Actuarial Modeling with MCMC and BUGS : Additional Worked Examples *

David P.M. Scollnik †

Abstract

This set of worked examples is a follow-up and companion to the paper entitled "Actuarial Modeling with MCMC and BUGS" by Scollnik (2000). The aim of both documents is towfold. First, to introduce the actuarial practitioner to Markov chain Monte Carlo (MCMC) and illustrate its application in a variety of actuarial contexts. Second, to introduce the actuarial practitioner to the BUGS and WinBUGS software programs for the analysis of complex statistical models using MCMC.

*This paper was supported by a grant from the Actuarial Education and Research Fund (AERF).

†David P.M. Scollnik, A.S.A., Ph.D., is Associate Professor, Department of Mathematics and Statistics, University of Calgary, Calgary, Alberta, Canada, T2N 1N4, e-mail, scollnik@ucalgary.ca .

Contents

1 Introduction

This set of worked examples is a follow-up and companion to the paper entitled "Actuarial Modeling with MCMC and BUGS" by Scollnik (2000). The aim of both documents is towfold. First, to introduce the actuarial practitioner to Markov chain Monte Carlo (MCMC) and illustrate its application in a variety of actuarial contexts. Second, to introduce the actuarial practitioner to the BUGS and WinBUGS software programs for the analysis of complex statistical models using MCMC. The reader is expected to have read Scollnik (2000) in advance of the current document. Illustrative BUGS / WinBUGS program files for the worked examples described in this paper are available from the author's website at www.math.ucalgary.ca/∼scollnik/abcd/.

The BUGS project home page is located at www.mrc-bsu.cam.ac.uk/bugs. This page links to others making available the 'classic' BUGS and WinBUGS software. After downloading the WinBUGS software, be sure to also complete and submit the request form for the key for unrestricted use of WinBUGS. Without it, you will be unable to run models containing 100 nodes or more.

A large number of articles concerned with MCMC and its applications are available on the MCMC Preprint Service at www.mcs.surrey.ac.uk/Personal/S.Brooks/MCMC/. Although many of the papers at this site are quite technical in nature, a number of accessible survey papers are available there as well. Another good reference is the book "Bayesian Data Analysis" by Gelman *et alia* (1995). It provides an excellent overview of current approaches to Bayesian modeling and computation in applied statistics. This readable and accessible text is highly recommended.

2 Exact : Size of Loss Distributions for Exact Data

2.1 Introduction

This example will illustrate the Bayesian analysis using WinBUGS of several severity models applied to exact size of loss data. The particular models we target are the gamma, inverse gamma, loggamma, lognormal, (two-parameter) Pareto, inverse (two-parameter) Pareto, Weibull, and inverse Weibull distributions. Each of these will be applied to the size of loss data in Table 2.1. It should be possible to implement additional size of loss models, including those for truncated data, using methods analogous to those described below.

It turns out that BUGS / WinBUGS explicitly supports relatively few continuous size of loss models. These distributions include the beta, chi-squared, double exponential, exponential, gamma, normal, t, (single-parameter) Pareto, uniform, and Weibull. **Before using any one of these distributions, the practitioner would be wise to note its parametrization given in Table II of the *User Manual* in order to avoid any possibility of confusion.** The reader will note that several of our targeted models are not included in the list above, nor in Table II. Some of these can be built up from those available in BUGS / WinBUGS using mixtures of distributions and / or by applying a simple transformation (e.g., the inverse or logarithmic) to the data. Others can be implemented using the 'ones' trick described below in our discussion of the Pareto models.

2.2 Size of Loss Model Specification in WinBUGS

In this section we will review the definitions for our targeted models, and discuss how they may be coded in WinBUGS. It should be understood that the code can be ported over to 'classic' BUGS with little effort.

2.2.1 The Gamma, Inverse Gamma, and Loggamma Models

Let x denote an observed exact size of loss value. In WinBUGS, the declaration

```
x ~ dgamma( alpha, beta )
```

corresponds to a definition of the gamma model with density function

$$f(x|\ \alpha,\beta)\ =\ \frac{\beta^{\,\alpha}}{\Gamma(\alpha)}\,x^{\,\alpha-1}\exp(-\beta\,x)\ ,\ x\ >\ 0\ , \tag{2.1}$$

with $\alpha > 0$ and $\beta > 0$.

Consider what happens if we assign a gamma distribution as in (2.1) to the transformed variable $y = \log(x)$ instead. Then, in accordance with the statistical change of variable technique, we will have

$$g(x|\ \alpha,\beta)\ =\ f(y|\ \alpha,\beta)\,\frac{1}{x}\ =\ \frac{\beta^{\,\alpha}\,[\,\log(x)\,]^{\,\alpha-1}}{\Gamma(\alpha)\,x^{\,\beta+1}}\ ,\ x\ >\ 1\ , \tag{2.2}$$

with $\alpha > 0$ and $\beta > 0$. This is the definition of the density function for the loggamma model. In WinBUGS, the lines of code

```
y <- log( x )
y ~ dgamma( alpha, beta )
```

describe the loggamma model defined above. Specifically, the first line states the relationship between x and y and the second line assigns the relevant density type to y. The order of the two lines is actually immaterial to WinBUGS.

As before, let x denote the exact size of loss. This time assign a gamma distribution as in (2.1) to the transformed variable $y = x^{\,-1}$. Then, in accordance with the statistical change of variable technique, we have

$$h(x|\ \alpha,\beta)\ =\ f(y|\ \alpha,\beta)\,\frac{1}{x^{\,2}}\ =\ \frac{\beta^{\,\alpha}\exp(-\beta/x)}{\Gamma(\alpha)\,x^{\,\alpha+1}}\ ,\ x\ >\ 0\ , \tag{2.3}$$

with $\alpha > 0$ and $\beta > 0$. This is the definition of the density function for the inverse gamma model. In WinBUGS, the lines of code

```
y <- 1 / x
y ~ dgamma( alpha, beta )
```

describe the inverse gamma model.

All of the models described above will need to be completed by adding prior density specifications for the model parameters. Suppose that we are interested in modeling an

inverse gamma model to the twenty exact size of loss observations appearing in Table 2.1. Then our specification of a complete inverse gamma model in the BUGS language might proceed as shown below.

CODE FOR THE INVERSE GAMMA MODEL

```
model;
{
   # Compute the negative loglikelihood (NLL) in terms of x.
   NLL <- - sum( loglik[] )
   for( i in 1 : N ) {
      loglik[i] <- alpha * log( beta ) - loggam( alpha ) -
                   beta / x[i] - ( alpha + 1 ) * log( x[i] )
   }

   # Define the exact size of loss random variables.
   for( i in 1 : N ) {
      y[i] <- 1 / x[i]
      y[i] ~ dgamma( alpha, beta )
   }

   # Define 'naive' prior densities for the founder nodes.
     alpha ~ dgamma( 0.001, 0.001 )
     beta ~ dgamma( 0.001, 0.001 )

   # More informative priors, each with mean = mle and sd = 5 x mle.
   # See discussion below for more details.
   #
   # alpha ~ dgamma( aparm1, aparm2 )
   # beta ~ dgamma( bparm1, bparm2 )
   #
   # amle <- 0.5661338 ; aparm1 <- 0.04 ; aparm2 <- aparm1 / amle
   # bmle <- 193.6986  ; bparm1 <- 0.04 ; bparm2 <- bparm1 / bmle

}

DATA

list( N = 20,
      x = c( 59, 71, 127, 217, 223, 524, 537, 1089, 1127, 1181,
             1189, 1516, 1681, 1708, 1784, 3639, 5386, 6100, 9945, 15295 ) )
```

```
INITS

list( alpha = 2, beta = 2 )
```

Note that the deterministic node NLL represents the negative log of the likelihood function for the observed exact size of loss values. In the case of the inverse gamma model as above, it is constructed in accordance with the density given by (2.3). Its value depends on the unobserved model parameters, and the different values it takes on as the model is updated in WinBUGS can be monitored like those of any other node. Dempster (1974; reprinted in 1997) suggested that one might examine the posterior distribution of the loglikelihood to assist with model selection, and we will adopt this simple strategy by monitoring the value of NLL and favouring the model with the smallest posterior expected value for this variable. See Spiegelhalter, Best, and Carlin (1998) for modifications of this approach which are particularly useful when the models under consideration differ in complexity (the number of free parameters).

The prior density specifications assigned to the random parameters α and β in the sample code above are rather naive. It is difficult to believe that an experienced actuarial practitioner can say no more than that these parameters are distributed *a priori* identically and independently as gamma random variables with mean and variance of 1 and 1000, respectively. When all else fails, it may be reasonable - or at least, be not too objectionable from a pragmatic point of view - to assign each variable a gamma distribution *a priori* with mean and standard deviation equal to its maximum likelihood estimate (mle) and, say, 5 times its mle, respectively. Such a distribution is approximately 'centered' in the appropriate region yet is still quite widely spread out. Code implementing this second strategy is provided up above for illustration's sake, but is commented out. The mle values themselves were determined outside of WinBUGS using standard techniques.

Illustrative BUGS code for the various models described above appear in the file exact.odc available on this author's website at www.math.ucalgary.ca/∼scollnik/abcd/. The same is true for each of the models described in the following sections.

2.2.2 The Lognormal Model

In WinBUGS, the declaration

```
x ~ dnorm( mu, tau )
```

corresponds to a definition of the normal model with density function

$$f(x|\,\mu,\tau) \;=\; \frac{\sqrt{\tau}}{\sqrt{2\,\pi}}\exp\left[-\frac{\tau}{2}(x-\mu)^{\,2}\right]\;,\;\; -\infty \;<\; x \;<\; \infty\;, \tag{2.4}$$

with $-\infty < \mu < \infty$ and $\tau > 0$. In this parametrization, τ is called the precision or inverse variance parameter.

Consider what happens if we assign a normal distribution as in (2.4) to the transformed variable $y = \log(x)$. Then, in accordance with the statistical change of variable technique, we will have

$$g(x|\,\mu,\tau) \;=\; f(y|\,\mu,\tau)\,\frac{1}{x} \;=\; \frac{\sqrt{\tau}}{x\,\sqrt{2\,\pi}}\exp\left(-\frac{\tau}{2}\,[\,\log(x)-\mu\,]^{\,2}\right)\;,\; x \;>\; 0\;, \tag{2.5}$$

with $-\infty < \mu < \infty$ and $\tau > 0$. This is the definition of the density function for the lognormal model. In WinBUGS, the lines of code

```
y <- log( x )
y ~ dnorm( mu, tau )
```

describe the lognormal model defined above. Another way in which to define the same lognormal model is with the declaration

```
x ~ dlnorm( mu, tau )
```

This appears to work for all recent versions of WinBUGS, even though the `dlnorm` density has been left undocumented in Table II of the *User Manual* for some of them.

As usual, we still need to complete the model with a prior density specification and also define the data and initial values. Our complete model specification might proceed as shown below.

CODE FOR THE LOGNORMAL MODEL

```
model;
{
   # Compute the negative loglikelihood (NLL) in terms of x.
   NLL <- - sum( loglik[] )
   for( i in 1 : N ) {
      loglik[i] <- - log( sqrt( 2 * Pi / tau ) ) - log( x[i] ) -
                   pow( log( x[i] ) - mu, 2 ) * tau / 2
   }
   Pi <- 3.14159265

   # Define the exact size of loss random variables.
   for( i in 1 : N ) {
      y[i] <- log( x[i] )
      y[i] ~ dnorm( mu, tau )
   }

   # Define 'naive' prior densities for the founder nodes.
     mu ~ dnorm( 0, 0.001 )
     tau ~ dgamma( 0.001, 0.001 )

   # More informative priors, each with mean = mle and sd = 5 x mle.
   # See discussion below for more details.
   #
   # mu ~ dnorm( mparm1, mparm2 )
   # tau ~ dgamma( tparm1, tparm2 )
   #
   # mmle <- 6.936106 ; mparm1 <- mmle ; mparm2 <- 1 / pow( 5 * mmle, 2 )
   # tmle <- 0.432222 ; tparm1 <- 0.04 ; tparm2 <- tparm1 / tmle

}
```

DATA

```
list( N = 20,
      x = c( 59, 71, 127, 217, 223, 524, 537, 1089, 1127, 1181,
             1189, 1516, 1681, 1708, 1784, 3639, 5386, 6100, 9945, 15295 ) )
```

INITS

```
list( mu = 2, tau = 2 )
```

As before, we included a definition of the NLL in the code. Its values can be monitored and used to assist with model selection. Note that we adopted a prior normal distribution for the parameter μ (i.e., instead of a gamma distribution) as the support for this parameter is the entire real number line. Included for illustration's sake, but commented out, is a more informative prior density specification for each model parameter - as before, centered at that parameter's mle and with standard deviation equal to 5 times the mle.

2.2.3 The Weibull and Inverse Weibull Models

In WinBUGS, the declaration

```
x ~ dweib( tau, lambda )
```

corresponds to a definition of the Weibull model with density function

$$f(x|\ \tau, \lambda)\ =\ \tau\lambda x^{\tau-1} \exp\left(-\lambda x^{\tau}\right)\ ,\ x\ >\ 0\ , \tag{2.6}$$

with $\tau > 0$ and $\lambda > 0$.

Consider what happens if we assign a Weibull distribution as in (2.6) to the transformed variable $y = x^{-1}$. Then, in accordance with the statistical change of variable technique, we have

$$h(x|\ \alpha, \beta)\ =\ f(y|\ \alpha, \beta)\,\frac{1}{x^{\,2}}\ =\ \frac{\tau\lambda \exp(-\lambda/x^{\tau})}{x^{\tau+1}}\ ,\ x\ >\ 0\ , \tag{2.7}$$

with $\tau > 0$ and $\lambda > 0$. This is the definition of the density function for the inverse Weibull model. In WinBUGS, the lines of code

```
y <- 1 / x
y ~ dweib( tau, lambda )
```

describe the inverse Weibull model.

As usual, we still need to complete either model with a prior density specification and also define the data and initial values. We omit a presentation of either complete model specification as they are both very similar to those presented earlier in this section. As mentioned earlier, the code is available on this author's website.

2.2.4 The Pareto and Inverse Pareto Models

The discussion of the Pareto and inverse Pareto models have been left for last, as the tricks used to implement these models have more general application and deserve to be emphasized. It turns out that in WinBUGS, (beta) version 1.2 (May, 1999) or later, a version of the Pareto model is available with the declaration

```
x ~ dpar( alpha, theta )
```

However, this declaration corresponds to the single-parameter Pareto model with density function

$$f(x|\ \alpha,\theta)\ =\ \frac{\alpha\,\theta^{\,\alpha}}{x^{\,\alpha+1}}\ ,\ x\ >\ \theta\ ,$$

with $\alpha > 0$ and $\theta > 0$. This form of the Pareto distribution may be appropriate in certain instances, for example when modeling losses above a given deductible. This distribution is utilized in Section 4 of this paper in the discussion of the motor example.

As the data in Table 2.1 has no deductible associated with it, a more sensible version of the Pareto distribution for this context would be the two-parameter model with density function

$$f(x|\ \alpha,\theta)\ =\ \frac{\alpha\,\theta^{\,\alpha}}{(x+\theta)^{\,\alpha+1}}\ ,\ x\ >\ 0\ , \tag{2.8}$$

with $\alpha > 0$ and $\theta > 0$. A related distribution is the inverse Pareto model which arises in the expected manner by assigning a Pareto distribution as in (2.8) to the transformed variable $y = x^{-1}$. Then, in accordance with the statistical change of variable technique, we have

$$h(x|\ \alpha,\theta)\ =\ f(y|\ \alpha,\theta)\,\frac{1}{x^{\,2}}\ =\ \frac{\alpha\,\theta^{\,\alpha}\,x^{\,\alpha-1}}{(1+x\,\theta)^{\,\alpha+1}}\ ,\ x\ >\ 0\ , \tag{2.9}$$

with $\alpha > 0$ and $\theta > 0$. Although neither of these distributions is explicitly supported in WinBUGS, we are aware of two ways in which to implement them.

The first is based on a trick that was originally found on the FAQ (frequently asked questions) page of the BUGS website www.mrc-bsu.cam.ac.uk/bugs. This 'ones' trick now appears in Section 3.2 of the WinBUGS *User Manual* and its discussion there reads as follows:

Suppose your data is y (of length n) and you want to fit the model p(y) = f(y,t) where t are the unknown parameters and f is the formula of the density that is not currently handled by BUGS.

The trick is to create a new vector 'ones', that comprises just 1's and is of length n (note the use of the data transformation ability described in Section 3.7). Then use the BUGS code:

```
for(i in 1 : n) {
    ones[i] <- 1
    ones[i] ~ dbern( p[i] )
    p[i] <- f(y[i],t) / K
}
```

where K is a sufficiently large constant to ensure that all sampled values of p[i] are less than one. This should provide a likelihood term proportional to f(y,t).

To illustrate, in the case of a random sample from the two-parameter Pareto model, with density function (2.8), we would assign

```
p[i] <- alpha * pow( theta, alpha ) / pow( x[i] + theta, alpha + 1 )
```

When using the inverse Pareto model, with density function (2.9), we would use the lines of code

```
y[i] <- 1 / x[i]
p[i] <- alpha * pow( theta, alpha ) / pow( y[i] + theta, alpha + 1 ) /
        pow( x[i], 2 )
```

It should be apparent to the reader that this 'ones' trick can be used to construct the likelihood function for a sample drawn from any continuous distribution, including truncated models, provided that the relevant density function may be expressed using the operators $+$, $-$, $*$, $/$, and the standard mathematical functions (e.g., *exp*, *log*, *abs*, and *sqrt*) listed in Table I of the WinBUGS *User Manual*. Incidentally, as the likelihood function for the observed data is the product of the `p[i]` terms, it will be an easy matter to calculate the node NLL as it is simply the negative logarithm of this product.

The second method available with which to implement the two-parameter Pareto and inverse Pareto models is actually just an application of the observation that a two-parameter Pareto random variable can be defined as a mixture of two gamma random variables (Hogg and Klugman, 1984, page 54). Specifically, if the distribution of x given τ is gamma(1, τ) and the distribution of τ given α and θ is gamma(α, θ), then the distribution of x given α and θ has the density function (2.8). To code this relationship in WinBUGS, we would use the lines of code

```
x[i] ~ dgamma( 1, tau[i] )
tau[i] ~ dgamma( alpha, theta )
```

It is important to note that each observation requires its own mixing parameter `tau[i]` (see Section 2.7.3.4 of Klugman *et alia*, 1998, for a further discussion of this point and of mixture modeling in general). The lines

```
y[i] <- 1 / x[i]
y[i] ~ dgamma( 1, tau[i] )
tau[i] ~ dgamma( alpha, theta )
```

serve to define the inverse Pareto model. Other distributions with interpretations as mixture models may be implemented in an analogous manner. However, although hard and fast advice is difficult to give, our experience suggests that the 'ones' trick leads to complete model specifications which update more quickly, and also take fewer updates to converge, in WinBUGS. The mixture modeling approach is still valuable, though, as it may be used to generate posterior predictive draws from the two-parameter Pareto models described above. This is discussed below.

No matter which of the two methods we adopt, we still need to complete the model with a prior density specification and also define the data and initial values in the usual way. Illustrative code for a complete model specification appears in the file exact.odc available on this author's website at www.math.ucalgary.ca/∼scollnik/abcd/.

2.3 Implementing Posterior Predictive Draws and Model Checks

The preceding discussion described how a variety of size of loss models can be implemented in WinBUGS. By examining the values of the NLL node associated with each model, selection between competing models is facilitated. That is to say, models with low values of the NLL are generally preferred, *ceteris paribus*. Although examination of the values taken on by the NLL node will provide some guidance as to how well a particular model fits a given data set, it does not tell the complete story. Model checking is also important and is discussed in the text by Gelman, Carlin, Stern, and Rubin (1995, especially Chapters 6 and 18). One method presented by these authors, that of posterior predictive checks, involves drawing simulated values from the posterior predictive distribution of replicated data and comparing these samples to the observed data (Gelman, *et alia*, 1995, pages 162-174). Systematic differences between the simulations and observed data indicate potential failings of the model.

The method of posterior predictive checks is fairly simple to implement using WinBUGS. The first step is to generate a *replicated* sample from the same model (i.e., from the same distribution and with the same model parameter values) that is assumed to have generated the observations at hand. The replicated sample is of the same size as the original and would utilize the identical covariate values if the model happened to contain explanatory variables. Often, this replicated sample is easily obtained by essentially duplicating the code used to model the original observations. For example, suppose we were assuming a loggamma model as in (2.2) for the data in Table 2.1 and so had specified

```
y[i] <- log( x[i] )
y[i] ~ dgamma( alpha, beta )
```

Then the replicated data would be defined analogously with the lines

```
x.rep[i] <- exp( y.rep[i] )
y.rep[i] ~ dgamma( alpha, beta )
```

Note that the transformations are now coded from `y.rep[i]` to `x.rep[i]` as the former is logically defined in advance of the latter. The same idea works for all of the distributions

previously discussed except the Pareto and inverse Pareto. As these two are not explicitly supported in WinBUGS, we utilize the mixture model interpretation of the Pareto distribution in order to generate the predictive draws. In the case of the Pareto model with density function (2.8), this is accomplished with the lines of code

```
x.rep[i] ~ dgamma( 1, tau.rep[i] )
tau.rep[i] ~ dgamma( alpha, theta )
```

whereas the code segment

```
x.rep[i] <- 1 / y.rep[i]
y.rep[i] ~ dgamma( 1, tau.rep[i] )
tau.rep[i] ~ dgamma( alpha, theta )
```

would be appropriate if we were assuming the inverse Pareto model with density function (2.9).

The next step is to compare the simulated values from the posterior predictive distribution to the observed data. This may be accomplished using *graphical summaries* or through the use of *test quantities*. Here, we will briefly describe the latter approach and direct the reader to Figures 6.3-6.5, 13.2, and 16.2-16.3 in Gelman *et alia* (1995) for examples of the former. In any case, the reader is once again referred to Gelman *et alia* (1995, pages 162-174) for a more extensive discussion of posterior predictive model checking.

Let x be the observed data, let θ be the vector of unknown model parameters, and let x^{rep} be the replicated data as defined above that might have been observed if a new sample of observations were sampled from the same distribution and with the same model parameter values used to generate x. A test quantity, also called a *discrepancy measure*, $T(x, \theta)$ is a scalar summary of the parameters and data that is used as a standard when comparing the observed data to the replicate simulations. The possibilities include, but are certainly not limited to, $T(x, \theta) = \min(x_i)$, $T(x, \theta) = \sum x_i$, and $T(x, \theta) = \bar{x} - E(X_i|\, \theta)$. Test quantities are suggested by the problem context, and some examples are considered below. Any given discrepancy measure can also be calculated using the posterior simulations of (x^{rep}, θ) in order to obtain values we denote $T(x^{rep}, \theta)$.

The Bayesian posterior predictive p-value is defined as the probability that the replicated data could be more extreme than the observed data, as measured by the test quantity and given the assumed model. Mathematically, we write

$$\text{Bayes } p\text{-value} \quad = \quad \Pr(\, T(x^{rep}, \theta) \geq T(x, \theta) |\, x \,) \;, \tag{2.10}$$

with the probability understood to be taken over the joint posterior distribution of (x^{rep}, θ).

When the tail-area probability (2.10) is close to 0 or 1 for some meaningful test quantity, the assumed model is suspect. In this case, the definition of the discrepancy measure might suggest how the model can be improved. For example, suppose $T(x, \theta) = \max(x_i)$ and the Bayes p-value is approximately 0.84. This says that nearly 17 times out of 20 the assumed model will generate a predictive sample containing a maximum value greater than that observed in the original sample. The practitioner will have to decide whether or not this is a crucial model failing, given the problem context. It needn't be, say, if the practitioner's real interest is in developing inferences with respect to the distribution of total future claims and the test quantity $T(x, \theta) = \sum x_i$ happens to yield a Bayes p-value close to 0.50. But if it is judged to be a crucial failing, the practitioner might try a model with a thinner tail. As an alternative course of action, the practitioner may keep the assumed model for the original sample but impose a reasonable *a priori* upper bound on each predictive draw. See Gelman *et alia* (1995, pages 463-468) for an example of this sort.

When inference is proceeding on the basis of a Markov chain Monte Carlo (MCMC) simulation, as with BUGS / WinBUGS, it is easy to estimate the Bayes p-value by monitoring the values taken on by an indicator variable assigned equal to 1 when $T(x^{rep}, \theta) \geq T(x, \theta)$, and 0 otherwise. The average of these values is an estimate of (2.10). In the particular case of the exact size of losses in Table 2.1, it may make sense to monitor the minimum, maximum, and total losses in each of the replicated data sets. The BUGS code following below could be used to implement the appropriate posterior predictive checks. The approximate Bayes p-values are equal to the estimated posterior means of the nodes `p.repmin`, `p.repmax`, and `p.repsum`.

```
CODE ILLUSTRATING THE IMPLEMENTATION OF POSTERIOR PREDICTIVE CHECKS

# Define indicator variables with which to estimate the Bayes p-values.

p.repmin <- step( x.repmin - x.min )  # 1 if x.repmin >= x.min
p.repmax <- step( x.repmax - x.max )  # 1 if x.repmax >= x.max
p.repsum <- step( x.repsum - x.sum )  # 1 if x.repsum >= x.sum

# Calculate the minimum, maximum, and total of the observed data.

x.min <- ranked( x[], 1 )
x.max <- ranked( x[], N )
x.sum <- sum( x[] )

# Calculate the minimum, maximum, and total of the replicated data.

x.repmin <- ranked( x.rep[], 1 )
x.repmax <- ranked( x.rep[], N )
x.repsum <- sum( x.rep[] )
```

2.4 Fitting the Models to the Data in Table 2.1

Finally, we are ready to apply the models and methods discussed in this section to the exact size of loss data in Table 2.1. For this illustration, in each case we have assumed independent prior distributions for all model parameters. Postive model parameters were assigned prior gamma distributions and the lognormal model's real parameter μ was assigned a normal prior distribution. Each model parameter had its prior distribution assigned a mean and standard deviation equal to its mle and 5 times its mle, respectively. These distributions are clearly informative, but we would argue only very weakly so. In practice, the actuarial practitioner will often be able to ascertain more informative prior distributions than these from past experience.

