A Set of Bayesian Models for Analysis of Claim Lags

Dale Lamps

Abstract

This paper presents Bayesian models for analysis of claim lag data. The analyses of claim runoff patterns investigate the extraction of information on trend and seasonal patterns in incurred claims. This is investigated on the basis of individual group data, and also relative to a larger aggregation of pooled experience. The models are implemented based on Markov chain Monte Carlo methods using BUGS software. The models are applied to examples using simulated claim lag data.

Introduction

Claim lag analysis.

Effective analysis of claim lags plays a crucial role in setting reserves, analysis of profitability and experience rating. In this paper we explore a number of probabilistic models for analyzing claim lag data with the aim of projecting future runoff, with specific recognition of exposure changes, including extraction of information on trend and seasonal variations in incurred claims.

Description of claim lag table.

The claim lag data consists of monthly payments P_{ij} incurred in month i and paid in lag month j. For example, Case 1 has 48 months of incurred claims and 24 lag months. Thus $i = 1, 2, \ldots, 48$ and $j = 1, 2, \ldots, 24$. For $(i + j) > 49$, P_{ij} represents future unknown payments.

The claim lag data were generated using a spreadsheet as described in the appendix.

Sources of variation in incurred claims and in patterns of claim payments.

The following sources of variations in incurred claims and claim payments are considered.

- Random effects.
- Exposure. This would conceptually include the count of risk units (e.g. number of persons in a group health program) and an index of variation among individuals (e.g. for group health coverage: age, sex, geographic location).
- Trend. This includes all long term cost change effects (e.g. unit cost, utilization).
- Seasonal claims incurred.

In addition, the experience of a separate but similar experience pool is considered.

The analyses were conducted using the following three cases.

- Case 1: Random variations only. The lag table example for Case 1 was generated based on zero trend, constant exposure and no seasonal incurred claim variations.
- Case 2: Trend, exposure and seasonal variations are incorporated in the lag table.
- Case 3: The experience of a similar larger pool of experience is incorporated into the analysis.

WinBUGS Software as a basis for the analyses.
The analyses described below are based on Markov chain Monte Carlo (MCMC) techniques using WinBUGS Version 1.3 software. This software and its use for actuarial modeling is addressed in the excellent work by Scollnik, especially in Scollnik (2000).

<u>Case 1:</u> This case addresses random effects only. The sample lag table for Case 1 was generated based on zero trend, constant exposure and no seasonal incurred claim variations. Case 1 has 48 months of incurred claims and 24 lag months. Thus i = 1, 2, …, 48 and j = 1, 2, ..., 24. For $(i + j) > 49$, P_{ij} represents future unknown payments. The data are presented in the appendix.

A. Log-normal models.

P_{ij} are assumed to be log-normal distributed as follows:

$$\ln(P_{ij}) \sim N(\mu_{ij}, 1/\tau_{ij}),$$

where τ is "precision" which is defined in terms of variance as $\tau = 1/\sigma^2$. This notation is used in the interest of consistency with the coding convention of the WinBUGS programming language used for these analyses.

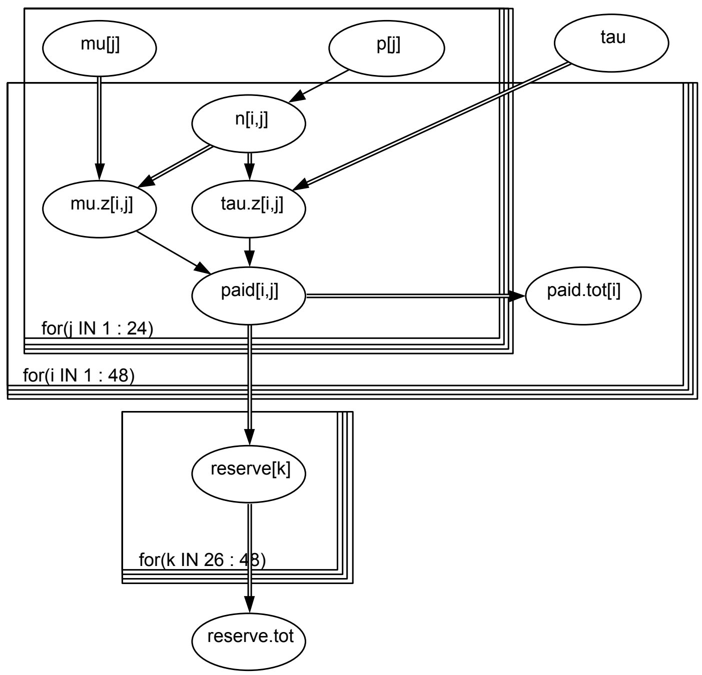
Many cells at longer lags may have values of $P_{ij} = 0$. This is accommodated by the following stratagem.

1. Replace all $P_{ij} = 0$ cells with $P_{ij} = 1$, to assure that $\ln(P_{ij})$ is defined.
2. For all non-zero cells, $\ln(P_{ij}) \sim N(\mu_j, 1/\tau_j)$.
3. Let p_{ij} be the probability that $P_{ij} = 1$.
4. An informative prior is used for p_{ij} based on an examination of several sets of simulated claims data based on the spreadsheet simulator. This particular prior may or may not be appropriate for a "real-life" set of claim lag data. The prior is based on the finding that there is an approximately linear relationship between p_{ij} and μ_p as follows:
   $$p_{ij} = 2.22 - .3\,\mu_p .$$
   I used the following prior:
   $$p_{ij} = 2.22 - .3\mu_{ij} + z_j$$
   where z_j is N(0, 1/100).
5. If $P_{ij} = 1$, then we force $\mu_{ij} = -.005$ and $\tau_{ij} = 100$. Thus $E(P_{ij}) = 1$ and $Var(P_{ij})$ = .01, which is believed to be a sufficiently small variance for this purpose.

Note that if $P_{ij} \neq 1$, then μ_{ij} and τ_{ij} should, in principle, be adjusted so that both E($P_{ij}|P_{ij} \neq 1$) and Var($P_{ij}|P_{ij} \neq 1$) are increased by a factor of $\frac{1}{1-p_{ij}}$. The case 1 log-normal models do not incorporate this adjustment.

Log-normal Model a. For the first model, let $\tau_j = \tau$ be independent of j for all non-zero cells.

The preliminary WinBUGS graph is as follows:



Using vague prior distributions for μ and τ, the WinBUGS coding is as follows:

```
#  Log-Normal model a, Case 1
#  IBNP24-1  ...  6/29/2
#  10,000 iterations - 527 sec. (single chain)

model;
{
  tau ~ dgamma(0.001,0.001)
  for( j in 1 : 24 ) {
    mu[j] ~ dnorm(0,1.0E-6)
    for( i in 1 : 48 ) {
      paid[i , j] ~ dlnorm(mu.z[i,j],tau.z[i,j])
      mu.z[i,j] <-  mu[j]*(1-n[i,j])-.005*n[i,j]      #  mean = 1 if no claims; no recognition of effect of p[j]
      tau.z[i,j] <-  tau*(1-n[i,j])+100*n[i,j]      #  variance = .01 if no claims; no recognition of effect of p[j]
    }
  }
```

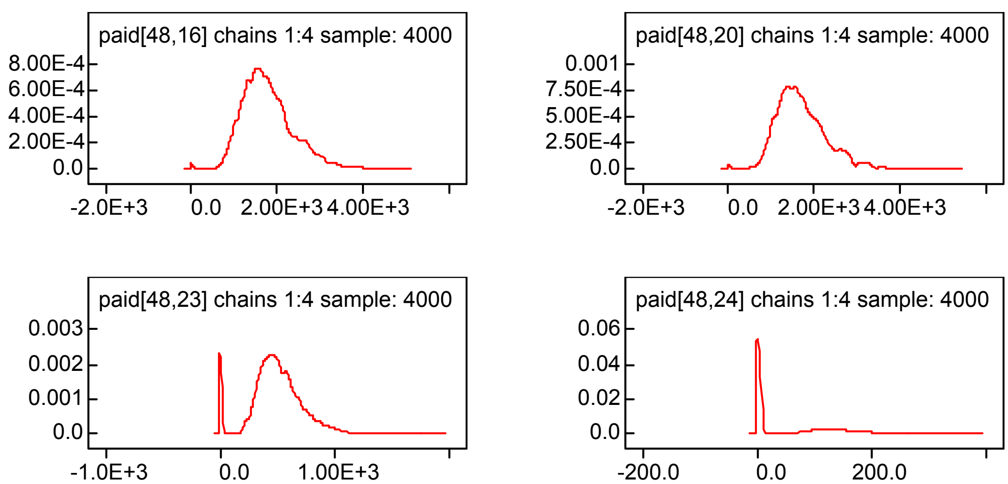
```
# Determine zero cells
  for( j in 1 : 24 ) {
    z[j] ~ dnorm( 0.0,100)
    p[j] <- max(0,min(2.22 -  0.3 * mu[j] + z[j],0.99))
    for( i in 1 : 48 ) {
      n[i , j] ~ dbern(p[j])           #  n = 1 for zero cell
    }
  }

# Summary results
  for( i in 1 : 48 ) {
    paid.tot[i] <- sum(paid[i , ])
  }
  for( i in 26 : 48 ) {
    reserve[i] <- sum(paid[i , 50 - i:24])
  }
  reserve.tot <- sum(reserve[26:48])
}
```

To confirm convergence I ran four chains with 8000 iterations. The chains were based on the following initial values for μ_j.

```
mu[] 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
mu[] 9 8 8 7 7 6 6 5 5 4 4 4 4 3 3 3 3 3 3 2 2 2 2 2
mu[] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
mu[] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

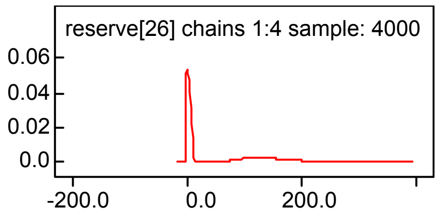
Gelman-Rubin convergence statistic values are presented in the appendix. Also in the appendix are selected graphs for iterations 0 to 600 to illustrate the convergence of the model.

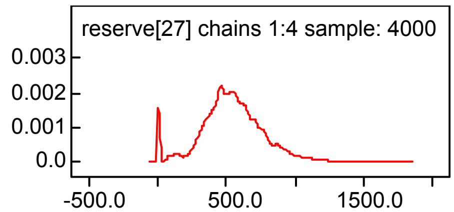
The kernel density graphs of paid[i,j] are of interest, showing the bi-modal features generated by the zero claim cells.



This illustrates one of the interesting things MCMC analysis does: it provides stochastic results for every unknown paid / incurred claims cell. These results can be seen more fully in the "node statistics" tables in the appendix.

I would offer a brief comment on two of the reserve graphs. The kernel for reserve[26] is identical to paid[26,24], and reserve[27] represents the combination of paid[27,23] and paid[27,24]. The reserve[27] graph shows a tri-modal pattern: observe the small bump on the lower left slope of the main arc. This results from being the combination of two bi-modal distributions.





Statistics for total reserve are:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	3.004E+6	277100.0	2872.0	2.527E+6	2.981E+6	3.171E+6	3.362E+6	3.5E+6	3.628E+6

The data and selected additional results are shown in the appendix.

Log-normal Model b. For the second log-normal model, let τ_j be sampled separately for each j for non-zero cells. The WinBUGS coding is as follows:

```
#  Log-Normal model b, Case 1
#  IBNP24-1  ...  7/4/2
#  12,000 iterations - 491 sec. (single chain)

model;
{
   for( j in 1 : 24 ) {
      mu[j] ~ dnorm(0,1.0E-6)
      tau[j] ~ dgamma(0.001,0.001)
      for( i in 1 : 48 ) {
         paid[i , j] ~ dlnorm(mu.z[i,j],tau.z[i,j])
         mu.z[i,j] <-  mu[j]*(1-n[i,j])-.00005*n[i,j]      #  mean = 1 if no claims
         tau.z[i,j] <-  tau[j]*(1-n[i,j])+10000*n[i,j]     #  variance = .0001 if no claims
         n[i , j] ~ dbern(p[j])                          #  = 1 if no claims
   } }

#  Determine zero cells
   for( j in 1 : 24 ) {
      z[j] ~ dnorm( 0.0,100)
      p[j] <- max(0,min(2.22 -  0.3 * mu[j] + z[j],0.99))}

#  Summary results
   for( i in 1 : 48 ) {
      paid.tot[i] <- sum(paid[i , ])
   }
   for( i in 26 : 48 ) {
      reserve[i] <- sum(paid[i , 50 - i:24])
   }
   reserve.tot <- sum(reserve[26:48])}
```

In comparison with model a, paid[,] means and standard deviations are generally lower for short lags and for $j = 24$. For reserve.tot, the mean is about 2% lower and the standard deviation is significantly lower. Consequently, the reserve.tot values at higher percentiles are more greatly reduced as the percentile level increases.

	Model a		Model b	
node	mean	sd	mean	sd
paid[48,2]	546600.0	186800.0	525200.0	94660.0
paid[48,3]	349600.0	119400.0	335700.0	60940.0
paid[48,4]	151500.0	51870.0	146300.0	32820.0
paid[48,5]	77110.0	26300.0	75200.0	19950.0
paid[48,6]	39920.0	13520.0	39730.0	12270.0
paid[48,7]	21900.0	7414.0	21650.0	6299.0
paid[48,8]	15040.0	5059.0	15060.0	5094.0
paid[48,9]	9195.0	3142.0	9349.0	3652.0
paid[48,10]	6723.0	2292.0	6646.0	2013.0
paid[48,11]	4938.0	1682.0	5101.0	2289.0
paid[48,12]	4130.0	1413.0	4205.0	1703.0
paid[48,13]	3473.0	1178.0	3713.0	1904.0
paid[48,14]	2799.0	959.9	2869.0	1220.0
paid[48,15]	2412.0	839.6	2536.0	1198.0
paid[48,16]	1803.0	621.2	1865.0	810.1
paid[48,17]	1721.0	606.1	1815.0	871.0
paid[48,18]	1724.0	602.2	1734.0	620.1
paid[48,19]	1847.0	643.1	1905.0	806.2
paid[48,20]	1709.0	602.3	1715.0	608.7
paid[48,21]	1790.0	625.1	1799.0	674.9
paid[48,22]	1112.0	410.6	1141.0	499.3
paid[48,23]	484.6	232.5	504.6	278.4
paid[48,24]	48.04	74.43	42.81	62.66

Statistics for total reserve are:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.944E+6	153100	1709.0	2.659E+6	2.937E+6	3.042E+6	3.142E+6	3.205E+6	3.271E+6

The corresponding results from model a are:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	3.004E+6	277100.0	2872.0	2.527E+6	2.981E+6	3.171E+6	3.362E+6	3.5E+6	3.628E+6

Selected additional results are shown in the appendix.

Log-normal Model c. This model is built on model b, but with the specification of p[] defined in terms of four stochastic parameters, as follows.

$$p_{ij} = x - y\,\mu_p + z_j\,,$$

where $x \sim N(2.2, 1/25)$,
$y \sim N(.3, 1/1000)$
$z_j \sim N(0, 1/200)$.

The relevant BUGS coding is:

```
# Determine zero cells
  x ~ dnorm(2.2, 25)
  y ~ dnorm(.3, 1000)
  for( j in 1 : 24 ) {
    z[j] ~ dnorm( 0.0,200)
    p[j] <- max(0,min(x -  y * mu[j] + z[j],0.99))
  }
```

The results for p[] are summarized below, in comparison with the values from model b.

	Model b		**Model c**	
node	**mean**	**sd**	**mean**	**sd**
p[10]	0.0	0.0	0.0	0.0
p[11]	7.17E-7	6.847E-5	0.0	0.0
p[12]	2.778E-5	8.162E-4	0.0	0.0
p[13]	1.168E-4	0.00181	0.0	0.0
p[14]	6.954E-4	0.00498	0.0	0.0
p[15]	0.00116	0.007521	0.0	0.0
p[16]	0.003947	0.0124	2.147E-5	8.21E-4
p[17]	0.004841	0.01462	2.315E-5	8.968E-4
p[18]	0.004851	0.01399	9.181E-6	5.015E-4
p[19]	0.004232	0.0139	9.556E-6	4.191E-4
p[20]	0.005251	0.0156	1.509E-5	6.798E-4
p[21]	0.004925	0.01504	8.885E-6	4.913E-4
p[22]	0.02065	0.03321	4.638E-4	0.004377
p[23]	0.08919	0.06914	0.02107	0.03562
p[24]	0.6841	0.06503	0.535	0.07427

Statistics for total reserve are:

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**75.0%**	**90.0%**	**95.0%**	**97.5%**
reserve.tot	2.946E+6	151600	1132.0	2.668E+6	2.939E+6	3.046E+6	3.142E+6	3.204E+6	3.265E+6

The corresponding results from model b are:

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**75.0%**	**90.0%**	**95.0%**	**97.5%**
reserve.tot	2.944E+6	153100	1709.0	2.659E+6	2.937E+6	3.042E+6	3.142E+6	3.205E+6	3.271E+6

B. Gamma Models.

P_{ij} are assumed to be gamma distributed as follows:

$$P_{ij} \sim \text{gamma}(r_{ij}, \mu_{ij}),$$

with mean r_{ij} / μ_{ij} and variance r_{ij} / μ_{ij}^2.

Handling of zero cells is similar to the log-normal model c: the logarithm of mean claims is assumed to follow a linear stochastic model.

For $P_{ij} \neq 1$ (i.e. non-zero cells), the gamma models incorporate an adjustment of r_{ij} and μ_{ij} to provide that both $E(P_{ij})$ and $Var(P_{ij})$ are increased by a factor of $\dfrac{1}{1-p_{ij}}$.

Gamma model a.

Assume the ratio R of standard deviation to mean is independent of j. Therefore $r_j = R^{-2}$, which implies r is independent of j.

The WinBUGS coding is as follows:

```
# 6/30/02
# Gamma model: no trend, seasonal, exposure change.
# Assume constant std dev / mean => r independent of j.
# 10000 iterations took 1005 sec.

model;
{
  r ~ dunif(0 , 30)
  for( j in 1 : 24 ) {
    mu[j] ~ dgamma(0.00000001,0.001)
    for( i in 1 : 48 ) {
      r.p[i , j] <- (r * (1 - n[i , j])) / (1 - p[j]) + 10000 * n[i , j]
      mu.p[i , j] <- mu[j] * (1 - n[i , j]) + 10000 * n[i , j]
      paid[i , j] ~ dgamma(r.p[i , j],mu.p[i , j])
    }
  }

# Determine zero cells
  x ~ dnorm(2.2, 25)
  y ~ dnorm(.3, 1000)
  for( j in 1 : 24 ) {
    z[j] ~ dnorm( 0.0,200)
    p[j] <- max(0,min(x - y * log(r / mu[j]) + z[j],0.99))
    for( i in 1 : 48 ) {
      n[i , j] ~ dbern(p[j])          # n = 1 for zero cell
    }
  }

# Summary results
  for( i in 1 : 48 ) {
    paid.tot[i] <- sum(paid[i , ])
  }
  for( i in 26 : 48 ) {
    reserve[i] <- sum(paid[i , 50 - i:24])
  }
  reserve.tot <- sum(reserve[26:48])
}
```

I ran 10,000 iterations after a burn-in of 4000.

The following is a comparison of p[] with the results of log-normal model c.

	LN Model c		Gamma Model a	
node	mean	sd	mean	sd
p[15]	0.0	0.0	0.0	0.0
p[16]	2.147E-5	8.21E-4	0.0	0.0
p[17]	2.315E-5	8.968E-4	0.0	0.0
p[18]	9.181E-6	5.015E-4	0.0	0.0
p[19]	9.556E-6	4.191E-4	0.0	0.0
p[20]	1.509E-5	6.798E-4	0.0	0.0
p[21]	8.885E-6	4.913E-4	0.0	0.0
p[22]	4.638E-4	0.004377	1.472E-5	8.971E-4
p[23]	0.02107	0.03562	0.001997	0.009551
p[24]	0.535	0.07427	0.7123	0.08579

Statistics for total reserve are:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.938E+6	254100	2344.0	2.485E+6	2.923E+6	3.1E+6	3.275E+6	3.379E+6	3.479E+6

Selected additional results are shown in the appendix.

Gamma model b.

For the second gamma model, let r_j be sampled separately for each j for non-zero cells. The WinBUGS coding is as follows:

```
# 6/27/02
# Gamma model b:  no trend, seasonal, exposure change.
# 10000 iterations took 1275 sec.

model;
{
  for( j in 1 : 24 ) {
    r[j] ~ dgamma(0.001,0.001)
    mu[j] ~ dgamma(0.001,0.001)
    for( i in 1 : 48 ) {
      r.p[i , j] <- (r[j] * (1 - n[i , j])) / (1 - p[j]) + 100 * n[i , j]
      mu.p[i , j] <- mu[j] * (1 - n[i , j]) + 100 * n[i , j]
      paid[i , j] ~ dgamma(r.p[i , j],mu.p[i , j])
    }
  }

# Determine zero cells
  x ~ dnorm(2.2, 25)
  y ~ dnorm(.3, 1000)
  for( j in 1 : 24 ) {
    z[j] ~ dnorm( 0.0,200)
    p[j] <- max(0,min(x -  y * log(r[j] / mu[j]) + z[j],0.99))
    for( i in 1 : 48 ) {
      n[i , j] ~ dbern(p[j])          # n = 1 for zero cell
    }
  }
```

```
# Summary results
  for( i in 1 : 48 ) {
    paid.tot[i] <- sum(paid[i , ])
  }
  for( i in 26 : 48 ) {
    reserve[i] <- sum(paid[i , 50 - i:24])
  }
  reserve.tot <- sum(reserve[26:48])
}
```

Statistics for total reserve are:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.936E+6	144900	1328.0	2.661E+6	2.932E+6	3.031E+6	3.126E+6	3.181E+6	3.226E+6

C. Bivariate Normal Regression Model.

Define $S_{ij} = \sum_{k \le j} P_{ik}$. For fixed j (j = 2, 3, …, 24), assume that $S_{i,j\text{-}1}$ and P_{ij} have a jointly normal interdependence. For a given j let σ_S^2 be the variance of $S_{i,j\text{-}1}$ and σ_P^2 be the variance of P_{ij} and ρ be the correlation coefficient. The covariance matrix is

$$\begin{pmatrix} \sigma_S^2 & \rho\sigma_S\sigma_P \\ \rho\sigma_S\sigma_P & \sigma_P^2 \end{pmatrix}.$$

The conditional distribution of P_{ij} for a given value of $S_{i,j\text{-}1}$ is normal with mean $\mu_{P|S}$ and standard deviation $\sigma_{P|S}$ where

$$\mu_{P|S} = \mu_P + \beta(S - \mu_S)$$

$$\beta = \rho\frac{\sigma_P}{\sigma_S} \text{ and}$$

$$\sigma_{P|S}^2 = \sigma_P^2(1 - \rho^2)\,.$$

The following relationships were used in construction of the model. (τ is the precision, related to variance by $\tau = 1/\sigma^2$).

$$(P_{ij} \mid S_{i,j-1}) \sim \mathrm{N}(\mu_{P|S;ij}, 1/\tau_{P|S;j})$$

$$S_{i,j-1} \sim \mathrm{N}(\mu_{S;j}, 1/\tau_{S;j})$$

$$\mu_{P|S;ij} = \mu_{P;j} + \beta(S_{i,j-1} - \mu_{S;j-1})$$

$$\rho_j = \beta_j\frac{\sigma_{S;j-1}}{\sigma_{P;j}}$$

$$\sigma_{P;j}^2 = \sigma_{P|S;j}^2 + \beta_j^2\sigma_{S;j-1}^2$$

The last two equations (for ρ and σ_P^2) are only for the purpose of accumulating ancillary information and have no effect on results. For each j, $\mu_{P;j}$ is estimated based on observed values of P_{ij}, and $\mu_{S;j}$ is based on observed values of S_{ij}. Noninformative prior distributions are assigned to $\tau_{P|S}$, τ_S and β. The distribution of P_{ij} is restricted to nonnegative numbers. Beyond that the model does not include any special provision for zero cells.

The WinBUGS coding is as follows.

```
# 7/19/02
# Bivariate Normal Regression Model – Case 1;   no trend, seasonal, exposure change.
# 10000 iterations took 90 sec.
# NO SPECIAL PROVISION FOR ZERO CELLS

model;
{
for( j in 1 : 23 ) {
  for( i in 1 : 48 ) {
    paid.sum[ i , j ] ~ dnorm( mu.s[ j ] , tau.s[ j ] )
                  # Note: for j > 49 - i this is a dummy sum, not equal to sum(paid[ i , 1 : j ]).
} }
for( j in 2 : 24 ) {
  for( i in 1 : 48 ) {
    paid[ i , j ] ~ dnorm( mu.p.s[ i , j ], tau.p.s[ j ] )I(0,)
    mu.p.s[ i , j ] <- mu.p[ j ]  + beta[ j ] * (paid.sum[ i , j - 1 ]- mu.s[ j - 1 ])
  }
  rho[ j ] <- beta[ j ] * sqrt( sig.sq.s[ j - 1 ] / sig.sq.p[ j ] )    # for info only; no impact on results
  mu.p[ j ] <- mean( paid[ 1 : 48 - j + 1 , j ] )
  sig.sq.p[ j ] <- sig.sq.p.s[ j ] + sig.sq.s[ j - 1 ] * pow( beta[ j ] , 2 ) # for info only; no impact on results
  sig.sq.p.s[ j ] <- 1 / tau.p.s[ j ]
}
for( j in 1 : 23 ) {
    mu.s[ j ] <- mean(paid.sum[ 1 : 48 - j , j ] )
    sig.sq.s[ j ] <- 1 / tau.s[ j ]
}
# Summary statistics
for( i in 26 : 48 ) {
  reserve[i] <- sum(paid[i , 50 - i : 24])
}
reserve.tot <- sum(reserve[26:48])

