CAS RESEARCH PAPERS

HIERARCHICAL COMPARTMENTAL RESERVING MODELS

Markus Gesmann and Jake Morris

CASUALTY ACTUARIAL SOCIETY

Hierarchical Compartmental Reserving Models

Markus Gesmann and Jake Morris

Hierarchical compartmental reserving models provide a parametric framework for describing aggregate insurance claims processes using differential equations. We discuss how these models can be specified in a fully Bayesian modeling framework to jointly fit paid and outstanding claims development data, taking into account the random nature of claims and underlying latent process parameters. We demonstrate how modelers can utilize their expertise to describe specific development features and incorporate prior knowledge into parameter estimation. We also explore the subtle yet important difference between modeling incremental and cumulative claims payments. Finally, we discuss parameter variation across multiple dimensions and introduce an approach to incorporate market cycle data such as rate changes into the modeling process. Examples and case studies are shown using the probabilistic programming language Stan via the brms package in R.

Citation:

Gesmann, M., and Morris, J. "Hierarchical Compartmental Reserving Models." *Casualty Actuarial Society*, CAS Research Papers, 19 Aug. 2020, www.casact.org/research/research-papers/Compartmental-Reserving-Models-GesmannMorris0820.pdf.

Contents

1. Introduction 5

1.1 OUTLINE OF THE DOCUMENT 7

2. Modeling the average claims development process 9

2.1 INTRODUCTION TO COMPARTMENTAL MODELS 9

2.2 MULTICOMPARTMENTAL CLAIMS MODELING 9

2.3 TWO-STAGE OUTSTANDING COMPARTMENTAL MODEL 13

2.4 MULTISTAGE EXPOSURE MODEL 15

3. Modeling parameter and process uncertainties 21

3.1 DATA-GENERATING ("PROCESS") DISTRIBUTION 21

3.2 PRIOR PARAMETER DISTRIBUTIONS 22

3.3 CUMULATIVE VERSUS INCREMENTAL DATA 24

3.4 PRIOR PREDICTIVE EXAMPLES 29

3.4.1 Negative binomial process distribution 30

3.4.2 Gaussian process distribution with constant coefficient of variation 31

3.4.3 Compartmental model with lognormal distribution 32

3.4.3.1 Compartmental model setup 33

4. Modeling hierarchical structures and correlations 38

4.1 INTRODUCTION TO HIERARCHICAL MODELS 38

4.2 SPECIFYING A HIERARCHY 39

4.3 REGULARIZATION 40

4.4 MARKET CYCLES 40

4.5 SINGLE-TRIANGLE HIERARCHICAL COMPARTMENTAL MODEL 42

4.6 EXPECTED VERSUS ULTIMATE LOSS 48

4.7 CORRELATIONS ACROSS EFFECTS 50

5. Compartmental reserving case study 53

5.1 DATA PREPARATION 54

5.2 MODEL BUILDING 55

5.2.1 Model 1: Gaussian distribution *55*

5.2.2 Model 2: Lognormal distribution and additional structure *57*

5.2.3 Model 3: Pricing and reserving cycle submodel *58*

5.3 TRAINING 60

5.3.1 Training model 1 .... *60*

5.3.2 Training model 2 .... *62*

5.3.3 Training model 3 .... *68*

5.4 TESTING AND SELECTION 71

5.4.1 Testing model 2 .... *71*

5.4.2 Testing model 3 .... *72*

5.4.3 Model selection *73*

5.5 FITTING 74

5.5.1 Fitting the selected model *74*

5.6 RESERVING 75

5.7 DISCUSSION 76

6. Summary and future developments 78

6.1 EXTENSIONS 80

About the authors 81

Acknowledgments 81

7. Appendix 82

7.1 R CODE FROM SECTION 4 82

7.1.1 Multilevel effects with narrow priors *85*

7.1.2 Multilevel effects with wider priors *88*

7.2 R CODE FROM CASE STUDY IN SECTION 5 91

7.2.1 Data *91*

7.2.2 Model 1 .... *94*

7.2.3 Model 2 .... *98*

7.2.4 Model 3 .... *103*

7.3 SESSION INFORMATION 108

8. References 109

1. Introduction

Claims reserving, pricing, and capital modeling are core to actuarial functions. The assumptions used in the underlying actuarial models play a key role in the management of any insurance company.

Knowing when those underlying assumptions are no longer valid is critical for the business to initiate change. Transparent models that clearly state the underlying assumptions are easier to test and challenge, and hence can speed up the process for change.

Unfortunately, many underlying risk factors in insurance are not directly measurable and are latent in nature. Although prices are set for all policies, only a fraction of policies will incur losses. Reserving is often based on relatively sparse data to make predictions about future payments, potentially over long time horizons.

Combining judgment about future developments with historical data is therefore common practice for many reserving teams, particularly when entering a new product, line of business, or geography, or when changes to products and business processes would make past data a less credible predictor. Modern Bayesian modeling provides a rich tool kit for bringing together the expertise and business insight of the actuary and augmenting and updating it with data.

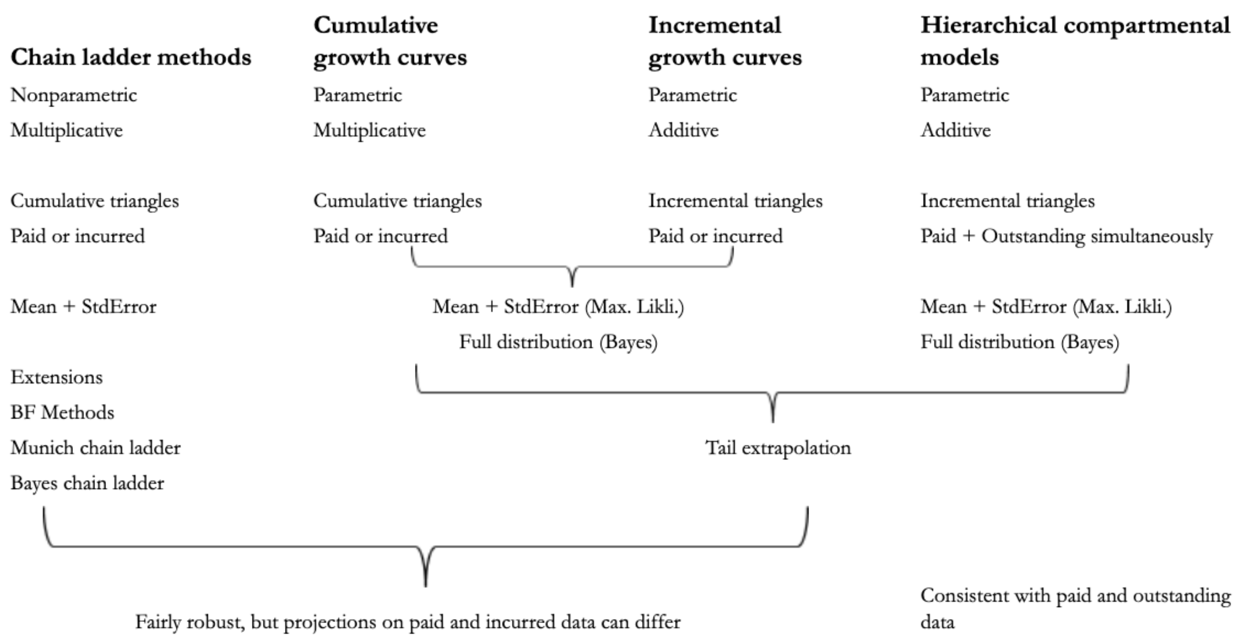
In situations where the actuary has access to large volumes of data, nonparametric machine learning techniques might provide a better approach. Some of these are based on enhancement of traditional approaches such as the chain-ladder method (Wüthrich 2018, Carrato and Visintin 2019), with others using neural networks (Kuo 2018); (Gabrielli, Richman and Wüthrich 2018) and Gaussian processes (Lally and Hartman 2018).

With small and sparse data, parametric models such as growth curves (Sherman 1984, Clark 2003, Guszcza 2008) can help the actuary capture key claims development features without overfitting, however, the actuary may require expertise and judgement in the selection of the growth curve and its parameterization.

Hierarchical Compartmental Reserving Models

Hierarchical compartmental reserving models provide an alternative parametric framework for describing the high-level business processes driving claims development in insurance (Morris 2016). Rather than selecting a growth curve, the experienced modeler can build loss emergence patterns from first principles using differential equations. Additionally, these loss emergence patterns can be constructed in a way that allows outstanding and paid data to be described simultaneously (see Figure 1.1).

Figure 1.1. Comparison of reserving methods and models



The starting point mirrors that of a scientist trying to describe a particular process in the real world using a mathematical model. By its very nature, the model will only be able to approximate the real world. We derive a “small-world” view that makes simplified assumptions about the real world, but which may allow us to improve our understanding of key processes. In turn, we can attempt to address our real-world questions by testing various ideas about how the real world functions.

Compared with many machine-learning methods, which are sometimes described as “black boxes”, hierarchical compartmental reserving models can be viewed as “transparent boxes.” All modeling assumptions must be articulated by the practitioner, with the benefit that expert knowledge can be incorporated, and each modeling assumption can be challenged more easily by other experts.

Finally, working within a parametric framework allows us to simulate artificial data in advance of fitting any models. An a priori understanding of model suitability should steer practitioners to align modeling assumptions with their expectations of reality, and therefore may improve predictive performance.

1.1 Outline of the document

This document builds on the original paper by Morris (2016). It provides a practical introduction to hierarchical compartmental reserving in a Bayesian framework and is outlined as follows:

- In Section 2 we develop the original ordinary differential equation (ODE) model and demonstrate how the model can be modified to allow for different claims processes, including different settlement speeds for standard versus disputed claims and different exposure to reporting processes.
- In Section 3 we build the stochastic part of the model and provide guidance on how to parameterize prior parameter distributions to optimize model convergence. Furthermore, we discuss why one should model incremental paid data in the context of underlying statistical assumptions and previously published methodologies.
- In Section 4 we add hierarchical structure to the model, which links compartmental models back to credibility theory and regularization. The “GenIns” data set is used to illustrate these concepts as we fit the model to actual claims data, and we highlight the conceptual differences between expected and ultimate loss ratios when interpreting model outputs.
- Section 5 concludes with a case study demonstrating how such models can be implemented in “RStan” using the “brms” package. Models of varying complexity are tested against each other, with add-ons such as parameter variation by both origin and

development period, and market cycle submodels. Model selection and validation is demonstrated using posterior predictive checks and holdout sample methods.

