# Loss Reserving Using Estimation Methods Designed for Error Reduction

Gary Venter
University of New South Wales

**Abstract:** Maximum likelihood estimation has been the workhorse of statistics for decades, but alternative methods are proving to give more accurate predictions. The rather vague-sounding term "regularization" is used for these. Their basic feature is shrinking fitted values towards the overall mean, much like in credibility. These methods are introduced and applied to loss reserving.

Improved estimation of ranges is also addressed, in part by a focus on the variance and skewness of residual distributions. For variance, if large losses pay later, as is typical, the variance in the later columns does not reduce as fast as the mean does. This can be modeled by making the variance proportional to a power of the mean less than 1.

Skewness can be modeled using the three-parameter Tweedie distribution, which for a variable $Z$ has variance $= \phi\mu^p$, $p \geq 1$. It is reparameterized here in $a, b, p$ to have mean $= ab$, variance $= ab^2$, and skewness $= pa^{-1/2}$. Then the distribution of the sum of $N$ individual claims has parameters $aN, b, p$, and $cZ$ has parameters $a, bc, p$. These properties are both useful in actuarial applications.

**Keywords:** MCMC, Loss reserving, Shrinkage priors, Tweedie distribution, Lasso.

## 1 Background

Over-parameterized models have less accurate predictions and are generally avoided by modelers, although in loss reserving they are still in use. Fitting parameterized curves to row or column factors is one way to mitigate this, but finding the right curves can be an issue. Often actuaries keep a parameter for every row and every column even though many of these are not statistically significant, in part because it is not clear how to eliminate them. The methodology here addresses that. More recently, statisticians have found that even more parsimonious models can be built by shrinking fitted values towards the overall mean, which can provide more accurate forecasts.

Credibility theory shrinks class estimates towards the overall mean using the average for the variance of individual classes over time and the variance of the class averages. The James-Stein estimator of Stein (1956) does this as well, but uses model assumptions to quantify the average individual variance. Starting with Hoerl and Kennard (1970), statistical methods have been developed that shrink the estimated mean for each observed point towards the overall mean by using a shrinkage parameter $\lambda$ which is selected based on how well the

model works on predictions for holdout samples. Typically $\lambda$ is tested by dividing the dataset into 4 – 10 groups, which are left out one at a time and predicted by the model fit on all the other groups, with various values of $\lambda$. This is called "cross-validation."

The original regularization method is ridge regression, which with parameters $\beta_j$ minimizes the negative loglikelihood NLL plus $\lambda \sum \beta_j^2$. More popular recently is lasso, or least absolute shrinkage and selection operator, which minimizes NLL plus $\lambda \sum |\beta_j|$. This has the practical advantage that as $\lambda$ increases, more and more parameters, and eventually all but the mean, go to exactly zero. This makes it a method of variable selection as well as estimation, so the modeler can start with a large number of variables and the estimation will eliminate most of them. As $\lambda$ gets smaller, the parameter-size penalty vanishes, so the MLE estimate is obtained. What is interesting is that some $\lambda > 0$ almost always performs better on the cross-validation, so shrinkage usually improves predictive accuracy.

Usually all the variables are standardized by a linear transform to make them mean zero, variance one. That way parameter size is comparable across variables. The additive part of the variable transforms gets picked up by the mean, which is not included in the penalty for sum of parameters and so is not shrunk. The other parameters end up pushed towards zero, which in turn pushes each fitted value towards the mean. Blei (2015) and Hastie, Tibshirani, and Wainwright (2015) are good references.

Bayesian versions of regularization work by giving the parameters shrinkage priors, which are mean-zero prior distributions – normal for ridge regression and double exponential for lasso. There are generalizations that use other shrinkage priors. The advantage of the Bayesian form is that it gives a distribution of parameters for parameter uncertainty calculations and it has a goodness-of-fit measure analogous to AIC for model comparisons. AIC, BIC, etc. do not work with regularized models due to parameter counting problems with shrinkage. MCMC estimation (Markov Chain Monte Carlo) can numerically produce samples of the posterior distribution without needing to specify conjugate distributions for the priors.

A classical approach similar to Bayesian estimation is to use random effects. Instead of parameters having distributions, as they do in Bayesian statistics, the effects being modeled have shrinkage distributions, like mean-zero normal. Then the effects are projected instead of the parameters being estimated. For instance, the differences between the territory frequency and the statewide frequency could be a mean-zero random effect. The only parameter would be the variance of these effects, but each territory's effect can be projected. One common method of projection is to maximize the product of the likelihood function with the probability of the effects. This turns out to be the same thing as computing the posterior mode in the Bayesian interpretation, but it can be done as a classical optimization. Ridge regression and lasso are thus special cases of random effects.

A typical assumption in random effects is that each random effect has its own variance parameter. But using the generalized degrees of freedom approach of Ye (1998), G. G. Venter, Gutkovich, and Gao (2017) found that having so many scale parameters can use up many degrees of freedom – that is, including them in the model makes the fitted values much more responsive to hypothetical small changes in the data points. Most random effects software allows users to specify having just one variance parameter for the whole model, which seems

to give considerably more parsimonious models without sacrificing too much in goodness of fit. This would get to the same result as ridge regression or lasso.

For reserve applications the starting point is a row-column factor model. To make it applicable in this context, the fitted value is the row parameter times the column parameter times a constant. For identifiability, there is no parameter for the first row or column other than the constant – the factor is 1.0. The problem with applying parameter shrinkage in this form is that if any parameter is eliminated, that row or column also gets the constant only. However if the model is set up so that each parameter is the change in the row or column factor from the previous one, then when a variable is eliminated, that row or column just gets the factor for the previous row or column. Since the first row and column get 1.0 anyway, the factor for the second row or column is its parameter change plus 1.

Here this is taken one step further – instead of the the parameters being these first differences, they are the second differences in the factors at each point. Then if one of these is zero, the modeled first difference does not change at that point, so the factor is on a line defined by the previous two factors. This seems to be a bit more realistic in actual triangles, and allows for more parsimonious models.

The row-column model is a special case of the row-column diagonal model, which includes calendar-year effects. That model is actually in wide use in the social-sciences, where it is called the age-period-cohort (APC) model. Cohorts are the rows, so could be the accident years in reserving, or more generally the years of origin. Ages are the columns, so lags, and periods are the years that the events, like payments, happen in, so here are the calendar years or payment years. The history of the models in all three directions traces back to Greenberg, Wright, and Sheps (1950), who in turn refer to data analysis by Frost (1939). In actuarial work, a column-diagonal model was discussed in G. Taylor (1977), and is called the separation model from his terminology. The first actuarial reference to the full APC model appears to be the reserve model of Barnett and Zehnwirth (2000). Mortality modelers have been using various forms of APC models fairly widely since A. E. Renshaw and Haberman (2006).

Parameter shrinkage methodology is starting to be applied in actuarial modeling. G. G. Venter, Gutkovich, and Gao (2017) model loss triangles with row, column, and diagonal parameters in slope change form fit by random effects and lasso. G. Venter and Şahin (2017) use Bayesian shrinkage priors for the same purpose in a mortality model that is similar to reserve models. G. Gao and Meng (2017) use shrinkage priors on cubic spline models of loss development. Some precursors include Barnett and Zehnwirth (2000), who apply shrinkage to reduce or omit piecewise linear slope changes in reserve modeling and Gluck (1997) who did something similar for the Cape Cod model.

Section 2 discusses the basic row-column model for cell means and goes into more detail on the parameter shrinkage approaches. Section 3 discusses loss distributions for individual cells given their fitted means. The fitting methods and properties of the distributions are illustrated in Section 4 by fitting to frequency, severity, and aggregate loss data from a published triangle. Extensions of the row-column model are discussed in Section 5. Section 6 concludes.

## 2 Parameter Shrinkage Methodology

The data is assumed to be arranged in a rectangle with a row for each year of origin (from now on called accident year for simplicity) and a column for each lag. A constant term $C$ is included and the first row factor and first column factor are set to 1.0. In the basic row-column model, the mean (or a parameter closely related to the mean, depending on the distributional assumptions) for the $[w, u]$ cell is the product of row and column factors:

$$\mu_{w,u} = A_w B_u C$$

Here $A_w$ is the parameter for accident year $w$ and $B_u$ is the parameter for lag $u$. This basic model will be used for frequency, severity, and aggregate losses by cell.

There can get to be a lot of parameters, with one for every row and column. Parameter shrinkage aims at getting more parsimonious models that avoid over-fitting and so predict better. This is the goal of regularization in general. Here there will still be a parameter for every row and every column, but several adjacent parameters could be on line segments.

When all the observations are positive, the estimation is often more efficient if the logs of the losses are modeled. Then the fitted values are the sums of the row and column log parameters, plus a constant. This can be set up in regression format with 0, 1 dummy variables identifying the row and column an observation is in. This allows the use of commonly available estimation applications. The model where the parameters are second differences can still be set up this way, but the variables become sums of 0, 1 dummies. This is illustrated in the example.

Some background on MCMC will help clarify the methodology. MCMC numerically generates a collection of samples from the posterior distribution when only the likelihood and prior are known. With data $X$ and parameters $\beta$, Bayes Theorem says:

$$p(\beta|X) = \frac{p(X|\beta)p(\beta)}{p(X)}$$

The left side is the posterior distribution of the parameters given the data, and the numerator of the right side is the likelihood times the prior. The denominator $p(X)$ is a constant for a given dataset, so maximizing the numerator maximizes the posterior. In random effects the numerator is called the joint likelihood, so maximizing it gives the posterior mode. Just using the numerator is the key to the original MCMC methodology, the Metropolis sampler. It uses a proposal generator to create a possible sample of the parameters from the latest accepted sample. If this produces a new maximum for the numerator, it is added to the collection of samples. If it doesn't, there is an acceptance rule to put it in or not, based on a [0,1] random draw. After a warmup period, the retained samples end up being representative of the posterior.

A refined version of that, the Hastings-Metropolis sampler, is more efficient. Further refinements include Hamiltonian mechanics and the no-U-turn sampler, which evolve the proposal generator dynamically. The latter is the basis of the Stan MCMC package, which is available in R and Python language applications, and some others. Another methodology is the Gibbs sampler, which draws parameters sequentially from the posterior distribution of each

parameter given the data and the latest sample of all the other parameters. The JAGS package uses that.

Basically then, MCMC is looking for parameters that give relatively high values to the loglikelihood plus the sum of the log of the probabilities of the parameters, using their priors. The posterior mode is at the set of probabilities that maximize this sum. In section 3d we show that the posterior mode using the normal or Laplace prior gives the ridge regression or lasso estimated parameters.

*2a Posterior Mean vs. Posterior Mode*

While classical shrinkage methods agree with the Bayesian posterior mode, the posterior mean is the basic Bayesian estimator. The mode is very similar to classical estimation in that it optimizes a probability – such a the NLL or joint likelihood.

The posterior mean is a fundamentally different approach. It does not maximize a probability. Instead it looks at all the parameter sets that could explain the data, and weights each according to its probability. The most likely set of parameters has appeal, but it has more risk of being a statistical fluke. If it is similar to many other possible parameter sets, then it would probably be only very slightly higher in posterior probability and not much different than the mean. But if it is very different, it could be overly tailored to that specific data set. In that case, only a small percentage of the MCMC samples would be close to that point. The posterior mean is aimed at getting an estimate that would still perform well on other samples.

*2b Measuring Goodness of Fit*

Traditional goodness-of-fit measures, like AIC, BIC, etc., penalize the loglikelihood with parameter-count penalties. This is already problematic for non-linear models, as the parameter count does not necessarily measure the same thing for them. Ye (1998) developed a way to count parameters using what he calls generalized degrees of freedom. These measure how sensitive the fitted values are to slight changes in the corresponding data points. This is accomplished by taking the derivative of each fitted value with respect to the data point, usually numerically. It agrees with the standard parameter count given by the diagonal of the hat matrix for linear models.

Parameter shrinkage also makes the parameter count ambiguous, and from Ye's perspective, the shrunk parameters do not allow as much responsiveness to changes in the data, so do not use up as many degrees of freedom. For lasso, the gold standard of model testing is leave-one-out estimation, or loo. The model is fit over and over, each time leaving out a single observation, with the loglikelihood computed for the omitted point. The sum of those loglikelihoods is the loo fit measure.

Both loo and Ye's method are computationally expensive, and do not work well with MCMC anyway because of sampling uncertainty. To address this, Gelfand (1996) developed an approximation for a sample point's out-of-sample loglikelihood using a numerical integration technique called importance sampling. In his implementation, that probability is estimated

as its weighted average over all the samples using weights proportional to the reciprocal of the point's likelihood under each sample. That gives greater weight to the samples that fit that point poorly, which would be more likely to occur if that point had been omitted. The estimate of the probability of the point comes out to be the reciprocal of the average over all the samples of the reciprocal of the point's probability in a sample. With this, the sample of the posterior distribution of all the parameters generated by MCMC is enough to do the loo calculation.

That gave good but still volatile estimates of the loo loglikelihood. Vehtari, Gelman, and Gabry (2017) addressed that by something akin to extreme value theory – fitting a Pareto to the probability reciprocals and using the fitted Pareto values instead of the actuals for the largest 20% of the sample. They call this "Pareto-smoothed importance sampling." It has been extensively tested and has become widely adopted. The penalized likelihood measure is labeled $\widehat{elpd}_{loo}$, standing for "expected log pointwise predictive density." It aims at doing what AIC etc. were trying to address as well – adjusting the loglikelihood for sample bias.

The Stan software provides a loo estimation package that can work on any posterior sample, even those not from Stan. It outputs $\widehat{elpd}_{loo}$ as well as the implied loglikelihood penalty and something they call looic – the loo information criterion – which is $-2\widehat{elpd}_{loo}$ in accord to standards of information theory. Since the factor is not critical, here the term looic is used for $-\widehat{elpd}_{loo}$, which is the negative loglikelihood (NLL) increased by the penalty.

*2c Selecting the Degree of Shrinkage*

Selecting the scale parameter of the Laplace or Cauchy prior for MCMC, or the $\lambda$ shrinkage parameter for lasso or ridge regression, requires a balancing of parsimony and goodness of fit. Taking the parameter that optimizes $\widehat{elpd}_{loo}$ is one way to proceed, and that was the approach taken in G. Venter and Şahin (2017). However this is not totally compatible with the posterior mean philosophy, as it is a combination of Bayesian and predictive optimization. An alternative would be to give a sufficiently wide prior to the scale parameter itself and include that in the MCMC estimation. This is called a fully Bayesian method and produces a range of sample values of $\lambda$. G. Gao and Meng (2017) is a loss reserving paper using the fully Bayesian approach. That is the approach taken here.

Lasso applications, like the R package glmnet, use cross-validation to select a range of candidate $\lambda$ values. An alternative is to build in more of the Bayesian approach. The Laplace = double exponential prior is discussed in Section 3d as well. There the log density is given as $log[f(\beta|\sigma)] = -log(2) - log(\sigma) - |\beta|/\sigma$, with $\sigma = 1/\lambda$. Summing over the $k$ parameters makes the negative log probability $= k * log(2) - k * log(\lambda) + \lambda \sum |\beta_j|$. This is the lasso penalty to the NLL of the data, but if $\lambda$ is a given constant, the first two terms are dropped. However if $\lambda$ itself is given a uniform prior with density $= C$ over some interval, the second term needs to be included, but the uniform density is a constant that can be dropped. Thus the quantity to be minimized over $\lambda, \beta_j$ is:

$$NLL - k * log(\lambda) + \lambda \sum |\beta_j|$$

The uniform prior is an arbitrary but reasonable choice, so values of $\lambda$ that are not at the exact minimum of this are possible candidates as well.