# priors
for( j in 1 : 23 ) {
  tau.s[ j ] ~ dunif( 0 , 100 )
  }
for( j in 2 : 24 ) {
  beta[ j ] ~ dnorm( 0 , 1.0E-6 )
  tau.p.s[ j ] ~ dunif( 0 , 100 )
} }
```

The following are results for total reserve.

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**75.0%**	**90.0%**	**95.0%**	**97.5%**
reserve.tot	2.944E+6	137300	1597.0	2.674E+6	2.943E+6	3.036E+6	3.122E+6	3.176E+6	3.213E+6

The mean and median results are in the same neighborhood as the other models, but the standard deviation is a bit lower. Thus reserves at the higher percentile levels are generally lower. It may be noted that one possible reason for the lower standard deviation may be that the structure of the bivariate normal model is more similar to the simulation spreadsheet used to generate the test data.

Selected additional results are presented in the appendix.

Case 1 reserve.tot **summary statistics:**

Model	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
log-norm a	3.004E+6	277100	2872.0	2.527E+6	2.981E+6	3.171E+6	3.362E+6	3.5E+6	3.628E+6
log-norm b	2.944E+6	153100	1709.0	2.659E+6	2.937E+6	3.042E+6	3.142E+6	3.205E+6	3.271E+6
log-norm c	2.946E+6	151600	1132.0	2.668E+6	2.939E+6	3.046E+6	3.142E+6	3.204E+6	3.265E+6
gamma a	2.938E+6	254100	2344.0	2.485E+6	2.923E+6	3.1E+6	3.275E+6	3.379E+6	3.479E+6
gamma b	2.936E+6	144900	1328.0	2.661E+6	2.932E+6	3.031E+6	3.126E+6	3.181E+6	3.226E+6
bivariate	2.944E+6	137300	1597.0	2.674E+6	2.943E+6	3.036E+6	3.122E+6	3.176E+6	3.213E+6

Case 2: Trend, exposure variations and seasonal variations are incorporated in the simulated claim lag table. Case 2 has 60 months of incurred claims and 24 lag months.

Trend. Let the monthly effect of trend be $(1 + t)$. Where the trend operates over a period of m months the cumulative effect is $(1 + t)^m$. t is assumed to be constant for all the incurred claim months represented in the model. It is assumed that the effect of trend is to proportionately increase medical care costs for all injuries or conditions, and that the effects of trend on mean and variance are as follows:

Mean: changed in proportion to $(1 + t)^m$.

Variance: changed in proportion to $(1 + t)^{2m}$.

Exposure Variation. Let the effect of exposure variation for a given incurred month i relative to a base month be k_i. For example, exposure variation would include the effects of change in group size or change in employee / family, employee / spouse, or employee age / sex distribution. Exposure variation is assumed to be equivalent to changing the size of the population. Therefore variance is assumed to be proportional to k_i. (Note that this assumption may be inappropriate for many forms of benefit change.) The effects on mean and variance are assumed to be as follows:

Mean: changed in proportion to k_i.

Variance: changed in proportion to k_i.

Seasonal Variation. Let the effect of seasonal variations on incurred claims in month i be s_i. Assume that seasonal variation is consistent from year to year. Therefore $s_i = s_{i+12m}$ for integers m. It is assumed that the effect of seasonal variation relates primarily to variations in the number or severity of injuries or conditions, and is not driven by shifts in price of treatment of particular injuries or conditions. Therefore variance is assumed to be proportional to s_i. The effects on mean and variance are assumed to be as follows:

Mean: changed in proportion to s_i.

Variance: changed in proportion to s_i.

Zero cell probability. $E(P_{ij}|\text{ non-zero}) = E(P_{ij}|\ n_{ij} = 0) = E(P_{ij}) / (1 - p_{ij})$, and $\text{Var}(P_{ij}|\ n_{ij} = 0) = \text{Var}(P_{ij}) / (1 - p_{ij})$. I.e.,

Mean: changed in proportion to $\dfrac{1}{1-p_{ij}}$.

Variance: changed in proportion to $\dfrac{1}{1-p_{ij}}$.

Summary. The following summarizes the effects of trend, exposure, seasonal and zero cells ("TESZ") on mean and variance.

Mean: changed in proportion to $\dfrac{(1+t)^m k_i s_i}{1-p_{ij}}$.

Variance: changed in proportion to $\dfrac{(1+t)^{2m} k_i s_i}{1-p_{ij}}$.

Log-normal Model.

Let μ and τ be the parameters of the log-normal distribution with random effects only. Let μ_p and τ_p be the parameters incorporating TESZ.

The mean is:

$$e^{\mu_p + \frac{1}{2\tau_p}} = \frac{(1+t)^i k_i s_i}{1-p_{ij}}\ e^{\mu + \frac{1}{2\tau}}.$$

The variance is:

$$e^{2\mu_p + \frac{1}{\tau_p}}(e^{\frac{1}{\tau_p}} - 1) = \frac{(1+t)^{2i} k_i s_i}{1-p_{ij}}\ e^{2\mu + \frac{1}{\tau}}(e^{\frac{1}{\tau}} - 1).$$

Therefore,

$$\mu_p = \mu + \ln\left(\frac{(1+t)^i \left(\dfrac{k_i s_i}{1-p_{ij}}\right)^{3/2}}{\left(1 + \left(\dfrac{k_i s_i}{1-p_{ij}} - 1\right) e^{-1/\tau}\right)^{1/2}} \right)$$

and

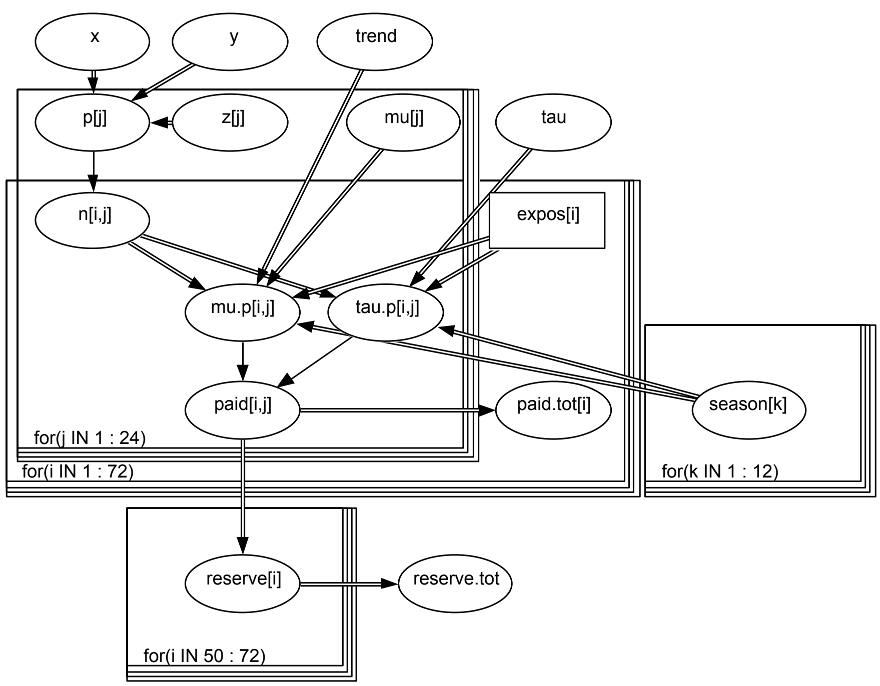
$$\tau_p = \frac{1}{\ln\left(1 + \frac{(1-p_{ij})\left(e^{1/\tau} - 1\right)}{k_i s_i}\right)}.$$

For Case 2, the prior adopted for p_{ij} has the same structure as that for Case 1 log-normal model c.

$$p_{ij} = x - y\,\mu_p + z_j\ ,$$

where $x \sim N(2.2, 1/25)$,
$y \sim N(.3, 1/1000)$
$z_j \sim N(0, 1/200)$.

The preliminary WinBUGS graph is as follows:



The WinBUGS coding is as follows:

```
#  Log-normal model, Case 2
#  IBNP4 ... 7/6/2.
#  10000 iterations - 2829 seconds (single chain).

model;
{
   for( i in 1 : I ) {
      for( j in 1 : 24 ) {
         paid[i , j] ~ dlnorm(mu.p1[i , j],tau.p1[i , j])
         mu.p1[i,j] <- mu.p[i,j]  * (1 - n[i , j]) - 0.005 * n[i , j]
         mu.p[i , j] <- mu[j] + log(trend) * (i - 1) + 1.5 * log((expos[ i ] / expos[ I ]) * season[mod12[ i ]]
/ ( 1 - p[ i , j ] )) -  0.5 * log(1 + ((expos[ i ] / expos[ I  ]) * season[mod12[i]] / ( 1 - p[ i , j ]) - 1) *
exp(( -1) / tau))
         mu.prelim[i , j] <- mu[j] + log(trend) * (i - 1) + 1.5 * log((expos[ i ] / expos[ I ]) *
season[mod12[ i ]] ) -  0.5 * log(1 + ((expos[ i ] / expos[ I  ]) * season[mod12[i]] - 1) * exp(( -1) /
tau))                  #  used to determine p[ i , j ]
         tau.p1[ i , j ] <- tau.p[ i , j ]  * (1 - n[ i , j ]) + 100 * n[ i , j ]
         tau.p[ i , j ] <- 1 / log(( exp(1 / tau) - 1 ) / (( expos[ i ] / expos[ I ] ) * season[mod12[ i ]] / ( 1 -
p[ i , j ])  ) + 1 )
      }
      paid.tot[i] <- sum( paid[ i , ])
      mod12[i] <- ((i - 1) / 12 - trunc((i - 1) / 12)) * 12 + 1
   }
   for( j in 1 : 24 ) {
      mu[j] ~ dnorm(5,1.0E-2)
   }
   tau ~ dgamma(0.001,0.001)I(1.0E-6,)
#  Determine zero claim cells.
   x ~ dnorm(2.2, 25)
   y ~ dnorm(.3, 1000)
   for( j in 1:24 ) {
      z[j] ~ dnorm( 0.0,200)
      for( i in 1:I ) {
         n[ i , j ] ~ dbern(p[ i , j ])
                                    #  n = 1 means no claims
         p[ i , j ] <- max( 0, min( x - y * mu.prelim[ i , j ] + z[ j ], .99 ))     #  prior based on anal of simulator spreadsheet
      }
   }

   trend ~ dunif( .98,1.03)
   trnd <- pow(trend,12)-1                        #  annual trend rate

   for( k in 1 : 11 ) {
      season[k] ~ dnorm(1,15)
   }
   season[12] <- 12 - sum(season[1:11])                            #  force seasonal factors to average 1.0

   for( i in I - 22 : I ) {
      reserve[i] <- sum(paid[ i , I + 2 - i :24])
   }
   reserve.tot <- sum(reserve[ I - 22 : I ])
}
```

Trend results. The following are the summary statistics for annualized trend. For comparison, the sample data was generated using a value of .12 in the spreadsheet simulator.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
trnd	0.1303	0.01364	6.399E-4	0.1037	0.1305	0.157	4001	10000

Seasonal variation results. The following are the summary statistics for seasonal variation. The column headed "True Value" is the seasonal factor used by the spreadsheet simulator to produce the sample data.

node	mean	sd	MC error	2.5%	median	97.5%	True Value	Error
season[1]	0.9145	0.0461	0.00192	0.8268	0.9147	1.004	1.00	-.09
season[2]	1.038	0.04929	0.001762	0.9394	1.037	1.133	1.09	-.05
season[3]	1.116	0.05486	0.002304	1.011	1.115	1.226	1.18	-.06
season[4]	1.144	0.05266	0.002271	1.042	1.143	1.248	1.09	.05
season[5]	0.9899	0.04903	0.002018	0.8943	0.9918	1.083	.91	.08
season[6]	1.09	0.05232	0.001914	0.9892	1.089	1.194	1.00	.09
season[7]	1.098	0.05126	0.002023	0.9934	1.098	1.199	1.09	.01
season[8]	0.9087	0.04765	0.001664	0.8176	0.9081	1.001	1.00	-.09
season[9]	0.9828	0.05078	0.002157	0.8875	0.9818	1.085	.91	.07
season[10]	1.003	0.04997	0.001967	0.9061	1.003	1.101	1.00	.00
season[11]	0.8872	0.04743	0.00168	0.7933	0.8868	0.98	.91	-.02
season[12]	0.8291	0.04666	9.959E-4	0.7396	0.8278	0.9223	.82	.01

The sum of the absolute errors is .62. The sum of the deviations of the "true value" from 1.00 is .90. In every case, the "true value" is within the 95% credible interval.

The seasonal factors for the last months of the year can be particularly important if the model were being used to determine reserves for claims incurred but not paid. To the extent late-year seasonal factors are understated, the reserve would be correspondingly understated. Conversely, a seasonal factors overstatement could lead to a reserve overstatement.

The summary statistics for total reserve (10,000 iterations) are as follows.

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.959E+6	600900	6887.0	2.071E+6	2.865E+6	3.267E+6	3.709E+6	4.05E+6	4.413E+6

Compared to the results of Case 1, the standard deviation is higher and the 95% credible interval is considerably broader. I have not analyzed the question of the extent to which this is due to additional volatility resulting from trend, seasonal and exposure variations being included in the original data, versus characteristics of the model with its numerous additional parameters.

In order to gain a sense of the impact of the seasonal factor, the model was run with seasonal factors forced to 1.0. Summary statistics for total reserve, based on a run of 1000 iterations after a burn-in of 4000, are as follows.

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	3.238E+6	631700	17670.0	2.267E+6	3.145E+6	3.626E+6	4.087E+6	4.374E+6	4.612E+6

This result would represent an overstatement in the reserve, due to elimination of the seasonal factor from the model. Note the increase in standard deviation. The values of reserve.tot at most higher percentile levels increased more than proportionately. Thus, not only would the estimated reserve be higher, but estimates of uncertainty are also higher. Conversely, if the true December factor had been over 1.0, to ignore analysis of the seasonal factor would increase the risk of understating the reserve. This suggests that, at least as far as this example is concerned, it is helpful to include seasonal variations in the analysis, even though there is significant uncertainty. Given the uncertainties involved (as with any parameter) judgment is called for.

Bivariate Normal Regression Model.

The approach for this model is to remove the effects of exposure, trend and seasonal from the data values S_{ij} and P_{ij}. The resulting adjusted data is assumed to be a bivariate normal distribution as analyzed in Case 1. The model incorporates the following relationships.

$$P_{ij}^{base} = P_{ij} / (1+t)^i s_i k_i$$

$$S_{ij}^{base} = S_{ij} / (1+t)^i s_i k_i$$

$\mu_{P;j}^{base}$ = Sum of relevant known values for each j.

$\mu_{S;j}^{base}$ = Sum of relevant known values for each j.

$$\mu_{P|S;ij}^{base} = \mu_{P;j}^{base} + \beta(S_{i,j-1}^{base} - \mu_{S;j-1}^{base})$$

$\tau_{P|S}^{base}$, τ_{S}^{base} and β are assigned non-informative priors.

$$\rho_j = \beta_j \frac{\sigma_{S;j-1}^{base}}{\sigma_{P;j}^{base}}$$

$$(\sigma_{P;j}^{base})^2 = (\sigma_{P|S;j}^{base})^2 + \beta^2 (\sigma_{S;j}^{base})^2$$

$$\tau_{P|S;ij} = \tau_{P|S;j}^{base} / (1+t)^{2i} s_i k_i$$

$$\tau_{S;ij} = \tau_{S;j}^{base} / (1+t)^{2i} s_i k_i$$

$$\mu_{P|S;ij} = \mu_{P|S;j}^{base} (1+t)^i s_i k_i$$

$$\mu_{S;ij} = \mu_{S;j}^{base} (1+t)^i s_i k_i$$

$$(P_{ij} \mid S_{i,j-1}) \sim \mathrm{N}(\mu_{P|S;ij}, 1/\tau_{P|S;ij})$$

$$S_{i,j} \sim \mathrm{N}(\mu_{S;ij}, 1/\tau_{S;ij})$$

The WinBUGS coding is presented below. An unfortunate feature of the model is its excruciating slowness. After convergence, each iteration took an average of 44 seconds, which averages out to about 12 hours per 1000 iterations. Thus a run of 4500 iterations on my 1000Mhz IBM R30 took in excess of two days.

WinBUGS coding is as follows.

```
#  7/19/02
#  Case 2 - Bivariate Normal Regression Model.
#  500 iterations took 22094 sec.
#  NO SPECIAL PROVISION FOR ZERO CELLS

model;
{
for( j in 1 : 23 ) {
  for( i in 1 : I ) {
    paid.sum[ i , j ] ~ dnorm( mu.s[ i , j ] , tau.s[ i , j ] )
         #  Note: for j > I + 1 - i this is a dummy sum, not equal to sum(paid[ i , 1 : j ]).
    paid.base.sum[ i , j ] <- paid.sum[ i , j ] / (expos[ i ] * pow(trend, i - 1) * season[mod12[ i ]])
    mu.s[ i , j ] <- mubase.s[ j ] * expos[ i ] * pow(trend, i - 1) * season[mod12[ i ]]
    tau.s[ i , j ] <- taubase.s[ j ] / (expos[ i ] * pow(trend, 2 * i - 2) * season[mod12[ i ]])
  }
  mubase.s[ j ] <- mean(paid.base.sum[ 1 : I - j , j ])
  sig.sq.s[ j ] <- 1 / taubase.s[ j ]
}

#  Transformations for exposure, trend, seasonal.
for( j in 2 : 24 ) {
  for( i in 1 : I ) {
    paid[ i , j ] ~ dnorm( mu.p.s[ i , j ], tau.p.s[ i , j ] )I(0,)
    paid.base[ i , j ] <- paid[ i , j ] / (expos[ i ] * pow(trend, i - 1) * season[mod12[ i ]])
    mu.p.s[ i , j ] <- mubase.p.s[ i , j ] * expos[ i ] * pow(trend, i - 1) * season[mod12[ i ]]
    tau.p.s[ i , j ] <- taubase.p.s[ j ] / (expos[ i ] * pow(trend, 2 * i - 2) * season[mod12[ i ]])
  }
  mubase.p[ j ] <- mean( paid.base[ 1 : I - j + 1 , j ] )
  sig.sq.p.s[ j ] <- 1 / taubase.p.s[ j ]
}

#  Bivariate normal relationships
for( j in 2 : 24 ) {
  for( i in 1 : I ) {
    mubase.p.s[ i , j ] <- mubase.p[ j ] + beta[ j ] * (paid.base.sum[ i , j - 1 ] - mubase.s[ j - 1 ])
  }
  rho[ j ] <- beta[ j ] * sqrt( sig.sq.s[ j - 1 ] / sig.sq.p[ j ] )  # for info only; no impact on results
  sig.sq.p[ j ] <- sig.sq.p.s[ j ] + sig.sq.s[ j - 1 ] * pow( beta[ j ] , 2 )  # for info only; no impact on results
}

trend ~ dunif( .98,1.04)           #  monthly trend factor
trnd <- pow(trend,12)-1            #  annual trend rate

for( k in 1 : 11 ) {                              #  Seasonal factors
  season[k] ~ dnorm(1, 15)
}
season[12] <- 12 - sum(season[1:11])                   #  force seasonal factors to average 1.0

#  Summary statistics
for( i in I - 22 : I ) {
  reserve[i] <- sum(paid[i , I + 2 - i : 24])
}
reserve.tot <- sum(reserve[ I - 22 : I ])
```

```
# priors
for( j in 1 : 23 ) {
  taubase.s[ j ] ~ dunif( 0 , 100 )
  }
for( j in 2 : 24 ) {
  beta[ j ] ~ dnorm( 0 , 1.0E-6 )
  taubase.p.s[ j ] ~ dunif( 0 , 100 )
}

# month index
for( i in 1 : I ) {
  mod12[i] <- ((i - 1) / 12 - trunc((i - 1) / 12)) * 12 + 1
} }
```

Because of the slow speed of this model, only 500 iterations were run after a burn-in of 4000.

Trend results. The following are the summary statistics for annualized trend. For comparison, the sample data was generated using a value of .12.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
trnd	0.1218	0.00225	1.728E-4	0.117	0.122	0.1259	4002	500

The results are closer to the "true" value of .12, the standard deviation is much lower (.00225 compared to .01364 under the log-normal model), and the ranges of results at various percentile levels are much narrower.

Seasonal Variation Results. The results for the seasonal factors are:

node	mean	sd	MC error	2.5%	median	97.5%	True Value	Error
season[1]	0.9689	0.01057	0.00155	0.9493	0.9692	0.9899	1.00	-.03
season[2]	1.151	0.01385	0.00234	1.127	1.15	1.186	1.09	.06
season[3]	1.133	0.009905	0.001499	1.111	1.132	1.154	1.18	-.05
season[4]	1.058	0.01181	0.001745	1.034	1.058	1.079	1.09	-.03
season[5]	0.9644	0.01038	0.00148	0.9395	0.9651	0.9847	.91	.05
season[6]	1.03	0.01081	0.001474	1.007	1.032	1.048	1.00	.03
season[7]	1.047	0.01214	0.001788	1.028	1.048	1.071	1.09	-.04
season[8]	0.9746	0.01215	0.001886	0.9512	0.9754	0.9972	1.00	-.03
season[9]	0.8567	0.01025	0.001392	0.838	0.8568	0.874	.91	-.05
season[10]	0.9939	0.01121	0.001423	0.9711	0.9959	1.014	1.00	-.01
season[11]	1.008	0.01196	0.002113	0.9845	1.008	1.039	.91	.10
season[12]	0.8147	0.01116	5.931E-4	0.7934	0.8141	0.8375	.82	-.01

The sum of the absolute errors is .49 – a somewhat better result than for the log-normal model at .62.

For eleven of the twelve variables, the true value falls outside of the 95% credible interval, suggesting that more work needs to be done and this model should be approached cautiously.

Total reserve results are as follows.

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.675E+6	138400	6768.0	2.418E+6	2.677E+6	2.768E+6	2.848E+6	2.912E+6	2.957E+6

In comparison, the results for the log-normal model were:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.959E+6	600900	6887.0	2.071E+6	2.865E+6	3.267E+6	3.709E+6	4.05E+6	4.413E+6

Under the bivariate model, the standard deviation is much lower and the ranges of results at various percentile levels are much narrower.

In consideration of the slow iteration time and the poor 95% credible interval results for seasonal variations, additional future work is called for here.

For those who may be curious, the "actual" runoff claims generated by the simulator spreadsheet were 2,669,715.

Case 3. For Case 3, assume there exists a separate body of pooled claim experience with characteristics similar to the group of interest. The data includes 60 months of paid experience with twelve months of claim lags for both the group and the pool.

Let E_{Gi} be the group's exposure for month i. Let E_{-Gi} be the pool's exposure. Let P_{Gij} and P_{-Gij} be the corresponding paid claims.

Then $\frac{E(P_{Gij})}{E_{Gi}} = q\frac{E(P_{-Gij})}{E_{-Gi}}$ where q represents the relative loss level of the group versus the pool.

The relative variances of P_{Gij} and P_{-Gij} depend on whether the differences in loss level arise from differences in utilization or in unit costs.
Assume $\frac{Var(I_{Gi})}{E_{Gi}} = q^a\frac{Var(P_{-Gi})}{E_{-Gi}}$ where $a \in [0, 1]$; $a = 1$ would correspond to cost differences arising entirely from utilization, and $a = 2$ would correspond to differences arising from unit costs. Note it is also conceivable that $a < 0$ or $a > 1$ where, for example, unit costs are lower but utilization rates are higher.

Case 3 was analyzed only under the log-normal model.

Log-normal Model.

For the log-normal model, the above imply that the expressions developed for Case 2 using TESZ are modified as follows.

The mean is:

$$e^{\mu_p+\frac{1}{2\tau_p}} = \frac{(1+t)^i k_i s_i q}{1-p_{ij}}\, e^{\mu+\frac{1}{2\tau}}.$$

The variance is:

$$e^{2\mu_p+\frac{1}{\tau_p}}\left(e^{\frac{1}{\tau_p}}-1\right) = \frac{(1+t)^{2i} k_i s_i q^a}{1-p_{ij}}\, e^{2\mu+\frac{1}{\tau}}\left(e^{\frac{1}{\tau}}-1\right).$$

Therefore,

$$\mu_p = \mu + \ln\left(\frac{(1+t)^i\left(\frac{k_i s_i}{1-p_{ij}}\right)^{3/2} q^{2-a/2}}{\left(1+\left(\frac{k_i s_i q^{2-a}}{1-p_{ij}}-1\right)e^{-1/\tau}\right)^{1/2}}\right)$$

and

$$\tau_p = \frac{1}{\ln\left(1+\frac{(1-p_{ij})\left(e^{1/\tau}-1\right)}{k_i s_i q^{2-a}}\right)}.$$

The WinBUGS coding is as follows:

```
#  Log-normal model, Case 3
#  IBNP6 ... 6/14/2
#  500 iterations took 890 secs.
#  10000 iterations after burn-in of 4000 - 13407 seconds.

model;
{
#  Pool results
   for( i in 1 : I ) {
      for( j in 1 : 12 ) {
         paid.pool[ i , j ] ~ dlnorm(mu.pool[ i , j ],tau.pool[ i ])
         mu.pool[ i , j ] <- mu[ j ] + log(trend) * (i - 1) + 1.5 * log((expos.pool[ i ] / expos.pool[ I ]) *
season[mod12[ i ]]) -  0.5 * log(1 + ((expos.pool[ i ] / expos.pool[ I ]) * season[mod12[ i ]] - 1) *
exp(( -1) / tau))
      }
      tau.pool[ i ] <- 1 / log(( exp(1 / tau) - 1 ) / (( expos.pool[ i ] / expos.pool[ I ] ) * season[mod12[ i
]] ) + 1 )
      paid.pool.tot[ i ] <- sum(paid.pool[ i , ])
      mod12[ i ] <- ((i - 1) / 12 - trunc((i - 1) / 12)) * 12 + 1
   }
```

```
for( j in 1 : 12 ) {
    mu[ j ] ~ dnorm(5,1.0E-2)
  }
  tau ~ dgamma(0.001,0.001)

# Group results
  for( i in 1 : I ) {
    for( j in 1 : 12 ) {
      paid[ i , j ] ~ dlnorm(mu.p1[ i , j ],tau.p1[ i , j ])
      mu.p1[ i , j ] <- mu.p[ i , j ]  * (1 - n[ i , j ]) - 0.005 * n[ i , j ]
      mu.p[ i , j ] <- mu[ j ] + log(trend) * (i - 1) + 1.5 * log((expos[ i ] / expos.pool[ I ]) *
season[mod12[ i ]] / ( 1 - p[ i , j ]) ) + (( 2 - a / 2) * log(q)) -  0.5 * log(1 + ((expos[ i ] / expos.pool[ I
]) * season[mod12[ i ]] * pow( q , 2 - a ) / ( 1 - p[ i , j ])  - 1) * exp(( -1) / tau))
      mu.prelim[ i , j ] <- mu[ j ] + log(trend) * (i - 1) + 1.5 * log((expos[ i ] / expos.pool[ I ]) *
season[mod12[ i ]] ) + (( 2 - a / 2) * log(q)) -  0.5  * log(1 + ((expos[ i ] / expos.pool[ I ]) *
season[mod12[ i ]] * pow( q , 2 - a ) - 1) * exp(( -1) / tau))          #  used to determine p[i,j]
      tau.p1[ i , j ] <- tau.p[ i , j ]  * (1 - n[ i , j ]) + 100 * n[ i , j ]
      tau.p[ i , j ] <- 1 / log(( exp(1 / tau) - 1 ) / (( expos[ i ] / expos.pool[ I ] ) * season[mod12[ i ]] *
pow( q , 2 - a ) / ( 1 - p[ i , j ])  ) + 1 )
    }
    paid.tot[ i ] <- sum( paid[ i , ])
  }

# Group experience factors
  q ~ dnorm(1,100)I(.01,)          #  group experience ratio;  sd .10
  a ~ dunif(1,2)              #  extent to which q is from utilization ( 1 ) or price differences ( 2 ).

