Severity Curve Fitting for Long-Tailed Lines: An Application of Stochastic Processes and Bayesian Models

by Greg McNulty

ABSTRACT

I present evidence for a model in which parameters fit to the severity distribution at each report age follow a smooth curve with random error. More formally, this is a stochastic process, and it allows us to estimate parameters of the ultimate severity distribution. I detail a Bayesian hierarchical model that takes a modestly sized dataset of triangulated individual claim data and returns posterior distributions for the parameters of the ultimate severity distribution, trend and loss to an excess layer. Currently available methods are also discussed. Full code and data are provided in the appendices.

KEYWORDS

Bayesian hierarchical models, stochastic processes, severity distributions, reinsurance pricing

1. Introduction

Severity distributions are very important in actuarial science. They allow for the calculation of rating parameters such as increased limits factors and reinsurance costs. Fitting parametric curves to severity distributions is highly desirable for smoothing the data and adequately modeling tail losses. A difficult problem faced by actuaries is how to produce parametric or semi-parametric severity distributions representing ultimate settlement values for long-tailed lines of business where losses develop slowly over time.

For lines of business in which all claims and their ultimate values are known soon after occurrence, fitting a curve to the individual claim amounts is a straightforward exercise. For long-tailed lines, however, any finite set of individual claim data is biased due to the tendency for individual claim amounts to increase as the time since claim occurrence increases. Also, some claims are not reported until years after they occur, so the actuary's dataset may not be complete, nor will they know if it is complete.

In this paper I will present evidence that a parameter fit to the severity distribution for each accident year and age will appear to follow a stochastic process. Then I detail a Bayesian hierarchical model which reasonably reproduces the theoretical true values for the critical parameters of the stochastic process when given a modestly sized simulated dataset. These critical values will give us estimates of trend, the ultimate severity distribution and excess layer losses. Finally, I will contrast with current methods.

2. Stochastic severity distribution parameters

2.1. Empirical evidence for the model

The method I will present in this paper was inspired by patterns found in actual data. Figure 1 shows the μ parameter of a lognormal distribution fit for each accident year and evaluation age from a grouped dataset of incurred amounts from commercial general liability claims. The number of claims in that dataset is on the order of hundreds of thousands. Lognormal was the best fit among about a dozen parametric distributions of one to four parameters. Each line is a single accident year.

Figure 2 shows the μ parameters of a lognormal distribution fit to each accident year and age of a ground up dataset of approximately three thousand individual professional liability claims.

The remainder of this paper will show how we can build a model around this pattern we see in the data. The purpose of the model is to solve the problem stated in the introduction: how to produce parametric

Figure 1. Fitted lognormal μ by AY and age from large dataset of grouped liability claims
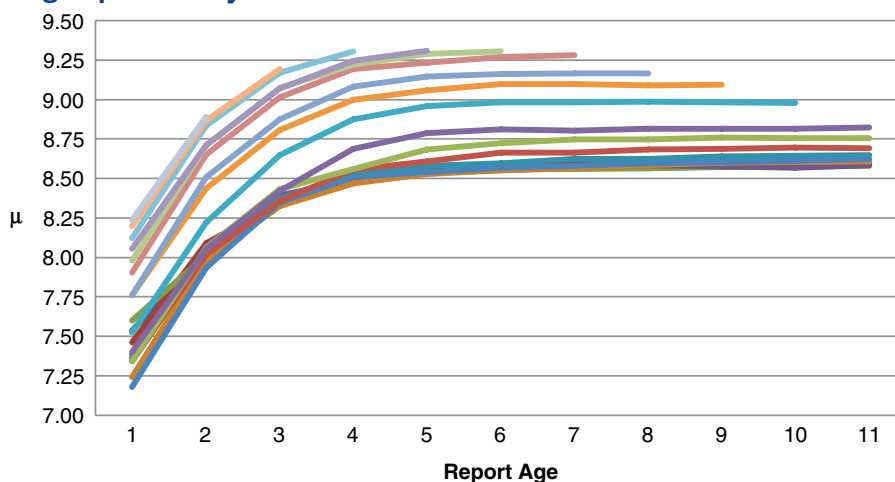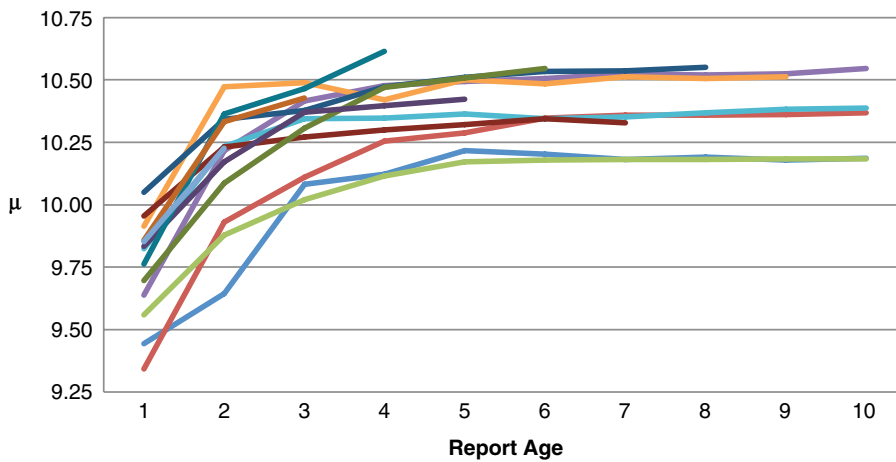
Figure 2. Fitted lognormal μ by AY and age from dataset of professional liability claims

or semi-parametric severity distributions representing ultimate settlement values for long-tailed lines of business where losses develop slowly over time.

2.2. Initial data and the basic model

The goal is to produce a parametric severity distribution representing sizes of loss for claims occurring in the prospective accident period. I'm assuming the user is starting with a dataset as shown in Table 1 with annual evaluations of individual claim amounts from a set of accident years.