*2d Estimation Issues*

Instead of doing MCMC, a non-linear optimizer like Nelder-Mead could be used to get the posterior mode through classical estimation. Good starting parameters seem to be needed, however. One advantage of MCMC is that it seems to be able to find reasonable parameter sets better than classical optimization. That might in fact be one of its historical attractions. However MCMC can also find a lot of local maxima that are not that good fits. The lingo of MCMC appears to be that this will happen if the model is "poorly specified." In practice that seems to mean that the priors are too wide. Running the estimation with starting values from the better previous fits also can help avoid bad local maxima.

Starting with lasso can give a starting point for MCMC. Stan gives good output on which parameters are not contributing to the fit, but the second difference variables are negatively correlated so work in groups, which makes some individual parameter ranges less indicative of the value of those parameters. Lasso gives parameter sets that work together at each value of $\lambda$.

The Stan software used here is not able to include R packages like Tweedie and gamlss.dist. With good starting parameters from related Stan fits, classical estimation in R can maximize the posterior mode for the Tweedie and PiG distributions discussed in Section 3, and at least compare fits by the posterior mode probabilities. Some of that was done in the examples below. Unfortunately, neither the posterior mean nor looic can be computed this way, so the comparisons are more suggestive.

## 3 Distributions for Reserve Modeling

Detailed distribution formulas follow, but there are a few key takeaways:

- Development triangles are subject to a unique form of heteroskedasticity. The variance is not constant among the cells but it often decreases less than the mean does across the triangle, due to volatile large losses paying later. This is addressed here by introducing an additional variance parameter. The easiest example is for the normal distribution – instead of a constant variance, the variance, and so the standard deviation, is $s\mu^k$. If $k < 1$, the variance decreases slower than the mean. Something similar can be done for any distribution and is labeled as the k form. The Weibull-k is particularly interesting as its skewness changes more than is seen in other distributions, often in a helpful way,
- The Tweedie distribution, usually parameterized with variance $= \phi\mu^p$, $p \geq 1$, is reparameterized in $a, b, p$ to have mean $= ab$, variance $= ab^2$, and skewness $= pa^{-1/2}$. Then the distribution of the sum of variables with the same $b$ and $p$ parameters is Tweedie in $\sum a_j, b, p$. Also if $Z$ is Tweedie in $a, b, p$, then $cZ$ has parameters $a, bc, p$. This puts the focus on controlling the skewness with the $p$ parameter. In the usual form, the skewness is still $p$CV, but the skewness relationship is overshadowed by the variance. The additive feature makes it possible to fit a severity distribution if only the number and total value of payments are known for each cell – the individual payments are not needed. This is the case for the normal-k as well, but with a slightly different

formula. The reparameterization also makes it easier to represent mixtures of Poissons by a Tweedie, which generalizes the negative binomial and Poisson–Inverse Gaussian.

- Choosing which parameter of a distribution to fix among the cells can also change the mean-variance relationship across the triangle. For example, the gamma with mean $\mu = ab$ and variance $ab^2$ has variance $= b\mu = \mu^2/a$, so fixing $a$ in all the cells makes the variance proportional to mean squared, but fixing $b$ makes it proportional to the mean. This then works the same way with any Tweedie distribution, which allows either relationship with any skewness/CV, as determined by $p$. The form with variance proportional to mean often works fairly well, depending on how the larger loss payments are arranged. The Tweedie mixed Poissons like the negative binomial are related to this. They come in two forms with different mean-variance relationships, which arise from the mixing Tweedie having $a$ or $b$ fixed across the cells. When fitted to a single population, that is to only one cell, the fits from the two forms are identical.
- The typical ODP assumption has variance proportional to mean, but the actual ODP in the exponential family takes values only at integer multiples of $b$, which is not what is needed for losses. Thus usually the ODP is applied to reserving with the quasi-likelihood specified but without any identified distribution function. The essential feature of this is that the variance is proportional to the mean, so any Tweedie with fixed $b, p$ could represent such an ODP, and in fact the gamma is often used in ODP simulations, where an actual distribution function is needed. But the gamma can be fit directly by MLE, which would allow the use of the Fisher information for parameter uncertainty instead of bootstrapping. (The parameters are asymptotically normal, but for positive parameters and usual sample sizes, a gamma with a normal copula usually works better for the parameter distribution.)

Details are also given for the shrinkage distributions for MCMC, and generalizations of classical lasso and ridge regression are discussed with them. Some of the distributional discussions can be skipped and referred back to in the examples, depending on reader interest.

*3a Aggregate Loss Distributions*

*3a.1 Tweedie*

The Tweedie distribution is usually parameterized so that $EX = \mu$ and $VarX = \phi\mu^p$. However its derivation starts out as a member of a class called the exponential dispersion family, with parameters $p$, $\lambda$ and $\theta$ having $EX = \lambda\theta$ and $VarX = \lambda\theta^p$. Then taking $\mu = \lambda\theta$ and $\phi = \lambda^{1-p}$ gives the usual form. This form has computational advantages relating to quasi-likelihood estimation, but as computation gets less expensive, this issue declines in importance. Good references for these distributions include Jørgensen (1987), Jørgensen (1997), and Arthur E. Renshaw (1994). The Wikipedia article on the Tweedie distribution gives a good summary as well.

Getting the variance $= \phi\mu^p$ requires making $\phi$ a function of $p$. With parameters $s, k$ with $s = \theta^{p-k}\lambda^{1-k}$, the variance becomes

$$s\mu^k = \theta^{p-k}\lambda^{1-k}(\lambda\theta)^k = \lambda\theta^p$$

for any $k$, which is an additional parameter.

For a single cell, it is meaningless to say the variance is proportional to a given power of the mean, as you can make two fixed numbers proportional with any power you want. It is when you make some parameters constant across all the cells that the variance can be proportional to a power of the mean for the whole dataset. So if you make $\phi$ and $p$ constant across the cells, you get the variance proportional to $\mu^p$. But if you make $s$, $k$, and $p$ constant across the cells, the variance is then proportional to $\mu^k$ across the dataset.

The Tweedie family in the original parameterization with fixed $p, \theta$ is closed under addition of independent variates, with $\sum X_j$ having parameters $\lambda_0 = \sum \lambda_j$, $\theta$, $p$. In the common parameterization, $\lambda = \phi^{1/(1-p)}$, the $\phi$ parameter for $\sum X_j$ is

$$\phi_0 = \left[ \sum \phi_j^{\frac{1}{1-p}} \right]^{1-p}$$

This supposes that $\theta = \mu_j/\lambda_j = \mu_j \phi_j^{1/(p-1)}$ is constant among the summands. Then $\mu_0 = \theta\lambda_0$.

The family with fixed $p, \lambda$ is closed under multiplication by a constant $c$. In the $\phi, \mu, p$ form, suppose that the resulting parameters are $\phi_0, \mu_0, p$. Since $E(cX) = cEX$ and $Var(cX) = c^2 VarX$, we must have $\mu_0 = c\mu$ and

$$\phi_0 (c\mu)^p = c^2 \phi \mu^p$$

This leads to $\phi_0 = c^{2-p}\phi$.

Another parameterization, which makes the sum and scale results much more convenient, has parameters $a, b, p$ with $a = \theta^{2-p}\lambda$ and $b = \theta^{p-1}$. Then $ab = \lambda\theta = EX$, and $ab^2 = \lambda\theta^p = VarX$. Looking at $\sum X_j$ with fixed $\theta, p$: since $\lambda_j = \theta^{p-2} a_j$, we have

$$a_0 = \theta^{2-p}\lambda_0 = \theta^{2-p} \sum \lambda_j = \theta^{2-p} \sum \theta^{p-2} a_j = \sum a_j$$

For $cX$, we have $EcX = cEX = cab = a_0 b_0$ and $Var(cX) = c^2 ab^2 = a_0 b_0^2$. Then dividing variance by mean and mean-squared by variance produces $b_0 = cb$ and $a_0 = a$. Thus $b$ is a scale parameter, and the $a$ shape parameters add across independent distributions. This can be used for instance in simulating the sum of individual claims from a Tweedie severity.

Although $p$ does not appear in the mean and variance formulas, it is still part of the distribution. In fact, $Skw(X) = p/\sqrt{a}$. More generally for the Tweedie, $Skw(X) = pCV(X)$, where CV is the coefficient of variation, i.e., the standard deviation divided by the mean. This follows from a more general formula of Arthur E. Renshaw (1994) for skewness in the linear exponential family. Thus in the $\mu, \phi, p$ parameterization, $Skw(X) = p\sqrt{\phi}\mu^{p/2-1}$. In the $\mu, s, k, p$ parameterization, $Skw(X) = p\sqrt{s}\mu^{k/2-1}$. The $p$ parameter may or may not appear in the variance of the Tweedie, but it is key in the skewness. That is the fundamental significance of the choice of $p$.

In the $a, b, p$ parameterization, fixing $b$ across the cells makes the variance proportional to the mean for any choice of $p$. This is possibly useful for modeling aggregate losses. On the

other hand, fixing $a$ across the cells makes the variance proportional to the mean squared, which could be useful for severity. In this parameterization, the mean and variance are the same as are usually given for the gamma distribution. Thus the Tweedie can be looked on as a generalization of the gamma where there is another parameter $p$ for the skewness.

In general, $E(X-EX)^3 = EX^3 - 3EX^2EX + 2(EX)^3$ and
$Skw(X) = E(X-EX)^3 Var(X)^{-1.5}$. In terms of $a, b, p$ some moments are:

- $EX^2 = ab^2 + a^2b^2$
- $EX^3 = b^3(pa + 3a^2 + a^3)$
- $E(X-EX)^3 = pab^3$

These combine to give $Skw(X) = p/\sqrt{x}$.

The Tweedie with $1 < p < 2$ in particular has been used for aggregate losses. It can be derived as a Poisson frequency and a gamma severity with frequency and severity both smaller in smaller cells. See Meyers (2009) or G. G. Venter (2007). In loss triangles, however, the smaller cells often have larger severity. The gamma/Poisson interpretation is not necessary to use these values of $p$, but there still will be a positive probability at zero.

The gamma distribution is the Tweedie with $p = 2$, and $p = 3$ gives the inverse Gaussian. With $p = 1$, the probability is only positive at integer multiples of $b$. This is sometimes called the over-dispersed Poisson, but it could be under-dispersed as well. The Poisson is when $b = p = 1$. The only other closed-form density is $p = 0$, the normal distribution. For the inverse Gaussian density in the $a, b, p = 3$ parameterization the density is:

$$f(x|a,b) = \sqrt{\frac{a^2b}{2\pi x^3}} exp\left(\frac{-(x-ab)^2}{2bx}\right)$$

The R package Tweedie has distribution and density functions and inverses for simulation for $p \geq 1$. It uses the $\mu, \phi, p$ parameterization, so to use it for the $a, b, p$ parameterization, set $\mu = ab$ and $\phi = a^{1-p}b^{2-p}$. To use the $\mu, s, k, p$ parameterization, set $\phi = s\mu^{k-p}$.

There is no Tweedie with $0 < p < 1$. For $p < 0$, the Tweedie is very heavy tailed but is shaped like a negatively skewed mean zero distribution on the real line. It is a generalization of the standard normal called an extreme stable distribution. The density contains an infinite sum and is a function of $p$ and $\lambda$ in the original parameterization. See Jørgensen (1997). For the standard normal, $(X_1 + ... + X_n)/\sqrt{n}$ is also standard normal. For the Tweedie with $p \leq 0$ and $X_j$ iid in $\lambda, p$, $(X_1 + ... + X_n)n^{(p-1)/(2-p)}$ is also Tweedie in $\lambda, p$. This is the basic requirement for a distribution to be stable. The standard normal is the case $p = 0$.

*3a.2 Normal-k*

The constant variance of the normal does not work for triangle fits because the variance decreases for the later cells. One way to address this heteroskedasticity is to set $\sigma^2_{w,u} = s\mu^k_{w,u}$ for parameters $s, k$. This adds an additional parameter. It is not meaningful when fitting a normal to a single distribution, because for any two values of $k$, you can find two values of

$s$ that will give the same $\sigma^2$. It is only when you need distributions for each cell that this becomes useful. The main drawback of this distribution is that it has zero skewness.

*3a.3 Gaussian Inverse Gaussian – GiG*

The inverse Gaussian distribution is the Tweedie with $p = 3$. The name arises for some abstract reason not usually relevant. It has skewness = 3CV, which is more skewed than the gamma but less than the lognormal. Most reserve cell distributions have less skewness than this, so a weighted average of the Gaussian and inverse Gaussian distributions with the same mean and variance can encompass a good deal of the triangles actuaries have to deal with. The GiG here is built around $\sigma^2_{w,u} = s\mu^k_{w,u}$, so the parameters will be $s$, $k$, the row and column parameters defining the cell means, and a parameter $v$ in [0,1] for percent Gaussian.

The inverse Gaussian density is closed form but a bit complicated so it is often easier to use a packaged function. Most published density functions and programmed software use the $\mu, \phi$ parameterization, often in $1/\phi$, so set $1/\phi = a^2 b$ to match mean and variance.

*3a.4 Weibull-k*

The Weibull distribution with parameters $\lambda, h$ has $f(x) = \frac{hx^{h-1}}{\lambda^h} e^{-(x/\lambda)^h}$ and $F(x) = 1 - e^{-(x/\lambda)^h}$. The moments are gamma functions and are more compact with the notation $n! = \Gamma(1+n)$, which agrees on the integers. Then $EX = \lambda \frac{1}{h}!$ and $VarX = \lambda^2 \left[\frac{2}{h}! - (\frac{1}{h}!)^2\right]$. These give $CV^2 = \frac{2}{h}!/(\frac{1}{h}!)^2 - 1$. The skewness is $[\frac{3}{h}!/(\frac{1}{h}!)^3 - 1]/CV^3 - 3/CV$. The skewness is negative for $h > 3.60235$ or so and gets large for small $h$. This gives a range of distribution shapes.