# Determine zero claim cells.
  x ~ dunif(0, 5)
  y ~ dunif(0, .6)
  for( j in 1:12 ) {
    z[ j ] ~ dnorm( 0, 200 )
    for( i in 1:I ) {
      n[ i , j ] ~ dbern(p[ i , j ])
                              #  n = 1 means no claims
      p[ i , j ] <- max( 0, min( x - y * mu.prelim[ i , j ] + z[ j ], .99 ))
    }
  }

# Determine trend
  trend ~ dunif( .98,1.03)
  trnd <- pow(trend,12)-1                 #  annual trend rate

# Determine seasonal variations
  for( k in 1 : 11 ) {
    season[ k ] ~ dnorm(1,15)             #  sd .26
  }
  season[12] <- 12 - sum(season[ 1 : 11 ])                #  force seasonal factors to average 1.0

# Determine reserve
  for( i in I - 10 : I ) {
    reserve[ i ] <- sum(paid[ i , I + 2 - i : 12 ])
  }
  reserve.tot <- sum(reserve[ I - 10 : I ])
}
```

The simulator spreadsheet used an experience factor *q* of 1.00. The results are:

node	mean	sd	sample
q	1.008	0.025	10000

With simulated q so close to 1.0, the results for *a* are predictably unhelpful: approximately a uniform distribution on [0, 1].

node	mean	sd	sample
a	1.507	0.288	10000

The results for trend are close to the mark:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
trnd	0.121	0.006196	2.444E-4	0.1088	0.1209	0.1333	4001	10000

Recall that Case 2 log-normal results were:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
trnd	0.1303	0.01364	6.399E-4	0.1037	0.1305	0.157	4001	10000

Sources of the Case 3 greater accuracy and lower standard deviation can derive from the greater information of the pooled data, and also from the fact that the data only has twelve months of lag, compared to 24 for the Case 2 data.

The results for season are:

node	mean	sd	MC error	2.5%	median	97.5%	True Value	Error
season[1]	0.9697	0.02224	6.95E-4	0.9252	0.9701	1.012	1.00	-.03
season[2]	1.098	0.02258	7.398E-4	1.054	1.098	1.142	1.09	.01
season[3]	1.168	0.02429	8.404E-4	1.12	1.167	1.214	1.18	-.01
season[4]	1.123	0.02311	7.197E-4	1.077	1.123	1.167	1.09	.03
season[5]	0.9255	0.02202	7.203E-4	0.885	0.9249	0.9702	.91	.02
season[6]	0.9319	0.0226	7.702E-4	0.8892	0.9318	0.9767	1.00	-.07
season[7]	1.092	0.02312	7.894E-4	1.047	1.092	1.139	1.09	.00
season[8]	1.071	0.02366	7.658E-4	1.023	1.072	1.117	1.00	.07
season[9]	0.9028	0.02218	6.459E-4	0.8581	0.9034	0.9469	.91	-.01
season[10]	1.008	0.0232	6.835E-4	0.963	1.008	1.053	1.00	.01
season[11]	0.936	0.02275	7.98E-4	0.8914	0.9357	0.9805	.91	.03
season[12]	0.7739	0.02079	2.018E-4	0.7336	0.7737	0.8147	.82	-.05

The sum of the absolute errors is .34. For season[6], [8] and [12], the "true value" falls outside of the 95% credible interval. The high rate of over-dispersed values may be related to the fact that P_{ij} values are correlated across j, while the model treats them as independent.

The results for reserve.tot are:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	929800.0	300400.0	3496.0	549300	865900.0	1.056E+6	1.295E+6	1.492E+6	1.694E+6

See the appendix for additional results.

Case 3, no pooling. To get a sense of the value added by incorporation of the pooled elements into the model, the group data was processed under the same model, but without reference to the pooled experience. The results are summarized below.

node	With Pooling mean	sd	Without Pooling mean	sd	True Value	With Pooling Error	Without Pooling Error
reserve.tot	929800	300400	9.98E+5	342600			
trnd	0.121	0.006196	0.109	0.02154	.12	.001	-.011
season[1]	0.9697	0.02224	0.9327	0.06343	1.00	-.03	-.07
season[2]	1.098	0.02258	1.106	0.06803	1.09	.01	.02
season[3]	1.168	0.02429	1.041	0.0656	1.18	-.01	-.14
season[4]	1.123	0.02311	1.095	0.07003	1.09	.03	.01
season[5]	0.9255	0.02202	0.966	0.06617	.91	.02	.06
season[6]	0.9319	0.0226	0.9004	0.06346	1.00	-.07	-.10
season[7]	1.092	0.02312	1.067	0.06936	1.09	.00	-.02
season[8]	1.071	0.02366	0.9865	0.06895	1.00	.07	-.01
season[9]	0.9028	0.02218	1.168	0.07568	.91	-.01	.26
season[10]	1.008	0.0232	1.125	0.07135	1.00	.01	.13
season[11]	0.936	0.02275	0.8086	0.0621	.91	.03	-.10
season[12]	0.7739	0.02079	0.8047	0.06487	.82	-.05	-.02

These results suggest value was successfully added by incorporating pooling into the model. With pooling, standard deviations of results are significantly lower, and results for trend and seasonal variation are significantly more accurate.

Case 3, alternative data set. Primarily for the purpose of observing the performance of q and a, I generated a separate set of claim lag data with q set to .90. The only change in the model was to increase the range of the prior on a to [0, 3]. The results based on the last 2000 iterations of 6000 are summarized below.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
q	0.9322	0.02609	9.827E-4	0.8834	0.9322	0.983	4001	2000
a	2.318	0.622	0.02061	0.6289	2.506	2.98	4001	2000

The results for q appear reasonable. The results for a show a quite broad distribution, of little potential help in revealing anything meaningful about the underlying data. Given the nature of the relationships, this is not unexpected.

The results for seasonal variations are:

node	mean	sd	MC error	2.5%	median	97.5%	True Value	Error
season[1]	1.038	0.02245	0.001682	0.9952	1.04	1.081	1.00	.04
season[2]	1.103	0.02404	0.001896	1.058	1.101	1.15	1.09	.01
season[3]	1.204	0.02291	0.001722	1.157	1.205	1.246	1.18	.02
season[4]	1.107	0.02283	0.00156	1.065	1.108	1.148	1.09	.02
season[5]	0.8446	0.01931	0.001451	0.8062	0.845	0.8814	.91	-.07
season[6]	1.037	0.02101	0.001189	0.9943	1.037	1.08	1.00	.04
season[7]	1.116	0.02259	0.001354	1.07	1.116	1.162	1.09	.03
season[8]	0.9696	0.02103	0.001507	0.9284	0.9699	1.017	1.00	-.03
season[9]	0.914	0.01981	0.001457	0.8739	0.9134	0.9514	.91	.00
season[10]	0.9541	0.02214	0.001711	0.9139	0.9533	0.9989	1.00	-.05
season[11]	0.8772	0.01955	0.001445	0.8397	0.8784	0.9141	.91	-.03
season[12]	0.8344	0.01979	5.021E-4	0.7955	0.8344	0.8734	.82	.01

For two of the twelve variables, the true value falls outside of the 95% credible interval. The sum of the absolute errors is .35.

Execution

I would like to pose a question of execution: *Can it become practical to apply these tools to the routine and efficient solution of actuarial tasks?*

- Is the software ready to rely upon for development of results for clients? The software is free. Being free, is it worth what you pay for it?
- In the working environment of an insurance or consulting firm, can a sufficient number of actuaries be trained to a sufficient level of proficiency?
- How might a knowledge base of actuaries proficient in Bayesian modeling be developed within a firm?
  - What would be the economics of developing such knowledge using internal firm resources?
  - To what extent can such knowledge be built with the assistance of outside resources? How would that be structured to ensure permanence, i.e. to ensure vitality and strengthening of the methodology over time within the firm culture?
  - Does deep insight into the methodology need to be resident in senior actuarial personnel? … Or is it sufficient to concentrate deep knowledge in people with under five or perhaps ten years of experience? Is it appropriate for senior people to exercise strong leadership roles in the sale, execution and / or delivery of a project involving Bayesian modeling if they have not had in-depth experience in the Bayesian tools being utilized?
- From the perspective of project documentation and quality assurance standards and procedures, what features of Bayesian analysis might suggest different standards compared to other actuarial projects? What unique procedures should be established to ensure competent completion of projects, along with attendant review and documentation?

Summary and Conclusions

The analyses presented above offer support for the proposition that an explicit Bayesian approach using WinBUGS has much to offer in the analysis of claim runoff reserves, and for other purposes as well. The word "potential" is operative: while there is evidence of potential, there is also evidence of aspects requiring careful thought and additional work. The tool offers the potential to squeeze a great deal of information out of a limited amount of data. A variety of important parameters can be brought into the analyses, in a completely integrated fashion. Specific stochastic results are produced, offering actuaries the potential of firmer ground in estimating appropriate margins for various contingencies.

Can concrete Bayesian methodologies be further developed and applied to the routine solution of actuarial tasks?

Would such methodologies present particular challenges with respect to project documentation and quality assurance standards and procedures?

It occurs to me that use of methodologies similar to those discussed above may present challenging questions for existing insurance company consulting firm project documentation and quality assurance standards and procedures. An explicit Bayesian methodology has unique characteristics. A firm using these tools should give careful thought as to any special procedures or safeguards that should be established for the application and documentation of these tools, the testing of models, the processes related to each project, and independent review of projects and results. I am concerned that these projects might become "black boxes", where a given worker may not have sufficient insight into the dynamics within the model.

For the actuary who may wish to begin exploring the use of these tools, perhaps a suggestion is in order: to echo the advice given in the WinBUGS documentation, I urge you to **keep it simple**. Pare your model down to the absolute minimum number of parameters with the simplest relationships. Then, when your model is working at a simplified level, enhance it with additional parameters.

Finally, permit me to confess that working in a Bayesian medium using WinBUGS has significant entertainment value … it is just plain fun!

Appendix

Spreadsheet Simulation of Claim Lag Data.

Paid claims were generated using a spreadsheet constructed as follows.

Incurred base claims I_i^{base} for month i are assumed to be log-normal with parameters μ and σ. (σ represents the conventional notation of standard deviation of the logarithm of the random variable.) Incurred claims I_i were generated based on the following parameters.

$$\mu_i = \mu + \ln(k_i) + i\ln(1+t) + \ln(s_i) + \ln(x) \text{ and}$$

$$\sigma_i^2 = \sigma^2 + \ln(k_i) + 2i\ln(1+t) + \ln(s_i) + \ln(x),$$

where k_i is the exposure index, $(1+t)$ is the monthly trend factor, s_i is the seasonal index, and x is the group experience level.

Paid claims are based on the assumption that, for each lag duration, the ratio of paid claims to claims incurred but not paid is beta distributed.

$$\theta_{ij} \sim \text{Beta}(\alpha_j, \beta_j)$$

$$P_{ij} = \theta_{ij}\left(I_i - \sum_{k \le j-1} P_{ik}\right).$$

Finally, the model incorporates a linear transformation algorithm to permit the runoff to be compressed or stretched out by a factor, and/or to be accelerated or delayed by a specific number of months.

Model A, Case 1.

Case 1 data is as follows.

paid[,1] paid[,2] paid[,3] paid[,4] paid[,5] paid[,6] paid[,7] paid[,8] paid[,9] paid[,10] paid[,11]
paid[,12] paid[,13] paid[,14] paid[,15] paid[,16] paid[,17] paid[,18] paid[,19] paid[,20] paid[,21]
paid[,22] paid[,23] paid[,24]
211624 647221 385891 138545 82125 53069 21963 13816 12668 5758 3643 7105 5241
2030 3371 1719 1186 2971 1379 1307 1839 1187 555 142
269340 505290 317900 155199 70073 30447 19278 8622 4520 6702 2728 2695 1932 2301
1144 2955 2255 900 1043 1042 755 409 182 1
373689 551046 405628 147995 75434 38185 22945 20491 11896 9049 7082 4056 3532
3985 4754 2680 2831 2457 2074 1877 1775 1119 550 148
253633 577563 315134 149181 83618 41923 23040 8151 7163 7777 7992 6547 2245 4693
2563 1121 2459 2443 1012 1343 1523 1417 791 132
162174 576547 421885 175306 66551 51722 18939 12382 7287 6152 3347 5020 3367
2438 2314 1287 1390 1312 1246 1247 1614 857 379 1
199657 455336 314833 129502 41741 24734 14666 10951 11677 5635 3269 2698 2254
2516 1474 1279 991 1136 1490 1222 920 739 492 1
316728 398045 297925 100047 38269 48688 20926 17261 10963 9217 5725 3103 3085
2168 1898 1432 983 1202 912 1347 1207 764 220 1

251010 633650 379466 211127 86310 47078 21716 12336 11006 5041 4774 1959 4654
3368 2199 3142 1648 1195 2309 2175 1471 1017 676 131
235048 513811 342384 102204 90468 43850 23376 25934 16831 5617 4483 4551 2488
1570 1956 1246 1156 1771 1979 2103 1925 1631 550 123
273067 526505 332636 167084 54590 54576 18027 14262 8575 4045 4220 2482 2795
1462 849 2334 1261 1755 1453 1583 1874 1290 563 1
190093 403813 389857 193303 64830 36123 24022 7573 9121 6035 4573 6025 3370 2479
2014 1950 2586 1661 1448 1160 1248 506 270 1
228303 604526 373144 156780 54624 40900 24809 21942 12304 3923 6910 4656 3160
2860 2697 1403 1534 2322 2145 1957 1830 1130 402 1
200425 556621 331735 149507 81987 50265 15289 19209 12281 6382 5519 4595 3581
3010 3701 1041 2331 2164 1128 1355 1746 1097 389 1
300203 532019 300084 183558 75862 38974 22849 16921 7843 9368 5829 3419 4635
2352 4270 2034 1800 1649 2069 2072 2455 1261 439 1
284247 509527 382039 193408 102584 41832 34237 14640 6004 9615 6943 4081 4278
3799 4164 2241 3267 1791 3537 3435 2768 1893 927 129
215477 582423 330778 142859 72560 51014 13656 13180 8572 6330 4956 4699 6153
3153 2638 1573 1257 690 1256 2332 1783 892 496 1
273657 498423 301260 182077 111135 64140 27079 13347 18272 9115 8020 4287 1587
2408 4651 1668 1910 2393 1820 1726 1015 819 413 1
214513 469650 336803 91762 49248 32241 17892 12615 8767 6143 6565 2022 2245 1253
1435 846 1958 1363 1271 1474 1720 932 596 1
244772 522127 236137 185926 70271 42146 20004 9241 4320 7264 6339 5861 4130 2832
1601 1695 1177 1402 1548 1665 2487 1292 476 1
257812 493305 291596 132435 54776 28761 17790 17412 7076 7450 3604 3823 1602
4523 2643 2376 801 1485 3301 1771 1526 1309 375 1
283886 556543 419986 145064 65331 54593 24764 22305 10739 4437 5818 6064 3654
3570 3422 3422 3985 2446 4134 2664 2970 1888 825 143
276589 597774 313537 147279 86945 46532 44499 24781 9771 8725 3439 3378 3709
4652 1549 3730 2312 2326 3189 2558 3248 1880 803 146
251490 464166 345800 102889 58520 29872 11511 17804 7706 7846 3657 2345 4181
1655 1781 1230 1103 1275 1628 1325 1656 1196 681 120
213226 525236 208173 161182 66603 43581 24712 13122 5474 5288 5720 4419 2507
2206 1930 1416 1134 1722 1419 1402 1256 1101 462 1
196356 302271 356974 134035 67969 31285 22407 17074 9348 7840 4568 4260 3347
1930 2420 1219 1447 1729 2315 1591 1700 722 514 1
228426 594260 329747 136792 88340 36906 28747 13461 7306 5925 5900 4916 3567
3458 1466 1539 2978 1273 1820 1665 2266 1312 506 NA
100101 373820 289194 134129 68675 27079 16309 10103 8029 3612 3309 2131 1231 951
1472 974 766 1633 1701 852 1540 826 NA NA
198862 592969 362630 120056 86409 34295 22879 13266 12247 7250 3417 3219 3364
2132 1738 2086 2660 1272 1698 1070 1626 NA NA NA
215671 612693 371002 151341 106655 35164 29800 20737 9251 7343 7862 3668 6188
2986 4039 1393 1690 1486 2559 2054 NA NA NA NA
341805 597439 432143 132394 87287 21964 21739 15472 10048 5948 4924 3331 4865
4121 1926 3157 1786 1752 1646 NA NA NA NA NA
278182 555949 315948 106309 100308 38574 16846 13338 8172 3935 2149 2221 4401
3005 1969 1637 1114 2098 NA NA NA NA NA NA
230516 545786 365907 194185 76801 27033 22522 17044 9319 9003 2145 4118 2781
1820 2274 2126 1284 NA NA NA NA NA NA NA
270993 673294 422807 148456 113850 54072 35030 9040 4881 5787 2327 4095 5820
3282 4073 1070 NA NA NA NA NA NA NA NA
135121 530928 443876 185537 107310 67478 23006 19517 10309 6748 8246 3882 6407
4436 2565 NA NA NA NA NA NA NA NA NA
218966 372683 328286 141021 60228 38279 17019 12788 8036 8093 1857 4083 925 3524
NA NA NA NA NA NA NA NA NA NA

153083 447527 287253 155208 99823 20573 23014 14057 11794 5374 4980 6170 6518 NA NA NA NA NA NA NA NA NA NA NA
297650 504690 344910 166929 56767 41820 16372 12411 11753 3186 10221 8382 NA NA NA NA NA NA NA NA NA NA NA NA
215051 448699 240914 115089 65581 23498 14580 7690 3051 6111 5020 NA NA NA NA NA NA NA NA NA NA NA NA NA
227785 669756 357519 134850 55605 28379 14773 21563 8014 8212 NA NA NA NA NA NA NA NA NA NA NA NA NA NA
263284 473299 297653 194968 86284 38160 16340 10516 6255 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
296694 572351 341244 129087 67680 25048 15222 15391 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
143172 577736 212259 92381 68885 36820 18931 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA 245584 476931 279505 111174 75177 36555 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
232485 356735 373007 120249 61174 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
268188 485038 298287 131333 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA 314671 644861 289566 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
266533 548646 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
256759 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

Case 1, log-normal model a selected results

Gelman-Rubin Convergence Statistic Values:

mu[1]

	80% interval				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.1007	0.1008	0.7852	0.7856	0.9996
4101	0.1172	0.1071	0.9138	0.8351	1.094
4151	0.112	0.1063	0.8725	0.8286	1.053
4201	0.1206	0.1171	0.9397	0.9126	1.03
4251	0.1202	0.121	0.9366	0.9429	0.9933
4301	0.1191	0.1149	0.9282	0.8955	1.036
4351	0.1194	0.1184	0.9304	0.9226	1.008
4401	0.1242	0.1235	0.9678	0.9623	1.006
4451	0.1251	0.1269	0.9752	0.9889	0.9861
4501	0.1264	0.1253	0.9853	0.9763	1.009
4551	0.1256	0.1242	0.9788	0.968	1.011
4601	0.1264	0.1261	0.9853	0.9826	1.003
4651	0.1257	0.1262	0.98	0.9835	0.9965
4701	0.1251	0.1259	0.9747	0.9812	0.9934
4751	0.1243	0.1249	0.9688	0.9733	0.9954
4801	0.1255	0.1244	0.9778	0.9694	1.009
4851	0.1251	0.1255	0.9748	0.9779	0.9968
4901	0.1272	0.127	0.9916	0.9899	1.002
4951	0.1277	0.1272	0.9951	0.9916	1.004
5001	0.1283	0.1282	1.0	0.999	1.001

mu[24]

	80% interval				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.2636	0.2389	0.946	0.8574	1.103
4101	0.2572	0.2617	0.923	0.9391	0.9829

4151	0.2786	0.2624	1.0	0.9416	1.062
4201	0.2719	0.2669	0.9757	0.9577	1.019
4251	0.2667	0.2622	0.9573	0.9409	1.017
4301	0.2666	0.2648	0.9567	0.9501	1.007
4351	0.2669	0.2687	0.9579	0.9641	0.9936
4401	0.2685	0.2714	0.9635	0.9739	0.9893