As in the real-life examples of Section 2.1, suppose we observe individual claim incurred loss amounts for a given accident year for ten years. At the end of each year we record the total incurred amount for every claim, whether open or closed, and then fit a parametric distribution, e.g., log-normal, to the claim amounts. This gives us ten μ parameters and ten σ parameters. Suppose we observe the pattern

in Figure 3: each parameter appears to be following a smooth curve. By extrapolating the curves beyond the observation period, say to infinity, we will have an estimate for the distribution of ultimate settlement amounts.

These points can be fitted by exponential decay in the first differences, e.g., following the formulas:

$$\mu(t) - \mu(t-1) = \delta_\mu e^{-(t-1)/\theta_1}$$

$$\sigma(t) - \sigma(t-1) = \delta_\sigma e^{-(t-1)/\theta_2}.$$

Applying the geometric series formula to both sides, we have, for example:

$$\sum_1^s \mu(t) - \mu(t-1) = \mu(s) - \mu(0)$$

$$\sum_1^s \delta_\mu e^{-(t-1)/\theta_1} = \delta_\mu \frac{1 - e^{-s/\theta_1}}{1 - e^{-1/\theta_1}} = \frac{\delta_\mu}{1 - e^{-1/\theta_1}} \left(1 - e^{-s/\theta_1}\right).$$

Defining

$$\Delta_\mu = \frac{\delta_\mu}{1 - e^{-1/\theta_1}}$$

as a closed-form solution to the above recursive relations, it can then be written as

$$\mu(t) = \mu(0) + \Delta_\mu \left(1 - e^{-t/\theta_1}\right)$$

$$\sigma(t) = \sigma(0) + \Delta_\sigma \left(1 - e^{-t/\theta_2}\right).$$

Table 1. Assumed format of available data

Claim	Accident Year	Age 1 Incurred	Age 2 Incurred	Age 3 Incurred
1	1	500	1000	1100
2	2	100	500	
3	2	100	300	
4	3	200		

Figure 3. Smooth curve of severity distribution parameter by report age
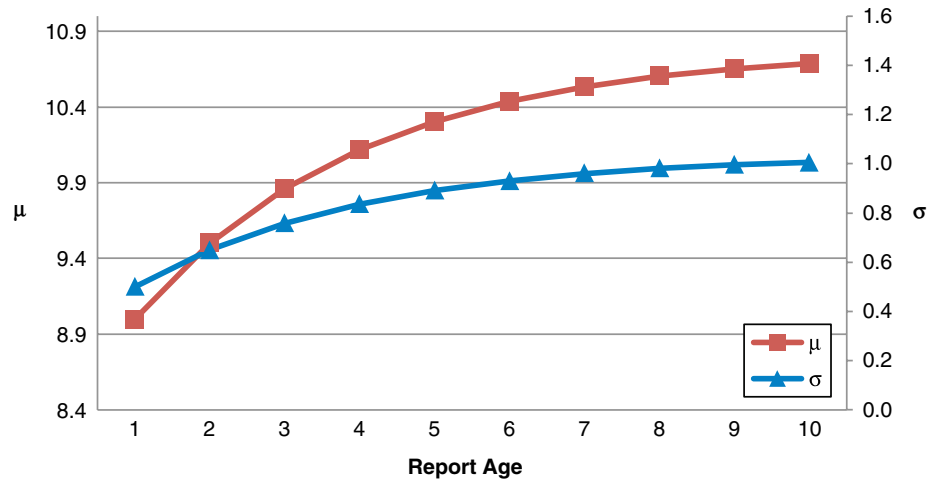
Note that the Δ terms depend on θ and δ but not t. Now with these formulas we can determine the parameters of the ultimate severity distribution:

$$\lim_{t \to \infty} \mu(t) = \mu(0) + \Delta_\mu$$

$$\lim_{t \to \infty} \sigma(t) = \sigma(0) + \Delta_\sigma.$$

This would give us an estimate of the ultimate average log-severity. We could also use these parameters to determine the amount of loss that would ultimately be in an excess layer or calculate ILFs.

2.3. Stochastic processes

A perfectly smooth curve is obviously an idealized model of the loss process and in reality will be subject to multiple layers of parameter risk. One risk is the potential for the actual severity distribution parameters to stray from the assumed deterministic path. The modern mathematical framework for random variables indexed by time like this is stochastic processes.

Suppose we observe multiple accident periods and see the pattern in the fitted log-normal mean parameters as shown in Figure 4, which more closely resembles the real data.

Figure 4. Smooth curve of severity distribution parameter μ with stochastic error
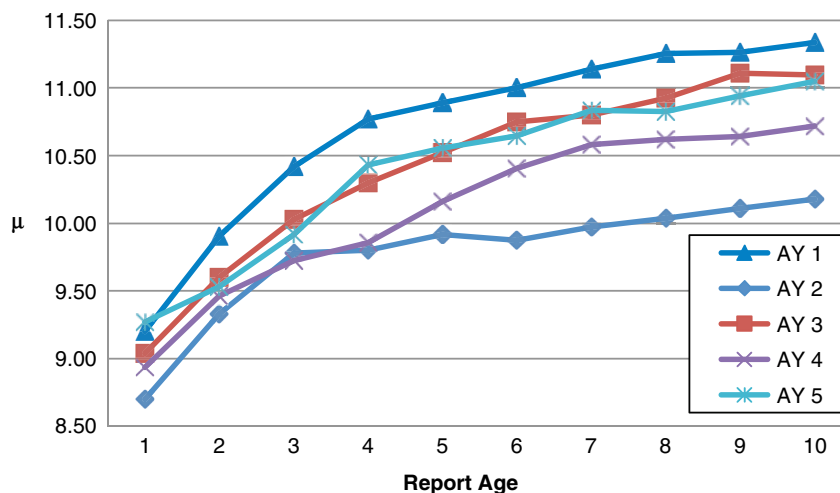
The μ parameter still appears to follow a smooth curve, but with a random error component between each reporting age. Also note that the paths appear to follow a moving average trend with drift, meaning each point is influenced by the prior points on the same graph, not the overall average among all paths. If we wanted to be more formal we could write the difference equation of the severity distribution parameters as

$$\mu(t) - \mu(t-1) = \delta_\mu e^{-(t-1)/\theta_1} + \epsilon_\mu(t)$$

$$\sigma(t) - \sigma(t-1) = \delta_\sigma e^{-(t-1)/\theta_2} + \epsilon_\sigma(t).$$

With error variances that decay in time,

$$Var\left(\epsilon_\mu(t)\right) = v_\mu e^{(-t/\eta_1)}$$

$$Var\left(\epsilon_\sigma(t)\right) = v_\sigma e^{(-t/\eta_2)}.$$

From here we could compute the ultimate severity distribution by taking limits of sums and also their variances by using stochastic calculus (beyond the scope of this paper).