- Section 6 summarizes the document and provides an outlook for future research.
- The appendix presents the R code to replicate the models in Sections 4 and 5.

We assume the reader is somewhat familiar with Bayesian modeling concepts. Good introductory textbooks on Bayesian data analysis are those by McElreath (2015), Kruschke (2014), and Gelman et al. (2014). For hierarchical models we recommend Gelman and Hill (2007), and for best practices on a Bayesian work flow, see Betancourt (2018).

In this document we will demonstrate practical examples using the brms (Bürkner 2017) interface to the probabilistic programming language Stan (Stan Development Team 2019) from R (R Core Team 2019).

The brm function—short for "Bayesian regression model"—in brms allows us to write our models in a way similar to a generalized linear model or multilevel model with the popular R functions glm or lme4::lmer (Bates et al. 2015). The Stan code is generated and executed by brm. Experienced users can access all underlying Stan code from brms as required.

Stan is a C++ library for Bayesian inference using the No-U-Turn Sampler, also known as NUTS (a variant of Hamiltonian Monte Carlo, or HMC), or frequentist inference via limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) optimization (Carpenter et al. 2017). For an introduction to HMC see Betancourt (2017).

The Stan language is similar to Bayesian Inference Using Gibbs Sampling, or BUGS (Lunn et al. 2000), and Just Another Gibbs Sampler, or JAGS (Plummer 2003), which use Gibbs sampling instead of HMC. BUGS was used by Morris (2016) and has been used for Bayesian reserving models by others (Scollnik 2001; Verrall 2004; Zhang, Dukic, and Guszcza 2012), while Schmid (2010) and Meyers (2015) have used JAGS. Examples of reserving models built in Stan can be found in Cooney (2017) and Gao (2018).

2. Modeling the average claims development process

Many different approaches have been put forward to model the average claims development process. The most well-known is perhaps the nonparametric chain-ladder method, which uses average loss development factors to model loss emergence (Schmidt 2006). Parametric approaches, such as growth curve models, have also been widely documented (Sherman 1984; Clark 2003; Guszcza 2008).

2.1 Introduction to compartmental models

Compartmental models are a popular tool in many disciplines to describe the behavior and dynamics of interacting processes using differential equations.

Disciplines in which compartmental models are used include the following:

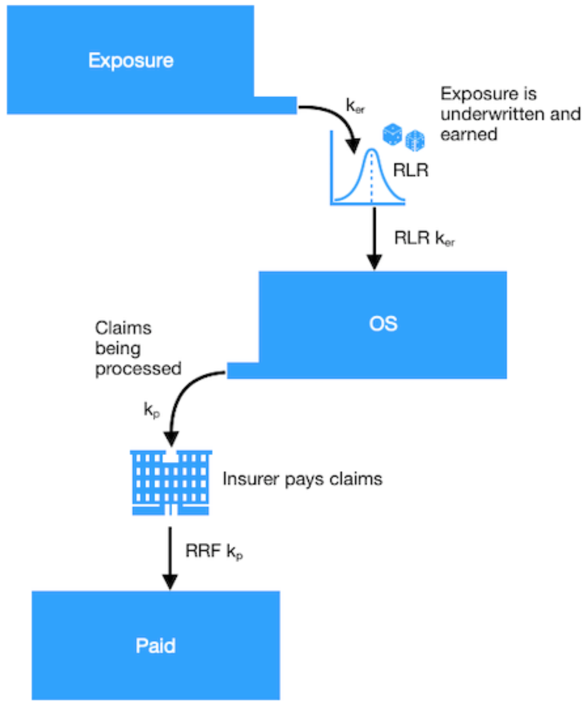
- Pharmaceutical sciences, to model how drugs interact with the body
- Electrical engineering, to describe the flow of electricity
- Biophysics, to explain the interactions of neurons
- Epidemiology, to understand the spread of diseases
- Biology, to describe the interaction of different populations

Each compartment typically relates to a different stage or population of the modeled process, usually described with its own differential equation.

2.2 Multicompartmental claims modeling

Similar to salt-mixing problem models, which describe the flow of fluids from one tank into another (Winkel 1994), we can model the flow of information or monetary amounts between exposure, claims outstanding, and claims payment states for a group of policies.

Figure 2.1. Schematic diagram of claims flow



The diagram in Figure 2.1 gives a schematic view of three compartments ("tanks") and the flow of monetary amounts between them. We start with a "bucket" of exposure or premium, which outflows into a second bucket, labeled OS, for reported outstanding claims.

The parameter k_{er} describes how quickly the exposure expires as claims are reported. For a group of risks, it is unlikely that 100% of exposure will convert to claims. Therefore, a proportion, or multiple of exposure (RLR = reported loss ratio), is assumed to convert to outstanding claim amounts.

Once claims have been processed, the insurer proceeds to pay its policyholders. The parameter k_p describes the speed of claims settlement, and the parameter RRF (reserve robustness factor) denotes the proportion of outstanding claims that are paid. An RRF greater than 1 would indicate case underreserving, whereas an RRF less than 1 would indicate case overreserving.

The set of compartments (the "state-space") and the claims processed through them can be expressed with a set of ordinary differential equations (ODEs). Denoting the "state-variables" EX = exposure, OS = outstanding claims, and PD = paid claims (i.e., the individual compartments), we have the following:

$$\begin{aligned} \mathrm{dEX/dt} &= -\mathrm{k_{er}} \cdot \mathrm{EX} \\ \mathrm{dOS/dt} &= \mathrm{k_{er}} \cdot \mathrm{RLR} \cdot \mathrm{EX} - \mathrm{k_p} \cdot \mathrm{OS} \\ \mathrm{dPD/dt} &= \mathrm{k_p} \cdot \mathrm{RRF} \cdot \mathrm{OS} \end{aligned} \tag{1}$$

The initial conditions at time 0 are typically set as $\mathrm{EX}(0) = \Pi$ (ultimate earned premiums), $\mathrm{OS}(0) = 0,\ \mathrm{PD}(0) = 0$ for accident period cohorts. Alternative approaches can be taken for policy year cohorts, which are discussed later in this section.

For exposure defined in terms of ultimate earned premium amounts, the parameters describe the following:

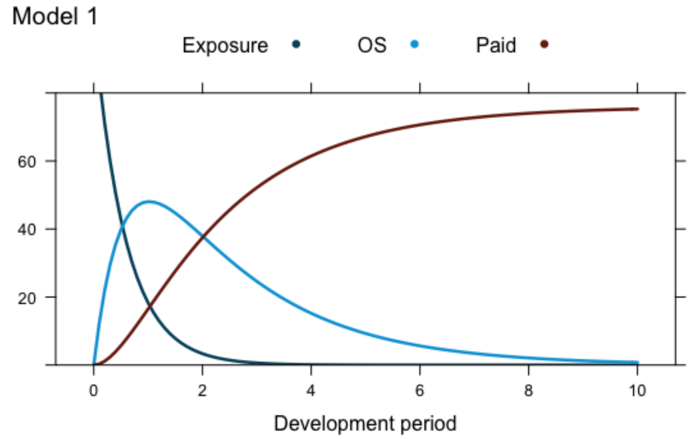
- **Rate of earning and reporting** ($\mathrm{k_{er}}$): the rate at which claim events occur and are subsequently reported to the insurer
- **Reported loss ratio** (RLR): the proportion of exposure that becomes reported claims
- **Reserve robustness factor** (RRF): the proportion of outstanding claims that are eventually paid
- **Rate of payment** ($\mathrm{k_p}$): the rate at which outstanding claims are paid

Here we assume that parameters are time independent, but in later sections we will allow for increased structural flexibility.

The expected loss ratio, ELR (expected ultimate losses ÷ ultimate premiums), can be derived as the product of RLR and RRF (the reported loss ratio scaled by the reserve robustness factor).

Setting parameters $\mathrm{k_{er}} = 1.7$, $\mathrm{RLR} = 0.8$, $\mathrm{k_p} = 0.5$, and $\mathrm{RRF} = 0.95$ produces the output shown in Figure 2.2.

Figure 2.2. Illustration of the different compartment amounts for a group of policies over time



The autonomous system of ODEs above can be solved analytically by iterative integration:

$$\begin{aligned}
\mathrm{EX}(t) &= \Pi \cdot \exp(-k_{er}t) \\
\mathrm{OS}(t) &= \frac{\Pi \cdot \mathrm{RLR} \cdot k_{er}}{k_{er} - k_p} \cdot \left(\exp(-k_p t) - \exp(-k_{er}t)\right) \\
\mathrm{PD}(t) &= \frac{\Pi \cdot \mathrm{RLR} \cdot \mathrm{RRF}}{k_{er} - k_p}\left(k_{er} \cdot (1 - \exp(-k_p t) - k_p \cdot (1 - \exp(-k_{er}t)\right)
\end{aligned} \tag{2}$$

The first equation describes an exponential decay of exposure over time.

Outstanding claims are modeled as the difference between two exponential decay curves with different time scales, which determine how the reported losses ($\Pi \cdot \mathrm{RLR}$) are spread out over time and how outstanding losses decay as payments are made.

The paid curve is an integration of the outstanding losses curve. It represents a classic loss emergence pattern with two parameters, k_{er} and k_p, multiplied by an expected ultimate claims cost, represented by the product of $\Pi \cdot \mathrm{RLR} \cdot \mathrm{RRF}$.

The peak of the outstanding claims cost is at $t = \log(k_p/k_{er})/(k_p - k_{er})$, representing the inflection point in paid loss emergence. Note that the summation of $\mathrm{OS}(t)$ and $\mathrm{PD}(t)$ gives us the implied incurred losses at time t.