4451	0.2644	0.2639	0.9488	0.9469	1.002
4501	0.2685	0.2649	0.9635	0.9508	1.013
4551	0.2689	0.271	0.965	0.9724	0.9923
4601	0.273	0.2786	0.9799	0.9998	0.98
4651	0.2718	0.2756	0.9752	0.9891	0.986
4701	0.2752	0.2744	0.9878	0.9846	1.003
4751	0.271	0.2761	0.9727	0.9908	0.9817
4801	0.2671	0.2712	0.9585	0.9732	0.9849
4851	0.2743	0.2707	0.9845	0.9715	1.013
4901	0.2731	0.2724	0.9802	0.9776	1.003
4951	0.2737	0.2709	0.9821	0.9721	1.01
5001	0.2715	0.2703	0.9743	0.9701	1.004

tau

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	1.172	1.132	0.9435	0.9115	1.035
4101	1.236	1.228	0.9949	0.9881	1.007
4151	1.24	1.242	0.9981	1.0	0.9981
4201	1.22	1.215	0.9817	0.9781	1.004
4251	1.186	1.181	0.9543	0.9504	1.004
4301	1.196	1.191	0.9624	0.959	1.004
4351	1.178	1.168	0.9479	0.9398	1.009
4401	1.178	1.168	0.9485	0.9405	1.009
4451	1.17	1.156	0.9414	0.9301	1.012
4501	1.178	1.168	0.9482	0.9401	1.009
4551	1.188	1.183	0.9564	0.9525	1.004
4601	1.182	1.186	0.9518	0.9545	0.9972
4651	1.177	1.179	0.9475	0.9492	0.9982
4701	1.166	1.17	0.9389	0.9415	0.9971
4751	1.174	1.174	0.9446	0.9451	0.9994
4801	1.171	1.176	0.9426	0.9465	0.9958
4851	1.166	1.168	0.9388	0.9405	0.9981
4901	1.162	1.163	0.9357	0.9362	0.9994
4951	1.162	1.161	0.9352	0.9348	1.0
5001	1.158	1.156	0.9317	0.9307	1.001

z[23]

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.148	0.156	0.7861	0.8286	0.9487
4101	0.1694	0.1702	0.9002	0.9043	0.9954
4151	0.162	0.1636	0.8606	0.8694	0.9899
4201	0.1814	0.1783	0.9637	0.9474	1.017
4251	0.1794	0.1854	0.9531	0.9849	0.9678
4301	0.1761	0.1829	0.9356	0.9716	0.963
4351	0.1779	0.1858	0.9452	0.9873	0.9573
4401	0.1779	0.1864	0.9452	0.9901	0.9547
4451	0.1777	0.1857	0.9443	0.9867	0.957
4501	0.1789	0.1882	0.9504	1.0	0.9504
4551	0.1793	0.1851	0.9528	0.9836	0.9687
4601	0.18	0.1831	0.9565	0.9726	0.9834
4651	0.1796	0.1838	0.9544	0.9766	0.9773
4701	0.1786	0.1822	0.949	0.9681	0.9802
4751	0.1789	0.182	0.9503	0.967	0.9827
4801	0.1773	0.1761	0.9418	0.9354	1.007
4851	0.1789	0.1815	0.9506	0.9645	0.9856

4901	0.1798	0.1824	0.9555	0.9691	0.9859
4951	0.18	0.1834	0.9561	0.9744	0.9813
5001	0.1801	0.1836	0.9569	0.9754	0.9811

paid[48,2]

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	424600.0	443500.0	0.9025	0.9427	0.9573
4101	464700.0	464400.0	0.9877	0.9872	1.001
4151	463700.0	452300.0	0.9856	0.9613	1.025
4201	464700.0	463900.0	0.9877	0.986	1.002
4251	464200.0	459200.0	0.9866	0.9761	1.011
4301	466800.0	468300.0	0.9923	0.9954	0.9969
4351	462600.0	459300.0	0.9832	0.9763	1.007
4401	461500.0	463300.0	0.9809	0.9848	0.9961
4451	469700.0	466800.0	0.9983	0.9921	1.006
4501	469300.0	4.66E+5	0.9975	0.9905	1.007
4551	464900.0	465400.0	0.9881	0.9892	0.999
4601	462900.0	462900.0	0.9838	0.9839	0.9999
4651	460900.0	461400.0	0.9796	0.9808	0.9988
4701	466800.0	467700.0	0.9921	0.994	0.9981
4751	467800.0	469800.0	0.9943	0.9986	0.9957
4801	468300.0	470500.0	0.9954	1.0	0.9954
4851	466100.0	466700.0	0.9907	0.992	0.9986
4901	461500.0	461600.0	0.981	0.9812	0.9998
4951	460300.0	462400.0	0.9784	0.9828	0.9955
5001	461100.0	462100.0	0.9802	0.9822	0.9979

paid[48,13]

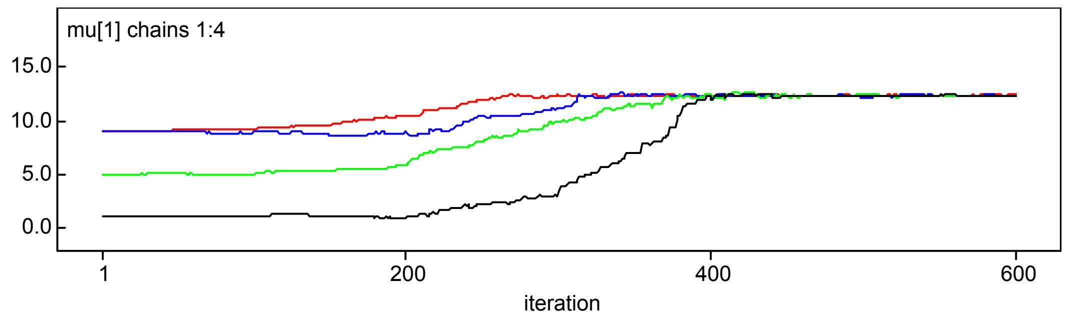
	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	3008.0	3266.0	0.921	1.0	0.921
4101	2894.0	2923.0	0.8861	0.8948	0.9903
4151	2817.0	2842.0	0.8625	0.8701	0.9912
4201	2756.0	2778.0	0.8439	0.8507	0.992
4251	2720.0	2750.0	0.8327	0.8421	0.9888
4301	2757.0	2769.0	0.8441	0.8477	0.9958
4351	2764.0	2775.0	0.8462	0.8496	0.996
4401	2760.0	2748.0	0.845	0.8415	1.004
4451	2817.0	2798.0	0.8625	0.8567	1.007
4501	2815.0	2798.0	0.862	0.8566	1.006
4551	2816.0	2802.0	0.8621	0.858	1.005
4601	2831.0	2816.0	0.8667	0.8622	1.005
4651	2834.0	2828.0	0.8676	0.8658	1.002
4701	2833.0	2838.0	0.8675	0.8688	0.9984
4751	2820.0	2825.0	0.8635	0.8651	0.9982
4801	2839.0	2858.0	0.8692	0.875	0.9934
4851	2833.0	2832.0	0.8675	0.867	1.001
4901	2837.0	2834.0	0.8686	0.8679	1.001
4951	2838.0	2841.0	0.8689	0.87	0.9987
5001	2834.0	2838.0	0.8678	0.869	0.9985

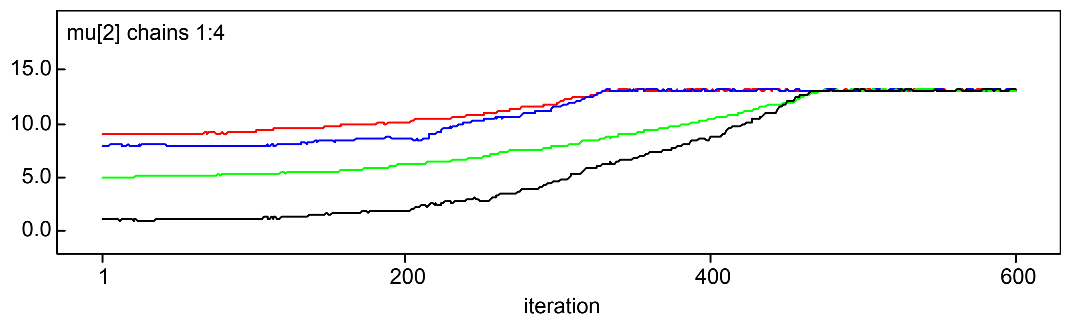
paid[48,24]

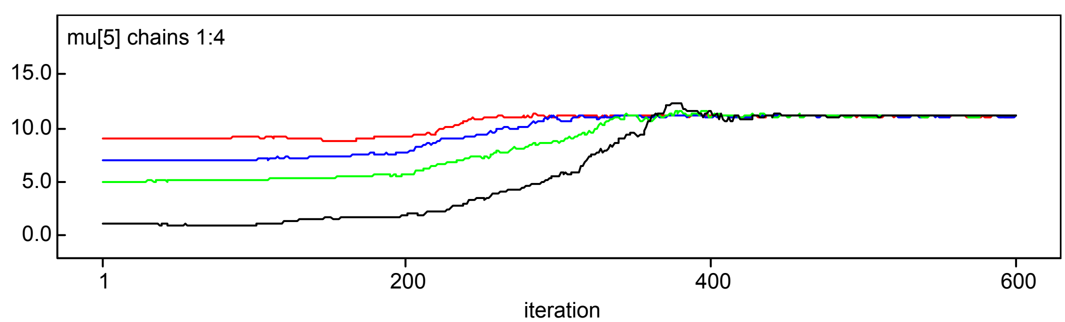
	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	158.1	159.9	0.9761	0.9869	0.9891
4101	156.2	157.7	0.9641	0.9736	0.9903
4151	154.7	154.3	0.9546	0.9524	1.002
4201	151.8	149.9	0.937	0.9251	1.013
4251	154.3	152.9	0.9525	0.9439	1.009

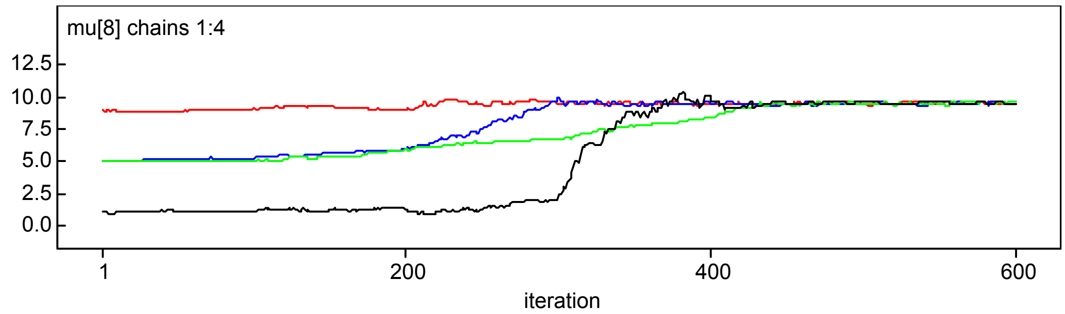
4301	155.5	155.5	0.96	0.9597	1.0
4351	156.6	157.2	0.9667	0.9702	0.9963
4401	158.0	158.1	0.9752	0.9761	0.9991
4451	158.0	158.4	0.9752	0.9778	0.9973
4501	158.2	159.4	0.9767	0.984	0.9925
4551	159.1	159.4	0.9823	0.984	0.9983
4601	159.8	160.6	0.9863	0.991	0.9953
4651	161.1	161.1	0.9945	0.9945	1.0
4701	161.6	162.0	0.9975	1.0	0.9975
4751	161.5	161.5	0.9967	0.9971	0.9996
4801	161.5	161.4	0.9967	0.9965	1.0
4851	161.1	161.0	0.9945	0.9936	1.001
4901	160.2	160.6	0.9891	0.9914	0.9977
4951	159.4	160.1	0.9836	0.9883	0.9952
5001	159.1	159.9	0.9823	0.9868	0.9954

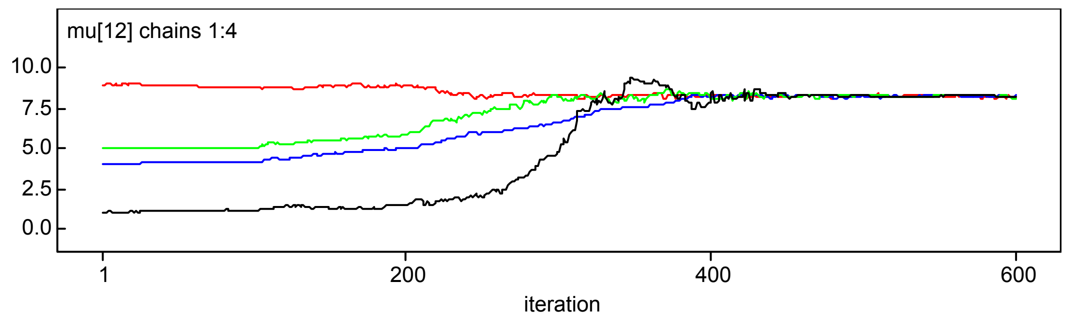
The following selected graphs for iterations 0 to 600 illustrate the convergence of the various parameters.

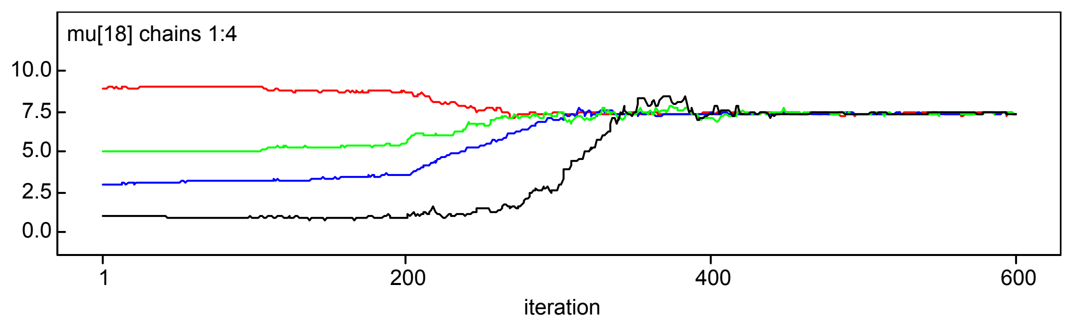


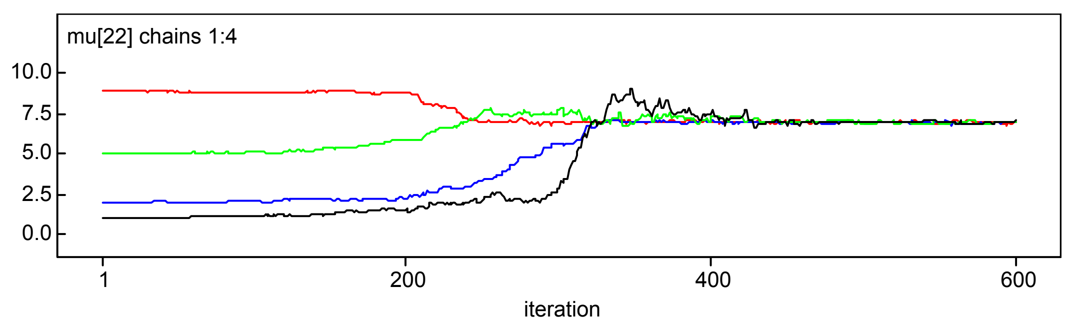


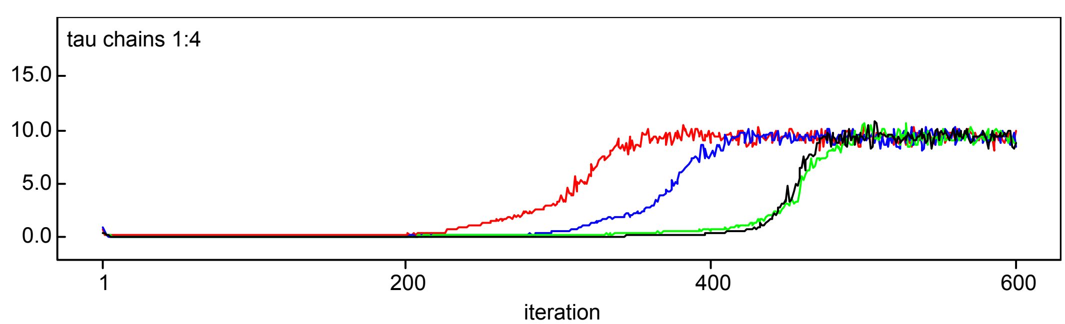


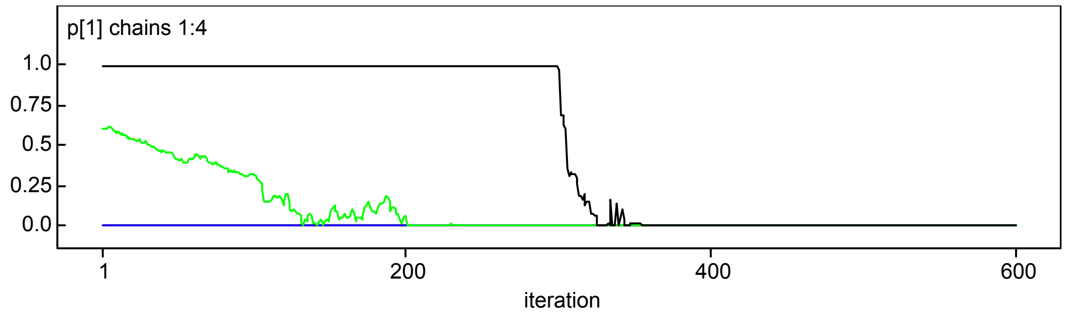

mu[8] chains 1:4
12.5
10.0
7.5
5.0
2.5
0.0
1
200
400
600
iteration

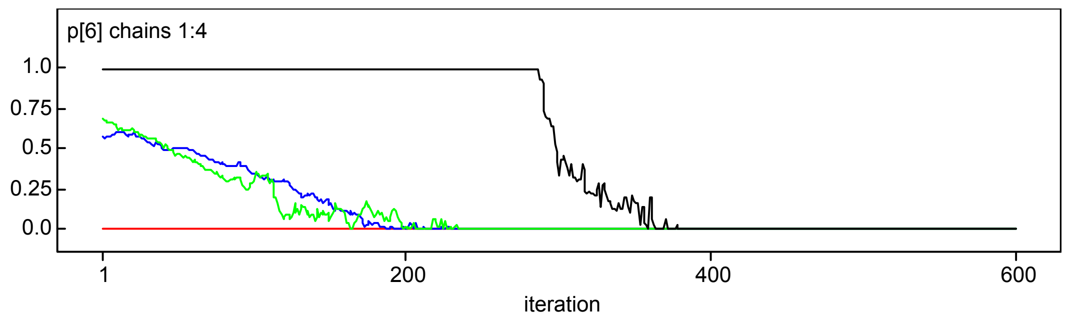

mu[12] chains 1:4
10.0
7.5
5.0
2.5
0.0
1
200
400
600
iteration

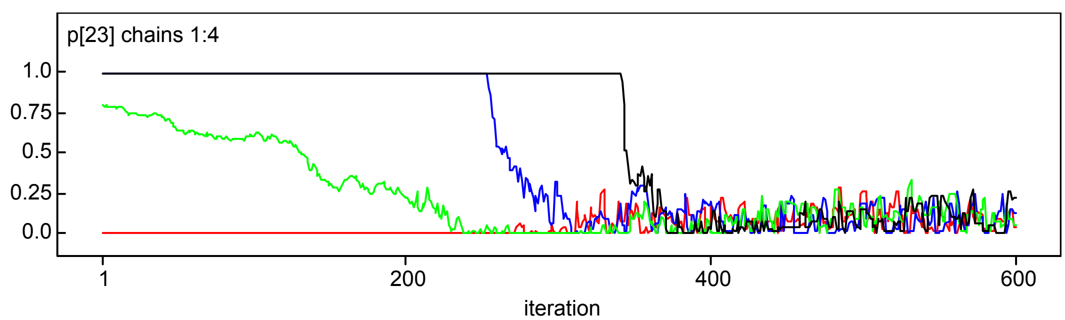

mu[18] chains 1:4
10.0
7.5
5.0
2.5
0.0
1
200
400
600
iteration

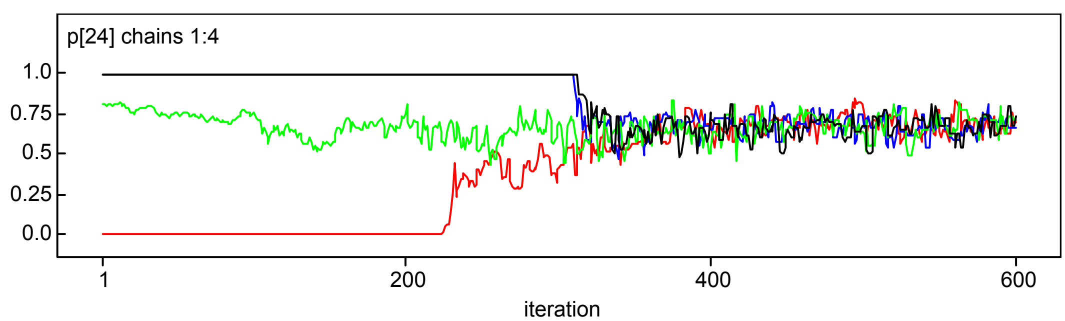

mu[22] chains 1:4
10.0
7.5
5.0
2.5
0.0
1
200
400
600
iteration

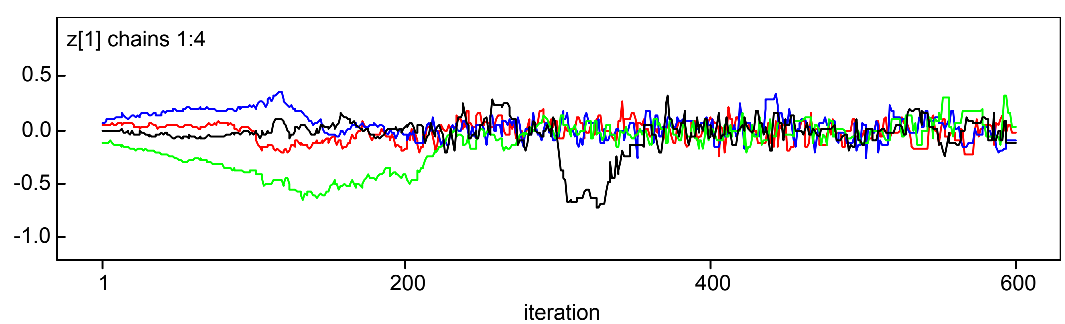

tau chains 1:4
15.0
10.0
5.0
0.0
1
200
400
600
iteration

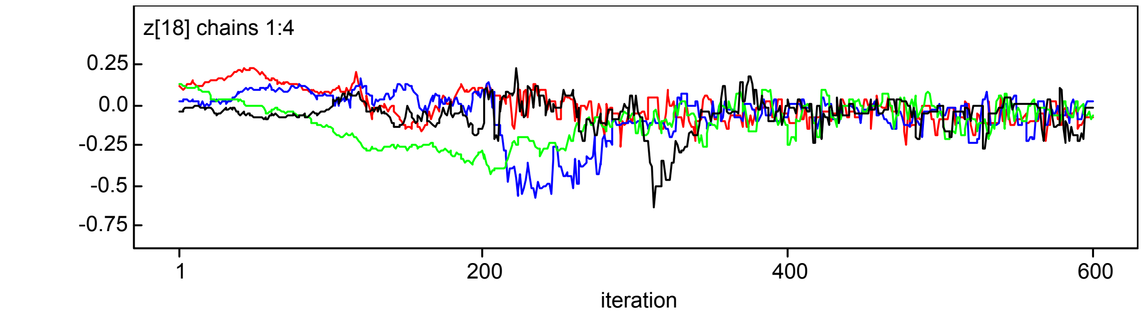

p[1] chains 1:4
1.0
0.75
0.5
0.25
0.0
1
200
400
600
iteration

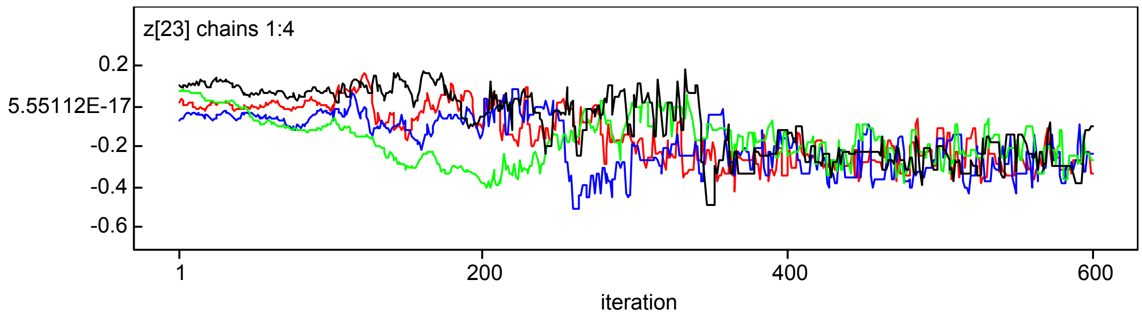

p[6] chains 1:4
1.0
0.75
0.5
0.25
0.0
1
200
400
600
iteration

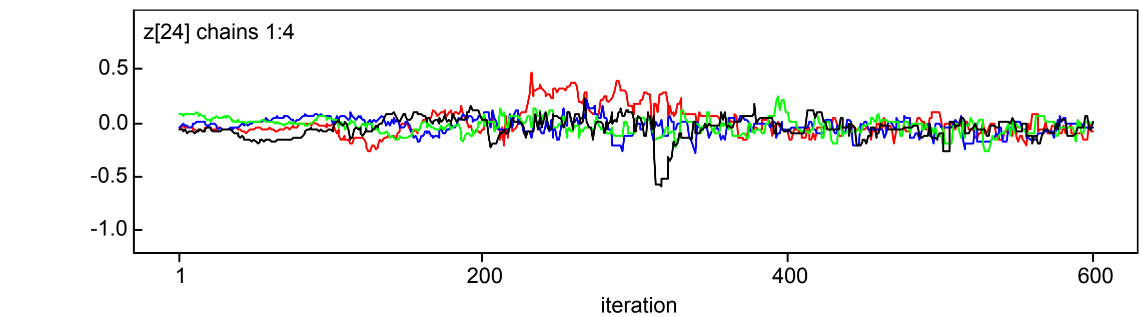

p[23] chains 1:4
1.0
0.75
0.5
0.25
0.0
1
200
400
600
iteration


p[24] chains 1:4
1.0
0.75
0.5
0.25
0.0
1
200
400
600
iteration

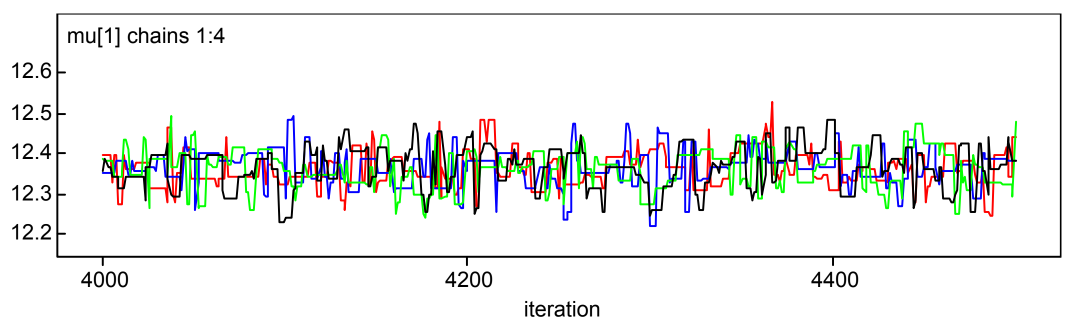

z[1] chains 1:4
0.5
0.0
-0.5
-1.0
1
200
400
600
iteration

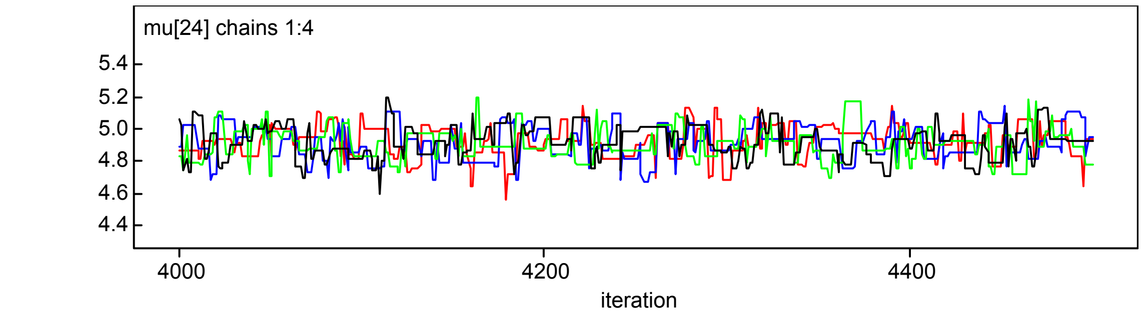


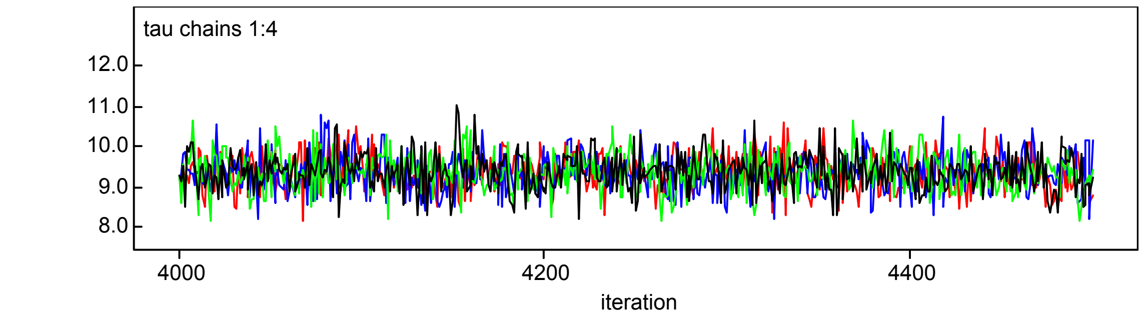


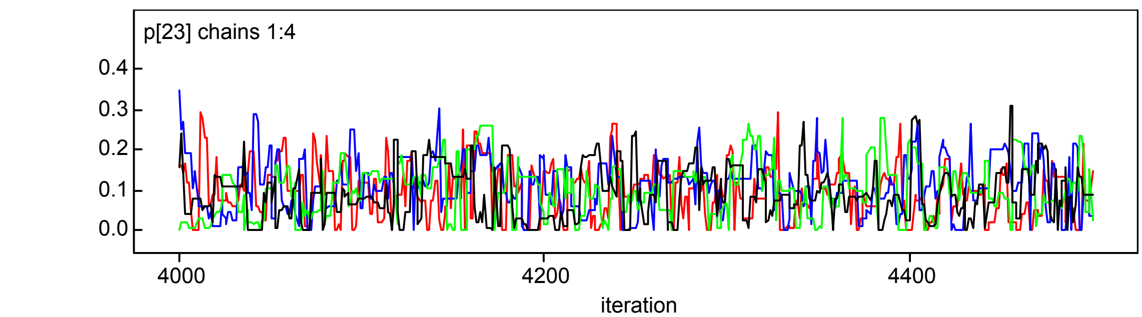


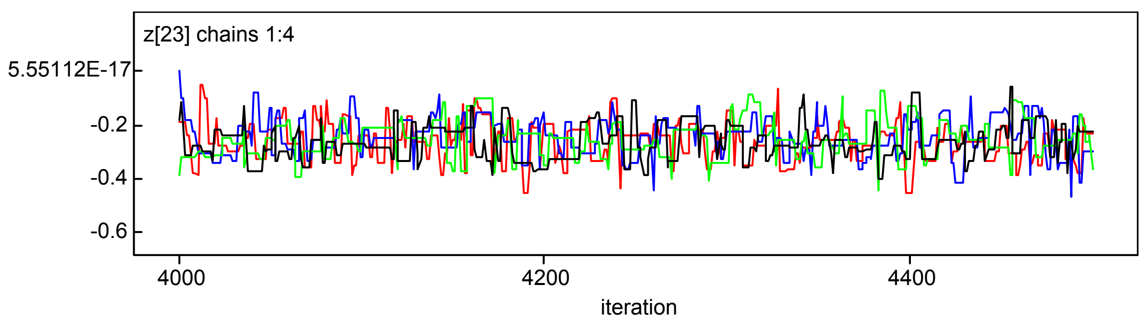
The following are selected graphs for iterations 4000 to 4500.











The following statistics are based on 10,000 iterations.

Total reserve statistics:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	3.004E+6	277100.0	2872.0	2.527E+6	2.981E+6	3.171E+6	3.362E+6	3.5E+6	3.628E+6

Reserve by incurred month:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
reserve[26]	45.98	73.23	0.8225	0.832	1.056	222.1	4001	10000
reserve[27]	535.7	244.6	2.77	1.955	535.9	1034.0	4001	10000

reserve[28]	1645.0	477.0	5.418	740.6	1614.0	2695.0	4001	10000
reserve[29]	3442.0	802.5	8.606	2075.0	3371.0	5232.0	4001	10000
reserve[30]	5144.0	994.8	10.32	3422.0	5066.0	7351.0	4001	10000
reserve[31]	6994.0	1182.0	14.04	4899.0	6924.0	9619.0	4001	10000
reserve[32]	8706.0	1332.0	15.14	6375.0	8615.0	11520.0	4001	10000
reserve[33]	10470.0	1455.0	15.1	7829.0	10380.0	13550.0	4001	10000
reserve[34]	12290.0	1603.0	16.75	9385.0	12200.0	15710.0	4001	10000
reserve[35]	14660.0	1782.0	17.26	11400.0	14570.0	18370.0	4001	10000
reserve[36]	17510.0	2041.0	22.18	13870.0	17410.0	21830.0	4001	10000
reserve[37]	20940.0	2337.0	24.9	16760.0	20790.0	25910.0	4001	10000
reserve[38]	25100.0	2731.0	30.12	20110.0	24930.0	30840.0	4001	10000
reserve[39]	30030.0	3205.0	34.04	24360.0	29840.0	36870.0	4001	10000
reserve[40]	36790.0	3919.0	43.22	29870.0	36510.0	45160.0	4001	10000
reserve[41]	45870.0	5054.0	48.55	37040.0	45530.0	56700.0	4001	10000
reserve[42]	60960.0	7190.0	82.97	48790.0	60380.0	76490.0	4001	10000
reserve[43]	82720.0	10380.0	102.6	65140.0	81740.0	105700.0	4001	10000
reserve[44]	122700.0	16980.0	162.7	94490.0	121100.0	162200.0	4001	10000
reserve[45]	200300.0	31320.0	326.9	149100.0	1.97E+5	272200.0	4001	10000
reserve[46]	350200.0	59980.0	632.1	253600.0	343600.0	489200.0	4001	10000
reserve[47]	699700.0	133200.0	1388.0	486500.0	683900.0	1.01E+6	4001	10000
reserve[48]	1.248E+6	228400.0	2063.0	873800.0	1.222E+6	1.771E+6	4001	10000

Total paid claims by incurred month:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
paid.tot[26]	1.503E+6	73.23	0.8225	1.503E+6	1.503E+6	1.503E+6	4001	10000
paid.tot[27]	1.049E+6	244.6	2.77	1.048E+6	1.049E+6	1.049E+6	4001	10000
paid.tot[28]	1.477E+6	477.0	5.418	1.476E+6	1.477E+6	1.478E+6	4001	10000
paid.tot[29]	1.597E+6	802.5	8.606	1.596E+6	1.597E+6	1.599E+6	4001	10000
paid.tot[30]	1.699E+6	994.8	10.32	1.697E+6	1.699E+6	1.701E+6	4001	10000
paid.tot[31]	1.463E+6	1182.0	14.04	1.461E+6	1.463E+6	1.466E+6	4001	10000
paid.tot[32]	1.523E+6	1332.0	15.14	1.521E+6	1.523E+6	1.526E+6	4001	10000
paid.tot[33]	1.769E+6	1455.0	15.1	1.767E+6	1.769E+6	1.772E+6	4001	10000
paid.tot[34]	1.568E+6	1603.0	16.75	1.565E+6	1.568E+6	1.571E+6	4001	10000
paid.tot[35]	1.23E+6	1782.0	17.26	1.227E+6	1.23E+6	1.234E+6	4001	10000
paid.tot[36]	1.253E+6	2041.0	22.18	1.249E+6	1.253E+6	1.257E+6	4001	10000
paid.tot[37]	1.496E+6	2337.0	24.9	1.492E+6	1.496E+6	1.501E+6	4001	10000
paid.tot[38]	1.17E+6	2731.0	30.12	1.165E+6	1.17E+6	1.176E+6	4001	10000
paid.tot[39]	1.556E+6	3205.0	34.04	1.551E+6	1.556E+6	1.563E+6	4001	10000
paid.tot[40]	1.424E+6	3919.0	43.22	1.417E+6	1.423E+6	1.432E+6	4001	10000
paid.tot[41]	1.509E+6	5054.0	48.55	1.5E+6	1.508E+6	1.519E+6	4001	10000
paid.tot[42]	1.211E+6	7190.0	82.97	1.199E+6	1.211E+6	1.227E+6	4001	10000
paid.tot[43]	1.308E+6	10380.0	102.6	1.29E+6	1.307E+6	1.331E+6	4001	10000
paid.tot[44]	1.266E+6	16980.0	162.7	1.238E+6	1.265E+6	1.306E+6	4001	10000
paid.tot[45]	1.383E+6	31320.0	326.9	1.332E+6	1.38E+6	1.455E+6	4001	10000
paid.tot[46]	1.599E+6	59980.0	632.1	1.503E+6	1.593E+6	1.738E+6	4001	10000
paid.tot[47]	1.515E+6	133200.0	1388.0	1.302E+6	1.499E+6	1.825E+6	4001	10000
paid.tot[48]	1.504E+6	228400.0	2063.0	1.131E+6	1.479E+6	2.028E+6	4001	10000

Selected paid claims:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
paid[48,2]	546600.0	186800.0	1740.0	2.7E+5	517300.0	9.99E+5	4001	10000
paid[48,3]	349600.0	119400.0	1120.0	172700.0	330800.0	6.31E+5	4001	10000
paid[48,4]	151500.0	51870.0	504.8	75330.0	142800.0	276300.0	4001	10000
paid[48,5]	77110.0	26300.0	263.3	38410.0	7.3E+4	139600.0	4001	10000
paid[48,6]	39920.0	13520.0	139.8	19670.0	37870.0	71570.0	4001	10000
paid[48,7]	21900.0	7414.0	80.38	10900.0	20670.0	39540.0	4001	10000
paid[48,8]	15040.0	5059.0	57.38	7426.0	14240.0	26970.0	4001	10000
paid[48,9]	9195.0	3142.0	29.62	4506.0	8722.0	16630.0	4001	10000
paid[48,10]	6723.0	2292.0	24.32	3340.0	6372.0	12190.0	4001	10000
paid[48,11]	4938.0	1682.0	17.83	2458.0	4660.0	8947.0	4001	10000
paid[48,12]	4130.0	1413.0	13.98	2037.0	3908.0	7523.0	4001	10000
paid[48,13]	3473.0	1178.0	13.32	1710.0	3295.0	6285.0	4001	10000
paid[48,14]	2799.0	959.9	10.79	1378.0	2653.0	5052.0	4001	10000
paid[48,15]	2412.0	839.6	8.211	1184.0	2265.0	4415.0	4001	10000
paid[48,16]	1803.0	621.2	6.313	869.9	1715.0	3243.0	4001	10000
paid[48,17]	1721.0	606.1	6.255	827.3	1631.0	3169.0	4001	10000

paid[48,18]	1724.0	602.2	6.27	839.2	1636.0	3127.0	4001	10000
paid[48,19]	1847.0	643.1	6.33	894.6	1743.0	3352.0	4001	10000
paid[48,20]	1709.0	602.3	6.98	806.9	1624.0	3154.0	4001	10000
paid[48,21]	1790.0	625.1	6.22	853.2	1704.0	3263.0	4001	10000
paid[48,22]	1112.0	410.6	3.928	479.4	1070.0	2027.0	4001	10000
paid[48,23]	484.6	232.5	2.662	0.9384	483.7	954.5	4001	10000
paid[48,24]	48.04	74.43	0.7952	0.8326	1.06	229.3	4001	10000

mu[j]:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[1]	12.37	0.04685	0.001151	12.27	12.37	12.46	4001	10000
mu[2]	13.16	0.04782	9.869E-4	13.06	13.16	13.25	4001	10000
mu[3]	12.71	0.04899	0.001165	12.61	12.71	12.8	4001	10000
mu[4]	11.87	0.04813	0.00105	11.78	11.87	11.97	4001	10000
mu[5]	11.2	0.04739	9.66E-4	11.1	11.2	11.29	4001	10000
mu[6]	10.54	0.04949	9.752E-4	10.45	10.54	10.64	4001	10000
mu[7]	9.937	0.05135	0.001235	9.837	9.936	10.04	4001	10000
mu[8]	9.562	0.05081	0.001089	9.464	9.563	9.662	4001	10000
mu[9]	9.068	0.05076	0.001179	8.966	9.068	9.164	4001	10000
mu[10]	8.759	0.05325	0.001171	8.655	8.76	8.861	4001	10000
mu[11]	8.448	0.05291	0.001112	8.347	8.45	8.552	4001	10000
mu[12]	8.272	0.05313	0.00123	8.166	8.271	8.376	4001	10000
mu[13]	8.098	0.0542	0.001263	7.99	8.101	8.202	4001	10000
mu[14]	7.882	0.05407	0.001297	7.771	7.884	7.982	4001	10000
mu[15]	7.738	0.05543	0.001364	7.627	7.738	7.852	4001	10000
mu[16]	7.453	0.05627	0.001236	7.341	7.453	7.562	4001	10000
mu[17]	7.403	0.05802	0.001319	7.294	7.403	7.515	4001	10000
mu[18]	7.403	0.05879	0.001157	7.285	7.404	7.519	4001	10000
mu[19]	7.473	0.05892	0.001298	7.36	7.472	7.587	4001	10000
mu[20]	7.399	0.0612	0.001444	7.282	7.396	7.518	4001	10000
mu[21]	7.436	0.06067	0.001431	7.319	7.437	7.557	4001	10000
mu[22]	6.983	0.06191	0.001242	6.864	6.984	7.105	4001	10000
mu[23]	6.223	0.0616	0.001473	6.099	6.224	6.343	4001	10000
mu[24]	4.923	0.1076	0.002576	4.718	4.923	5.141	4001	10000

tau:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
tau	9.372	0.4645	0.004576	8.496	9.364	10.32	4001	10000

The percentage of ‘0’ cells, p[j]:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p[1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[3]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[4]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[5]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[6]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[7]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[8]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[9]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[10]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[11]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[12]	1.892E-5	5.352E-4	8.136E-6	0.0	0.0	0.0	4001	10000
p[13]	2.36E-4	0.002916	6.427E-5	0.0	0.0	0.0	4001	10000
p[14]	5.138E-4	0.004278	7.819E-5	0.0	0.0	0.00249	4001	10000
p[15]	0.001058	0.006208	9.411E-5	0.0	0.0	0.01621	4001	10000
p[16]	0.003601	0.01209	2.519E-4	0.0	0.0	0.04361	4001	10000
p[17]	0.004771	0.01398	2.529E-4	0.0	0.0	0.05245	4001	10000
p[18]	0.005033	0.01494	2.99E-4	0.0	0.0	0.05374	4001	10000
p[19]	0.003915	0.01346	2.587E-4	0.0	0.0	0.04557	4001	10000
p[20]	0.005322	0.0154	3.334E-4	0.0	0.0	0.05577	4001	10000
p[21]	0.005462	0.01657	3.664E-4	0.0	0.0	0.05844	4001	10000
p[22]	0.01861	0.03125	6.517E-4	0.0	0.0	0.1079	4001	10000
p[23]	0.09223	0.07063	0.001394	0.0	0.084	0.2503	4001	10000
p[24]	0.6821	0.06733	0.001391	0.5459	0.6819	0.8082	4001	10000

z[j]:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
z[1]	-0.003956	0.1023	0.002017	-0.2066	-0.002981	0.1962	4001	10000
z[2]	0.001356	0.09625	0.002151	-0.1904	-5.813E-5	0.1869	4001	10000
z[3]	0.001842	0.1013	0.002186	-0.2019	0.004743	0.2028	4001	10000
z[4]	0.001022	0.1028	0.002249	-0.1949	0.003037	0.2002	4001	10000
z[5]	-0.002246	0.1001	0.00214	-0.1946	-0.005418	0.2013	4001	10000
z[6]	-0.001747	0.09732	0.002311	-0.1892	-0.002266	0.1929	4001	10000
z[7]	-0.001587	0.1003	0.002262	-0.1919	-0.003512	0.1989	4001	10000
z[8]	-0.001945	0.09917	0.002354	-0.1879	-0.005965	0.1953	4001	10000
z[9]	-5.898E-4	0.1012	0.002317	-0.1981	3.216E-4	0.21	4001	10000
z[10]	0.001699	0.1	0.002322	-0.191	5.971E-4	0.2035	4001	10000
z[11]	3.221E-4	0.09848	0.002139	-0.1984	-0.001058	0.1936	4001	10000
z[12]	-0.003668	0.09847	0.002129	-0.2026	-0.002262	0.1927	4001	10000
z[13]	-0.006286	0.09784	0.002683	-0.2041	-0.004393	0.1858	4001	10000
z[14]	-0.01025	0.09324	0.001996	-0.2034	-0.00766	0.1476	4001	10000
z[15]	-0.0193	0.08386	0.001853	-0.2004	-0.01184	0.1182	4001	10000
z[16]	-0.05322	0.07047	0.001906	-0.2114	-0.04476	0.06239	4001	10000
z[17]	-0.05862	0.068	0.001496	-0.2107	-0.05141	0.05404	4001	10000
z[18]	-0.05902	0.06997	0.001615	-0.2216	-0.05001	0.05643	4001	10000
z[19]	-0.04976	0.0738	0.001692	-0.2161	-0.03984	0.07134	4001	10000
z[20]	-0.05622	0.06757	0.001759	-0.2108	-0.04408	0.05795	4001	10000
z[21]	-0.05242	0.07356	0.00167	-0.2171	-0.04264	0.07069	4001	10000
z[22]	-0.134	0.06135	0.001555	-0.2656	-0.1309	-0.01938	4001	10000
z[23]	-0.2631	0.07507	0.00157	-0.3927	-0.269	-0.1037	4001	10000
z[24]	-0.06112	0.06792	0.00164	-0.2005	-0.06138	0.06921	4001	10000

Case 1, log-normal model b selected results

The following statistics are based on 10,000 iterations.

Total reserve statistics:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.944E+6	153100	1709.0	2.659E+6	2.937E+6	3.042E+6	3.142E+6	3.205E+6	3.271E+6

Reserve by incurred month:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
reserve[26]	44.18	63.14	0.631	0.9773	1.001	153.9	4001	10000
reserve[27]	548.7	287.6	3.465	1.988	532.5	1182.0	4001	10000
reserve[28]	1694.0	585.2	5.541	700.4	1629.0	3016.0	4001	10000
reserve[29]	3500.0	887.5	9.635	1985.0	3405.0	5494.0	4001	10000
reserve[30]	5199.0	1073.0	11.66	3350.0	5108.0	7537.0	4001	10000
reserve[31]	7101.0	1356.0	15.38	4779.0	6969.0	10120.0	4001	10000
reserve[32]	8854.0	1502.0	16.56	6202.0	8732.0	12160.0	4001	10000
reserve[33]	10670.0	1698.0	16.61	7694.0	10550.0	14290.0	4001	10000
reserve[34]	12520.0	1914.0	19.85	9161.0	12390.0	16730.0	4001	10000
reserve[35]	15090.0	2286.0	20.17	11230.0	14910.0	20100.0	4001	10000
reserve[36]	17950.0	2551.0	27.07	13520.0	17790.0	23560.0	4001	10000
reserve[37]	21650.0	3240.0	33.4	16240.0	21350.0	28980.0	4001	10000
reserve[38]	25870.0	3628.0	36.23	19740.0	25540.0	34040.0	4001	10000
reserve[39]	30990.0	4295.0	44.22	23820.0	30580.0	40550.0	4001	10000
reserve[40]	37530.0	4693.0	47.92	29260.0	37190.0	47600.0	4001	10000
reserve[41]	46920.0	5988.0	64.66	36630.0	46400.0	60090.0	4001	10000
reserve[42]	61860.0	7749.0	79.89	48410.0	61270.0	79440.0	4001	10000
reserve[43]	83560.0	10010.0	99.23	66280.0	82840.0	105200.0	4001	10000
reserve[44]	123300.0	15950.0	167.1	96060.0	121700.0	158800.0	4001	10000
reserve[45]	198500.0	25350.0	257.6	1.55E+5	196200.0	253600.0	4001	10000
reserve[46]	3.45E+5	41400.0	392.1	272500.0	341800.0	434200.0	4001	10000
reserve[47]	680100.0	72400.0	773.8	550800.0	675400.0	836500.0	4001	10000
reserve[48]	1.206E+6	119600.0	1170.0	987300.0	1.2E+6	1.463E+6	4001	10000

Total paid claims by incurred month:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
paid.tot[26]	1.503E+6	63.14	0.631	1.503E+6	1.503E+6	1.503E+6	4001	10000
paid.tot[27]	1.049E+6	287.6	3.465	1.048E+6	1.049E+6	1.05E+6	4001	10000