2.4. Caveat on real-life sigma parameters

In the interest of full disclosure, the charts of the σ parameters for the real datasets in Figures 1 and 2 also follow what look like a stochastic process around a smooth curve, but the smooth curve is not as simple as the exponential decay model developed above. They look more like a bell curve with a local maximum within the first several years. For simplicity I will assume it does follow the exponential decay model, but if you look at real data, expect to see a more complicated pattern. The model developed in this paper can be expanded to handle other types of smooth curves. I would just urge the reader to visually inspect graphs of their data and fitted parameters to make sure your model is reasonable.

2.5. Trend, the link between accident years

So far we have seen multiple accident years together in Figures 1, 2, and 4 but we have only given formulas for a single accident year. A complete model must incorporate the data from multiple accident years simultaneously. This can be done by assuming all accident years follow different realizations of the same stochastic process, as in Figure 4. Additionally, we can incorporate the traditional assumption that losses in each successive accident year will be higher than the previous by a fixed rate of inflation, or trend.

Using the notation $\mu(i, j)$ where the first variable is the accident year and the second is the report age, we can incorporate trend very simply as follows:

$$\mu(i, 1) = \mu(i-1, 1) + trend.$$

Equivalently, if we define μ to be a fixed number representing a baseline value for μ at accident year 0 and age 1, then we can write

$$\mu(i, 1) = \mu + trend * i.$$

In our final model below, we will also add a random error term to each $\mu(i, 1)$.

3. Application of Bayesian statistics

3.1. Extension to the small or medium dataset case

Even with a large volume of data, there will always be a range of possible estimates for the parameters of the smooth curve, the error variances, and all the other parameters of the stochastic process. This is the parameter risk of the stochastic model. How can we confidently use this model when we don't have a very large dataset?

Instead of taking our small- to medium-sized dataset and trying to fit a model from scratch, we can start with an a priori set of possible models and use the data to modify the probability that each model is correct. The modern tool for this type of statistical analysis is a Bayesian hierarchical model. An excellent reference is Zhang, Dukic, and Guszcza (2012).

To put our stochastic process in the framework of a Bayesian hierarchical model, instead of simply saying the accident year inflation rate of the μ parameter is

5% per year, we specify that it is normally distributed with mean 5% and standard deviation 1%, or that the exponential decay of the first differences in the σ parameter isn't 30%, but normally distributed with mean 30% and standard deviation 5%. These are our prior distributions. Then, given a dataset of individual claim amounts indexed by accident year and report age, we can calculate the Bayesian posterior distributions of all our severity distribution stochastic process parameters.

3.2. Bayesian hierarchical model of stochastic severity development

I created an implementation of this model in Stan, a free and open-source R package for building Bayesian models and calculating posterior estimates. The full code of the model can be found in Appendix A. For readers who are not familiar with Bayesian modeling and would like more background, an excellent reference is Scollnik (2001).

I assume that claims in each accident year and age are log-normally distributed with the μ and σ parameters following discrete time stochastic processes defined by the following formulas.

$$\mu(i,1) = \mu_{start} + trend * i + \epsilon_{\mu}(i,1)$$

$$\sigma(i,1) = \sigma_{start} + \epsilon_{\sigma}(i,1)$$

$$\mu(i,j) = \mu(i,j-1) + \mu_{incr} * e^{-(j-1)/\mu_{growth}} + \epsilon_u(i,j)$$

$$\sigma(i,j) = \sigma(i,j-1) + \sigma_{incr} * e^{-(j-1)/\sigma_{growth}} + \epsilon_{\sigma}(i,j)$$

$$\epsilon_{\mu}(i,j) \sim N\left(0, p_{\mu} * e^{-(j-1)*r_{\mu}}\right)$$

$$\epsilon_{\sigma}(i,j) \sim N\left(0, p_{\sigma} * e^{-(j-1)*r_{\sigma}}\right).$$

Explanations of each variable are given in Table 2.

The input data, shown in Appendix B, represents 20 claim amount observations per accident year and report year from the upper half of a 10 by 10 accident year/report year triangle. There will be 10 report ages observed from the oldest accident year and only one

Table 2. Variable definitions

Parameter	Definition
mu_start	The theoretical value of μ at report time 1 without process risk
mu_incr	The amount μ is expected to increase between report ages, subject to decay
mu_growth	Rate of decay in the amount μ is expected to increase between report ages
sigma_start	The theoretical value of σ at report time 1 without process risk
sigma_incr	The amount σ is expected to increase between report ages, subject to decay
sigma_growth	Rate of decay in the amount σ is expected to increase between report ages
trend	Expected increase in μ at age 1 between successive accident years
p_mu	Standard deviation of error term for μ in age 1
r_mu	Rate of decay in variance term for μ between successive report ages
p_sigma	Standard deviation of error term for σ in age 1
r_sigma	Rate of decay in variance term for σ between successive report ages

from the most recent. The result is 1,100 data points. The data is simulated randomly from a process identical to the model so that we know the answer and can compare the Bayesian posterior estimates with the true theoretical values.