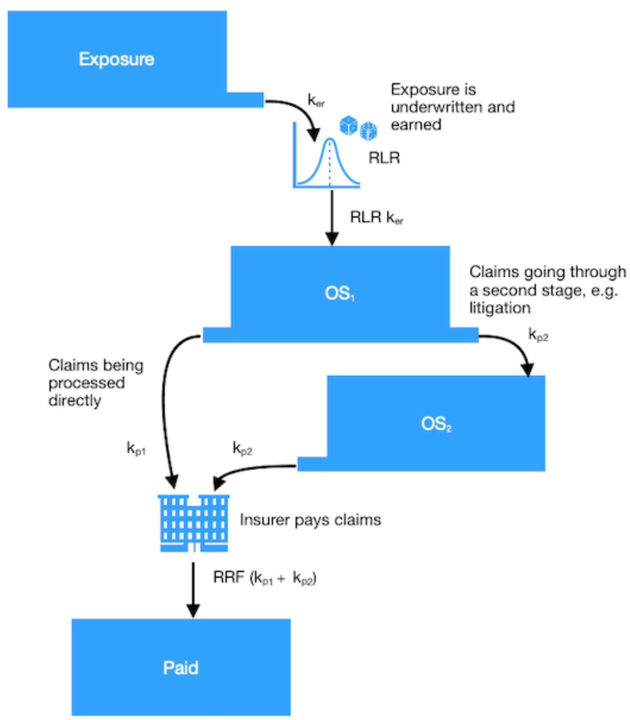
2.3 Two-stage outstanding compartmental model

We can increase the flexibility of the model in many ways, for example by introducing time-dependent parameters or adding one or more compartments to the model, as outlined by Morris (2016).

Adding compartments keeps our ODEs autonomous, which makes them easier to solve analytically and to visualize in a single diagram.

The diagram in Figure 2.3 depicts a compartmental model that assumes reported claims fall into two categories: they are either dealt with by the insurance company quickly, with claims paid to policyholders in a timely fashion, or they go through another, more time-consuming process (such as investigation, dispute, and/or litigation).

Figure 2.3. Two compartments for outstanding claims to allow some claims to be settled faster than others



We translate this diagram into a new set of ODEs:

$$
\begin{aligned}
\mathrm{dEX/dt} &= -\mathrm{k_{er}\,EX} \\
\mathrm{dOS_1/dt} &= \mathrm{k_{er}\,RLR \cdot EX} - \left(\mathrm{k_{p_1}} + \mathrm{k_{p_2}}\right) \mathrm{OS_1} \\
\mathrm{dOS_2/dt} &= \mathrm{k_{p_2}\,(OS_1 - OS_2)} \\
\mathrm{dPD/dt} &= \mathrm{RRF\,(k_{p_1}\,OS_1 + k_{p_2}\,OS_2)}
\end{aligned}
\tag{3}
$$

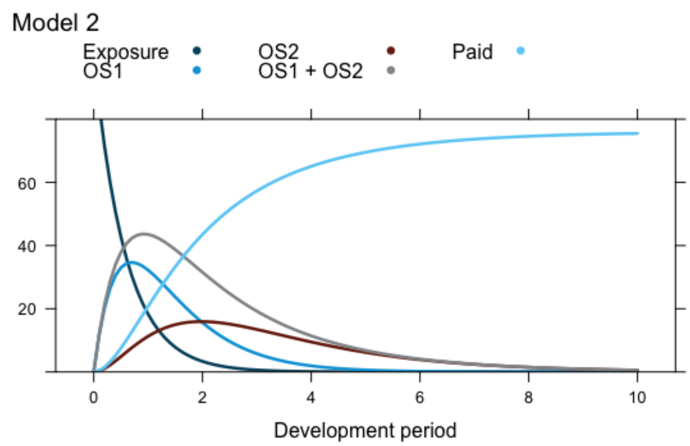
Solving the system of autonomous ODEs can be done iteratively, resulting in the solutions below. However, numerical solvers are typically preferred to reduce algebraic computation and minimize risk of error.

$$
\begin{aligned}
\mathrm{EX(t)} = {} & \Pi \exp(-\mathrm{k_{er}t}) \\
\mathrm{OS_1(t)} = {} & \frac{\Pi\ \mathrm{RLR\ k_{er}}}{\mathrm{k_{er} - k_{p_1} - k_{p_2}}}\left[\exp(-(k_{p_1} + k_{p_2})t) - \exp(-k_{er}t)\right] \\
\mathrm{OS_2(t)} = {} & \frac{\Pi\ \mathrm{RLR\ k_{er}\ k_{p_2}}}{\mathrm{k_{p_1}(k_{p_2} - k_{er})(k_{er} - k_{p_1} - k_{p_2})}}\Big[\\
& \exp(-(\mathrm{k_{p_1}} + \mathrm{k_{p_2}})\mathrm{t})(\mathrm{k_{er}} - \mathrm{k_{p_2}}) - \\
& \exp(-\mathrm{k_{p_2}t})(\mathrm{k_{er}} - \mathrm{k_{p_1}} - \mathrm{k_{p_2}}) - \\
& \exp(-\mathrm{k_{er}t})\,\mathrm{k_{p_1}}\Big] \\
\mathrm{PD(t)} = {} & \frac{\Pi\ \mathrm{RLR\ RRF}}{\mathrm{k_{p_1}(k_{p_2} - k_{er})(k_{er} - k_{p_1} - k_{p_2})}}\Big[\\
& \left(\mathrm{k_{p_1}}(\mathrm{k_{er}}\,(\mathrm{k_{p_1}} - \mathrm{k_{er}}) - \mathrm{k_{p_2}}\,(\mathrm{k_{p_1}} + \mathrm{k_{p_2}})) + 2\mathrm{k_{er}k_{p_2}}\right) + \\
& \exp(-(\mathrm{k_{p_1}} + \mathrm{k_{p_2}})\mathrm{t})\left(\mathrm{k_{er}}(\mathrm{k_{er}k_{p_1}} - \mathrm{k_{er}k_{p_2}} + \mathrm{k_{p_2}^2} - \mathrm{k_{p_1}k_{p_2}})\right) + \\
& \exp(-\mathrm{k_{p_2}t})\left(\mathrm{k_{er}k_{p_2}}(\mathrm{k_{er}} - \mathrm{k_{p_1}} - \mathrm{k_{p_2}}\right) + \\
& \exp(-\mathrm{k_{er}t})\left(\mathrm{k_{p_1}}(\mathrm{k_{p_1}k_{p_2}} + \mathrm{k_{p_2}^2} - \mathrm{k_{er}k_{p_1}})\right)\Big]
\end{aligned}
$$

Plotting the solutions illustrates faster and slower processes for the two distinct groups of outstanding claims, producing a paid curve that exhibits a steep start followed by a longer tail, shown in Figure 2.4.

Many data sets will not separate outstanding claims into different categories, in which case, the sum of $\mathrm{OS_1}$ and $\mathrm{OS_2}$ will be used for fitting purposes.

Figure 2.4. Example of a two-stage outstanding model, with a portion of the claims settled at a faster rate (0.7) than others (0.5)



It is trivial to expand this approach by adding further compartments to allow for more than two distinct settlement processes. The next section introduces a multistage exposure compartment model in which the number of compartments becomes a variable itself.

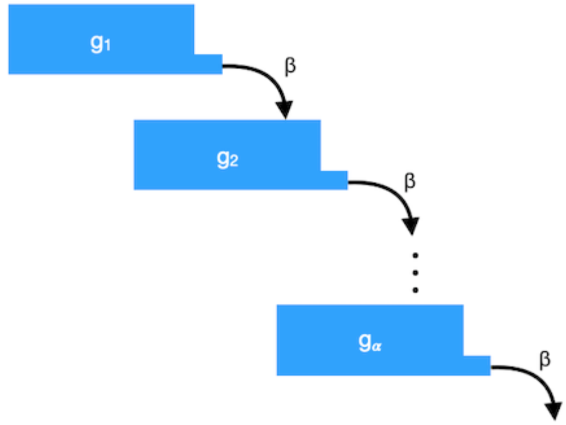
2.4 Multistage exposure model

The models thus far have assumed an exponential decay of exposure over time. Furthermore, we have assumed that the exposure at $t = 0$ can be represented by ultimate earned premiums.

In reality, at $t = 0$ we may expect some exposures to still be earning out (on an accident year basis) or not yet be written (on a policy year basis). If we have a view on how exposures have earned in the past and may earn into the future (e.g., from our business plan), then we can feed blocks of exposure into the compartmental system over development time (Morris 2016), with the result that k_{er} simplifies to k_r.

Alternatively, we can use a cascading compartmental model to allow for different earning and reporting processes as part of the modeling process, as in Figure 2.5.

Figure 2.5 Schematic of a multistage transition model



We assume that risks are written and earned at a constant rate, analogous to water flowing from one tank to the next at a constant rate. The claims reporting delay is then modeled by the number of different compartments.

We can express this compartmental model as a system of α ODEs:

$$\begin{aligned}
\dot{g}_1 &= -\beta\, g_1 \\
\dot{g}_2 &= \beta\, g_1 - \beta\, g_2 \\
&\vdots \\
\dot{g}_\alpha &= \beta g_{\alpha-1} - \beta g_\alpha
\end{aligned}$$

More succinctly, we express the system as an ODE of α order:

$$\frac{d^\alpha}{dt^\alpha} g(t;\alpha,\beta) = -\sum_{i=1}^{\alpha} \binom{\alpha}{i} \beta^i \frac{d^{(\alpha-i)}}{dt^{(\alpha-i)}} g(t;\alpha,\beta)$$

For $\alpha = 1$ we get back to an exponential decay model.

This ODE can be solved analytically (Gesmann 2002):

$$g(t;\alpha,\beta) = \begin{cases} \dfrac{\beta^\alpha\, t^{\alpha-1}}{(\alpha-1)!} e^{-\beta t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

Relaxing the assumption that α is a positive integer gives

$$g(t;\alpha,\beta) = \frac{\beta^\alpha\, t^{\alpha-1}}{\Gamma(\alpha)} e^{-\beta t} \quad \text{for } t \geq 0,$$

with the gamma function $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}\, e^{-x} dx$.

The trained eye may recognize that $g(t; \alpha, \beta)$ is the probability density function (PDF) of the gamma distribution, which is commonly used to model waiting times.

