", header = TRUE, sep = " ")

x.tau<-read.table(file="C:\\Documents and Settings\\user name\\My Documents\\S-Plus
Projects v6.1\\Project 3\\R export\\xtau.txt", header = FALSE, sep = "\n")
x.tau<-x.tau[,]

z.tau<-read.table(file="C:\\Documents and Settings\\user name\\My Documents\\S-Plus
Projects v6.1\\Project 3\\R export\\ztau.txt", header = FALSE, sep = "\n")
z.tau<-z.tau[,]
```

Of course the path must be change to point to the datasets on a particular computer, `CC.df.txt` being the *y, x, z* dataset, and `x.tau` and `z.tau` are the post-convergence WinBUGS draws from the precision densities. 20,000 draws were used in this example.

The following code can then be copied directly into **R**, and immediately executed (after having removed the comments, of course).

```
# Model 1 Prior parameter densities
mu.C1.prior <- 3000.
sd.C1.prior <- 1000.
mu.C2.prior <- 185.
sd.C2.prior <- 100.
shape.tau.prior <- 3.
scale.tau.prior <- 1.8E5

# Parameter estimates from original draws.
C1.star <- 2992.0
sd.C1.star <- 51.51
C2.star <- 184.5
sd.C2.star <- 11.62
tau.star <- 9.277E-6
sd.tau.star <- 1.921E-6
sig.star <- 333.7

# Estimate the  density of tau.star from auxiliary draws.
tau.posterior <- 9.197E-6
sd.tau.posterior <- 1.384E-6
shape.tau.posterior <- (tau.posterior/sd.tau.posterior)^2
scale.tau.posterior <-  tau.posterior/(sd.tau.posterior^2)
shape.tau.posterior
scale.tau.posterior
```

```
G <- length(x.tau)
p.tau <- matrix(NA, nrow = G, ncol = 1)
for(g in 1:G) {
       p.tau[g] <- dgamma(x.tau[g], shape=shape.tau.posterior,
rate=scale.tau.posterior)
       }
density.tau.star <- mean(p.tau)
```

Programming Aside: Loops, like the one above, are inefficient in **S/R** an interpreted, object-oriented language that treats arrays as a single object. The entire `density.tau.star` loop above, including its initial zeroing, could be carried out much faster using this single line of **R**-code:

```
density.tau.star <- mean(dgamma(z.tau, shape=shape.tau.posterior,
   rate=scale.tau.posterior))
```

The loop syntax is used here to make the calculation more understandable to those less familiar with **S/R**.

```
# likelihood
log.likelihood <- 0.
for(i in 1:length(CC.df$y)){
log.likelihood <- log.likelihood +
log(dnorm(x=CC.df$y[i], mean=(C1.star + C2.star*CC.df$x.m.xbar[i]), sd=sig.star))
  }
log.likelihood

# prior
log.prior.C1 <- log(dnorm(C1.star, mean=mu.C1.prior, sd=sd.C1.prior))
log.prior.C2 <- log(dnorm(C2.star, mean=mu.C2.prior, sd=sd.C2.prior))
log.prior.tau <- log(dgamma(tau.star, shape=shape.tau.prior, rate=scale.tau.prior))
log.prior <- log.prior.C1 + log.prior.C2 + log.prior.tau
log.prior

# posterior
log.posterior.C1 <- log(dnorm(C1.star, mean=C1.star, sd=sd.C1.star))
log.posterior.C2 <- log(dnorm(C2.star, mean=C2.star, sd=sd.C2.star))
log.posterior <- log.posterior.C1 + log.posterior.C2 + log(density.tau.star)
log.posterior

log.marginal.density.x <- log.likelihood + log.prior - log.posterior
log.marginal.density.x

# Model 2 Prior parameter densities
mu.C1.prior <- 3000.
sd.C1.prior <- 1000.
mu.C2.prior <- 185.
sd.C2.prior <- 100.
shape.tau.prior <- 3.
scale.tau.prior <- 1.8E5

# Parameter estimates from original draws.
C1.star <- 2992.0
sd.C1.star <- 42.93
C2.star <- 183.1
sd.C2.star <- 9.387
tau.star <- 1.341E-5
sd.tau.star <- 2.794E-6
sig.star <- 277.6

# Estimate the density of tau.star from auxiliary draws.
```

```
tau.posterior <- 1.353E-5
sd.tau.posterior <- 2.044E-6
shape.tau.posterior <- (tau.posterior/sd.tau.posterior)^2
scale.tau.posterior <-  tau.posterior/(sd.tau.posterior^2)
shape.tau.posterior
scale.tau.posterior

G <- length(z.tau)
p.tau <- matrix(NA, nrow = G, ncol = 1)
for(g in 1:G) {
       p.tau[g] <- dgamma(z.tau[g], shape=shape.tau.posterior,
rate=scale.tau.posterior)
       }
density.tau.star <- mean(p.tau)

# likelihood
log.likelihood <- 0.
for(i in 1:length(CC.df$y)){
log.likelihood <- log.likelihood +
log(dnorm(x=CC.df$y[i], mean=(C1.star + C2.star*CC.df$z.m.zbar[i]), sd=sig.star))
  }
log.likelihood

# prior
log.prior.C1 <- log(dnorm(C1.star, mean=mu.C1.prior, sd=sd.C1.prior))
log.prior.C2 <- log(dnorm(C2.star, mean=mu.C2.prior, sd=sd.C2.prior))
log.prior.tau <- log(dgamma(tau.star, shape=shape.tau.prior, rate=scale.tau.prior))
log.prior <- log.prior.C1 + log.prior.C2 + log.prior.tau
log.prior

# posterior
log.posterior.C1 <- log(dnorm(C1.star, mean=C1.star, sd=sd.C1.star))
log.posterior.C2 <- log(dnorm(C2.star, mean=C2.star, sd=sd.C2.star))
log.posterior <- log.posterior.C1 + log.posterior.C2 + log(density.tau.star)
log.posterior

log.marginal.density.z <- log.likelihood + log.prior - log.posterior
log.marginal.density.z


1./exp(log.marginal.density.x - log.marginal.density.z)
```

The example here produced this result:

```
> 1./exp(log.marginal.density.x - log.marginal.density.z)
[1] 4852.227
```

To use this code the numerical values here would be replaced by those produced by the appropriate Bayesian regression models of the two new models being compared.

Bayes Factor comparison of Smith-Watson-Topper and $S_{equivalent}$ s-N Models

The code is presented below for computing the Bayes factors to compare the Smith-Watson-Topper parameter and the Walker equivalent stress parameter in describing s-N behavior using only those data with cycle counts less than 1.1×10^5 in the Ti-6Al-4V dataset and a model without a mixture of probability densities.

```
RFL.df<-read.table(file="C:\\Documents and Settings\\user name\\My Documents\\S-
Plus Projects v6.1\\Project 3\\R export\\HCF6.txt", header = TRUE, sep = " ")
```

```
x.tau<-read.table(file="C:\\Documents and Settings\\user name\\My Documents\\S-Plus
Projects v6.1\\Project 3\\R export\\RFLSWTtau.txt", header = TRUE, sep = "\n")
x.tau<-x.tau[,]

z.tau<-read.table(file="C:\\Documents and Settings\\user name\\My Documents\\S-Plus
Projects v6.1\\Project 3\\R export\\RFLSeqTau.txt", header = TRUE, sep = "\n")
z.tau<-z.tau[,]

# Model 1 Prior parameter densities
mu.C1.prior <- 4.515
sd.C1.prior <- 10.
mu.C2.prior <- 1.82
sd.C2.prior <- 10.
shape.tau.prior <- 0.001
scale.tau.prior <- 0.001

# Parameter estimates from original draws.
C1.star <- 4.515
sd.C1.star <- 0.02464
C2.star <- -4.799
sd.C2.star <- 0.2805
tau.star <- 56.84
sd.tau.star <- 14.97
sig.star <- 0.1362