I have assumed mostly uninformative uniform distributions over wide ranges relative to the true parameter values for purposes of this example. In a real application, the true benefit of this model is the ability to use an informative prior. For example, if 10 similar datasets have been analyzed and μ_{growth} appears to be normally distributed with mean 5 and standard deviation 2, then that would be a reasonable prior to use in analyzing the 11th dataset. Even for a brand new dataset, there is likely some type of expectation by the analyst of the range of parameters and the prior is how those expectations enter the model.

In this example, the priors for μ_{start} and σ_{start} are very specific, however. These variables represent the parameters of the distribution of claims at the first reporting age. Since these claim amounts are immediately available, this is a classical fitting exercise. I assume the variance between our estimate and the

true values (9 for μ_{start} and 0.5 for σ_{start}) is given by Fisher's information; see Example 15.9 in Klugman, Panjer, and Wilmot (2008).

Near the end of the code I define the outputs of interest, μ_{ult} and σ_{ult}. These are calculated by extrapolating the smooth curve out to report time infinity. We are not limited to the estimate for $\mu(i, 10)$ or $\sigma(i, 10)$.

Finally, we calculate the expected layer loss and frequency to a $1M excess $1M layer for the next accident year. The output is an approximation to the Bayesian posterior distribution of the expected layer loss. Note that this is per claim since the model does not have a frequency component.

3.3. Model output and performance

Table 3 summarizes the model output and compares the posterior estimates to the true values used to simulate the data.

Table 3 shows the model coming within approximately one standard deviation of the true value for the critical output parameters of μ_{ult}, σ_{ult} and trend, but the posterior means differ somewhat from the true values. The point is not that the model can exactly reproduce the true parameter values but that in the case of a small- to medium-sized dataset the model is able to produce a reasonable estimate of the stochastic process parameters, and also give an estimate of the uncertainty around those estimates. The posterior

Table 3. Bayesian model output

Parameter	Model Mean	Model Std. Dev.	True Value
mu_start	9.00	0.03	9
mu_incr	0.75	0.15	0.79
mu_growth	3.89	1.17	3
mu_ult	11.40	0.39	11
sigma_start	0.52	0.02	0.5
sigma_incr	0.43	0.75	0.20
sigma_growth	2.36	1.57	3
sigma_ult	1.08	0.15	1
trend	0.03	0.02	0.05
layer_loss	19,969	43,592	*
layer_freq	0.04	0.06	*

means will be closer if either we have more data or supply informative priors.

Since μ_{ult} and σ_{ult} are stochastic, they do not have a true value but rather a true mean. For any combination of these variables, we will get a different layer loss and frequency which is why the "true value" is ambiguous. Parameter variance in the model for ground up losses becomes leveraged and adds to the expected layer loss and frequency; therefore, the expected value depends on which types of parameter risk are considered to be contributors. As defined in the code, layer_loss and layer_freq depend only on μ_{ult}, σ_{ult} and the trend and the model means reflect the uncertainty in those parameters. One could argue that they should also depend on the process variance of a generic accident year, introducing p_{μ} and p_{σ} into the equation.

Figures 5 through 8 show histograms of the posterior distribution of the critical output variables.

4. Survey of current methods

4.1. ISO: Fitting by closure age

ISO provides ultimate severity distributions for many long-tailed liability lines of business. Their method of accounting for severity development is to fit a distribution to claims closed at each reporting age, then weight those curves together by the proportion of claims expected to be settled in each age. Once a claim is closed, then the loss amount is fixed, so there is no bias in the fit to the data for a given closure period.

In the later ages the data thins out and there may be a problem fitting credible curves. ISO's solution is to group all closed claims beyond a certain age together and give them a single severity distribution. Industry excess loss triangles would indicate that this assumption is not accurate, but it may be immaterial for the volume of data and number of years ISO has available. For an application with less data or a shorter history, it may be necessary to make explicit adjustment for development beyond the reporting ages observed.

Figure 5. Histogram of posterior mu_ult
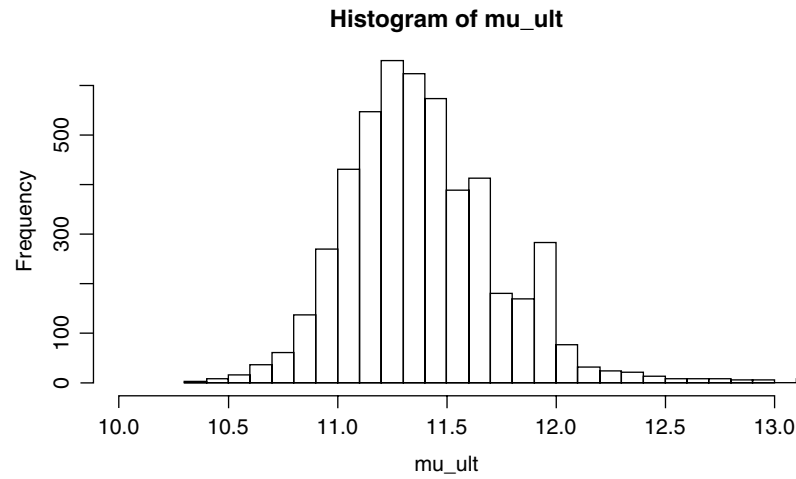
Histogram of mu_ult