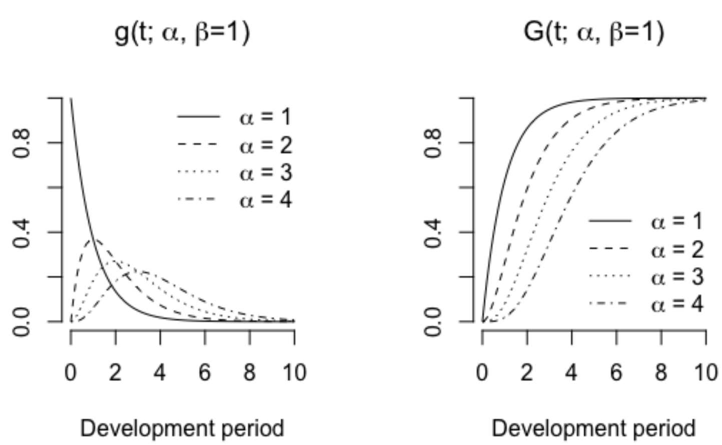
Finally, let's imagine we collect the outflowing water in another tank, with amounts in this compartment calculated by integrating $g(t; \alpha, \beta)$ over time. This integration results in a gamma cumulative distribution function (CDF),

$$G(t; \alpha, \beta) = \int_0^t g\,(x; \alpha, \beta) dx = \frac{\gamma(\alpha, t\,\beta)}{\Gamma(\alpha)} \quad \text{for } t \geq 0,$$

using the incomplete gamma function $\gamma(s, t) = \int_0^x t^{(s-1)}\, e^{-t} dt$.

Visualizing the functions $g(t; \alpha, \beta)$ and $G(t; \alpha, \beta)$ shows that for a fixed β the parameter α can be used to determine how quickly the curves converge; see Figure 2.6.

Figure 2.6. Visualization of the gamma function for different values of α



The gamma function has previously been proposed to model loss emergence patterns, labeled as the "Hoerl curve" (England and Verrall 2001).

For our purpose of modeling exposure decay, we introduce parameters k_e, describing the earning speed, and d_r, describing the reporting delay.

We define $k_e = \beta$ and $d_r = \alpha$; which implies that the speed of earning k_e matches the flow of water from one tank into the next, while d_r can be linked to the number of transient tanks.

The analytical form for the exposure can be then expressed as

$$EX(t) = \Pi \frac{k_e^{d_r} t^{d_r - 1}}{\Gamma(d_r)} e^{-k_e t}.$$

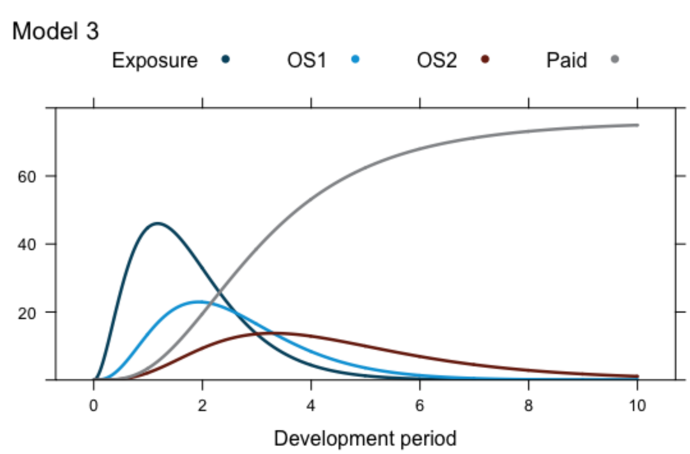
In other words, we model exposure as ultimate earned premium (Π) weighted over time with a gamma PDF.

Inserting the solution into the ODEs produces the following:

$$\begin{aligned} dOS_1/dt &= \Pi \cdot RLR \cdot \frac{k_e^{d_r} t^{d_r-1}}{\Gamma(d_r)} e^{-k_e t} - (k_{p_1} + k_{p_2}) \cdot OS_1 \\ dOS_2/dt &= k_{p_2} \cdot (OS_1 - OS_2) \\ dPD/dt &= RRF \cdot (k_{p_1} \cdot OS_1 + k_{p_2} \cdot OS_2) \end{aligned} \tag{4}$$

Figure 2.7 illustrates the impact the multistage exposure submodel has on the two-stage outstanding curves and paid loss emergence pattern. Claim reports and payments develop more slowly, as typically observed for longer-tailed business lines.

Figure 2.7. Example of multistage exposure model ($k_e = 3$, $d_r = 1.7$) with a two-stage outstanding process

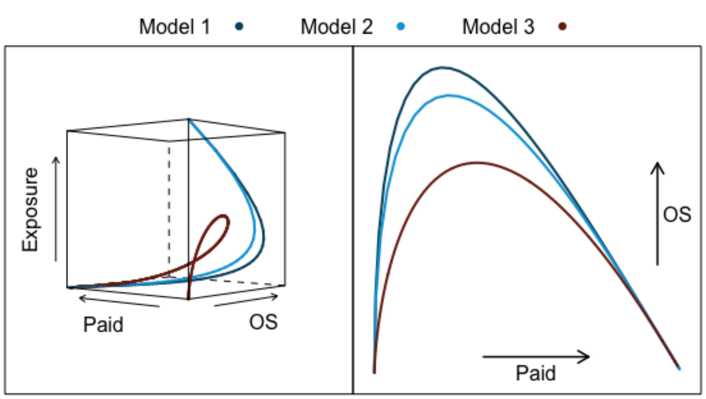


Note that with the proposed extensions, setting $d_r = 1$ and $k_{p_2} = 0$ gets us back to the original model (1), with one parameter, k_{er}, for the exposure and reporting process, and one parameter, k_p, for the payment process.

We can express our model as an autonomous ODE system for various extensions, but integrating the system is not always straightforward. Fortunately, as we will see in a later section, Stan (Stan Development Team 2019) has a Runge-Kutta solver to integrate the system numerically.

It is worth emphasizing that this framework allows us to build parametric curves that share parameters across paid and outstanding data. This enables us to learn from both data sources at the same time and have consistent ultimate projections of paid and incurred claims (see the phase plots in Figure 2.8). This is in contrast to fitting separate curves for paid and incurred data, resulting in two different answers.

Figure 2.8. Phase plot of models 1 – 3. The 3-D plot (left) illustrates that models 1 and 2 assume exposure to peak at t = 0, while model 3 assumes exposure to be 0 at t = 0, with gradual increase and decrease over time. Note that for models 2 and 3 OS displays the sum of OS1 + OS2. The 2-D plot (right) shows the relationship between outstanding and paid claims, which can be compared against actual data.



In practice, for some business lines, claim characteristics can be heterogeneous or case handling processes inconsistent, resulting in volatile outstanding claims development. The value of incorporating outstandings may be diminished if the data do not broadly conform to the model's assumption on how outstandings and payments link.

Similarly, for the model assumptions to hold, the process lifecycle from earning exposure through to paying claims should be approximable as continuous for a volume of policies.

In summary, compartmental models provide a flexible framework to build loss emergence patterns for outstanding and paid claims from first principles. We outline two extensions here, yet many more are feasible depending on the underlying features the practitioner is hoping to build within a structural model for the average development process. Getting a "feel" for the parameters, their interpretations, and how they determine loss emergence characteristics in each case will become important when we have to set prior distributions for them.

3. Modeling parameter and process uncertainties

In the previous section we developed an understanding of how to model the average behavior of the claims development process using compartmental models. In this section, we start to build statistical models around a central statistic such as the mean or median.

We will not model any data here; instead, the focus is on selecting distributions for the observation scale (the "process") and priors for the system parameters. The aim is to create a model that can simulate data that shares key characteristics of real data. This will lead to a discussion on modeling cumulative versus incremental paid claims.

We demonstrate how these models can be implemented in Stan (Stan Development Team 2019) using brms (Bürkner 2017) as an interface from R (R Core Team 2019) to generate prior predictive output.

3.1 Data-generating ("process") distribution

To model the data-generating process for our observations, y_j, we have to consider the likely distribution of our data (D) and how the process can be expressed in terms of parameters. In simple models, we often distinguish between parameters that are direct functions of variables in the data (Θ) and family-specific (Φ), which are either fixed or vary indirectly with respect to Θ in line with specific distributional assumptions. Examples of the latter include the standard deviation σ in Gaussian models or the shape α in gamma models.

The generic form of a univariate data-generating process for repeated measures data (such as claims development) can be written as follows:

$$y_j \sim D(f(t_j, \Theta), \Phi)$$

Note that in more complex models we can estimate specific relationships between Φ and data features by introducing additional parameters.

It can be helpful to think about how the variability in y_j is related to changes in the mean or median. In ordinary linear regression, where the process is assumed to follow a normal distribution (or, equivalently, to have a Gaussian error term), a constant variance is typically assumed:

$$y_j \sim \mathrm{Normal}(\mu(t_j), \sigma)$$

In the claims reserving setting it is often assumed that volatility changes with the mean. A multiplicative or overdispersed relationship is usually considered.

Given that claims are typically right skewed and that larger claims tend to exhibit larger variation, the lognormal or gamma distributions are often a good starting point. Other popular choices are the negative binomial (claim counts) and Tweedie (pure premium) distributions. However, for some problems, standard distributions will not appropriately characterize the level of zero-inflation or extreme losses observed without additional model structure.

For the lognormal distribution, a constant change assumption on the log scale translates to a constant coefficient of variation (CoV) on the original scale ($\mathrm{CoV} = \sqrt{\exp(\sigma^2) - 1}$):

$$y_j \sim \mathrm{Lognormal}(\mu(t_j), \sigma)$$

It can be helpful to model variables on a similar scale so that they have similar orders of magnitude. This can improve sampling efficiency and, in the case of the target variable, makes it easier apply the same model to different data sets. For this reason, we advise modeling loss ratios instead of loss amounts in the first instance. However, we also note that this approach will have an effect on the implicit volume weighting within the optimization routine for constant CoV distributions, and on occasion it may be preferable to target claim amounts.

The choice of the process distribution should be carefully considered, and the modeler should be able to articulate the selection criteria.