paid.tot[28]	1.477E+6	585.2	5.541	1.476E+6	1.477E+6	1.478E+6	4001	10000
paid.tot[29]	1.597E+6	887.5	9.635	1.596E+6	1.597E+6	1.599E+6	4001	10000
paid.tot[30]	1.699E+6	1073.0	11.66	1.697E+6	1.699E+6	1.701E+6	4001	10000
paid.tot[31]	1.463E+6	1356.0	15.38	1.461E+6	1.463E+6	1.466E+6	4001	10000
paid.tot[32]	1.524E+6	1502.0	16.56	1.521E+6	1.523E+6	1.527E+6	4001	10000
paid.tot[33]	1.77E+6	1698.0	16.61	1.767E+6	1.769E+6	1.773E+6	4001	10000
paid.tot[34]	1.568E+6	1914.0	19.85	1.565E+6	1.568E+6	1.572E+6	4001	10000
paid.tot[35]	1.231E+6	2286.0	20.17	1.227E+6	1.231E+6	1.236E+6	4001	10000
paid.tot[36]	1.253E+6	2551.0	27.07	1.249E+6	1.253E+6	1.259E+6	4001	10000
paid.tot[37]	1.497E+6	3240.0	33.4	1.491E+6	1.496E+6	1.504E+6	4001	10000
paid.tot[38]	1.171E+6	3628.0	36.23	1.165E+6	1.171E+6	1.179E+6	4001	10000
paid.tot[39]	1.557E+6	4295.0	44.22	1.55E+6	1.557E+6	1.567E+6	4001	10000
paid.tot[40]	1.424E+6	4693.0	47.92	1.416E+6	1.424E+6	1.434E+6	4001	10000
paid.tot[41]	1.51E+6	5988.0	64.66	1.499E+6	1.509E+6	1.523E+6	4001	10000
paid.tot[42]	1.212E+6	7749.0	79.89	1.199E+6	1.211E+6	1.23E+6	4001	10000
paid.tot[43]	1.308E+6	10010.0	99.23	1.291E+6	1.308E+6	1.33E+6	4001	10000
paid.tot[44]	1.267E+6	15950.0	167.1	1.24E+6	1.265E+6	1.302E+6	4001	10000
paid.tot[45]	1.381E+6	25350.0	257.6	1.338E+6	1.379E+6	1.436E+6	4001	10000
paid.tot[46]	1.594E+6	41400.0	392.1	1.522E+6	1.591E+6	1.683E+6	4001	10000
paid.tot[47]	1.495E+6	72400.0	773.8	1.366E+6	1.491E+6	1.652E+6	4001	10000
paid.tot[48]	1.463E+6	119600.0	1170.0	1.244E+6	1.456E+6	1.72E+6	4001	10000

Selected paid claims:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
paid[48,2]	525200.0	94660.0	956.8	361300.0	517400.0	734900.0	4001	10000
paid[48,3]	335700.0	60940.0	574.7	231700.0	330100.0	4.72E+5	4001	10000
paid[48,4]	146300.0	32820.0	319.0	91900.0	142800.0	220600.0	4001	10000
paid[48,5]	75200.0	19950.0	201.4	43920.0	72530.0	1.22E+5	4001	10000
paid[48,6]	39730.0	12270.0	135.6	21220.0	37970.0	69270.0	4001	10000
paid[48,7]	21650.0	6299.0	63.43	11880.0	20810.0	36430.0	4001	10000
paid[48,8]	15060.0	5094.0	51.1	7460.0	14300.0	27420.0	4001	10000
paid[48,9]	9349.0	3652.0	36.76	4208.0	8688.0	18360.0	4001	10000
paid[48,10]	6646.0	2013.0	21.95	3544.0	6356.0	11430.0	4001	10000
paid[48,11]	5101.0	2289.0	23.67	2038.0	4647.0	10680.0	4001	10000
paid[48,12]	4205.0	1703.0	18.49	1865.0	3898.0	8377.0	4001	10000
paid[48,13]	3713.0	1904.0	22.75	1273.0	3332.0	8482.0	4001	10000
paid[48,14]	2869.0	1220.0	14.26	1196.0	2657.0	5850.0	4001	10000
paid[48,15]	2536.0	1198.0	13.43	951.2	2297.0	5522.0	4001	10000
paid[48,16]	1865.0	810.1	7.386	740.1	1715.0	3866.0	4001	10000
paid[48,17]	1815.0	871.0	9.57	661.8	1651.0	3962.0	4001	10000
paid[48,18]	1734.0	620.1	7.108	821.7	1642.0	3187.0	4001	10000
paid[48,19]	1905.0	806.2	9.503	772.0	1770.0	3830.0	4001	10000
paid[48,20]	1715.0	608.7	6.211	818.8	1625.0	3177.0	4001	10000
paid[48,21]	1799.0	674.9	6.364	809.5	1697.0	3369.0	4001	10000
paid[48,22]	1141.0	499.3	4.897	385.1	1073.0	2312.0	4001	10000
paid[48,23]	504.6	278.4	2.596	0.9884	485.5	1126.0	4001	10000
paid[48,24]	42.81	62.66	0.669	0.9774	1.001	153.3	4001	10000

mu[j]:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[1]	12.37	0.0365	7.975E-4	12.29	12.37	12.44	4001	10000
mu[2]	13.16	0.02606	6.195E-4	13.11	13.16	13.21	4001	10000
mu[3]	12.71	0.02556	5.017E-4	12.66	12.71	12.76	4001	10000
mu[4]	11.87	0.03326	7.174E-4	11.81	11.87	11.94	4001	10000
mu[5]	11.2	0.03974	8.781E-4	11.12	11.2	11.28	4001	10000
mu[6]	10.55	0.0445	0.001021	10.46	10.55	10.63	4001	10000
mu[7]	9.938	0.04277	9.338E-4	9.854	9.938	10.02	4001	10000
mu[8]	9.564	0.05033	0.001155	9.465	9.564	9.663	4001	10000
mu[9]	9.069	0.05799	0.001369	8.955	9.069	9.187	4001	10000
mu[10]	8.758	0.04656	0.001014	8.665	8.76	8.848	4001	10000
mu[11]	8.448	0.06906	0.001496	8.308	8.449	8.579	4001	10000
mu[12]	8.273	0.06313	0.001315	8.151	8.272	8.395	4001	10000
mu[13]	8.097	0.07924	0.001868	7.938	8.097	8.253	4001	10000
mu[14]	7.882	0.06781	0.001547	7.747	7.881	8.017	4001	10000
mu[15]	7.742	0.07384	0.001821	7.597	7.741	7.888	4001	10000
mu[16]	7.455	0.06939	0.001553	7.321	7.454	7.593	4001	10000
mu[17]	7.411	0.07671	0.001857	7.261	7.411	7.563	4001	10000
mu[18]	7.403	0.05892	0.00122	7.286	7.404	7.517	4001	10000
mu[19]	7.477	0.06911	0.001535	7.34	7.479	7.611	4001	10000

mu[20]	7.397	0.05962	0.00124	7.278	7.397	7.52	4001	10000
mu[21]	7.434	0.06588	0.001539	7.301	7.435	7.565	4001	10000
mu[22]	6.99	0.07372	0.001841	6.852	6.988	7.143	4001	10000
mu[23]	6.235	0.07869	0.001577	6.081	6.231	6.396	4001	10000
mu[24]	4.904	0.02952	6.822E-4	4.846	4.904	4.965	4001	10000

tau[]:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
tau[1]	16.71	3.481	0.04274	10.65	16.44	24.19	4001	10000
tau[2]	33.49	6.921	0.07627	21.3	33.06	48.35	4001	10000
tau[3]	33.74	7.205	0.07977	21.25	33.21	49.31	4001	10000
tau[4]	22.22	4.728	0.05364	14.04	21.86	32.46	4001	10000
tau[5]	15.73	3.383	0.04077	9.773	15.5	22.94	4001	10000
tau[6]	12.16	2.648	0.03092	7.455	12.01	17.8	4001	10000
tau[7]	13.47	2.999	0.03243	8.236	13.24	19.77	4001	10000
tau[8]	9.888	2.224	0.02717	6.049	9.718	14.65	4001	10000
tau[9]	7.642	1.704	0.02168	4.669	7.507	11.33	4001	10000
tau[10]	12.45	2.846	0.03196	7.528	12.26	18.6	4001	10000
tau[11]	5.834	1.365	0.01533	3.471	5.731	8.827	4001	10000
tau[12]	7.6	1.812	0.02357	4.459	7.455	11.5	4001	10000
tau[13]	4.659	1.104	0.01242	2.753	4.564	7.037	4001	10000
tau[14]	6.71	1.626	0.01817	3.901	6.561	10.26	4001	10000
tau[15]	5.562	1.355	0.01455	3.275	5.441	8.52	4001	10000
tau[16]	6.583	1.649	0.02007	3.728	6.469	10.16	4001	10000
tau[17]	5.556	1.392	0.01666	3.171	5.446	8.576	4001	10000
tau[18]	9.516	2.47	0.02854	5.375	9.287	14.96	4001	10000
tau[19]	7.005	1.865	0.02053	3.845	6.862	11.1	4001	10000
tau[20]	10.14	2.701	0.03346	5.566	9.918	16.04	4001	10000
tau[21]	8.901	2.439	0.02845	4.805	8.665	14.26	4001	10000
tau[22]	7.266	2.03	0.02345	3.856	7.089	11.8	4001	10000
tau[23]	6.645	1.907	0.02347	3.479	6.454	10.99	4001	10000
tau[24]	164.9	82.15	1.075	45.0	151.6	355.9	4001	10000

The percentage of '0' cells, p[j]:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p[10]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[11]	7.17E-7	6.847E-5	6.844E-7	0.0	0.0	0.0	4001	10000
p[12]	2.778E-5	8.162E-4	1.003E-5	0.0	0.0	0.0	4001	10000
p[13]	1.168E-4	0.00181	2.894E-5	0.0	0.0	0.0	4001	10000
p[14]	6.954E-4	0.00498	9.67E-5	0.0	0.0	0.007052	4001	10000
p[15]	0.00116	0.007521	1.503E-4	0.0	0.0	0.01489	4001	10000
p[16]	0.003947	0.0124	2.298E-4	0.0	0.0	0.04358	4001	10000
p[17]	0.004841	0.01462	3.062E-4	0.0	0.0	0.05277	4001	10000
p[18]	0.004851	0.01399	2.831E-4	0.0	0.0	0.05256	4001	10000
p[19]	0.004232	0.0139	3.015E-4	0.0	0.0	0.04926	4001	10000
p[20]	0.005251	0.0156	2.854E-4	0.0	0.0	0.05637	4001	10000
p[21]	0.004925	0.01504	3.193E-4	0.0	0.0	0.05461	4001	10000
p[22]	0.02065	0.03321	8.791E-4	0.0	0.0	0.1119	4001	10000
p[23]	0.08919	0.06914	0.001413	0.0	0.07996	0.2407	4001	10000
p[24]	0.6841	0.06503	0.001428	0.5544	0.6859	0.8059	4001	10000

z[j]:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
z[1]	-8.752E-4	0.1002	0.002085	-0.1951	-0.001484	0.1919	4001	10000
z[2]	0.001684	0.1002	0.002093	-0.1851	3.38E-4	0.197	4001	10000
z[3]	2.528E-4	0.0992	0.002125	-0.1882	5.787E-4	0.201	4001	10000
z[4]	3.379E-4	0.09888	0.001962	-0.1956	0.00323	0.1915	4001	10000
z[5]	-0.002846	0.09998	0.002057	-0.1937	-0.003541	0.1933	4001	10000
z[6]	-0.00151	0.09987	0.002233	-0.1981	-0.005055	0.2072	4001	10000
z[7]	0.002844	0.1031	0.002456	-0.1993	0.001424	0.2082	4001	10000
z[8]	-0.003905	0.1021	0.002267	-0.206	-0.002319	0.1952	4001	10000
z[9]	-0.001957	0.09784	0.002267	-0.1942	-0.002757	0.1862	4001	10000
z[10]	0.00229	0.1007	0.002243	-0.1929	0.003382	0.205	4001	10000
z[11]	-0.001806	0.09896	0.002086	-0.1922	-9.389E-4	0.1913	4001	10000
z[12]	5.429E-4	0.09635	0.002017	-0.1835	-9.412E-4	0.1922	4001	10000
z[13]	-0.004098	0.09819	0.002418	-0.2006	-0.004381	0.1857	4001	10000
z[14]	-0.009692	0.09033	0.00203	-0.1878	-0.005636	0.1502	4001	10000
z[15]	-0.01933	0.08414	0.001997	-0.1987	-0.01078	0.1261	4001	10000

z[16]	-0.05314	0.07192	0.0016	-0.2096	-0.04371	0.06578	4001	10000
z[17]	-0.05938	0.07067	0.001756	-0.216	-0.05178	0.06105	4001	10000
z[18]	-0.05927	0.07045	0.00175	-0.2182	-0.05079	0.05287	4001	10000
z[19]	-0.04671	0.07295	0.001783	-0.2052	-0.03937	0.07579	4001	10000
z[20]	-0.06163	0.06989	0.001777	-0.2148	-0.05348	0.05425	4001	10000
z[21]	-0.05371	0.07115	0.001782	-0.2131	-0.04623	0.06652	4001	10000
z[22]	-0.1271	0.06111	0.001776	-0.2537	-0.1251	-0.0113	4001	10000
z[23]	-0.263	0.07371	0.001454	-0.3934	-0.2699	-0.1085	4001	10000
z[24]	-0.06473	0.06494	0.001442	-0.1954	-0.06208	0.05768	4001	10000

Case 1, log-normal model c, selected results.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[1]	12.37	0.03602	5.814E-4	12.29	12.37	12.43	4001	20000
mu[2]	13.16	0.02539	3.97E-4	13.11	13.16	13.21	4001	20000
mu[3]	12.71	0.02625	4.28E-4	12.66	12.71	12.76	4001	20000
mu[4]	11.87	0.03291	5.145E-4	11.81	11.87	11.94	4001	20000
mu[5]	11.2	0.03866	5.777E-4	11.12	11.2	11.27	4001	20000
mu[6]	10.55	0.04479	8.313E-4	10.46	10.54	10.63	4001	20000
mu[7]	9.938	0.04233	6.679E-4	9.854	9.939	10.02	4001	20000
mu[8]	9.561	0.05139	7.694E-4	9.463	9.562	9.663	4001	20000
mu[9]	9.07	0.05869	9.398E-4	8.953	9.07	9.184	4001	20000
mu[10]	8.758	0.04687	7.898E-4	8.665	8.758	8.85	4001	20000
mu[11]	8.449	0.06777	8.624E-4	8.31	8.449	8.581	4001	20000
mu[12]	8.273	0.06256	0.001003	8.149	8.273	8.396	4001	20000
mu[13]	8.095	0.07996	0.001234	7.94	8.095	8.252	4001	20000
mu[14]	7.882	0.06712	9.806E-4	7.751	7.883	8.012	4001	20000
mu[15]	7.739	0.07516	0.001116	7.591	7.739	7.884	4001	20000
mu[16]	7.447	0.07002	0.001186	7.309	7.448	7.587	4001	20000
mu[17]	7.395	0.07833	0.001245	7.24	7.393	7.548	4001	20000
mu[18]	7.398	0.05988	9.8E-4	7.278	7.399	7.514	4001	20000
mu[19]	7.468	0.07089	0.001121	7.331	7.469	7.611	4001	20000
mu[20]	7.39	0.06011	9.689E-4	7.275	7.39	7.511	4001	20000
mu[21]	7.432	0.06566	0.001118	7.302	7.43	7.563	4001	20000
mu[22]	6.968	0.07382	0.001367	6.821	6.968	7.111	4001	20000
mu[23]	6.214	0.0775	0.00124	6.067	6.213	6.37	4001	20000
mu[24]	4.898	0.03137	4.888E-4	4.833	4.899	4.957	4001	20000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
tau[1]	16.73	3.472	0.02468	10.63	16.48	24.2	4001	20000
tau[2]	33.55	7.023	0.05112	21.24	33.07	48.71	4001	20000
tau[3]	33.65	7.138	0.05877	21.18	33.13	49.15	4001	20000
tau[4]	22.32	4.779	0.03721	14.04	21.97	32.53	4001	20000
tau[5]	15.77	3.379	0.02883	9.824	15.52	23.04	4001	20000
tau[6]	12.2	2.669	0.0211	7.482	12.01	18.01	4001	20000
tau[7]	13.47	2.958	0.02269	8.308	13.28	19.85	4001	20000
tau[8]	9.869	2.25	0.01737	5.998	9.701	14.86	4001	20000
tau[9]	7.633	1.717	0.01223	4.644	7.51	11.31	4001	20000
tau[10]	12.46	2.844	0.02377	7.491	12.24	18.65	4001	20000
tau[11]	5.848	1.35	0.009562	3.495	5.746	8.788	4001	20000
tau[12]	7.604	1.799	0.01393	4.457	7.466	11.55	4001	20000
tau[13]	4.672	1.108	0.007989	2.749	4.582	7.077	4001	20000
tau[14]	6.705	1.624	0.01284	3.912	6.583	10.2	4001	20000
tau[15]	5.563	1.372	0.01074	3.211	5.447	8.542	4001	20000
tau[16]	6.576	1.655	0.01223	3.76	6.437	10.25	4001	20000
tau[17]	5.558	1.418	0.01179	3.138	5.436	8.642	4001	20000
tau[18]	9.542	2.467	0.01982	5.339	9.365	14.93	4001	20000
tau[19]	6.983	1.836	0.0134	3.868	6.824	10.99	4001	20000
tau[20]	10.17	2.742	0.02305	5.532	9.937	16.25	4001	20000
tau[21]	8.884	2.396	0.02028	4.872	8.682	14.16	4001	20000
tau[22]	7.286	2.016	0.01712	3.878	7.094	11.82	4001	20000
tau[23]	6.701	1.903	0.01614	3.501	6.529	10.86	4001	20000
tau[24]	162.6	82.42	0.7473	42.44	149.3	361.7	4001	20000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
x	2.1	0.1468	0.006714	1.804	2.104	2.386	4001	20000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
y	0.33	0.02505	0.00114	0.2807	0.3301	0.379	4001	20000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
z[1]	-0.001267	0.07286	0.001113	-0.1439	-0.002204	0.1415	4001	20000
z[2]	-0.001541	0.07094	0.001179	-0.1397	-4.41E-4	0.1374	4001	20000
z[3]	-0.002151	0.07045	0.001281	-0.1386	-0.003465	0.1385	4001	20000
z[4]	0.001011	0.07138	0.001199	-0.1419	0.001387	0.1411	4001	20000
z[5]	-3.594E-4	0.07043	0.001112	-0.1375	-1.993E-4	0.1391	4001	20000
z[6]	-8.687E-4	0.0711	0.0011	-0.141	-0.00231	0.1439	4001	20000
z[7]	4.811E-4	0.07194	0.001155	-0.1394	0.001473	0.1448	4001	20000
z[8]	5.923E-4	0.07081	0.001189	-0.1434	-2.995E-5	0.1367	4001	20000
z[9]	-1.29E-4	0.06937	0.00116	-0.1401	9.542E-4	0.1337	4001	20000
z[10]	6.942E-4	0.07064	0.001084	-0.1364	-2.954E-4	0.1408	4001	20000
z[11]	0.001579	0.07042	0.001228	-0.1378	2.873E-4	0.1393	4001	20000
z[12]	0.001047	0.07034	0.001107	-0.1343	0.001042	0.1391	4001	20000
z[13]	6.795E-4	0.07094	0.001033	-0.1391	-3.693E-4	0.1434	4001	20000
z[14]	3.326E-5	0.07081	0.001107	-0.1362	-3.194E-4	0.1424	4001	20000
z[15]	-0.001602	0.07089	0.001022	-0.1395	-0.001289	0.1387	4001	20000
z[16]	5.113E-4	0.07012	0.001152	-0.1367	4.787E-4	0.1406	4001	20000
z[17]	0.001994	0.06997	0.001035	-0.137	0.002035	0.137	4001	20000
z[18]	-9.701E-4	0.07006	0.001178	-0.1389	-2.083E-4	0.1351	4001	20000
z[19]	0.001689	0.07148	0.00124	-0.1363	3.662E-4	0.1445	4001	20000
z[20]	-7.089E-5	0.07086	0.001125	-0.135	-5.57E-4	0.1369	4001	20000
z[21]	4.076E-4	0.07098	0.001227	-0.1351	5.126E-4	0.1393	4001	20000
z[22]	-0.001157	0.06774	0.001097	-0.1339	-0.001072	0.1287	4001	20000
z[23]	-0.06108	0.06253	0.001293	-0.1857	-0.06144	0.06087	4001	20000
z[24]	0.05115	0.06309	0.001136	-0.07168	0.05056	0.1767	4001	20000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p[15]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[16]	2.147E-5	8.21E-4	1.15E-5	0.0	0.0	0.0	4001	20000
p[17]	2.315E-5	8.968E-4	1.283E-5	0.0	0.0	0.0	4001	20000
p[18]	9.181E-6	5.015E-4	4.76E-6	0.0	0.0	0.0	4001	20000
p[19]	9.556E-6	4.191E-4	3.906E-6	0.0	0.0	0.0	4001	20000
p[20]	1.509E-5	6.798E-4	6.262E-6	0.0	0.0	0.0	4001	20000
p[21]	8.885E-6	4.913E-4	5.437E-6	0.0	0.0	0.0	4001	20000
p[22]	4.638E-4	0.004377	4.059E-5	0.0	0.0	0.0	4001	20000
p[23]	0.02107	0.03562	4.553E-4	0.0	0.0	0.1228	4001	20000
p[24]	0.535	0.07427	0.00131	0.3891	0.5353	0.6796	4001	20000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
paid[48,2]	525900.0	94820.0	740.5	362900.0	517500.0	735700.0	4001	20000
paid[48,3]	335700.0	59900.0	415.1	232400.0	331200.0	467900.0	4001	20000
paid[48,4]	146300.0	32450.0	239.3	93260.0	142700.0	220100.0	4001	20000
paid[48,5]	75260.0	19900.0	134.4	43370.0	72850.0	121200.0	4001	20000
paid[48,6]	39740.0	12240.0	91.64	21120.0	37980.0	68260.0	4001	20000
paid[48,7]	21530.0	6191.0	41.36	11950.0	20700.0	35840.0	4001	20000
paid[48,8]	14990.0	5139.0	37.64	7408.0	14230.0	27210.0	4001	20000
paid[48,9]	9317.0	3657.0	26.31	4173.0	8663.0	18260.0	4001	20000
paid[48,10]	6650.0	2025.0	14.8	3517.0	6366.0	11440.0	4001	20000
paid[48,11]	5113.0	2353.0	16.27	1974.0	4645.0	10990.0	4001	20000
paid[48,12]	4230.0	1662.0	12.59	1856.0	3937.0	8247.0	4001	20000
paid[48,13]	3674.0	1936.0	15.7	1266.0	3262.0	8607.0	4001	20000
paid[48,14]	2874.0	1221.0	7.96	1197.0	2650.0	5837.0	4001	20000
paid[48,15]	2520.0	1206.0	9.327	958.2	2277.0	5497.0	4001	20000
paid[48,16]	1862.0	805.8	6.333	768.3	1708.0	3822.0	4001	20000
paid[48,17]	1797.0	843.1	6.594	690.7	1624.0	3923.0	4001	20000
paid[48,18]	1730.0	610.2	4.472	831.8	1632.0	3205.0	4001	20000
paid[48,19]	1902.0	810.0	6.453	802.5	1753.0	3882.0	4001	20000
paid[48,20]	1713.0	590.8	4.454	841.7	1621.0	3132.0	4001	20000
paid[48,21]	1796.0	666.6	5.434	833.6	1686.0	3398.0	4001	20000
paid[48,22]	1145.0	467.8	3.751	494.0	1063.0	2281.0	4001	20000
paid[48,23]	531.1	250.1	1.82	149.0	492.1	1136.0	4001	20000
paid[48,24]	63.16	67.14	0.472	0.9835	1.015	155.3	4001	20000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
paid.tot[26]	1.503E+6	67.33	0.4838	1.503E+6	1.503E+6	1.503E+6	4001	20000

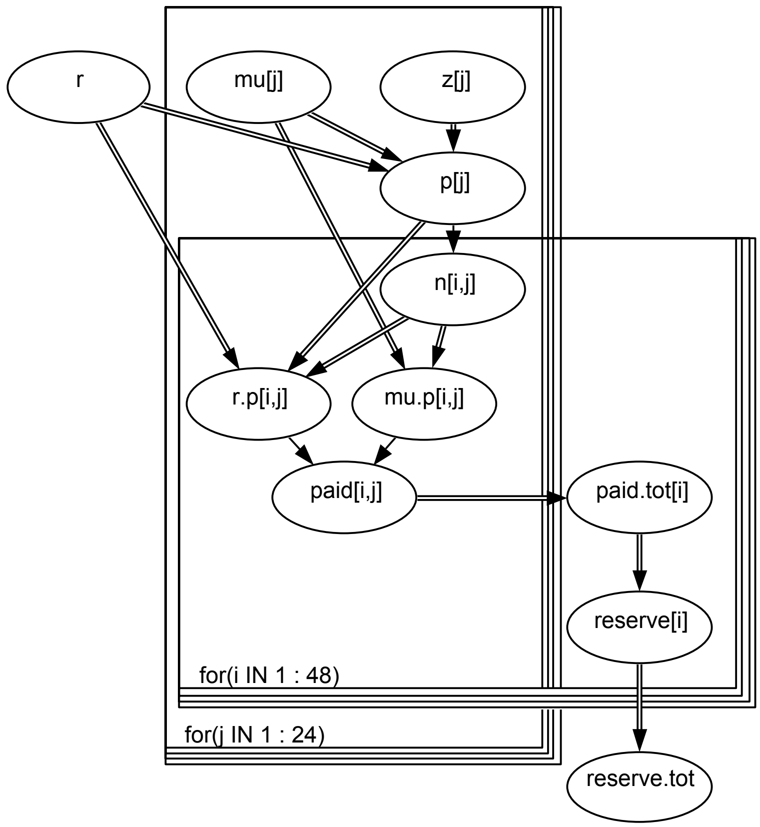
paid.tot[27]	1.049E+6	256.3	1.909	1.049E+6	1.049E+6	1.05E+6	4001	20000
paid.tot[28]	1.477E+6	539.2	4.563	1.476E+6	1.477E+6	1.478E+6	4001	20000
paid.tot[29]	1.597E+6	858.1	7.063	1.596E+6	1.597E+6	1.599E+6	4001	20000
paid.tot[30]	1.699E+6	1034.0	7.833	1.697E+6	1.699E+6	1.701E+6	4001	20000
paid.tot[31]	1.463E+6	1304.0	10.27	1.461E+6	1.463E+6	1.466E+6	4001	20000
paid.tot[32]	1.524E+6	1430.0	11.03	1.521E+6	1.523E+6	1.527E+6	4001	20000
paid.tot[33]	1.77E+6	1690.0	15.07	1.767E+6	1.769E+6	1.773E+6	4001	20000
paid.tot[34]	1.568E+6	1855.0	13.39	1.565E+6	1.568E+6	1.572E+6	4001	20000
paid.tot[35]	1.231E+6	2223.0	18.09	1.227E+6	1.231E+6	1.236E+6	4001	20000
paid.tot[36]	1.253E+6	2542.0	19.62	1.249E+6	1.253E+6	1.259E+6	4001	20000
paid.tot[37]	1.497E+6	3145.0	25.21	1.491E+6	1.496E+6	1.504E+6	4001	20000
paid.tot[38]	1.171E+6	3633.0	25.21	1.165E+6	1.171E+6	1.179E+6	4001	20000
paid.tot[39]	1.557E+6	4251.0	29.92	1.55E+6	1.557E+6	1.567E+6	4001	20000
paid.tot[40]	1.424E+6	4740.0	39.85	1.416E+6	1.424E+6	1.435E+6	4001	20000
paid.tot[41]	1.51E+6	5989.0	43.93	1.499E+6	1.509E+6	1.523E+6	4001	20000
paid.tot[42]	1.212E+6	7881.0	52.93	1.199E+6	1.211E+6	1.23E+6	4001	20000
paid.tot[43]	1.309E+6	9988.0	75.09	1.291E+6	1.308E+6	1.33E+6	4001	20000
paid.tot[44]	1.267E+6	15770.0	105.9	1.24E+6	1.265E+6	1.302E+6	4001	20000
paid.tot[45]	1.381E+6	25260.0	199.7	1.338E+6	1.379E+6	1.437E+6	4001	20000
paid.tot[46]	1.595E+6	41060.0	289.7	1.523E+6	1.592E+6	1.684E+6	4001	20000
paid.tot[47]	1.496E+6	72870.0	526.0	1.367E+6	1.491E+6	1.653E+6	4001	20000
paid.tot[48]	1.463E+6	119700.0	899.8	1.247E+6	1.457E+6	1.719E+6	4001	20000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
reserve[26]	63.43	67.33	0.4838	0.9835	1.015	156.3	4001	20000
reserve[27]	592.6	256.3	1.909	174.0	561.3	1194.0	4001	20000
reserve[28]	1744.0	539.2	4.563	923.8	1666.0	3035.0	4001	20000
reserve[29]	3545.0	858.1	7.063	2176.0	3437.0	5502.0	4001	20000
reserve[30]	5251.0	1034.0	7.833	3536.0	5150.0	7591.0	4001	20000
reserve[31]	7143.0	1304.0	10.27	4963.0	7008.0	10100.0	4001	20000
reserve[32]	8878.0	1430.0	11.03	6435.0	8750.0	12050.0	4001	20000
reserve[33]	10680.0	1690.0	15.07	7815.0	10530.0	14440.0	4001	20000
reserve[34]	12510.0	1855.0	13.39	9359.0	12350.0	16640.0	4001	20000
reserve[35]	15100.0	2223.0	18.09	11350.0	14880.0	20060.0	4001	20000
reserve[36]	17980.0	2542.0	19.62	13650.0	17790.0	23550.0	4001	20000
reserve[37]	21630.0	3145.0	25.21	16380.0	21330.0	28740.0	4001	20000
reserve[38]	25870.0	3633.0	25.21	19790.0	25500.0	34040.0	4001	20000
reserve[39]	30900.0	4251.0	29.92	23770.0	30520.0	40520.0	4001	20000