Figure 6. Histogram of posterior sigma_ult

Histogram of sigma_ult
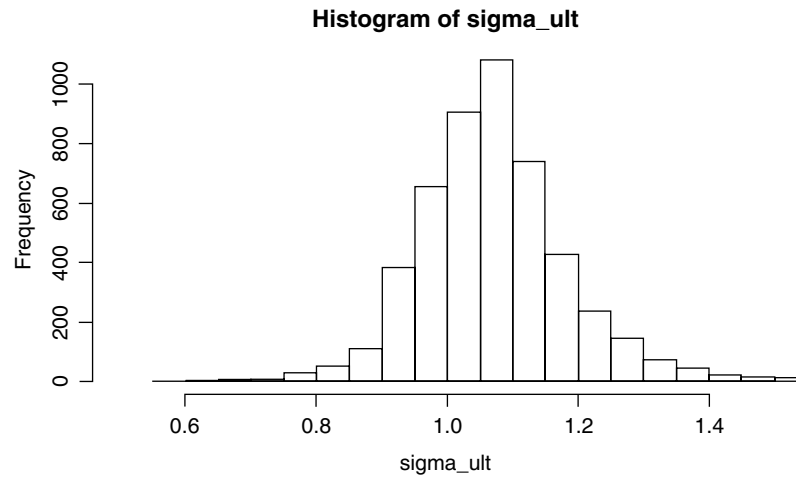
Figure 7. Histogram of posterior trend
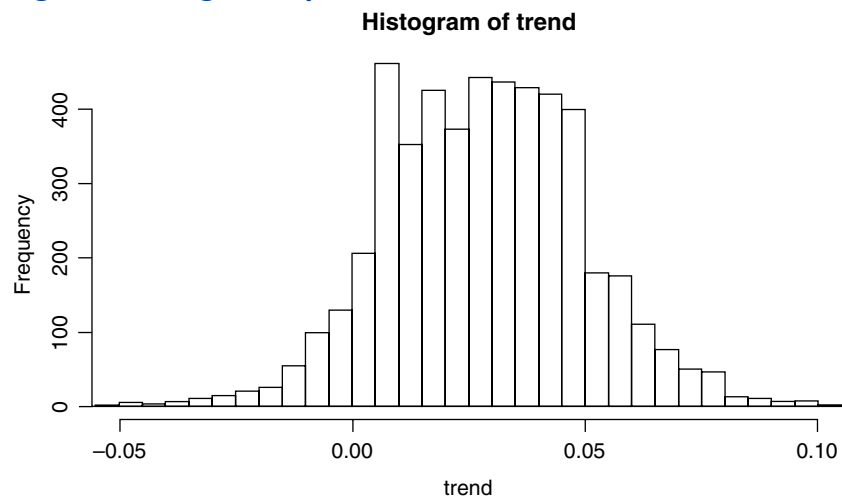
Histogram of trend

Figure 8. Histogram of posterior layer_loss

Histogram of layer_loss
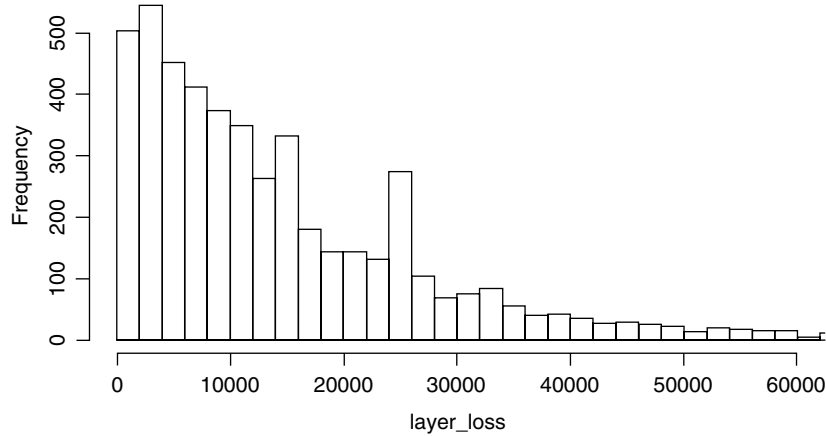
4.2. NCCI: Claims dispersion

The method put into practice by the NCCI and WCIRB for fitting ultimate severity distributions for worker's compensation losses is claims dispersion; see Corro and Engl (2006) and California WCIRB (2012). The basic idea behind claims dispersion is, given a claim amount, we can find the distribution of the ultimate settlement value by looking at the historical movement of individual claims over time. The aggregation of the ultimate settlement distribution for each current claim gives the ultimate severity distribution.

If we are working with a low to medium volume of data, it may not be possible to fit two or more parameters between every reporting age independently. Also, if the average IBNR claim is larger than the average previously reported claim, then some of the severity distribution development cannot be captured by measuring the change in existing claims.

4.3. Bayesian methods

An excellent application of Bayesian statistics to the fitting of loss distributions is given in Meyers (2005). Given a multitude of possible full severity distributions, Meyers uses the observed claim experience along with Bayes' Theorem to calculate posterior probabilities that each model is correct, then uses these probabilities to calculate the expected layer loss.

Meyers acknowledges the loss development problem and proposes a solution in which each possible loss model has an early report distribution and a corresponding ultimate loss distribution. The Bayesian calculation is done using the immature individual loss data and the early report severity curves, then the same posterior probabilities are given to the corresponding ultimate severity distributions. Construction of the ultimate severity distributions associated to immature report distributions is not described.

4.4. Stochastic models

The papers by Drieksens et al. (2012) and Korn (2015) develop more detailed stochastic models of the claims process in order to calculate reserve distributions and more. They use multiple evaluations for each claim, paid and incurred amounts, open/closed status and separate treatment of IBNR claims in a detailed stochastic framework that mimics the actual movement of individual claims.

Drieksens recognizes the need for a tail factor to account for development beyond observed ages, but the determination of the factor is left to ad hoc methods. Korn discusses two methods for determining severity distributions based on a Cox proportional hazard model in a GLM framework with semi-parametric distributions. These appear to be very robust approaches, although they may suffer in cases

of low data volume unless enhanced with Bayesian credibility as suggested by the author.

5. Conclusion

We saw evidence from actual data that when we observe a fitted parameter to the severity distribution for each single accident year and age, it will appear to follow a smooth curve with some random error. I formulated this more precisely in terms of stochastic processes. Then I showed how a Bayesian hierarchical model can reasonably reproduce the theoretical true values for the critical parameters of the stochastic process when given a modestly sized dataset of individual claim amounts at multiple evaluations. Readers now have at their disposal a new method for developing ultimate severity distributions for long-tailed lines of business.

Beyond simply being an additional method, I believe the "Bayesian stochastic severity" framework has many benefits over currently available methods. Our method has very minimal data requirements. Data in the format of Table 1 is required, although we never actually use the information on the movement of an individual claim. Only triplets of (accident year, report age, incurred amount) are necessary.