For the heteroskedasticity in a reserve triangle, it again might be helpful to be able to set $VarX = s(EX)^k$. This would require

$$1 + CV^2 = \frac{2}{h}!/\left(\frac{1}{h}!\right)^2 = 1 + s(EX)^{k-2}$$

Unfortunately this would have to be solved numerically. There are various root finding programs that can solve for $h$ inside of an estimation routine. This is easier in logs due to limitations of double-precision numbers. Calling the right side $v$, the equation to solve is

$$g(h) = log\left(\frac{2}{h}!\right) - 2log\left(\frac{1}{h}!\right) - log(v) = 0$$

This can be done for example by iterating with Newton's method starting at some value $h_0$ and setting $h_{j+1} = h_j - g(h_j)/g'(h_j)$. For this, $g'(h)$ is easy enough with the digamma function $\psi(x) = \partial log\Gamma(z)/\partial z$, which is widely available in software packages. With this,

$$g'(h) = 2\left[\psi\left(1 + \frac{1}{h}\right) - \psi\left(1 + \frac{2}{h}\right)\right]/h^2$$

*3b Severity Distributions*

Severity in loss triangles does not usually have the same heteroskedasticity problems that aggregate has, so any severity distribution can be tried. Typically the variance is proportional to mean squared for severity. Thus the $a, b, p$ form of the Tweedie with $a$ fixed across cells is a good starting point. The tail is not usually as heavy for individual cells as it is for the whole severity distribution used for pricing. The additive form property of the $a, b, p$ parameterization makes it easy to use when the data is only number of payments $n_{w,u}$ and total payments $x_{w,u}$ for the cell. Then $x_{w,u}$ is distributed $n_{w,u}a, b_{w,u}, p$. For the normal-k, $x_{w,u}$ is normal with mean $\mu_{w,u}n_{w,u}$ and variance $n_{w,u}s\mu_{w,u}^k$.

*3c Frequency Distributions*

*3c.1 Poisson*

The Poisson is the Tweedie with $p = b = 1$ and $a$ is usually called $\lambda$. Some moments are:

- $EN = Var(N) = \lambda$
- $Skw(N) = 1/\sqrt{\lambda}$
- $EN^2 = \lambda + \lambda^2$
- $EN^3 = \lambda + 3\lambda^2 + \lambda^3$
- $E(N - EN)^3 = \lambda$

One problem is that the variance of the cells have to pick up the Poisson variability as well as any specification error in the mean, and the Poisson variance can be too limited for this.

*3.c.2 TweeP – Tweedie Mixture of Poissons*

Adding some variability to the Poisson is often done by assuming the Poisson $\lambda$ is itself uncertain, and assigning a distribution for that. The most common case is to use a gamma distribution for $\lambda$, which yields the negative binomial. But this is often misapplied. If there is a population of drivers, for example, each with a Poisson distribution for number of accidents in a year, with $\lambda_j$ for driver $j$, then the number of accidents for the whole population is Poisson in $\sum \lambda_j$. This is just a case of the additive property of the Tweedie. The negative binomial arises if a driver is chosen at random, with unknown $\lambda_j$ that is gamma distributed.

Assume $\lambda$ is distributed Tweedie $a, b, p$. To get the moments, use the formula $Eg(N) = EE[g(N)|\lambda]$. Then

- $EN = EE[N|\lambda] = E\lambda = ab$
- $EN^2 = EE[N^2|\lambda] = E[\lambda + \lambda^2] = ab + ab^2 + a^2b^2$
- $Var(N) = ab(1 + b)$
- $EN^3 = EE[N^3|\lambda] = E[\lambda^3 + 3\lambda^2 + \lambda] = pab^3 + 3a^2b^3 + a^3b^3 + 3ab^2 + 3a^2b^2 + ab$
- $E(N - EN)^3 = EN^3 - 3EN^2EN + 2(EN)^3 = pab^3 + 3ab^2 + ab$

The last item requires a bit of algebra. After a little more,

$$Skw(N) = \frac{pb^2 + 3b + 1}{(1+b)\sqrt{ab(1+b)}}$$

In the frequency world, some, like Mathematica documentation, use the notation $r = a$, $q = b/(1+b)$. Then $b = q/(1-q)$, $1 + b = 1/(1-q)$ and $b(1+b) = q/(1-q)^2$. It is even more common to use $p$ instead of $q$, but here $p$ is already used for the Tweedie skewness parameter. Substituting this notation produces:

$$Skw(N) = \frac{(p-2)q^2 + q + 1}{\sqrt{rq}}$$

Hougaard, Mei-Ling, and Whitmore (1997) discuss the TweeP and provide a formula for computing the probabilities for any $p > 1$ except 2, which they say works up to about $n = 150$ before running into problems with double-precision representations. This would be fine for distributions with small counts, like claims per policy, but it would not handle aggregate claims from larger business units. Some special cases discussed below have closed form distributions for any $n$.

They start by introducing three transformed parameters to simplify the formulas, defined by $\alpha = (p-2)/(p-1)$, $1/\delta = \sqrt{2\phi}$, and $1/\theta = 2\mu^2\phi$, then define the coefficients $c_{n,j(\alpha)}$ recursively by: $c_{n,0}(\alpha) = \Gamma(n-\alpha)/\Gamma(1-\alpha)$ and $c_{n+1,j+1}(\alpha) = (n-(j+1)\alpha)c_{n,j+1}(\alpha) + c_{n,j}(\alpha)$. Then

$$f(0) = exp\Big[-\frac{\delta}{\alpha}\Big([\theta+1]^\alpha - \theta^\alpha\Big)\Big]$$

$$f(n) = \frac{f(0)}{n!}\sum_{j=1}^{n} c_{n,j}(\alpha)\delta^j(\theta+1)^{\alpha j - n}$$

*3c.3 Negative Binomial*

The negative binomial is the TweeP with $p = 2$. It has a closed form probability mass function. In the $q, r$ form it is

$$f(n; r, q) = \frac{\Gamma(r+n)}{n!\Gamma(r)}q^n(1-q)^r$$

It has mean $= ab = m = qr/(1-q)$, variance $= ab(1+b) = qr/(1-q)^2 = m/(1-q)$ and skewness $= (1+q)/\sqrt{qr} = (1+q)CV$.

Like with the Tweedie, two basic forms for cell distributions come about by fixing either $a$ or $b$ across the cells. If $b$ and so $q$ is fixed across the cells, then the variance is proportional to the mean. If $a$ and so $r$ is fixed, it is convenient to eliminate $q$ by $qr = m - qm$, so $q = m/(r+m)$. Then $1 - q = r/(r+m)$. The variance $m/(1-q)$ then becomes $m(r+m)/r$ or $m + m^2/r$. Thus the variance is a quadratic function of the mean.

The second is the form used in GLM and often works better as a distribution of residuals, perhaps because the part of the residual distribution that comes from estimation error for the mean is large enough in large cells to benefit from the mean$^2$ term. The probability mass function then is:

$$f(n;m,r) = \frac{\Gamma(r+n)m^n r^r}{n!\Gamma(r)(m+r)^{n+r}}$$

*3c.4 Poisson Inverse Gaussian – PiG*

The Poisson mixed by the inverse Gaussian is the TweeP with $p = 3$. It has the same mean and variance as the negative binomial. The skewness is $(1+q+q^2)/\sqrt{qr}$. It is thus a bit more skewed alternative to the negative binomial. It also has the two forms of parameterization across a data set. As usual, they both give the same distribution for a single sample, that is one not involving multiple cells, like statewide accident frequency. The density has calculation issues, but the probability generating function in the $m, r$ parameterization is:

$$P(z) = e^{r-r\sqrt{1-\frac{2m}{r}(z-1)}}$$

See Dean, Lawless, and Willmot (1989), who also give a recursive algorithm for calculating $f(n;m,r)$, which is a bit simpler than the algorithm of Hougaard, Mei-Ling, and Whitmore (1997) above with $p = 3, \alpha = -1/2$. There is an exact probability mass function involving modified Bessel functions. However these can run into problems with double precision representations if there are a large number of claims. Perhaps 40 – 50 digit precision could be needed to calculate them in some cases. R does have specialized functions for arbitrary-precision numbers, but not every Bessel function application uses them. The modified Bessel function of the first kind is defined as:

$$I_\alpha(x) = \sum_{j=0}^{\infty} \frac{1}{j!(\alpha+j)!}\left(\frac{x}{2}\right)^{2j+\alpha}$$

The modified Bessel function of the second kind, which is the same thing as the modified Bessel function of the third kind – an obsolete but stubborn term, is

$$K_\alpha(x) = \frac{\pi}{2}\frac{I_{-\alpha}(x) - I_\alpha(x)}{sin(\alpha\pi)}$$

With this (see Zha, Lord, and Zou (2016)), the PiG probability mass function is:

$$f(n;m,r) = \sqrt{\frac{2\alpha}{\pi}}\frac{m^n e^r K_{n-1/2}(\alpha)}{(\alpha/r)^n n!}$$

where $\alpha = \sqrt{r^2 + 2mr}$.

The dPIG function in the R package gamlss.dist seems to be able to calculate this with any $m$, so probably uses arbitrary precision numbers.

*3c.5 Sichel Distribution*

The Sichel is a three-parameter distribution that comes from mixing the Poisson by a generalization of the inverse Gaussian. Its skewness is greater than that of the negative binomial and can be greater than that of the PiG as well. Rigby, Stasinopoulos, and Akantziliotou (2008) is a good source for this and other heavier-tailed count distributions. G. G. Venter (2011) applies the Sichel to mortality data relevant for workers compensation and finds that it fits slightly better than the negative binomial.

The Sichel probability function is also closed form using the Bessel functions. It is:

$$f(n;m,r,\nu) = \frac{K_{n+\nu}(\alpha)(mc)^n}{K_\nu(r)n!(\alpha/r)^{n+\nu}}$$

where $\alpha = \sqrt{r^2+2mrc}$ and $c = K_\nu(r)/K_{\nu+1}(r)$.

It has the same mean and variance as the PiG, which is the special case $\nu = -\frac{1}{2}$. The parameter $\nu$ can be any real number. The negative binomial is a limiting case. Higher moments are shown in the appendix of Rigby, Stasinopoulos, and Akantziliotou (2008), but the $c$ there is $1/c$ here, and $\sigma$ is $1/r$. The density function is available in the R gamlss.dist package.

*3c.6 Etc.*

The zero-truncated frequency distributions, which eliminate the positive probability at zero, provide further choices, and there are other mixtures as well. The appendix of Klugman, Panjer, and Willmot (2008) is a good starting point for these.

*3d Distributions for Use as Shrinkage Priors*

Shrinkage priors are mean-zero priors that push parameters towards zero, which can be offset by the likelihood increase if the parameter is important to creating a better fitting model. In both classical and Bayesian estimation these offsetting priorities are balanced by finding parameters that give high values to the sum of the loglikelihood plus the log of the prior probabilities of those parameters. Shrinkage can be done towards any value, but only the mean-zero versions are used here.

*3d.1 Normal Distribution*

If the parameter $\beta$ is distributed normal(0,$\sigma$), the log of the density is:

$$log[f(\beta|\sigma)] = -log(2\pi)/2 - log(\sigma) - \frac{\beta^2}{2\sigma^2}$$

Constants – meaning any terms not having parameters being estimated – can be ignored in the estimation. In fact with a fixed value of $\sigma$, the estimation would look for higher values of [loglikelihood $-\frac{1}{2}\sum\beta_j^2/\sigma^2$]. This value is what is maximized in ridge regression, for selected values of $\lambda = 2/\sigma^2$.

*3d.2 Laplace Distribution*

The Laplace density on the real line is

$$f(\beta|\sigma) = \frac{1}{2\sigma} exp\Big( - \frac{|\beta|}{\sigma} \Big)$$

This has variance $= 2\sigma^2$ and kurtosis $= 6$.

Also $log[f(\beta|\sigma)] = -log(2) - log(\sigma) - |\beta|/\sigma$. With a fixed value of $\sigma$, the estimation seeks high values of [loglikelihood $- \sum |\beta_j|/\sigma$]. This is maximized in lasso. Shrinkage with the Laplace prior is thus called Bayesian lasso.

*3d.3 Cauchy Distribution*

The Cauchy is just the Students-t distribution with one degree of freedom, so is heavy-tailed. In fact the mean does not even exist as the integral defining it does not converge. The density and its log are:

$$f(\beta|\sigma) = \frac{\sigma}{\pi} \frac{1}{\sigma^2 + \beta^2}$$

$$log[f(\beta|\sigma)] = log(\sigma) - log(\pi) - log(\beta^2 + \sigma^2)$$

Thus ignoring constants and for a fixed value of $\sigma$, the optimization would be on [loglikelihood $- \sum log(\beta_j^2 + \sigma^2)$]. This is not a common classical method, but perhaps it should be.

The Cauchy prior is usually used with a smaller value of $\sigma$ than for the Laplace prior. It then puts more weight on small values of the parameters, but still allows occasional larger values if they provide enough improvement in the loglikelihood. In this way it usually produces more parsimonious models than the Laplace does, but often with only a slight reduction in loglikelihood. It is becoming more popular as a shrinkage prior, and the classical analogue could provide a similar improvement over lasso.

*3d.4 Scaled t-prior*

The scaled t distribution with $\nu$ degrees of freedom and its log are:

$$f(\beta|\nu,\sigma) = \frac{\Gamma\Big(\frac{1}{2} + \frac{\nu}{2}\Big)}{\sigma\sqrt{\pi\nu}\Gamma\Big(\frac{\nu}{2}\Big)} \left(1 + \frac{\beta^2}{\nu\sigma^2}\right)^{-(\nu+1)/2}$$

$$log[f(\beta|\nu,\sigma)] = log\Big[\Gamma\Big(\frac{1}{2} + \frac{\nu}{2}\Big)/\Gamma\Big(\frac{\nu}{2}\Big)\Big] - \frac{1}{2}\left[log(\sigma^2) + log(\nu) + log(\pi) + (\nu+1)log\left(1 + \frac{\beta^2}{\nu\sigma^2}\right)\right]$$

This distribution has variance $= \sigma^2\nu/(\nu-2)$ for $\nu > 2$ and kurtosis $= 3 + \nu/(\nu-4)$ for $\nu > 4$. The Cauchy is the special case $\nu = 1$, and the normal is the limiting case as $\nu \to \infty$. The case $\nu = 6$ provides a reasonable approximation to the Laplace. For this $\nu$, it and the Laplace have kurtosis of 6, and a Laplace $\sigma$ of $\sqrt{3}/2$ matches the variance of the t with $\sigma = 1$. As the odd moments are zero and only five moments exist for $\nu = 6$, the Laplace thus matches all existing moments of this t. Figure 1 graphs the densities.