reserve[40]	37580.0	4740.0	39.85	29360.0	37190.0	47930.0	4001	20000
reserve[41]	46950.0	5989.0	43.93	36760.0	46440.0	60060.0	4001	20000
reserve[42]	61940.0	7881.0	52.93	48530.0	61250.0	79400.0	4001	20000
reserve[43]	83580.0	9988.0	75.09	66270.0	82810.0	105400.0	4001	20000
reserve[44]	123300.0	15770.0	105.9	96490.0	121800.0	158500.0	4001	20000
reserve[45]	198400.0	25260.0	199.7	154800.0	196400.0	254500.0	4001	20000
reserve[46]	345500.0	41060.0	289.7	273600.0	342700.0	435300.0	4001	20000
reserve[47]	680600.0	72870.0	526.0	551600.0	675500.0	837700.0	4001	20000
reserve[48]	1.206E+6	119700.0	899.8	990500.0	1.2E+6	1.462E+6	4001	20000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**75.0%**	**90.0%**	**95.0%**	**97.5%**	**start**	**sample**
reserve.tot	2.946E+6	151600.0	1132.0	2.668E+6	2.939E+6	3.046E+6	3.142E+6	3.204E+6	3.265E+6	4001	20000

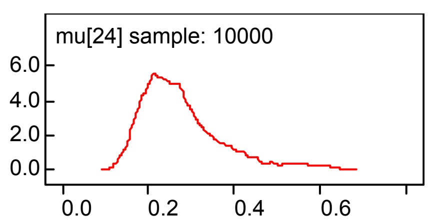
Case 1, gamma model a

The preliminary WinBUGS graph is as follows:



Case 1, gamma model a, selected statistics.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
r	9.842	0.4582	0.02357	9.011	9.813	10.78	4001	10000



node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[1]	4.079E-5	2.665E-6	1.013E-7	3.583E-5	4.069E-5	4.622E-5	4001	10000
mu[2]	1.876E-5	1.247E-6	4.94E-8	1.643E-5	1.87E-5	2.136E-5	4001	10000
mu[3]	2.936E-5	1.957E-6	7.359E-8	2.568E-5	2.93E-5	3.334E-5	4001	10000
mu[4]	6.737E-5	4.475E-6	1.681E-7	5.881E-5	6.726E-5	7.652E-5	4001	10000
mu[5]	1.311E-4	8.68E-6	3.17E-7	1.148E-4	1.308E-4	1.489E-4	4001	10000
mu[6]	2.491E-4	1.677E-5	6.246E-7	2.18E-4	2.486E-4	2.837E-4	4001	10000
mu[7]	4.576E-4	3.086E-5	1.144E-6	4.014E-4	4.562E-4	5.22E-4	4001	10000
mu[8]	6.599E-4	4.488E-5	1.612E-6	5.754E-4	6.586E-4	7.502E-4	4001	10000

mu[9]	0.001067	7.471E-5	2.739E-6	9.279E-4	0.001065	0.001221	4001	10000
mu[10]	0.001493	1.037E-4	3.686E-6	0.001303	0.00149	0.001707	4001	10000
mu[11]	0.001946	1.335E-4	5.051E-6	0.001693	0.001941	0.002214	4001	10000
mu[12]	0.002359	1.663E-4	5.711E-6	0.002046	0.002355	0.002701	4001	10000
mu[13]	0.002731	1.93E-4	6.754E-6	0.002377	0.002726	0.003123	4001	10000
mu[14]	0.003483	2.48E-4	8.501E-6	0.003029	0.003476	0.003986	4001	10000
mu[15]	0.003939	2.839E-4	1.004E-5	0.003409	0.003933	0.004526	4001	10000
mu[16]	0.005318	3.824E-4	1.314E-5	0.004603	0.005302	0.006116	4001	10000
mu[17]	0.005514	4.037E-4	1.401E-5	0.004765	0.005505	0.006347	4001	10000
mu[18]	0.005748	4.249E-4	1.418E-5	0.004952	0.005733	0.006632	4001	10000
mu[19]	0.005221	3.879E-4	1.321E-5	0.004493	0.00521	0.006024	4001	10000
mu[20]	0.005787	4.317E-4	1.44E-5	0.00497	0.005779	0.006669	4001	10000
mu[21]	0.005541	4.242E-4	1.413E-5	0.004734	0.005535	0.006409	4001	10000
mu[22]	0.008711	6.716E-4	2.235E-5	0.007454	0.008701	0.01012	4001	10000
mu[23]	0.01889	0.001488	4.847E-5	0.01616	0.01884	0.02195	4001	10000
mu[24]	0.2789	0.09743	0.005708	0.1496	0.2584	0.5538	4001	10000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p[1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[3]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[4]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[5]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[6]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[7]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[8]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[9]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[10]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[11]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[12]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[13]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[14]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[15]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[16]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[17]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[18]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[19]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[20]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[21]	0.0	0.0	1.0E-12	0.0	0.0	0.0	4001	10000
p[22]	1.472E-5	8.971E-4	1.441E-5	0.0	0.0	0.0	4001	10000
p[23]	0.001997	0.009551	2.488E-4	0.0	0.0	0.02964	4001	10000
p[24]	0.7123	0.08579	0.003388	0.5307	0.7173	0.865	4001	10000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
x	1.909	0.1089	0.00972	1.656	1.914	2.111	4001	10000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
y	0.3248	0.02374	0.002032	0.2768	0.3249	0.3728	4001	10000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
z[1]	6.263E-4	0.07088	0.001655	-0.1379	-0.002794	0.1414	4001	10000
z[2]	-0.002002	0.06968	0.001577	-0.139	-3.156E-5	0.1338	4001	10000
z[3]	0.002227	0.07105	0.001591	-0.1416	0.001805	0.1439	4001	10000
z[4]	-0.001225	0.07058	0.001686	-0.1399	-2.078E-5	0.1344	4001	10000
z[5]	-4.411E-4	0.0707	0.001702	-0.138	0.00179	0.1369	4001	10000
z[6]	0.00242	0.06984	0.001489	-0.1336	0.003826	0.1356	4001	10000
z[7]	0.001369	0.07087	0.001672	-0.1351	0.001787	0.1448	4001	10000
z[8]	9.369E-4	0.06987	0.001549	-0.1414	0.001353	0.141	4001	10000
z[9]	-3.677E-5	0.07044	0.001591	-0.145	0.003551	0.1304	4001	10000
z[10]	-0.001185	0.07251	0.001459	-0.1425	-0.001196	0.142	4001	10000
z[11]	-0.002186	0.06922	0.001537	-0.1354	-0.004336	0.1368	4001	10000
z[12]	0.001208	0.07246	0.001809	-0.1389	0.001053	0.1431	4001	10000
z[13]	0.002075	0.07166	0.001601	-0.1438	0.002378	0.1373	4001	10000
z[14]	-0.002616	0.07138	0.001573	-0.1445	-0.00156	0.1323	4001	10000
z[15]	-0.001193	0.07048	0.001397	-0.1362	-0.002302	0.1373	4001	10000
z[16]	0.001243	0.0737	0.001568	-0.149	0.002019	0.1393	4001	10000
z[17]	-1.495E-4	0.0703	0.001484	-0.1353	-9.959E-4	0.1366	4001	10000
z[18]	0.002792	0.07057	0.001542	-0.14	0.001859	0.1416	4001	10000

z[19]	3.847E-5	0.07147	0.001563	-0.1443	0.00135	0.134	4001	10000
z[20]	-0.001455	0.07032	0.001766	-0.1388	-0.002227	0.1377	4001	10000
z[21]	-4.698E-4	0.07072	0.001733	-0.1366	1.551E-4	0.1366	4001	10000
z[22]	-7.648E-4	0.07136	0.001628	-0.1449	-0.001006	0.1412	4001	10000
z[23]	-0.009111	0.06537	0.002093	-0.141	-0.00733	0.1163	4001	10000
z[24]	-0.02154	0.06253	0.004804	-0.1425	-0.02106	0.09809	4001	10000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
paid[48,2]	523100.0	170500.0	1674.0	248700.0	503800.0	910900.0	4001	10000
paid[48,3]	335400.0	1.08E+5	1097.0	160300.0	322700.0	580800.0	4001	10000
paid[48,4]	147400.0	47990.0	510.2	68610.0	1.43E+5	256500.0	4001	10000
paid[48,5]	75350.0	24440.0	241.8	35830.0	72800.0	129900.0	4001	10000
paid[48,6]	39710.0	12800.0	131.2	19010.0	38260.0	68420.0	4001	10000
paid[48,7]	21510.0	6990.0	78.46	10270.0	20730.0	37760.0	4001	10000
paid[48,8]	14910.0	4807.0	49.67	7122.0	14400.0	25740.0	4001	10000
paid[48,9]	9231.0	2947.0	27.56	4371.0	8911.0	15830.0	4001	10000
paid[48,10]	6628.0	2133.0	18.99	3175.0	6385.0	11430.0	4001	10000
paid[48,11]	5061.0	1633.0	15.67	2404.0	4884.0	8624.0	4001	10000
paid[48,12]	4184.0	1367.0	12.91	1968.0	4043.0	7316.0	4001	10000
paid[48,13]	3609.0	1153.0	10.43	1738.0	3492.0	6174.0	4001	10000
paid[48,14]	2809.0	917.4	8.407	1318.0	2708.0	4889.0	4001	10000
paid[48,15]	2507.0	813.0	7.645	1188.0	2423.0	4358.0	4001	10000
paid[48,16]	1850.0	600.1	6.509	868.3	1788.0	3222.0	4001	10000
paid[48,17]	1786.0	571.1	5.375	845.2	1726.0	3085.0	4001	10000
paid[48,18]	1718.0	556.7	4.956	822.2	1652.0	2965.0	4001	10000
paid[48,19]	1891.0	614.4	6.867	899.4	1826.0	3267.0	4001	10000
paid[48,20]	1709.0	556.9	4.818	804.0	1645.0	2984.0	4001	10000
paid[48,21]	1778.0	579.3	6.235	830.9	1711.0	3071.0	4001	10000
paid[48,22]	1139.0	372.5	3.845	529.9	1095.0	1995.0	4001	10000
paid[48,23]	524.4	171.5	2.001	246.2	508.0	910.9	4001	10000
paid[48,24]	39.79	62.3	0.6898	0.9821	1.005	171.0	4001	10000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
paid.tot[26]	1.503E+6	62.32	0.8724	1.503E+6	1.503E+6	1.503E+6	4001	10000
paid.tot[27]	1.049E+6	183.0	1.728	1.049E+6	1.049E+6	1.049E+6	4001	10000
paid.tot[28]	1.477E+6	416.6	4.426	1.476E+6	1.477E+6	1.478E+6	4001	10000
paid.tot[29]	1.597E+6	714.3	7.287	1.596E+6	1.597E+6	1.599E+6	4001	10000
paid.tot[30]	1.699E+6	893.5	8.627	1.697E+6	1.699E+6	1.701E+6	4001	10000
paid.tot[31]	1.463E+6	1098.0	11.24	1.461E+6	1.463E+6	1.466E+6	4001	10000
paid.tot[32]	1.523E+6	1226.0	12.32	1.521E+6	1.523E+6	1.526E+6	4001	10000
paid.tot[33]	1.769E+6	1354.0	13.32	1.767E+6	1.769E+6	1.772E+6	4001	10000
paid.tot[34]	1.568E+6	1495.0	12.51	1.565E+6	1.568E+6	1.571E+6	4001	10000
paid.tot[35]	1.231E+6	1692.0	17.26	1.228E+6	1.231E+6	1.234E+6	4001	10000
paid.tot[36]	1.253E+6	1895.0	16.69	1.25E+6	1.253E+6	1.257E+6	4001	10000
paid.tot[37]	1.497E+6	2237.0	21.37	1.492E+6	1.496E+6	1.501E+6	4001	10000
paid.tot[38]	1.171E+6	2629.0	26.44	1.166E+6	1.171E+6	1.176E+6	4001	10000
paid.tot[39]	1.557E+6	3097.0	30.63	1.551E+6	1.557E+6	1.564E+6	4001	10000
paid.tot[40]	1.424E+6	3777.0	34.34	1.417E+6	1.424E+6	1.432E+6	4001	10000
paid.tot[41]	1.509E+6	4822.0	46.27	1.501E+6	1.509E+6	1.519E+6	4001	10000
paid.tot[42]	1.212E+6	6812.0	64.57	1.199E+6	1.211E+6	1.226E+6	4001	10000
paid.tot[43]	1.308E+6	9728.0	86.15	1.29E+6	1.307E+6	1.329E+6	4001	10000
paid.tot[44]	1.266E+6	16140.0	143.3	1.238E+6	1.265E+6	1.301E+6	4001	10000
paid.tot[45]	1.38E+6	28890.0	284.3	1.329E+6	1.378E+6	1.442E+6	4001	10000
paid.tot[46]	1.593E+6	55100.0	543.0	1.497E+6	1.589E+6	1.713E+6	4001	10000
paid.tot[47]	1.496E+6	122200.0	1094.0	1.286E+6	1.488E+6	1.759E+6	4001	10000
paid.tot[48]	1.461E+6	209900.0	2016.0	1.096E+6	1.446E+6	1.918E+6	4001	10000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
reserve[26]	40.03	62.32	0.8724	0.982	1.006	170.5	4001	10000
reserve[27]	564.5	183.0	1.728	259.8	543.5	969.2	4001	10000
reserve[28]	1697.0	416.6	4.427	992.2	1658.0	2611.0	4001	10000
reserve[29]	3479.0	714.3	7.287	2252.0	3420.0	5052.0	4001	10000
reserve[30]	5177.0	893.5	8.627	3590.0	5113.0	7093.0	4001	10000
reserve[31]	7091.0	1098.0	11.24	5109.0	7022.0	9402.0	4001	10000
reserve[32]	8804.0	1226.0	12.32	6572.0	8750.0	11390.0	4001	10000
reserve[33]	10570.0	1354.0	13.32	8100.0	10510.0	13370.0	4001	10000
reserve[34]	12460.0	1495.0	12.51	9708.0	12410.0	15570.0	4001	10000
reserve[35]	14960.0	1692.0	17.26	11820.0	14880.0	18420.0	4001	10000

reserve[36]	17770.0	1895.0	16.69	14230.0	17710.0	21690.0	4001	10000
reserve[37]	21410.0	2237.0	21.37	17280.0	21320.0	26030.0	4001	10000
reserve[38]	25600.0	2629.0	26.44	20770.0	25490.0	31090.0	4001	10000
reserve[39]	30700.0	3097.0	30.63	25020.0	30550.0	37160.0	4001	10000
reserve[40]	37270.0	3777.0	34.34	30330.0	37080.0	45090.0	4001	10000
reserve[41]	46530.0	4822.0	46.27	37890.0	46320.0	56740.0	4001	10000
reserve[42]	61430.0	6812.0	64.57	49200.0	6.1E+4	75660.0	4001	10000
reserve[43]	83040.0	9728.0	86.15	65460.0	82470.0	103700.0	4001	10000
reserve[44]	122500.0	16140.0	143.3	93980.0	121600.0	1.57E+5	4001	10000
reserve[45]	197500.0	28890.0	284.3	146400.0	195400.0	258700.0	4001	10000
reserve[46]	344300.0	55100.0	543.0	2.48E+5	339900.0	4.64E+5	4001	10000
reserve[47]	681100.0	122200.0	1094.0	471200.0	673100.0	9.44E+5	4001	10000
reserve[48]	1.204E+6	209900.0	2016.0	839700.0	1.189E+6	1.661E+6	4001	10000

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.938E+6	254100	2344.0	2.485E+6	2.923E+6	3.1E+6	3.275E+6	3.379E+6	3.479E+6

Case 1, gamma model b, selected statistics.

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
reserve.tot	2.936E+6	144900	1328.0	2.661E+6	2.932E+6	3.031E+6	3.126E+6	3.181E+6	3.226E+6

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
reserve[26]	49.46	64.82	0.6925	0.8336	1.077	153.7	14001	10000
reserve[27]	574.1	217.3	2.372	218.3	549.4	1073.0	14001	10000
reserve[28]	1710.0	485.3	4.554	908.9	1662.0	2807.0	14001	10000
reserve[29]	3494.0	784.2	6.953	2142.0	3436.0	5184.0	14001	10000
reserve[30]	5192.0	964.7	9.644	3487.0	5127.0	7280.0	14001	10000
reserve[31]	7090.0	1232.0	13.04	4920.0	7007.0	9734.0	14001	10000
reserve[32]	8821.0	1363.0	15.07	6382.0	8737.0	11740.0	14001	10000
reserve[33]	10610.0	1594.0	17.43	7800.0	10520.0	14020.0	14001	10000
reserve[34]	12460.0	1758.0	18.01	9325.0	12350.0	16210.0	14001	10000
reserve[35]	14980.0	2085.0	18.19	11230.0	14840.0	19500.0	14001	10000
reserve[36]	17790.0	2309.0	25.22	13550.0	17700.0	22590.0	14001	10000
reserve[37]	21470.0	2873.0	27.12	16400.0	21280.0	27560.0	14001	10000
reserve[38]	25630.0	3211.0	36.47	19790.0	25450.0	32400.0	14001	10000
reserve[39]	30700.0	3873.0	42.29	23760.0	30470.0	39050.0	14001	10000
reserve[40]	37340.0	4262.0	44.09	29660.0	37090.0	46250.0	14001	10000
reserve[41]	46510.0	5392.0	59.1	36620.0	46180.0	58130.0	14001	10000
reserve[42]	61550.0	7172.0	73.67	48260.0	61230.0	76600.0	14001	10000
reserve[43]	82970.0	9417.0	88.38	65590.0	82560.0	102900.0	14001	10000
reserve[44]	122400.0	14940.0	150.5	95620.0	121500.0	153800.0	14001	10000
reserve[45]	197700.0	24180.0	243.4	153500.0	196600.0	248300.0	14001	10000
reserve[46]	344100.0	39910.0	386.3	271500.0	342400.0	429100.0	14001	10000
reserve[47]	679400.0	70370.0	714.3	551300.0	675700.0	825300.0	14001	10000
reserve[48]	1.203E+6	113700.0	1054.0	992400.0	1.2E+6	1.433E+6	14001	10000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
paid.tot[26]	1.503E+6	64.82	0.6925	1.503E+6	1.503E+6	1.503E+6	14001	10000
paid.tot[27]	1.049E+6	217.3	2.372	1.049E+6	1.049E+6	1.05E+6	14001	10000
paid.tot[28]	1.477E+6	485.3	4.554	1.476E+6	1.477E+6	1.478E+6	14001	10000
paid.tot[29]	1.597E+6	784.2	6.954	1.596E+6	1.597E+6	1.599E+6	14001	10000
paid.tot[30]	1.699E+6	964.7	9.644	1.697E+6	1.699E+6	1.701E+6	14001	10000
paid.tot[31]	1.463E+6	1232.0	13.04	1.461E+6	1.463E+6	1.466E+6	14001	10000
paid.tot[32]	1.523E+6	1363.0	15.07	1.521E+6	1.523E+6	1.526E+6	14001	10000
paid.tot[33]	1.769E+6	1594.0	17.43	1.767E+6	1.769E+6	1.773E+6	14001	10000
paid.tot[34]	1.568E+6	1758.0	18.01	1.565E+6	1.568E+6	1.572E+6	14001	10000
paid.tot[35]	1.231E+6	2085.0	18.19	1.227E+6	1.231E+6	1.235E+6	14001	10000
paid.tot[36]	1.253E+6	2309.0	25.22	1.249E+6	1.253E+6	1.258E+6	14001	10000
paid.tot[37]	1.497E+6	2873.0	27.12	1.491E+6	1.496E+6	1.503E+6	14001	10000
paid.tot[38]	1.171E+6	3211.0	36.47	1.165E+6	1.171E+6	1.178E+6	14001	10000
paid.tot[39]	1.557E+6	3873.0	42.29	1.55E+6	1.557E+6	1.566E+6	14001	10000
paid.tot[40]	1.424E+6	4262.0	44.09	1.416E+6	1.424E+6	1.433E+6	14001	10000
paid.tot[41]	1.509E+6	5392.0	59.1	1.499E+6	1.509E+6	1.521E+6	14001	10000
paid.tot[42]	1.212E+6	7172.0	73.67	1.198E+6	1.211E+6	1.227E+6	14001	10000
paid.tot[43]	1.308E+6	9417.0	88.38	1.291E+6	1.307E+6	1.328E+6	14001	10000
paid.tot[44]	1.266E+6	14940.0	150.5	1.239E+6	1.265E+6	1.297E+6	14001	10000

paid.tot[45]	1.381E+6	24180.0	243.4	1.336E+6	1.379E+6	1.431E+6	14001	10000
paid.tot[46]	1.593E+6	39910.0	386.3	1.521E+6	1.592E+6	1.678E+6	14001	10000
paid.tot[47]	1.495E+6	70370.0	714.3	1.366E+6	1.491E+6	1.641E+6	14001	10000
paid.tot[48]	1.46E+6	113700.0	1054.0	1.249E+6	1.456E+6	1.69E+6	14001	10000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
paid[48,2]	523900.0	88830.0	930.0	365900.0	518200.0	7.11E+5	14001	10000
paid[48,3]	335200.0	57890.0	555.8	230700.0	331900.0	457600.0	14001	10000
paid[48,4]	146100.0	31270.0	289.6	90420.0	144200.0	2.14E+5	14001	10000
paid[48,5]	75110.0	19330.0	202.8	42020.0	73630.0	117600.0	14001	10000
paid[48,6]	39470.0	11350.0	111.7	20590.0	38350.0	65210.0	14001	10000
paid[48,7]	21550.0	6069.0	62.73	11480.0	20970.0	34840.0	14001	10000
paid[48,8]	15040.0	4830.0	50.01	7123.0	14530.0	26070.0	14001	10000
paid[48,9]	9229.0	3278.0	35.68	3956.0	8855.0	16750.0	14001	10000
paid[48,10]	6574.0	1853.0	15.77	3392.0	6438.0	10730.0	14001	10000
paid[48,11]	5123.0	2107.0	21.29	1856.0	4834.0	9998.0	14001	10000
paid[48,12]	4190.0	1542.0	16.7	1735.0	4010.0	7707.0	14001	10000
paid[48,13]	3626.0	1639.0	15.44	1146.0	3395.0	7491.0	14001	10000
paid[48,14]	2843.0	1090.0	9.995	1116.0	2701.0	5342.0	14001	10000
paid[48,15]	2512.0	1078.0	11.46	851.2	2361.0	5023.0	14001	10000
paid[48,16]	1862.0	785.5	7.736	653.1	1747.0	3688.0	14001	10000
paid[48,17]	1799.0	796.0	7.838	592.9	1681.0	3661.0	14001	10000
paid[48,18]	1715.0	565.1	5.461	787.7	1651.0	2982.0	14001	10000
paid[48,19]	1902.0	770.1	8.575	700.4	1803.0	3680.0	14001	10000
paid[48,20]	1701.0	561.8	5.326	782.1	1640.0	2975.0	14001	10000
paid[48,21]	1782.0	624.2	6.287	786.5	1713.0	3181.0	14001	10000
paid[48,22]	1141.0	433.7	4.005	451.8	1092.0	2131.0	14001	10000
paid[48,23]	523.1	206.1	2.032	201.0	495.8	996.5	14001	10000
paid[48,24]	51.07	65.5	0.7271	0.832	1.08	153.5	14001	10000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
x	1.94	0.1221	0.009858	1.678	1.942	2.178	4001	20000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
y	0.3291	0.0265	0.002115	0.2692	0.3282	0.3799	4001	20000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
z[1]	4.494E-5	0.07187	0.001138	-0.1442	-7.0E-4	0.1406	4001	20000
z[2]	8.431E-4	0.07003	0.001149	-0.134	0.001581	0.1417	4001	20000
z[3]	0.001118	0.07043	0.001047	-0.1355	3.966E-4	0.1414	4001	20000
z[4]	-4.784E-5	0.06979	0.001017	-0.139	1.612E-4	0.136	4001	20000
z[5]	-0.001846	0.0708	0.001104	-0.1393	-0.003077	0.1361	4001	20000
z[6]	0.00112	0.07039	0.001162	-0.1378	0.001632	0.1409	4001	20000
z[7]	0.001091	0.07256	0.00111	-0.1408	6.524E-4	0.1453	4001	20000
z[8]	-2.065E-4	0.0713	0.001094	-0.1394	-6.394E-4	0.1401	4001	20000
z[9]	7.574E-4	0.0701	0.001051	-0.136	0.001077	0.1375	4001	20000
z[10]	1.035E-4	0.07049	0.001283	-0.1418	4.345E-6	0.1373	4001	20000
z[11]	4.231E-4	0.07011	0.001164	-0.1358	2.093E-4	0.1392	4001	20000
z[12]	-8.688E-4	0.07	0.001086	-0.1351	-0.00147	0.1365	4001	20000
z[13]	4.216E-4	0.0692	0.001142	-0.1331	6.007E-4	0.1367	4001	20000
z[14]	-4.666E-4	0.07088	0.001047	-0.1387	-9.909E-4	0.1394	4001	20000
z[15]	-1.916E-4	0.07037	0.00108	-0.1388	-0.001587	0.1364	4001	20000
z[16]	-9.451E-4	0.07191	0.001213	-0.144	-0.001503	0.1362	4001	20000
z[17]	-0.001091	0.07073	0.001086	-0.1424	-0.002015	0.1384	4001	20000
z[18]	-5.012E-4	0.07014	0.001168	-0.1379	3.517E-4	0.1344	4001	20000
z[19]	0.001325	0.07084	0.001141	-0.1374	7.753E-4	0.1396	4001	20000
z[20]	-6.046E-4	0.07067	0.00105	-0.1409	-2.985E-4	0.1338	4001	20000
z[21]	-0.002306	0.07162	0.001203	-0.1452	-0.001254	0.1398	4001	20000
z[22]	4.234E-5	0.06969	0.00105	-0.1342	-0.002492	0.1404	4001	20000
z[23]	-0.009785	0.06523	0.001454	-0.142	-0.008308	0.1147	4001	20000
z[24]	-0.03123	0.06042	0.004309	-0.144	-0.03234	0.1012	4001	20000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
p[1]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[2]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[3]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[4]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000