# Estimate the  density of tau.star from auxiliary draws.
tau.posterior <- 58.88
sd.tau.posterior <- 10.83
shape.tau.posterior <- (tau.posterior/sd.tau.posterior)^2
scale.tau.posterior <-  tau.posterior/(sd.tau.posterior^2)
shape.tau.posterior
[1] 29.55825
scale.tau.posterior
[1] 0.5020083

density.tau.star <- mean(dgamma(x.tau, shape=shape.tau.posterior,
rate=scale.tau.posterior))
density.tau.star
[1] 0.02640857

# likelihood
log.likelihood <- sum(log(dnorm(x=RFL.df$y, mean=(C1.star + C2.star*RFL.df$x.SWT),
sd=sig.star)))
log.likelihood
[1] 19.62342

# prior
log.prior.C1 <- log(dnorm(C1.star, mean=mu.C1.prior, sd=sd.C1.prior))
log.prior.C2 <- log(dnorm(C2.star, mean=mu.C2.prior, sd=sd.C2.prior))
log.prior.tau <- log(dgamma(tau.star, shape=shape.tau.prior, rate=scale.tau.prior))
log.prior <- log.prior.C1 + log.prior.C2 + log.prior.tau
log.prior
[1] -17.66923

# posterior
log.posterior.C1 <- log(dnorm(C1.star, mean=C1.star, sd=sd.C1.star))
log.posterior.C2 <- log(dnorm(C2.star, mean=C2.star, sd=sd.C2.star))
log.posterior <- log.posterior.C1 + log.posterior.C2 + log(density.tau.star)
log.posterior
[1] -0.4973782

log.marginal.density.x <- log.likelihood + log.prior - log.posterior
```

```
log.marginal.density.x
[1] 2.45157

# Model 2 Prior parameter densities
mu.C1.prior <- 4.515
sd.C1.prior <- 10.
mu.C2.prior <- 1.82
sd.C2.prior <- 10.
shape.tau.prior <- 0.001
scale.tau.prior <- 0.001

# Parameter estimates from original draws.
C1.star <- 4.515
sd.C1.star <- 0.02751
C2.star <- -5.08
sd.C2.star <- 0.3355
tau.star <- 47.19
sd.tau.star <- 8.679
sig.star <- 0.1475

# Estimate the density of tau.star from auxiliary draws.
tau.posterior <- 47.19
sd.tau.posterior <- 8.679
shape.tau.posterior <- (tau.posterior/sd.tau.posterior)^2
scale.tau.posterior <-  tau.posterior/(sd.tau.posterior^2)
shape.tau.posterior
[1] 29.56382
scale.tau.posterior
[1] 0.6264849

density.tau.star <- mean(dgamma(z.tau, shape=shape.tau.posterior,
rate=scale.tau.posterior))
density.tau.star
[1] 0.03295491

# likelihood
log.likelihood <- sum(log(dnorm(x=RFL.df$y, mean=(C1.star + C2.star*RFL.df$x.Seq),
sd=sig.star)))
log.likelihood
[1] 16.27062

# prior
log.prior.C1 <- log(dnorm(C1.star, mean=mu.C1.prior, sd=sd.C1.prior))
log.prior.C2 <- log(dnorm(C2.star, mean=mu.C2.prior, sd=sd.C2.prior))
log.prior.tau <- log(dgamma(tau.star, shape=shape.tau.prior, rate=scale.tau.prior))
log.prior <- log.prior.C1 + log.prior.C2 + log.prior.tau
log.prior
[1] -17.4927

# posterior
log.posterior.C1 <- log(dnorm(C1.star, mean=C1.star, sd=sd.C1.star))
log.posterior.C2 <- log(dnorm(C2.star, mean=C2.star, sd=sd.C2.star))
log.posterior <- log.posterior.C1 + log.posterior.C2 + log(density.tau.star)
log.posterior
[1] -0.5651531

log.marginal.density.z <- log.likelihood + log.prior - log.posterior
log.marginal.density.z
[1] -0.6569247