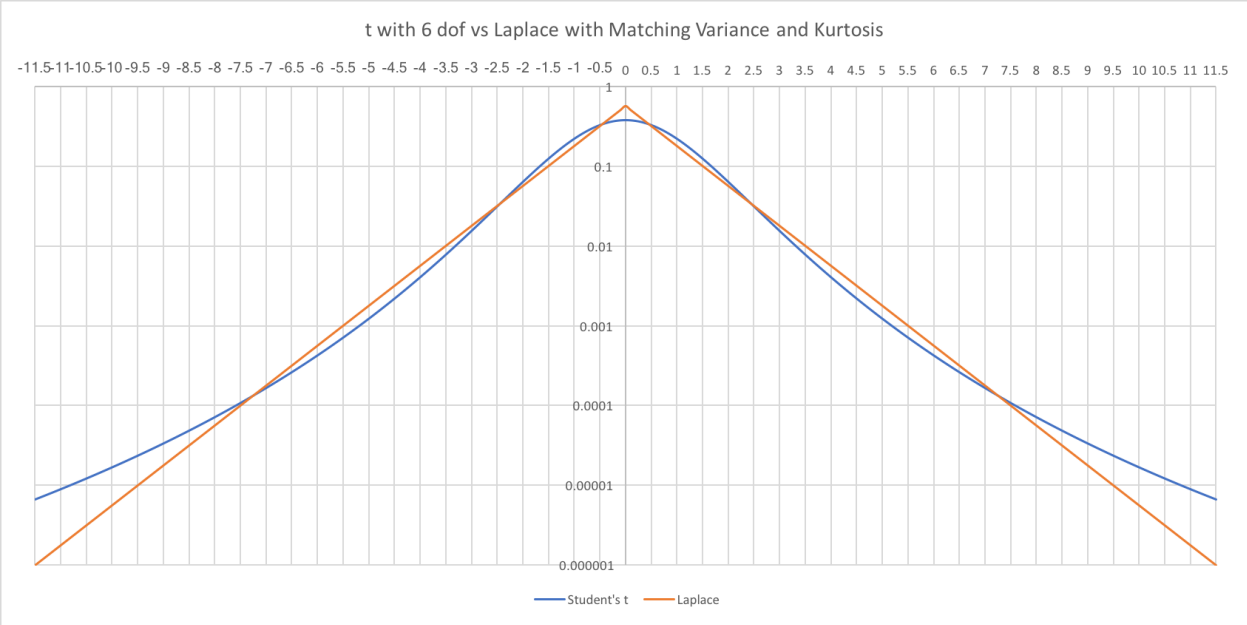



Figure 1: Students-t with 6 Degrees of Freedom and Laplace Densities

*3d.5 Estimating $\sigma$*

The fully Bayesian approach includes $\sigma$ as a parameter to be estimated. If it has a uniform prior with density $K$ over an appropriate interval, the log density in the Laplace case becomes:

$$log[f(\beta|\sigma)] = log(K) - log(2) - log(\sigma) - |\beta|/\sigma$$

In the estimation, $K$ drops out as a constant, but now the $log(\sigma)$ has to be included, since $\sigma$ is a parameter. The posterior mode with $n$ parameters then maximizes [loglikelihood $-n * log(\sigma) - \sum |\beta_j|/\sigma$]. This is one possibility for a classical lasso estimate of $\sigma$, and so $\lambda$, but the uniform prior is just one possible choice, so other values of $\lambda$ might be worth considering as well.

Initially I also tried putting a prior on the $\nu$ parameter of the scaled t distribution. That would provide a Bayesian estimate of the heaviness of tail the prior should have. Initial model runs always ended up with $\nu$ somewhere between 0.8 and 1.2 for the data here, which is pretty close to the value of 1.0 that produces the Cauchy. However estimates for the Laplace and Cauchy priors were very close for these small models, and lasso was used as an intermediate step, so the Laplace prior was used in the estimates in the example.

## 4 Example

As an example of this methodology, a loss triangle including exposures, counts, and amounts from Wüthrich (2003) is modeled. With the additive property of the Tweedie, only counts and amounts are needed to model the severity distributions across the cells, and with exposures, the frequency distributions also can be modeled. The triangles are shown in Tables 1 and 2.

Table 1: Development Triangle – Losses by AY and Lag

| AY | Lag: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 157.95 | 65.89 | 7.93 | 3.61 | 1.83 | 0.55 | 0.14 | 0.22 | 0.01 | 0.14 |
| 1 | 176.86 | 60.31 | 8.53 | 1.41 | 0.63 | 0.34 | 0.49 | 1.01 | 0.38 | 0.23 |
| 2 | 189.67 | 60.03 | 10.44 | 2.65 | 1.54 | 0.66 | 0.54 | 0.09 | 0.19 | 0 |
| 3 | 189.15 | 57.71 | 7.77 | 3.03 | 1.43 | 0.95 | 0.27 | 0.61 | 0 | 0 |
| 4 | 184.53 | 58.44 | 6.96 | 2.91 | 3.46 | 1.12 | 1.17 | 0 | 0 | 0 |
| 5 | 185.62 | 56.59 | 5.73 | 2.45 | 1.05 | 0.93 | 0 | 0 | 0 | 0 |
| 6 | 181.03 | 62.35 | 5.54 | 2.43 | 3.66 | 0 | 0 | 0 | 0 | 0 |
| 7 | 179.96 | 55.36 | 5.99 | 2.74 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 188.01 | 55.86 | 5.46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 2: Payment Counts by Lag and Exposures by AY

| AY | Lag: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Exposures |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 6229 | 3500 | 425 | 134 | 51 | 24 | 13 | 12 | 6 | 4 | 112.953 |
| 1 | 6395 | 3342 | 402 | 108 | 31 | 14 | 12 | 5 | 6 | 5 | 110.364 |
| 2 | 6406 | 2940 | 401 | 98 | 42 | 18 | 5 | 3 | 3 | 0 | 105.400 |
| 3 | 6148 | 2898 | 301 | 92 | 41 | 23 | 12 | 10 | 0 | 0 | 102.067 |
| 4 | 5952 | 2699 | 304 | 94 | 49 | 22 | 7 | 0 | 0 | 0 | 99.124 |
| 5 | 5924 | 2692 | 300 | 91 | 32 | 23 | 0 | 0 | 0 | 0 | 101.460 |
| 6 | 5545 | 2754 | 292 | 77 | 35 | 0 | 0 | 0 | 0 | 0 | 94.753 |
| 7 | 5520 | 2459 | 267 | 81 | 0 | 0 | 0 | 0 | 0 | 0 | 92.326 |
| 8 | 5390 | 2224 | 223 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 89.545 |

*4a Exploratory Analysis*

*4a.1 Design Matrix*

It is often useful before fitting the models to do some simple fits on an exploratory basis. In particular, multiple regression on the logs of the losses can reveal much of the structure of the data, before getting to better models. To set up multiple regression, the whole triangle has to be put into a single column as the dependent variable. An easy way to do this for this 9 x 10 triangle is to start with two columns for the row and column number of each cell - the first column having the number 1 ten times, the number 2 ten times, up to 9 ten times, with the second column repeating 1, 2,... 10 over and over. For later use, another column for the diagonal each cell is on can be added, set this column to row + column $-1$. In Excel then the index function can be used to put the loss values for each cell into a single column. Usually it will be convenient to put in a low value, maybe 0, or $-99$, for the cells not yet emerged – i.e., the lower triangle.

For the design matrix, a column parallel to the loss column is needed for each variable, starting with the constant, which is all 1's. Here for specificity the first row and column are not given parameters, so design matrix columns are needed for the variables for triangle rows 2 – 9 and columns 2 – 10. It usually helps to put in names for each column and the triangle row or column number above each name. Then the variable for a row parameter will be 1 if the row number equals the parameter number, and zero otherwise, and similarly for the column variables. Doing copy - paste values of all of this to another area then sorting by the loss size will put the not emerged cells at the bottom, and then a column of log losses can be added as the dependent variable. The dummy variables for the bottom triangle cells are there to make projections more convenient.

The same thing can be done with slope change dummy variables, but they are more complicated. Say a row parameter is the sum of its previous first differences, written as $p_w = \sum_{j=2}^{w} f_j$, and further that the first differences are sums of the previous second differences, so $f_j = \sum_{i=2}^{j} a_i$. Then $p_2 = f_2 = a_2$, $p_3 = f_2 + f_3 = 2a_2 + a_3$, $p_4 = f_2 + f_3 + f_4 = 3a_2 + 2a_3 + a_4$, etc. It comes down to that row parameter dummy $a_i$ for a cell in row $w$ gets value $1 + w - i$, with a minimum of zero. The same thing holds for column and diagonal parameters, using $u$ or $w + u - 1$ in place of $w$, so it can be used to fill out the design matrix, which is then sorted and the log column added. The entries for an observation in the design matrix are the number of times any slope change is added up for that observation.

Regressions can be done on both matrices. Calling the log column $y$ and the design matrix $x$, this is easy enough to do in Excel with matrix functions, giving the parameter vector $\beta = (x'x)^{-1}x'y$. It is even easier with regression functions, such as in the package Real Statistics. That, and in fact all packages used here, assume that the constant term is not in the design matrix, so from now on, $x$ refers to the design matrix without the constant term.

*4a.2 Regression*

Both the level and slope change regressions give the same overall fit – see Table 3 – but the t-statistics are different. Tables 4 and 5 show these for the two regressions. Usually

Table 3: Full Regression

| | |
|---|---|
| Multiple R | 0.978 |
| R Square | 0.956 |
| Adjusted R Square | 0.940 |
| Standard Error | 0.592 |

Table 4: Level Parameters and t-Statistics

| | cn | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| coef | 4.80 | 0.45 | 0.39 | 0.47 | 0.74 | 0.33 | 0.51 | 0.36 | 0.31 | -1.12 | -3.26 | -4.26 | -4.71 | -5.55 | -6.10 | -6.23 | -7.50 | -6.75 |
| s err | 0.28 | 0.26 | 0.28 | 0.29 | 0.30 | 0.32 | 0.34 | 0.37 | 0.41 | 0.28 | 0.28 | 0.29 | 0.30 | 0.32 | 0.34 | 0.37 | 0.41 | 0.48 |
| t sta | 17.4 | 1.71 | 1.41 | 1.64 | 2.46 | 1.03 | 1.52 | 0.97 | 0.75 | -4.01 | -11.7 | -14.7 | -15.5 | -17.4 | -17.9 | -16.9 | -18.2 | -1.0 |

t-statistics with absolute values $> 2$ are considered significant. By that measure, most of the row parameters in the levels regression are not significant, although the columns are. That might make this triangle a good candidate for the Cape Cod model. Parameter reduction will end up allowing some degree of variability among the rows, much like the Generalized Cape Cod of Gluck (1997).

The trend regression parameters are in general less significant, but a lower threshold for t may be appropriate in that adjacent parameters are strongly negatively correlated – raising one and lowering the next would offset for all but one row. Thus together they are more significant than they are individually. When a trend change is low, that means that the previous trend continues. The a2 parameter is probably significant, which would show a general upward trend from the first row. The column trend changes are significant in the beginning, with some fluctuation in direction, then lose significance, which would mean a continuing trend.

*4a.3 Lasso*

The design matrix can feed right into lasso software to get a start on parameter reduction. Illustrated here is the R package glmnet. The data $y$ and the design matrix $x$ are put in text files swissy.txt and swissx.txt first. This R code sets up and runs glmnet, given that it has already been installed. Standardization is turned off because the design matrix consists of dummy variables that count how many times a slope change is added in.

```
library(glmnet)
  y = scan('swissy.txt')
  x = read.table('swissx.txt', header = FALSE)
  x = as.matrix(x)
  N = length(y)
```

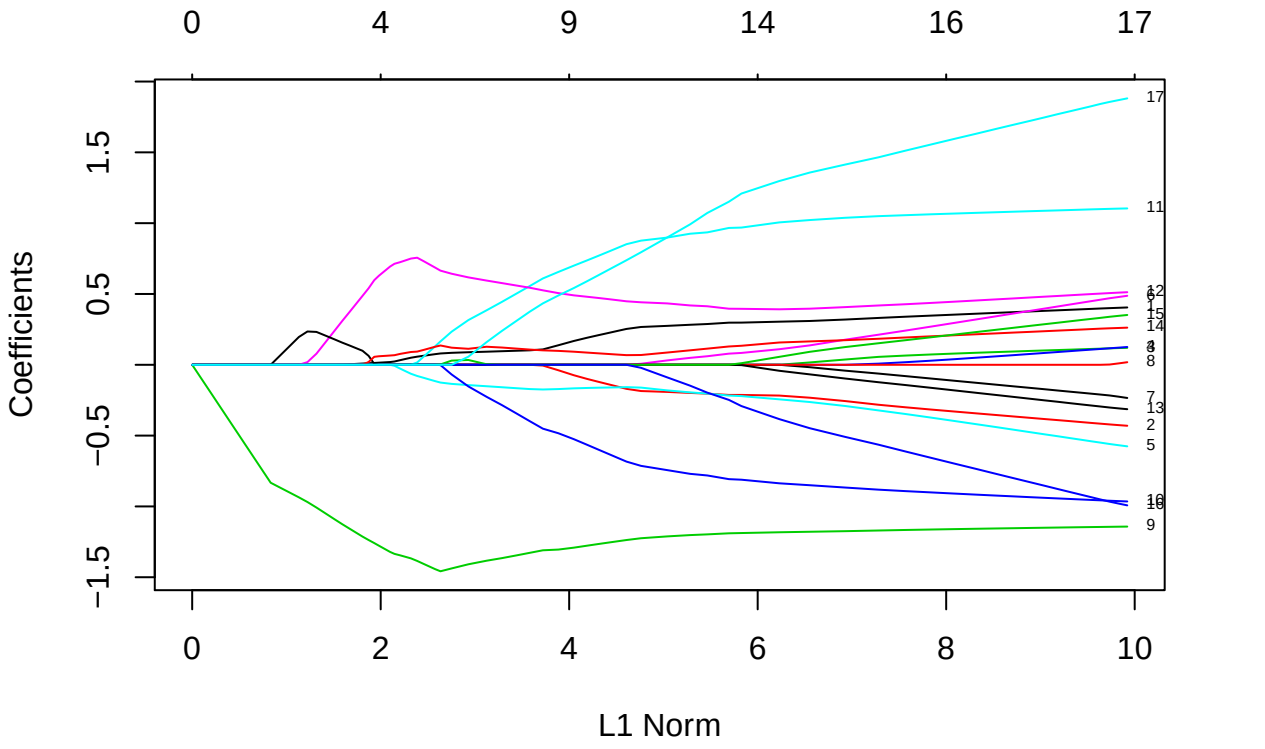
Table 5: Trend Change Parameters and t-Statistics

| | cn | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| coef | 4.80 | 0.45 | -0.52 | 0.15 | 0.19 | -0.68 | 0.60 | -0.34 | 0.11 | -1.12 | -1.01 | 1.13 | 0.56 | -0.39 | 0.28 | 0.42 | -1.13 | 2.01 |
| s err | 0.28 | 0.26 | 0.47 | 0.49 | 0.52 | 0.55 | 0.60 | 0.66 | 0.74 | 0.28 | 0.48 | 0.49 | 0.52 | 0.55 | 0.60 | 0.66 | 0.74 | 0.86 |
| t sta | 17.4 | 1.71 | -1.11 | 0.30 | 0.36 | -1.24 | 1.00 | -0.52 | 0.14 | -4.01 | -2.10 | 2.30 | 1.08 | -0.70 | 0.47 | 0.65 | -1.54 | 2.33 |

```
  U = ncol(x)
fit1 = glmnet(x, y, standardize = FALSE)
```

The program estimates the parameters for up to 100 values of $\lambda$, depending on some internal settings. This function prints out a graph of the parameter values as $\lambda$ decreases, going from left to right, with the variables numbered 1–19. The top axis is the number of non-zero parameters, and the bottom is the L1 norm, $\sum |\beta_j|$, both of which increase as $\lambda$ decreases.

```
plot(fit1, label=TRUE)
```




The parameters can increase and decrease as $\lambda$ changes since they are negatively correlated, and thus to some degree can substitute for each other. Variable 9 at the bottom is the parameter for the second column, which is a significant drop, and it is the last one to leave the model as $\lambda$ increases.