p[5]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[6]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[7]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[8]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[9]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[10]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[11]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[12]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[13]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[14]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[15]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[16]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[17]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[18]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[19]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[20]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[21]	0.0	0.0	7.071E-13	0.0	0.0	0.0	4001	20000
p[22]	1.184E-6	8.811E-5	8.524E-7	0.0	0.0	0.0	4001	20000
p[23]	0.002624	0.01202	2.976E-4	0.0	0.0	0.03791	4001	20000
p[24]	0.6346	0.08543	0.001995	0.46	0.638	0.7909	4001	20000

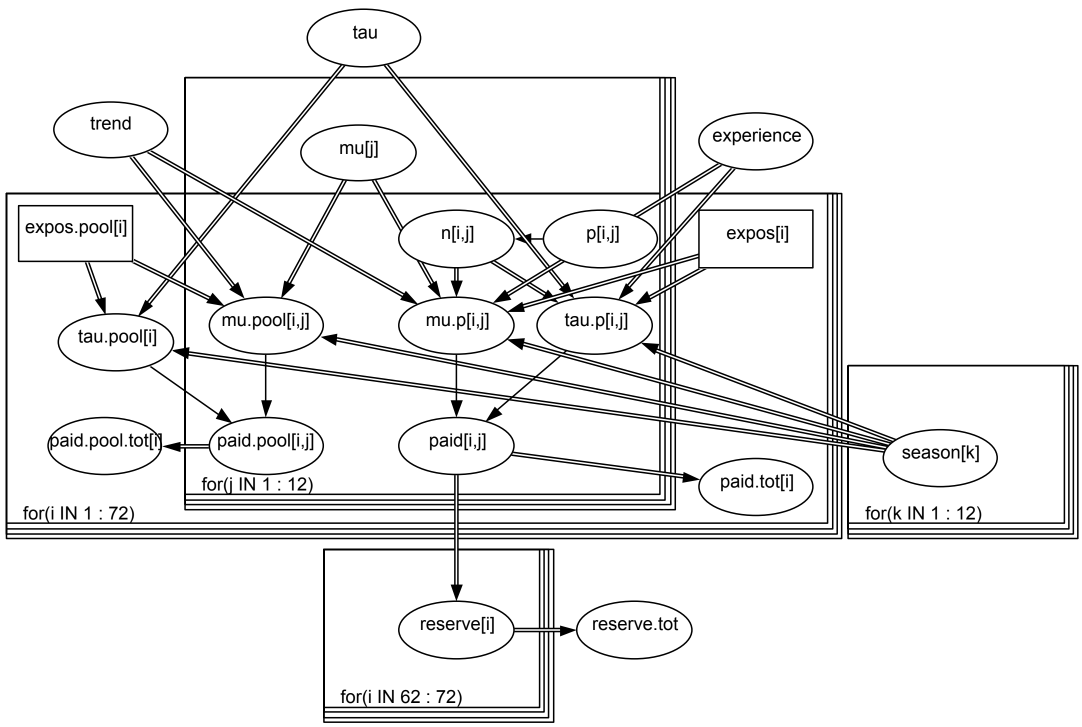
node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
r[1]	18.2	3.916	0.212	11.55	17.94	26.53	4001	20000
r[2]	36.15	7.396	0.4739	23.88	35.4	52.77	4001	20000
r[3]	35.22	7.41	0.4609	22.77	34.5	51.75	4001	20000
r[4]	23.28	4.982	0.279	14.21	23.09	33.78	4001	20000
r[5]	16.29	3.55	0.1817	10.09	16.01	23.78	4001	20000
r[6]	12.71	2.759	0.1262	7.898	12.49	18.68	4001	20000
r[7]	13.26	3.01	0.1496	7.88	13.04	19.89	4001	20000
r[8]	10.38	2.177	0.09114	6.528	10.2	15.04	4001	20000
r[9]	8.481	1.791	0.06894	5.317	8.36	12.37	4001	20000
r[10]	13.49	3.373	0.1675	7.84	13.21	20.79	4001	20000
r[11]	6.337	1.477	0.05182	3.828	6.193	9.549	4001	20000
r[12]	8.019	1.853	0.06569	4.735	7.876	12.03	4001	20000
r[13]	5.304	1.248	0.03715	3.17	5.203	7.978	4001	20000
r[14]	7.342	1.692	0.05836	4.349	7.218	11.01	4001	20000
r[15]	5.896	1.451	0.04944	3.438	5.788	9.041	4001	20000
r[16]	6.547	1.639	0.0589	3.737	6.42	10.18	4001	20000
r[17]	5.668	1.418	0.0486	3.307	5.547	8.888	4001	20000
r[18]	10.14	2.566	0.1091	5.807	9.948	15.74	4001	20000
r[19]	6.753	1.786	0.06786	3.77	6.572	10.75	4001	20000
r[20]	10.21	2.694	0.1059	5.53	10.03	16.02	4001	20000
r[21]	9.325	2.524	0.09709	4.977	9.122	14.93	4001	20000
r[22]	8.018	2.162	0.07891	4.195	7.828	12.88	4001	20000
r[23]	7.478	2.128	0.07755	3.965	7.279	12.16	4001	20000
r[24]	63.25	34.52	1.144	14.97	56.61	146.1	4001	20000

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**97.5%**	**start**	**sample**
mu[1]	7.546E-5	1.645E-5	8.904E-7	4.775E-5	7.431E-5	1.105E-4	4001	20000
mu[2]	6.892E-5	1.419E-5	9.095E-7	4.549E-5	6.745E-5	1.007E-4	4001	20000
mu[3]	1.051E-4	2.227E-5	1.386E-6	6.763E-5	1.03E-4	1.544E-4	4001	20000
mu[4]	1.592E-4	3.443E-5	1.926E-6	9.653E-5	1.578E-4	2.317E-4	4001	20000
mu[5]	2.168E-4	4.796E-5	2.458E-6	1.335E-4	2.129E-4	3.181E-4	4001	20000
mu[6]	3.219E-4	7.117E-5	3.257E-6	1.979E-4	3.154E-4	4.752E-4	4001	20000
mu[7]	6.165E-4	1.427E-4	7.094E-6	3.625E-4	6.068E-4	9.257E-4	4001	20000
mu[8]	6.957E-4	1.493E-4	6.242E-6	4.324E-4	6.834E-4	0.001016	4001	20000
mu[9]	9.205E-4	2.004E-4	7.686E-6	5.665E-4	9.074E-4	0.001357	4001	20000
mu[10]	0.002045	5.211E-4	2.59E-5	0.001171	0.002001	0.003176	4001	20000
mu[11]	0.001253	3.038E-4	1.07E-5	7.357E-4	0.001224	0.001918	4001	20000
mu[12]	0.001924	4.592E-4	1.629E-5	0.00111	0.001891	0.002916	4001	20000
mu[13]	0.00147	3.63E-4	1.078E-5	8.475E-4	0.001437	0.002255	4001	20000
mu[14]	0.002597	6.187E-4	2.128E-5	0.001493	0.002554	0.003934	4001	20000
mu[15]	0.00236	6.05E-4	2.055E-5	0.001334	0.002313	0.003683	4001	20000
mu[16]	0.003541	9.222E-4	3.304E-5	0.001967	0.003468	0.005576	4001	20000
mu[17]	0.003179	8.324E-4	2.863E-5	0.001789	0.003108	0.005073	4001	20000
mu[18]	0.005926	0.001538	6.534E-5	0.003342	0.005796	0.009261	4001	20000
mu[19]	0.003586	9.829E-4	3.73E-5	0.001962	0.003495	0.005782	4001	20000
mu[20]	0.006002	0.001623	6.378E-5	0.003174	0.005897	0.009487	4001	20000
mu[21]	0.00525	0.001457	5.605E-5	0.002751	0.005139	0.008486	4001	20000

mu[22]	0.007098	0.001969	7.176E-5	0.003621	0.006936	0.0115	4001	20000
mu[23]	0.01435	0.004217	1.532E-4	0.007332	0.01393	0.02362	4001	20000
mu[24]	1.286	0.6338	0.02036	0.3381	1.191	2.78	4001	20000

Case 2.

The preliminary WinBUGS graph is as follows:



Case 2 data is as follows.

expos[] paid[,1] paid[,2] paid[,3] paid[,4] paid[,5] paid[,6] paid[,7] paid[,8] paid[,9] paid[,10] paid[,11] paid[,12] paid[,13] paid[,14] paid[,15] paid[,16] paid[,17] paid[,18] paid[,19] paid[,20] paid[,21] paid[,22] paid[,23] paid[,24]
7000 86816 216800 98317 55486 30844 17959 6213 11121 4189 2467 1170 1154 650 622 255 238 1 658 1 957 1 292 375 1
7000 81769 162997 143574 36709 23537 5886 8772 3963 7474 3667 4151 993 1205 300 843 213 1511 1285 571 1853 268 1 255 1
7000 60108 225248 155455 35357 21330 13276 10803 3673 2658 4698 262 861 478 257 2268 667 1 284 284 560 1 1 262 448
7000 114421 186747 159929 68347 40418 20121 10963 6873 4180 4705 2335 2429 1767 282 1 1581 345 1 1 693 684 460 380 1
7000 168747 166579 115787 49301 31721 13241 7627 2286 1134 1195 1830 815 399 539 795 682 213 1093 1 282 1 797 463 1559
7000 143870 249262 123045 55311 43809 23210 12186 12148 9633 4589 3066 878 241 925 2068 1277 257 1061 1481 765 1830 1 1 1
7000 80699 273008 148497 54222 34213 6624 6527 2205 2017 1969 2777 1293 1111 876 632 1 384 218 639 296 1 1 616 364
7000 64028 218861 172072 40786 38353 23485 7861 9106 5036 1153 4166 286 338 215 688 1635 1 1814 246 278 1 1 396 227
7000 90845 149604 88201 33982 21816 10894 11901 4189 1400 3366 378 1233 235 1017 238 218 1 1 1171 430 1219 270 1 1

7000 129086 150054 110216 56322 21046 12387 9355 6331 2778 1640 6263 649 1275
868 221 1841 396 331 1 1 262 1 1 220
7000 118904 172019 134211 74210 21315 12692 6335 3590 3107 7920 1767 496 2544
1280 1025 1007 2882 294 1 1 493 1 1 1
7000 91608 195812 105834 70655 50450 29905 8464 6872 6868 4392 2004 616 4784
1764 386 3348 1 289 6083 819 1 326 250 508

7000 114072 242603 131239 75548 62062 16792 12171 5744 1094 2347 291 924 1 453
1170 455 3543 658 258 256 1 1729 352 1
7000 96856 275734 158214 75360 24909 18686 7397 6243 2710 349 644 1 286 1265 3343
271 561 236 267 1007 1545 1 1121 1
8000 110540 274596 180333 52314 28516 14407 8816 9491 4540 4805 1733 2968 2875
730 1254 1419 2780 1 1 2170 1473 1 279 1
8000 126239 222427 210130 62963 27709 13746 9897 4287 6728 2198 1548 949 1468
1735 3216 1 2041 219 330 1380 1132 1 1 1
8000 145256 256145 234171 64235 28341 20959 9514 4696 11926 8580 2982 760 3833
1866 3051 1 1759 292 314 1 511 206 1093 343
8000 143516 245405 204784 94076 35149 15402 11464 9198 5850 6488 3301 912 284
2512 1 1226 1230 1102 292 647 294 300 525 296
8000 126956 278438 170439 75153 41790 24818 9386 10870 3533 1577 5218 1578 3808
324 1767 2300 2693 1382 935 524 681 265 1 1
8000 137001 247025 111395 46820 27528 12850 13302 7878 2774 1376 636 2090 1676
2146 217 784 326 1 219 1 3746 1 1 1
8000 100330 221625 105534 58036 23708 19279 8061 3567 2052 3193 9428 1637 678
2035 672 3088 1367 502 1 1039 1 279 1 271
8000 166517 253663 156820 77509 69586 15783 6234 7041 3881 4581 4440 664 853
1446 242 338 1607 1412 839 295 535 1 1 268
8000 120027 299088 201973 59948 43599 13534 8814 7265 2419 2982 2197 4839 2404
7646 1 904 277 1 1 1203 469 242 1 1
8000 113868 190406 128469 64172 20761 16561 12348 6428 3330 2447 4749 1147 1068
232 1404 4114 1767 317 212 1 1737 1 270 1

8000 133564 284546 219423 88291 34330 26245 10430 11235 2476 1727 515 3034 919
1221 314 1 1320 2431 703 970 493 1 248 1
8000 131542 220067 125285 82929 28438 18795 9942 7504 2593 3215 2707 1507 420
537 2736 1 1 1496 486 558 656 1 1 1
8000 185360 282581 147556 95291 44032 14920 11945 8636 3666 1416 3358 4287 2289
514 992 2432 1052 276 329 1 1 429 1 1
8000 146041 225452 263343 73365 43089 17702 6139 9144 8692 4992 3169 3468 290
605 897 512 3101 1 1292 1649 437 1944 457 265
8000 134096 176581 155182 54729 66552 19258 17684 6577 7043 3342 1608 1764 1590
2317 617 1888 1 3263 1804 881 354 1 1 1
8000 87686 232124 246038 90189 23213 18899 10388 3583 4738 3092 5141 1922 1016
1551 1534 2041 950 385 610 1607 1 284 640 245
8000 95000 275579 205353 105624 68162 19753 10291 12313 4494 3109 9620 1958 6177
442 3239 1357 6340 2036 366 245 3069 554 1 270
7000 152499 187442 168633 44110 37106 7825 9228 3693 6513 2464 2087 635 3078 376
237 5253 1060 3301 215 254 1 300 1 1
7000 153716 232351 122659 33835 13194 23286 16582 5505 2792 1772 6256 1954 550
1797 940 1988 1 1917 319 1969 1849 1 1 1
7000 169446 214118 157902 65956 24618 14575 11861 4551 1582 3907 712 1142 262
1076 1 1185 1 413 1752 3667 946 294 226 1
7000 194507 240782 164444 87312 31946 23260 12522 5135 2643 6008 1374 2608 1001
1181 1788 1 1 413 1545 1196 263 1 1 1
7000 109816 206099 137075 49060 13799 10874 5751 3607 3951 1248 1916 234 1153
264 468 212 672 487 807 226 1 424 1 1

7000 88556 210175 179759 56281 23480 13764 6525 4708 3407 3096 4927 898 843 1564 298 1 1 241 1839 1 841 310 284 1
7000 186152 362103 177967 111944 53261 26389 14668 7425 7198 6457 2201 925 1621 3168 1452 231 272 1 1504 1 1 1 1 1
7000 113245 350576 297316 117191 61876 47203 6584 19441 5227 1832 6099 2897 2604 10643 1223 518 1 1579 590 714 1 288 1 1
7000 133301 258574 229926 85961 25734 29691 10713 15145 11713 1700 2998 1457 1483 1304 1592 1673 2041 2056 1260 412 270 1 709 548
8000 80602 289007 184488 82839 51070 26964 10082 4628 5014 4293 9031 3922 237 2373 659 3294 1435 298 299 456 762 562 253 1
8000 105792 298609 187895 93696 37372 21922 10473 16714 4213 5410 7783 725 2243 496 1970 398 3152 1047 936 1 206 1 1 1
8000 115337 460154 160173 113481 60991 26735 9882 7865 5876 7614 1548 526 1261 3706 2860 3627 293 569 1592 999 2702 1064 933 362
8000 84459 266411 128725 62307 48258 24187 15487 6907 8121 9603 4443 1975 270 1213 1 1765 831 1 1439 654 1364 1 1 276
8000 176166 298027 196979 71734 37868 17913 16184 12561 13995 3003 3671 3125 2452 1409 3774 853 2131 525 705 1080 276 422 1 1
8000 103254 368371 222945 64930 52702 47743 26579 9271 14446 3898 6873 1814 3833 1107 1 988 1 613 3973 661 1 3357 575 1
8000 116473 291780 327050 47525 59336 23891 16160 2935 5599 3321 1232 1556 2687 854 1 356 1697 212 280 354 385 329 278 306
8000 121473 212389 126004 61019 36141 26096 5346 5526 3113 3760 770 1 991 834 1 661 475 821 1 1 1181 1078 1 248

8000 166705 254223 165098 62368 62540 28019 13492 16275 3456 6954 2731 1471 1282 524 510 387 1441 3847 2432 1434 1 284 1 1
8000 190929 572151 324517 201685 61060 26077 10394 9738 7598 11206 6242 3111 1523 2858 2511 519 1051 1786 1213 1 1 980 220 NA
8000 188658 419150 245021 81121 65813 34202 21861 11599 9326 7971 6890 4837 1298 2129 3106 2504 1508 7475 6335 1833 8007 1 NA NA
9000 108792 440106 331281 94601 51031 22633 24272 13026 13978 4191 11891 8955 3060 580 4384 702 1251 512 2068 1226 1015 NA NA NA
9000 174411 341673 223826 82830 55584 31411 27941 7578 9922 7601 5821 3650 248 3550 2188 1 4249 890 553 561 NA NA NA NA
9000 186694 476252 204697 79958 51094 34726 15969 6232 6495 5151 6283 2852 4245 401 1616 2033 2291 2547 673 NA NA NA NA NA
9000 84620 389662 253896 121324 75262 31814 25096 14509 7058 3244 6440 4525 3545 569 1080 1 2638 8081 NA NA NA NA NA NA
9000 232996 455749 464896 137734 55779 41070 10734 7339 10506 7644 2837 3038 2995 2308 6464 282 406 NA NA NA NA NA NA NA
9000 238851 355587 240618 89568 62221 23748 12071 10044 13000 6981 8541 3285 882 2162 2623 485 NA NA NA NA NA NA NA NA
9000 200019 349144 297872 134561 93997 46159 21619 9665 5435 15328 17701 3666 407 1503 3144 NA NA NA NA NA NA NA NA NA
9000 175749 428763 297497 80779 49175 39672 14371 18738 5683 5109 1542 9991 262 271 NA NA NA NA NA NA NA NA NA NA
9000 232380 333770 241532 112576 86405 40956 17470 9130 13021 10605 6633 6575 2350 NA NA NA NA NA NA NA NA NA NA NA

9000 127164 380391 220934 88864 61997 21316 20899 16218 16233 8395 636 1910 NA NA NA NA NA NA NA NA NA NA NA NA
9000 230999 447766 297169 106097 87315 38391 15363 12123 12420 13455 6316 NA NA NA NA NA NA NA NA NA NA NA NA NA

10000 195065 632609 347009 137445 69395 34419 26140 5771 8026 7029 NA NA NA NA NA NA NA NA NA NA NA NA NA NA
10000 320489 492999 328015 139510 67614 40668 37382 19088 5470 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
10000 256464 533983 315845 133144 69377 12180 13458 6762 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
9000 150895 494938 339410 115690 76872 41168 22150 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
9000 208900 417336 348930 188564 75629 39444 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
9000 215406 513780 391848 146544 44400 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
9000 183445 356127 234208 89427 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
9000 272695 490904 329819 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
9000 152413 437059 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
10000 226426 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

Case 2, log-normal model selected results

The following selected Gelman-Rubin Convergence Statistic values are based on a separate run of three chains. They are based on a run of 500 after a burn-in of 4000.

mu[1]

	---------------------------80% interval		----------------------------		
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.2144	0.1918	0.9186	0.8219	1.118
4101	0.2227	0.2282	0.9543	0.9775	0.9762
4151	0.1905	0.1992	0.816	0.8535	0.956
4201	0.1945	0.1923	0.8335	0.824	1.012
4251	0.22	0.221	0.9424	0.9467	0.9954
4301	0.227	0.2288	0.9725	0.9803	0.992
4351	0.228	0.2334	0.9767	1.0	0.9767
4401	0.2284	0.2291	0.9787	0.9815	0.9971
4451	0.227	0.225	0.9725	0.9641	1.009
4501	0.2152	0.2186	0.9219	0.9365	0.9845

mu[2]

	---------------------------80% interval		----------------------------		
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.1448	0.1197	0.8267	0.6834	1.21
4101	0.153	0.1432	0.8733	0.8171	1.069
4151	0.1668	0.1679	0.952	0.9581	0.9936
4201	0.1539	0.1581	0.8782	0.9024	0.9731
4251	0.1687	0.1716	0.9631	0.9794	0.9834
4301	0.1656	0.1672	0.9453	0.9545	0.9904
4351	0.167	0.1719	0.9533	0.981	0.9717
4401	0.1668	0.1658	0.9518	0.9464	1.006
4451	0.168	0.1721	0.9589	0.9825	0.976
4501	0.1712	0.1752	0.9769	1.0	0.9769

mu[12]

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.2371	0.2341	0.9705	0.9584	1.013
4101	0.2313	0.22	0.947	0.9007	1.051
4151	0.2443	0.2363	1.0	0.9673	1.034
4201	0.2418	0.2348	0.9898	0.9613	1.03
4251	0.227	0.2293	0.9291	0.9389	0.9896
4301	0.2403	0.2319	0.9836	0.9495	1.036
4351	0.2313	0.2206	0.947	0.903	1.049
4401	0.2351	0.2268	0.9624	0.9285	1.037
4451	0.2305	0.2244	0.9435	0.9187	1.027
4501	0.2329	0.2223	0.9536	0.9102	1.048

mu[24]

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.4161	0.3582	0.9805	0.8441	1.162
4101	0.4244	0.3917	1.0	0.9229	1.084
4151	0.3918	0.3892	0.9233	0.9171	1.007
4201	0.407	0.4203	0.959	0.9903	0.9683
4251	0.3838	0.3822	0.9043	0.9005	1.004
4301	0.4073	0.4047	0.9597	0.9536	1.006
4351	0.392	0.388	0.9236	0.9143	1.01
4401	0.3828	0.3832	0.902	0.903	0.9988
4451	0.3863	0.3795	0.9103	0.8942	1.018
4501	0.3896	0.3767	0.9179	0.8877	1.034

tau

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.3237	0.3291	0.9409	0.9567	0.9835
4101	0.3197	0.3213	0.9294	0.934	0.9952
4151	0.3332	0.3263	0.9687	0.9485	1.021
4201	0.344	0.3397	1.0	0.9875	1.013
4251	0.3386	0.3421	0.9844	0.9945	0.9898
4301	0.3395	0.3417	0.9868	0.9934	0.9933
4351	0.3397	0.3382	0.9874	0.983	1.004
4401	0.3375	0.3381	0.9813	0.9828	0.9985
4451	0.3329	0.331	0.9679	0.9622	1.006
4501	0.3332	0.332	0.9686	0.9652	1.004

trend

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.002266	0.001779	0.9682	0.7602	1.273
4101	0.002167	0.002061	0.9258	0.8805	1.051
4151	0.002104	0.002032	0.8991	0.8682	1.036
4201	0.002171	0.002098	0.9278	0.8965	1.035
4251	0.00234	0.002289	1.0	0.978	1.023
4301	0.002306	0.002256	0.9855	0.9642	1.022
4351	0.002287	0.002144	0.9772	0.9162	1.067
4401	0.002203	0.002132	0.9414	0.9112	1.033
4451	0.002145	0.002099	0.9166	0.8968	1.022
4501	0.002149	0.002097	0.9182	0.8962	1.025

season[1]

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.0769	0.06248	0.7567	0.6148	1.231
4101	0.08439	0.07938	0.8305	0.7811	1.063
4151	0.09124	0.09126	0.8979	0.898	0.9998
4201	0.09701	0.0956	0.9546	0.9407	1.015
4251	0.09881	0.09772	0.9723	0.9616	1.011
4301	0.1012	0.09913	0.9957	0.9755	1.021
4351	0.1004	0.09471	0.9878	0.9319	1.06
4401	0.1007	0.09647	0.9905	0.9493	1.043
4451	0.1007	0.09865	0.9905	0.9708	1.02
4501	0.1016	0.09979	1.0	0.9819	1.018

season[2]

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.07251	0.07448	0.5439	0.5587	0.9736
4101	0.08987	0.08843	0.674	0.6632	1.016
4151	0.1258	0.1333	0.9435	1.0	0.9435
4201	0.1238	0.1243	0.9288	0.9323	0.9962
4251	0.1109	0.1101	0.832	0.8257	1.008
4301	0.1232	0.1193	0.924	0.8945	1.033
4351	0.1095	0.1141	0.8214	0.8561	0.9594
4401	0.1049	0.1133	0.7869	0.8497	0.9261
4451	0.1088	0.1089	0.8158	0.8169	0.9987
4501	0.1011	0.1098	0.7586	0.8235	0.9212

season[3]

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.1433	0.09328	0.9796	0.6379	1.536
4101	0.1462	0.1254	1.0	0.8575	1.166
4151	0.1354	0.1293	0.926	0.8839	1.048
4201	0.1374	0.1304	0.9399	0.8915	1.054
4251	0.1352	0.1343	0.9247	0.9181	1.007
4301	0.1288	0.1318	0.8806	0.9016	0.9766
4351	0.1279	0.1277	0.8744	0.8735	1.001
4401	0.1252	0.1266	0.8563	0.8657	0.9891
4451	0.1242	0.1202	0.8491	0.8221	1.033
4501	0.1216	0.1187	0.8317	0.8115	1.025

season[10]

	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
End iteration of bin	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
4051	0.09163	0.0956	0.7243	0.7557	0.9585
4101	0.1078	0.1013	0.8521	0.8004	1.065
4151	0.1049	0.1013	0.8291	0.8006	1.036
4201	0.09308	0.09768	0.7358	0.7721	0.9529
4251	0.1106	0.1126	0.8742	0.8903	0.9819
4301	0.1099	0.1103	0.8689	0.8716	0.9969
4351	0.1148	0.1155	0.9075	0.9128	0.9942
4401	0.1199	0.117	0.948	0.9249	1.025
4451	0.1189	0.1183	0.9396	0.9349	1.005
4501	0.1265	0.1249	1.0	0.9876	1.013

season[11]

End iteration of bin	80% interval: Unnormalized of pooled chains	mean within chain	Normalized as plotted of pooled chains	mean within chain	BGR ratio
4051	0.07366	0.06661	0.7413	0.6704	1.106
4101	0.06886	0.07397	0.6929	0.7444	0.9308
4151	0.07606	0.07651	0.7655	0.77	0.9942