exp(log.marginal.density.x - log.marginal.density.z)
[1] 22.38732
```

Appendix

E

A Review of Statistical Fundamentals

What *IS* a "Bayes Factor" anyway?

A **Bayes Factor** is defined as the **posterior odds** favoring one model versus another when the prior odds of the two models are equal. After mathematical simplification that is:

$$B_{1,2} = P(\text{data} \mid \text{explanatory model}_1) / P(\text{data} \mid \text{explanatory model}_2)$$

The **odds**[10] for a binary outcome, is the ratio of favorable to unfavorable probabilities. The odds = *P*(yes) / *P*(no) = *P*(yes) / (1 - *P*(yes)), or in other words, the ratio of the probability *for* an event to the probability *against*.[11]

Since these definitions require several other, perhaps unfamiliar, statistical concepts, and since our development of Bayes factors depends upon it, the following paragraphs review some of this background material, beginning with the difference between "probability" and "statistics," two terms often mistaken to be synonymous.

Some Statistical Concepts:

Statistics are functions of the data (observations) that do not contain any unknown parameters. Some statistics have interesting and useful properties, like the sample mean, a statistic, that always tends to a normal distribution[12].

statistic: A numerical measure of the sample (the data) or measurable characteristic of the sample (data) such as the sample average, or the largest or smallest value observed in the sample. This is distinguished from a similar characteristic of the population, such as its mean, which is called a **parameter**. .It is important to recognize that a sample statistic, which changes from sample to sample, is *not* the population parameter, which is fixed but unknown, except for an *estimate* of it provided by the sample statistic.

[10] Note that "odds" is often used with a singular verb as is "mathematics." The log of the odds is called the **logit**, and is the basis for logistic regression used in modeling phenomena having a binary outcome, such as some types of NDE (nondestructive evaluation).

[11] The odds is not to be confused with the **odds ratio**, which is the ratio of the odds under two different scenarios, for example the failure odds with a proposed maintenance intervention, to the odds without the proposed maintenance, or the odds of the data under model 1 to the odds under model 2.

[12] The distribution of an average will tend to be Normal as the sample size increases, regardless of the distribution from which the average is taken except when the moments of the parent distribution do not exist. . This is a statement of the **Central Limit Theorem**. All practical distributions in statistical engineering have defined moments, and thus the CLT applies. The Cauchy is an example of a pathological distribution with nonexistent moments. Thus the mean (the first statistical moment) doesn't exist. If the mean doesn't exist, then we might expect some difficulties with an estimate of the mean like Xbar.

Probability itself has two, sometimes conflicting, definitions. The frequentist definition sees probability as the long-run expected frequency of occurrence. $P(A) = n/N$, where n is the number of times event A occurs in N opportunities. The Bayesian view of probability is related to degree of belief. It is a measure of the plausibility of an event given incomplete knowledge. Both definitions of probability follow the same rules, however.

Rules of probability (abridged): The rules of probability form an axiomatic system the salient features of which are:

Probability must be between zero and one inclusive: $0 \le P \le 1$.

The probabilities in a space sum to 1.

Addition Law: $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$. Note that $P(A \text{ and } B)$ must be subtracted from the sum to avoid double counting situations when A and B both occur. When more than two events are involved this construct becomes unwieldy and requires a method for organizing events, such as the directed acyclic graph (**DAG)**.

Multiplication Law [13] : $P(A \text{ and } B) = P(A) \times P(B|A)$. Notice that the common practice of "multiplying the probabilities" of A and B is only valid when events A and B are independent, so that $P(B|A) = P(B)$.

probability density or **distribution:** f(x | **θ**) where f is the probability density of x, given the distribution parameters, **θ**. For a normal distribution, $\boldsymbol{\theta} = (\mu, \sigma^2)^T$ where μ is the mean, and σ is the standard deviation. This is sometimes called a **pdf**, probability density function. The integral of a pdf, the area under the curve (corresponding to the probability) between specified values of x, is a **cdf**, **cumulative distribution function,** F(x | **θ**). For discrete f, F is the corresponding summation.

multivariate distribution: A joint probability density two or more variables. It is often summarized by a vector of parameters, which may or may not be sufficient to characterize the distribution completely. Example, the normal is summarized (sufficiently) by a mean vector and covariance matrix.

joint probability: f(x, y | **θ**) where f is the probability of x and y together as a pair, given the distribution parameters, **θ**.

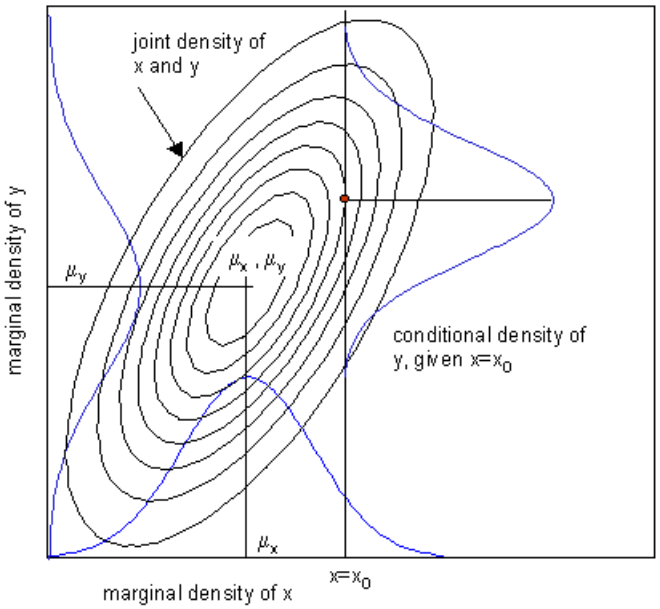
marginal probability: f(x | **θ**) where f is the probability density of x, for all possible values of y, given the distribution parameters, **θ**. The marginal probability is determined from the joint distribution of x and y by integrating over all values of y, called "integrating out" the variable y. In applications of Bayes's Theorem, y is often a matrix of possible parameter values. Joint, marginal, and conditional probability are illustrated in figure 8.

conditional probability: $f(x \mid y; \boldsymbol{\theta})$ where f is the probability of x by itself, given specific value of variable y, and the distribution parameters, $\boldsymbol{\theta}$. (See figure 8.) If x *and* y represent events A *and* B, then $P(A|B) = n_{AB}/n_B$,where n_{AB} is the number of times both A *and* B occur, and n_B is the number of times B occurs. $P(A|B) = P(AB)/P(B)$, since $P(AB) = n_{AB}/N$ *and* $P(B) = n_B/N$ so that $P(A|B) = \dfrac{n_{AB}/N}{n_B/N} = n_{AB}/n_B$ Note that in general

[13] $P(A \text{ and } B) = P(A \mid B)\, P(B)$ but more generally, $P(A \text{ and } B \mid C) \quad P(A \mid B, C)\, P(B \mid C)$

the conditional probability of A given B is **not** the same as B given A. The probability of **both** A *and* B together is $P(AB)$, and $P(A|B) \times P(B) = P(AB) = P(B|A) \times P(A)$, if both $P(A)$ *and* $P(B)$ are non-zero. This leads to a statement of **Bayes's Theorem**:
$P(B|A) = P(A|B) \times P(B)/P(A)$. Conditional probability is also the basis for **statistical dependence** and **independence**.

• Figure 8 Schematic showing joint, marginal and conditional densities



independence: Two variables, A and B, are independent if their conditional probability is equal to their unconditional probability. In other words, A and B are independent if, and only if, P(A|B)=P(A), and P(B|A)=P(B). In engineering terms, A and B are independent if knowing something about one tells nothing about the other. This is the origin of the familiar, but often misused, formula P(AB) = P(A) × P(B), which is true only when A and B are independent.

conditional independence: A and B are conditionally independent, given C, if Prob(A=a, B=b | C=c) = Prob(A=a | C=c) × Prob(B=b | C=c) whenever Prob(C=c) > 0. So the joint probability of ABC, when A and B are conditionally independent, given C, is then Prob(C) × Prob(A | C) × Prob(B | C) A directed graph illustrating this conditional independence is A ← C → B.

Bayes's Theorem: A statement of the relationship between sequential events, used to infer an antecedent, A, after observing what followed, B: $P(A|B) = P(B|A) \times P(A) / P(B)$
Bayes's Theorem is discussed in detail in Section 3.

Bayesian prior and **posterior distributions**: Using Bayes's Theorem, the **prior** distribution is what is known about an event before observing the results of an experiment, and is updated to the **posterior** density, by incorporating information learned from the experiment. (See **conditional probability**)

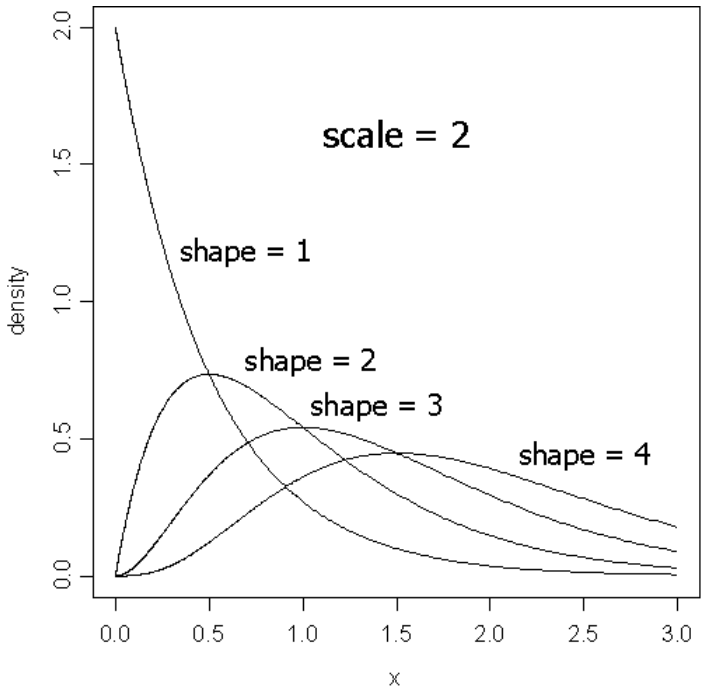