Each model compiled readily in WinBUGS and updated fairly quickly. The loggamma model was typical of the majority, and took 3 seconds to burn-in for 5000 updates, and then 25 seconds to run for an additional 20,000 iterations on a dual 200 MHz Pentium Pro PC. The Pareto and inverse Pareto models were slowest and each took about twice as long to run

as the others. Summary statistics for the eight models appear in Table 2.2. The estimates from WinBUGS are based on the final 20,000 of the 25,000 iterations performed for each model. On the basis of the summary statistics for the NLL node, the lognormal and Pareto models rank as our first and second choices. The posterior predictive checks we monitored give us no reason to suspect either model.

It is important to note that WinBUGS will always output estimated posterior means and SDs for the nodes `x.repmin`, `x.repmax`, and `x.repsum` using sample moment calculations applied to the 20,000 simulated values of each, even though the corresponding theoretical posterior predictive moments may not exist under the assumed model. In these cases, the posterior mean and SD estimates should be ignored. If it is believed *a priori* that certain predictive moments do exist, then the model parameters should be constrained appropriately. In the case of the Pareto model, for example, the restrictions $\alpha > 1$ and $\alpha > 2$ would need to be imposed in order to ensure the existence of a finite posterior predictive mean and variance, respectively (Klugman *et alia*, 1998, page 575). The posterior probability attached to these restrictions can be checked by monitoring the frequency with which they arise in the MCMC simulation-based analysis of the unconstrained model. This procedure is illustrated in the analysis of the motor example in Section 4 of this paper, and in the analysis of the grouped example ("Modeling Grouped Size of Loss Data in WinBUGS") in Scollnik (1999).

2.5 Implementing Predictive Inference

Suppose that $f(x|\,\psi)$ is the loss model responsible for generating the original observed losses, and that $g(y|\,\psi)$ is the loss model responsible for generating the losses that will be observed in the next period. Given the model parameters, ψ, we assume that the past and future losses are all independent of one another. The predictive density $h(y|\,Obs.\ Data\,)$ associated with a future loss is defined as the theoretical average of $g(y|\,\psi)$ taken with respect to the posterior distribution of the model parameters. That is,

$$h(y|\,Obs.\ Data\,) = \int g(y|\,\psi)\,p(\psi|\,Obs.\ Data\,)\,\mathrm{d}\psi\ . \tag{2.11}$$

We have already discussed how to simulate a dependent sequence of random draws from a posterior distribution of model parameters, like $p(\psi|\, Obs.\ Data\ )$, using WinBUGS. Let us assume that WinBUGS has been used in this manner to generate a sequence of such draws, which we will denote as $\psi^{(t)}$, for $t = m, \ldots, n$ (m = 5,001 and n = 25,000, in the example above). Provided that the model parameter vector ψ was monitored in WinBUGS over these $n-m+1$ iterations, we can click the *Coda* button on the *Sample Monitor Tool* dialog box to dump an ascii representation of its simulated values. These can be read into a spreadsheet or mathematical / statistical package and then used to estimate (2.11) on the basis of the ergodic sample average

$$h(y|\, Obs.\ Data\ ) \;\approx\; \frac{1}{n-m+1} \sum_{t=m}^{n} g(y|\, \psi^{(t)}) \ . \tag{2.12}$$

This is easily evaluated for any value(s) of y desired. Note that the conditional model $g(y|\, \psi)$ needn't be identical to the model $f(x|\, \psi)$ responsible for generating the original observed losses. In particular, it may be modified in accordance with the effect(s) of inflation and / or policy limit modifications. For instance, if the original model was

$$f(x|\, \alpha, \theta) \;\sim\; \text{Pareto}(\alpha, \theta)$$

and inflation through the next period was r percent, then the conditional loss model at the end of this period would be

$$g(y|\, \alpha, \theta) \;\sim\; \text{Pareto}(\alpha, [1 + \text{r}]\, \theta) \ ,$$

as noted in Table 5.1 of Hogg and Klugman (1984, page 180).

Finally, if we wanted to simulate a variable representing a predictive draw from the loss model, we can implement this with lines of code reminiscent of Section 2.3. For the Pareto loss model with inflation as described in the paragraph above, for example, we would code

```
y ~ dgamma( 1, tau.y )
tau.y ~ dgamma( alpha, theta.y )
theta.y <- ( 1 + r ) * theta
```

The value of r would be set as a constant, loaded in as part of the data list.

Table 2.1

Twenty Exact Size of Losses.

59	71	127	217
223	524	537	1,089
1,127	1,181	1,189	1,516
1,681	1,708	1,784	3,639
5,386	6,100	9,945	15,295

Table 2.2

Estimated Posterior Summary Statistics.

		Estimates from WinBUGS				
Model	Parameter	Mean	SD	2.5%	Median	97.5%
gamma	NLL	177.3	1.03	176.3	177.0	180.1
	alpha	0.6241	0.1699	0.341	0.6075	0.9989
	beta	2.35E-4	9.23E-5	8.72E-5	2.23E-4	4.45E-4
	p.repmin	0.3686	0.4824	0.0	0.0	1.0
	p.repmax	0.1905	0.3927	0.0	0.0	1.0
	p.repsum	0.4953	0.5	0.0	0.0	1.0
	x.repmin	82.57	131.9	0.02948	31.4	466.0
	x.repmax	11160.0	7002.0	3519.0	9392.0	2.9E+4
	x.repsum	58390.0	26510.0	22660.0	53140.0	123200.0
inverse	NLL	179.0	1.045	178.0	178.7	181.8
gamma †	alpha	0.5504	0.1476	0.3059	0.536	0.8788
	beta	188.0	75.82	67.81	178.5	359.8
	p.repmin	0.7805	0.4139	0.0	1.0	1.0
	p.repmax	0.7222	0.4479	0.0	1.0	1.0
	p.repsum	0.7859	0.4102	0.0	1.0	1.0
	x.repmin	106.0	63.92	27.7	92.03	263.1
	x.repmax	9.8E+11	9.9E+13	2482.0	50780.0	1.117E+8
	x.repsum	6.4E+13	6.9E+15	16160.0	214500.0	6.082E+8

† Note the discussion in the main text concerning the existence of the theoretical posterior predictive moments for the nodes `x.repmin`, `x.repmax`, and `x.repsum`.

Table 2.2 (continued)

Estimated Posterior Summary Statistics.

		Estimates from WinBUGS				
Model	Parameter	Mean	SD	2.5%	Median	97.5%
loggamma †	NLL	177.0	0.9604	176.0	176.7	179.6
	alpha	18.52	5.826	9.211	17.72	31.67
	beta	2.669	0.8512	1.311	2.553	4.575
	p.repmin	0.7228	0.4476	0.0	1.0	1.0
	p.repmax	0.5332	0.4989	0.0	1.0	1.0
	p.repsum	0.6401	0.48	0.0	1.0	1.0
	x.repmin	117.2	91.01	18.08	93.77	352.6
	x.repmax	175700.0	4.432E+6	2672.0	17010.0	590800.0
	x.repsum	608500.0	2.988E+7	17600.0	73250.0	1.432E+6
lognormal	NLL	176.5	1.017	175.5	176.2	179.2
	mu	6.933	0.365	6.22	6.934	7.66
	tau	0.4105	0.1335	0.1969	0.3956	0.712
	p.repmin	0.6247	0.4842	0.0	1.0	1.0
	p.repmax	0.4294	0.495	0.0	0.0	1.0
	p.repsum	0.5647	0.4958	0.0	1.0	1.0
	x.repmin	109.6	102.0	6.599	81.54	380.4
	x.repmax	32870.0	161600.0	2764.0	12900.0	159100.0
	x.repsum	1.02E+5	310700.0	18770.0	59560.0	397900.0

† Note the discussion in the main text concerning the existence of the theoretical posterior predictive moments for the nodes `x.repmin`, `x.repmax`, and `x.repsum`.

Table 2.2 (continued)

Estimated Posterior Summary Statistics.

		Estimates from WinBUGS				
Model	Parameter	Mean	SD	2.5%	Median	97.5%
Pareto †	NLL	176.7	0.9122	175.7	176.4	179.1
	alpha	3.484	4.27	0.6302	2.098	15.65
	theta	6827.0	10740.0	453.5	3182.0	38770.0
	p.repmin	0.6236	0.4845	0.0	1.0	1.0
	p.repmax	0.3059	0.4608	0.0	0.0	1.0
	p.repsum	0.5065	0.5	0.0	1.0	1.0
	x.repmin	126.8	132.1	3.411	85.76	481.9
	x.repmax	3.585E+7	3.827E+9	2919.0	9876.0	236600.0
	x.repsum	4.725E+7	4.06E+9	20910.0	53880.0	683400.0
inverse	NLL	177.1	0.987	176.1	176.8	179.7
Pareto †	alpha	1.536	1.166	0.4978	1.231	4.387
	theta	0.002069	0.00274	2.69E-4	0.001299	0.008517
	p.repmin	0.593	0.4913	0.0	1.0	1.0
	p.repmax	0.5238	0.4994	0.0	1.0	1.0
	p.repsum	0.6594	0.4739	0.0	1.0	1.0
	x.repmin	106.7	104.6	1.152	78.39	374.3
	x.repmax	96650.0	1.412E+6	2703.0	16390.0	406200.0
	x.repsum	219100.0	1.668E+6	18490.0	74010.0	962200.0

† Note the discussion in the main text concerning the existence of the theoretical posterior predictive moments for the nodes `x.repmin`, `x.repmax`, and `x.repsum`.

Table 2.2 (continued)

Estimated Posterior Summary Statistics.

		Estimates from WinBUGS				
Model	Parameter	Mean	SD	2.5%	Median	97.5%
Weibull	NLL	176.8	0.9955	175.8	176.5	179.5
	alpha	0.7236	0.1226	0.4974	0.7195	0.9698
	beta	0.006146	0.006806	4.528E-4	0.003937	0.02473
	p.repmin	0.4091	0.4917	0.0	0.0	1.0
	p.repmax	0.2338	0.4232	0.0	0.0	1.0
	p.repsum	0.4978	0.5	0.0	0.0	1.0
	x.repmin	84.74	120.3	0.1849	40.89	421.9
	x.repmax	12590.0	11780.0	3277.0	9712.0	38800.0
	x.repsum	60370.0	35610.0	21760.0	53260.0	1.41E+5
inverse	NLL	178.1	1.002	177.2	177.8	180.9
Weibull †	tau	0.6671	0.108	0.4647	0.6645	0.8875
	lambda	71.95	51.04	17.23	59.21	202.9
	p.repmin	0.745	0.4359	0.0	1.0	1.0
	p.repmax	0.6937	0.4609	0.0	1.0	1.0
	p.repsum	0.7718	0.4197	0.0	1.0	1.0
	x.repmin	109.5	75.39	20.96	92.0	298.4
	x.repmax	1.796E+8	1.314E+10	2719.0	34900.0	9.502E+6
	x.repsum	8.665E+9	1.161E+12	17700.0	143100.0	3.507E+7

† Note the discussion in the main text concerning the existence of the theoretical posterior predictive moments for the nodes `x.repmin`, `x.repmax`, and `x.repsum`.

3 Bivreg : Regression Models for Bivariate Loss Data

3.1 Introduction

This example illustrates the Bayesian analysis of two possible regression models for bivariate claims data. The context and data are borrowed from Section 2.11.3 ("Bivariate Models") in Klugman *et alia* (1998). In casualty insurance, the allocated loss adjustment expenses (ALAE) are directly related to the payment of the loss itself. Loss and ALAE values for twenty-four such claims are given in Table 3.1. Our problem is twofold. First, we want to consider how a fully Bayesian analysis of a regression loss model can be implemented in WinBUGS. Second, we want to estimate the predictive total loss distribution under each of the following two situations: (1) a policy limit of 50,000 and an assumption that the ALAE is related to the actual size of loss, not the amount paid; (2) a per loss reinsurance where the reinsurer pays the excess of each loss over 50,000 with a maximum of 50,000 per loss, with any ALAE paid in the same proportion as is paid on the loss.

Let's begin the discussion by considering how we might proceed if we were not operating in a Bayesian framework. Let X and Y be the loss and ALAE random variables, respectively, and let ψ represent the complete vector of model parameters for their joint distribution, which we denote as $f(x, y|\,\psi)$. If we adopt a classical parametric estimation approach (e.g, like maximum likelihood, method of moments, or minimum distance) in order to obtain a point estimate $\widehat{\psi}$ based on a random sample, then our fitted model for future losses and ALAEs (barring effects like inflation) would be $f(x, y|\,\widehat{\psi})$. A transformation of variable corresponding to the policy coverage under consideration can be applied to this bivariate model in order to obtain the univariate model describing the future total loss. This theoretical derivation can be difficult to implement in practice. An alternative is to use simulation. Specifically, generate many draws of pairs (x, y) from $f(x, y|\,\widehat{\psi})$, and apply the terms of coverage under examination to each. Under coverage (1), the total loss is equal to $x_{\,cov\,1} + y_{\,cov\,1}$ with

$$\begin{aligned} x_{\,cov\,1} &= \min(\,x,\,50,000\,) \\ y_{\,cov\,1} &= y\ . \end{aligned}$$

Under coverage (2), the total loss is equal to $x_{\,cov\,2} + y_{\,cov\,2}$ with

$$\begin{aligned} x_{\,cov\,2} &= \min(\,\max(\,0,\,x - 50,000\,),\,50,000\,) \\ y_{\,cov\,2} &= \left(\,\frac{x_{\,cov\,2}}{x}\,\right)\,y\ . \end{aligned}$$

Once these simulated total losses are determined, use them to develop the inferences of interest. Quantile estimates of the distribution of future total losses can be obtained as the corresponding empirical quantiles from the simulated data. An empirical central $100(1-\alpha)\%$ interval estimate for the total loss can be obtained taking the endpoints of this interval equal to the empirical $100(\alpha/2)\%$ and $100(1-\alpha/2)\%$ points for the set of simulated total loss values (set $\alpha = 0.05$ for an empirical 95% interval estimate). The expected value of the total loss, assuming that this expected value exists and is finite, can be estimated using the average loss observed for the simulated data. An approximate 95% confidence interval for this expected value can be obtained using the method described in Klugman *et alia* (1998, page 151). Specifically, calculate the sample mean and standard deviation of the simulated total losses and then take the limits of the confidence interval equal to the sample mean plus / minus twice the sample standard deviation divided by the square root of the number of simulated draws. The reader should note that a 95% confidence interval for the expected value of the total loss will be significantly tighter than the empirical 95% interval estimate of the total loss. A value at risk summary statistic is usually defined as the upper bound of the latter type of interval, not that of the former.

In any case, all of the inferences discussed in the paragraph above are conditional upon the value of $\widehat{\psi}$ being the 'correct' value of the model parameters. This ignores inherent parameter uncertainty and may lead to inferences which can understate the true risk. One way in which to address this problem is to adopt a fully Bayesian approach and direct attention to the posterior predictive distribution of future losses and ALAEs given the observed data. The predictive distribution is defined as the theoretical average of $f(x, y|\,\psi)$ taken with respect to the posterior distribution of the model parameters given the observed data, that is,

$$f(x, y|\,Obs.\ Data\ ) = \int f(x, y|\,\psi)\,p(\psi|\,Obs.\ Data\ )\,\mathrm{d}\psi\ . \tag{3.1}$$

In this definition, we made the usual assumption that the future and past observations are independent of one another, given the model parameters. If this is not the case, replace $f(x, y|\,\psi)$ in (3.1) with $f(x, y|\,Obs.\ Data\ , \psi)$. The posterior predictive distribution accounts for parameter uncertainty by assigning more or less weight through the posterior to those values of ψ that are more or less probable in light of the data, respectively. See Dickson, Tedesco, and Zehnwith (1998) (and the motor example in Section 4 of this report) for further discussion and references concerning the ability of the Bayesian predictive distribution to incorporate parameter uncertainty into the modeling process.

3.2 Two Regression Models for Bivariate Loss Data

We now describe two regression models for the bivariate data in Table 3.1. We begin our description of the first by assuming a Pareto distribution for X with parameters α and θ, and a conditional gamma distribution for Y with parameters δ and $\lambda_x = \exp(\gamma + \beta \log x)$. The joint density function for this model is

$$f(x, y|\,\psi) \;=\; \frac{\alpha\,\theta^{\alpha}}{(\,x\,+\,\theta\,)^{\alpha+1}}\;\frac{\lambda_x^{\delta}\;y^{\delta-1}}{\Gamma(\delta)\;e^{y\,\lambda_x}}\;, \tag{3.2}$$

with $\psi = (\alpha, \theta, \delta, \gamma, \beta)$. The parameters α, θ, and δ must be positive whereas γ and β can be positive or negative valued. Note that the conditional expectation $E(Y \,|\, X = x, \psi)$ always exists and has value $\delta \,/\, \lambda_x$. From the data in Table 3.1, the maximum likelihood estimates (determined using standard methods outside of WinBUGS) are $\widehat{\alpha} = 2.4461$, $\widehat{\theta} = 32,248.8$, $\widehat{\delta} = 0.4497009$, $\widehat{\gamma} = -7.439761$, and $\widehat{\beta} = -0.2058762$. The value of the negative log likelihood (NLL) evaluated at the maximum likelihood estimates is 483.5047.

In a MCMC-based analysis, experience has shown that it is often beneficial to center the covariates in order to speed convergence of the simulation. In this case, the model stated in (3.2) would be modified so that the conditional distribution of Y is gamma with parameters δ and $\tilde{\lambda}_x = \exp(\gamma + \beta\,[\,\log x - k\,]\,)$, where k is set equal to the observed average value of $\log x$. The joint density function for this model is given by (3.2) with λ_x replaced by $\widetilde{\lambda}_x$. For the data in Table 3.1, we have $k = 9.293716$. The maximum likelihood estimates are

unchanged for the parameters α, θ, δ, and β, and that for γ is shifted to $\widehat{\gamma} = -7.439761 + (-0.2058762)\, k = -9.353116$. The minimum value of the NLL is unaffected by this or any reparametrization.

The second model we consider will assume a Pareto distribution for X with parameters α and θ as before, with a conditional Pareto distribution for Y with parameters δ and $\lambda_x = \exp(\gamma + \beta \log x)$. The joint density function for this model is

$$f(x, y|\, \psi) \;=\; \frac{\alpha\, \theta^{\,\alpha}}{(\, x \,+\, \theta\,)^{\,\alpha+1}} \; \frac{\delta\, \lambda_x^{\,\delta}}{(\, y \,+\, \lambda_x\,)^{\,\delta+1}} \;, \tag{3.3}$$

with $\psi = (\alpha, \theta, \delta, \gamma, \beta)$. The parameters α, θ, and δ must be positive whereas γ and β can be positive or negative valued. This model is presented in Klugman *et alia* (1988, page 153). From the data in Table 3.1, the maximum likelihood estimates (determined using standard methods outside of WinBUGS) are $\widehat{\alpha} = 2.44612$, $\widehat{\theta} = 32,248.86$, $\widehat{\delta} = 0.8484348$, $\widehat{\gamma} = 1.106727$, and $\widehat{\beta} = 0.6200453$. The value of the NLL evaluated at the maximum likelihood estimates is 482.59. Note that the conditional expectation of $E(Y \,|\, X = x, \psi)$ does not exist when $\delta \leq 1$. Otherwise, it is equal to $\lambda_x \,/\, (\delta - 1)$.

The centered version of the model stated in (3.3) would be modified so that the conditional distribution of Y is Pareto with parameters δ and $\widetilde{\lambda}_x = \exp(\gamma + \beta\,[\,\log x - k\,]\,)$, with k set equal to the observed average value of $\log\, x$ as before. The joint density function for this model is given by (3.3) with λ_x replaced by $\widetilde{\lambda}_x$. For the data in Table 3.1, recall that $k = 9.293716$. The maximum likelihood estimates are unchanged for the parameters α, θ, δ, and β, and that for γ is shifted to $\widehat{\gamma} = 1.106727 + 0.6200453\, k = 6.869252$. The minimum value of the NLL remains the same.

For both the Pareto-gamma and Pareto-Pareto models, the value of the maximum likelihood estimate $\widehat{\alpha} = 2.4461$ suggests that the first two moments of X exist. However, consider testing the null hypothesis that the Pareto model for X has $\alpha = 1$ against the unrestricted alternative. Under this null hypothesis, the restricted maximum likelihood estimate of the remaining Pareto model parameter for X is $\widehat{\theta} = 10{,}554.15$ and the minimum value of the NLL is 262.9672. Under the alternative, the unrestricted maximum likelihood estimates are $\widehat{\alpha} = 2.4461$ and $\widehat{\theta} = 32,248.8$, and the corresponding value of the NLL is 261.4931. The

likelihood ratio test statistic defined in Theorm 2.20 of Klugman *et alia* (1998, page 81) can now be calculated as W = 2 (262.9672 - 261.4931) = 2.9482. As the null hypothesis has one restriction, we compare the value of the test statistic to a critical value based on the chi-square distribution with one degree of freedom. At a 5% significance level, the critical value is 3.8414 and so we cannot reject the null hypothesis. The p-value is 0.0860. The implication is that predictive inferences conditioned on the value $\widehat{\alpha} = 2.4461$ may be subject to significantly less variability than warranted.

3.3 Regression Model Specification in WinBUGS

In this section we will discuss how the targeted regression models may be coded in WinBUGS. It should be understood that the code can be ported over to 'classic' BUGS with little effort. Note, that as the two-parameter Pareto model is not explicitly supported in BUGS / WinBUGS, we will make use of the 'ones' trick found in Section 3.2 of the WinBUGS *User Manual* to model this distribution. The 'ones' trick was previously used to implement the Pareto model in the context of the exact example in Section 2.2.4 of this report. The code below implements the Pareto-Pareto model, and is available in the file bivregpp.odc available on this author's website at www.math.ucalgary.ca/∼scollnik/abcd/. Very similarly styled code for the Pareto-gamma model is available in the file bivregpg.odc.

Comments interspersed throughout the code below hopefully clarify some of its trickier parts. The code begins by defining the Pareto-Pareto model generating the observed sample with the 'ones' trick as discussed above. The NLL for the observed data is then evaluated using the current model parameter values. See Section 2.2.1 for a discussion concerning the use of the NLL in model selection. Independent prior distributions are next assigned to the remaining model parameters. For illustration's sake, we assign gamma distributions to positive parameters and normal distributions to parameters that can take on positive or negative values. Each model parameter has its prior distribution assigned a mean and standard deviation equal to its mle and 3 times its mle, respectively. Although informative, these priors are still fairly spread out. In practice, the actuarial practitioner will often be able

to ascertain more informative prior distributions than these from past experience. Near the end of the file, the nodes `eofx.exist` and `eofy.exist` are defined. These can be monitored in order to observe the frequency that $\alpha \geq 1$ and $\delta \geq 1$, respectively. If one of these posterior probabilities is sufficiently large, then the practitioner may be willing to impose the strict 'greater than' constraint directly using the `I( 1, )` construct in the specification of the corresponding model parameter's prior density. See Section 2.4 of this report for more information and an illustration of this procedure.