In the next block of R code, print(fit1) calculates and prints out 3 columns (not shown) for each $\lambda$: df is number of non-zero parameters, %Dev is R-squared in this regression case, and then $\lambda$ itself, in decreasing order of $\lambda$, increasing order of df. I call these dof, rsq, and lambda, and use them to calculate NLL $+\lambda \sum |\beta_j| - k * log\lambda$, the quantity to be minimized if $\lambda$ has a uniform prior, for each row. That is called min and is printed out with $\lambda$ next.

```
  answer = print(fit1)
  lambda = answer[,3]
  rsq = answer[,2]
  dof = answer[,1]
  sst = sum((y-mean(y))^2)
```

```
  ssr = (1-rsq)*sst
  sigsq = ssr/(N-dof)
  NLL = N*log(sigsq)+ssr/2/sigsq
  beta = fit1[[2]]
  h=dim(answer)[1]
  k=dim(x)[2]
  L1 = c(1:h)
  for(i in 1:h) L1[i] = sum(abs(beta[2:k,i])) #sum of absolute values
  min =  NLL-dof*log(lambda)+lambda*L1
```

```
 lambda
```

```
##   [1] 5.6980000 5.1920000 4.7310000 4.3110000 3.9280000 3.5790000 3.2610000
##   [8] 2.9710000 2.7070000 2.4670000 2.2480000 2.0480000 1.8660000 1.7000000
##  [15] 1.5490000 1.4120000 1.2860000 1.1720000 1.0680000 0.9729000 0.8865000
##  [22] 0.8077000 0.7360000 0.6706000 0.6110000 0.5567000 0.5073000 0.4622000
##  [29] 0.4212000 0.3837000 0.3496000 0.3186000 0.2903000 0.2645000 0.2410000
##  [36] 0.2196000 0.2001000 0.1823000 0.1661000 0.1514000 0.1379000 0.1257000
##  [43] 0.1145000 0.1043000 0.0950500 0.0866100 0.0789200 0.0719100 0.0655200
##  [50] 0.0597000 0.0543900 0.0495600 0.0451600 0.0411500 0.0374900 0.0341600
##  [57] 0.0311300 0.0283600 0.0258400 0.0235500 0.0214500 0.0195500 0.0178100
##  [64] 0.0162300 0.0147900 0.0134700 0.0122800 0.0111900 0.0101900 0.0092870
##  [71] 0.0084620 0.0077100 0.0070250 0.0064010 0.0058320 0.0053140 0.0048420
##  [78] 0.0044120 0.0040200 0.0036630 0.0033380 0.0030410 0.0027710 0.0025250
##  [85] 0.0023000 0.0020960 0.0019100 0.0017400 0.0015860 0.0014450 0.0013160
##  [92] 0.0011990 0.0010930 0.0009958 0.0009073 0.0008267 0.0007533 0.0006864
##  [99] 0.0006254 0.0005698
```

```
min
```

```
##   [1] 139.1546578 128.5099482 119.2727479 110.2654986 101.5464556
##   [6]  93.1729916  85.1716841  77.6168526  70.5099937  63.9140876
##  [11]  57.8211853  52.2532368  47.2379003  42.7012551  38.6963721
##  [16]  35.1559199  32.0395586  29.3520893  27.0057140  25.0460777
##  [21]  23.3507511  21.9024199  20.6774497  19.6905216  18.8214565
##  [26]  18.1204360  17.5483675  17.0637192  16.7143903  16.4110022
##  [31]  16.1538759  15.9435496  15.8276049  15.7128830  13.4839918
##  [36]   9.3518397   5.5858747   4.4655629   1.1074878  -2.3413246
##  [41]  -5.3021017  -5.2037415  -7.4618867  -9.2628828 -10.8067730
##  [46]  -9.2122464 -10.2469277 -14.1664652 -14.8062671 -15.2758039
##  [51] -15.5672680 -12.3776308 -12.9763013 -13.3960749  -9.7473662
##  [56] -10.0955550 -10.3551425 -10.4168976 -10.4854673  -1.7043340
##  [61]  -2.0199045   1.9086412   1.3079381  -3.7673173  -4.0607240
##  [66]  -4.1319612   0.8366980   0.8321761   0.9443275   1.1637626
##  [71]   1.5035898   1.8398307   2.4239278   2.8805064  15.0096378
##  [76]  15.2633089  15.7676314  16.3980934  23.1392631  30.4838019
```

```
##  [81]  37.3736464  44.8752963  45.7278083  53.3901917  54.6076325
##  [86]  55.6780471  56.8859274  58.2370121  59.5790237  60.9277201
##  [91]  62.2826403  63.6308198  64.9699640  66.4593997  67.8064965
##  [96]  69.2943538  70.6396882  80.2739272  81.8555927  83.4379203
```

The minimum of this function is at the $51^{st}$ cell where $\lambda = 0.05439$. Since the uniform prior is only one possible choice, other values of $\lambda$ should be considered as well. Adding a few more variables is a sound choice, as they can be eliminated later in Bayesian lasso if they are not needed. The $59^{th}$ value is at the end of the area of low values of min, with $\lambda = 0.02584$. Cross validation is done in a function called cv.glmnet, which produces its own target range for $\lambda$ between lambda.min and lambda.1se.

```
cvfit = cv.glmnet(x, y, standardize = FALSE)
cvfit$lambda.1se
```

```
## [1] 0.1256568
```

```
cvfit$lambda.min
```

```
## [1] 0.01019241
```

The variables and coefficients for selected values of $\lambda$ are given by the coef function.

```
coef(cvfit, s=c(0.01118616, 0.02584, 0.05439, 0.11449))
```

```
## 18 x 4 sparse Matrix of class "dgCMatrix"
##                       1             2             3            4
## (Intercept)  4.96801195  5.014613e+00  4.960018e+00  4.775570560
## V1           0.13797768  7.511297e-02  1.940272e-02  0.003674287
## V2          -0.03868051  .             .             .
## V3           .           .             .             .
## V4           .           .             .             .
## V5          -0.17133069 -1.153241e-01 -5.443126e-06  .
## V6           .           .             .             .
## V7           .           .             .             .
## V8           .           .             .             .
## V9          -1.30479742 -1.442692e+00 -1.321330e+00 -1.157271291
## V10         -0.48333616 -1.432428e-06  .             .
## V11          0.65565734  1.302843e-01  .             .
## V12          0.50612353  6.865177e-01  6.936972e-01  0.373278485
## V13          .           .             .             0.136026729
## V14          0.09814611  1.273081e-01  6.540153e-02  .
## V15          .           1.700821e-06  .             .
## V16          .           .             .             .
## V17          0.48927296  .             .             .
```

Bayesian lasso has several advantages over classical lasso, including giving a sample distribution of parameters for risk analysis, being able to include a distribution of values of $\lambda$, and having a goodness of fit measure looic. It does not eliminate variables, but it provides a probability

range for each parameter, and those near zero with a wide range of positive and negative values can be eliminated, which usually improves looic. Estimation is faster if some variables are eliminated before running it, however, and lasso output can give some guidance to that. The second and third columns of coefficients have the same non-zero variables, except for V10, V11, and V15. Keeping the variables in the second column except for V!5 leaves a2, a6, b2, b3, b4, b5 and b7, plus the constant. These are used in a reduced design matrix in Stan to do the MCMC estimation. See Table 6.

*4b Aggregate Triangle*

Stan contains a programming language for building models to be estimated by MCMC. Below is the code used for estimating the gamma distribution with fixed $b$, so with variance proportional to mean, from the reduced design matrix. Most of the code is setup – declaring the variable types and dimensions, etc. Now the $y$ variable is in monetary units, but the model is still fit in logs. The cell gamma mean is the exponentiation of the sum of the log parameters for that cell, which makes the parameters slightly different than they would be for estimating the mean of the log.

```
data {
  int N;     // number of obs
  int U;    // number of variables
  vector[N] y;
  matrix[N,U] x1;          //design matrix with U columns
}
parameters {  // all except v will get uniform prior, which is default
 real<lower=4, upper=16> cn;             //constant term, starting in known range
  vector[U] v;                         // the parameters
 real<lower=-5, upper = -0.2> logs;   //log of s, related to lambda, not too high
 real<lower=-20, upper = 20> logbeta;   //log of beta
}
transformed parameters {
 real beta;
 real s;                            // shrinkage parameter, like lambda
 vector[N] alpha;                    //fitted means
 beta = exp(logbeta); //for positive parameter, uniform on log is like 1/X
 s = exp(logs);  // 1/X gives more weight to lower values, which is good if X not big
 alpha = exp(x1*v+cn)*beta;
}
model {  // gives priors for those not assumed uniform. Choose this one for lasso.
   for (i in 1:U)  v[i] ~ double_exponential(0, s);  // more weight to close to 0
for (j in 1:N) y[j] ~ gamma(alpha[j], beta);
}
generated quantities { //outputs log likelihood for testing purposes
  vector[N] log_lik;
for (j in 1:N) log_lik[j] = gamma_lpdf(y[j] | alpha[j],beta);
```

Table 6: Reduced Regression Variables

| Loss | Row | Col | y | cn | a2 | a6 | b2 | b3 | b4 | b5 | b7 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 189.67 | 3 | 1 | 5.245 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 189.15 | 4 | 1 | 5.243 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 188.01 | 9 | 1 | 5.236 | 1 | 8 | 4 | 0 | 0 | 0 | 0 | 0 |
| 185.62 | 6 | 1 | 5.224 | 1 | 5 | 1 | 0 | 0 | 0 | 0 | 0 |
| 184.53 | 5 | 1 | 5.218 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 181.03 | 7 | 1 | 5.199 | 1 | 6 | 2 | 0 | 0 | 0 | 0 | 0 |
| 179.96 | 8 | 1 | 5.193 | 1 | 7 | 3 | 0 | 0 | 0 | 0 | 0 |
| 176.86 | 2 | 1 | 5.175 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 157.95 | 1 | 1 | 5.062 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 65.89 | 1 | 2 | 4.188 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 62.35 | 7 | 2 | 4.133 | 1 | 6 | 2 | 1 | 0 | 0 | 0 | 0 |
| 60.31 | 2 | 2 | 4.099 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 60.03 | 3 | 2 | 4.095 | 1 | 2 | 0 | 1 | 0 | 0 | 0 | 0 |
| 58.44 | 5 | 2 | 4.068 | 1 | 4 | 0 | 1 | 0 | 0 | 0 | 0 |
| 57.71 | 4 | 2 | 4.055 | 1 | 3 | 0 | 1 | 0 | 0 | 0 | 0 |
| 56.59 | 6 | 2 | 4.036 | 1 | 5 | 1 | 1 | 0 | 0 | 0 | 0 |
| 55.86 | 9 | 2 | 4.023 | 1 | 8 | 4 | 1 | 0 | 0 | 0 | 0 |
| 55.36 | 8 | 2 | 4.014 | 1 | 7 | 3 | 1 | 0 | 0 | 0 | 0 |
| 10.44 | 3 | 3 | 2.346 | 1 | 2 | 0 | 2 | 1 | 0 | 0 | 0 |
| 8.53 | 2 | 3 | 2.144 | 1 | 1 | 0 | 2 | 1 | 0 | 0 | 0 |
| 7.93 | 1 | 3 | 2.071 | 1 | 0 | 0 | 2 | 1 | 0 | 0 | 0 |
| 7.77 | 4 | 3 | 2.050 | 1 | 3 | 0 | 2 | 1 | 0 | 0 | 0 |
| 6.96 | 5 | 3 | 1.940 | 1 | 4 | 0 | 2 | 1 | 0 | 0 | 0 |
| 5.99 | 8 | 3 | 1.790 | 1 | 7 | 3 | 2 | 1 | 0 | 0 | 0 |
| 5.73 | 6 | 3 | 1.746 | 1 | 5 | 1 | 2 | 1 | 0 | 0 | 0 |
| 5.54 | 7 | 3 | 1.712 | 1 | 6 | 2 | 2 | 1 | 0 | 0 | 0 |
| 5.46 | 9 | 3 | 1.697 | 1 | 8 | 4 | 2 | 1 | 0 | 0 | 0 |
| 3.66 | 7 | 5 | 1.297 | 1 | 6 | 2 | 4 | 3 | 2 | 1 | 0 |
| 3.61 | 1 | 4 | 1.284 | 1 | 0 | 0 | 3 | 2 | 1 | 0 | 0 |
| 3.46 | 5 | 5 | 1.241 | 1 | 4 | 0 | 4 | 3 | 2 | 1 | 0 |
| 3.03 | 4 | 4 | 1.109 | 1 | 3 | 0 | 3 | 2 | 1 | 0 | 0 |
| 2.91 | 5 | 4 | 1.068 | 1 | 4 | 0 | 3 | 2 | 1 | 0 | 0 |
| 2.74 | 8 | 4 | 1.008 | 1 | 7 | 3 | 3 | 2 | 1 | 0 | 0 |
| 2.65 | 3 | 4 | 0.975 | 1 | 2 | 0 | 3 | 2 | 1 | 0 | 0 |
| 2.45 | 6 | 4 | 0.896 | 1 | 5 | 1 | 3 | 2 | 1 | 0 | 0 |
| 2.43 | 7 | 4 | 0.888 | 1 | 6 | 2 | 3 | 2 | 1 | 0 | 0 |
| 1.83 | 1 | 5 | 0.604 | 1 | 0 | 0 | 4 | 3 | 2 | 1 | 0 |
| 1.54 | 3 | 5 | 0.432 | 1 | 2 | 0 | 4 | 3 | 2 | 1 | 0 |
| 1.43 | 4 | 5 | 0.358 | 1 | 3 | 0 | 4 | 3 | 2 | 1 | 0 |
| 1.41 | 2 | 4 | 0.344 | 1 | 1 | 0 | 3 | 2 | 1 | 0 | 0 |
| 1.17 | 5 | 7 | 0.157 | 1 | 4 | 0 | 6 | 5 | 4 | 3 | 1 |
| 1.12 | 5 | 6 | 0.113 | 1 | 4 | 0 | 5 | 4 | 3 | 2 | 0 |
| 1.05 | 6 | 5 | 0.049 | 1 | 5 | 1 | 4 | 3 | 2 | 1 | 0 |
| 1.01 | 2 | 8 | 0.010 | 1 | 1 | 0 | 7 | 6 | 5 | 4 | 2 |
| 0.95 | 4 | 6 | -0.051 | 1 | 3 | 0 | 5 | 4 | 3 | 2 | 0 |
| 0.93 | 6 | 6 | -0.073 | 1 | 5 | 1 | 5 | 4 | 3 | 2 | 0 |
| 0.66 | 3 | 6 | -0.416 | 1 | 2 | 0 | 5 | 4 | 3 | 2 | 0 |
| 0.63 | 2 | 5 | -0.462 | 1 | 1 | 0 | 4 | 3 | 2 | 1 | 0 |
| 0.61 | 4 | 8 | -0.494 | 1 | 3 | 0 | 7 | 6 | 5 | 4 | 2 |
| 0.55 | 1 | 6 | -0.598 | 1 | 0 | 0 | 5 | 4 | 3 | 2 | 0 |
| 0.54 | 3 | 7 | -0.616 | 1 | 2 | 0 | 6 | 5 | 4 | 3 | 1 |
| 0.49 | 2 | 7 | -0.713 | 1 | 1 | 0 | 6 | 5 | 4 | 3 | 1 |
| 0.38 | 2 | 9 | -0.968 | 1 | 1 | 0 | 8 | 7 | 6 | 5 | 3 |
| 0.34 | 2 | 6 | -1.079 | 1 | 1 | 0 | 5 | 4 | 3 | 2 | 0 |
| 0.27 | 4 | 7 | -1.309 | 1 | 3 | 0 | 6 | 5 | 4 | 3 | 1 |
| 0.23 | 2 | 10 | -1.470 | 1 | 1 | 0 | 9 | 8 | 7 | 6 | 4 |
| 0.22 | 1 | 8 | -1.514 | 1 | 0 | 0 | 7 | 6 | 5 | 4 | 2 |
| 0.19 | 3 | 9 | -1.661 | 1 | 2 | 0 | 8 | 7 | 6 | 5 | 3 |
| 0.14 | 1 | 7 | -1.966 | 1 | 0 | 0 | 6 | 5 | 4 | 3 | 1 |
| 0.14 | 1 | 10 | -1.966 | 1 | 0 | 0 | 9 | 8 | 7 | 6 | 4 |
| 0.09 | 3 | 8 | -2.408 | 1 | 2 | 0 | 7 | 6 | 5 | 4 | 2 |
| 0.01 | 1 | 9 | -4.605 | 1 | 0 | 0 | 8 | 7 | 6 | 5 | 3 |

```
}
```

Stan parameterizes the gamma with parameter $beta = 1/b$, and $alpha_{w,u}$ is set so the mean is $alpha_{w,u}/beta$. Stan also uses the parameter $s = 1/\lambda$ for the Laplace = double exponential prior, and here $s$ is taken as a parameter to be estimated. Unless otherwise stated, all parameters are taken to have uniform priors over their defined ranges. The transformed parameters are intermediate calculations and do not have priors and are not estimated. The generated quantities section creates additional outputs, here the loglikelihood for each point for each parameter sample for the looic calculation.