Null hypothesis, H_0: In Frequentist reasoning it is a statement that is to be *dis*proved by the data. H_0 is usually in the form of a probability density indexed by $\boldsymbol{\theta}$ - e.g.: normal, where $\boldsymbol{\theta} = (\mu, \sigma^2)^T$ In contrast, Bayes Factors use the data to provide support *for* H_0.

Gamma distribution: Often used as a prior for **precision** = $1/\sigma^2$, or for $log(\sigma^2)$.

$$f(x \mid \alpha = shape, \beta = scale) = \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta x}; x > 0 \text{ and } \Gamma(\alpha) = \int_0^\infty u^{\alpha-1} exp(-u)\, du$$

The mean is α/β, and variance is α/β^2. A special case is $f(x \mid 1, \beta)$ = **exponential**$(x \mid \beta)$.

Figure 9 Gamma Probability Density is Often Used as a Prior for Precision



Some Necessary Advanced Statistical Concepts

hyperparameters: Non-constant parameters for another distribution. For example, suppose $x \sim N(\mu, \sigma^2)$ where the mean, μ, is *not* constant but is described by its own distribution, a **hyperdistribution**, $f(\mu \mid \alpha, \beta)$, where α and β are **hyperparameters**, and f is any proper density. (A proper density integrates to unity.) (Under certain limited conditions the requirement for a proper hyperparameterization can be relaxed.) In most practical instances all parameters, (μ, σ^2) in this example, must be considered jointly. Note the *symbol* "~" is read " is distributed as...." For example $x \sim N(0, 1)$ is read "x is distributed as normal, with mean zero, variance 1," i.e., x is standard normal.

Conjugacy: A conjugate relationship between the prior and posterior densities insures that the mathematical form of the posterior is known if the prior takes a given form. For

example, it the prior for a normal mean is normal, then the posterior mean will be normal. If the prior for a binomial parameter is a beta density, then the posterior will also be beta. Many real applications are not conjugate, however, and thus were all but impossible to evaluate until recently.

likelihood: can be thought of as the "probability of the data," given specific model parameters. Likelihood is *proportional* to probability, but the proportionality constant is often unknown. Thus while probabilities must sum to unity, likelihoods, typically, do not. Likelihood has the same functional form as a probability density, but whereas with probability, the parameter values are assumed known and the probability of a future observation is sought, with likelihood the data, having been observed, are known, and the parameter values are sought, usually as those that will either maximize the likelihood (i.e. the probability that the data would be as there were observed to be) or maximize the Bayesian posterior parameter density. Likelihood is proportional to the ordinate of the density function, for **uncensored** observations.

maximum likelihood: a goodness-of-fit criterion that selects model parameters that produce the maximum value of the likelihood function. Likelihood is sometimes interpreted as "the probability of the data," making a maximum likelihood estimator one which maximizes the probability that the experiment turned out the way it did. Maximum likelihood parameter estimates are *identical* with **least-squares** (LS) estimates, when there is no data censoring. This is encouraging to engineers who have used LS estimators for centuries.

maximum likelihood estimator, mle: Value of a parameter in a statistical model that maximizes the probability that the experiment turned out the way it did. The Bayesian analog is **maximum *a posteriori***, **MAP**, estimator - the value that maximized the posterior density.

likelihood ratio: a goodness-of-fit criterion that compares the ratio of the likelihood for competing values of the model parameter set with the set that produces the maximum likelihood. Used in constructing **confidence regions** in the neighborhood of the maximum. The likelihood ratio has a Chi-square distribution[14] with degrees of freedom equal to the number of parameters in the model. Likelihood ratios are similar to **Bayes factors**, but require nested models so that the likelihood proportionality constants in the ratio are equal (but unknown), and thus cancel in the ratio. Bayes factors are based on the marginal density, and thus can be compared directly.

least-squares: (or least-squared error): A goodness-of-fit criterion that compares the model prediction with the data that produced it. An "error" is defined as the difference between an observed response value and the predicted response. The criterion selects model parameters that produce the smallest summed squared error. This method has been used successfully by engineers for over 200 years since Gauss popularized it. The method breaks down however when the data are **censored**, since the true value of the response is *not* observed, only that it exceeds some censoring value.

censored data: An observation known to be greater than (or less than, or bounded by) some censoring value, while it's exact value is unknown. e.g.: a fatigue test **runout**. The **likelihood** for a right-censored observation (*e.g.* a test terminated after N cycles, without failing) is equal to one minus the cumulative probability of failure at N cycles. .

[14] Strictly speaking these are **asymptotic distributions**, i.e., for large sample sizes the distribution of the parameter approaches the given distribution.

Markov chain: Given the current state E_i for a series of events, E_1 ,E_2 , ... E_n, and that the probability that the next state is E_j depends only on the current state and not on how the current state was achieved. Then given an $n \times n$ matrix of transition probabilities, $p_{i,j}$, sometimes called the **transition kernel**, the probabilities can be calculated for a series of trials (samples from the original series) knowing only the initial state, and the transition probabilities. (Notice that you don't need the entire transition matrix beforehand, only the ability to calculate $p_{i,j}$ when in current state i.) In certain situations the long-run behavior of this sample, a Markov chain, becomes independent of the starting state, and **converges in distribution** to some probability density of interest.

convergence in probability, convergence in distribution: Engineers are familiar with mathematical convergence - that the terminal value of a series approaches some limit as the number of terms increases.. They are less familiar with an analogous statistical concept of "**convergence in distribution**," where the characteristic of the limit isn't a single value, but rather that the character of the sequence itself approaches some specific distribution. An example is the **central limit theorem**:

central limit theorem, CLT: If x_1, x_2 , ... x_n are a sequence of independent identically distributed (***iid***) random variables, with finite mean μ_x and variance σ^2_x then z_n *converges in distribution* to $N(0, 1)$ as *n* becomes large, and

$$z_n = (\bar{x}_n - E(\bar{x}_n) / \sqrt{\text{var}(\bar{x}_n)} = (\bar{x}_n - \mu_x) / (\sigma_x / \sqrt{n})$$

where $E(.)$ is the **expectation operator**. This result does *not* depend on the original distribution of x, only that the mean and variance are finite. And "large" *n* may be on the order of a dozen observations. This concept of convergence in distribution is fundamental to the workings of the **Metropolis-Hastings MCMC** algorithm, and the **Gibbs sampler**, where the behavior of the collection of sampled values approaches the desired probability density as *n* increases. (Here *n* typically exceeds 1000.) In simpler terms the CLT says that for large *n*, the *average* of *n* samples taken from *any* distribution with finite mean and variance will have a *normal* distribution with mean equal to the parent distribution's mean, and variance equal to the parent variance divided by the sample size, *n*. Much Frequentist inference is based on asymptotic theory of the CLT, which in turn is based on long-run behavior (large *n*). The "**Bayesian Central Limit Theorem**" is analogous, and states that the *posterior density* of a continuous parameter, θ, is asymptotically normal.