CODE FOR THE PARETO-PARETO MODEL

```
model;
{

   # The regression model assumption we make in this file is that :
   #
   # (1)  x[i] ~ Pareto( alpha, theta )
   #
   # and either
   #
   # (2a)  y[i] | x[i] ~ Pareto( delta, lambda[i] )
   #
   #       lambda[i] <- exp( gamma + beta * log( x[i] ))
   #
   # or
   #
   # (2b)  y[i] | x[i] ~ Pareto( delta, lambda[i] )
   #
   #       lambda[i] <- exp( gamma + beta * ( log( x[i] - mean(logx[]) ))))
   #
   # The difference between (2a) and (2b) is that the covariate values of
   # log( x[i] ) are centered in (2b). This speeds convergence of the MCMC
   # simulation.

   # Define the Pareto-Pareto model.
   # Assign center = 1 for model (2b), and center = 0 for model (2a).

   center <- 1
```

```
for( i in 1 : N ) {

   # Model y[i] using the 'ones' trick. Note the use of 'center' below.
   ones.y[i] <- 1
   ones.y[i] ~ dbern( p.y[i] )
   p.y[i] <- delta * pow( lambda[i] / ( y[i] + lambda[i] ), delta ) /
             ( y[i] + lambda[i] )
   log( lambda[i] ) <- gamma + beta * ( logx[i] - center * mean( logx[] ))
   logx[i] <- log( x[i] )

   # Model x[i] using the 'ones' trick.
   ones.x[i] <- 1
   ones.x[i] ~ dbern( p.x[i] )
   p.x[i] <- alpha * pow( theta / ( x[i] + theta ) , alpha ) /
             ( x[i] + theta )

}

# Define the NLL.

NLL <- - sum( loglik[] )
for( i in 1 : N ) {
   loglik[i] <- log( p.x[i] ) + log( p.y[i] )
}

# Define mildly informative priors, 'centered' at each parameter's mle
# and with sd = 'numsd' x mle.

numsd <- 3
parm <- 1 / pow( numsd, 2 )

alpha ~ dgamma( parm, aparm )
aparm <- parm / 2.44612

theta ~ dgamma( parm, tparm )
tparm <- parm / 32248.86

delta ~ dgamma( parm, dparm )
dparm <- parm / 0.8484348
```

```
   gamma ~ dnorm( gparm1, gparm2 )
   gparm1 <- ( 1 - center ) * 1.106727 + center * 6.869252
   gparm2 <- 1 / pow( numsd * gparm1, 2 )

   beta ~ dnorm( bparm1, bparm2 )
   bparm1 <- 0.6200453
   bparm2 <- 1 / pow( numsd * bparm1, 2 )

   # Monitor two statistics of possible interest.

   eofx.exist <- step( alpha - 1 )
   eofy.exist <- step( delta - 1 )

}

DATA

list( N=24,
      x = c( 1500,  2000,  2500,  2500,  4500,  5000,  5750,  7000,
             7000,  7500,  9000,  10000, 11750, 12500, 14000, 14750,
             15000, 17500, 19833, 30000, 33033, 44887, 62500, 210000 ),

      y = c( 301,   3043,  415,   4940,  395,   25,    34474, 50,
             10593, 50,    406,   1174,  2530,  165,   175,   28217,
             2072 , 6328,  212,   2172,  7845,  2178,  12251, 7357 ))


INITS

# Possible starting values.

list( alpha = 8, theta = 40000, delta = 1, gamma = 1, beta = 0.0 )

list( alpha = 2, theta = 32000, delta = 0.8, gamma = 1.1, beta = 0.6 )

list( alpha = 1, theta = 15000, delta = 1.2, gamma = 0.9, beta = 0.3 )

list( alpha = 2, theta = 3000, delta = 0.8, gamma = 1.1, beta = 0.6 )
```

3.4 Fitting the Models to the Data in Table 3.1

To simplify the presentation, we will restrict our main focus in this section to the Pareto-Pareto model. An analysis of the Pareto-gamma model would proceed in a similar manner. We compiled the non-centered version of the Pareto-Pareto model to run 4 chains in parallel using WinBUGS. We updated the model for 10,000 iterations using a different set of initial values for each chain. This model updated very slowly and it took 926 seconds to complete 10,000 iterations (over 4 chains) on a dual 200 MHz Pentium Pro PC. (This time could be significantly improved if more informative prior distributions were available and / or better initial values were provided, particularly as much of the time was spent in the early iterations while WinBUGS was adapting.) The model parameters `alpha`, `theta`, `delta`, `gamma`, and `beta` were among those nodes monitored and their partial trace plots are presented in Figure 3.1. Inspection of these plots suggests that the chains are slow to mix, especially in the case of the variables `gamma` and `beta`. (This should become even more evident if the reader elects to run the example for him or herself and monitors the trace plots as they are dynamically generated. It may also help to examine the complete history of the sample paths obtained by clicking the *history* button on the *Sample Monitor Tool*.)

The Gelman-Rubin convergence diagnostic plots for the monitored nodes are presented in Figure 3.2. These plots were obtained in WinBUGS by clicking the *GR diag* button on the *Sample Monitor Tool*. The modified Gelman-Rubin convergence diagnostic underlying these plots is based on a generalization due to Brooks and Gelman (1998) of a method proposed by Gelman and Rubin (1992), and is described in Sections 1.2 and 6.1 of the WinBUGS *User Manual*. Essentially, for each monitored node we have plots of measures of the within and between variability in the traces and their ratio, using the methodology of Brooks and Gelman (1998). These measures are all calculated dynamically, as the simulation progresses. One should be concerned with convergence of the ratio to 1, and also with convergence of both the within and between variability measures to stability. A more detailed description of the rationale underlying these modified Gelman-Rubin convergence diagnostic plots is given in the next paragraph.

Suppose we have run m chains in parallel and the parameter ν was one of the monitored nodes. Any summary estimate (either point or interval) of the parameter ν can be constructed using either the total pooled output across all m chains or on the basis of the output from each chain individually. The former would lead to a single pooled or total-sequence estimate, while the latter would lead to m within-sequence estimates. Consider the determination of an empirical central $100(1-\alpha)\%$ (posterior) interval estimate for ν using the pooled output. The endpoints of this interval are given by the empirical $100(\alpha/2)\%$ and $100(1-\alpha/2)\%$ points for the complete set of values of ν, that is, pooled across all m chains. Denote this pooled or total-sequence interval width estimate as R_p. Considering the output from each chain individually, we can also obtain m within-chain empirical central $100(1-\alpha)\%$ interval estimates for ν and then determine their mean width. Denote this mean within-sequence interval width estimate as R_w. Finally, calculate the ratio R of the total-sequence interval width estimate R_p to the mean within-sequence interval width estimate R_w, that is, $R = R_p/R_w$. All of these values can be calculated on the fly and recorded as the MCMC simulation progresses. As the MCMC simulation converges, the values of the total-sequence and mean within-sequence interval width estimates should each converge to a stable value, and their ratio R (the modified Gelman-Rubin convergence diagnostic) should converge to one. If this is not the case and R is still much greater than one, this suggests that the practitioner should either continue the simulation for a longer while or assume that there is some feature of the model which is causing slow convergence. The process just described for ν should generally be applied to each of the model parameters of main interest.

When the *GR diag* button is clicked, WinBUGS calculates these interval width quantities, after discarding a portion of the initial iterations as burn-in, for each selected node in bins of length 50 using $\alpha = 0.20$. WinBUGS also displays a plot of the three quantities R_p, R_w, and R for each selected node as in Figure 3.2. Note that the actual plots in WinBUGS are colour coded with the sequence of R_p, R_w, and R values in green, blue, and red, respectively. For plotting purposes, the total-sequence and within-sequence interval width estimates are also normalised to have an overall maximum of one. In Figure 3.2, we can observe that the

convergence diagnostic R does appear to converge to one near about iteration 7,000 in the case of each monitored node. The measures R_p and R_w have also stabilized in each plot by this time. Note that the actual values underlying any one of these diagnostic plots can be listed to a window by double-clicking on that plot followed by a ctrl-left-mouse-click on the window.

Continuing with the analysis, we next compiled the centered version of the model to run 4 chains in parallel. WinBUGS took 1,048 seconds to perform 10,000 updates of this model. Partial trace plots appear in Figure 3.3 and the Gelman-Rubin diagnostic plots appear in Figure 3.4. From the trace plots, we can observe that the chains appear to mix better for the centered model than did those for the non-centered one. Also, effective convergence can be diagnosed much sooner than before from the Gelman-Rubin diagnostic plots - near about iteration 5,000. In summary, we have some justification that the centered version of the model converges faster than the non-centered one and so we will make use of the former's MCMC simulation-based output to fashion our posterior and predictive inferences.

With this being the case, we updated the centered model for an additional 20,000 iterations. This took 522 seconds in WinBUGS. Summary statistics based on the final 25,000 of the 30,000 total iterations are given in Table 3.2 (the first 5,000 we discarded as burn-in). We selected the final 25,000 iterations by setting the "beg" slot equal to 5001 and the "end" slot to 30000, on the *Sample Monitor Tool*. As we ran 4 chains in parallel, these summary statistics are based on 100,000 = 4 × 25,000 simulated values of each parameter. Estimated density plots for the five model parameters appear in Figure 3.5.

If we want to implement posterior predictive inference, the next step is to dump a text representation of the simulated values of the monitored nodes `alpha`, `theta`, `delta`, `gamma`, and `beta` nodes using the *coda* button on the *Sample Monitor Tool*. These values can now be read into an application like EXCEL or S-PLUS that will enable the practitioner to simulate a random draw (x, y) from the centered version of (3.3) for each quintuple of simulated model parameters. Thus, the 100,000 quintuples of simulated model parameters we have generated can be used in this way to obtain 100,000 predictive draws from the centered Pareto-Pareto

model for bivariate losses. The coverages (1) and (2) under consideration can be applied to this predictive sample and inferences developed on the basis of the resulting values. Following just this procedure led us to the estimated 50th, 75th, 90th, 95th, and 99th percentiles of the posterior predictive total loss distribution presented in Table 3.3. (It is possible that we may need to increase the length of the simulation in order to obtain good estimates of the 95th and 99th percentiles. Klugman *et alia* (1998, pages 325-327) discuss some sample size considerations in the context of traditional Monte Carlo simulations.) For comparison, we have included the same estimated percentiles of the mle-fitted predictive distribution for the total loss under each coverage. It is apparent that the Bayesian predictive distribution is considerably more dispersed, at least in this example.

Note that we have not calculated sample means in order to estimate the corresponding population means. This is for the simple reason that the population means do not exist. In the case of the predictive total loss distribution conditioned on the mles, this is due to the fact that $\hat{\delta} = 0.8484348 \leq 1$. In the case of the fully Bayesian predictive distribution, this is the case as positive posterior probability is assigned to both the event that $\alpha \leq 1$ and $\delta \leq 1$. Specifically, the estimated posterior means of the nodes eofx.exist and eofy.exist reported in Table 3.2 tell us that the events $\alpha \leq 1$ and $\delta \leq 1$ have posterior probabilities approximately equal to 0.0353 and 0.7039, respectively. If the posterior evidence were not so convincing, we might be willing to impose strict 'greater than' constraints on the parameters α and δ using the `I( 1, )` construct in the specification of each model parameter's prior density. If we strongly believe that future loss and ALAE values are limited in scope for all practical purposes, then we can impose educated guestimates as upper bounds on the value of each and / or their sum as we sample the predictive draws (x, y). This may be a reasonable course of action to follow in practice and would permit meaningful estimates of expected predictive total losses to be calculated.

For comparison's sake we also ran the centered version of the Pareto-gamma model in WinBUGS for 25,000 iterations following a burn-in of 5,000, using 4 chains as before. The 100,000 quintuples of simulated model parameters thus generated were used to obtain 100,000

predictive draws from the centered Pareto-gamma model for bivariate losses. The coverages (1) and (2) were applied to this predictive sample in order to obtain the summary statistics given in Tables 3.4 and 3.5.

Table 3.1

Twenty-four Losses with ALAE

(Klugman, Panjer, and Willmot, 1998).

Loss	ALAE	Loss	ALAE
1,500	301	11,750	2,530
2,000	3,043	12,500	165
2,500	415	14,000	175
2,500	4,940	14,750	28,217
4,500	395	15,000	2,072
5,000	25	17,500	6,328
5,750	34,474	19,833	212
7,000	50	30,000	2,172
7,000	10,593	33,033	7,845
7,500	50	44,887	2,178
9,000	406	62,500	12,251
10,000	1,174	210,000	7,357

Table 3.2

Estimated Posterior Summary Statistics

for the Centered Pareto-Pareto Model.

	Estimates from WinBUGS				
Parameter	Mean	SD	2.5%	Median	97.5%
NLL	485.1	1.562	483.0	484.8	489.0
alpha	3.33	2.647	0.9188	2.565	10.39
theta	51,670.0	54,260.0	8,341.0	35,180.0	196,400.0
delta	0.9158	0.522	0.3621	0.7951	2.194
gamma	6.783	0.8842	5.074	6.769	8.588
beta	0.6204	0.3402	- 0.03507	0.6164	1.317
eofx.exist	0.9647	0.1845	0.0	1.0	1.0
eofy.exist	0.2961	0.4565	0.0	0.0	1.0

Table 3.3

Estimated Percentiles of the Predictive Total Loss Distributions for the Pareto-Pareto Model.

		Predictive Distribution Type	
Coverage	Percentile	Fully Bayesian	MLE-Based
1	50th	15,176	14,156
	75th	37,110	33,159
	90th	58,015	54,088
	95th	94,920	71,413
	99th	1,081,144	281,900
2	50th	0	0
	75th	0	0
	90th	6,699	677
	95th	48,525	31,839
	99th	68,257	58,664

Table 3.4

Estimated Posterior Summary Statistics
for the Centered Pareto-Gamma Model.

	Estimates from WinBUGS				
Parameter	Mean	SD	2.5%	Median	97.5%
NLL	485.9	1.491	483.9	485.6	489.6
alpha	3.349	2.821	0.9166	2.545	10.98
theta	52,060.0	57,670.0	8,222.0	34,680.0	209,000.0
delta	0.4295	0.1025	0.2546	0.4203	0.654
gamma	- 9.52	0.424	- 10.45	- 9.486	- 8.788
beta	- 0.2307	0.2827	- 0.807	- 0.2221	0.3001
eofx.exist	0.961	0.1935	0.0	1.0	1.0

Table 3.5

Estimated Percentiles of the Predictive Total Loss Distributions for the Pareto-Gamma Model.

		Predictive Distribution Type	
Coverage	Percentile	Fully Bayesian	MLE-Based
1	50th	16,710	15,382
	75th	35,444	31,739
	90th	52,650	50.926
	95th	61,642	57,125
	99th	99,797	76,873
2	50th	0	0
	75th	0	0
	90th	6,470	472
	95th	43,658	30,738
	99th	55,489	53,288

Figure 3.1

Multiple Chain Trace Plots for Monitored Nodes
(for the Non-Centered Pareto-Pareto Model).

Figure 3.2

Gelman-Rubin Convergence Statistic Diagnostic Plots
(for the Non-Centered Pareto-Pareto Model).

Figure 3.3

Multiple Chain Trace Plots for Monitored Nodes
(for the Centered Pareto-Pareto Model).

Figure 3.4

Gelman-Rubin Convergence Statistic Diagnostic Plots
(for the Centered Pareto-Pareto Model).

Figure 3.5

Density Plots for Monitored Nodes
(for the Centered Pareto-Pareto Model).

4 Motor : A Predictive Aggregate Claims Distribution

4.1 Introduction

One of the most common and elementary of statistical problems faced by an actuarial practitioner is that of forecasting future aggregate claims. The traditional aggregate risk model assumes that

$$S_t = Y_{t,1} + \ldots + Y_{t,N_t} \ .$$

Here, S_t represents the aggregate claim amount in the t-th time period, N_t represents the number of claims occuring in that same period, and $Y_{t,1}$, $Y_{t,2}$, ..., represent the amounts of the successive claims in the period. Define $S_t = 0$ when $N_t = 0$. We suppose that the frequency and severity components of the process are independent, and that the individual claim amounts $\{Y_{t,i}\}$ are independent and identically distributed.

A number of recursive algorithms are available with which to compute the aggregate claims distributions associated with a great many combinations of parametric frequency and severity models (e.g., Klugman, Panjer, and Willmot, 1988, pages 309-316). The parameters of the frequency and severity models must be determined in advance of the application of any one of these recursive algorithms. Typically, this is accomplished by fitting the models to sample data using maximum likelihood estimation (or possibly method of moments estimation or percentile matching). For all intents and purposes, the parameters of these models *are* the point estimates. As noted by Dickson, Tedesco, and Zehnwirth (1998, page 695), a major drawback with this approach is that the recursive algorithms all assume that the frequency and severity model parameters are known with certainty. By failing to incorporate parameter uncertainty into the analysis, the true variability of the future aggregate claims distribution may be significantly understated and this can have a major impact on its moments and percentiles (Dickson, Tedesco, and Zehnwirth, 1998, page 706).

In this example, we illustrate how parameter uncertainty can be incorporated into the analysis of 5 years worth of reinsurance data appearing in Rytgaard (1990). This data is given in Table 4.1. The reinsurer has information about all single loss amounts exceeding

$c = 1.5$ million over a five year period. The losses have already been indexed for inflation and it is assumed that no such problem as IBNR exists. Rytgaard (1990) fit a compound Poisson model with a one parameter Pareto model severity component to this data using maximum likelihood and moment based estimation. The same data was subsequently considered from a Bayesian perspective by Hesselager (1993), who assumed a Poisson model for the number of claims reported and a one parameter Pareto model for claim severities and adopted conjugate priors. Pai (1997) also adopted a Bayesian approach for modelling this data, and considered a variety of frequency and severity models. We will assume a Poisson distribution for claim frequencies and use the same Pareto distribution for claim severities found in Pai (1997) and Klugman *et alia* (1998, page 584). Our model is described in detail below.

While Rytgaard's (1990) analysis assumed that the lower limit $c = 1.5$ was known as very often the reinsurer will receive information about all losses exceeding a certain limit, he also noted that the value of c might be unknown and have to be estimated in the case that the reinsurer receives only a list of the largest losses. Our Pareto model is flexible enough to cover both contingencies and is given by

$$f(y) = \frac{ab^a}{y^{a+1}} \ , \ y \ \geq \ b \ > \ c \ , \tag{4.2}$$

with $a > 0$ and $c \ \geq \ 0$. Depending on the context, the parameter b may be assigned equal to c (which may be known or unknown) or else be treated as an extra parameter to be estimated with c then interpreted as a lower bound on its value. In accordance with Pai (1997), we adopt the second interpretation with $c = 1.5$. The Pareto distribution defined by (4.2) corresponds to `dpar(a,b)` in BUGS. Our complete model is as follows :

$$\begin{aligned}
Y_{t,i} &\sim \text{Pareto } (a, b) \ , \ t = 1, \ldots, 5 \ , \ i = 1, \ldots, N_t \\
N_t &\sim \text{Poisson } (\lambda) \ , \ t = 1, \ldots, 5 \\
a &\sim \text{gamma } (0.001, 0.001) \\
b &\sim \text{gamma } (0.001, 0.001) \\
\lambda &\sim \text{gamma } (0.001, 0.001) \ .
\end{aligned} \tag{4.3}$$

Relatively noninformative priors are assigned to the parameters a, b, and λ. These may be overridden with informative priors when such are available. It is also to be understood that $min(Y_{t,i}) \geq b > c$, and this restriction is coded in the file motor.bug. In addition to posterior inference with respect to the model parameters a, b, and λ, we wish to implement predictive inference with respect to the next period's aggregate claims distribution. We will return to this aspect of the analysis shortly.

4.2 Model Specification and BUGS Files for the Motor Example

Note that since each of the 16 size of loss random variables is identically and independently distributed given the model parameters a and b, we can dispense with the double subscript notation $Y_{t,i}$ and work with a simple vector of length 16 instead. The complete files required to implement an analysis of the motor model using 'classic' BUGS are given below. They are presently maintained on the world wide web at www.math.ucalgary.ca/∼scollnik/abcd/.

One observation with respect to the Pareto density (4.2) should be made at this time. Note that the k-th moment of this distribution exists provided that $a > k$ (Klugman, *et alia*, 1998, page 584). For any particular value of k, this event can be monitored during the running of the MCMC simulation. If the posterior probability of the monitored event is judged to be sufficiently large, then the practitioner may decide to impose the corresponding restriction and rerun the MCMC simulation on that basis. In the motor.bug file, the deterministic nodes eofY.exist and eofY2.exist are used to monitor the events that $a > 1$ and $a > 2$, respectively.

4.2.1 The File **motor.bug**

```
model motor;

const
  T = 5,         # number of years with observed data
  L = 16,        # number of size of loss data values
  bmin = 1.5,    # the minimum value the second Pareto parameter can take
  bmax = 1.625; # the maximum value the second Pareto parameter can take
                 # the value of bmax is the smallest value observed in Y
var
  N[T],          # number of claims in years 1 through 5
  N6,            # future predicted number of claims for year 6
  lambda,        # frequency distribution (Poisson) parameter

  Y[L],          # size of loss data values in years 1 through 5
  Y6,            # future predicted individual size of loss value for year 6
  a, b,          # severity distribution (Pareto) parameters

  eofY.exist,    # to monitor whether the first moment of Y will exist
  eofY2.exist;   # to monitor whether the second moment of Y will exist

data in "motor-s.dat";          # data in S-PLUS format
inits in "motor.in";

{

# Model structure for Pareto modelling.

  Y6 ~ dpar( a, b );
  for( i in 1:L ){
      Y[i] ~ dpar( a, b );
  }

# Define prior densities on a and b, with or without a restriction on a.

  a ~ dgamma( 0.001, 0.001 );
  # a ~ dgamma( 0.001, 0.001 ) I( 1, ); # for finite 1st moment
  # a ~ dgamma( 0.001, 0.001 ) I( 2, ); # for finite 1st and 2nd moments
  b ~ dgamma( 0.001, 0.001 ) I( bmin, bmax );
```

```
# Indicator variables to monitor whether  a > 1  and / or  a > 2.
# These are the conditions which must be satisfied in order to ensure
# the existence of the first and second moments of the Pareto random
# variables, respectively.

  eofY.exist <- step( a - 1 );
  eofY2.exist <- step( a - 2 );

# Model structure for Poisson modelling.

  N6 ~ dpois( lambda );
  for( i in 1:T ){
      N[i] ~ dpois( lambda );
  }

  lambda ~ dgamma( 0.001, 0.001 );

}
```

4.2.2 The Data File **motor-s.dat**

```
list( N = c( 5, 3, 4, 0, 4 ),

      Y = c( 2.495,  2.120, 2.095, 1.700, 1.650,
             1.985,  1.810, 1.625,
             3.215,  2.105, 1.765, 1.715,
             19.180, 1.915, 1.790, 1.755 ) )
```

4.2.3 The Initial Value File **motor.in**

```
list(a = 3, b = 1.6, lambda=15)
```

4.3 Output and Analysis

After a burn-in of 5,000 iterations, it took BUGS 11 seconds to run for a further 15,000 iterations on a 150 MHz Pentium PC. The parameters a, b, λ, N_6, and Y_6 were monitored and the summary statistics for these parameters over the last 15,000 iterations appear in Table 4.2. Partial sample paths and empirical histograms of the simulated values of the parameters a, b, and λ appear in Figures 4.1, 4.2, and 4.3, respectively. Note that N_6 corresponds to the future number of claims in year 6 and Y_6 is an individual future loss in that same year. For the predictive density of N_6, we have

$$\begin{aligned} Pr(N_6 = n \mid N_1, \ldots, N_5) &= \int_0^\infty Pr(N_6 = n \mid \lambda)\ f(\lambda \mid N_1, \ldots, N_5)\ d\lambda \\ &\approx \frac{1}{15,000} \sum_{i\ =\ 5001}^{20,000} Pr(N_6 = n \mid \lambda^{(i)}) \\ &= \frac{1}{15,000} \sum_{i\ =\ 5001}^{20,000} \frac{e^{-\lambda^{(i)}} \left[\lambda^{(i)}\right]^n}{n!} \ . \end{aligned}$$

The estimated density on the right hand side of this expression is equal to the average of the conditional Poisson probability mass function for N_6 given the Poisson parameter λ, taken over that parameter's sampled draws. The values of this estimated predictive density are given in Table 4.3 and are plotted in Figure 4.4. The predictive density of Y_6 can be estimated in a like manner, by averaging (4.2) over the sampled values of a and b.

In this example, our real interest is in the future distribution of the aggregate claims. This is given by

$$f(S_6 \mid Obs.\ Data) = \int_0^\infty \int_0^\infty \int_{1.5}^{1.625} f(S_6 \mid \lambda, a, b)\ f(\lambda, a, b \mid Obs.\ Data)\ d\lambda\ da\ db \ ,$$

with $S_6 = Y_{6,1} + \ldots + Y_{6,N_6}$ when $N_6 > 0$, and $S_6 = 0$ otherwise. Given the model parameters, we assume as before that :

$$\begin{aligned} Y_{6,i} &\sim \text{Pareto}(a, b)\ ,\ i = 1, \ldots, N_6 \\ N_6 &\sim \text{Poisson}(\lambda)\ . \end{aligned}$$

The conditional compound distribution $f(S_6 \mid \lambda, a, b)$ appearing in the expression above may be calculated recursively, and then averaged over the sampled values of a, b, and λ. This

can be a time-consuming process, though, and so we may prefer to estimate the predictive aggregate distribution on the basis of a further simulation. In particular, for each triplet of values (λ, a, b) previously sampled, proceed as follows :

1. Sample $N_6 \sim \text{Poisson}(\lambda)$;

2. If $N_6 = 0$, set $S_6 = 0$;

3. If $N_6 > 0$, sample $Y_{6,i} \sim \text{Pareto}(a, b)$, $i = 1, \ldots, N_6$, and then set $S_6 = Y_{6,1} + \ldots + Y_{6,N_6}$.

Then predictive inference may be conducted on the basis of the empirical distribution of the sampled values of S_6. We implemented these 3 steps for each of the 15,000 sets of parameters we previously generated and graphed the empirical density of the resulting draws from the predictive aggregate distribution in Figure 4.5 (NB, one particularly large realization of S_6 exceeding 1200 was truncated at 300 for plotting convenience). This portion of our simulation was implemented using the statistical software package S-PLUS, but it may be implemented in any programming environment permitting the requisite draws to be made from the Poisson and Pareto distributions described above.