The range defined for the constant was informed by the lasso result but is wider than it needs to be. $Beta$ is defined by giving its log a wide uniform prior. That is similar to giving it a prior of $1/x$. This is appropriate for a parameter for which it or its reciprocal could be used. $1/beta$ would have the same prior – its log would be uniform on the real line (limited by $\pm 10^{310}$ or so by double precision numbers). Also the $1/x$ prior often gives the classical unbiased estimate for a positive parameter. This is similar for $s$, but it was given a smaller range. Too high a value can get into convergence problems. After some experimentation, a6 was replaced by a4, which gave a better fit by looic and NLL.

Output available includes a graph of (0.05, 0.95) and (0.2, 0.8) percentile ranges for the parameters. See Figure 2. This is where parameters that are near zero with large positive and negative ranges can be reviewed for removal from the model. None of these are like that. The resulting row and column parameters are compared to those from the full lognormal regression in Figure 3. Not shown is the $s$ parameter, which is in the range [0.12, 0.14].

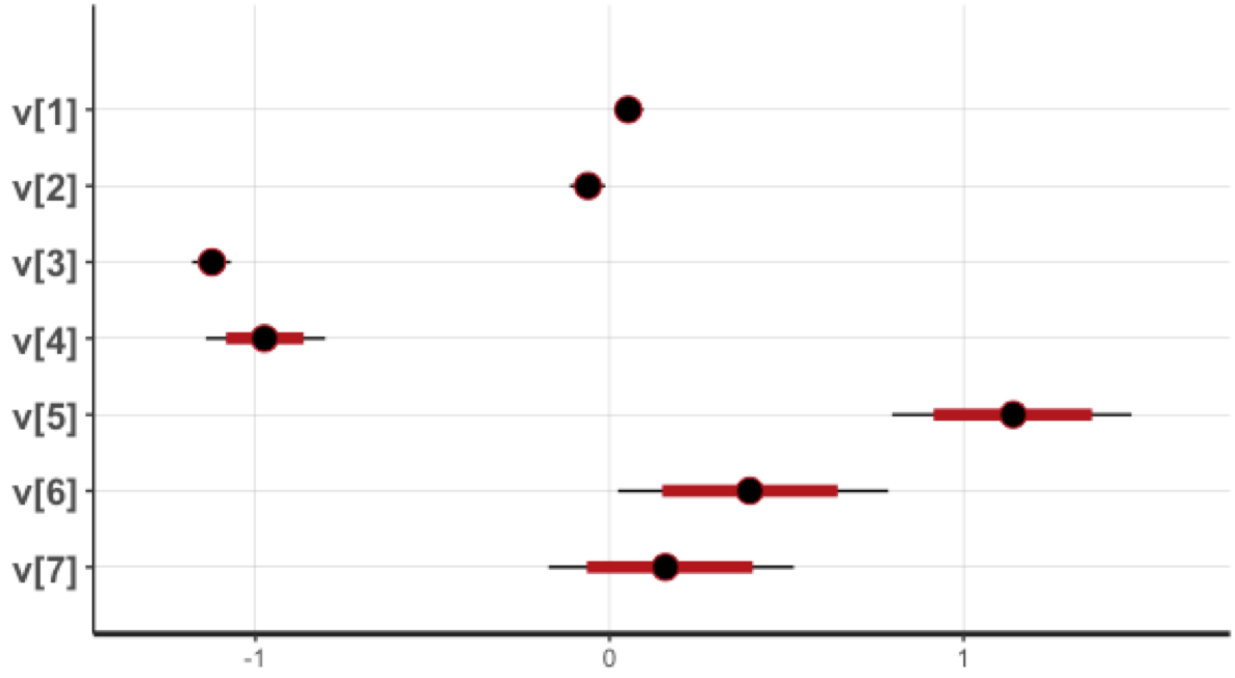



Figure 2: Parameter Ranges Stan Gamma Fit

Normal-k, GiG, gamma, and Weibull-k distributions were fit to the triangle. All have very similar row and column parameters but different looic, due to the different distribution shapes. Table 7 shows looic, the NLL, and their difference, the parameter penalty. All except the gamma have a parameter for the power in the variance $= s * mean^k$ relationship, but here all those powers came out very close to 1.0. The gamma was fit with the $b$ parameter constant across the cells, so it also has the power $k = 1$ implicitly. It thus saves a parameter. The GiG has one more parameter for the percent normal, which was 30%.

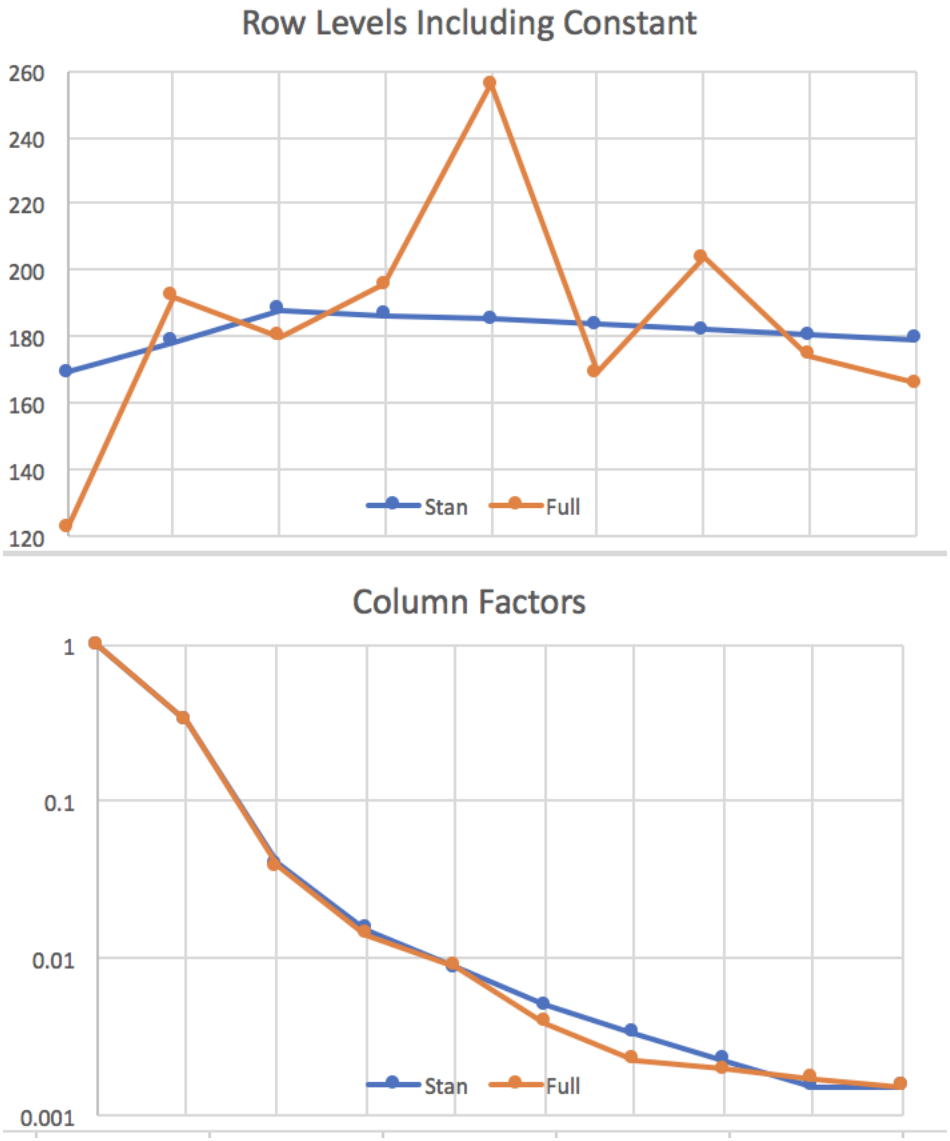



Figure 3: Row and Column Parameters for Gamma in Stan and Full Regression Lognormal

The best fitting distribution was the Weibull-k, but it is not significantly better than the gamma. It is the only one that does not have skewness proportional to CV, and the extra variability in skewness apparently helped for this data set. It seems that the zero skewness of the normal-k did not work well for this data.

Table 7: Aggregate Triangle Model Fits

| Distribution | looic | NLL | Penalty |
|---|---|---|---|
| Normal-k | 111.2 | 98.9 | 12.3 |
| GiG | 106.2 | 94.7 | 11.5 |
| Gamma | 102.4 | 92.1 | 10.3 |
| Weibull-k | 101.8 | 92.3 | 9.5 |

The Weibull-k and gamma fits had about the same mean and CV by cell, but the skewnesses were different. Figure 4 graphs the common CV and the two skewnesses by lag for the second row, the last one that had all columns. Because the rows are all pretty similar, this graph would look about the same for any row. The gamma skewness is twice the CV, but the Weibull's is consistently lower. This appears to provide a better representation of the observations under the row-column model. Possibly a Tweedie with $p < 2$ would fit better than the gamma, but its skewness would be positive, so would be more like the gamma than the Weibull-k.

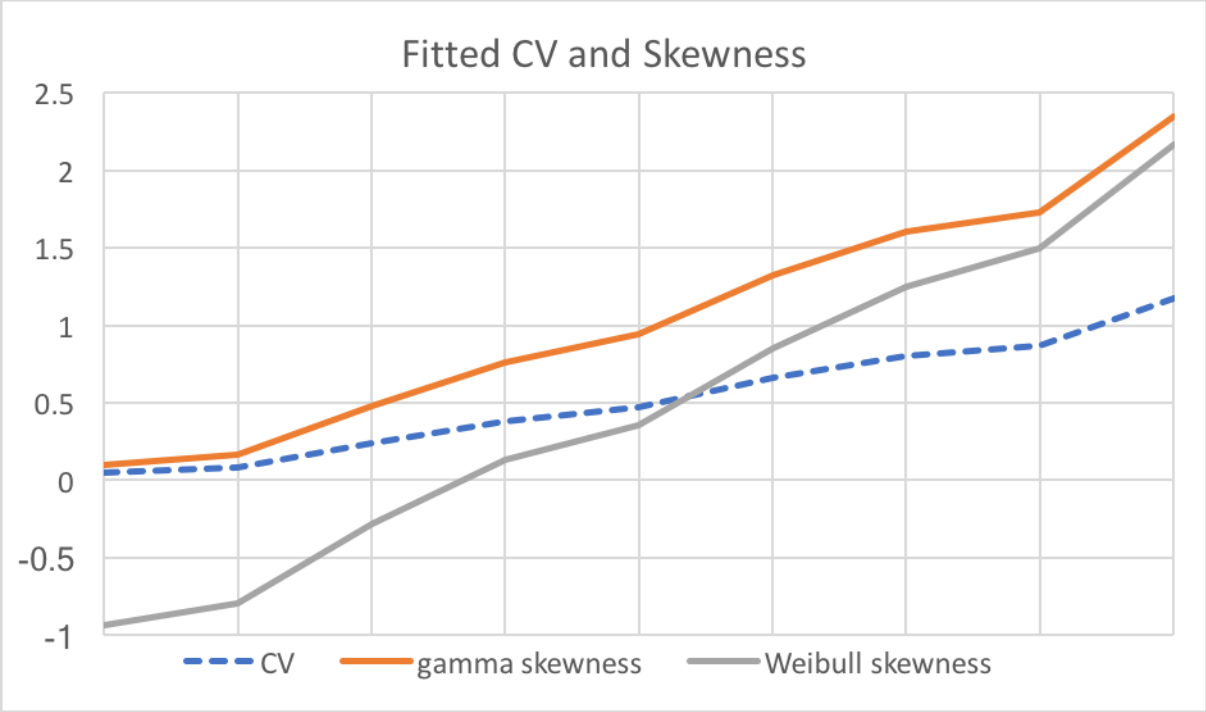



Figure 4: Fitted CV and Skewness for Gamma and Weibull-k Fits

*4c Severity*

The data does not have individual payment observations, but due to the additive property of the Tweedie, the counts and total payments in a cell are enough to model the severity distribution. Severity is typically modeled with a constant CV across the cells. That requires the Tweedie severity $a$ parameter to be constant. Each cell gets its own $b$ parameter from the

row-column model. Then the losses in a cell are modeled as Tweedie in $a$ times the number of payments in the cell and the $b$ for the cell, with any $p$. Here $p = 2$ and $p = 3$, so the gamma and inverse Gaussian, are fit. The model with constant $b$, so variance proportional to mean, was tested for comparison. If severity is normal-k distributed in $\mu_{w,u}, s, k$, the payment total is distributed normal with mean = $\mu_{w,u}$*(counts) and variance = $s\mu_{w,u}^{k}$*(counts).

The starting point was to use the same seven variables that were optimal for aggregate losses. For the gamma distribution, the parameter graph with (5%, 95%) and (20%, 80%) ranges is shown in Figure 5. From the graph, v[6], which is the coefficient for the column 5 slope change, has mean close to zero and a wide range. That is the sort of graph that indicates that a parameter is not needed. Eliminating it improved the looic. The remaining variables are the slope changes for rows 2 and 4, and for columns 2, 3, 4, and 7.