expectation: The average. The **expectation operator** for a discrete density, $f(x)$, is $E(x)=\Sigma x f(x)$, and for continuous density $f(x)$, $E(x)=\int x f(x) dx$.

ergodic: Time-dependent and other sequential processes are called **ergodic** if the eventual distribution of states in the system does not depend on the starting state so the random sequence S_m from time = t_n to time = t_{n+m} does not depend on *n* as $n \to \infty$.

(direct-sampling) Monte Carlo: A kind of simulation for finding approximate solutions to statistical problems that are resistant to closed-form methods. The procedure is often naively misused by simply sampling independently from the various distributions with little regard for the interactions among the variables. Directed acyclic graphs can help organize these interrelationships to take advantage of conditional independences, making **MCMC** or **Gibbs sampling** possible.

MCMC, Markov Chain Monte Carlo: An iterative sampling procedure of which the **Gibbs Sampler** is a special case (Chib and Greenberg, 1995), based on substitution sampling, where the true, but unknown, joint posterior density is sampled from using a local proposal density and a Markov state-change probability function to accept or reject the current offering from the proposal density. The sequence of samples **converges in distribution** to the desired joint posterior density.

Gibbs Sampler: An iterative statistical sampling procedure that samples a variable, conditioned on the current values of all other variables, then samples the next variable, conditioned on the current, but continuously updated, values of the other variables. The process is repeated some large number of times (say, 1000) until convergence. It can be shown (Carlin and Louis, 1996, pp. 163 ff.) that the sequence of samples **converges in distribution** to the joint posterior density of interest. Geman and Geman (1984) gave the somewhat incongruous name to their algorithm because they saw the distribution of pixel "states" in an image is being analogous to the Gibbs distribution of states in solid-state physics.

Bayesian network: a visual representation of a joint probability density over a set of random variables linked together by Bayes's theorem where the posterior distribution of one application can provide the prior for a subsequent one.

inference: In Bayesian networks **inference** is concerned with calculating the conditional probability distribution of a subset of nodes in a graph, given another subset of the nodes.

forward sampling: Sampling in advance of the data. This is similar to ordinary MC simulation for frequentists for whom simulation is detached from data analysis.

confidence interval (credibility interval): (Frequents' definition) A numerical interval said to contain the true parameter value in some percentage of similar repeated intervals. A confidence interval for the binomial parameter, p, for example, could be constructed assuming asymptotic normal behavior of **maximum likelihood estimators** and parameters $mean = \hat{p}\ and\ \mathrm{var} = \hat{p}(1-\hat{p})/n$ So for a $(1\text{-}\alpha/2)$ confidence interval, where $z_{0.05}=1.645$, $\left(\hat{p} - z_{\alpha}\sqrt{\hat{p}(1-\hat{p})/n} \le p \le \hat{p} + z_{\alpha}\sqrt{\hat{p}(1-\hat{p})/n} \right)$ is a 90% CI. (Bayesian definition) An interval said to contain the true value with some given probability, also called a Bayesian probability interval.

Appendix

F

Regression: Building a Mathematical Model

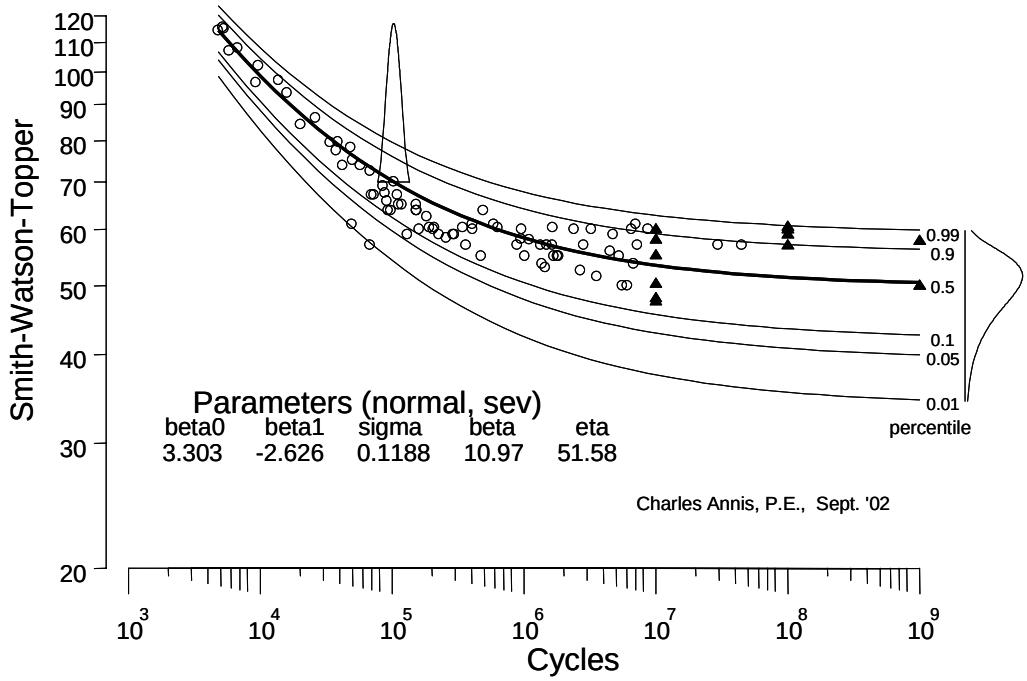
Fundamentals of Mathematical Regression

Ordinary Least Squares (OLS):

When all fatigue specimens fail (no runouts), Ordinary Least-Squares (OLS) is the accepted method for estimating the parameters of the s-N model. This method has been the basis of engineering data analysis for the 200 years since Gauss popularized it.

Some mathematical model is proposed which relates stress (or strain, and/or temperature, or other relevant parameter) with cycles to failure, N. The goal is to choose parameters for the model which "best" fits the data. Gauss said that "best" means that the summed squared error of the residuals is a minimum. (A residual is the difference between an observation and the model prediction.) Another way of saying the same thing is that the variance of the observations about the predicted behavior is as small as possible.

Figure 10 RFL Data, Cycles vs. SWT, log-log axes.



Given this criterion for goodness, the OLS method first writes the equation for the sum of the squares of the differences between the observed and expected lives. This relationship is then differentiated with respect to each of the model parameters, and these derivatives set equal to zero. The simultaneous solution to these equations (the "Normal" equations)

provides the desired least-squares estimates of the parameter values. (Statisticians don't talk about "measuring" a parameter value; they "estimate" it. That's because the estimate will change slightly given different or new data, something that wouldn't happen with something which could be measured without error.)

Now, if the equation chosen to represent the s-N behavior is linear in the model parameters, then the solution to the Normal equations can be written down directly: Consider $\boldsymbol{y}=\boldsymbol{Xb}$, where $\boldsymbol{y}$ is the column vector of the dependent variable observations (such as $log(N)$) and $\boldsymbol{X}$ is an n by m matrix of life-controlling variables (such as stress or log(stress), and/or temp, or whatever), n is the number of observations, and m is the number of model parameters, including the offset. $\boldsymbol{b}$ is the column vector of length, m, of model parameters. (The first column of $\boldsymbol{X}$ is all ones, unless the offset is defined to be zero, and the equation is forced to go through the origin.) The general solution is $\boldsymbol{b}$-hat = $(\boldsymbol{X}^T\boldsymbol{X})inv\ \boldsymbol{X}^T\boldsymbol{Y}$, where the superscript ^{T} indicates matrix transpose, and the inv indicates its antecedent is to be inverted. The *"hat"* (a carat above a parameter) indicates an estimate rather than a known value[15].

Censored Data (Runouts):

The forgoing is a summary of current engineering practice, used with success for 200 years. This approach is unworkable, however, if some of the observations are “censored,” that if the actual failure lifetime is unknown because the test was stopped before the specimen failed. The OLS approach is based on minimizing the summed squared residuals, but because the specimen could have failed at any point after the test was suspended, the "error" (residual) can not be defined, thus it can't be included in the summed squared error to be minimized, and so the LS method breaks down.

This problem was solved in the comparatively recent past by R.A. Fisher in the early decades of the last century, and brought into engineering practice only about 15 years ago. Fisher looked at the problem of parameter estimation using a different criterion for goodness. Fisher said the "best" parameter value would be the one maximizing the likelihood that the experiment would have turned out the way it actually did. He said that you could choose any parameter values you wanted, but some would be more likely to be the true values, given the experimental results.

Likelihood:

Picture the s-N data with a best-fit line through it (*e.g.*: figure 10). Now imagine a (normal) distribution of lives scattered about the line, at a constant stress parameter, SWT, for example. The likelihood (of the line's being in the right place) is the ordinate of the probability distribution which is centered at the model value of N. Obviously, if the line is nowhere near the data, the normal distribution won't be centered appropriately, and the ordinates evaluated at the N values will be low. We want to put the curve through the data so its likelihood is maximized.

The method of maximum likelihood is approached as was the method of least-squares: beginning with the likelihood equation, which is just the product of all the individual likelihoods. For practical purposes it's helpful to take the log of the likelihood equation

[15] One of the most serious sources of confusion about statistics is the difference between the true – but unknowable – value of a parameter, and an *estimate* of it, based on the data. Much of contemporary engineering “probabilisitcs” is questionable, if not altogether wrong, due to ignorance of this difference.