As noted earlier, the traditional approach to computing an aggregate claims distribution involves estimating the frequency and severity distributional parameters using a method like maximum likelihood, and then compounding the frequency and severity model with the parameters fixed at their estimated values. In this example, the maximum likelihood estimates are $\hat{\lambda} = 3.2$, $\hat{a} = 3.076351$, and $\hat{b} = 1.625$. We sampled 15,000 draws from the future aggregate claims distribution conditioned on the maximum likelihood estimates and graphed the empirical density of the resulting draws in Figure 4.5 directly below the Bayesian predictive density estimate. It may be discerned from the two plots in Figure 4.5 that the Bayesian predictive density is significantly less concentrated and more skew than the future aggregate claims distribution conditioned on the maximum likelihood estimates. This is also illustrated in Figure 4.6, in which the empirical quantiles for the samples from these two distributions are plotted against one another. These figures support the assertion of Dickson, Tedesco, and Zehnwirth (1998, page 706) that failing to incorporate parameter uncertainty into the analysis can result in the true variability of the future aggregate claims distribution

being significantly understated and so adversely affect moments and percentiles.

The output of the MCMC simulation presented in this example may also be used to price per claim excess of loss reinsurance, aggregate excess of loss reinsurance, and quoata share reinsurance treaties. Details may be found in Pai (1997).

4.4 Further Discussion of the Motor Example

This example involved the analysis of severity data with a truncated Pareto distribution. Our analysis was not complicated by the truncation since the Pareto distribution used in 'classic' BUGS conveniently incorporates an explicit lower limit for the data values. It should be remarked that the present version of BUGS / WinBUGS does not support the truncation of arbitrary distributions for modelling observed data : see Section 4.6 of the BUGS 0.5 Manual (version ii) (Spiegelhalter, Thomas, Best, and Gilks, 1996) for more information.

Originally, we implemented this example in 'classic' BUGS as the Pareto distribution was not supported in WinBUGS at the time (i.e., in version 1.1). This changed with the release of WinBUGS (beta) version 1.2 (May, 1999). However, using a trick that was originally found on the FAQ (frequently asked questions) page of the BUGS website www.mrc-bsu.cam.ac.uk/bugs, we were able to port the example over to the earlier version of WinBUGS. The advice on the FAQ page read as follows :

> Suppose your data is y (of length n) and you want to fit the model p(y) = f(y,t) where t are the unknown parameters and f is the distribution that is not currently handled by BUGS. Create a new vector 'ones', that comprises just 1's and is of length n. Then use the BUGS code:
>
> ```
> for( i in 1 : n ){
>    ones[i] ~ dbern( p[i] )
>    p[i] <- f(y[i],t)
> }
> ```
>
> This should provide a likelihood term proportional to f(y,t).

This 'ones' trick has since been documented in Section 3.2 of the WinBUGS *User Manual* and it is incorporated in the illustrative WinBUGS code given below for the motor model. The corresponding motorold.odc file is available on this author's website at www.math.ucalgary.ca/∼scollnik/abcd/ for use with WinBUGS version 1.1. The simpler motor.odc file can be used with WinBUGS (beta) version 1.2.

```
CODE FOR THE MOTOR ( PARETO / POISSON ) MODEL, for WinBUGS version 1.1

model;
{
   # Generate a predictive draw from the Pareto distribution using a
   # definition of the shifted Pareto distribution given in terms of
   # mixtures.
   Y6 <- X6 + b
   X6 ~ dexp( theta )
   theta ~ dgamma( a, b )

   # The 'ones' trick.
   for( i in 1 : L ){
      ones[i] <- 1
      ones[i] ~ dbern( p[i] )
      p[i] <- a * pow( b / Y[i], a ) / ( bignum * Y[i] )
   }
   # The value of 'bignum' is rather arbitrary. It should be large enough
   # so that each p[i] is always less than one. Otherwise an error will
   # eventually arise when an update results in a value of a Bernoulli
   # probability outside of the (0,1) range. However, the value of bignum
   # plays no role in the analysis, as it is simply a constant being
   # multiplied into the likelihood.
   bignum <- 100

   a ~ dgamma( 0.001, 0.001 )
   # a ~ dgamma( 0.001, 0.001 ) I( 1, ) # for finite 1st moment
   # a ~ dgamma( 0.001, 0.001 ) I( 2, ) # for finite 1st and 2nd moments
   b ~ dgamma( 0.001, 0.001 ) I( bmin, bmax )

   eofY.exist <- step( a - 1 )
   eofY2.exist <- step( a - 2 )
```

```
   N6 ~ dpois( lambda )
   for( i in 1 : T ){
      N[i] ~ dpois( lambda )
   }

   lambda ~ dgamma( 0.001, 0.001 )
}

DATA

list( N = c( 5, 3, 4, 0, 4 ),
      Y = c( 2.495, 2.120, 2.095, 1.700, 1.650,
             1.985, 1.810, 1.625,
             3.215, 2.105, 1.765, 1.715,
             19.180, 1.915, 1.790, 1.755 ),
       N6 = NA,
       Y6 = NA,
       T = 5, L = 16,
       bmin = 1.5, bmax = 1.625 )
```

Table 4.1

Claims Exceeding $c = 1.5$ Million During a 5 Year Period.

Year Number				
1	2	3	4	5
2.495	1.985	3.215	-	19.180
2.120	1.810	2.105	-	1.915
2.095	1.625	1.765	-	1.790
1.700	-	1.715	-	1.755
1.650	-	-	-	-

Table 4.2

Estimated Posterior Summaries for Monitored Parameters of Interest.

Parameter	Mean	SD	2.5%	Median	97.5%
λ	3.209	0.8064	1.824	3.143	4.970
a	2.911	0.7443	1.637	2.848	4.534
b	1.594	0.0272	1.522	1.602	1.624
eofY.exist	0.999	0.0199	1.000	1.000	1.000
eofY2.exist	0.907	0.2898	0.000	1.000	1.000
N_6	3.203	1.940	0.000	3.000	7.000
Y_6	2.583 †	2.421 †	1.602	2.032	6.679

† Whereas these sample moments will always be finite, one or more of the corresponding theoretical predictive moments of Y_6 may in fact equal infinity. To ensure the existence of the first and second predictive moments, it will be necessary to restrict $a > 1$ and $a > 2$, respectively. The monitored values of eofY.exist and eofY2.exist indicate that it may be reasonable to rerun the simulation incorporating the second (i.e., stronger) of these restrictions.

Table 4.3

Estimated Density of the Future Claim Number.

n	$\Pr(\ N_6 = n \mid \text{Data}\ )$	n	$\Pr(\ N_6 = n \mid \text{Data}\ )$
0	0.0538373	8	0.0159708
1	0.1438783	9	0.0071456
2	0.2040566	10	0.0030041
3	0.2041858	11	0.0011975
4	0.1617578	12	0.0004561
5	0.1079688	13	0.0001670
6	0.0631147	14	0.0000592
7	0.0331700	≥ 15	0.0000304

5 Norweg : Heterogeneity in Group Life Insurance

5.1 Introduction

Haastrup (1997, page 1) described one possible approach for the modelling of heterogeneity between policy holders. To begin with, assume that each policy holder is equipped with a heterogeneity parameter. The heterogeneity parameters are assumed to be independent and identically distributed draws from an unknown heterogeneity distribution. A tractable parametric distribution (e.g., a conjugate prior) is imposed on the heterogeneity distribution. The parameters in the heterogeneity distribution are then estimated from the observed data, and based on these estimates the posterior distribution of the individual heterogeneity parameters is found. This describes what is called a parametric empirical Bayesian approach. Using this approach and two others (i.e., parametric fully Bayesian and non-parametric fully Bayesian), Haastrup re-examined the heterogeneity in a portfolio of group life insurances previously considered by Norberg (1989). Haastrup's parametric fully Bayesian analysis was implemented using the Gibbs sampler and we are interested in implementing the same analysis using BUGS.

The data appear in Table 5.1 and arise from 1125 groups insured through the whole or parts of the period 1982-85 by a major Norwegian insurance company. There are $J = 72$ classes distinguished by occupational category. The j-th class has risk exposure W_j and the observed number of deaths N_j. Class j has a level of heterogeneity represented by a parameter θ_j , and the number of deaths N_j in this class follows a Poisson distribution with mean $W_j\theta_j$. The classes are assumed to be mutually independent given the heterogeneity parameters $\theta_1, \ldots, \theta_J$. The heterogeneity parameters themselves are independent draws from a common gamma distribution. Our model so far is :

$$N_j \sim \text{Poisson } (W_j\theta_j) \ , \ j = 1, \ldots, J, \tag{5.1}$$

$$\theta_j \sim \text{gamma } (\gamma, \delta) \ , \ j = 1, \ldots, J. \tag{5.2}$$

Specification of the parameters γ and δ remains. A parametric fully Bayesian analysis pro-

Bayesian approach would use the data to determine point estimates of γ and δ, typically on the basis of a maximum likelihood or possibly a method of moments based estimation procedure. (It should be noted that such an empirical Bayesian approach technically contradicts the rules of conditional probability by using the data twice ; once in the likelihood and again in shaping the form of the prior. Nevertheless, it is sometimes adopted in the literature and in practice as a concession to those who demand that an analysis be conducted in a manner independent of the practitioner's bias or personal prior beliefs.) We will consider both analyses below.

Figure 4.2

Partial Sample Path and Empirical Histogram of the Simulated Values of the Pareto Parameter a.

Partial Sample Path and Empirical Histogram of the Simulated Values of the Poisson Parameter λ.

5.2 Parametric Empirical Bayesian Analysis

Let us first consider a parametric empirical Bayesian styled analysis. If we wish to estimate γ and δ using maximum likelihood, we start by noting that

$$\begin{aligned} f(N_j \mid W_j, \gamma, \delta) &= \int_0^\infty f(N_j \mid W_j\theta_j) f(\theta_j \mid \gamma, \delta)\, d\theta_j \qquad (5.3) \\ &= \frac{\delta^\gamma\, W_j^{N_j}}{\Gamma(\gamma)\,\Gamma(N_j+1)} \int_0^\infty \theta_j{}^{\gamma+N_j-1}\, e^{-[\delta+W_j]\theta_j} d\theta_j \\ &= \frac{\Gamma(\gamma+N_j)}{\Gamma(\gamma)\,\Gamma(N_j+1)}\, \frac{\delta^\gamma\, W_j^{N_j}}{(\delta+W_j)^{\gamma+N_j}} \\ &= \frac{\Gamma(\gamma+N_j)}{\Gamma(\gamma)\,\Gamma(N_j+1)} \left(\frac{\delta}{\delta+W_j}\right)^{N_j+\gamma} \left(\frac{W_j}{\delta}\right)^{N_j}, \end{aligned}$$

for $N_j = 0, 1, \ldots$, which is the form of a negative binomial (γ , $\frac{W_j}{\delta}$) distribution with mean $\gamma\frac{W_j}{\delta}$ and variance $\gamma\frac{W_j}{\delta}(1+\frac{W_j}{\delta})$. Applying the maximum likelihood estimation procedure to this model with the data in Table 5.1 yields estimated values $\hat{\gamma} = 6.20$ and $\hat{\delta} = 5.45$. It should be remarked that finding these maximum likelihood estimates is not a particularly easy task to accomplish by hand, and need not be trivial on a computer either unless the appropriate software resides on one's computer. We solved the problem using the statistical software package S-PLUS. The code appears below :

Figure 4.3

Partial Sample Path and Empirical Histogram of

```
# Define the 72 counts and exposures.

N <- c( 16, 0, 3, ... , 2, 0 )
W <- c( 9.75, 0.52, 0.45, ... , 1.85, 0.12 )

# Set up the model and define the parameters.

norweg <- data.frame( data1=N , data2=W )
param( norweg , "g" )
param( norweg , "d" )

# Numerically solve for the mles by minimizing the negative log-likelihood.

ms( ~ - ( log( d ** g ) + lgamma( g + data1 ) -
        lgamma( g ) - ( g + data1 ) * log( d + data2 ) ) ,
        norweg , start = list( g = 6 , d = 5 , trace = T )
```

The S-PLUS function `ms` minimizes a sum of nonlinear functions over the parameters appearing in a data structure called a data frame. In this context, we are using it to determine the values of the parameters g (for γ) and d (for δ) which minimize the negative log-likelihood and so define the maximum likelihood estimates. For more information, type `?ms` in an S-PLUS commands window. (Incidentally, a pracitioner experienced with EXCEL may prefer to attempt maximizing the likelihood using EXCEL Solver. However, this task is left as an exercise for the reader.)

Once the values of the maximum likelihood estimates have been determined as above, we simply insert them into (5.2) and then derive the posterior distribution of the parameters θ_j, $j = 1, \ldots, J$. The gamma distribution is the well-known conjugate prior for the Poisson model appearing in (5.1), meaning the posterior distribution for each heterogeneity parameter will be gamma as well with updated parameters. Specifically,

$$f(\theta_j \mid N_j, W_j) \;\sim\; \text{gamma}\,(N_j + \hat{\gamma}, W_j + \hat{\delta})\;,\; j = 1, \ldots, J. \tag{5.4}$$

Figure 4.4

Figure 4.5

Estimated Density of the Future Claim Number
Estimated Density of the Future Aggregate Claim.

The first plot is the Bayesian predictive density. The second plot is of the density of the future aggregate claim conditioned on the mles.

Figure 4.6

Empirical Quantile-Quantile Plot of the Two Future Aggregate Claim Distributions.

The empirical quantiles of the Bayesian predictive distribution are on the x-axis and are plotted against the corresponding empirical quantiles of the future aggregate claim distribution conditioned on the mles.

Hence, the estimated posterior mean and variance of the heterogeneity parameter θ_j are given by

$$\hat{E}(\theta_j \mid N_j, W_j) \;=\; \frac{N_j + \hat{\gamma}}{W_j + \hat{\delta}} \tag{5.5}$$

and

$$\hat{V}(\theta_j \mid N_j, W_j) \;=\; \frac{N_j + \hat{\gamma}}{(W_j + \hat{\delta})^2}\;, \tag{5.6}$$

respectively. Expression (5.5) is sometimes called the empirical Bayes estimate of θ_j , and (5.6) is the associated estimated risk. The empirical Bayes estimate (5.5), the square-root of the associated estimated risk (5.6), and the 2.5th, 50th, and 97.5th percentiles of the estimated posterior density (5.4) are given in Table 5.2 for six of the observed groups. The mean and standard deviation values differ very slightly from those reported in Haastrup (1997), even though they should agree exactly. The differences are attributed to the fact that the data appearing in Haastrup (1997) is very possibly rounded and not exactly as presented.

For a new class of group insurance with heterogeneity parameter θ_f and exposure W_f, suppose that $N_f \sim \text{Poisson}(W_f\theta_f)$ and $\theta_f \sim \text{gamma}(\hat{\gamma}, \hat{\delta})$. Along the lines of (5.3), we have

$$f(N_f \mid W_f, \hat{\gamma}, \hat{\delta}) \;\sim\; \text{negative binomial } \left(\hat{\gamma}, \frac{W_j}{\hat{\delta}}\right)\;.$$

Thus,

$$E(N_f \mid W_f, \hat{\gamma}, \hat{\delta}) \;=\; \hat{\gamma}\frac{W_j}{\hat{\delta}}$$

and

$$V(N_f \mid W_f, \hat{\gamma}, \hat{\delta}) \;=\; \hat{\gamma}\frac{W_j}{\hat{\delta}}\left(1 + \frac{W_j}{\hat{\delta}}\right)\;.$$

If the exposure for the new class was unknown, then it can be modelled probabilistically as were the other model parameters. This possibility will be explicitly considered in the next section.

5.3 Parametric Fully Bayesian Analysis

A parametric fully Bayesian analysis would augment (5.1) and (5.2) with a parametric prior density specification for the unknown parameters γ and δ. As in Haastrup (1997), we will

assume that these two parameters are independent *a priori* with

$$\gamma \sim \text{gamma}\,(1.2, 0.6) \quad \text{and} \quad \delta \sim \text{gamma}\,(1.2, 0.6)\ .$$

As previously noted, it may be desired to implement predictive inference with respect to the number of claims in a new class with an unknown level of exposure. In order to accommodate this eventuality, we will treat the exposures as random variables with

$$\begin{aligned} W_j &\sim \text{gamma}\,(a, b)\ ,\ j = 1, \ldots, J, \\ a &\sim \text{uniform}\,(0, 100)\ , \\ b &\sim \text{uniform}\,(0, 100)\ . \end{aligned}$$

Of course, the priors assigned to the parameters a and b may be replaced with more informative priors when such are available. We now suppose the new class of group insurance possesses heterogeneity parameter θ_f and an exposure W_f forecast to lie between, say, 0.1 and 2. Then we have

$$\begin{aligned} N_f &\sim \text{Poisson}\,(W_f \theta_f)\ , \\ W_f &\sim \text{gamma}\,(a, b)\ \text{I}(0.1, 2.0)\ , \\ \theta_f &\sim \text{gamma}\,(\gamma, \delta)\ . \end{aligned}$$

Note the use of the `I(0.1,2.0)` construct to restrict the range of permissible values for the parameter W_f. Of course, should W_f be known precisely it may be coded as a constant in the corresponding BUGS files.

5.4 Model Specification and BUGS Files for the Norweg Example

The graph for this example's complete probabilistic model is given in Figure 5.1. The complete files required to implement an analysis of the norweg model using 'classic' BUGS are given below. They are presently maintained on the world wide web, along with a counterpart file norweg.odc for use with WinBUGS, at www.math.ucalgary.ca/∼scollnik/abcd/.

5.4.1 The File norweg.bug

```
model norweg;

const
  J = 72;           # number of classes

var
  N[J],             # deaths for the J = 72 groups
  W[J],             # exposures for the J = 72 groups

  lambda[J],        # Poisson parameter
  theta[J],         # heterogeneity parameter
  parms[6],         # a vector for simultaneous storage of certain thetas

  Nf, Wf,           # variables for predictive analysis
  lambdaf,
  thetaf,

  gamma, delta,
  a, b;

data in "norweg-s.dat";     # data in S-PLUS format
inits in "norweg.in";

{

# Model structure for prediction wrt a new class.

  Nf ~ dpois( lambdaf );
  lambdaf <- Wf * thetaf;
  Wf ~ dgamma( a, b ) I( 0.1, 2 );
  thetaf ~ dgamma( gamma, delta );
```

```
# Model structure for Poisson-gamma modelling.

  for( j in 1:J ){
      N[j] ~ dpois( lambda[j] );
      lambda[j] <- W[j] * theta[j];
      theta[j] ~ dgamma( gamma, delta );
      W[j] ~ dgamma( a, b );
  }

  parms[1] <- theta[61];
  parms[2] <- theta[14];
  parms[3] <- theta[40];
  parms[4] <- theta[17];
  parms[5] <- theta[66];
  parms[6] <- theta[8];

  gamma ~ dgamma( 1.2, 0.6 );
  delta ~ dgamma( 1.2, 0.6 );
  a ~ dunif( 0, 100 );
  b ~ dunif( 0, 100 );

}
```

5.4.2 The Data File norweg-s.dat

```
list( N = c( 16, 0,  3,  0,  10, 12,
             3,  5,  15, 0,  3,  6,
             5,  24, 13, 15, 49, 5,
             0,  3,  18, 7,  0,  4,
             0,  9,  2,  5,  0,  6,
             0,  0,  14, 0,  0,  2,
             0,  1,  3,  57, 0,  21,
             2,  1,  11, 20, 2,  0,
             0,  2,  10, 2,  0,  14,
             4,  0,  2,  0,  1,  0,
             11, 1,  5,  4,  3,  12,
             0,  0,  26, 0,  2,  0 ),

      W = c( 9.75, 0.52,  0.45,  0.62,  6.06, 11.95,
             4.8,  8.32,  13.09, 0.01,  4.52, 6.68,
             4.06, 11.49, 21.91, 15.53, 77.9, 7.17,
             0.29, 3.28,  11.53, 9.49,  0.33, 2.74,
             1.40, 12.61, 1.25,  3.6,   0.07, 3.56,
             0.08, 0.28,  16.06, 0.07,  0.43, 0.68,
             0.07, 0.32,  3.05,  19.14, 0.05, 24.57,
             2.22, 0.4,   7.33,  26.3,  2.41, 0.11,
             0.03, 1.89,  6.97,  0.79,  0.61, 8.01,
             2.16, 0.22,  0.25,  0.09,  2.26, 0.68,
             6.62, 3.81,  7.0,   3.31,  0.8,  5.98,
             0.31, 1.89,  56.72, 0.13,  1.85, 0.12 ) )
```

5.4.3 The Initial Value File norweg.in

```
list( gamma = 1, delta = 1.6 )
```

5.5 Output and Analysis

After a burn-in of 10,000 iterations, it took BUGS 2 minutes and 34 seconds to run for a further 20,000 iterations on a 150 MHz Pentium PC. The heterogeneity parameters for six of the groups were monitored and the summary statistics for these parameters over the last 20,000 iterations appear in Table 5.3. Also included in this table are the summary statistics given for the same parameters from the analysis conducted by Haastrup (1997). The two analyses evidently agree, particularly when we recall that the data reported in Haastrup (1997) may have been rounded from the exact values. It is worth noting that the Markov chain Monte Carlo sampling scheme implemented for the parametric fully Bayesian analysis in Haastrup (1997) was therein advertised as slow to mix and was run for 100,000 iterations. However, Haastrup (1997) does not provide the specific details regarding the implementation of that simulation. Our results in this example suggest that the Markov chain Monte Carlo simulation coded in BUGS may run more efficiently and converge slightly faster.

Although we will not run through the entire gamut, a number of the other model parameters were also monitored throughout our simulation and some summary statistics are listed in Table 5.4. Note that the estimated posterior expectation of (γ,δ) under the parametric fully Bayesian analysis is equal to (4.62,4.02), which is a compromise between the prior expectation of (2,2) and the maximum likelihood estimate of (6.20,5.45). Using the terminal 20,000 simulated pairs $(\gamma^{(i)},\delta^{(i)})$, $i = 10,001,\ldots,30,000$, the posterior predictive distribution of the heterogeneity parameter θ_f under this analysis may be estimated using the result that

$$\begin{aligned} f(\theta_f \mid \textit{Obs. Data}) &= \int_0^\infty \int_0^\infty f(\theta_f \mid \gamma,\delta)\; f(\gamma,\delta \mid \textit{Obs. Data})\; d\gamma\; d\delta \\ &\approx \frac{1}{20,000} \sum_{i\,=\,10,001}^{30,000} f(\theta_f \mid \gamma^{(i)},\delta^{(i)}) \\ &= \frac{1}{20,000} \sum_{i\,=\,10,001}^{30,000} \frac{(\delta^{(i)})^{\gamma^{(i)}}}{\Gamma(\gamma^{(i)})}\; {\theta_f}^{\gamma^{(i)}-1}\; e^{-\delta^{(i)}\theta_f}\;. \end{aligned} \tag{5.7}$$

Under the parametric empirical Bayesian analysis, the corresponding distribution is simply

given by

$$f(\theta_f \mid Obs.\ Data) \ = \ f(\theta_f \mid \hat{\gamma}, \hat{\delta}) \ \sim \ \text{gamma}\ (\hat{\gamma}, \hat{\delta})\ . \tag{5.8}$$

Both (5.7) and (5.8) are plotted in Figure 5.2, from which it is apparent that the density estimate under the parametric fully Bayesian analysis is more dispersed. This was to be expected since it incorporates the extra parameter uncertainty with respect to γ and δ.

Finally, the predictive distribution for N_f under the parametric fully Bayesian analysis may be estimated using the result that

$$\begin{aligned} Pr(N_f = n \quad | \quad & Obs.\ Data) \\ = \ & \int_{0.1}^{2.0} \int_{0}^{\infty} Pr(N_f = n \mid W_f, \theta_f)\ f(W_f, \theta_f \mid Obs.\ Data)\ d\theta_f\ dW_f \\ \approx \ & \frac{1}{20,000} \sum_{i\ =\ 10{,}001}^{30{,}000} Pr(N_f = n \mid W_f^{(i)}, \theta_f^{(i)}) \\ = \ & \frac{1}{20,000} \sum_{i\ =\ 10{,}001}^{30{,}000} \frac{e^{-W_f^{(i)}\theta_f^{(i)}} \left[W_f^{(i)}\theta_f^{(i)}\right]^n}{n!}\ . \end{aligned}$$

The values of this estimated predictive density are listed in Table 5.5.

5.6 Further Discussion of the Norweg Example

The parametric fully Bayesian analysis appearing in Haastrup (1997) of the data in Table 5.1 was quickly and easily reproduced using BUGS. As in Haastrup (1997), we disregarded group variation with respect to additional classification factors like sex and age. However, the models considered in this example can be extended into models that recognize differences due to more than one factor. See the credit example for a model incorporating two classification factors.

Table 5.1

Exposures (W) and Deaths (N) for 72 Groups.

W	N	W	N	W	N	W	N	W	N	W	N
9.75	16	4.06	5	1.40	0	0.07	0	0.03	0	6.62	11
0.52	0	11.49	24	12.61	9	0.32	1	1.89	2	3.81	1
0.45	3	21.91	13	1.25	2	3.05	3	6.97	10	7	5
0.62	0	15.53	15	3.60	5	19.14	57	0.79	2	3.31	4
6.06	10	77.90	49	0.07	0	0.05	0	0.61	0	0.80	3
11.95	12	7.17	5	3.56	6	24.57	21	8.01	14	5.98	12
4.80	3	0.29	0	0.08	0	2.22	2	2.16	4	0.31	0
8.32	5	3.28	3	0.28	0	0.40	1	0.22	0	1.89	0
13.09	15	11.53	18	16.06	14	7.33	11	0.25	2	56.72	26
0.01	0	9.49	7	0.07	0	26.30	20	0.09	0	0.13	0
4.52	3	0.33	0	0.43	0	2.41	2	2.26	1	1.85	2
6.68	6	2.74	4	0.68	2	0.11	0	0.68	0	0.12	0

Table 5.2

Parametric Empirical Bayesian Posterior Summary Statistics for the Heterogeneity Parameters of Six Observed Groups.