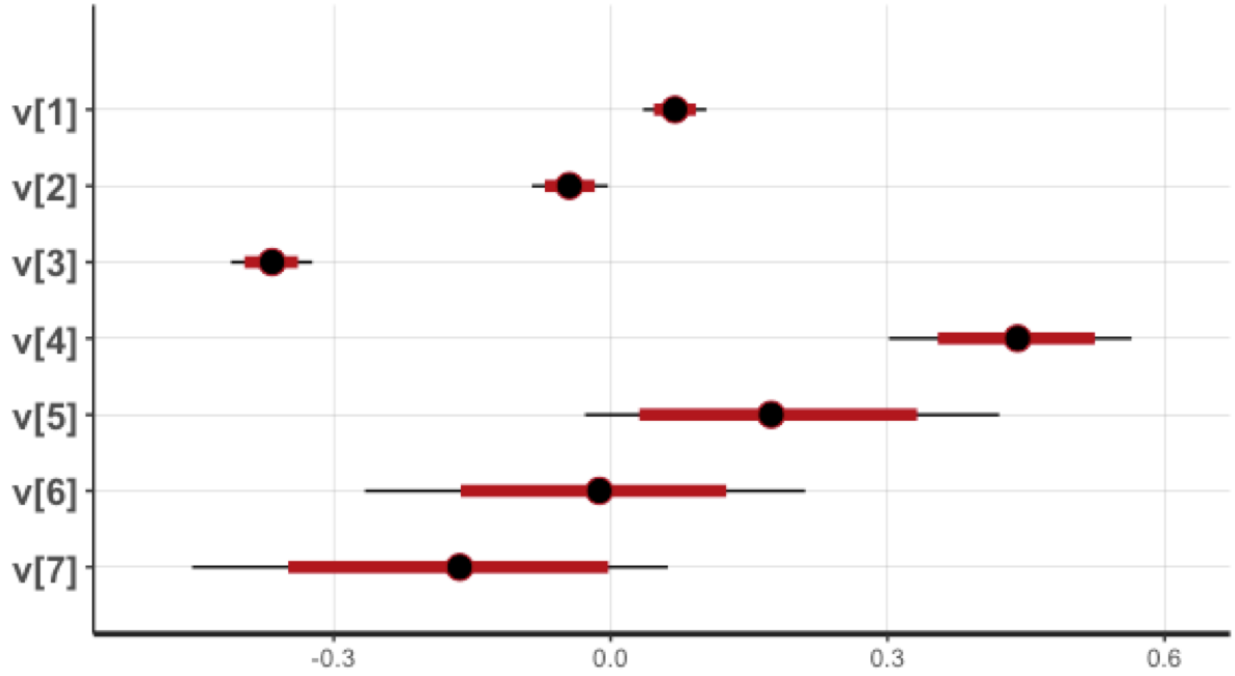



Figure 5: Gamma Severity Parameter Ranges – 7 Variables

The design matrix for that data was used for the three distributions. The gamma with $a$ fixed was the best fit. The inverse Gaussian was actually slightly better with $b$ held constant. Fit measures and the fitted moments are in Table 8. For these distributions, variance is proportional to a power of the mean and the skewness is a multiple of the CV. From the table, the power appears to bear an inverse relationship to the skewness – the more skewed distributions have the lowest power.

Figure 6 graphs the resulting level factors (not differences) for the gamma and the inverse Gaussian. The column factors are indistinguishable for the two distributions. Severity is growing fairly steadily across the accident years, and is highest at the fifth lag. The raw severity mean is highest for the 7th and 8th columns but is highly volatile there.

It is not possible to use the R Tweedie package within Stan, but it can be used with a non-

Table 8: Severity Fit

| | Power | Skw/CV | Looic | Penalty | NLL |
|---|---|---|---|---|---|
| Normal-k | 3.2 | 0 | 97.2 | 11.6 | 85.6 |
| Gamma | 2.0 | 2 | 87.0 | 7.4 | 79.6 |
| Inverse Gaussian | 1.0 | 3 | 94.0 | 9.8 | 84.2 |

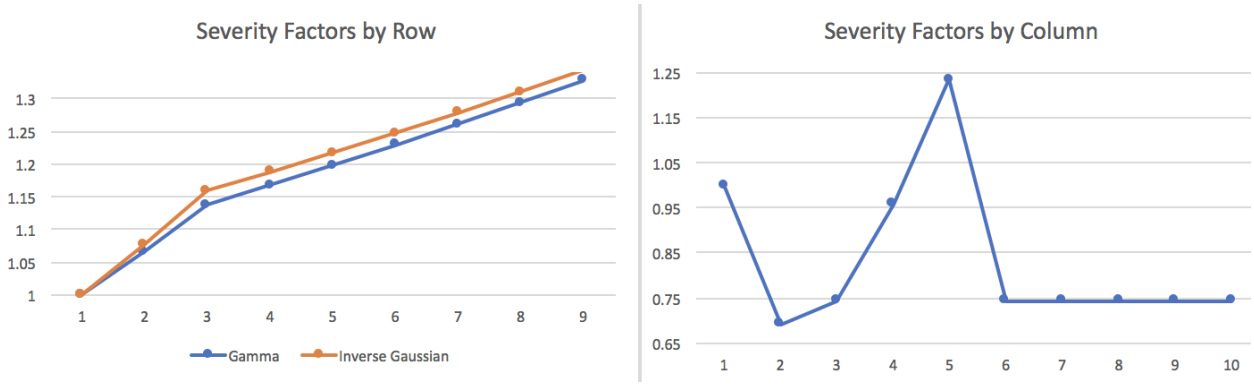



Figure 6: Row and Column Severity Level Parameters

linear optimizer, like optimx, to estimate parameters at the posterior mode with shrinkage priors. I tried that for the Tweedie with $a$ fixed across the cells using a Cauchy prior with $\sigma = 0.1$. That produced an estimate of $p = 2.1$. This is close to the gamma distribution value of $p = 2$.

*4d Frequency*

There are cell counts and AY exposures, so mean frequency in a cell is modeled with the row-column model, and the number of claims is modeled with its mean equal to the cell frequency mean times the row exposure. The Poisson distribution and two forms of the negative binomial were fit. NB1 is the one with variance proportional to the mean, and NB2 has variance a quadratic function of the mean.

The fit measures are shown in Table 9. The NB2 is clearly the best fit. Its row and column factors for six chains are graphed in Figure 7. Payment frequency is declining somewhat by row and sharply by column.

The PiG distribution was fit by maximizing the posterior using Nelder-Mead optimization starting with the parameters of the NB2. The PiG NLL was 272.1, so is a little worse than the NB2 if you assume the shrinkage is comparable. It is a more skewed distribution, so the NB2 appears to have enough skewness for this data.

## 5 Extensions of Row-Column Model

A few extensions of the basic row-column model are discussed for this methodology. The aggregate triangle with the gamma distribution is used with fixed $b$, so variance is proportional

Table 9: Frequency Fits

| Distribution | looic | NLL | Penalty |
|---|---|---|---|
| Poisson | 365.1 | 306.1 | 59.0 |
| NB1 | 302.8 | 283.8 | 19.0 |
| NB2 | 284.6 | 271.4 | 13.2 |

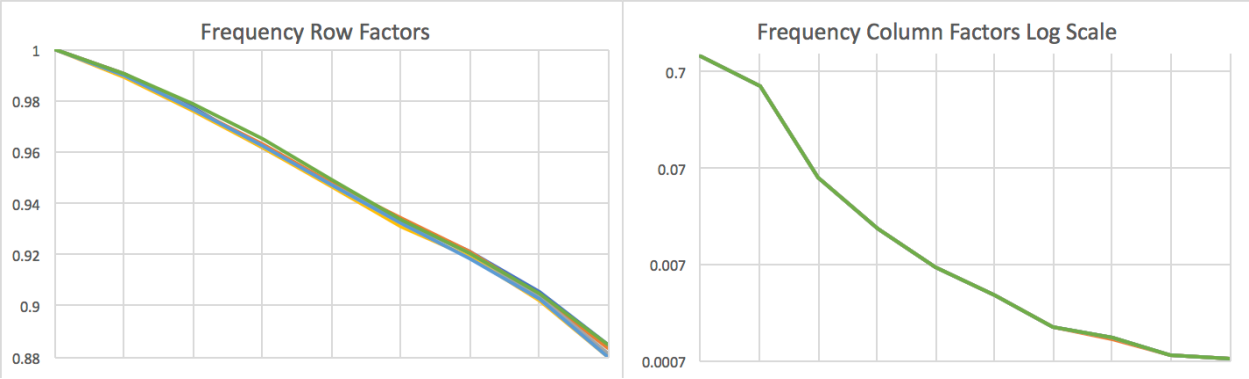



Figure 7: Frequency Row and Column Factors

to mean, as it is a good fitting model and its estimation is fast – one or two seconds typically.

*5a Additive Component*

Müller (2016) suggests expanding the multiplicative model with an additive component. He argues that some part of loss development is from late reported claims, and these could be more related to exposure than to losses already emerged. Any accident-year exposure variable, like premium or policy count, would be the starting point. This would be multiplied by coefficients by column, and added to the row*column mean for the cell. Even a constant for all the rows could be used if exposure is not available. Also the coefficients could be from a curve fit across the columns. The resulting model for the cell mean $\mu_{w.u}$ would be:

$$\mu_{w,u} = A_w B_u C + D_u E_w$$

where $E_w$ is the exposure for AY $w$ (or just a constant) and $D_u$ are column parameters.

Having the column factor on a piecewise-linear curve with slope changes shrunk with a shrinkage prior would be consistent with the approach here. This could end up using few parameters so could be as parsimonious as a fitted curve, but more flexible in shape. The idea that this comes from late-reported claims would suggest that the coefficients all be positive, but another possible justification is that this is an additive term to adjust for bias. Then it would not necessarily have to remain positive. Here a positive factor by column is fit and applied to AY exposures, with the result added to the row*column means. This can be done in logs with another design matrix for the slope changes for the column parameters. This design matrix would be the same as the column parameter design matrix, except it would include a dummy variable for the first column as well.

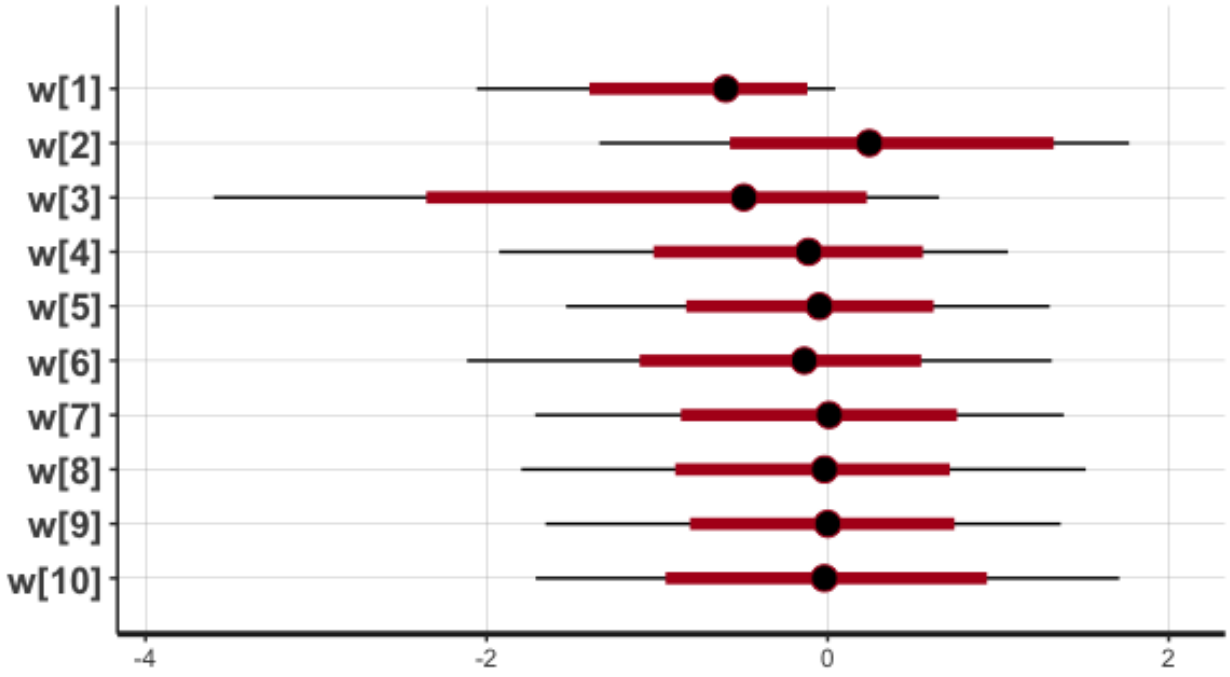



Figure 8: Column Parameter Ranges for Exposure Log Slope Change Variables

In the code below, w is the vector of coefficients for the exposure column parameters, and x_expo is the corresponding design matrix. The exposure by row is in a vector expo, but this is divided by 10,000 to put it on a more useful scale. The alpha by cell is built up from the row-column mean, the exposure component, and beta. Losses are assumed to be gamma distributed.

```
 alpha = exp(x_expo*w); //expo design matrix for log 2nd diff * parameters
 for (i in 1:N) alpha[i] = alpha[i]*expo[i]/10000; // multiply by row exposure
 alpha = (alpha + exp(x1*v+cn))*beta; // add in row-col mean to give mean, alpha
}
model {  // gives priors for those not assumed uniform. This one for lasso.
   for (i in 1:U)  v[i] ~ double_exponential(0, s);
   for (i in 1:V)  w[i] ~ double_exponential(0, s);
for (j in 1:N) y[j] ~ gamma(alpha[j], beta);
}
```

Resulting parameter ranges are in Figure 8. Most of these are centered near zero, with wide ranges. Keeping just the first three gave a good fit to the triangle, with looic and NLL of 99.9 and 90.1, compared to 102.4 and 92.1 for the row-column gamma model. There are nominally three extra parameters here, but the loo parameter penalty was actually slightly less, at 9.9, compared to 10.3 for the base model. The penalty comes from the out-of-sample fit, which was apparently better with the exposures included. Perhaps the exposures allowed more shrinkage of the other parameters. The exposure factor was 0.653 for the first column and 0.606 for the second. After that it falls by a multiple of 0.545 for each subsequent column.

This is believable as an IBNR effect, as it is strongest early on then practically disappears by the end.

*5b Calendar-Year Effects*

Inflation can operate on payment years more than on accident years per se, as jury awards and building costs are typically based on price levels at the time of payment. This can be modeled by adding calendar-year factors to the model, or by using them instead of accident-year factors. With just diagonal and column factors, this is called the separation model after G. Taylor (1977).

Another type of calendar-year effect comes from changes in loss processing, which could speed up or slow down payments in just a few diagonals. Only one or two diagonal parameters could model this. Such effects would not need to be projected, but adjusting for them could reduce estimation errors on the other parameters. G. G. Venter (2007) applies that to the triangle of G. C. Taylor and Ashe (1983), for example.

Either way, the mean for the multiplicative model with CY effects included is

$$\mu_{w,u} = A_w B_u G_{w+u-1} C$$

The cell in row $w$ and column $u$ will be on diagonal $w+u-1$, assuming the columns start at 1 and rows and diagonals start with the same number. $G_{w+u-1}$ is thus the trend factor, and in this framework it is a cumulative sum of the modeled second differences that have shrinkage priors, just like the $A$s and $B$s are.