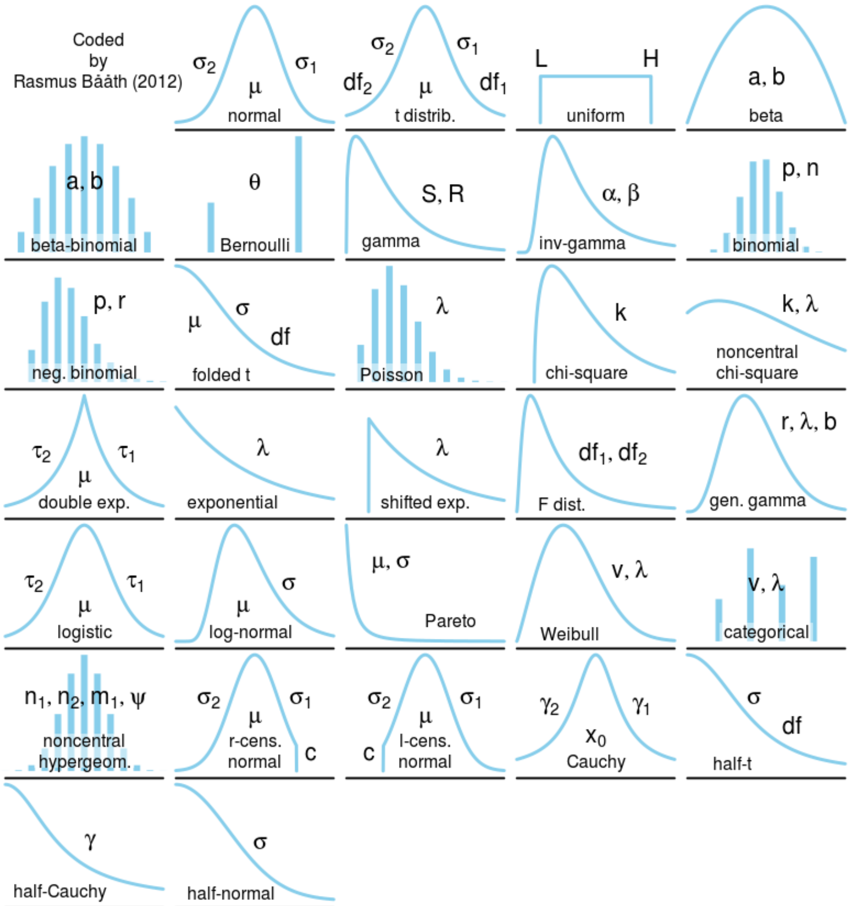
3.2 Prior parameter distributions

The concept of analyses beginning with "prior" assumptions, which are updated with data, is fundamental to Bayesian inference. In the claims reserving setting, the ability to set prior distributional assumptions for the claims process parameters also gives the experienced practitioner an opportunity to incorporate his or her knowledge into an analysis independently of the data.

An expert may have a view on the required shape of a parameter distribution and the level of uncertainty. Figure 3.1 (Bååth 2011) provides an overview of typical distributions. In order to select a sensible distribution, it can be helpful to consider the following questions:

- Is the data/parameter continuous or discrete?
- Is the data/parameter symmetric or asymmetric?
- Is the data/parameter clustered around a central value?
- How prevalent are outliers?
- Are the outliers positive or negative?

Figure 3.1. Schematic diagram of popular distributions and their parameters



There is no concept of a "prior" in frequentist procedures, and hence Bayesian approaches can offer greater flexibility when such prior knowledge exists. However, note that priors are starting points only, and the more data that are available, the less sensitive posterior inferences will be to those starting points.

3.3 Cumulative versus incremental data

Since we intend to model the full aggregated claims distribution at each development time, we have to carefully consider the impact the process variance assumption has on model behavior. This is particularly true for paid claims. Actual payments are incremental by nature, but we have the option to model cumulative payments. Does our choice matter?

Many traditional reserving methods (including the chain-ladder technique) take cumulative claims triangles as an input. Plotting cumulative claims development allows the actuary to quickly understand key data features by eye and identify the appropriateness of the selected projection technique.

In compartmental reserving models we estimate cumulative paid claims in the final compartment—a scaled (RRF) and delayed (k_p) version of the integrated outstanding claims—so it is also natural to visualize cumulative paid claims development. However, if we assume a constant (e.g., lognormal) CoV process distribution and model cumulative claims, this would imply more volatile paid claims over development time as payments cumulate. As a result, changes from one development period to the next would become more volatile. This feature is in direct contradiction to our intuition and the mean compartmental model solution, which expects less movement in the aggregate cumulative paid claims as fewer claims are outstanding.

To illustrate this concept and get us started with Bayesian model notation, we consider a simple growth curve model for cumulative paid loss ratio development.

Let's assume the loss ratio data-generating process can be modeled using a lognormal distribution, with the median loss ratio over time following a simple exponential growth curve. The loss ratio (ℓ_j) at any given development time (t_j) is modeled as the product of an expected loss ratio (ELR) and loss emergence pattern $G(t;\theta)$:

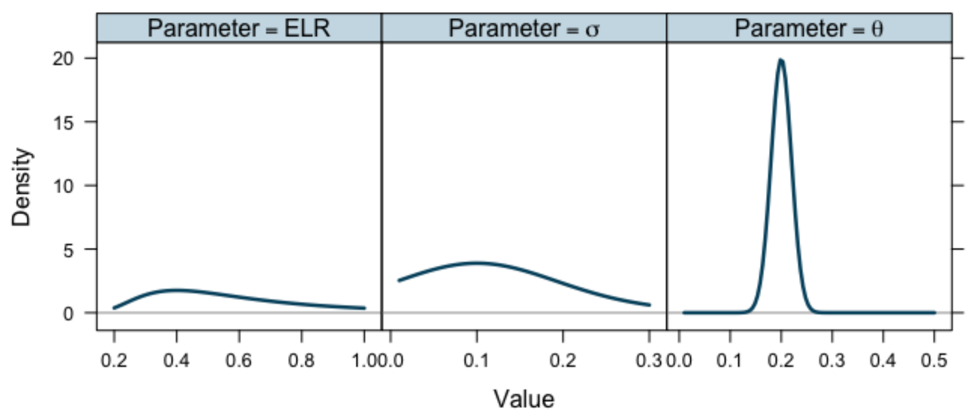
$$
\begin{aligned}
\ell_j &\sim \text{Lognormal}(\eta(t_j;\theta,\text{ELR}),\sigma)\\
\eta(t;\theta,\text{ELR}) &= \log(\text{ELR}\cdot G(t;\theta))\\
G(t;\theta) &= 1-e^{-\theta t}\\
\text{ELR} &\sim \text{InvGamma}(4,2)\\
\theta &\sim \text{Normal}(0.2,0.02)\\
\sigma &\sim \text{StudentT}(10,0.1,0.1)^{+}
\end{aligned}
$$

Note that we specify prior distributions for the parameters ELR, θ, and σ to express our uncertainty in these quantities. We assume that the expected loss ratio (ELR) follows an inverse gamma distribution to ensure positivity, but also allow for potential larger losses and hence poorer performance.

The parameter θ describes loss emergence speed, with $\ln(2)/\theta$ being the expected halfway-time of ultimate development. We set a Gaussian prior for this parameter, with a mean of 0.2 and standard deviation of 0.02, which implies that we expect 50% development of claims after around 3.5 years, but perhaps this occurs a month earlier or later.

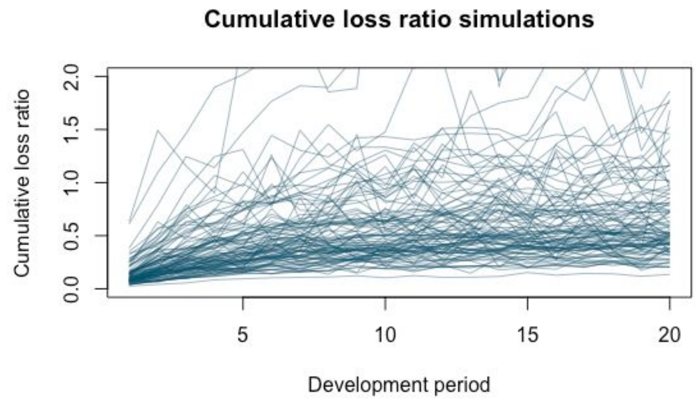
The process uncertainty (σ) has to be positive, so we assume a Student t-distribution left-truncated at 0. Figure 3.2 illustrates the prior parameter distributions.

Figure 3.2. Density plots of prior parameter distributions



Sampling from this model produces payment patterns with many negative increments in later development periods, as depicted in Figure 3.3.

Figure 3.3. Spaghetti plot of 100 simulated cumulative loss ratios



The reason for this behavior is the lognormal constant CoV, σ. As the mean loss ratio increases with development time, volatility increases as well, and there is no constraint in the model for the lognormal realizations to be increasing by development time.

However, this is not what we typically observe in development data. To account for this discrepancy, Meyers (2015) imposes a monotone decreasing constraint on the σ_j parameter with respect to development time, while Zhang, Dukic, and Guszcza (2012) and Morris (2016) include a first-order autoregressive error process.