Group		Posterior Summary Statistics				
N_j	W_j	Mean	SD	2.5%	Median	97.5%
11	6.62	1.4250	0.3436	0.8331	1.3975	2.1732
24	11.49	1.7828	0.3244	1.2046	1.7631	2.4725
57	19.14	2.5701	0.3233	1.9758	2.5566	3.2414
49	77.90	0.6623	0.0891	0.4992	0.6583	0.8481
12	5.98	1.5923	0.3732	0.9468	1.5632	2.4029
5	8.32	0.8134	0.2430	0.4090	0.7893	1.3544

Table 5.3

Parametric Fully Bayesian Posterior Summary Statistics for the Heterogeneity Parameters of Six Observed Groups.

Group		Haastrup's estimates		Estimates from BUGS				
N_j	W_j	Mean	SD	Mean	SD	2.5%	Median	97.5%
11	6.62	1.4725	0.3781	1.473	0.3800	0.8291	1.439	2.311
24	11.49	1.8485	0.3514	1.850	0.3539	1.227	1.824	2.613
57	19.14	2.6632	0.3482	2.666	0.3490	2.032	2.650	3.399
49	77.90	0.6546	0.0894	0.6546	0.0894	0.4906	0.6511	0.8412
12	5.98	1.6690	0.4173	1.670	0.4198	0.9584	1.632	2.298
5	8.32	0.7785	0.2536	0.7775	0.2538	0.3580	0.7497	1.347

Table 5.4

Parametric Fully Bayesian Posterior Summary Statistics for Some Additional Model Parameters of Interest.

	Estimates from BUGS				
Parameter	Mean	SD	2.5%	Median	97.5%
N_f	0.957	1.312	0	1	4
W_f	0.818	0.547	0.118	0.702	1.916
θ_f	1.156	0.570	0.327	1.064	2.517
γ	4.620	1.214	2.577	4.431	7.342
δ	4.020	1.111	2.139	3.841	6.511
a	0.482	0.064	0.365	0.479	0.618
b	0.077	0.016	0.048	0.076	0.110

Table 5.5

Estimated Predictive Density of N_f.

n	$Pr(N_f = n \mid Obs.\ Data)$
0	0.4949
1	0.2661
2	0.1276
3	0.0603
4	0.0280
5	0.0128
6	0.0058
7	0.0026
≥ 8	0.0020

Figure 5.1

Graphical Model for the Norweg Example.

Figure 5.2

Posterior Density of the Heterogeneity Parameter θ_f.

6 Credit : An Unbalanced 2-Way Crossed Classification Model

6.1 Introduction

A great many models of credibility have been developed over the past few decades. These include the Bühlmann-Straub model, Jewell's hierarchical model, crossed classification models, De Vylder's credibility IBNR model, and Hachmeister's credibility for regression models. A recent review and analysis of these models is provided in the text by Dannenburg, Kaas, and Goovaerts (1996). The analysis of these credibility models usually proceeds on the basis of best linear unbiased estimation, or via a parametric empirical Bayesian approach (e.g., Sections 5.4 and 5.5 of Klugman, Panjer, Willmot, 1998). In the past, parametric fully Bayesian analyses leading to exact Bayesian predictive premiums have often been avoided due to the analytical difficulties they can introduce. The Markov chain Monte Carlo (MCMC) methodology can often free the actuarial practitioner from this constraint. To illustrate, we will use BUGS to implement a parametric fully Bayesian analysis of an unbalanced two-way crossed classification model.

Dannenburg *et alia* (1996, Section 6.5) analyze data arranged in a two-way table and corresponding to payments of a credit insurer to several banks to cover losses caused by clients defaulting on private loans. The data appears in Table 6.1 and will supply our context. The observations are categorized according to two risk factors relating to the marital status of the debtors and the length of time they have worked with their current employer. Let I and J denote the number of levels or categories of the first and second risk factors, respectively. For the data in Table 6.1, $I = J = 3$. The first risk factor classifies the debtor (by row) as single, divorced, or otherwise ($i = 1, 2, 3$). The second risk factor classifies the time the debtor spent with his or her current employer (by column) as less than two years, from two to ten years, or more than ten years ($j = 1, 2, 3$). An interaction effect between the two factors (row and column) will also be included in the model. Let K_{ij} denote the number of observations in marriage class i and employment class j. We will refer to this intersection of classes as cell (i,j) and let X_{ijk} represent the k-th ($k = 1, \ldots, K_{ij}$) claim belonging to

it. Notice that the total number of observations is 401 and that the data is unbalanced - that is, the number of observations differs from cell to cell. The average amounts paid by the credit insurer within each cell are given in Table 6.2. The two-way crossed classification model assumes that

$$X_{ijk} \;=\; m + \alpha_i + \beta_j + \gamma_{ij} + \delta_{ijk} \;. \tag{6.1}$$

The components α_i, β_j, γ_{ij}, and δ_{ijk} listed from left to right in (6.1) are regarded as independent random variables with mean zero and with variances σ_α^2 , σ_β^2 , σ_γ^2 , and σ_δ^2/w_{ijk} , respectively. The exposures w_{ijk} are assumed to be known. In this particular example, we have $w_{ijk} = 1$ for all values of i, j, and k. The model parameters m , σ_α^2 , σ_β^2 , σ_γ^2 , and σ_δ^2 are all unknown, and must be estimated from the available data.

Within the context of (6.1), Dannenburg *et alia* (1996, pages 94-103) describe in detail a complicated procedure with which to obtain a credibility estimate of the next observation $Y_{ij} \equiv X_{ij,K_{ij}+1}$ in each cell (i,j). Their credibility estimator of Y_{ij} takes the form :

$$m + z_{ij}(X_{ijw} - m) + z_i^\alpha(1 - z_{ij})(Y_{izw} - m) + z_j^\beta(1 - z_{ij})(Y_{zjw} - m) \;. \tag{6.2}$$

In this expression, the parameters z_i^α , z_j^β , and z_{ij} are the credibility factors associated with level i of the first risk factor, level j of the second, and cell (i,j), respectively. A weighted average of the observations in cell (i,j) is denoted by X_{ijw}, and specially adjusted weighted averages for the risk experiences of the i-th row and j-th column are denoted by Y_{izw} and Y_{zjw}, respectively. (The determination of the adjusted weighted averages entails solving a system of $I + J$ linear equations. In general, this system cannot be solved explicitly.) The credibility factors and the specially adjusted weighted averages are all functions of the model parameters m , σ_α^2 , σ_β^2 , σ_γ^2 , and σ_δ^2, which are regarded as fixed but unknown constants. Consequently, these parameters must be estimated before (6.2) can be evaluated. Dannenburg *et alia* (1996) suggest that a weighted average of the class averages can be used to estimate m, and note that the method of moments can be used to obtain many different estimators of the variance parameters. It follows that a number of different sets of credibility premiums can be determined. The set of credibility premiums determined by Dannenburg

et alia (1996) are reproduced in Table 6.3. They utilized the estimated model parameter values $\hat{m} = 242.51$, $\hat{\sigma}_\alpha^2 = (31.00)^2$, $\hat{\sigma}_\beta^2 = (51.44)^2$, $\hat{\sigma}_\gamma^2 = (5.86)^2$, and $\hat{\sigma}_\delta^2 = (164.28)^2$. As these model parameters are treated as fixed constants once estimated, the variability associated with the credibility premiums will be understated.

We now discuss how BUGS can be used to develop exact Bayesian premiums in accordance with a parametric fully Bayesian analysis of the following unbalanced two-way crossed classification model :

$$\begin{aligned} X_{ijk} &\sim \text{normal } (\mu_{ij}, \tau_{ijk}) \ , \\ \mu_{ij} &= m + \alpha_i + \beta_j + \gamma_{ij} \ , \\ \alpha_i &\sim \text{normal } (0, \tau_\alpha) \ , \\ \beta_j &\sim \text{normal } (0, \tau_\beta) \ , \\ \gamma_{ij} &\sim \text{normal } (0, \tau_\gamma) \ , \\ \tau_{ijk} &= \tau_\delta \ w_{ijk} \ , \\ w_{ijk} &\sim \text{gamma } (a_{ij}, b_{ij}) \ . \end{aligned} \tag{6.3}$$

We emphasize that this model is not quite identical to (6.1). In particular, we have now introduced normal distributions for the random parameters α_i, β_j, γ_{ij}, and δ_{ijk}. In order to maintain consistency with the BUGS language, the second parameter of each normal distribution is a precision or inverse variance so that $\sigma_\alpha^2 = \tau_\alpha^{-1}$, $\sigma_\beta^2 = \tau_\beta^{-1}$, $\sigma_\gamma^2 = \tau_\gamma^{-1}$, and $\sigma_\delta^2 = \tau_\delta^{-1}$. This means that the conditional variance of X_{ijk} in (6.3) decreases as the exposure w_{ijk} increases. The exposures w_{ijk} have themselves been modelled as independent draws from gamma distributions with parameters varying from class to class. This is an unnecessary step in this particular example as the exposures are all known and equal to 1, but would prove a useful feature of the model if some of the exposures were missing or otherwise unavailable.

Recall that the model parameters m , σ_α^2 , σ_β^2 , σ_γ^2 , and σ_δ^2, were previously regarded as fixed but unknown constants. In the parametric fully Bayesian setting they will all be modeled as random parameters in order to incorporate an additional level of parameter uncertainty in the development of exact Bayesian premiums. If no significant information

about these parameters is available *a priori*, then the following distributions might reasonably be adopted :

$$\begin{aligned}
m &\sim \text{normal } (\mu_m, \tau_m) \ , \\
\mu_m &\sim \text{normal } (0, 0.000001) \ , \\
\tau_m &\sim \text{gamma } (0.001, 0.001) \ , \\
\tau_\alpha &\sim \text{gamma } (0.001, 0.001) \ , \\
\tau_\beta &\sim \text{gamma } (0.001, 0.001) \ , \\
\tau_\gamma &\sim \text{gamma } (0.001, 0.001) \ , \\
\tau_\delta &\sim \text{gamma } (0.001, 0.001) \ , \\
a_{ij} &\sim \text{uniform } (0, 100) \ , \\
b_{ij} &\sim \text{uniform } (0, 100) \ .
\end{aligned}$$

The small precision associated with the normal distribution for μ_m effectively assigns this random parameter a prior distribution that is approximately locally uniform over the range of values it might reasonably be expected to take on (say, from 0 to 1000). The remaining hyperparameters τ_m, τ_α, τ_β, τ_γ, τ_δ, a_{ij}, and b_{ij} have also been assigned relatively noninformative priors. Any of these distributions may be replaced with more informative ones when appropriate. Our probabilistic model will be complete once we model the predictive draws. Assuming unit exposures, this is easily done as follows :

$$Y_{ij} \ \sim \ \text{normal } (\mu_{ij}, \tau_\delta) \ .$$

The predictive draws generated for each cell will be monitored, and these will be used to estimate each cell's exact Bayesian predictive premium.

The model described above can be modified in any number of ways. For example, it is possible to impose order restrictions on the row and / or column main effects. Requiring $\beta_1 < \beta_2 < \beta_3$ may make sense since those persons with more working experience, and correspondingly higher salaries and job security, are likely to have higher average loan amounts.

Another modification would be to require the parameter m to be drawn from a normal distribution that is truncated from below and / or above at reasonable values (say, 0 and 5000). These restrictions are easily coded via the `I(,)` construct in BUGS as shown below.

6.2 Model Specification and BUGS Files for the Credit Example

There is one minor difficulty with coding (6.3) in BUGS. Since the data is unbalanced, the array representing the values X_{ijk} will have a ragged form (i.e., the number of observations across the third dimension varies from cell to cell). To work around this, we will store all of the claims in a single vector X with length 401, and introduce vectors $c1$ and $c2$ in which we store the values of the first and second risk factors for each observed claim. Likewise, the vector w will store the exposures.

The graph for this example's probabilistic model is given in Figure 6.1. The complete files required to implement an analysis of the credit model using 'classic' BUGS are given below. They are presently maintained on the world wide web, along with a counterpart file credit.odc for use with WinBUGS, at www.math.ucalgary.ca/∼scollnik/abcd/.

6.2.1 The File credit.bug

```
model credit;

const
  N = 401,     # number of observations in total
  I = 3,       # number of categories in classification 1
  J = 3;       # number of categories in classification 2

var
  X[N],        # observations ( amounts paid by credit insurer )
  c1[N],       # category in classification 1 for each observation
  c2[N],       # category in classification 2 for each observation
  tau[N],      # precisions associated with X[N]
  w[N],        # exposures associated with X[N]
  a[I,J],      # parameters associated with exposures
  b[I,J],      # parameters associated with exposures

  Y[I,J],      # predictive draws from each cell
  mu[I,J],     # mean for observations in cell (i,j)

  m,           # main effect
  alpha[I],    # classification 1 effect
  beta[J],     # classification 2 effect
  gamma[I,J],  # classification interaction effect

  mu.m, tau.m, tau.alpha, tau.beta, tau.gamma, tau.delta,
  sigma2.m, sigma2.alpha, sigma2.beta, sigma2.gamma, sigma2.delta;

data X, c1, c2 in "credit.dat";      # data in tabular format
inits in "credit.in";

{

# Define the two-way crossed classification model structure.

  for( i in 1:N ){
    X[i] ~ dnorm( mu[c1[i],c2[i]], tau[i] );
    tau[i] <- tau.delta * w[i];
    w[i] ~ dgamma( a[c1[i],c2[i]], b[c1[i],c2[i]] );
    w[i] <- 1;
  }
```

```
  # Define the predictive draws and the class means.
  for( i in 1:I ){
    for( j in 1:J ){
      Y[i,j] ~ dnorm( mu[i,j], tau.delta );
      mu[i,j] <- m + alpha[i] + beta[j] + gamma[i,j];
    }
  }

  # Remove the comment to implement the order restriction.
  m ~ dnorm( mu.m, tau.m ); # I(0,5000);

  alpha[1] ~ dnorm( 0, tau.alpha );
  alpha[2] ~ dnorm( 0, tau.alpha );
  alpha[3] ~ dnorm( 0, tau.alpha );

  # remove the comments to implement the order restrictions
  beta[1] ~ dnorm( 0, tau.beta ); # I(,beta[2]);
  beta[2] ~ dnorm( 0, tau.beta ); # I(beta[1],beta[3]);
  beta[3] ~ dnorm( 0, tau.beta ); # I(beta[2],);

  for( i in 1:I ){
    for( j in 1:J ){
      gamma[i,j] ~ dnorm( 0, tau.gamma );
      a[i,j] ~ dunif( 0, 100 );
      b[i,j] ~ dunif( 0, 100 );
    }
  }

  mu.m ~ dnorm( 0, 0.000001 );
  tau.m ~ dgamma( 0.001, 0.001 );
  tau.alpha ~ dgamma( 0.001, 0.001 );
  tau.beta  ~ dgamma( 0.001, 0.001 );
  tau.gamma ~ dgamma( 0.001, 0.001 );
  tau.delta ~ dgamma( 0.001, 0.001 );

  sigma2.m <- 1 / tau.m;
  sigma2.alpha <- 1 / tau.alpha;
  sigma2.beta  <- 1 / tau.beta;
  sigma2.gamma <- 1 / tau.gamma;
  sigma2.delta <- 1 / tau.delta;

}
```

6.2.2 The Data File **credit.dat** (an excerpt)

```
 75.30  1  1
  7.72  1  1
133.23  1  1
     .  .  .
     .  .  .
     .  .  .
 54.92  2  1
164.22  2  1
396.50  2  1
554.12  2  1
163.77  2  1
     .  .  .
     .  .  .
     .  .  .
509.60  3  3
274.23  3  3
469.48  3  3
```

6.2.3 The Initial Value File **credit.in**

```
list( tau.alpha = 0.01, tau.beta = 0.01, tau.gamma = 0.01, tau.delta = 0.01,
      a = c(1,1,1,1,1,1,1,1,1), b = c(1,1,1,1,1,1,1,1,1) )
```

6.3 Output and Analysis

After a burn-in of 50,000 iterations taking 362 seconds, it took BUGS 412 seconds to run for an additional 50,000 iterations on a 150 MHz Pentium PC (WinBUGS took 766 and 776 seconds, respectively). The predictive draws Y_{ij} were monitored for each cell, and summary statistics for these parameters over the last 50,000 iterations appear in Table 6.4. The exact Bayesian premiums would usually be determined on the basis of the estimated predictive means, but the reported predictive probability intervals and standard deviations may also be useful in order to assess the variability associated with the predictive distributions. Also included in Table 6.4, are the summary statistics for the parameters m, σ^2_α, σ^2_β, σ^2_γ, and σ^2_δ. The summary statistics appearing in Table 6.4 are not directly comparable with the estimates given by Dannenburg *et alia* (1996), since the models under the two corresponding analyses are not identically interpreted, but there is definately a close correspondence.

6.4 Further Discussion of the Credit Example

According to Schnieper (1995) :

> In practical applications of Credibility Theory the structure parameters usually have to be estimated from the data. This leads to an estimator of the *a posteriori* mean which is often biased and where the credibility factor depends on the data. A more coherent approach to the problem would be to also treat the unknown parameters as random variables and to simultaneously estimate the *a posteriori* mean and the structure parameters.

The Bayesian analysis of the credit example discussed above takes this advice to heart.

A general discussion of Bayesian inference for cross classification models can be found in Box and Tiao (1973; see chapter 6).

Table 6.1

Amounts Paid by Credit Insurer for Debtors, by Marital Status ($i = 1, 2, 3$) and Time Worked with Current Employer ($j = 1, 2, 3$).

$i = 1,\ j = 1$									
75.30	7.72	133.23	202.63	223.68	199.95	49.62	63.97	293.02	218.24
242.69	113.68	196.30	380.65	388.63	326.63	91.41	326.63	106.38	266.48
156.06	43.80	484.22	90.07	27.06	122.02	80.97	61.24	133.68	445.67
265.84	231.10	10.29	325.23	68.63	87.61	84.79	27.48	314.31	248.60
$i = 2,\ j = 1$									
54.92	164.22	396.50	554.12	163.77	52.67	77.99	65.40	102.35	107.72
79.17	208.78	106.38	268.77	209.69	80.80	166.02	148.62	71.84	84.18
254.47	97.79	80.97	665.58	73.22	109.34	154.87	184.61	55.57	173.10
7.06	486.38	176.24	62.66	67.74	26.78	305.27	213.05	47.18	87.88
321.16	86.70	425.24	300.52	69.13	141.89	134.91	118.57	180.37	106.38
491.71	5.05	321.15	93.98						
$i = 3,\ j = 1$									
363.10	267.58	66.82	93.05	295.78	59.36	313.61	395.56	118.71	150.95
371.92	166.47	312.18	103.37	452.58	293.79	286.09	96.27	403.13	540.88
523.66	103.35	140.82	196.93	298.89	96.28	187.40	136.81	24.92	94.36
261.02	109.34	81.57	150.11	87.44	137.64	113.57	332.67	51.60	

Table 6.1 (cont.)

Amounts Paid by Credit Insurer for Debtors, by Marital Status ($i = 1, 2, 3$) and Time Worked with Current Employer ($j = 1, 2, 3$).

$i = 1,\ j = 2$									
158.72	616.01	405.37	53.58	123.04	34.98	316.44	323.07	364.56	221.55
165.78	477.92	107.48	253.04	406.20	58.31	77.93	103.37	400.06	550.82
101.68	358.21	56.18	513.20	82.79	151.52	342.59	402.08	413.70	257.69
47.99	638.10	97.31	133.48	113.57	177.84	406.86	225.10	234.76	406.86
56.09	25.05	147.45							
$i = 2,\ j = 2$									
513.14	40.35	529.67	260.59	425.27	163.77	640.47	53.00	272.63	81.67
378.23	494.40	264.02	95.01	316.37	133.37	532.93	145.99	42.67	157.20
202.68	331.85	27.32	563.61	273.65	97.04	150.13	332.02	184.14	102.29
398.88	77.34	47.94	319.02	97.72	416.72	456.57	107.96	150.90	483.78
134.27	147.06	192.16	149.56	309.78	98.96	111.04	166.00	56.23	134.27
46.66	228.23	194.81							
$i = 3,\ j = 2$									
173.69	63.47	414.06	205.19	295.89	536.37	48.29	74.51	100.01	20.97
403.24	221.20	196.30	400.06	405.70	203.34	398.01	90.92	609.78	12.82
142.28	608.41	311.53	496.28	268.45	111.55	168.30	33.02	84.40	350.82
497.90	12.11	122.62	457.48	168.02	309.32	381.98	748.68	366.05	

Table 6.1 (cont.)

Amounts Paid by Credit Insurer for Debtors, by Marital Status ($i = 1, 2, 3$) and Time Worked with Current Employer ($j = 1, 2, 3$).

$i = 1,\ j = 3$									
293.78	185.62	488.89	314.31	46.47	49.86	483.03	73.09	527.52	214.25
37.12	295.07	447.74	242.96	298.26	222.98	219.79	539.43	22.68	420.87
155.79	34.23	170.73	252.84	84.31	358.32	287.12	411.58	434.72	502.44
247.09	157.63	145.27	314.85	400.06	56.56	210.21	251.92	561.44	117.82
145.92									
$i = 2,\ j = 3$									
215.06	440.24	48.16	10.29	382.19	374.19	563.36	271.07	37.10	36.43
324.44	133.16	392.61	135.26	238.92	313.01	388.53	139.42	77.37	314.01
60.60	134.11	163.87	290.97	361.60	414.72	529.82	471.13	272.92	520.18
258.75	232.75	94.52	421.51	487.45	380.65	416.76	277.07	214.37	157.63
89.62	36.85	99.73	290.97	78.43	209.32	209.51	143.69		
$i = 3,\ j = 3$									
48.56	57.06	518.60	287.28	480.38	423.87	711.86	259.00	505.18	177.51
394.34	145.99	95.04	110.22	463.98	129.01	469.67	516.97	203.34	386.37
103.11	331.60	105.88	643.38	724.22	648.69	783.54	814.95	251.56	685.96
162.40	602.69	203.00	591.94	239.66	221.54	182.72	297.93	556.78	208.54
133.16	509.60	274.23	469.48						

Table 6.2

Average Amounts Paid by the Credit Insurer.

	Number of Years a Debtor Has Worked With Current Employer:		
Marital Status:	< 2 (j=1)	2 to 10 (j=2)	≥ 10 (j=3)
single (i=1)	180.39	246.71	261.58
divorced (i=2)	172.04	232.67	253.22
otherwise (i=3)	212.30	269.57	366.61

Table 6.3

Credibility Estimates from Dannenburg *et alia* (1996).

	Number of Years a Debtor Has Worked With Current Employer:		
Marital Status:	< 2 (j=1)	2 to 10 (j=2)	≥ 10 (j=3)
single (i=1)	181.05	238.44	277.33
divorced (i=2)	172.24	229.38	268.40
otherwise (i=3)	224.76	281.38	324.49

Table 6.4

Parametric Fully Bayesian Posterior Summary Statistics for Certain Model Parameters of Interest.

	Estimates from BUGS				
Parameter	Mean	SD	2.5%	Median	97.5%
Y_{11}	186.2	166.5	-140.9	186.5	512.8
Y_{12}	241.4	166.1	-82.79	241.3	568.2
Y_{13}	274.6	166.2	-53.05	274.8	600.8
Y_{21}	177.7	166.1	-145.8	178.2	500.2
Y_{22}	234.9	166.9	-93.18	234.3	562.4
Y_{23}	267.2	166.8	-60.32	266.6	594.2
Y_{31}	215.4	166.5	-113.2	214.9	542.3
Y_{32}	270.5	166.5	-57.92	270.4	598.0
Y_{33}	325.7	167.2	-2.801	325.7	655.0
m	241.8	57.4	109.1	242.7	365.2
σ^2_α	3635.0	56330.0	0.00298	380.6	21750.0
σ^2_β	10470.0	72870.0	0.01791	2457.0	63020.0
σ^2_γ	828.3	1884.0	0.00169	21.61	5649.0
σ^2_δ	27210.0	1950.0	23630.0	27130.0	31270.0

Figure 6.1

Graphical Model for the Credit Example.

7 Health : A Normal Model for Order Restricted Graduation

7.1 Introduction

Following the lead of Broffit's papers on the subject (1984, 1986, 1988), Carlin (1992c) described the Bayesian analysis of two models for graduation incorporating order restrictions on the parameters of interest. Carlin's first model has a normal-normal distributional structure and was applied to a set of observed health claim cost aging factors. We are interested in reproducing Carlin's analysis with the aid of BUGS. His model will be described momentarily. The data appears in Table 7.1. At $N = 13$ ages, we have available raw health claim cost aging factors. These raw aging factors y_i were computed from average claim cost data presented in a paper by Hutchings and Ullman (1983) based on 676,000 Blue Cross and Blue Shield of Greater New York contracts for the calendar year 1978. These factors are unisex and are further described in Carlin (1992c). Basically, these factors describe the increase in health care cost as an individual ages. If $C_{[x_i-n_1,x_i]}$ and $C_{[x_i,x_i+n_2]}$ represent true monthly costs per person for two consecutive age brackets having lengths n_1 and n_2 years, respectively, and we represent these age intervals by their midponts, then the i-th true aging factor θ_i satisfies the relation

$$(1+\theta_i)^{\frac{n_1+n_2}{2}} \;=\; \frac{C_{[x_i-n_1,x_i]}}{C_{[x_i,x_i+n_2]}} \;.$$

An estimate y_i of θ_i may be obtained by replacing the true but unknown costs C in the expression above with estimated costs $\hat{C}$ from employee claim cost data. Then the estimate y_i would be defined by

$$y_i \;=\; \left(\frac{\hat{C}_{[x_i-n_1,x_i]}}{\hat{C}_{[x_i,x_i+n_2]}}\right)^{\frac{2}{n_1+n_2}} \;-\; 1 \;.$$

These are the raw aging factors appearing in Table 7.1 for the data under study.