because it turns all the products into sums (of logarithms). Next, differentiate this equation with respect the model parameters. (It's much easier to differentiate a sum than a series of products.) These derivatives are set equal to zero and the resulting equations are solved simultaneously. This usually requires an iterative solution. Now, because the logarithm is a monotone function, it reaches a maximum when the variable of which it the logarithm reaches a maximum, so the solution to the maximum of the log of the likelihood occurs at the same parameter values as the maximum of the likelihood function itself.

OLS estimators ARE ML estimators:

How do parameters estimated with Fisher's maximum likelihood criterion compare with those estimated using Gauss's least-squares criterion? They are EXACTLY the same. Not close - exact - (given that the errors are normally distributed, which is usually the case.) That means that if there were NO censored observations, the ML method produces the identical results as the method we've been using for 200 years, a comforting situation.

So what about a runout? It could be represented by the ordinate at the N cycles where it was discontinued, OR at the ordinate at a few cycles more, OR at even more cycles after that, since it could have failed at any of those cycle-counts. Since exactly where the failure would have occurred is unknown, only that it has to be **after** the N observed cycles, the relative likelihood (of the curve being in the right place) is that fraction of the area under the normal curve to the right of the suspension, since the data were **right censored**. This definition of likelihood also works for left censored observations and for interval censored tests. (An example of interval censoring could be a test which failed over the weekend. The cycle counter was working Friday afternoon, but the specimen was found failed Monday morning, and the cycle count is in doubt. Here the likelihood would be the area under a normal curve between the last known cycle count, and the cycle count estimated by the test frequency and the duration of the interval.)

This approach to modeling fatigue data works well for $N \leq 10^7$ cycles but begins to suffer when many high cycle runouts must be considered. This suggests a model that can consider not only the variability in N, but also variability in runout stress or fatigue limit.

Appendix

G

Notes on Goodness-of-Fit Tests for Statistical Distributions

Anderson-Darling

Of the many quantitative goodness-of-fit techniques (e.g.: Komolgorov-Smirnov, Anderson-Darling, Shipiro-Wilk, von Mises), the Anderson-Darling test seems best for our purposes, since it is more sensitive to deviations in the tails of the distribution than is the older Komolgorov-Smirnov test.

Anderson-Darling can be applied to any distribution, but finding the necessary tables of critical values may require purchase of D'Agostino and Stephens (1986). Include here are the two most useful tables, for the normal and lognormal, and for the Weibull, exponential, and Gumbel.

For the **normal** and **lognormal** distributions, the test statistic, A^2 is calculated from

$$A^2 = -n - (1/n)\sum_{i=1}^{n}(2i-1)[\ln(w_i + \ln(1-w_{n-i+1})] \qquad \text{Equation 17}$$

where n is the sample size, and w is the standard normal cdf, $\Phi[(x-\mu)/\sigma]$.

This formula needs to be modified for small samples,

$$A_m^2 = A^2\left(1+\frac{0.75}{n}+\frac{2.25}{n^2}\right) \qquad \text{Equation 18}$$

and then compared to an appropriate critical value from the table below.

α	0.1	0.05	0.025	0.01
A^2_{crit}	0.631	0.752	0.873	1.035

Reference: D'Agostino and Stephens, _Goodness-Of-Fit Techniques, Marcel-Dekker, New York, 1986, Table 4.7, p.123. All of Chapter 4, pp.97-193, deals with goodness-of-fit tests based on empirical distribution function (EDF) statistics.

The other popular family of distributions includes the Weibull for distributions of minima, and Gumbel for distributions of maxima. The Gumbel variable X, and Weibull variable Y are related by $X=\ln(1/Y)$. A Weibull distribution with the shape parameter equal to one produces the exponential distribution as a special case.

For the **Weibull** (and **Gumbel**) distributions, the test statistic, A^2 is again calculated from

$$A^2 = -n - (1/n)\sum_{i=1}^{n}(2i-1)[\ln(w_i + \ln(1-w_{n-i+1})] \qquad \text{Equation 19}$$

just as for the normal, but *w* is the cdf for the distribution under consideration. For the Weibull, $w_i = F(x) = 1 - \exp(-(x_i/\eta)^{\beta})$, and η, β, are the model scale and shape parameters.

This formula needs to be modified for small samples,

$$A_m^2 = A^2\left(1 + \frac{0.2}{\sqrt{n}}\right) \qquad \text{Equation 20}$$

and then compared to an appropriate critical value from the table below.

α	0.1	0.05	0.025	0.01
A^2_{crit}	0.637	0.757	0.877	1.038

(Ref: D'Agostino and Stephens, 1986, Table 4.17, p.146)

NOTE: Although the Weibull, a distribution of "weakest-link" minima, is more widely known, it may not be the best choice for some of our distributions, as its sister, the Gumbel, the asymptotic distribution of maxima.

Graphical Methods: The InterOcular Trauma Test.

The InterOcular Taruma Test is simple: Plot the data. If it hits you between the eyes, it's significant. While perhaps facetious, this simple graphical method is very powerful and should be part of any statistical analysis.

Using an appropriate probability grid, plot the cdf (cumulative distribution function) data. The data will appear as a straight line on the correct grid. Fortunately, there are two grids that will cover the most common distributions and making additional ones isn't too complicated.

To make a Normal (or Lognormal) grid notice that the y-axis is in terms of number of standard deviations, although it's not labeled that way. So the middle of the graph is at y=0 and that corresponds to cdf, *F(x)* = 0.5 = 50%. One standard deviation unit up (or down) is *F(x)* = 0.8413 (or 0.1587). Two units up (or down) is 0.9772 (or 0.0228). Three units up (down) is 9987 (0.0013). And so on. These values, and intermediate values chosen for graphing purposes, are tabulated everywhere and can be found using MS EXCEL also. If the x-axis is to represent a normally distributed x, then it's Cartesian. If lognormal is what you want, then the x-axis is logarithmic.

To make a grid for the exponential distribution, we can take advantage of knowing that the exponential distribution is a special case of the Weibull, when the slope parameter, beta, equals one. The Weibull grid is even easier to make than the Normal grids because *F(x)* has a closed form (unlike the Normal), viz. $F(x) = 1 - \exp(-(x_i/\eta)^{\beta})$. A little arithmetic shows that

$$\exp(-(x/\eta)^{\beta}) = 1 - F(x) \qquad \text{Equation 21}$$

$$-(x/\eta)^{\beta} = \ln(1 - F(x)) \qquad \text{Equation 22}$$

$$\beta \ln(x/\eta) = \ln(-\ln(1 - F(x))) = \beta \ln(x) - \beta \ln(\eta) \qquad \text{Equation 23}$$

This is a linear equation, $Y=M*X+B$, where $X=ln(x)$ and $Y=ln(-ln(1-F(x)))$, with slope of $M = \beta$, and intercept $= -\beta\, ln(\eta)$. (Remember that β and η are constants for a given fit so $\beta\, ln(\eta)$ is also a constant.)

The grid then is simply $Y=ln(-ln(1-F(x)))$, and $X=ln(x)$. (Logarithmic x-axis, and log(logarithmic) y-axis.) Notice that the (0, 0) point occurs at x =1 (so that $ln(x)=0$, and $y = \approx 0.632$, since $ln(-ln(1-0.632)) \approx 0$. (The actual value is at $y=1-exp(-1) \approx 0.6321205588$)

To plot the data mean ranks are used because they more closely agree with MLE cdf plots, than median rank plotting positions. The mean rank (for the *i*-th *un*censored observation,

$$y_i = \frac{i}{n+1} \qquad \text{Equation 24}$$

where $i = 1, 2, 3, \ldots n$, and n is the sample size.

Appendix

H

A (Brief) Recent History of Bayesian Methods

Overview of Bayesian Computational Methods

Consider equation 25, Bayes's Theorem that was stated earlier and is repeated here:

$$P(\theta \mid y) = \frac{P(y \mid \theta)\pi(\theta)}{P(y)} \qquad \text{Equation 25}$$

where

$$P(y) = \iint \ldots \int P(y \mid \theta)\pi(\theta)\, d\theta$$