Many others, including Zehnwirth and Barnett (2000) and Clark (2003), model incremental payments, for example as follows:

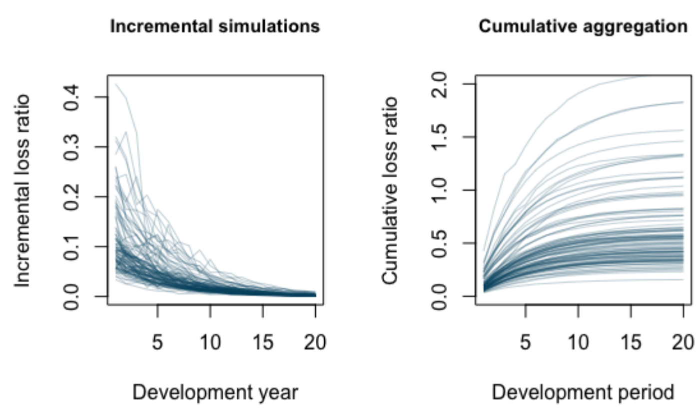
$$\eta(t_j; \theta, \mathrm{ELR}) = \log\big(\mathrm{ELR} \cdot [G(t_j; \theta) - G(t_{j-1}; \theta)]\big)$$

Modeling the incremental median payments with a lognormal distribution and a constant CoV is not only straightforward in the brms package in R, as shown in the code below, but the resultant simulations from the model appear more closely aligned to development data observed in practice, as shown in Figure 3.4.

```
myFun <- "
real incrGrowth(real t, real tfreq, real theta){
 real incrgrowth;
```

```
  incrgrowth = (1 - exp(-t * theta));
  if(t > tfreq){
   incrgrowth = incrgrowth - (1 - exp(-(t - tfreq) * theta));
  }
  return(incrgrowth);
}
"
prior_lognorm <- brm(
  bf(incr_lr ~ log(ELR * incrGrowth(t, 1.0, theta)),
     ELR ~ 1, theta ~ 1, nl=TRUE),
  prior = c(prior(inv_gamma(4, 2), nlpar = "ELR", lb=0),
            prior(normal(0.2, 0.02), nlpar = "theta", lb=0),
            prior(student_t(10, 0.1, 0.1), class = "sigma")),
  data = dat, file = "models/section_3/prior_lognorm",
  stanvars = stanvar(scode = myFun, block = "functions"),
  family = brmsfamily("lognormal"),
  sample_prior = "only")
```

Figure 3.4. Simulations of incremental claims payments and cumulative aggregation across development period



Additional factors that lead us to favor the use of incremental data include the following:

- Missing or corrupted data for some development periods can be problematic when we require cumulative data from the underlying incremental cash flows. Manual interpolation techniques can be used ahead of modeling, but a parametric growth curve applied to incremental data will deal with missing data as part of the modeling process.
- Changes in underlying processes (claims handling or inflation) causing effects in the calendar period dimension can be masked in cumulative data and are easier to identify and model using incremental data.
- Predictions of future payments are put on an additive scale, rather than a multiplicative scale, which avoids ad hoc anchoring of future claims projections to the latest cumulative data point.

3.4 Prior predictive examples

In this section, we provide three more examples of simulation models with different process distributions. These models are generative insofar as they are intended to emulate the data-generating process. However, their parameters are set manually as priors rather than estimated from data, so we term them "prior predictive" models.

The prior predictive distribution ($p(y)$) is also known as the marginal distribution of the data. It is the integral of the likelihood function with respect to the prior distribution

$$p(y) = \int p(y|\theta)p(\theta)d\theta,$$

and is not conditional on observed data.

Clark (2003) demonstrates how an overdispersed Poisson model can be fitted using maximum likelihood, and Guszcza (2008) illustrates the use of a Gaussian model with constant CoV.

Below, we showcase how these ideas can be implemented in Stan with brms.

At the end of the section we outline a prior predictive model for compartmental model (1) in the previous section.

3.4.1 Negative binomial process distribution

An overdispersed Poisson process distribution is assumed in Clark (2003), but here we will use a negative binomial distribution to model overdispersion generatively. This is also a standard family distribution in brms.[1]

The negative binomial distribution can be expressed as a Poisson(μ) distribution where μ is itself a random variable, coming from a gamma distribution with shape $\alpha = r$ and rate $\beta = (1 - p)/p$:

$$\begin{aligned} \mu &\sim \text{Gamma}(r, (1-p)/p) \\ y &\sim \text{Poisson}(\mu) \end{aligned}$$

Alternatively, we can specify a negative binomial distribution with mean parameter μ and dispersion parameter ϕ:

$$\begin{aligned} y &\sim \text{NegativeBinomial}(\mu, \phi) \\ E(y) &= \mu \\ \text{Var}(y) &= \mu + \mu^2/\phi \end{aligned}$$

The support for the negative binomial distribution is ℕ, and therefore we model dollar-rounded loss amounts (*Lj*) instead of loss ratios.

A growth curve model can be written as follows, with a log-link for the mean and shape parameters:

$$\begin{aligned} L_j &\sim \text{NegativeBinomial}(\mu(t_j; \theta, \Pi, \text{ELR}), \phi) \\ \mu(t_j; \theta, \text{ELR}) &= \log\Big(\Pi\,\text{ELR}\big(G(t_j;\theta) - G(t_{j-1};\theta)\big)\Big) \\ \theta &\sim \text{Normal}(0.2, 0.02)^+ \\ \text{ELR} &\sim \text{InvGamm}(4,2) \\ \phi &\sim \text{StudentT}(10, 0, \log(50))^+ \end{aligned}$$

This is straightforward to specify in brms:

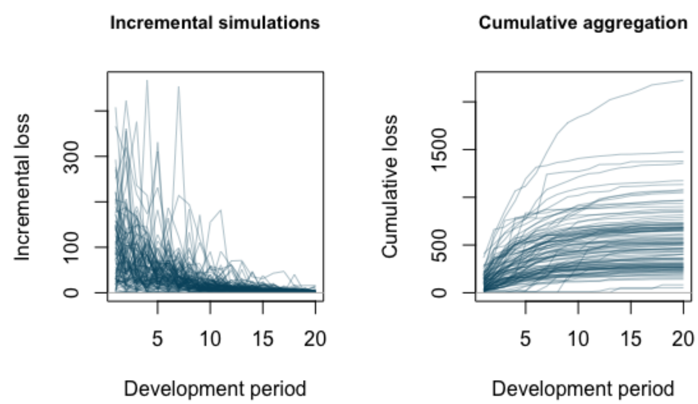
```
prior_negbin <- brm(
bf(incr  ~ log(premium * ELR * incrGrowth(t, 1.0, theta)),
```

[1] Custom distributions can be defined in brms too; see Bürkner (2020).

```
  ELR ~ 1, theta ~ 1, nl = TRUE),
 prior = c(prior(inv_gamma(4, 2), nlpar = "ELR"),
       prior(normal(0.2, 0.02), nlpar = "theta", lb=0),
       prior(student_t(10, 0, log(50)), class = "shape")),
data = dat, family = negbinomial(link = "log"),
stanvars = stanvar(scode = myFun, block = "functions"),
file="models/section_3/prior_negbin", sample_prior = "only")
```

This specification gives the outputs shown in Figure 3.5.

Figure 3.5. Prior predictive simulations of 100 losses with a negative binomial process distribution assumption



3.4.2 Gaussian process distribution with constant coefficient of variation

Guszcza (2008) proposes a Gaussian model with constant CoV to force the standard deviation to scale with the mean.

We can re-express our loss ratio growth curve model from earlier as

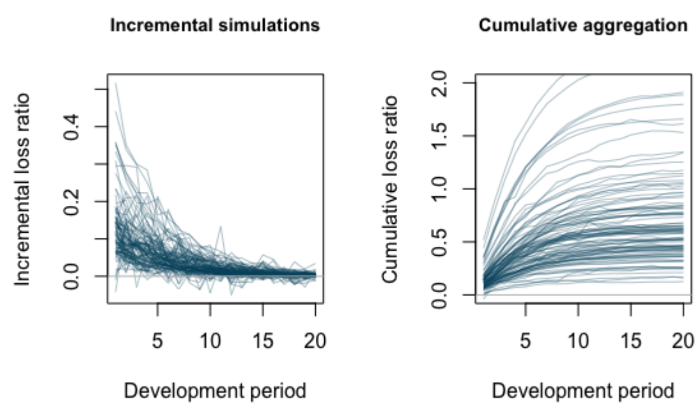
$$\begin{aligned}
\ell_j &\sim \text{Normal}(\eta(t_j;\Theta,\text{ELR}),\sigma_j)\\
\sigma_j &= \sigma\sqrt{\eta_j}\\
\eta(t_j;\theta,\text{ELR}) &= \text{ELR}\big(G(t_j;\Theta)-G(t_{j-1};\Theta)\big)\\
\theta &\sim \text{Normal}(0.2,0.02)^+\\
\text{ELR} &\sim \text{InvGamma}(4,2)\\
\sigma &\sim \text{StudentT}(10,0,0.1)^+,
\end{aligned}$$

which we can specify in brms once more (note the use of the nlf function here, which maps variables to nonlinear functions):

```
  prior_gaussian <- brm(
bf(incr_lr ~ eta,
  nlf(eta ~ ELR * incrGrowth(t, 1.0, theta)),
  nlf(sigma ~ tau * sqrt(eta)),
  ELR ~ 1, theta ~ 1, tau ~ 1, nl = TRUE),
data = dat, family = brmsfamily("gaussian", link_sigma = "identity"),
prior = c(prior(inv_gamma(4, 2), nlpar = "ELR"),
        prior(normal(0.2, 0.02), nlpar = "theta", lb=0),
        prior(student_t(10, 0, 0.1), nlpar = "tau", lb = 0)),
stanvars = stanvar(scode = myFun, block = "functions"),
file = "models/section_3/prior_gaussian",
sample_prior = "only")
```

Figure 3.6 illustrates the output of this approach.

Figure 3.6. Prior predictive simulations of 100 losses with a Gaussian process distribution assumption



3.4.3 Compartmental model with lognormal distribution

Finally, we simulate model output for our first compartmental model (1).

Compartmental models have a little more complexity than the growth curve models above, and so we have additional considerations for their implementation with brms and Stan:

- How to deal with the multivariate nature of the compartmental model, which is specified to fit paid and outstanding claims data simultaneously
- How to solve the ODEs numerically
- How to ensure that as the number of prior assumptions grows, their initialization values are valid

3.4.3.1 Compartmental model setup

To model paid and outstanding loss ratio data simultaneously, we stack both into a single column and add another column with an indicator variable. This indicator (δ) allows us to switch between the two claim stages and specify different variance levels (with a log link):

$$\begin{aligned}
y_j &\sim \text{Lognormal}(\mu(t_j;\Theta,\delta),\sigma_{[\delta]}) \\
\mu(t_j;\Theta,\delta) &= \log\Big((1-\delta)OS_j + \delta\big(PD_j - PD_{j-1}\big)\Big) \\
\sigma_{[\delta]} &= \exp\big((1-\delta)\beta_{OS} + \delta\,\beta_{PD}\big) \\
\delta &= \begin{cases} 0 \text{ if } y_j \text{ is outstanding claims} \\ 1 \text{ if } y_j \text{ is paid claims} \end{cases} \\
\Theta &= (k_{er}, k_p, RLR, RRF) \\
dEX_j/dt &= -k_{er}\cdot EX_j \\
dOS_j/dt &= k_{er}\cdot RLR\cdot EX_j - k_p\cdot OS_j \\
dPD_j/dt &= k_p\cdot RRF\cdot OS_j
\end{aligned} \tag{5}$$

Some of the more complex compartmental models described in the previous section have no analytical solutions for their ODE systems, forcing us to rely on numerical integration.