The true aging factors θ_i describe increases in medical costs due to the aging of a population. As such, these factors should generally be positive. Carlin (1992c) notes that some health-care professionals further believe that the true aging factors should be near zero at the

very youngest ages and also at the oldest. These prior opinions suggest that the sequence of true aging factors should start small but positive at the early ages, increase up to some age near retirement with factor θ_s, and then decrease smoothly until the end of the table where the factors are again small but positive. We may also wish to impose some upper bound B that no θ_i may exceed. Although these prior opinions nicely describe the behaviour of the true aging factors, the true factors θ_i are not actually observed : only the estimates y_i are observed, and they are likely to form an irregular sequence at odds with the prior opinion. The observations are reconciled with the prior opinion through the process of graduation.

The model under consideration here supposes that the observed aging factors y_i arise from normal populations with means θ_i and common precision (i.e., inverse variance) σ. The prior distribution for the true aging factors θ_i is taken to be a product of independent and identical normal distributions with common mean μ and precision τ, with imposed constraints that $0 < \theta_1 < \ldots < \theta_s < B > \theta_s > \theta_{s+1} > \ldots > \theta_N > 0$. That is,

$$\begin{aligned} y_i &\sim \text{normal } (\theta_i, \sigma) \ , \ i = 1, \ldots, N, \\ \theta_i &\sim \text{normal } (\mu, \tau) \ , \ i = 1, \ldots, N, \end{aligned}$$

with $0 < \theta_1 < \ldots < \theta_s < B > \theta_s > \theta_{s+1} > \ldots > \theta_N > 0$. Notice that this model permits observed aging factors to take on negative values - thus accommodating y_{12} in Table 7.1, for instance - but the unobserved true aging factors are restricted to positive values. Specification of the parameters B, s, μ, σ, and τ remains. As in Carlin (1992c), we accept $B = 0.15$ as a reasonable upper bound for the aging factors and also adopt age 60 (corresponding to $s = 7$) as the age at which the maximum aging factor will appear. For the remaining parameters, we will adopt the following hyperprior distributions :

$$\begin{aligned} \mu &\sim \text{normal } (\alpha, \beta) \ , \\ \sigma &\sim \text{gamma } (a_1, b_1) \ , \\ \tau &\sim \text{gamma } (a_2, b_2) \ . \end{aligned}$$

We consider three different hyperprior parameter specifications labelled Smooth 1, Smooth

2, and Smooth 3, respectively :

Smooth 1 : $\alpha = 0.035$, $\beta = 400$, $a_1 = a_2 = 3.0$, $b_1 = 1/1250$, $b_2 = 1/1250$,

Smooth 2 : $\alpha = 0.035$, $\beta = 400$, $a_1 = a_2 = 3.0$, $b_1 = 1/50$, $b_2 = 1/1250$,

Smooth 3 : $\alpha = 0.035$, $\beta = 400$, $a_1 = a_2 = 3.0$, $b_1 = 1/50$, $b_2 = 1/50$.

As we will see, these hyperprior specifications reflect an increasing variability in the prior beliefs that translates into increasingly variable aging factors. (It should be noted that while our hyperprior parameter specifications are equivalent to those in Carlin (1992c), our parametrization corresponds with that of BUGS. In particular, whereas Carlin would say that v is a random variance parameter with an inverse (reciprocal) gamma (a, b) distribution, we say that the random precision parameter $u = 1/v$ is gamma (a, $1/b$) as defined in BUGS.)

The graph for this example's probabilistic model is given in Figure 7.1. Note that the graph is simplified insomuch as the various order restrictions on the parameters are not explicitly shown, but are nevertheless understood to be present. The complete files required to implement an analysis of the health model using 'classic' BUGS are given below. They are presently maintained on the world wide web, along with a counterpart file health.odc for use with WinBUGS, at www.math.ucalgary.ca/∼scollnik/abcd/.

7.2 Model Specification and BUGS Files for the **Health** Example

In the file health.bug , note the frequent use of `I(,)` to restrict the range of permissible values for the `theta[i]` parameters. We found it necessary to insert starting values for the `theta[i]` parameters in the file health.in in order to avoid termination of the BUGS program with the error "`Unable to generate initial values for this model`". These starting values should satisfy the same order restrictions mentioned above.

7.2.1 The File **health.bug**

```
model health;

const
  N = 13,       # number of age classes
  s = 7,        # age class presumed to have the largest HCCAF
  B = 0.15;     # presumed largest HCCAF

var
  y[N],         # observed health claim cost aging factors

  theta[N],     # conditional means of y
  mu,           # conditional means of theta
  sigma,        # common precision for y
  tau,          # common precision for theta

  lbound,       # lower bound for theta[s]

  a1, b1,       # hyperparameters for sigma
  a2, b2;       # hyperparameters for tau

data in "health-s.dat";    # data in S-PLUS format
inits in "health.in";
```

```
{

# Model structure for normal-normal modelling of the HCCAFs.

  for( i in 1:N ){
    y[i] ~ dnorm( theta[i], sigma );
  }

  theta[1] ~ dnorm( mu, tau ) I( 0, theta[2] );
  for( i in 2:(s-1) ) {
    theta[i] ~ dnorm( mu, tau ) I( theta[i-1], theta[i+1] );
  }
  lbound <- max( theta[s-1], theta[s+1] );
  theta[s] ~ dnorm( mu, tau ) I( lbound, B );
  for( i in (s+1):(N-1) ) {
    theta[i] ~ dnorm( mu, tau ) I( theta[i+1], theta[i-1] );
  }
  theta[N] ~ dnorm( mu, tau ) I( 0, theta[N-1] );

  mu ~ dnorm( 0.035, 400 );
  sigma ~ dgamma( a1, b1 );
  tau ~ dgamma( a2, b2 );

# Define the hyperprior parameters according to the graduation desired.

  a1 <- 3;
  a2 <- 3;

# Smooth 1
  b1 <- 1 / 1250;
  b2 <- 1 / 1250;

# Smooth 2
# b1 <- 1 / 50;
# b2 <- 1 / 1250;

# Smooth 3
# b1 <- 1 / 50;
# b2 <- 1 / 50;

}
```

7.2.2 The Data File health-s.dat

```
list( y = c( 0.0047, 0.0341, 0.0403, 0.0503, 0.0467,
             0.0184, 0.0713, 0.0454, 0.0390, 0.0244,
             0.0240, -0.0021, 0.0001 ) )
```

7.2.3 The Initial Value File health.in

```
list( theta = c( 0.003, 0.004, 0.005, 0.006, 0.007,
                 0.008, 0.009, 0.008, 0.007, 0.006,
                 0.005, 0.004, 0.003 ),
      mu = 0.035, sigma = 1, tau = 1 )
```

7.3 Output and Analysis

For each of the three hyperprior parameter specifications (Smooth 1, Smooth 2, and Smooth 3), it took BUGS about 8 seconds to run for 20,000 additional iterations after a burn-in of 10,000 iterations on a 150 MHz Pentium PC. The true aging factors were monitored over the last 20,000 iterations of each specification's simulation, and the summary posterior statistics for these aging factors under the first specification appear in Table 7.2 (like summary statistics were also calculated for the other two prior specifications, but will not be reported here). As with any Bayesian graduation, a question still remains : namely, which aspect of the posterior distribution will define our graduated values ? The posterior mean is a common choice and is the one we adopt. In Figure 7.2, we have plotted the raw aging factors and the graduated factors (i.e., the estimated posterior expected values of the true aging factors) resulting under each of the three hyperprior parameter specifications. Our figure precisely mirrors the graph of graduated aging factors appearing as Figure 1 in Carlin (1992c).

7.4 Further Discussion of the **Health** Example

The order restricted Bayesian graduation of health claim cost aging factors appearing in Carlin (1992c) was quickly and easily reproduced using BUGS. It is worthwhile to note that this method is easily adapted when health care costs arising from more than just one study are available. Let us suppose that $k = 3$ studies have yielded observed aging factors at $N = 5$ different ages, with the understanding that not all studies necessarily cover the same ages. In the health.bug file, we would need to change the lines

```
const
  N = 13,       # number of age classes
  s = 7,
var
  y[N],
```

to

```
const
  N = 5,        # number of age classes
  k = 3,        # number of studies
  s = 3,        # suppose the 3rd class has the largest HCCAF

var
  y[N,k],
```

and the lines

```
  for( i in 1:N ){
    y[i] ~ dnorm( theta[i], sigma );
  }
```

to

```
  for( i in 1:N ){
    for( j in 1:k ){
      y[i,j] ~ dnorm( theta[i], sigma );
    }
  }
```

Of course, the files health.dat and health.in would also need to be updated. For instance, the data file might look like

```
list( y = structure(
          .Data = c( 0.0039, 0.0219, 0.0325, 0.0471, 0.0325,
                     0.0047, 0.0341, 0.0403, NA,     NA,
                     NA,     NA,     0.0448, 0.0422, 0.0313 ),
          .Dim = c( 5, 3 ) )
```

The symbol `NA` appears in this file in those spots corresponding to aging factors not observed in particular studies. In this hypothetical example, the second study only observed aging factors at the first three ages whereas the third study only observed them at the last three. The file of starting values might look like

```
list( theta = c( 0.003, 0.004, 0.005, 0.004, 0.003 ),
      mu = 0.035, sigma = 1, tau = 1 )
```

Of course, and as the reader has undoubtedly noted, the graduation methodology just described is a general one and is applicable not just for graduating aging factors in the context of a health care cost study.

Table 7.1

Observed Health Claim Cost Aging Factors.

i	age_i	y_i	i	age_i	y_i
1	17.50	0.0047	8	70.00	0.0454
2	30.00	0.0341	9	75.00	0.0390
3	38.75	0.0403	10	80.00	0.0244
4	45.00	0.0503	11	85.00	0.0240
5	50.00	0.0467	12	90.00	-0.0021
6	55.00	0.0184	13	95.00	0.0001
7	60.00	0.0713			

Table 7.2

Estimated Posterior Summary Statistics for the Health Claim Cost Aging Factors (HCCAFs) Under the First Hyperprior Parameter Specification.

	Estimated Posterior Statistics from BUGS				
HCCAF	Mean	SD	2.5%	Median	97.5%
θ_1	0.0139	0.0074	0.0012	0.0136	0.0286
θ_2	0.0250	0.0067	0.0111	0.0252	0.0376
θ_3	0.0313	0.0062	0.0189	0.0313	0.0433
θ_4	0.0364	0.0062	0.0243	0.0364	0.0484
θ_5	0.0405	0.0063	0.0285	0.0405	0.0532
θ_6	0.0449	0.0068	0.0322	0.0447	0.0587
θ_7	0.0597	0.0096	0.0428	0.0590	0.0801
θ_8	0.0455	0.0078	0.0310	0.0452	0.0615
θ_9	0.0373	0.0070	0.0243	0.0371	0.0519
θ_{10}	0.0300	0.0067	0.0175	0.0299	0.0435
θ_{11}	0.0237	0.0066	0.0112	0.0236	0.0371
θ_{12}	0.0158	0.0066	0.0040	0.0156	0.0296
θ_{13}	0.0090	0.0061	0.0005	0.0082	0.0226

Figure 7.1

Graphical Model for the Health Example.

Figure 7.2

Original and Graduated Health Claim Cost Aging Factors.

8 Broff : Increasing Ordered Graduation of Mortality Rates

8.1 Introduction

Broffit (1988) described a model for the Bayesian graduation of mortality rates subject to one of two restrictions - either that the force of mortality is increasing or that it is increasing and convex. We are interested in considering a Bayesian analysis of this model's first variation with the use of BUGS. We note that Carlin (1992c) has previously discussed how one might implement the Bayesian analysis of either model variation using the Gibbs sampler, and we will make specific references to Carlin's work later on.

The context of the problem under consideration is as follows. Let $q_x, \ldots, q_{x+k-1}$ be the mortality rates at ages $x, \ldots, x+k-1$ for a particular population. Suppose that data available from a mortality study of a group of independent lives from this population under observation within these age intervals include values for d_i, the number of deaths observed between ages $x+i-1$ and $x+i$, and e_i, the exposure or total number of years the lives were under observation between ages $x+i-1$ and $x+i$. Under a model that assumes independent random times to death and withdrawal along with a constant force of mortality θ_i in effect within each unit year interval between ages $x+i-1$ and $x+i$, Broffit (1988) obtains the likelihood function

$$L(\theta_1, \ldots, \theta_k \mid Obs.\ Data) \ \propto \ \prod_{i=1}^{k} {\theta_i}^{d_i} \exp(-e_i\theta_i) \ . \tag{8.1}$$

This likelihood function is equivalent to that obtained by assuming that each d_i given e_i has an independent Poisson distribution with mean $e_i\theta_i$. The unrestricted maximum likelihood estimate of θ_i is $\theta_i^M = d_i/e_i$. These unrestricted maximum likelihood estimates serve as our 'raw' or ungraduated estimates. We wish to produce a graduated sequence of estimates that conform to the increasing condition $0 < \theta_1 < \ldots < \theta_k < B$. This is a reasonable assumption for human ages $x \geq 30$ or so. We assign the label (I) to this increasing condition.

In his analysis, Broffit (1988) originally assumed each θ_i to be independently generated from a gamma (α_i, β_i) distribution,

$$\theta_i \ \sim \ \text{gamma}\ (\alpha_i, \beta_i)\ ,\ i = 1, \ldots, k\ , \tag{8.2}$$

thus yielding the posterior distribution

$$f(\theta_1, \ldots, \theta_k \mid \textit{Obs. Data}) \ \propto \ \prod_{i\,=\,1}^{k} \ {\theta_i}^{\alpha_i + d_i - 1} \ \exp(-[\beta_i + e_i]\ \theta_i) \tag{8.3}$$

or

$$f(\theta_i \mid \textit{Obs. Data}) \ \sim \ \text{gamma}\ (\alpha_i^*, \beta_i^*)\ ,\ i = 1, \ldots, k\ , \tag{8.4}$$

with $\alpha_i^* = \alpha_i + d_i$ and $\beta_i^* = \beta_i + e_i$. As noted in Carlin (1992c, page 59), imposing the increasing condition (I) on the prior (8.2) simply constrains the posterior (8.3) or (8.4) in the same manner. In BUGS, there are two ways in which we can code the model developed thus far. Using (8.4) directly with the increasing condition (I) suggests the BUGS code :

```
theta[1] ~ dgamma( alphastar[1], betastar[1] ) I(,theta[2]);
for( i in 2:(N-1) ){
  theta[i] ~ dgamma( alphastar[i], betastar[i] ) I(theta[i-1],theta[i+1]);
}
theta[N] ~ dgamma( alphastar[N], betastar[N] ) I(theta[N-1],B);

for( i in 1: N ){
  alphastar[i] <- alpha[i] + d[i];
  betastar[i] <- beta[i] + e[i];
}
```

On the other hand, the Poisson model interpretation of the likelihood and (8.2) with the increasing condition (I) speaks to the BUGS code :

```
for( i in 1:k ){
  d[i] ~ dpois( lambda[i] );
  lambda[i] <- e[i] * theta[i];
}

theta[1] ~ dgamma( alpha[1], beta[1] ) I(,theta[2]);
for( i in 2:(k-1) ){
  theta[i] ~ dgamma( alpha[i], beta[i] ) I(theta[i-1],theta[i+1]);
}
theta[k] ~ dgamma( alpha[k], beta[k] ) I(theta[k-1],B);
```

Both codings effectively describe the same model. We adopt the second since it is a more explicit description. Note the use of `I(,)` to restrict the range of permissible values for the `theta[i]` parameters. Specification of the parameters α_i and β_i, for $i = 1, \ldots, k$, remains. Some ways in which this specification might proceed are discussed in the context of a particular example below.

The mortality data we are interested in graduating is given in Table 8.1 and originally appeared in Broffit (1988). The data represent male ultimate (duration ≥ 16) experience for premium-paying policies with face amounts between \$10,000 and \$24,900. Prior values of the forces of mortality, θ^P, are also available and are based on a graduation from an earlier mortality study involving policies similar to those comprising the present data set. The data will be graduated in accordance with a Bayesian analysis of Broffit's model subject to the increasing condition (I). The analysis will proceed under four different illustrative prior density specifications. In each case, our graduated values of θ_i, for $i = 1, \ldots, k$, will be set equal to the estimated posterior expected values. The four graduations will be referred to as Smooths 1, 2, 3, and 4, and their corresponding prior density specifications are as follow :

Smooth 1. If we were to suppose that, prior to the imposition of the increasing condition (I), the individual θ_i were each independent draws from a common gamma (α, β) distribution, then it would not be unreasonable to treat the prior graduated values as like draws. We estimate the unknown prior parameters α and β by equating the sample mean and variance of the prior graduated values to the gamma population mean and variance of α/β and α/β^2, respectively. This process yields $\alpha = 1.5079$ and $\beta = 229.3094$. In this manner, we incorporate an aspect of the information provided by the prior graduated values into the analysis. Finally, we set $B = 0.025$ (recall that B is the upper bound for θ_k) in order to complete the model specification.

Smooth 2. As in Smooth 1, with $B = 0.020$.

Smooth 3. Prior to the imposition of the increasing condition (I), we suppose that θ_i and θ_i^P are independent draws from the same gamma (α_i, β_i) distribution, for $i = 1, \ldots, k$. By

explicitly coding each prior graduated value θ_i^P as a draw from a gamma (α_i, β_i) distribution in BUGS, we will incorporate the prior knowledge provided by these values into the analysis. To each parameter α_i and β_i, we assign weakly informative and independent gamma (1, 0.01) priors (i.e., independent exponential distributions with a common mean of 100). We set $B = 0.025$.

Smooth 4. As in Smooth 3, with $B = 0.020$.

The four specifications described above are illustrative and the graduator should only treat them as possible guides. Other specifications are entirely possible. For instance, Carlin (1992c, pages 69-70) describes the following empirical Bayes approach to the problem. Suppose, as in Smooths 1 and 2, that each θ_i is a draw from a common gamma (α, β) distribution. Then a method of moments argument can be used to determine α and β. Specifically, note that the first two sample moments of the unrestricted maximum likelihood estimates are given by $\overline{\theta}^M = \sum_{i=1}^k \theta_i^M/k$ and $S_M^2 = \sum_{i=1}^k (\theta_i^M - \overline{\theta}^M)^2/(k-1)$, while the corresponding population moments in the unconstrained marginal family $f(\theta_i^M \mid \alpha, \beta)$ are

$$E(\theta_i^M) \;=\; \frac{1}{e_i} E(d_i) \;=\; \frac{1}{e_i} E\left[E(d_i \mid \theta_i)\right] \;=\; \frac{1}{e_i} E(e_i \theta_i) \;=\; \frac{\alpha}{\beta}$$

and

$$\begin{aligned} Var(\theta_i^M) &= \frac{1}{e_i^2} Var(d_i) \;=\; \frac{1}{e_i^2}\left\{Var\left[E(d_i \mid \theta_i)\right] \;+\; E\left[Var(d_i \mid \theta_i)\right]\right\} \\ &= \frac{1}{e_i^2}\left\{Var(e_i\theta_i) \;+\; E(e_i\theta_i)\right\} \\ &= \frac{\alpha}{\beta^2} \;+\; \frac{\alpha}{e_i \beta}\,. \end{aligned}$$

These population moments were derived using the Poisson distribution of d_i given θ_i and e_i and the gamma distribution of θ_i given α and β. Now, set

$$\overline{\theta}^M = \frac{\alpha}{\beta}$$

and

$$S_M^2 \;=\; \frac{1}{k} \sum_{i=1}^{k} \left\{ \frac{\alpha}{\beta^2} \;+\; \frac{\alpha}{e_i \beta} \right\}\,.$$

Solving this system of two equations and two unknowns yields values $\alpha = 1.4928$ and $\beta = 230.4133$. Since these are nearly identical to the values of α and β associated with Smooths 1 and 2, we will not consider this specification any further.

The graph for this example's probabilistic model corresponding to Smooths 3 and 4 is given in Figure 8.1. Note that the graph is simplified insomuch as the various order restrictions on the parameters are not explicitly shown, but are nevertheless understood to be present. The complete files required to implement an analysis of the broff model using 'classic' BUGS are given below. They are presently maintained on the world wide web, along with counterpart files broffS12.odc and broffS34.odc for use with WinBUGS, at www.math.ucalgary.ca/∼scollnik/abcd/.

8.2 Model Specification and BUGS Files for the Broff Example

In the file broff.bug , note the frequent use of `I(,)` to restrict the range of permissible values for the `theta[i]` parameters. We found it necessary to insert starting values for the `theta[i]` parameters in the file broff.in in order to avoid termination of the BUGS program with the error "`Unable to generate initial values for this model`". These starting values should satisfy the increasing condition (I). Starting values for the `alpha[i]` and `beta[i]` parameters are not required for Smooths 3 and 4, but specifying reasonable values can dramatically improve the time spent by BUGS on the first 10 or so iterations.

8.2.1 The File broff.bug

```
model broff;

const
  k = 30,       # number of age classes
  B = 0.025;    # an upper bound for theta[k]

var
  d[k], e[k],   # observed deaths and exposures
  theta[k],     # underlying forces of mortality
  lambda[k],    # Poisson means for d[k]
  thetap[k],    # prior values for theta[k]

  alpha[k],     # prior density parameters
  beta[k];      # prior density parameters

data in "broff-s.dat";    # data in S-PLUS format
inits in "broff.in";

{

# Model structure for observed deaths.
#
  for( i in 1:k ){
    d[i] ~ dpois( lambda[i] );
    lambda[i] <- e[i] * theta[i];
  }
```

```
# Model structure for increasing thetas.

  theta[1] ~ dgamma( alpha[1], beta[1] ) I(,theta[2]);
  for( i in 2:(k-1) ){
    theta[i] ~ dgamma( alpha[i], beta[i] ) I(theta[i-1],theta[i+1]);
  }
  theta[k] ~ dgamma( alpha[k], beta[k] ) I(theta[k-1],B);

# Remainder of model description, for Smooths 1 and 2.
#
# Also, adjust the value of B in the const declarations.
#
  for( i in 1:k ){
    alpha[i] <- 1.507857773;
    beta[i] <- 229.3094159;
  }

# Remainder of model description, for Smooths 3 and 4.
#
# Also, adjust the value of B in the const declarations.
#
# for( i in 1:k ){
#   thetap[i] ~ dgamma( alpha[i], beta[i] );
#   alpha[i] ~ dexp( 0.01 ); # dgamma( 1, 0.01 );
#   beta[i] ~ dexp( 0.01 ); # dgamma( 1, 0.01 );
# }

}
```

8.2.2 The Data File **broff-s.dat**

```
list( d = c(  3,  1,  3,  2,  2,
              4,  4,  7,  5,  2,
              8, 13,  8,  2,  7,
              4,  7,  4,  4, 11,
             11, 13, 12, 12, 19,
             12, 16, 12,  6, 10 ),

      e = c( 1771.5, 2126.5, 2743.5, 2766.0, 2463.0,
             2368.0, 2310.0, 2306.5, 2059.5, 1917.0,
             1931.0, 1746.5, 1580.0, 1580.0, 1467.5,
             1516.0, 1371.5, 1343.0, 1304.0, 1232.5,
             1204.5, 1113.5, 1048.0, 1155.0, 1018.5,
              945.0,  853.0,  750.0,  693.0, 594.0 ),

 thetap = c( 0.0012308, 0.0012808, 0.0013609, 0.0014811, 0.0016213,
             0.0017816, 0.0019519, 0.0021423, 0.0023628, 0.0026134,
             0.0028942, 0.0031951, 0.0035362, 0.0039377, 0.0044097,
             0.0049422, 0.0054850, 0.0060382, 0.0066017, 0.0072663,
             0.0080523, 0.0090710, 0.0101210, 0.0111823, 0.0122548,
             0.0133386, 0.0145047, 0.0158753, 0.0174514, 0.0192848 ) )
```

8.2.3 The Initial Value File **broff.in**

```
list( theta = c( 0.0001, 0.0002, 0.0003, 0.0004, 0.0005,
                 0.0006, 0.0007, 0.0008, 0.0009, 0.0010,
                 0.0011, 0.0012, 0.0013, 0.0014, 0.0015,
                 0.0016, 0.0017, 0.0018, 0.0019, 0.0020,
                 0.0021, 0.0022, 0.0023, 0.0024, 0.0025,
                 0.0026, 0.0027, 0.0028, 0.0029, 0.0030 )
#
#   , alpha = c( 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
#                1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
#                1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ),
#
#     beta = c( 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,
#               100, 100, 100, 100, 100, 100, 100, 100, 100, 100,
#               100, 100, 100, 100, 100, 100, 100, 100, 100, 100 )
#
      )
```

8.3 Output and Analysis

The times it took BUGS to run for 25,000 iterations on a 150 MHz Pentium PC are described in Table 8.2. Times are given for the analysis under each of the specifications considered (i.e., Smooths 1, 2, 3, and 4). These are running times under 'classic' BUGS with only the `theta[i]` parameters initialized in the broff.in file. Since most of the running time is spent on the early iterations, it may be advantageous to undertake a more careful selection of the initial values if possible. The use of prior graduated values might be explored in this context. We also coded the same four model specifications in WinBUGS, and it is interesting to note that it took WinBUGS significantly less time than 'classic' BUGS to update each of the probabilistic models under consideration. The running times under WinBUGS are given in Table 8.3.