Until recently it was simply not possible to perform the necessary integrations. Conventional Gaussian quadrature is impractical for models with more than a few parameters, because the number of function evaluations increases exponentially with the number of dimensions. (Carlin and Louis, 1996), so Bayesians were forced to use conjugate relationships[16] that were only grossly approximate, and then use approximate methods to evaluate the result. While Bayesian philosophy was very appealing because it provides a mechanism for combining prior with current knowledge, in practice Bayesian methods were simply too impractical for many applications.

Rediscovery of **Markov Chain Monte Carlo (MCMC)** methods (Metropolis, et al. 1953, Chib and Greenberg, 1995), especially the **Gibbs Sampler** (e.g. Casella and George, 1992), has removed this impediment. Powerful software, such as **WinBUGS**, (Spiegelhalter, et al.,1996) for performing Bayesian regression and other Bayesian computations now put the benefits of Bayesian methods within the reach of the practicing engineer.

While Markov Chain Monte Carlo sounds like the Monte Carlo method familiar to most engineers, the two methods have very little in common, other than their name. To understand MCMC it is helpful to put aside what you may know about its namesake.

Different Flavors of Monte Carlo

Direct-Sampling Monte Carlo

The direct methods generate random samples from probability densities and use these samples to evaluate some function of interest. This is usually accomplished through the

[16] Conjugate relationships between the prior and posterior densities were the foundation for applied Bayesian methods for more than 200 years. Conjugacy insures that the mathematical form of the posterior is known if the prior takes a given form. For example, it the prior for a normal mean is normal, then the posterior mean will be normal. If the prior for a binomial parameter is a beta density, then the posterior will also be beta. Many real applications are not conjugate, however, and thus were all but impossible to evaluate before the advent of Markov Chain Monte Carlo methods during the 1990s.

inverse of the marginal cumulative distribution function. (Implicit here, yet often ignored by engineers, is that the model parameters are independent, making the closed-form statement of the marginal density feasible.) In practice the (assumed independent) parameters are sampled in sequence, and then used in some physical model. The result is recorded and the process repeated many times. Finally the statistics of the aggregate behavior, the recorded values, are computed.

MCMC, by contrast, samples from a multivariate poster distribution of model parameters. The object in Bayesian regression is to determine the highest posterior density for the regression model parameters.

Markov Chain Monte Carlo

Most engineers are familiar with direct-sampling Monte Carlo, described above, and perhaps familiar with some of its variants, indirect methods like importance sampling, rejection sampling, and expectation maximization (EM). We are less familiar with **Markov Chain** methods like substitution sampling, Gibbs sampling and the Metropolis-Hastings algorithm (cf. Carlin and Lewis 1996; MacKay 1999; Ripley, 1996). Only the **MCMC** methods are unaffected by the "curse of dimensionality," and of these the **Metropolis-Hastings** algorithm and particularly the **Gibbs Sampler** are best suited for evaluating Bayesian networks.

Direct-sampling methods attempt to sample from the entire probability space and thus from the joint probability density of interest indirectly, usually inversely through the marginal **cdf**s. **Markov Chain Monte Carlo**, **MCMC**, methods, in contrast, *sample directly from the joint probability density itself* (!) Because they do not have to sample everywhere in the probability space, only where the variables most probably reside, MCMC methods are not fettered by the problem of large dimensions. There is a price for all this, of course. MCMC methods, like all Monte Carlo methods including direct-sampling, produce, not the joint density itself, but only a *sample* from it. (The sample can be as large as desired, however.) Furthermore, the individual elements of the sample are autocorrelated. These problems are easily overcome, however, and are a small price to escape the shackles of the "curse."

These are iterative, rather than direct, sampling methods, and rely on the idea of **convergence in distribution**. It can be shown (cf.: Carlin and Louis, 1996; MacKay, 1999) that under suitable conditions that the *sequence* of samples taken under the Metropolis-Hastings or Gibbs Sampler algorithms ultimately becomes **ergodic**, with elements of the sequence representing samples from the joint probability density being simulated. The original idea was proposed nearly 50 years ago by Metropolis and colleagues at Los Alamos National Laboratory to simulate atomic physics (Metropolis, et al., 1953).

To implement the **transition kernel** (e.g.: Ripley, 1996) the original Metropolis algorithm required a symmetric function as part of its transition decision rule. The algorithm was improved by Hastings, (Hastings, 1970), who removed the requirement for symmetry in the candidate density and devised a more refined transition rule. For some well-posed problems, including many of those that can be represented by a directed, acyclic graph, DAG, the **Gibbs Sampler** is even more effective. Although Gemen and Geman (1984) noted the similarity of their algorithm with that of Metropolis, et al., it is only recently recognized as being a special case of **Metropolis-Hastings** (Chib and Greenberg, 1995). The Gibbs sampler is easier to implement if all the probabilities are supplied in terms of their conditional densities (MacKay, 1999). Many implementations of the Gibbs sampler, such as **WinBUGS**, can take advantage of any **conjugate** relationships, resorting to the

more general, but slower to converge, Metropolis Hastings algorithm when necessary (Spiegelhalter, et al.,1996, see also Carlin and Louis, 1996).

It could be argued that MCMC methods are the most important discovery in applied statistics since R.A. Fisher proposed likelihood as a criterion for parameter estimation, replacing the method-of-moments in the early decades of the last century. While Bayes factors themselves are not new -Jeffries first proposed them in 1935, (Kass and Raffery, 1995) - MCMC methods, especially the ideas of Chib (1995) and Chib and Jeliazkov (2001, 2002), have finally made computing them feasible.

Appendix

I

Bibliography

1) Annis, Charles, "Bayesian Network Analysis of the 1-D Oscillator Problem - A Directed Acyclic Graph Paradigm for Probabilistic High Cycle Fatigue Risk Assessment," supported under Independent Contractor Agreement 01-S441-48-01-C4, Universal Technology Corporation Prime Contract F33615-98-C-2807, September, 2001

2) Annis, Charles, "Modeling High Cycle Fatigue with Markov Chain Monte Carlo: A New Look at an Old Idea," AIAA 2002-13800, presented at 43rd AIAA/ASME/ASCE/AHS Structures and Dynamics Conference, Denver, CO, 22-25 April, 2002

3) Annis, C., "Probabilistic Life Prediction Isn't as Easy as It Looks," **Probabilistic Aspects of Life Prediction, ASTM STP-1450**, W. S. Johnson and B.M. Hillberry, Eds., ASTM International, West Conshohocken, PA, 2003.

4) Berger, James O. **Statistical Decision Theory and Bayesian Analysis**, 2nd ed., Springer-Verlag, 1985

5) Best, N.G., and M. K. Cowles and S. K. Vines (1995). **CODA Convergence Diagnosis and Output Analysis software for Gibbs Sampler output**: Version 0.3

6) Bos, Charles S., "A Comparison of Marginal Likelihood Computational Methods," Faculty of Economics and Operations Research, *Vrije Universiteit Amsterdam*, Feb., 2002

7) Box, George E. P., William G. Hunter, and J. Stewart Hunter, **Statistics for Experimenters**, Wiley, 1978

8) Box, George. E.P. and George C. Tiao, **Bayesian Inference in Statistical Analysis**, Addison Wesley, 1973

9) Box, Joan Fisher, **R.A. Fisher, the Life of a Scientist**, John Wiley & Sons ISBN: 0-471-09300-9 June 1978

10) Carlin, Bradley P., and Siddhartha Chib, "Bayesian Model Choice via Markov Chain Monte Carlo Methods," *Journal of the Royal Society* **B** Vol. 57, No. 3 pp 173-484, 1995