4201	0.08533	0.08149	0.8587	0.8201	1.047
4251	0.09009	0.08849	0.9066	0.8905	1.018
4301	0.09215	0.08786	0.9273	0.8842	1.049
4351	0.09361	0.08894	0.9421	0.8951	1.052
4401	0.09673	0.09293	0.9735	0.9352	1.041
4451	0.09552	0.09059	0.9613	0.9116	1.054
4501	0.09706	0.09937	0.9768	1.0	0.9768

season[12]

End iteration of bin	80% interval: Unnormalized of pooled chains	mean within chain	Normalized as plotted of pooled chains	mean within chain	BGR ratio
4051	0.1084	0.1083	0.9505	0.9498	1.001
4101	0.114	0.1103	1.0	0.9672	1.034
4151	0.1046	0.1043	0.9178	0.915	1.003
4201	0.1032	0.1036	0.9046	0.9088	0.9954
4251	0.1057	0.1047	0.9269	0.9179	1.01
4301	0.1035	0.1045	0.9078	0.9161	0.991
4351	0.1048	0.1041	0.9188	0.9127	1.007
4401	0.1065	0.1052	0.9342	0.9228	1.012
4451	0.1063	0.1048	0.9326	0.9189	1.015
4501	0.1051	0.1036	0.9217	0.909	1.014

The following results are based on 2000 iterations after a burn-in of 4000.

Total reserve statistics:

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**75.0%**	**90.0%**	**95.0%**	**97.5%**
reserve.tot	2.885E+6	502100.0	13160.0	2.122E+6	2.818E+6	3.154E+6	3.529E+6	3.785E+6	4.076E+6

Reserve by incurred month:

node	**mean**	**sd**	**MC error**	**2.5%**	**median**	**75.0%**	**90.0%**	**95.0%**	**97.5%**
reserve[50]	274.6	396.1	9.018	0.8321	1.322	458.5	747.7	992.8	1296.0
reserve[51]	812.8	680.4	14.11	1.848	699.8	1184.0	1695.0	2078.0	2413.0
reserve[52]	1660.0	975.9	20.07	3.019	1537.0	2238.0	2956.0	3361.0	3798.0
reserve[53]	2523.0	1389.0	30.11	397.7	2344.0	3283.0	4340.0	5018.0	5744.0
reserve[54]	4278.0	1815.0	53.44	1430.0	4047.0	5310.0	6631.0	7416.0	8347.0
reserve[55]	5903.0	2043.0	55.16	2606.0	5695.0	7092.0	8591.0	9588.0	10490.0
reserve[56]	5866.0	2105.0	61.12	2508.0	5575.0	7060.0	8512.0	9658.0	10860.0
reserve[57]	8409.0	2514.0	79.45	4088.0	8182.0	9960.0	11650.0	12750.0	14020.0
reserve[58]	10380.0	3007.0	78.8	5173.0	10120.0	12130.0	14250.0	15950.0	16900.0
reserve[59]	10870.0	3199.0	76.38	5584.0	10510.0	12780.0	14910.0	16430.0	18020.0
reserve[60]	12110.0	3325.0	72.87	6453.0	11860.0	14100.0	16450.0	18100.0	19690.0
reserve[61]	15250.0	3821.0	103.3	8829.0	14820.0	17570.0	20090.0	21740.0	23770.0
reserve[62]	21200.0	4450.0	142.4	13410.0	20940.0	24030.0	27010.0	29190.0	31040.0
reserve[63]	33210.0	6739.0	185.5	22200.0	32560.0	37100.0	42250.0	45760.0	48300.0
reserve[64]	43060.0	8326.0	202.6	28670.0	42140.0	47950.0	54310.0	58210.0	61230.0
reserve[65]	48280.0	10570.0	273.3	31790.0	46680.0	54340.0	61480.0	67370.0	73460.0
reserve[66]	61940.0	14040.0	395.8	39940.0	60310.0	69440.0	79580.0	87730.0	95070.0
reserve[67]	89700.0	21270.0	589.7	58120.0	86230.0	102200.0	1.17E+5	127400.0	138400.0
reserve[68]	109300.0	31650.0	786.3	63740.0	104500.0	1.24E+5	149400.0	170100.0	190200.0
reserve[69]	2.0E+5	61790.0	1324.0	112200.0	188300.0	228600.0	283400.0	320100.0	361500.0
reserve[70]	349200.0	112100.0	3204.0	197200.0	328500.0	403300.0	482800.0	562900.0	623300.0
reserve[71]	632500.0	232100.0	5862.0	325400.0	585800.0	740500.0	912700.0	1.06E+6	1.208E+6
reserve[72]	1.218E+6	416600.0	8719.0	631100.0	1.146E+6	1.411E+6	1.731E+6	2.024E+6	2.256E+6

Results of paid.tot[] are as follows. Note that because paid.tot[] includes unknown future claim payments, this is equal to incurred claims.

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%
paid.tot[50]	1.438E+6	396.1	9.018	1.437E+6	1.437E+6	1.438E+6	1.438E+6	1.438E+6	1.439E+6
paid.tot[51]	1.131E+6	680.4	14.11	1.131E+6	1.131E+6	1.132E+6	1.132E+6	1.133E+6	1.133E+6
paid.tot[52]	1.141E+6	975.9	20.07	1.14E+6	1.141E+6	1.142E+6	1.143E+6	1.143E+6	1.143E+6
paid.tot[53]	9.87E+5	1389.0	30.11	984900.0	986800.0	987800.0	988800.0	989500.0	990200.0
paid.tot[54]	1.094E+6	1815.0	53.44	1.092E+6	1.094E+6	1.096E+6	1.097E+6	1.098E+6	1.099E+6
paid.tot[55]	1.039E+6	2043.0	55.16	1.036E+6	1.039E+6	1.04E+6	1.042E+6	1.043E+6	1.044E+6
paid.tot[56]	1.449E+6	2105.0	61.12	1.445E+6	1.448E+6	1.45E+6	1.451E+6	1.452E+6	1.454E+6
paid.tot[57]	1.079E+6	2514.0	79.45	1.075E+6	1.079E+6	1.081E+6	1.082E+6	1.083E+6	1.085E+6
paid.tot[58]	1.211E+6	3007.0	78.8	1.205E+6	1.21E+6	1.212E+6	1.214E+6	1.216E+6	1.217E+6
paid.tot[59]	1.138E+6	3199.0	76.38	1.133E+6	1.138E+6	1.14E+6	1.143E+6	1.144E+6	1.146E+6
paid.tot[61]	980200.0	3821.0	103.3	973800.0	979800.0	982500.0	9.85E+5	986700.0	988700.0
paid.tot[62]	1.289E+6	4450.0	142.4	1.281E+6	1.288E+6	1.291E+6	1.294E+6	1.297E+6	1.298E+6
paid.tot[63]	1.496E+6	6739.0	185.5	1.485E+6	1.495E+6	1.5E+6	1.505E+6	1.509E+6	1.511E+6
paid.tot[64]	1.494E+6	8326.0	202.6	1.48E+6	1.493E+6	1.499E+6	1.506E+6	1.509E+6	1.512E+6
paid.tot[65]	1.389E+6	10570.	273.3	1.373E+6	1.388E+6	1.396E+6	1.403E+6	1.409E+6	1.415E+6
paid.tot[66]	1.303E+6	14040.	395.8	1.281E+6	1.301E+6	1.311E+6	1.321E+6	1.329E+6	1.336E+6
paid.tot[67]	1.369E+6	21270.	589.7	1.337E+6	1.365E+6	1.381E+6	1.396E+6	1.406E+6	1.417E+6
paid.tot[68]	1.421E+6	31650.	786.3	1.376E+6	1.416E+6	1.436E+6	1.461E+6	1.482E+6	1.502E+6
paid.tot[69]	1.063E+6	61790.	1324.0	975400.0	1.052E+6	1.092E+6	1.147E+6	1.183E+6	1.225E+6
paid.tot[70]	1.443E+6	112100.	3204.0	1.291E+6	1.422E+6	1.497E+6	1.576E+6	1.656E+6	1.717E+6
paid.tot[71]	1.222E+6	232100.	5862.0	914900.0	1.175E+6	1.33E+6	1.502E+6	1.649E+6	1.798E+6
paid.tot[72]	1.445E+6	416600.	8719.0	857500.0	1.372E+6	1.637E+6	1.957E+6	2.251E+6	2.482E+6

mu[j]:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%	start	sample
mu[1]	11.75	0.07115	0.003954	11.61	11.75	11.8	11.84	11.86	11.88	4001	2000
mu[2]	12.5	0.07757	0.004544	12.36	12.5	12.56	12.61	12.63	12.67	4001	2000
mu[3]	12.11	0.07863	0.003979	11.96	12.11	12.16	12.22	12.24	12.26	4001	2000
mu[4]	11.19	0.0757	0.003844	11.04	11.18	11.24	11.29	11.32	11.34	4001	2000
mu[5]	10.58	0.0772	0.004552	10.43	10.59	10.63	10.68	10.7	10.74	4001	2000
mu[6]	9.91	0.07739	0.004481	9.752	9.909	9.958	10.01	10.04	10.06	4001	2000
mu[7]	9.334	0.08136	0.006255	9.182	9.332	9.388	9.442	9.47	9.493	4001	2000
mu[8]	8.872	0.08141	0.005261	8.711	8.878	8.927	8.966	9.016	9.036	4001	2000
mu[9]	8.504	0.08391	0.005351	8.338	8.508	8.561	8.611	8.642	8.672	4001	2000
mu[10]	8.235	0.07768	0.003746	8.084	8.235	8.281	8.334	8.382	8.403	4001	2000
mu[11]	7.973	0.08759	0.004794	7.813	7.97	8.035	8.088	8.115	8.145	4001	2000
mu[12]	7.45	0.07569	0.005205	7.309	7.448	7.499	7.552	7.576	7.606	4001	2000
mu[13]	7.04	0.08247	0.006124	6.876	7.04	7.098	7.142	7.165	7.191	4001	2000
mu[14]	6.949	0.08619	0.004856	6.786	6.951	7.004	7.05	7.09	7.129	4001	2000
mu[15]	7.087	0.08873	0.005996	6.918	7.081	7.149	7.201	7.231	7.263	4001	2000
mu[16]	6.886	0.09755	0.007234	6.705	6.884	6.952	7.017	7.044	7.083	4001	2000
mu[17]	7.029	0.09771	0.005464	6.834	7.026	7.095	7.151	7.19	7.217	4001	2000
mu[18]	6.752	0.08318	0.006209	6.577	6.751	6.808	6.861	6.886	6.917	4001	2000
mu[19]	6.639	0.09351	0.006207	6.426	6.644	6.704	6.753	6.79	6.807	4001	2000
mu[20]	6.67	0.1075	0.007286	6.461	6.668	6.744	6.804	6.846	6.868	4001	2000
mu[21]	6.694	0.1043	0.005316	6.488	6.694	6.759	6.827	6.872	6.922	4001	2000
mu[22]	6.212	0.1153	0.007283	5.988	6.218	6.293	6.359	6.397	6.441	4001	2000
mu[23]	6.108	0.1237	0.007432	5.858	6.105	6.182	6.281	6.314	6.342	4001	2000
mu[24]	5.862	0.1484	0.009259	5.585	5.864	5.964	6.061	6.119	6.167	4001	2000

Tau:

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%	start	sample
tau	3.2	0.1284	0.003158	2.943	3.201	3.289	3.364	3.407	3.455	4001	2000

For the percentage of ‘0’ cells, p[i,j], note the heavily skewed results for shorter lags.

node	mean	sd	MC error	2.5%	median	75.0%	90.0%	95.0%	97.5%	start	sample
p[1,1]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[1,2]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[1,3]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[1,4]	2.427E-5	0.001085	2.439E-5	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[1,5]	3.761E-5	0.00153	3.431E-5	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[1,6]	4.31E-4	0.005793	2.438E-4	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000

p[1,7]	8.622E-4	0.006517	2.998E-4	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[1,8]	0.001547	0.009637	3.179E-4	0.0	0.0	0.0	0.0	0.0	0.02669	4001	2000
p[1,9]	0.005209	0.01907	8.165E-4	0.0	0.0	0.0	0.005889	0.04208	0.07461	4001	2000
p[1,10]	0.008414	0.024	0.00107	0.0	0.0	0.0	0.03361	0.06998	0.09377	4001	2000
p[1,11]	0.01056	0.02893	0.001219	0.0	0.0	0.0	0.04031	0.07632	0.1116	4001	2000
p[1,12]	0.1965	0.03414	0.002053	0.1366	0.1934	0.217	0.2439	0.2575	0.2736	4001	2000
p[1,13]	0.1178	0.04011	0.001894	0.0524	0.114	0.1432	0.1729	0.1851	0.2019	4001	2000
p[1,14]	0.04003	0.04828	0.002451	0.0	0.01702	0.07213	0.1114	0.1331	0.1544	4001	2000
p[1,15]	0.3014	0.04372	0.00241	0.2233	0.3001	0.327	0.3594	0.3775	0.3938	4001	2000
p[1,16]	0.3638	0.04018	0.002476	0.2884	0.3628	0.3892	0.4174	0.4312	0.4461	4001	2000
p[1,17]	0.3507	0.05163	0.00289	0.2572	0.3491	0.3854	0.415	0.4355	0.4608	4001	2000
p[1,18]	0.3021	0.04352	0.001953	0.2277	0.2955	0.3292	0.3621	0.3796	0.4043	4001	2000
p[1,19]	0.2946	0.04731	0.002457	0.2112	0.2924	0.3254	0.3565	0.374	0.3994	4001	2000
p[1,20]	0.364	0.04833	0.002889	0.281	0.3613	0.3974	0.4267	0.4489	0.4643	4001	2000
p[1,21]	0.4406	0.05538	0.003087	0.3338	0.4395	0.4767	0.5141	0.5313	0.5525	4001	2000
p[1,22]	0.5638	0.05597	0.002225	0.4486	0.5667	0.6019	0.6325	0.6508	0.6682	4001	2000
p[1,23]	0.6102	0.05918	0.002544	0.4888	0.6105	0.6523	0.6812	0.7069	0.7283	4001	2000
p[1,24]	0.7181	0.06415	0.003191	0.5866	0.721	0.7629	0.797	0.8216	0.8394	4001	2000
p[72,1]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,2]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,3]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,4]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,5]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,6]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,7]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,8]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,9]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,10]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,11]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,12]	5.564E-5	0.001142	2.411E-5	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,13]	5.875E-5	0.001924	5.773E-5	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,14]	0.0	0.0	2.236E-12	0.0	0.0	0.0	0.0	0.0	0.0	4001	2000
p[72,15]	0.02332	0.03358	0.001696	0.0	0.00437	0.03937	0.07164	0.09146	0.1139	4001	2000
p[72,16]	0.07192	0.04393	0.002361	0.0	0.06796	0.09915	0.1318	0.1507	0.1655	4001	2000
p[72,17]	0.06663	0.05379	0.002896	0.0	0.0597	0.1038	0.1397	0.161	0.1845	4001	2000
p[72,18]	0.01693	0.02885	0.00108	0.0	0.0	0.02529	0.06028	0.08091	0.09897	4001	2000
p[72,19]	0.01297	0.0258	0.001303	0.0	0.0	0.01431	0.04809	0.07005	0.09334	4001	2000
p[72,20]	0.06447	0.05208	0.002938	0.0	0.05671	0.09743	0.1383	0.1628	0.1826	4001	2000
p[72,21]	0.1519	0.06339	0.003187	0.02978	0.1526	0.1954	0.2348	0.2544	0.2779	4001	2000
p[72,22]	0.2721	0.0671	0.002594	0.136	0.2739	0.3178	0.3549	0.3793	0.4027	4001	2000
p[72,23]	0.3215	0.07127	0.003017	0.1697	0.3255	0.3721	0.4085	0.428	0.4579	4001	2000
p[72,24]	0.4362	0.07626	0.00376	0.2834	0.4386	0.4882	0.5315	0.5576	0.5788	4001	2000

Case 3, selected statistics.

node	mean	sd	sample
p[1,1]	0.000	0.000	10000
p[1,2]	0.000	0.000	10000
p[1,3]	0.000	0.000	10000
p[1,4]	0.000	0.001	10000
p[1,5]	0.000	0.004	10000
p[1,6]	0.002	0.008	10000
p[1,7]	0.004	0.012	10000
p[1,8]	0.015	0.022	10000
p[1,9]	0.082	0.021	10000
p[1,10]	0.080	0.024	10000
p[1,11]	0.158	0.028	10000
p[1,12]	0.357	0.043	10000
p[30,1]	0.000	0.000	10000
p[30,2]	0.000	0.000	10000
p[30,3]	0.000	0.000	10000
p[30,4]	0.000	0.000	10000
p[30,5]	0.000	0.002	10000
p[30,6]	0.000	0.003	10000
p[30,7]	0.001	0.004	10000
p[30,8]	0.004	0.010	10000
p[30,9]	0.046	0.018	10000

p[30,10]	0.043	0.022	10000
p[30,11]	0.122	0.024	10000
p[30,12]	0.321	0.040	10000
p[60,1]	0.000	0.000	10000
p[60,2]	0.000	0.000	10000
p[60,3]	0.000	0.000	10000
p[60,4]	0.000	0.000	10000
p[60,5]	0.000	0.000	10000
p[60,6]	0.000	0.000	10000
p[60,7]	0.000	0.000	10000
p[60,8]	0.000	0.001	10000
p[60,9]	0.001	0.006	10000
p[60,10]	0.002	0.006	10000
p[60,11]	0.045	0.021	10000
p[60,12]	0.244	0.038	10000

node	**mean**	**sd**	**sample**
x	1.175	0.236	10000

node	**mean**	**sd**	**sample**
y	0.159	0.034	10000

node	**mean**	**sd**	**sample**
z[1]	-0.003	0.072	10000
z[2]	0.001	0.072	10000
z[3]	0.000	0.071	10000
z[4]	0.001	0.072	10000
z[5]	-0.004	0.069	10000
z[6]	-0.007	0.068	10000
z[7]	-0.025	0.058	10000
z[8]	-0.044	0.055	10000
z[9]	0.027	0.038	10000
z[10]	-0.006	0.036	10000
z[11]	0.007	0.034	10000
z[12]	0.057	0.048	10000

node	**mean**	**sd**	**sample**
paid.pool.tot[50]	14370000.000	1496.000	10000
paid.pool.tot[51]	15470000.000	4325.000	10000
paid.pool.tot[52]	14910000.000	7387.000	10000
paid.pool.tot[53]	12170000.000	9496.000	10000
paid.pool.tot[54]	14130000.000	12490.000	10000
paid.pool.tot[55]	14990000.000	18810.000	10000
paid.pool.tot[56]	14430000.000	27480.000	10000
paid.pool.tot[57]	13260000.000	42010.000	10000
paid.pool.tot[58]	15020000.000	87410.000	10000
paid.pool.tot[59]	13480000.000	240600.000	10000
paid.pool.tot[60]	11820000.000	925500.000	10000

node	**mean**	**sd**	**sample**
paid[50,12]	642.400	551.700	10000
paid[51,11]	1952.000	1182.000	10000
paid[51,12]	778.900	612.400	10000
paid[52,10]	2903.000	1788.000	10000
paid[52,11]	1904.000	1179.000	10000
paid[52,12]	752.300	590.200	10000
paid[53,9]	2914.000	1934.000	10000
paid[53,10]	2380.000	1567.000	10000
paid[53,11]	1607.000	1149.000	10000
paid[53,12]	619.900	549.900	10000
paid[54,8]	3499.000	2342.000	10000
paid[54,9]	2978.000	1971.000	10000
paid[54,10]	2425.000	1608.000	10000
paid[54,11]	1595.000	1113.000	10000
paid[54,12]	639.600	568.000	10000
paid[55,7]	6002.000	3647.000	10000

paid[55,8]	4045.000	2431.000	10000
paid[55,9]	3499.000	2138.000	10000
paid[55,10]	2886.000	1771.000	10000
paid[55,11]	1908.000	1211.000	10000
paid[55,12]	757.100	595.400	10000
paid[56,6]	9269.000	5781.000	10000
paid[56,7]	5974.000	3594.000	10000
paid[56,8]	4085.000	2564.000	10000
paid[56,9]	3486.000	2205.000	10000
paid[56,10]	2845.000	1793.000	10000
paid[56,11]	1895.000	1208.000	10000
paid[56,12]	749.200	595.000	10000
paid[57,5]	14260.000	9652.000	10000
paid[57,6]	7812.000	5379.000	10000
paid[57,7]	5039.000	3468.000	10000
paid[57,8]	3470.000	2375.000	10000
paid[57,9]	2963.000	2020.000	10000
paid[57,10]	2466.000	1679.000	10000
paid[57,11]	1591.000	1105.000	10000
paid[57,12]	640.500	554.600	10000
paid[58,4]	34200.000	22240.000	10000
paid[58,5]	16100.000	10260.000	10000
paid[58,6]	8915.000	5661.000	10000
paid[58,7]	5735.000	3719.000	10000
paid[58,8]	3852.000	2477.000	10000
paid[58,9]	3332.000	2192.000	10000
paid[58,10]	2748.000	1753.000	10000
paid[58,11]	1805.000	1190.000	10000
paid[58,12]	708.300	585.600	10000
paid[59,3]	99900.000	65860.000	10000
paid[59,4]	32100.000	21460.000	10000
paid[59,5]	15100.000	10080.000	10000
paid[59,6]	8250.000	5553.000	10000
paid[59,7]	5388.000	3679.000	10000
paid[59,8]	3664.000	2428.000	10000
paid[59,9]	3127.000	2096.000	10000
paid[59,10]	2549.000	1651.000	10000
paid[59,11]	1702.000	1177.000	10000
paid[59,12]	669.700	577.100	10000
paid[60,2]	407400.000	281300.000	10000
paid[60,3]	91840.000	63950.000	10000
paid[60,4]	29970.000	21190.000	10000
paid[60,5]	13850.000	9503.000	10000
paid[60,6]	7784.000	5569.000	10000
paid[60,7]	4970.000	3388.000	10000
paid[60,8]	3362.000	2291.000	10000
paid[60,9]	2878.000	2027.000	10000
paid[60,10]	2370.000	1645.000	10000
paid[60,11]	1600.000	1164.000	10000
paid[60,12]	616.400	550.900	10000

References

Broffit, J.D. 1988. "Increasing and Increasing Convex Bayesian Graduation." *Transactions of the Society of Actuaries* 40(1):115-48.

Bühlmann, H. 1970. *Mathematical Methods in Risk Theory.* Berlin: Springer-Verlag.

Carlin, B.P. 1992a. "Analyzing Nonlinear and Non-Gaussian Actuarial Time Series." *Actuarial Research Clearing House* 1992 (1):27-60.

Carlin, B.P. 1992b. "A Simple Monte Carlo Approach to Bayesian Graduation." *Transactions of the Society of Actuaries* 44:55-76.

Congdon, Peter 2001. *Bayesian Statistical Modeling.* Chichester: John Wiley & Sons Ltd.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. 1995. *Bayesian Data Analysis.* London: Chapman and Hall.

Jones, D.A. 1965. "Bayesian Statistics." *Transactions of the Society of Actuaries* 17:33-57.

Kimeldorf, G.S., and Jones, D.A. 1967. "Bayesian Graduation." *Transactions of the Society of Actuaries* 19.

Morgan, I.M., and Hickman, J.C. 1993. "Conjugate Bayesian Analysis of the Negative Binomial Distribution." *Actuarial Research Clearing House* 1993 (1):97-113.

Neter, John, Kutner, M. H., Nachtsheim, C. J., and Wasserman, William 1996. *Applied Linear Statistical Models.* WCB/McGraw-Hill.

Ntzoufras, I., and Dellaportas, P. 2002. "Bayesian Modeling of Outstanding Liabilities Incorporating Claim Count Uncertainty," *North American Actuarial Journal* 6(1): 113-128.

Rosenberg, M., and Young, V.R. 1999. "A Bayesian Approach to Understanding Time Series Data." *North American Actuarial Journal* 3 (2):130-43.

Scollnik, D.P.M. 1995a. "Simulating Random Variates from Makeham's Distribution and form Others with Exact or Nearly Log-Concave Densities." *Transactions of the Society of Actuaries* 47:409-37.

Scollnik, D.P.M. 1995b. The Bayesian Analysis of Generalized Poisson Models for Claim Frequency Data Utilising Markov Chain Monte Carlo Methods." *Actuarial Research Clearing House* 1995 (1):339-56.

Scollnik, D.P.M. 1996. "An Introduction to Markov Chain Monte Carlo Methods and Their Actuarial Applications." *Proceedings of the Casualty Actuarial Society* 83:114-65.

Scollnik, D.P.M. 2000. "Actuarial Modeling with MCMC and BUGS: Additional Worked Examples." Working Paper.

Scollnik, D.P.M. 2001. "Actuarial Modeling with MCMC and BUGS." *North American Actuarial Journal* 5(2): 96-124.

Scollnik, D.P.M. 2002. "Implementation of Four Models for Outstanding Liabilities in WinBUGS: A Discussion of a Paper by Ntzoufras and Dellaportas." *North American Actuarial Journal* 6(1): 128-136.

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

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

Spiegelhalter, D.J., Thomas, A., and Best, N.G., 2000. *WinBUGS Version 1.3 User Manual.* Cambridge, U.K.: MRC Biostatistics Unit.

Spiegelhalter, D.J., Best, N.G., Carlin, B.P., and van der Linde, A. 2002. "Bayesian Measures of Model Complexity and Fit." *Journal of the Royal Statistical Society* B, 64, Part 3, 1-34.