For each smooth, we adopted the estimated posterior expected values of the `theta[i]` parameters as our graduated forces of mortality. These estimated posterior expected values were based on the output from the final 5000 iterations in BUGS. The resulting graduations are depicted in Figure 8.2 (along with the original unrestricted maximum likelihood estimates), and illustrate what a range of possible graduations under Broffit's model subject to the increasing condition (I) can look like. Of course, since $q_{x+i-1} = 1 - \exp(-\theta_i)$, for $i = 1, \ldots, k$, under the present assumption of a constant force of mortality within each year of age, it is a simple matter to transform our original and graduated forces of mortality to mortality rates if so desired.

8.4 Further Discussion of the **Broff** Example

In addition to the increasing condition (I), Broffit (1988) and Carlin (1992c) also considered the imposition of an increasing and convex condition on the values of the forces of mortality. We tried implementing this model in BUGS and effectively failed : BUGS was unable to update the model for even a single iteration within a reasonable amount of time. It is possible that success still may obtain with BUGS on a faster computer. Otherwise, a Gibbs sampler implementing Broffit's model subject to the increasing and convex condition may need to be coded by the practitioner directly in a language like APL, J, Fortan, or C. See Carlin (1992c) for a more detailed description of this problem and its solution.

Table 8.1

Raw Mortality Data and Prior Values of θ.

i	age_i	d_i	e_i	θ_i^P	θ_i^M
1	35	3	1771.5	0.0012308	0.0016935
2	36	1	2126.5	0.0012808	0.0004703
3	37	3	2743.5	0.0013609	0.0010935
4	38	2	2766.0	0.0014811	0.0007231
5	39	2	2463.0	0.0016213	0.0008120
6	40	4	2368.0	0.0017816	0.0016892
7	41	4	2310.0	0.0019519	0.0017316
8	42	7	2306.5	0.0021423	0.0030349
9	43	5	2059.5	0.0023628	0.0024278
10	44	2	1917.0	0.0026134	0.0010433
11	45	8	1931.0	0.0028942	0.0041429
12	46	13	1746.5	0.0031951	0.0074435
13	47	8	1580.0	0.0035362	0.0050633
14	48	2	1580.0	0.0039377	0.0012658
15	49	7	1467.5	0.0044097	0.0047700
16	50	4	1516.0	0.0049422	0.0026385
17	51	7	1371.5	0.0054850	0.0051039
18	52	4	1343.0	0.0060382	0.0029784
19	53	4	1304.0	0.0066017	0.0030675
20	54	11	1232.5	0.0072663	0.0089249
21	55	11	1204.5	0.0080523	0.0091324
22	56	13	1113.5	0.0090710	0.0116749
23	57	12	1048.0	0.0101210	0.0114504
24	58	12	1155.0	0.0111823	0.0103896
25	59	19	1018.5	0.0122548	0.0186549
26	60	12	945.0	0.0133386	0.0126984
27	61	16	853.0	0.0145047	0.0187573
28	62	12	750.0	0.0158753	0.0160000
29	63	6	693.0	0.0174514	0.0086580
30	64	10	594.0	0.0192848	0.0168350

Table 8.2

Updating Times in Classic BUGS.

Iterations	Smooth 1	Smooth 2	Smooth 3	Smooth 4
1-10	794 secs	249 secs	291 secs	299 secs
11-20,000	186 secs	180 secs	296 secs	321 secs
20,001-25,000	49 sec	46 secs	53 secs	81 secs

Table 8.3

Updating Times in WinBUGS.

Iterations	Smooth 1	Smooth 2	Smooth 3	Smooth 4
1-10	25 secs	25 secs	3 secs	2 secs
11-20,000	166 secs	165 secs	270 secs	293 secs
20,001-25,000	43 secs	61 secs	65 secs	65 secs

Figure 8.1

Graphical Model for the Broff Example.

Figure 8.2

Original and Graduated Forces of Mortality.

9 Kimjo : Kimeldorf-Jones Styled Bayesian Graduations

9.1 Introduction

One of the earliest Bayesian graduation methods was proposed by Kimeldorf and Jones (1967). The Kimeldorf-Jones (K-J) method assumes that the observed mortality rates are conditionally independent and normally distributed about some unknown true rates given the same, which are themselves positively correlated and normally distributed about known prior or standard rates (a more detailed description of the method follows below). A natural question to ask, is how successfully BUGS can be used to implement this early, yet still popular, graduation method. Further, what value has BUGS to add to the traditional K-J graduation method ? We will consider these questions below.

9.2 The Traditional K-J Graduation Method

The traditional K-J graduation method we are about to review is summarized in London (1985, pages 75-82). Let the $n \times 1$ vectors $\mathbf{u}$, $\mathbf{t}$, and $\mathbf{m}$ represent the observed, true, and prior or standard mortality rates over n adjacent age intervals, respectively. The vector $\mathbf{m}$ is assumed to be given and is typically obtained from a prior mortality study or a table of standard rates. We seek to estimate the unknown and unobserved true rates comprising the vector $\mathbf{t}$.

Begin by postulating independent normal distributions for each observed rate were the true rates known. That is,

$$u_i \ \sim \ \text{normal}\ (t_i, B_i^{-1}) \ , \ i = 1, \ldots, n \ . \tag{9.1}$$

In this expression, the element B_i^{-1} denotes the precision, or inverse variance, of u_i. Traditionally, this value is estimated by treating u_i as an approximate binomial proportion with associated sample size (exposure) e_i and $Var(u_i) = t_i(1 - t_i)/e_i$. Since the value of t_i is unknown, we use $B_i = m_i(1 - m_i)/e_i$ instead.

The traditional K-J model specification continues by postulating the conditional distribution of $\mathbf{t}$ to be normal also - specifically, multivariate normal with mean vector $\mathbf{m}$ and positive definite and non-singular covariance matrix $\mathbf{A}$ (with elements A_{ij}). That is,

$$\mathbf{t} \ \sim \ \text{multivariate normal } (\mathbf{m}, \mathbf{A}^{-1}) \ , \tag{9.2}$$

with $\mathbf{A}^{-1}$ denoting the precision, or inverse covariance, matrix. Usually, the covariance matrix is designed so that a positive (or at least non-negative) correlation exists between any two variables t_i and t_j, with the correlation decreasing as the distance between the index values i and j increases. The correlation structure here is being used to model "relations between neighbouring rates" (Elphistone, 1951, page 18; London, 1985, page 79) and is the principal mechanism being used to define the smoothness inherent in past knowledge (Hickman and Miller, 1977, page 13). Kimeldorf and Jones (1967) described two particular covariance structures of interest in practice. Their matrix class a_1 is defined by

$$A_{ij} = p^2 r^{|i-j|} \ , \ p > 0 \ , \ 0 \leq r < 1 \ ,$$

while their matrix class a_2 supposes that

$$A_{ij} = p^2 r_i r_{i+1} \cdots r_{j-1} \ , \ p > 0 \ , \ 0 \leq r_k < 1 \ ,$$

for all k. Matrix class a_1 is clearly contained within class a_2.

The next step in the K-J graduation method involves the derivation of the posterior distribution of the vector $\mathbf{t}$ given $\mathbf{u}$. This is a straightforward derivation provided the actuarial practitioner is familiar with multivariate normal theory. The resulting posterior distribution is well known to be multivariate normal with mean $\mathbf{v}$ and precision matrix $\mathbf{C}^{-1}$, that is

$$f(\mathbf{t} \mid \mathbf{u}) \ \sim \ \text{multivariate normal } (\mathbf{v}, \mathbf{C}^{-1}) \ , \tag{9.3}$$

with

$$\mathbf{v} \ = \ (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1} (\mathbf{B}^{-1}\mathbf{u} + \mathbf{A}^{-1}\mathbf{m}) \tag{9.4}$$

and

$$\mathbf{C} = (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1} \ .$$

Here, we have used $\mathbf{B}$ to denote a matrix with the elements B_i on the main diagonal and with zeroes everywhere else. For the graduated vector, it is common to adopt the posterior mean $\mathbf{v}$. Recall, the mean of a normal distribution is also equal to its median and modal values.

At this point, it may be worthwhile to illustrate the K-J method with an application to an actual data set. The data we select is taken from Miller (1946, page 11) and appears in the first four columns of Table 9.1. This data also appears in London (1985, page 20). The standard rates appearing in column five are from the 1969-71 U.S. Life Table for White Males and are also used in London (1985, page 24). Our illustrative graduation (Smooth 1) will incorporate the matrix class a_1 with p^2 set equal to the average of the B_i values (i.e., $p^2 = 0.000588$) and $r = 0.8$. The graduated values defined by (9.4) appear in the second column of Table 9.2 and are plotted in Figure 9.1 as Smooth 1 (Exact).

9.3 Model Specification and BUGS Files for the Kimjo Example

The K-J model specification described above is relatively simple to code in BUGS, although there are two important points to keep in mind. The first is that the syntax of the 'classic' BUGS language differs slightly in some respects from that of the text based specification language associated with WinBUGS. To illustrate, consider expression (9.2). In terms of 'classic' BUGS code, we have :

```
t[] ~ dmnorm( m[], Ainv[,] );
Ainv[,] <- inverse( A[,] );
```

On the other hand, in WinBUGS this specification is coded as :

```
t[1:n] ~ dmnorm( m[], Ainv[,] );
for( i in 1:n ){
  for( j in 1:n ){
    Ainv[i,j] <- inverse( A[,], i, j );
  }
}
```

We observe that WinBUGS requires all multivariate nodes appearing on the left-hand-side of statements to have their indexes explicitly defined. Also, the `inverse()` function is defined slightly differently in the 'classic' BUGS and WinBUGS environments. The second point to keep in mind is that the `dmnorm()` function limits the vector **t** to 25 or fewer elements. See Section 9.10 of the BUGS 0.5 Manual (version ii) for a discussion of this point and others relating to the use of multivariate normal nodes. In Section 9.4, we will discuss how to circumvent this restriction in BUGS.

The complete files required to implement the illustrative K-J graduation using 'classic' BUGS are given below. They are presently maintained on the world wide web at www.math.ucalgary.ca/∼scollnik/abcd/. A counterpart file kimjo.odc for use with WinBUGS is available there also. In the file kimjo.bug, observe that the exposures `e[i]` are modelled probabilistically as gamma (a, b) variables (with vague priors on the hyperparameters a and b). This permits unobserved or missing values of `u[i]` and `e[i]` to simply be entered as `NA` in the appropriate spots in the data file kimjo.dat.

9.3.1 The File kimjo.bug

```
model kimjo;

const
  n = 15,       # number of age classes

var
  u[n],         # observed mortality rates
  e[n],         # observed exposures
  a, b,         # parameters for modelling exposures

  t[n],         # true rates
  m[n],         # prior rates

  A[n,n],
  Ainv[n,n],
  Binv[n],
  p2, r,

  fit[n], fitotal,
  smooth[n-3], smtotal;

data in "kimjo-s.dat";    # data in S-PLUS format
inits in "kimjo.in";

{

# Model structure for a traditional K-J graduation, provided n <= 25.

  for( i in 1:n ){
    u[i] ~ dnorm( t[i], Binv[i] );
  }

  for( i in 1:n ){
    Binv[i] <- e[i] / ( m[i] * ( 1 - m[i] ) );
    e[i] ~ dgamma( a, b );
  }

  a ~ dexp( 0.01 );
  b ~ dexp( 0.01 );
```

```
# Fit of t[] to u[].

  fitotal <- sum( fit[] );
  for( i in 1:n ){
    fit[i] <- Binv[i] * pow( t[i] - u[i] , 2);
  }

# Smoothness of t[].

  smtotal <- sum( smooth[] );
  for( i in 1:(n-3) ){
    smooth[i] <- pow( t[i+3] - 3 * t[i+2] + 3 * t[i+1] - t[i] , 2);
  }

  t[] ~ dmnorm( m[], Ainv[,] );

  Ainv[,] <- inverse( A[,] );
  for( i in 1:(n-1) ){
    A[i,i] <- p2;
    for( j in (i+1):n ){
      A[i,j] <- p2 * pow( r, j-i );
      A[j,i] <- A[i,j];
    }
  }
  A[n,n] <- p2;

  p2 <- 0.0005882139;
  r <- 0.8;

}
```

9.3.2 The Data File kimjo-s.dat

```
list( e = c( 135, 143, 140, 144, 149,
             154, 150, 139, 145, 140,
             137, 136, 126, 126, 109 ) ,

      u = c( 0.0444, 0.0839, 0.0714, 0.0764, 0.0403,
             0.1039, 0.1600, 0.0575, 0.1103, 0.0929,
             0.1387, 0.1544, 0.1825, 0.2063, 0.2385 ) ,

      m = c( 0.049, 0.053, 0.057, 0.061, 0.066,
             0.072, 0.078, 0.085, 0.091, 0.097,
             0.105, 0.113, 0.121, 0.130, 0.139 ) )
```

9.3.3 The Initial Value File kimjo.in

```
list( t = c( 0.01, 0.02, 0.03, 0.04, 0.05,
             0.06, 0.07, 0.08, 0.09, 0.10,
             0.11, 0.12, 0.13, 0.14, 0.15 ) )
```

9.3.4 Output and Analysis

It took BUGS about 34 seconds to run for 15,000 iterations on a 150 MHz Pentium PC. The graduated values were set equal to the estimated posterior means based upon the last 5000 iterations, and are plotted in Figure 9.1 as Smooth 1 (Estimated). These estimated means are indistinguishable from the exact values. Instead of using the means, we might have wanted more conservative mortality rates to use in our calculations and utilized, say, 75th percentiles. For illustration's sake, the empirical 75th percentiles from the simulation's last 5000 iterations are also plotted in Figure 9.1.

Any graduation is a compromise between the fit of the graduated vector to the initial rates and the smoothness inherent in the graduated vector. These aspects of fit and smoothness associated with a graduated vector $\mathbf{v}$ have traditionally been measured quantitatively using statistics like

$$\mathrm{F_V} \;=\; \sum_{i\,=\,1}^{n} \frac{(u_i \;-\; v_i)^2}{B_i} \tag{9.5}$$

and

$$\mathrm{S_V} \;=\; \sum_{i\,=\,1}^{n-3} (\Delta^3 v_i)^2 \;, \tag{9.6}$$

respectively. Of course, the symbol Δ in (9.6) denotes the forward difference operator, so that Δ^3 produces third forward differences (see Appendix D in London, 1985). Small values of (9.5) indicate a good fit, while small values of (9.6) indicate a good smooth. We calculated the value of these two statistics for Smooth 1 and report them in Table 9.3.

In the Bayesian context, the graduated values are representatives from the posterior distribution. Instead of calculating (9.5) and (9.6) soley for the graduated vector $\mathbf{v}$, we might monitor their posterior distribution of values when $\mathbf{v}$ is replaced with $\mathbf{t}$. That is, monitor the random variables

$$\mathrm{F_T} \;=\; \sum_{i\,=\,1}^{n} \frac{(u_i \;-\; t_i)^2}{B_i} \tag{9.7}$$

and

$$\mathrm{S_T} \;=\; \sum_{i\,=\,1}^{n-3} (\Delta^3 t_i)^2 \;. \tag{9.8}$$

This is easy to do in BUGS, and the relevant code already appears in the kimjo.bug file with the variables fitotal and smtotal corresponding to (9.7) and (9.8), respectively. The estimated posterior mean for each of these two random variables is given in Table 9.3. Observe that the graduated vector $\mathbf{v}$ defined by (9.4) is a better fit to the initial rates, and is also smoother, than the sequence of true rates $\mathbf{t}$ is on average.

The discussion above applies to any function of mortality rates, such as a future life expectancy or a present value of an annuity or insurance. If this function is viewed as a function of the graduated rates, then the function reduces to a fixed number. If the function is viewed as a function of the true rates, then the function is a random variable and draws from its posterior distribution can be generated and monitored within BUGS, or else produced afterwards by simply applying the function in question to each draw of $\mathbf{t}$ using a spreadsheet or statistical computing package.

9.4 An Order-Restricted K-J Styled Graduation Method

As noted above, the `dmnorm()` function limits the vector **t** to 25 or fewer elements. The reason we required the use of this function in the first place was to model the correlated multivariate normal distribution in expression (9.2). The smoothness inherent in past knowledge was modelled through a non-negative correlation structure between any two variables t_i and t_j, with the correlation decreasing as the distance between the index values i and j increases. A similar style of correlation structure can be induced by initially assuming that each t_i, $i = 1, \ldots, n$, is conditionally independent and then imposing order constraints so that $0 < t_1$, $t_n < 1$, and

$$t_{i-1}/(1 + f_{i-1}) < t_i < (1 + f_i)t_{i+1}, \tag{9.9}$$

for $i = 2, \ldots, n-1$, with $f_i \geq 0$, for $i = 1, \ldots, n-1$. This order structure will induce positive correlations between any two variables t_i and t_j. In practice, we might let $f_i = f \geq 0$, for all values of i, and try several values of f until a satisfactory graduation obtains. This order-restricted K-J style of graduation circumvents the vector size restriction associated with the `dmnorm()` function in BUGS which complicated our traditional K-J analysis, since (9.9) can be modelled in BUGS using interval restricted univariate normal `dnorm()` functions.

Only minor changes to the code in the file kimjo.bug are required to implement the order-restricted K-J model in BUGS. The revised model is coded in the file kimjor.bug. Note the use of the `lbound[]` and `ubound[]` arrays to specify the relevant bounds for each element of `t[]`. The files kimjo.dat and kimjo.in are unchanged from before.

9.4.1 The File kimjor.bug

```
model kimjor;

const
  n = 15;       # number of age classes

var
  u[n],         # observed mortality rates
  e[n],         # observed exposures
  a, b,         # parameters for modelling exposures

  t[n],         # true rates
  m[n],         # prior rates

  A, Ainv,
  Binv[n],
  p2,

  lbound[n], ubound[n], f,

  fit[n], fitotal,
  smooth[n-3], smtotal;

data in "kimjo-s.dat";    # data in S-PLUS format
inits in "kimjo.in";

{

# Model structure for order-restricted K-J graduation.

  for( i in 1:n ){
    u[i] ~ dnorm( t[i], Binv[i] );
  }

  for( i in 1:n ){
    Binv[i] <- e[i] / ( m[i] * ( 1 - m[i] ) );
    e[i] ~ dgamma( a, b );
  }

  a ~ dexp( 0.01 );
  b ~ dexp( 0.01 );
```

```
# Fit of t[] to u[].

  fitotal <- sum( fit[] );
  for( i in 1:n ){
    fit[i] <- Binv[i] * pow( t[i] - u[i] , 2);
  }

# Smoothness of t[].

  smtotal <- sum( smooth[] );
  for( i in 1:(n-3) ){
    smooth[i] <- pow( t[i+3] - 3 * t[i+2] + 3 * t[i+1] - t[i] , 2);
  }

  for( i in 1:n ){
    t[i] ~ dnorm( m[i], Ainv ) I(lbound[i],ubound[i]);
  }

  lbound[1] <- 0;
  ubound[1] <- t[2] * ( 1 + f );
  for( i in 2:(n-1) ){
    lbound[i] <- t[i-1] / ( 1 + f );
    ubound[i] <- t[i+1] * ( 1 + f );
  }
  lbound[n] <- t[n-1] / ( 1 + f );
  ubound[n] <- 1;

  f <- 0;
# f <- 0.25;
# f <- 1;

  Ainv <- 1 / A;
  A <- p2;

  p2 <- 0.0005882139;

}
```

9.4.2 Output and Analysis

It took BUGS about 13 seconds to run the kimjor model for 15,000 iterations on a 150 MHz Pentium PC. Three different settings of f were used for comparison : $f = 0, 0.25$, and 1. We refer to these three parameter settings as Smooths 2, 3, and 4, respectively. In each case, the graduated values were set equal to the estimated posterior means based upon the last 5000 iterations of the corresponding BUGS run. The resulting sequences of graduated values are plotted in Figure 9.2, and recorded in Table 9.2. The posterior standard deviations appearing in this table were also estimated using the final 5000 iterations of the corresponding BUGS run.

In Table 9.3, we have recorded the corresponding values of the fit and smooth statistics (9.5) and (9.6). For each graduation, we evaluated these statistics using the final graduated vector. We also monitored (9.7) and (9.8) under each of the three graduations, and have also recorded their estimated posterior means in Table 9.3. As we might expect in light of (9.9), we sacrifice smoothness in the graduated vector for a better fit with the initial rates as the value of f increases. Once again, observe that in all three cases the graduated vector defined by the mean of the posterior distribution is a better fit to the initial rates, and is also smoother, than the sequence of true rates $\mathbf{t}$ is on average.

9.5 Further Discussion of the Kimjo and Kimjor Examples

Although the traditional K-J graduation method can be implemented in BUGS, the present implementation of the multivariate normal distribution in BUGS effectively restricts us to data sets with 25 or fewer rates. An alternative is to utilize interval restricted univariate normal random variables for the elements of $\mathbf{t}$. This is easy to code in BUGS and can yield a reasonable spectrum of candidate graduated vectors.

Table 9.1

Raw Mortality Data and Standard Rates.

Age	Exposure	Deaths	Initial Rate	Standard Rate
70	6	135	0.044	0.049
71	12	143	0.084	0.053
72	10	140	0.071	0.057
73	11	144	0.076	0.061
74	6	149	0.040	0.066
75	16	154	0.104	0.072
76	24	150	0.160	0.078
77	8	139	0.058	0.085
78	16	145	0.110	0.091
79	13	140	0.093	0.097
80	19	137	0.139	0.105
81	21	136	0.154	0.113
82	23	126	0.183	0.121
83	26	126	0.206	0.130
84	26	109	0.239	0.139

Table 9.2

Graduated Rates : Posterior Expected Values (SDs).

Age	Smooth 1	Smooth 2	Smooth 3	Smooth 4
70	0.055 (0.011)	0.039 (0.011)	0.045 (0.014)	0.046 (0.015)
71	0.068 (0.010)	0.055 (0.008)	0.067 (0.012)	0.072 (0.014)
72	0.070 (0.010)	0.062 (0.008)	0.067 (0.010)	0.066 (0.014)
73	0.072 (0.010)	0.068 (0.008)	0.068 (0.010)	0.068 (0.014)
74	0.072 (0.010)	0.074 (0.008)	0.066 (0.010)	0.056 (0.013)
75	0.093 (0.010)	0.084 (0.008)	0.090 (0.015)	0.091 (0.015)
76	0.108 (0.010)	0.092 (0.008)	0.105 (0.012)	0.120 (0.015)
77	0.098 (0.010)	0.096 (0.008)	0.093 (0.011)	0.079 (0.013)
78	0.106 (0.010)	0.103 (0.008)	0.099 (0.013)	0.100 (0.017)
79	0.113 (0.010)	0.111 (0.008)	0.101 (0.014)	0.095 (0.017)
80	0.131 (0.010)	0.122 (0.009)	0.120 (0.016)	0.121 (0.018)
81	0.148 (0.010)	0.134 (0.010)	0.132 (0.016)	0.131 (0.017)
82	0.165 (0.011)	0.148 (0.011)	0.146 (0.017)	0.147 (0.019)
83	0.179 (0.011)	0.164 (0.012)	0.161 (0.017)	0.160 (0.019)
84	0.188 (0.013)	0.185 (0.015)	0.175 (0.019)	0.174 (0.020)

Table 9.3

Measures of Fit and Smooth.

Measure	Smooth 1	Smooth 2	Smooth 3	Smooth 4
F_V	16.502	26.540	20.678	14.923
$E[F_T]$	21.443	28.869	25.792	21.836
1000 S_V	4.578	0.223	6.475	44.838
1000 $E[S_T]$	20.738	5.680	37.893	100.370

Figure 9.1

Original and Graduated Mortality Rates.

Figure 9.2

Original and Graduated Mortality Rates.

10 Monty : The Monty Hall, or Let's Make a Deal, Problem

10.1 Introduction

This example is a consideration of simple conditional probabilities in a confusing context, with no explicit actuarial content. Coincidentally, though, this puzzle was posted (most assuredly not by this author) to discussion forums on both the Society of Actuaries and Casualty Actuarial Society websites, and also to the email discussion list of the Canadian Institute of Actuaries, during the spring / summer of 1999. Hundreds of responses were generated - many of which were incorrect. The problem is stated below, after which we consider its analysis.

THE MONTY HALL PROBLEM

> Pretend that you are a contestant on the classic television gameshow "Let's Make a Deal" with host Monty Hall. You are faced with 3 (or, more generally, N) doors, behind each of which is a prize. Behind one of the doors is a new car, while there is a goat behind each of the other 2 (or $N-1$). We proceed under the assumption that, as a contestant, you'd rather win the new car instead of a goat.
>
> You select a door, but do not yet know what is behind it. Monty, **knowing what is behind each door**, opens one of the other 2 (or $N-1$) remaining doors **specifically** in order to reveal one of the 2 (or $N-1$) goats. The rules of the game specify that Monty **must** open a door (i.e., he cannot stand pat and do nothing).
>
> You are now given the chance to change your selection to the other (or any one of the other $N-2$) unopened door(s). In the end, you will win whatever is behind the door you choose.
>
> Should you switch ? Why or why not ?