Fortunately, the Stan language contains a Runge-Kutta solver. We can write our solver in Stan and pass the code into brms in the same way as we did with the analytical growth curve solution earlier.

The Stan code below shows three functional blocks. The first function defines the ODE system, the second the solver, and the third the application to the data. Note the modeling of incremental paid claims for development periods greater than 1.

```
    myCompFun <- "
// ODE System
real[] ode_claimsprocess(real t, real [] y,  real [] theta,
                     real [] x_r, int[] x_i){
  real dydt[3];
  // Define ODEs
  dydt[1] = - theta[1] * y[1];
  dydt[2] = theta[1] * theta[3] * y[1] - theta[2] * y[2];
  dydt[3] = theta[2] * theta[4] * y[2];
   return dydt;
  }
//Priors & Solver
real int_claimsprocess(real t, real ker, real kp,
                    real RLR, real RRF, real delta){
  real y0[3];
  real y[1, 3];
  real theta[4];
  theta[1] = ker; theta[2] = kp;
  theta[3] = RLR; theta[4] = RRF;
  // Set initial values
  y0[1] = 1; y0[2] = 0; y0[3] = 0;

  y = integrate_ode_rk45(ode_claimsprocess,
                    y0, 0, rep_array(t, 1), theta,
                    rep_array(0.0, 0), rep_array(1, 1),
                    0.0001, 0.0001, 500); // tolerances, steps
     return (y[1, 2] * (1 - delta) + y[1, 3] * delta);
}
```

```
//Application to OS and Incremental Paid Data
real claimsprocess(real t, real devfreq, real ker, real kp,
                   real RLR, real RRF, real delta){
    real out;
    out = int_claimsprocess(t, ker, kp, RLR, RRF, delta);
    if( (delta > 0) && (t > devfreq) ){ // paid greater dev period 1
    // incremental paid
     out = out - int_claimsprocess(t - devfreq, ker, kp, RLR, RRF, delta);
    }
    return(out);
}
"
```

At the beginning of the HMC simulation Stan initializes all parameter values randomly between -2 and 2. Although these can be changed by the user, the default initializations can cause issues for parameters that cannot be negative in the model. To avoid setting multiple initial values, it is common practice to define parameters on an unconstrained scale and transform them to the required scale afterwards.

For our example, we will assume all compartmental model parameter priors are lognormally distributed. For the implementation, however, we use standardized Gaussians and transform them to lognormal distributions using the nlf function.

$$
\begin{aligned}
\mathrm{RLR} &\sim \mathrm{Lognormal}(\log(0.6), 0.1)\\
\mathrm{RRF} &\sim \mathrm{Lognormal}(\log(0.95), 0.05)\\
\mathrm{k_{er}} &\sim \mathrm{Lognormal}(\log(1.7), 0.02)\\
\mathrm{k_p} &\sim \mathrm{Lognormal}(\log(0.5), 0.05)
\end{aligned}
$$

We assume Gaussians for β_{OS} and β_{PD}, with the volatility for outstanding loss ratios slightly higher than for paid loss ratios:

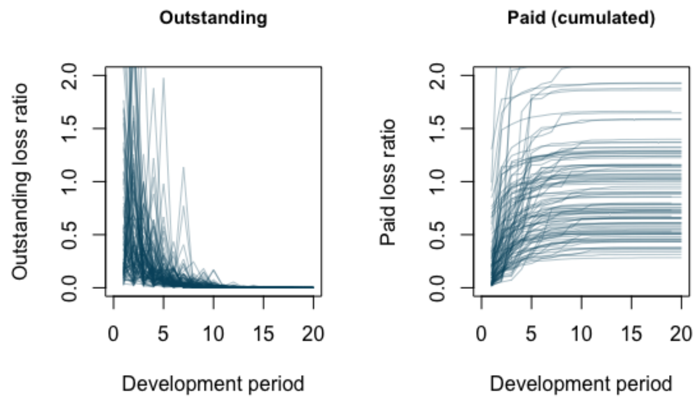
$$
\begin{aligned}
\beta_{OS} &\sim \mathrm{Normal}(0.15, 0.025)\\
\beta_{PD} &\sim \mathrm{Normal}(0.1, 0.02)
\end{aligned}
$$

The above implies a lognormal distribution for $\sigma_{[\delta]}$ given the log link.

Now that we have prepared our model, we can simulate from it with brms (below) and review the prior predictive output (Figure 3.7).

```
  frml <- bf(
 incr_lr ~ eta,
 nlf(eta ~ log(claimsprocess(t, 1.0, ker, kp, RLR, RRF, delta))),
 nlf(RLR ~ 0.6 * exp(oRLR * 0.1)),
 nlf(RRF ~ 0.95 * exp(oRRF * 0.05)),
 nlf(ker ~ 1.7 * exp(oker * 0.02)),
 nlf(kp ~ 0.5 * exp(okp * 0.05)),
 oRLR ~ 1, oRRF ~ 1, oker ~ 1, okp ~ 1, sigma ~ 0 + deltaf,
 nl = TRUE)
mypriors <- c(prior(normal(0, 1), nlpar = "oRLR"),
         prior(normal(0, 1), nlpar = "oRRF"),
         prior(normal(0, 1), nlpar = "oker"),
         prior(normal(0, 1), nlpar = "okp"),
         prior(normal(0.1, 0.02), class = "b",  coef="deltafpaid", dpar = "sigma"),
         prior(normal(0.15, 0.025), class = "b",  coef="deltafos", dpar= "sigma"))
prior_compartment_lognorm <- brm(frml, data = dat,
 family = brmsfamily("lognormal", link_sigma = "log"),
 prior = mypriors,
 stanvars = stanvar(scode = myCompFun, block = "functions"),
 file="models/section_3/prior_compartment_lognorm",
 sample_prior = "only")
```

Figure 3.7. Prior predictive simulations of 100 outstanding and paid development paths with a lognormal process distribution assumption



The prior predictive simulations appear to resemble development data, despite having not used any real data to generate them.

As part of a robust Bayesian work flow, one should next try to fit the model to a sample of the prior predictive distribution to establish whether the model parameters are identifiable (Betancourt 2018). This is left as an exercise for the reader.

4. Modeling hierarchical structures and correlations

In the previous section we discussed generative models for claims development. We will continue this line of thought and add more complexity in the form of hierarchies for describing claims emergence pattern variation by accident year.

4.1 Introduction to hierarchical models

Hierarchical and multilevel models are popular tools in the social and life sciences. A typical motivation for their use is to understand which characteristics are shared among individuals within a population as well as which ones vary, and to what extent. In the frequentist setting, these models are usually termed “mixed-effects” or “fixed- and random-effects” models.

In insurance we face similar challenges: we want to learn as much as possible at a credible “population” level and make adjustments for individual cohorts or policyholders. The Bühlmann-Straub credibility pricing model (Bühlmann and Straub 1970) is a special case of a hierarchical model.

Hierarchical models have been proposed in the reserving literature previously, for example by Antonio et al. (2006); Guszcza (2008); Zhang, Dukic, and Guszcza (2012); and Morris (2016).

When it comes to claims reserving, we typically consider the aspects of the data-generating process that we believe to be the same across a dimension and those that will vary “randomly,” for the purpose of the model.

It is generally assumed that the loss emergence pattern of claims is similar across accident years, while aggregate loss amounts themselves vary given the “random” nature of loss event occurrence and severity.

For example, the standard chain-ladder method derives a single loss emergence pattern from a claims triangle. The loss development factors are applied to the most recent cumulative claims positions to provide ultimate loss forecasts by accident year. However, the latest cumulative claims positions are the result of a random process for which volatility tends to dominate in earlier development periods (i.e., younger accident years), leading to highly sensitive projections from the chain-ladder approach. This issue is often addressed by using the Bornhuetter-Ferguson method (Bornhuetter and Ferguson 1972), which incorporates prior information on expected loss ratios and uses the loss emergence expectation as a credibility weight for the chain-ladder forecast.

Hierarchical compartmental models provide a flexible framework to simultaneously model the fixed and random components of the claim development process.

4.2 Specifying a hierarchy

The previous section presented models that we can extend to be hierarchical. For example, we could assume that the pattern of loss emergence is the same across accident years and that expected loss ratios vary "randomly" by accident year i around a central value ELR_c:

$$\begin{aligned}
\ell_{ij} &\sim \text{Lognormal}(\eta(t_j; \theta, \text{ELR}_{[i]}), \sigma) \\
\eta(t; \theta, \text{ELR}_{[i]}) &= \log(\text{ELR}_{[i]} \cdot (G(t_j; \theta) - G(t_{j-1}; \theta)) \\
&= \log\left(\text{ELR}_{[i]}\right) + \log\left(G(t_j; \theta) - G(t_{j-1}; \theta)\right) \\
\text{ELR}_{[i]} &\sim \text{Lognormal}(\log(\text{ELR}_c), \sigma_{[i]}) \\
\text{ELR}_c &\sim \text{Lognormal}(\log(0.6), 0.1) \\
\sigma_{[i]} &\sim \text{StudentT}(10, 0, 0.05)^+ \\
\theta &\sim \text{Normal}(0.2, 0.02) \\
\sigma &\sim \text{StudentT}(10, 0, 0.05)^+
\end{aligned}$$

This parameterization is known as the "centered" approach, whereby individual ELR estimates are distributed around an average or central value. For subsequent models, we replace lines 4–6 above with the following structure:

$$\begin{aligned}
\log(\text{ELR}_{[i]}) &= \mu_{\text{ELR}} + u_{[i]} \\
u_{[i]} &= \sigma_{[i]} z_{[i]} \\
\mu_{\text{ELR}} &\sim \text{Normal}(\log(0.6), 0.1) \\
\sigma_{[i]} &\sim \text{StudentT}(10, 0, 0.025)^+ \\
z_{[i]} &\sim \text{Normal}(0, 1)
\end{aligned}$$

In this specification, individual effects are estimated around the population as additive perturbations, relating naturally to the "fixed" and "random" effects terminology. However, this is a potential source of confusion in the Bayesian setting, where all parameters are random variables. We therefore opt for the terms “population” and “varying” in lieu of “fixed” and “random” to avoid such confusion.