Although the link between reports for each individual claim is lost, data from multiple accident years and report ages are used simultaneously in a unified and visually intuitive way through the stochastic process. The smooth curve portion is easy to communicate and compare between applications.

The Bayesian model responds appropriately to the volume of data provided, hence we can proceed with a small- to medium-sized dataset. If the dataset is limited in time, the method explicitly accounts for development beyond the latest observed report age. With a limited number of accident years, minimal information on trend is available and the model will fail to update the posterior significantly from the prior, which is a good thing in that case.

Finally, I am providing readers with full computer code and sample dataset for easy reproduction of the results presented and the ability to conduct further experimentation.

References

California WCIRB Actuarial Research, "2013 California Retrospective Rating Plan Technical Documentation," *California Retrospective Rating Plan, Supplemental Information*, 2012.

Corro, D., and G. Engl, "The 2004 NCCI Excess Loss Factors," Casualty Actuarial Society *Forum,* Fall 2006, pp. 513-571.

Drieksens, D., et al., "Stochastic Projection for Large Individual Losses," *Scandinavian Actuarial Journal* 2012, 1, 2012, pp. 1-39.

Klugman, S., H. Panjer, and G. Willmot, *Loss Models: From Data to Decisions*, 3rd ed., New York: Wiley, 2008.

Korn, U., "A Frequency-Severity Stochastic Approach to Loss Development," Casualty Actuarial Society *E-Forum,* Spring 2015.

Meyers, G., "On Predictive Modeling for Claim Severity," Casualty Actuarial Society *Forum,* Spring, 2005, pp. 215-254.

Scollnik, D., "Actuarial Modelling with MCMC and BUGS," *North American Actuarial Journal* 5:2, 2001, pp. 96-125.

Zhang, Y., V. Dukic, and J. Guszcza, "A Bayesian Nonlinear Model for Forecasting Insurance Loss Payments," *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 175(2), 2012, pp. 637-656.

6. Appendix A

Below is the code for the Stan model, stored in the file "model.stan" called from the R code:

```
data {
  int<lower=0> N;
  real size_of_loss[N];
  int<lower=0> year[N];
  int<lower=0> age[N];
}
parameters {
real mu_start;
real mu_incr;
real<lower=0> mu_growth;
real trend;
real<lower=0.01> sigma_start;
real<lower=0> sigma_incr;
real<lower=0> sigma_growth;
real<lower=0> p_mu;
real<lower=0> r_mu;
real<lower=0> p_sigma;
real<lower=0> r_sigma;
real err_mu[10,10];
real err_sigma[10,10];
}
transformed parameters {
real sd_mu[10];
real sd_sigma[10];
real mu[10,10];
real sigma[10,10];
real claim_mean[N];
real claim_sd[N];
    for( i in 1 : 10 ) {
        sd_mu[i] <- p_mu * exp(-(i - 1) * r_mu);
        sd_sigma[i] <- p_sigma * exp(-(i - 1) * r_sigma);

        mu[i , 1] <- mu_start + trend * i + err_mu[i , 1];
        sigma[i , 1] <- sigma_start + err_sigma[i , 1];

        for( j in 2 : 10 ) {
            mu[i , j] <- mu[i , j - 1] + mu_incr * exp(( -j + 1) / mu_growth) + err_mu[i , j];
            sigma[i , j] <- sigma[i , j - 1] + sigma_incr * exp(( -j + 1) / sigma_growth) +
              err_sigma[i , j];
        }
    }
    for( i in 1 : N ) {
            claim_mean[i] <- mu[year[i] , age[i]];
            claim_sd[i] <- sigma[year[i] , age[i]];
    }
}
model {
mu_start ~ normal(9, 0.035);
mu_incr ~ uniform(0.01, 20);
mu_growth ~ uniform(0.01, 20);
trend ~ normal(0, 0.1);
sigma_start ~ inv_gamma(400,200);
sigma_incr ~ uniform(0.01, 20);
sigma_growth ~ uniform(0.01, 20);
```

```
p_mu ~ uniform(0.01, 5);
r_mu ~ uniform(0.01, 20);
p_sigma ~ uniform(0.01, 5);
r_sigma ~ uniform(0.01, 20);
for (i in 1:10){
    for (j in 1:10){
        err_mu[i , j] ~ normal(0, sd_mu[j]);
        err_sigma[i , j] ~ normal(0, sd_sigma[j]);
    }
}
for (i in 1:N){
    size_of_loss[i] ~ lognormal(claim_mean[i], claim_sd[i]);
}
}
generated quantities {
real mu_ult;
real sigma_ult;
real a;
real b;
real layer_loss;
real layer_freq;
real mu_next;
mu_ult <- mu_start + mu_incr * exp(-1/mu_growth) / (1 - exp(-1/mu_growth));
sigma_ult <- sigma_start + sigma_incr * exp(-1/sigma_growth) / (1 - exp(-1/sigma_growth));
a <- 1000000;
b <- 2000000;
mu_next <- mu_ult + trend * 11;
layer_loss <- exp(mu_next + 0.5 * square(sigma_ult)) * Phi((log(b) - mu_next
  - square(sigma_ult)) / sigma_ult) + b * Phi((mu_next - log(b)) / sigma_ult)
  - exp(mu_next + 0.5 * square(sigma_ult)) * Phi((log(a) - mu_next - square(sigma_ult))
  / sigma_ult) - a * Phi((mu_next - log(a)) / sigma_ult);
layer_freq <- 1-Phi( (log(a)-mu_next) / sigma_ult);
}
```