You may wish to take a few minutes and try to determine answers to these questions for yourself in the simplest case when $N = 3$, before reading their correct solution below.

The game we've described is usually played with 3 doors, but can be played with any number of doors N as noted above, provided $N \geq 3$. When $N = 3$, the probability of winning the car is $\frac{1}{3}$ if you do not switch doors, and increases to $\frac{2}{3}$ if you do. The paradox, if it may be called that, is that many people's intuitions are easily confused by conditional probabilities and this leads them to mistakenly calculate the revised probability of winning the car as $\frac{1}{2}$ under the switching strategy. (In general, when $N \geq 3$ it may be shown that the probability of winning the car changes from $\frac{1}{N}$ to $\frac{1}{N} * \frac{N-1}{N-2}$ if one adopts the switching strategy. As this second expression is greater than $\frac{1}{N}$ when $N \geq 3$, it is always better to switch doors.) It is important to note that these results implicitly assume that Monty will randomly select a door hiding a goat, when he has more than one such door to choose from. If Monty uses a different strategy (e.g., always selects the lowest numbered door from among the alternatives) that is advertised to the contestant, or has the option of not opening a door at all, then the contestent may or may not be advised to switch (depending on the strategy adopted by Monty).

The result we want is easy to prove in the 3 door case by applying Bayes' Theorem. Without any loss in generality, suppose that you chose door 1, Monty opens door 2, and you now must decide whether to switch your choice from door 1 to door 3. Let:

A be the event that the car is behind door 1 ;

B be the event that the car is behind door 3 ;

C be the event that Monty opened door 2.

It should be apparent to the reader that the marginal probabilities $P(A)$ and $P(B)$ are both equal to $\frac{1}{3}$. However, we are interested in comparing the conditional probabilities $P(A|\, C)$ and $P(B|\, C)$. From the given information, it is fairly easy to see that $P(C|\, A) = \frac{1}{2}$ (as we are assuming that Monty chooses a door at random if more than one door hides a goat) and $P(C|\, B) = 1$ (as Monty as no choice in this case). Bayes' Theorem tells us that:

$$P(A|\, C) \;=\; \frac{P(A)\; P(C|\, A)}{P(A)\; P(C|A) \;+\; P(B)\; P(C|\, B)}$$

and

$$P(B|\,C) \;=\; \frac{P(B)\;P(C|\,B)}{P(A)\;P(C|A)\;+\;P(B)\;P(C|\,B)}\;.$$

Substituting the probabilities we've already identified into the right hand sides of these expressions gives us the desired result, namely, $P(A|\,C) = \frac{1}{3}$ and $P(B|\,C) = \frac{2}{3}$.

Many discussions of the Monty Hall problem are available. References to several can be found in an article published in the Paradox section of the February 1998 issue of *Liaison* (Volume 12, Number 1, February 1998), the newsletter of the Statistical Society of Canada. Many other discussions can be found on the World Wide Web, starting with these websites:

astro.uchicago.edu/rranch/vkashyap/Misc/mh.html

www.io.com/∼kmellis/monty.html

math.rice.edu/∼ddonovan/montyurl.html

In the next section, we will discuss how to simulate trials of the Monty Hall game for the general case ($N \geq 3$) using WinBUGS, and then perform a simulation-based check of the result presented earlier concerning the probabilities of winning the car, with and without the switching of doors.

10.2 Specification of the Monty Hall Problem in WinBUGS

The code required to implement the Monty Hall problem in WinBUGS is given below and is also available on this author's website at www.math.ucalgary.ca/∼scollnik/abcd/ in the file monty.odc. This code merely simulates one play of the Monty Hall game from the game's starting point. By simulating thousands of plays of the game, we can estimate the probabilities of winning the car with or without the use of the switching strategy.

In the code below, note that the variable `door[i]` takes on the value 1 or 0 in order to indicate whether the i-th door hides the car or a goat, respectively. A categorical distribution described with the statement `dcat( p[] )` is used to make the actual determination as to which of the doors hides the car. We assume uniform probabilities across the available

alternatives. Another categorical distribution is used to decide the contestant's first pick, again with uniform probabilities. The prize (1 or 0) associated with the contestant's selection is assigned to the node `firstpick`. Although the contestant would not know whether or not `firstpick` is equal to the car or a goat in real life, WinBUGS does in the context of the simulation. This knowledge is used to determine how many goats are left behind the remaining doors and hence the probability `carprob` of ending up with the car if a switch is made. The prize associated with the switch pick is determined using a Bernoulli random draw with probability `carprob` and is assigned to the node `switchpick`.

Finally, by monitoring the values of `firstpick` and `switchpick` (or `carprob`) we can estimate the conditional probabilities of winning the car, with and without the use of the switching strategy. The theoretical exact and simulation-based estimated probabilities of winning are given in Table 10.1, for $N = 3, \ldots, 20$. The estimated probabilities in each row are based on simulations that ran for 25,000 iterations in WinBUGS. These simulations took several seconds to run, apiece. Clearly, the simulation results are well in accordance with the theoretical.

CODE FOR THE MONTY HALL PROBLEM

```
model;
{

   # How many doors is the game played with ?
   # Define N using the data statement below.

   # Assign the goats and the car to different doors. Without any loss
   # in generality,  we could simply place the car behind door number 1
   # all of the time ...
   # j <- 1

   # but to convince skeptics, we randomly select the door behind which
   # the car is placed.
   for( i in 1:N ) { p[i] <- 1 / N }
   j ~ dcat( p[] )
```

```
   # Now, place the prizes.
   for( i in 1:N ) { door[i] <- equals( i, j ) }

   # Select a door using equal probabilities.
   k ~ dcat( p[] )

   # Firstpick is the prize you win if you do not switch doors and stick
   # with door number k.
   firstpick <- door[k]

   # Determine switchpick, the prize you would win by switching.

   # Remember,  there were N-1 goats to begin with.  As Monty revealed
   # one of them to you, there are either N-2 or N-3 behind the remaining
   # doors ( depending on whether your firstpick happens to be equal to 1
   # or 0. Of course, you do not know which is the actual case ). So, the
   # number of goats remaining is given by :
   goatsleft <- ( N - 2 ) - ( 1 - firstpick )

   # If you were to switch after Monty reveals one of the goats to you,
   # then the probability you select a door with one of the remaining
   # goats still behind it is given by :
   goatprob <- goatsleft / ( N - 2 )

   # Hence :
   carprob <- 1 - goatprob
   switchpick ~ dbern( carprob )

}

DATA

Classic context with 3 doors ( the value of N can be edited ) ...

list( N = 3 )


INITS

You do not need to load inits ... just use the ‘gen inits’ button.
```

Table 10.1

Exact and Estimated Probabilities of Winning.

Estimated Probabilities are Based on 25,000 Iterations in WinBUGS.

	Pr(Win | No Switch)		Pr(Win | Switch)	
N	Exact	Estimated	Exact	Estimated
3	0.33333333	0.3322	0.66666667	0.6678
4	0.25000000	0.2448	0.37500000	0.3746
5	0.20000000	0.1992	0.26666667	0.2668
6	0.16666667	0.165	0.20833333	0.2083
7	0.14285714	0.1429	0.17142857	0.1726
8	0.12500000	0.1222	0.14583333	0.1468
9	0.11111111	0.1091	0.12698413	0.1276
10	0.10000000	0.1006	0.11250000	0.1116
11	0.09090909	0.09012	0.10101010	0.1011
12	0.08333333	0.08084	0.09166667	0.09232
13	0.07692308	0.07584	0.08391608	0.084
14	0.07142857	0.07044	0.07738095	0.07708
15	0.06666667	0.06612	0.07179487	0.07132
16	0.06250000	0.06224	0.06696429	0.0672
17	0.05882353	0.05792	0.06274510	0.06308
18	0.05555556	0.05508	0.05902778	0.05828
19	0.05263158	0.05232	0.05572755	0.05444
20	0.05000000	0.05052	0.05277778	0.05232

Complete References (Paper and Examples)

Best, N.G., Spiegelhalter, D.J., Thomas, A., and Brayne, C.E.G. (1996). Bayesian Analysis of Realistically Complex Models. *Journal of the Royal Statistical Society, Series A.* **159**, Part 2, pages 323-342.

Box, G.E.P., and Tiao, G.C. (1973). *Bayesian Inference in Statistical Analysis.* Addison-Wesley.

Broffit, J.D. (1984). A Bayes Estimator for Ordered Parameters and Isotonic Bayesian Graduation. *Scandinavian Actuarial Journal*, pages 231-247.

Broffit, J.D. (1986). Isotonic Bayesian Graduation with an Additive Prior. In *Advances in the Statistical Sciences: Vol. 6, Actuarial Science*, pages 19-40. Edited by MacNeill, I.B., and Umphrey, G.J. D. Reidal Publishing Co., Boston.

Broffit, J.D. (1988). Increasing and Increasing Convex Bayesian Graduation. *Transactions of the Society of Actuaries* **XL** Part 1, pages 115-148.

Brooks, S.P., and Gelman, A. (1998). General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics* **7** (4), pages 434-455.

Carlin, B.P. (1992a). Analyzing Nonlinear and Non-Gaussian Actuarial Time Series. *Actuarial Research Clearing House* **1992.1**, pages 27-60.

Carlin, B.P. (1992b). State Space Modeling of Non-Standard Actuarial Time Series. *Insurance: Mathematics and Economics* **11**, pages 209-222.

Carlin, B.P. (1992c). A Simple Monte Carlo Approach to Bayesian Graduation. *Transactions of the Society of Actuaries* **XLIV**, pages 55-76.

Carlin, B.P., and Louis, T.A. (1996). *Bayes and Empirical Bayes Methods for Data Analysis.* Chapman and Hall, London.

Consul, P.C. (1989). *Generalized Poisson Distributions : Properties and Applications.* Marcel Dekker Inc., New York.

Cowles, M.K., and Carlin, B.P. (1996). Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association* **91**, pages 883-904.

Dannenburg, D.R., Kaas, R., and Goovaerts, M.J. (1996). *Practical Actuarial Credibility Models.* Ceuterick, Belgium.

Dempster, A.P. (1997). The Direct Use of Likelihood for Significance Testing. *Statistics and Computing* **7**, pages 247-252.

Dickson, D., Tedesco, L.M., and Zehnwith, B. (1998). Predictive Aggregate Claims Distributions. *Journal of Risk and Insurance* **65** (4), pages 689-709.

Duvall, R.M. (1999). A Bayesian Approach to Negative Binomial Parameter Estimation. *Casualty Actuarial Society Forum* Winter 1999 Edition, pages 377-385.

Elphistone, M.D.W. (1951). Summation and Some Other Methods of Graduation: The Foundations of Theory. *TFA* **XX**, pages 15-??.

Gelfand, A.E., and Smith, A.F.M. (1990). Sampling-based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association* **85**, pages 398-409.

Gelfand, A.E., Hills, S.E., Racine-Poon, A., and Smith, A.F.M. (1990). Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling. *Journal of the American Statistical Association* **85**, pages 972-985.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (1995). *Bayesian Data Analysis*. Chapman and Hall, New York.

Gelman, A., and Rubin, D. (1992). Inference from Iterative Simulation using Multiple Sequences. *Statistical Science* **7**, pages 457-511.

Geman, S., and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEE Transactions on Pattern Analysis and Machine Intelligence* **6**, pages 721-741.

Gilks, W.R., Wild, P. (1992). Adaptive Rejection Sampling for Gibbs Sampling. *Applied Statistics* **41** (2), pages 337-348.

Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. (1996). Introducing Markov Chain Monte Carlo. In *Markov Chain Monte Carlo in Practice*. Edited by Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. Chapman and Hall, New York.

Haastrup, S., and Arjas, E. (1996). Claims Reserving in Continuous Time: a Nonparametric Bayesian Approach. *ASTIN Bulletin* **26** (2), pages 139-164.

Haastrup, S. (1997). Comparison of Some Bayesian Analyses of Heterogeneity in Group Life Insurance. *Working Paper* **150**, Laboratory of Actuarial Mathematics, University of Copenhagen.

Hastings, W.K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika* **57**, pages 97-109.

Herzog, T.N. (1996). *Introduction to Credibility Theory*. ACTEX Publications, Winsted, Connecticut.

Hesselager, O. (1993). A Class of Conjugate Priors with Applications to Excess-of-Loss Reinsurance. *ASTIN Bulletin* **23** (1), pages 77-93.

Hickman, J.C., and Miller, R.B. (1977). Notes on Bayesian Graduation. *Transactions of the Society of Actuaries* **XXIX**, pages 7-21.

Hogg, R.V., and Klugman, S.A. (1984). *Loss Distributions*. John Wiley & Sons, Inc., New York.

Hutchings, P.L., and Ullman, R.E. (1983). Prepaid Hospital Care Age / Sex and Hospital Continuation Study. *Transactions of the Society of Actuaries* **XXXV**, pages 623-655.

Jacquier, E., Polson, N.G., and Rossi, P.E. (1994). Bayesian Analysis of Stochastic Volatility Models. *Journal of Business & Economic Statistics* **12**, pages 371-389.

Johnson, N.L., Kotz, S., and Balakrishnan, N. (1997). *Discrete Multivariate Distributions*. John Wiley & Sons, Inc., New York.

Kimeldorf, G.S., and Jones, D.A. (1967). Bayesian Graduation. *Transactions of the Society of Actuaries* **XIX**, pages 66-??.

Klugman, S.A. (1992). *Bayesian Statistics in Actuarial Science, with Emphasis on Credibility*. Kluwer Academic.

Klugman, S.A., and Carlin, B.P. (1993). Hierarchical Bayesian Whittaker Graduation. *Scandinavian Actuarial Journal*, pages 183-196.

Klugman, S.A., Panjer, H.H., and Willmot, G.E. (1998). *Loss Models: From Data to Decisions*. John Wiley & Sons, Inc., New York.

Krzanowski, W.J., and Marriott, F.H.C. (1994). *Multivariate Analysis Part 1 Distributions, Ordination, and Inference*. Edward Arnold, New York.

London, D.L. (1985). *Graduation: The Revision of Estimates*. ACTEX Publications, Winsted, Connecticut.

Makov, U.E, Smith, A.F.M., and Liu, Y.-H. (1996). Bayesian Methods in Actuarial Science. *The Statistician* **45** (4), pages 503-515.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953). Equations of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **21**, pages 1087-1092.

Meyers, G. (1994). Quantifying the Uncertainty in Claim Severity Estimates for an Excess Layer When Using the Single Parameter Pareto. *Proceedings of the Casualty Actuarial Society* **LXXXI**, pages 91-122 (including discussion).

Miller, M.D. (1946). *Elements of Graduation*. Actuarial Society of America and American Institute of Actuaries, New York.

Morgan, I.M., and Hickman, J.C. (1993). Conjugate Bayesian Analysis of the Negative Binomial Distribution. *Actuarial Research Clearing House* **1993.1**, pages 97-113.

Norberg, R. (1989). Experience Rating in Group Life Insurance. *Scandinavian Actuarial Journal*, pages 194-224.

Pai, J. (1997). Bayesian Analysis of Compound Loss Distributions. *Journal of Econometrics* **79**, pages 129-146.

Panjer, H.H., and Willmot, G.E. (1992). *Insurance Risk Models*. Society of Actuaries, Schaumburg.

Rosenberg, M. (1994). *A Hierarchical Bayesian Model of the Rate of Non-Acceptable In-Patient Hospital Utilization*. Doctoral Thesis, University of Michigan.

Rosenberg, M., and Young, V.R. (1997). A Flexible Model for Time-Dependent Data. Submitted for publication, and presented at the 32nd Actuarial Research Conference.

Rosenberg, M., and Young, V.R. (1999). A Bayesian Approach to Understanding Time Series Data. *North American Actuarial Journal* **3** (2), pages 130-143.

Rytgaard, M. (1990). Estimation in the Pareto Distribution. *ASTIN Bulletin* **20** (2), pages 201-216.

Schnieper, René. (1995). On the Estimation of the Credibility Factor: A Bayesian Approach. *ASTIN Bulletin* **25** (2), pages 137-151.

Scollnik, D.P.M. (1993). A Bayesian Analysis of a Simultaneous Equations Model for Insurance Rate-Making. *Insurance: Mathematics and Economics* **12**, pages 265-286.

Scollnik, D.P.M. (1995a). Simulating Random Variates from Makeham's Distribution and from Others with Exact or Nearly Log-Concave Densities. *Transactions of the Society of Actuaries*. **XLVII**, pages 409-437.

Scollnik , D.P.M. (1995b). Bayesian Analysis of Two Overdispersed Poisson Models. *Biometrics* **51**, pages 1117-1126.

Scollnik, D.P.M. (1995c). The Bayesian Analysis of Generalized Poisson Models for Claim Frequency Data Utilising Markov Chain Monte Carlo Methods. *Actuarial Research Clearing House* **1995.1**, pages 339-356.

Scollnik, D.P.M. (1996). An Introduction to Markov Chain Monte Carlo Methods and Their Actuarial Applications. *Proceedings of the Casualty Actuarial Society* **LXXXIII**, pages 114-165.

Scollnik, D.P.M. (1998). On the Analysis of the Truncated Generalized Poisson Distribution Using a Bayesian Method. *ASTIN Bulletin* **28** (1), pages 135-152.

Scollnik, D.P.M. (2000). Actuarial Modeling with MCMC and BUGS. *North American Actuarial Journal*. To appear.

Shaban, S.A. (1988). Poisson-Lognormal Distributions. In *Lognormal Distributions: Theory and Applications*. Edited by Crow, E.L., and Shimizu, K. Marcel Dekker Inc, New York and Basel.

Shephard, N., and Pitt, M.K. (1995). Parameter-Driven Exponential Family Models. *Technical Report, Nuffield College, Oxford*.

Smith, A.F.M., and Roberts, G.O. (1993). Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods (with discussion). *Journal of the Royal Statistical Society, Series B*. **55**, pages 3-23.

Smith, T.C., Spiegelhalter, D.J., and Parmar, M.K.B. (1996). Bayesian Meta-Analysis of Randomized Trials Using Graphical Models and BUGS. In *Bayesian Biostatistics*, pages 411-427. Edited by Berry, D.B., and Stangl, D.K. Marcel Dekker, Inc., New York.

Spiegelhalter, D.J., Best, N.G., and Carlin, B.P. (1998). Bayesian Deviance, the Effective Number of Parameters, and the Comparison of Arbitrarily Complex Models. Research Report 98-009, Division of Biostatistics, University of Minnesota. Preprint available at muskie.biostat.umn.edu/∼brad/. Submitted to *Journal of the Royal Statistical Society, Series B*.

Spiegelhalter, D.J., Thomas, A., and Best, N.G. (1996). Computation on Bayesian Graphical Models. In *Bayesian Statistics 5*, pages 407-425. Edited by Bernardo, J.M., Berger, J.O., Dawid, A.P., and Smith, A.F.M. Oxford University Press, Oxford.

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

Spiegelhalter, D.J., Thomas, A., Best, N.G., and Gilks, W.R. (1997). *BUGS 0.6: Bayesian inference Using Gibbs Sampling (Addendum to Manual)*. MRC Biostatistics Unit, Cambridge.

Tierney, L. (1994). Markov Chains for Exploring Posterior Distributions (with discussion). *Annals of Statistics* **22**, pages 1701-1762.

Venables, W.N., and Ripley, B.D. (1999). Modern Applied Statistics with S-PLUS, Third Edition. Springer, New York.

Wild, P., and Gilks, W.R. (1993). Adaptive Rejection Sampling from Log-concave Density Functions. *Applied Statistics* **42** (4), pages 701-709.

Bivreg Example References

Brooks, S.P., and Gelman, A. (1998). General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics* **7** (4), pages 434-455.

Dickson, D., Tedesco, L.M., and Zehnwith, B. (1998). Predictive Aggregate Claims Distributions. *Journal of Risk and Insurance* **65** (4), pages 689-709.

Gelman, A., and Rubin, D. (1992). Inference from Iterative Simulation using Multiple Sequences. *Statistical Science* **7**, pages 457-511.

Klugman, S.A., Panjer, H.H., and Willmot, G.E. (1998). *Loss Models: From Data to Decisions*. John Wiley & Sons, Inc., New York.

Scollnik, D.P.M. (2000). Actuarial Modeling with MCMC and BUGS. *North American Actuarial Journal*. To appear.

Broff / Health / Kimjo Example References

Broffit, J.D. (1984). A Bayes Estimator for Ordered Parameters and Isotonic Bayesian Graduation. *Scandinavian Actuarial Journal*, pages 231-247.

Broffit, J.D. (1986). Isotonic Bayesian Graduation with an Additive Prior. In *Advances in the Statistical Sciences: Vol. 6, Actuarial Science*, pages 19-40. Edited by MacNeill, I.B., and Umphrey, G.J. D. Reidal Publishing Co., Boston.

Broffit, J.D. (1988). Increasing and Increasing Convex Bayesian Graduation. *Transactions of the Society of Actuaries* **XL** Part 1, pages 115-148.

Carlin, B.P. (1992a). Analyzing Nonlinear and Non-Gaussian Actuarial Time Series. *Actuarial Research Clearing House* **1992.1**, pages 27-60.

Carlin, B.P. (1992b). State Space Modeling of Non-Standard Actuarial Time Series. *Insurance: Mathematics and Economics* **11**, pages 209-222.

Carlin, B.P. (1992c). A Simple Monte Carlo Approach to Bayesian Graduation. *Transactions of the Society of Actuaries* **XLIV**, pages 55-76.

Elphistone, M.D.W. (1951). Summation and Some Other Methods of Graduation: The Foundations of Theory. *TFA* **XX**, pages 15-??.

Hickman, J.C., and Miller, R.B. (1977). Notes on Bayesian Graduation. *Transactions of the Society of Actuaries* **XXIX**, pages 7-21.

Hutchings, P.L., and Ullman, R.E. (1983). Prepaid Hospital Care Age / Sex and Hospital Continuation Study. *Transactions of the Society of Actuaries* **XXXV**, pages 623-655.

Kimeldorf, G.S., and Jones, D.A. (1967). Bayesian Graduation. *Transactions of the Society of Actuaries* **XIX**, pages 66-??.

London, D.L. (1985). *Graduation: The Revision of Estimates.* ACTEX Publications, Winsted, Connecticut.

Miller, M.D. (1946). *Elements of Graduation.* Actuarial Society of America and American Institute of Actuaries, New York.

Credit Example References

Box, G.E.P., and Tiao, G.C. (1973). *Bayesian Inference in Statistical Analysis*. Addison-Wesley.

Dannenburg, D.R., Kaas, R., and Goovaerts, M.J. (1996). *Practical Actuarial Credibility Models*. Ceuterick, Belgium.

Schnieper, René. (1995). On the Estimation of the Credibility Factor: A Bayesian Approach. *ASTIN Bulletin* **25** (2), pages 137-151.

Exact Example References

Dempster, A.P. (1997). The Direct Use of Likelihood for Significance Testing. *Statistics and Computing* **7**, pages 247-252.

Dickson, D., Tedesco, L.M., and Zehnwith, B. (1998). Predictive Aggregate Claims Distributions. *Journal of Risk and Insurance* **65** (4), pages 689-709.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (1995). *Bayesian Data Analysis*. Chapman and Hall, New York.

Hogg, R.V., and Klugman, S.A. (1984). *Loss Distributions*. John Wiley & Sons, Inc., New York.

Klugman, S.A., Panjer, H.H., and Willmot, G.E. (1998). *Loss Models: From Data to Decisions*. John Wiley & Sons, Inc., New York.

Scollnik, D.P.M. (2000). Actuarial Modeling with MCMC and BUGS. *North American Actuarial Journal*. To appear.

Spiegelhalter, D.J., Best, N.G., and Carlin, B.P. (1998). Bayesian Deviance, the Effective Number of Parameters, and the Comparison of Arbitrarily Complex Models. Research Report 98-009, Division of Biostatistics, University of Minnesota. Preprint available at muskie.biostat.umn.edu/∼brad/. Submitted to *Journal of the Royal Statistical Society, Series B*.

Motor Example References

Dickson, D., Tedesco, L.M., and Zehnwith, B. (1998). Predictive Aggregate Claims Distributions. *Journal of Risk and Insurance* **65** (4), pages 689-709.

Hesselager, O. (1993). A Class of Conjugate Priors with Applications to Excess-of-Loss Reinsurance. *ASTIN Bulletin* **23** (1), pages 77-93.

Klugman, S.A., Panjer, H.H., and Willmot, G.E. (1998). *Loss Models: From Data to Decisions*. John Wiley & Sons, Inc., New York.

Pai, J. (1997). Bayesian Analysis of Compound Loss Distributions. *Journal of Econometrics* **79**, pages 129-146.

Rytgaard, M. (1990). Estimation in the Pareto Distribution. *ASTIN Bulletin* **20** (2), pages 201-216.

Norweg Example References

Haastrup, S. (1997). Comparison of Some Bayesian Analyses of Heterogeneity in Group Life Insurance. *Working Paper* **150**, Laboratory of Actuarial Mathematics, University of Copenhagen.

Norberg, R. (1989). Experience Rating in Group Life Insurance. *Scandinavian Actuarial Journal*, pages 194-224.