Including diagonal parameters can make row and column factors ambiguous, so some constraints are needed if all row, column and diagonal factors are to be used. One approach is to adjust for row levels, like by dividing by premiums or exposures. Then a fair assumption is that there is no overall trend in the accident year direction, so all the trend is on the diagonals. Still you can have row factors, but in the estimation you make them the residuals to a trend through them, so then a trend line fitted to them would just be the x-axis. This is discussed in more detail in G. Venter and Şahin (2017). But with parameter reduction eliminating a fair number of parameters, this might not be necessary.

A good starting point for the exploratory analysis is to fit both the row-column and diagonal-column models with log regressions in second difference form. This can give an indication as to whether the row or diagonal factors are more explanatory. Usually before this, the triangle should be divided by an appropriate accident-year exposure measure, like premiums, policy counts, etc. In a row-column model, the row parameters can pick up such known row effects, but even in that model, adjusting for them first can help with parameter reduction. This was applied to the sample triangle using the exposures above (divided by 100,000 to keep the loss numbers in the same range).

This triangle with 9 rows actually has 11 diagonals, as two short rows usually found at the bottom of the triangle are not provided. The two initial regressions with all rows and columns or all columns and diagonals have very similar r-squares: 95.75% with rows and 95.76% for diagonals. But since there are more diagonals, the respective adjusted r-squares reverse, at

94.1% and 93.8%. But none of the row or diagonal t-statistics were above 1.8 in absolute value. This again suggests a Cape Cod model. Just small differences among row effects ends up as an aspect of the resulting MCMC estimation.

Again lasso is a good starting point for parameter reduction. The negatively correlated variables make it difficult to know which individually insignificant variables to leave out. Lasso selects groups of variables for each $\lambda$. Running it for each of the two regressions gives possible variable sets for use in MCMC. Since all the row and diagonal parameters are individually insignificant, the lambda.min variables were taken, as of the choices set out above, this $\lambda$ gives the largest set of variables, some of which can be eliminated later. All the columns except 6, 8, and 9 were included, as were rows 2, 3, and 6 and diagonals 5, 8, and 11.

The best row model was with rows 2 and 3 and columns 2, 3, 4, 5, 7 and 9. It gave looic of 103.3 with NLL of 92.6 and penalty of 10.7. These are not strictly comparable to the results without the exposure adjustment. The best diagonal model was not as good, with looic of 111.1, NLL of 101, and penalty of 10.1. This included only diagonal 5, although including 5 and 8 worked about as well. Thus the rows provide a better account of this triangle than do the diagonals. In fact, when the calendar-year trend is fairly constant, there is usually no need for diagonal parameters, as the row and column factors pick it up.

Since there are only a few row and diagonal parameters, they all can be included in a single model. Doing this then eliminating zero parameters left just row 2, columns 2, 3, 4, 5 and 7, and diagonals 5 and 8. The looic and NLL are 99.0 and 89.7, with a penalty of 9.3. This is easily the best fitting model by these measures. Similarly to the exposure adjustment, a lower penalty resulted even with as many nominal parameters.

*5c Calendar-Year Effects with Exposure Adjustment*

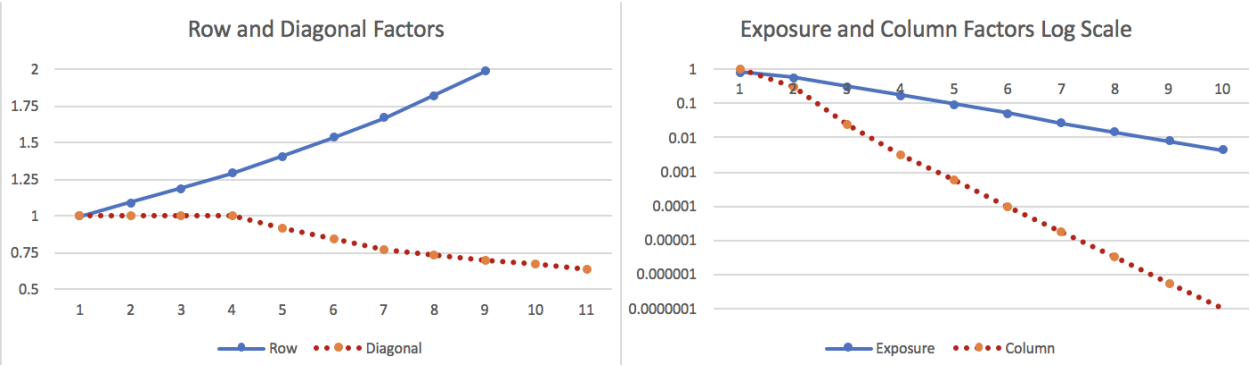



Figure 9: Factors

Finally, putting it all together, the exposure adjustment is included in the row-column-diagonal model. Since the whole triangle has already been divided by the exposures, just a constant is used instead of the actual exposures by row. This simplifies the coding. To keep factors in the same scale, the constant used was 10.

Table 10: Estimated Parameters

| cn | a2 | b2 | b3 | b4 | b5 | c5 | c8 | d1 | d2 | d3 | $\beta$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4.928 | 0.086 | -1.158 | -1.444 | 0.638 | 0.229 | -0.085 | 0.036 | -0.162 | -0.213 | -0.235 | 3.706 |

```
alpha = (10*exp(x_expo*w) + exp(x1*v+cn))*beta;
for (j in 1:N) y[j] ~ gamma(alpha[j], beta);
```

In this model, column 7 is no longer significant. Table 10 shows the estimated parameters, and the resulting factors for rows, columns, diagonals and exposures are in Figure 9. The exposure factors by column are denoted by $d$. The resulting looic and NLL are 97.4 and 87.1, with a penalty of 10.3. The exposure parameters did increase the penalty a bit in this case. There are nominally 12 parameters in this model, but since they have been shrunk, there are fewer degrees of freedom used – probably about 7. This is thus a fairly parsimonious model to produce the 40 row, column, diagonal and exposure factors plus the constant and $\beta$.

*5d Parameter Distributions*

It is easy in Stan to extract the sample distributions of the parameters. Here is some code, used here to make a correlation matrix of the parameters:

```
fit3p_ss = extract(fit3p, permuted = FALSE)  #Need FALSE to get array
fit3p_ss = fit3p_ss[,,1:14] #Only need first 14
dim(fit3p_ss) = c(4000,14) #Collapses dimensions
corrM = cor(fit3p_ss)   #Correlation matrix
write.csv(corrM, file = "cormatAPCexp.csv")
```

The extract function gives every variable or transformed variable plus other things. Here it is a (1000, 4, 139) array, so goes by sample then by chain. The parameters are in the first 14 elements, so only those are needed here. R keeps an array in a long vector with notation on how it is arranged. The dim function can collapse adjacent dimensions, giving just a table. Then the correlation matrix is computed by the cor function. It is shown in Table 11.

The diagonal parameters c5 and c8 have a lot of correlations with row and column parameters, as does the constant. The exposure parameters d1, d2, and d3 are negatively correlated with each other, as they are adjacent slope changes and somewhat offset each other. The first row and column parameters a2 and b2 have a degree of correlation as well, and are both negatively correlated with the constant, which offsets them to some degree – especially a2 as it is the only row parameter.

## 6 Conclusions

Reducing over-parameterization is known to improve the predictive accuracy of models, and now parameter shrinkage towards the mean provides further improvement, as it does with

Table 11: Parameter Correlation Matrix

| | cn | a2 | b2 | b3 | b4 | b5 | c5 | c8 | d1 | d2 | d3 | beta |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cn | 100% | -83% | -41% | 13% | -1% | 0% | 61% | -11% | -19% | 15% | 6% | -8% |
| a2 | -83% | 100% | 38% | -1% | 2% | 0% | -91% | 34% | 2% | -1% | -1% | 7% |
| b2 | -41% | 38% | 100% | -18% | 5% | -1% | -35% | 3% | -17% | 3% | 25% | 4% |
| b3 | 13% | -1% | -18% | 100% | 4% | 2% | 0% | -3% | -40% | 19% | 32% | -18% |
| b4 | -1% | 2% | 5% | 4% | 100% | -1% | -3% | 4% | -9% | 4% | 5% | 4% |
| b5 | 0% | 0% | -1% | 2% | -1% | 100% | -2% | 4% | 0% | -1% | 2% | 3% |
| c5 | 61% | -91% | -35% | 0% | -3% | -2% | 100% | -62% | 0% | 0% | 1% | -7% |
| c8 | -11% | 34% | 3% | -3% | 4% | 4% | -62% | 100% | 1% | 2% | -6% | 6% |
| d1 | -19% | 2% | -17% | -40% | -9% | 0% | 0% | 1% | 100% | -85% | -19% | 11% |
| d2 | 15% | -1% | 3% | 19% | 4% | -1% | 0% | 2% | -85% | 100% | -34% | -11% |
| d3 | 6% | -1% | 25% | 32% | 5% | 2% | 1% | -6% | -19% | -34% | 100% | -7% |
| beta | -8% | 7% | 4% | -18% | 4% | 3% | -7% | 6% | 11% | -11% | -7% | 100% |

credibility. In loss reserving, eliminating factors is not usually possible, but making the factors the cumulative sum of slope changes allows for parameter reduction. Building a design matrix of slope change variables is the starting point for this, and then lasso and Bayesian parameter shrinkage can be applied to do the estimation. There are R packages for these that require minimal programming.

In the end, lasso is more of a step towards MCMC estimation, as MCMC provides better tools for determining the best degree of shrinkage and for measuring predictive accuracy, as well as directly handling parameter uncertainty distributions. It also can handle most probability distributions. The negative correlation of the slope change variables makes lasso a very good starting point.

Extensions of the row-column factor model can improve performance. Here using diagonal trends and including an additive exposure-based component both proved helpful.

The gamma distribution with the scale parameter held constant across cells makes the variance proportional to the mean, which is a good starting point for reserve modeling. The variance-mean relationship can be further controlled by adding a parameter for that, as in the normal-k and GIG distributions. Modeling skewness can help with range predictions. Special cases of the Tweedie distribution are useful for that, and they also allow for modeling of the severity distribution with only counts and amounts in total, not individual claims. The Weibull-k distribution provides a different skewness effect, which can sometimes be appropriate. Mixing Poissons by the Tweedie gives two versions of popular frequency distributions, which can be fit with the same data across a triangle.

## References


Barnett, Glen, and Ben Zehnwirth. 2000. "Best Estimates for Reserves." *PCAS* 87: 245–303.

Blei, David M. 2015. "Regularized Regression." *Technometrics* http://www.cs.columbia.edu/∼blei/fogm/2015F/notes/regularized-regression.pdf.

Dean, C., J.F. Lawless, and G.E. Willmot. 1989. "A Mixed Poisson-Inverse-Gaussian Regression Model." *The Canadian Journal of Statistics* 17:2: 171–81.

Frost, Wade Hampton. 1939. "The Age Selection of Mortality from Tuberculosis in Successive Decades." *American Journal of Hygiene* 30:3A: 91–96.

Gao, Guangyuan, and S. Meng. 2017. "Stochastic Claims Reserving via a Bayesian Spline Model with Random Loss Ratio Effects." *ASTIN Bulletin*.

Gelfand, A. E. 1996. "Model Determination Using Sampling-Based Methods." *Markov Chain Monte Carlo in Practice, Ed. W. R. Gilks, S. Richardson, D. J. Spiegelhalter* London: Chapman and Hall: 145–62.

Gluck, Spencer M. 1997. "Balancing Development and Trend in Loss Reserve Analysis." *PCAS* 84: 482–532.

Greenberg, B. G., John J. Wright, and Cecil G. Sheps. 1950. "A Technique for Analyzing Some Factors Affecting the Incidence of Syphilis." *Journal of the American Statistical Association* 45:251: 373–99.

Hastie, T., R. Tibshirani, and M. Wainwright. 2015. "Statistical Learning with Sparsity." *CRC Press*.

Hoerl, A.E., and R. Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12: 55–67.

Hougaard, Philip, Ting Mei-Ling, and G.A. Whitmore. 1997. "Analysis of Overdispersed Count Data by Mixtures of Poisson Variables and Poisson Processes." *Biometrics* 53: 1225–38.

Jørgensen, Bent. 1987. "Exponential Dispersion Models." *Journal of the Royal Statistical Society. Series B (Methodological)* 49:2: 127–62.

———. 1997. "The Theory of Dispersion Models." *Chapman & Hall*.

Klugman, Stuart A., Harry H. Panjer, and Gorden E. Willmot. 2008. "Loss Models: From Data to Decisions." *Wiley* 3rd Edition: 669.

Meyers, Glenn. 2009. "Predictive Modeling with the Tweedie Distribution." *Casualty Annual Meeting Handouts* C-25: https://www.casact.org/education/annual/2009/handouts/c25–meyers.pdf.

Müller, Thomas. 2016. "Projection for Claims Triangles by Affine Age-to-Age Development." *Variance* 10:1: 121–44.

Renshaw, A. E., and S. Haberman. 2006. "A Cohort-Based Extension to the Lee-Carter

Model for Mortality Reduction Factors." *Insurance: Mathematics and Economics* 38: 556–70.

Renshaw, Arthur E. 1994. "Modelling the Claims Process in the Presence of Covariates." *Astin Bulletin* 24:2: 265–85.

Rigby, R.A., D.M. Stasinopoulos, and C. Akantziliotou. 2008. "A Framework for Modelling Overdispersed Count Data, Including the Poisson-Shifted Generalized Inverse Gaussian." *Computational Statistics and Data Analysis* 53: 381–93.

Stein, Charles. 1956. "Inadmissibility of the Usual Estimator of the Mean of a Multivariate Normal Distribution." *Proceedings of the Third Berkeley Symposium* 1: 197–206.

Taylor, Greg. 1977. "Separation of Inflation and Other Effects from the Distribution of Non-Life Insurance Claims Delays." *Astin Bulletin* 9: 217–30.

Taylor, Greg C., and Frank R. Ashe. 1983. "Second Moments of Estimates of Outstanding Claims." *Journal of Econometrics* 23: 37–61.

Vehtari, Aki, Andrew Gelman, and Jonah Gabry. 2017. "Practical Bayesian Model Evaluation Using Leave-One-Out Cross-Validation and Waic." *Journal of Statistcs and Computing* 27:5: 1413–32.

Venter, Gary G. 2007. "Generalized Linear Models Beyond the Exponential Family with Loss Reserve Applications." *Casualty Actuarial Society E-Forum* Summer: 1–25.

———. 2011. "Mortality Trend Models." *Casualty Actuarial Society E-Forum* Winter (2).

Venter, Gary G., Roman Gutkovich, and Qian Gao. 2017. "Parameter Reduction in Actuarial Triangle Models." *Variance* https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2992300.

Venter, Gary, and Şule Şahin. 2017. "Parsimonious Parameterization of Age-Period-Cohort Models by Bayesian Shrinkage." *Astin Bulletin*.

Wüthrich, Mario V. 2003. "Claims Reserving Using Tweedie's Compound Poisson Model." *Astin Bulletin* 33:2: 331–46.

Ye, J. 1998. "On Measuring and Correcting the Effects of Data Mining and Model Selection." *Journal of the American Statistical Association* 93: 120–31.

Zha, Liteng, Dominique Lord, and Yajie Zou. 2016. "The Poisson Inverse Gaussian (Pig) Generalized Linear Regression Model for Analyzing Motor Vehicle Crash Data." *Journal of Transportation Safety & Security* 8: 18–35.