Next is the companion R code for running the model:

```
library(rstan)
source("data.txt")
set.seed(123); fit<- stan(file = 'model.stan', data=data, iter=5000, chains = 2)
print(fit, pars=c("mu_start", "mu_incr", "mu_growth", "mu_ult", "p_mu", "r_mu",
  "sigma_start", "sigma_incr", "sigma_growth", "sigma_ult", "p_sigma", "r_sigma",
  "trend", "layer_loss", "layer_freq"))
traceplot(fit, pars=c("mu_ult", "sigma_ult", "trend", "layer_loss", "layer_freq"),
  inc_warmup = FALSE)
x<-extract(fit, pars=c("mu_ult", "sigma_ult", "trend", "layer_loss"), permuted = TRUE,
  inc_warmup = FALSE)
attach(x)
hist(mu_ult, breaks = 50, xlim = range(10,13))
hist(sigma_ult, breaks = 50, xlim = range(0.5,1.5))
hist(trend, breaks = 50, xlim = range(-0.05,0.1))
hist(layer_loss, breaks = 500, xlim = range(0,60000))
```

7. Appendix B

Below is the data used in the Stan model, stored in the file "data.txt" and called from the R code:

```
data<- list(N = 1100, size_of_loss = c(27391, 5247, 5626, 8498, 18732, 29296, 4640,
14200, 6993, 10798, 9891, 6765, 6784, 13998, 20273, 4723, 6620, 16061, 24055, 26241,
11551, 20260, 16801, 27092, 15732, 49591, 19493, 35771, 21991, 17305, 117541, 79469,
29029, 23251, 32613, 16767, 31875, 42086, 79740, 40686, 22013, 91399, 12690, 5225, 6159,
50562, 82580, 12879, 45203, 28509, 7504, 75341, 36510, 15472, 740283, 76052, 66960,
330890, 90372, 289358, 22775, 113945, 199806, 87982, 7115, 38619, 452900, 144612, 48155,
14116, 124650, 62953, 263282, 13678, 71177, 239511, 40455, 180691, 8042, 33629, 143648,
47681, 119368, 73799, 24349, 96274, 8424, 128084, 34222, 24907, 15400, 67429, 98406,
93242, 741785, 41395, 74027, 155411, 17781, 679444, 178249, 74269, 23854, 10552, 19624,
14673, 20129, 64344, 51649, 284291, 154198, 379181, 150379, 101411, 200118, 22749, 81948,
4888, 492670, 72041, 59343, 357935, 444824, 167078, 71670, 40626, 130953, 41356, 89383,
491058, 645837, 75096, 62314, 59012, 295248, 312625, 51843, 35446, 199074, 19205, 113792,
48475, 657834, 112167, 140254, 65132, 31324, 270400, 33393, 1203011, 67180, 112681, 41516,
334095, 47855, 263742, 13038, 91708, 12786, 24915, 292404, 42813, 44328, 106793, 197605,
73142, 9847, 140268, 195790, 291079, 18695, 45046, 495291, 148617, 169532, 567873, 70130,
20424, 19081, 72215, 59949, 24246, 689893, 315562, 44716, 75738, 44495, 1828832, 99438,
106000, 8664, 344798, 353197, 272017, 176736, 56721, 36720, 184906, 209506, 192954, 7463,
5557, 6625, 6425, 1936, 16045, 2351, 5989, 9643, 4951, 10521, 6381, 2494, 42536, 4930,
4366, 4511, 36721, 12348, 6422, 50323, 10163, 13721, 15319, 4188, 11599, 13316, 20149,
8805, 16261, 1512, 6248, 17815, 5567, 61026, 14356, 18175, 20636, 14121, 13975, 33310,
24455, 186401, 36094, 208388, 12510, 14165, 42712, 39457, 34097, 16094, 66558, 58838,
25800, 126406, 36057, 116894, 55234, 118106, 22765, 46170, 19395, 46569, 38020, 19006,
16416, 64137, 25676, 75195, 14450, 32161, 3263, 36796, 9969, 37453, 20506, 143506, 19716,
138778, 74603, 38587, 240670, 25254, 14047, 109233, 353390, 141065, 17313, 27969, 113089,
125258, 70727, 10955, 144470, 67605, 31260, 106717, 96548, 41624, 61368, 44722, 113191,
92237, 101002, 41609, 46192, 152294, 28334, 7599, 48700, 155066, 177534, 130983, 165544,
9338, 77806, 56396, 13707, 31609, 251547, 190457, 17546, 50830, 149937, 41248, 58396,
389251, 29244, 161436, 93445, 73427, 48216, 77142, 30448, 244308, 85139, 65575, 86181,
149661, 96704, 143106, 102066, 164646, 144970, 78455, 89236, 1336131, 41302, 69487, 75280,
341458, 37486, 165637, 142835, 97178, 229452, 112998, 43624, 19580, 215436, 163919,
130124, 156586, 67557, 59918, 122495, 109581, 127714, 84716, 201129, 143145, 164793,
44320, 24383, 131758, 371019, 7783, 83966, 90372, 15359, 3307, 13665, 12801, 7727, 9381,
6927, 9085, 16676, 10232, 10292, 9872, 17645, 5114, 3641, 6582, 5125, 6599, 10376, 9925,
8915, 7101, 9674, 39673, 44349, 11586, 8014, 47398, 9231, 8749, 8755, 35577, 26803, 11433,
5556, 5706, 7729, 27343, 3145, 9803, 9913, 14619, 31510, 2267, 23245, 41074, 23828, 6049,
39367, 138770, 4815, 53012, 80507, 1803, 13671, 25257, 8868, 14324, 32148, 75268, 6393,
12289, 42779, 12595, 69797, 6538, 54523, 67595, 11403, 46451, 9720, 27337, 2715, 11443,
8396, 30312, 29386, 122162, 34861, 4551, 4446, 177268, 65600, 70106, 15082, 106718,
278412, 3587, 758210, 4599, 383022, 36282, 24875, 125631, 74215, 5042, 6075, 11989, 27858,
10718, 27807, 18454, 60956, 109004, 138845, 23947, 118984, 25004, 3801, 122783, 17966,
24468, 99368, 67037, 83109, 15686, 24555, 28000, 88171, 32114, 83804, 406543, 85813,
118656, 23121, 14278, 5185, 19115, 91406, 764318, 645227, 7539, 64905, 27599, 44745,
50258, 49787, 100134, 22960, 195094, 177531, 89873, 24199, 171135, 369367, 4128, 17674,
17456, 26677, 28912, 27763, 21711, 58541, 132510, 2063, 11827, 3999, 112228, 89860, 22672,
39198, 9756, 8338, 3781, 9194, 7306, 10672, 24336, 1766, 3948, 17081, 2946, 14141, 3605,
7129, 7890, 20126, 16645, 4926, 11193, 8274, 12421, 15053, 27175, 6973, 9257, 29861,
11225, 2756, 7919, 23980, 21839, 11814, 9948, 11344, 4770, 21957, 6896, 33266, 3219, 8254,
16041, 72735, 6768, 14823, 48228, 40420, 12070, 28574, 23307, 11663, 12180, 7522, 13506,
1652, 39369, 6242, 52273, 12457, 28490, 5775, 9431, 156860, 85657, 18746, 15866, 21796,
13242, 71418, 17132, 15611, 50649, 5122, 48639, 16540, 59757, 15958, 27394, 9353, 46000,
38188, 46426, 19602, 144475, 25218, 96294, 14467, 8506, 4876, 8500, 33080, 66834, 17376,
86244, 57509, 45856, 2378, 3168, 19850, 34751, 12670, 8610, 59768, 11081, 188032, 3640,
20240, 88980, 12805, 33166, 9257, 2105, 211269, 38215, 12551, 63909, 221840, 24685,
176227, 59358, 10332, 233916, 295171, 36118, 4498, 200729, 73205, 20838, 136954, 67975,
```