11) Carlin, Bradley P. and Thomas A. Louis, **Bayes and Empirical Bayes Methods for Data Analysis**, Chapman and Hall/CRC, 1996

12) Casella, G. and George, E., "Explaining the Gibbs Sampler," *The American Statistician*, 46, 167-174, 1992

13) Chib, Siddhartha, “Marginal Likelihood from the Gibbs Output” *Journal of the American Statistical Association*, Vol. 90, No. 432. (Dec., 1995), pp. 1313-1321.

14) Chib, Siddhartha, “Bayesian Estimation and Comparison of Multiple Change Point Models,” John M. Olin School of Business, Washington University, April, 1996.

15) Chib, Siddhartha, and Ivan Jeliazkov, “Marginal Likelihood from the Metropolis-Hastings Output,” *Journal of the American Statistical Association*, Vol. 96, No. 453. (March, 2001), pp. 270-281.

16) Chib, Siddhartha, and Edward Greenberg, "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, November, 1995, Vol. 49, no. 4, pp. 327-335

17) Chib, Siddhartha, and Ivan Jeliazkov, “Accept-Reject Metropolis-Hastings Sampling and Marginal Likelihood Estimation,” July 2002, working paper to be published.

18) Congdon, Peter, **Bayesian Statistical Modeling**, Wiley, 2001

19) Cox, D. R. and D. V. Hinkley, **Theoretical Statistics**, Chapman and Hall, 1974

20) D'Agostino and Stephens, **Goodness-Of-Fit Techniques**, Marcel-Dekker, New York, 1986, Table 4.7, p.123. All of Chapter 4, pp.97-193, deals with goodness-of-fit tests based on empirical distribution function (EDF) statistics.)

21) DeGroot, Morris H., “Lindley’s Paradox: Comment,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp336-339

22) DeGroot, Morris H., “Lindley’s Paradox: Comment,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp336-339

23) Demster, A.P., “Lindley’s Paradox: Comment,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp339-341

24) Di Ciccio, Thomas J., Robert E. Kass, Adrian Raftery, and Larry Wasserman, “Computing Bayes Factors By Combining Simulation and Asymptotic Approximations,” *Journal of the American Statistical Association,* Volume 92, Number 439, September 1997, p.903

25) Efron, Bradley, "Why isn't everyone a *Bayesian*? With discussion and a reply by the author", *American Statistician*, 40-1, pp.1-11, ISSN:0003-1305, 1986

26) Fisher, Ronald A., **Statistical Methods for Research Workers***.* (First published in 1925; 14th edition was ready for publication in 1962, when Fisher died, and was published in 1990, by the Oxford University Press, along with **Experimental Design** and **Scientific Inference**, with corrections to the 1991 edition, in1993

27) Gelfand, Alan E. and Adrian F.M. Smith, “Sampling-Based Approaches to Calculating Marginal Densities,” *Journal of the American Statistical Association,* Volume 85, Number 410, June 1990, p.398-409

28) Geman, S. and Geman, D. "Stochastic Relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721-742, 1984

29) Good, I.J., “Lindley’s Paradox: Comment,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp342-344

30) Good, Irving John and James Flinn Crook, “The Bayes/Non-Bayes Compromise and the Multinomial Distribution,” *Journal of the American Statistical Association*, September, 1974, Volume 69, Number 347, pp. 711-720.

31) Green, Peter, and Tony O’Hagan, “Model choice with MCMC on product spaces without using pseudo-priors,” University of Bristol and University of Nottingham, Nottingham University Statistics Research Report 98-01, 1998, updated March 2000

32) Hahn, Gerald J. and William Q. Meeker, **Statistical Intervals, A Guide for Practitioners**, Wiley Interscience, 1991

33) Han, Cong and Bradley P. Carlin, “Markov Chain Monte Carlo Methods for Computing Bayes Factors: A Comparative Review,” *Journal of the American Statistical Association*, Vol. 96 No. 455, pp.1122-1132

34) Hastings, W..K., "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrica*, 57, 97-109, 1970

35) Hill, Bruce M., “Lindley’s Paradox: Comment,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp344-347

36) Johnson, Valen E., Mark Fitzgerald, Harry F. Martz, “Discussion: Pascual and Meeker - Estimating Fatigue Curves with the Random Fatigue-Limit Model," TECHNOMETRICS Vol. 41, No. 4, p.294-296, November 1999.

37) JRSSB (1993) "Discussion on the Meeting on the Gibbs Sampler and Other Markov Chain Monte Carlo Methods, *"Journal of the Royal Statistical Society (1993), Series B (Methodological)"* 55, 53-102

38) Kass, Robert E. (moderator) and Bradley P. Carlin, Andrew Gelman, and Radford M. Neal, "Markov Chain Monte Carlo in Practice: A Roundtable Discussion," Joint Statistical Meetings, August 4, 199, reprinted in *The American Statistician* , May 1998, Vol. 2, No. 2, pp.93-100

39) Kass, Robert E., “Lindley’s Paradox: Comment,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp347-349

40) Kass, Robert E., and Adrian E. Raftery, “Bayes Factors,” *Journal of the American Statistical Association*, June 1995, Volume: 90 Number: 430 Page: pp.773-795

41) Kendall, Maurice and A. Stuart**, The Advanced Theory of Statistics, Vol 2, Inference and Relationship**, 4th ed., Charles Griffin & Co., 1979

42) Lawless, Jerald F., **Statistical Models and Methods for Lifetime Data**, Wiley, 1982

43) Lindley, D.V., “Lindley’s Paradox: Comment,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp334-336

44) Lindley, D.V., **Bayesian Statistics, A Review**, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, 1972

45) Lindley, D.V., “A Statistical Paradox,” *Biometrika*, Volume 44, Issue ½ (June, 1975), pp.187-192

46) MacKay, D.J.C., "Introduction to Monte Carlo Methods," **Learning in Graphical Models**, MIT Press, pp175-204, 1999

47) Meeker and Escobar, **Statistical Methods for Reliability Data**, Wiley, 1998.

48) Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21(6), 1087--1092, 1953

49) Mood, Alexander M., Franklin A. Graybill, and Duane C. Boes, **Introduction to the Theory of Statistics**, 3rd ed., McGraw-Hill, 1974

50) Neal, Radford M., “Erroneous Results in ‘Marginal Likelihood from the Gibbs Output’“ Dept. of Statistics and Dept. Computer Science, University of Toronto, March 1999.

51) Neter, John, William Wasserman, Michael H. Kutner, **Applied Linear Statistical Models**, 2nd ed., IRWIN, 1985

52) Oberkampf, William L., “Investigation of Evidence Theory for Engineering Applications,” AIAA 2002-159, presented at the 4th National Non-Deterministic Approaches Forum, Denver, CO, April, 2002

53) Pascual and Meeker, "Estimating Fatigue Curves with the Random Fatigue-Limit Model," TECHNOMETRICS Vol. 41, No. 4, p.277-302, November 1999

54) Press, S. James, **Bayesian Statistics, Principles, Models, and Applications**, Wiley Interscience, 1989

55) Ripley, Brian D., "Belief Networks," **Pattern Recognition and Neural Networks"**, Cambridge University Press, Chapter 8, (1996)

56) Ritter, Christian, and Martin A. Tanner, “Facilitating the Gibbs Sampler: The Gibbs Stopper and the Griddy-Gibbs Sampler, *Journal of the American Statistical Association*, Vol. 87, No. 419, September, 1992, pp. 861-868

57) Shafer, Glenn, “Lindley’s Paradox,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp.325-348

58) Shafer, Glenn, “Rejoinder: Lindley’s Paradox,” *Journal of the American Statistical Association*, June 1982, Volume: 77 Number: 378 Page: pp.349-351

59) Sinharay, Sandip, and Hal S. Stern, “On the Sensitivity of Bayes Factors to the Prior Distributions,” *The American Statistician*, August 2002, Vol. 56, No. 3, pp.196-201

60) Spiegelhalter, D.J. and A Thomas and N G Best and W R Gilks (1996). **BUGS: Bayesian inference Using Gibbs Sampling, Version 0.5**, (version ii)

61) Spiegelhalter, D.J. and A Thomas and N G Best and W R Gilks (1996). **BUGS Examples Volume 1, Version 0.5**, (version ii)

62) Spiegelhalter, D.J. and A Thomas and N G Best and W R Gilks, **BUGS Examples Volume 2, Version 0.5**, (version ii), 1996

63) Spiegelhalter, David, Andrew Thomas, and Nicky Best, **WinBUGS Version 1.3 User Manual** (part of the WinBUGS package) April 2000.

64) Tierney, Luke, "Markov Chains for Exploring Posterior Distributions," Annals of Statistics, Vo. 22, Issue 4, Dec. 1994, pp.1701-1728

65) Verdinelli, Isabella and Larry Wasserman, “Computing Bayes Factors Using a Generalization of the Savage-Dickey Density Ratio,” *Journal of the American Statistical Association*, June 1995, Volume: 90 Number: 430 Page: pp.614-618