13481, 11789, 41257, 59247, 296266, 34576, 37410, 47969, 110565, 195800, 51182, 24509,
24511, 12829, 2310, 18035, 1379, 6263, 14536, 4973, 8308, 5339, 26998, 11820, 10889,
41262, 2379, 11452, 2066, 4539, 34569, 28295, 39552, 59826, 13050, 7006, 25295, 10214,
73978, 12991, 3177, 31040, 42004, 9617, 27460, 16233, 68031, 92051, 10424, 27124, 60105,
125429, 72524, 49698, 29015, 15517, 36180, 134900, 9296, 34172, 13368, 59032, 46054,
18931, 496097, 9841, 73585, 70621, 14348, 9129, 19698, 33035, 8323, 141706, 48913, 71584,
30464, 6933, 8451, 100269, 7579, 33592, 57368, 90100, 86556, 15448, 6820, 24334, 47515,
77507, 101397, 126396, 27546, 381525, 23828, 48660, 98779, 27822, 219425, 52089, 7611,
9098, 49513, 42052, 244808, 11640, 299800, 23857, 101466, 55359, 46966, 134068, 18618,
261676, 66430, 14123, 9805, 4683, 55212, 102559, 26864, 46025, 9077, 18518, 172388, 25343,
43835, 22867, 10910, 40012, 22372, 5179, 4124, 2595, 6968, 6652, 4845, 7074, 9383, 8012,
4084, 8949, 3618, 9118, 6683, 5552, 5289, 4054, 7074, 5292, 8292, 17568, 10965, 11926,
9943, 10456, 34421, 5836, 17285, 21685, 5709, 5734, 8264, 4641, 16110, 12007, 10848,
28022, 14285, 9405, 9705, 19794, 17890, 26057, 21008, 13247, 22175, 42704, 35119, 64196,
28956, 31347, 21549, 7746, 21572, 52759, 13975, 14236, 17333, 10527, 16173, 12615, 52775,
12137, 128238, 37771, 5743, 178456, 29841, 53745, 103006, 15033, 28217, 75519, 128955,
13608, 138403, 19301, 35530, 48332, 40114, 123332, 35887, 184976, 33680, 100884, 55896,
144192, 72605, 83371, 13180, 76799, 19945, 37723, 126639, 56243, 60166, 8729, 34179,
63260, 20302, 15928, 4552, 4489, 5875, 5997, 9998, 7607, 4550, 5823, 17420, 5521, 12223,
7149, 4895, 5323, 11346, 6839, 28098, 10006, 9231, 10000, 44015, 13689, 6844, 16760, 29511,
3108, 23621, 11691, 5926, 8725, 4631, 8913, 38441, 16933, 16518, 7714, 12639, 3423, 18180,
4916, 5068, 12989, 25711, 16723, 14176, 17281, 3032, 4160, 6838, 3999, 9527, 47089, 12226,
36726, 115216, 10511, 1925, 14189, 4563, 53425, 25311, 10557, 23210, 15156, 16759, 21798,
13990, 9988, 25392, 4907, 27560, 5149, 13007, 33751, 21484, 37233, 72361, 2648, 55069,
24033, 16207, 1832, 4120, 6292, 14665, 22673, 6664, 5021, 3537, 5298, 7419, 31747, 10902,
7122, 6316, 10638, 5370, 2262, 8558, 14902, 9535, 6206, 21237, 8085, 3994, 11773, 7096,
4200, 8465, 3751, 5945, 9725, 33511, 35704, 2010, 10870, 14557, 2910, 48417, 9400, 4993,
12413, 96128, 31515, 67238, 22102, 37779, 55615, 6371, 35608, 21921, 28420, 35139, 4114,
5023, 46936, 32402, 63517, 20610, 22862, 27657, 7008, 35614, 28462, 25641, 33311, 10685,
8902, 17272, 14586, 8149, 14887, 9687, 11534, 9932, 26758, 22743, 9157, 35436, 21014,
12393, 20690, 9959, 27647, 12830, 26770, 20651, 10932, 8520, 41510, 12694, 19110, 15630,
6532, 63756, 53265, 22679, 10150, 248012, 41556, 17472, 17723, 19403, 11277, 9885, 13352,
14362, 23355, 6851, 16872, 18915, 16844, 21619, 18243, 13301, 29362, 16026, 20139, 34050),
year = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,

```
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 9,
9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,
9, 9, 9, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10),
age = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,
9, 9, 9, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1))
```