The second “noncentered” parameterization is the default approach in brms for hierarchical models because it often improves convergence, so we adopt it for all hierarchical models fitted in this paper.

4.3 Regularization

Hierarchical models provide an adaptive regularization framework in which the model learns how much weight subgroups of data should get, which can help to reduce overfitting. This is effectively a credibility weighting technique. Setting a small value for $\sigma_{[i]}$ above ensures that sparse data (e.g., for the most recent accident year) has limited influence on $ELR_{[i]}$. In this scenario, our estimate of $\log(\mathrm{ELR}_{[\mathrm{i}]})$ will “shrink” more heavily toward μ_{ELR}.

Regularization allows us to estimate the parameters of more complex and thus more flexible models with greater stability and confidence. For example, as we noted earlier, the multistage model (4) collapses into the simpler model (2) with $\mathrm{k_e} = 1$ and $\mathrm{k_{p_2}} = 0$. We can therefore use the more complex model with priors centered on 1 and 0 to allow flexibility, but only where the data provide a credible signal away from our prior beliefs. In this sense, we can estimate parameters for models that would be considered “overparameterized” in a traditional maximum-likelihood setting.

4.4 Market cycles

For the compartmental models introduced in Section 2, hierarchical techniques allow us to estimate “random” variation in reported loss ratios and reserve robustness factors across accident years.

However, changes in the macroeconomic environment, as well as internal changes to pricing strategy, claims settlement processes, and teams, can also impact underwriting and reserving performance in a more systematic manner.

Where information relating to such changes exists, we can use it in our modeling to provide more informative priors for the reported loss ratios and reserve robustness factors by accident year.

One approach is to on-level the parameters across years. Suppose we have data on historical cycles in the form of indices, with RLM_i describing reported loss ratio multipliers and RRM_i describing reserve robustness change multipliers on a base accident year.

Sources for the reported loss ratio multipliers could be risk-adjusted rate changes or planning or pricing loss ratios, while the reserve robustness multipliers could be aligned with internal claims metrics.

This data (or judgment, or both) can be used to derive prior parameters $RLR_{[i]}$ and $RRF_{[i]}$ by accident year as follows:

$$\begin{aligned} \mathrm{RLR}_{[i]} &= \mathrm{RLR}_{\mathrm{base}} \cdot \mathrm{RLM}_{\mathrm{i}}^{\lambda_{\mathrm{RLR}}} \\ \mathrm{RRF}_{[i]} &= \mathrm{RRF}_{\mathrm{base}} \cdot \mathrm{RRM}_{\mathrm{i}}^{\lambda_{\mathrm{RRF}}} \end{aligned}$$

For each accident year, we specify parameter priors as the product of a base parameter (e.g., the expected loss ratio for the oldest year) and an index value for that year. We also introduce additional parameters $\lambda_{RLR}, \lambda_{RRF}$ to describe the extent to which expected loss ratios correlate with the indices.

On a log scale this implies a simple additive linear relationship:

$$\begin{aligned} \mu_{\mathrm{RLR}_{[i]}} &= \mu_{\mathrm{RLR}} + \lambda_{\mathrm{RLR}} \log(\mathrm{RLM}_{\mathrm{i}}) \\ \mu_{\mathrm{RRF}_{[i]}} &= \mu_{\mathrm{RRF}} + \lambda_{\mathrm{RRF}} \log(\mathrm{RRM}_{\mathrm{i}}) \end{aligned}$$

For interpretability, the reported loss ratio and reserve robustness multiplier should be set to 1 for the base accident year. Under this assumption it may be preferable to set prior assumptions for λ close to 1 also, provided the indices are considered reliable. Furthermore, the credibility parameters λ could be allowed to vary by accident year.

A weakness of this approach is that any index uncertainty observed or estimated in earlier years does not propagate into more recent years. Additionally, the λ parameters have minimal influence for years in which the indices are close to 1. Although this allows us to set loss ratio priors for each year individually, we could instead adopt time series submodels for the RLR and RRF parameters to address these limitations.

The next section illustrates how to build a hierarchical compartmental model for a single claims triangle. To keep the example compact, we will touch on market cycles but not model them directly. However, the case study in Section 5 will explicitly take market cycles into account. The corresponding R code is presented in the appendix.

4.5 Single-triangle hierarchical compartmental model

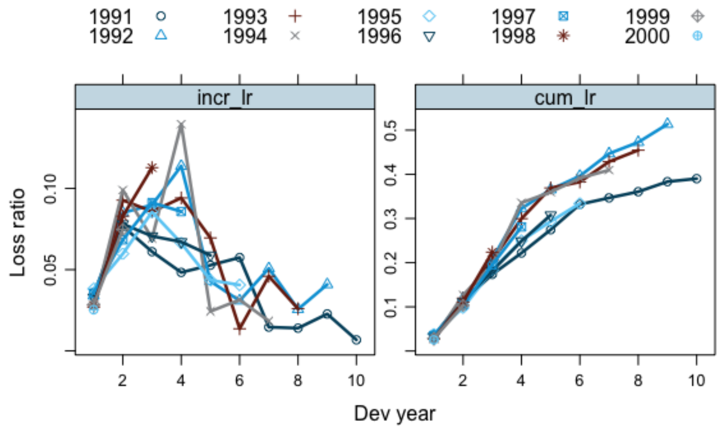
This example uses the classic "GenIns" paid triangle (Taylor and Ashe 1983) from the ChainLadder package (Gesmann et al. 2019). The triangle has been used in many reserving papers, including Mack (1993), Clark (2003), and Guszcza (2008). We also use the premium information given in Clark (2003) for our analysis (see Table 4.1).

Table 4.1 Premiums and cumulative paid claims triangle, with values shown in thousands

	Premium	1	2	3	4	5	6	7	8	9	10
1991	10,000	358	1,125	1,735	2,218	2,746	3,320	3,466	3,606	3,834	3,901
1992	10,400	352	1,236	2,170	3,353	3,799	4,120	4,648	4,914	5,339	
1993	10,800	291	1,292	2,219	3,235	3,986	4,133	4,629	4,909		
1994	11,200	311	1,419	2,195	3,757	4,030	4,382	4,588			
1995	11,600	443	1,136	2,128	2,898	3,403	3,873				
1996	12,000	396	1,333	2,181	2,986	3,692					
1997	12,400	441	1,288	2,420	3,483						
1998	12,800	359	1,421	2,864							
1999	13,200	377	1,363								
2000	13,600	344									

The incremental data in Figure 4.1 exhibit substantial volatility, in both quantum and development behavior. Some of the variance seen in the cumulative loss ratio development could be attributed to risk-adjusted rate changes across accident years.

Figure 4.1. Example triangle of incremental and cumulative paid loss ratio development by accident year



With a single payment triangle we are still able to use a hierarchical compartmental model, such as model (4), to model paid loss ratio loss emergence. This is similar to fitting a hierarchical growth curve model; however, we will not be able to make inferences about case reserve robustness.

We allow all compartmental model parameters to vary by accident year and again use the nlf function to transform parameters from `Normal(0,1)` into lognormal priors:

```
frml <- bf(incr_lr ~ eta,
    nlf(eta ~ log(ELR * lossemergence(dev, 1.0, ke, dr, kp1, kp2))),
    nlf(ke ~ exp(oke * 0.5)),
    nlf(dr ~ 1 + 0.1 * exp(odr * 0.5)),
    nlf(kp1 ~ 0.5 * exp(okp1 * 0.5)),
    nlf(kp2 ~ 0.1 * exp(okp2 * 0.5)),
    ELR ~ 1 + (1 | AY),
    oke  ~ 1 + (1 | AY), odr ~ 1 + (1 | AY),
    okp1 ~ 1 + (1 | AY), okp2 ~ 1 + (1 | AY),
    nl = TRUE)
```

We set prior parameter distributions similar to those in the previous section and add priors for the Gaussian perturbation terms of the varying effects. The standard deviations for these are set to narrow Student's t-distributions as regularization to prevent overfitting:

```
mypriors <- c(prior(inv_gamma(4, 2), nlpar = "ELR", lb=0),
        prior(normal(0, 1), nlpar = "oke"),
        prior(normal(0, 1), nlpar = "odr"),
        prior(normal(0, 1), nlpar = "okp1"),
        prior(normal(0, 1), nlpar = "okp2"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "ELR"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "oke"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "odr"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "okp1"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "okp2"),
        prior(student_t(10, 0, 1), class = "sigma"))
```

Now we can estimate the posterior distributions for all of the parameters in our model:

```
fit_loss <- brm(frml, prior = mypriors,
         data = lossDat, family = lognormal(), seed = 12345,
         stanvars = stanvar(scode = myFuns, block = "functions"),
         file="models/section_4/GenInsIncModelLog")
```

The model run does not report any obvious warnings. Diagnostics such as $\hat{R}$ and effective sample size look good, so we move on to reviewing the outputs. The case study in the next section will cover model review and validation in more detail; hence we keep it brief here.

We note that the population k_e and k_{p_2} from the extended compartmental model are identified with 95% posterior credible intervals that scarcely contain 1 and 0, respectively, indicating possible support for this model structure (see Table 4.2).

Table 4.2. Population-level estimates

	Estimate	Est. Error	l–95% CI	u–95% CI
ELR	0.491	0.033	0.428	0.559
ke	0.660	0.179	0.382	1.070
dr	1.145	0.077	1.045	1.344
kp1	0.428	0.134	0.249	0.752
kp2	0.113	0.059	0.039	0.272

Notwithstanding data volatility, the model appears reasonably well behaved against the historical data (see Figure 4.2).

Figure 4.2. Posterior predictive distribution for each accident and development year, showing the predicted means and 95 percent predictive intervals

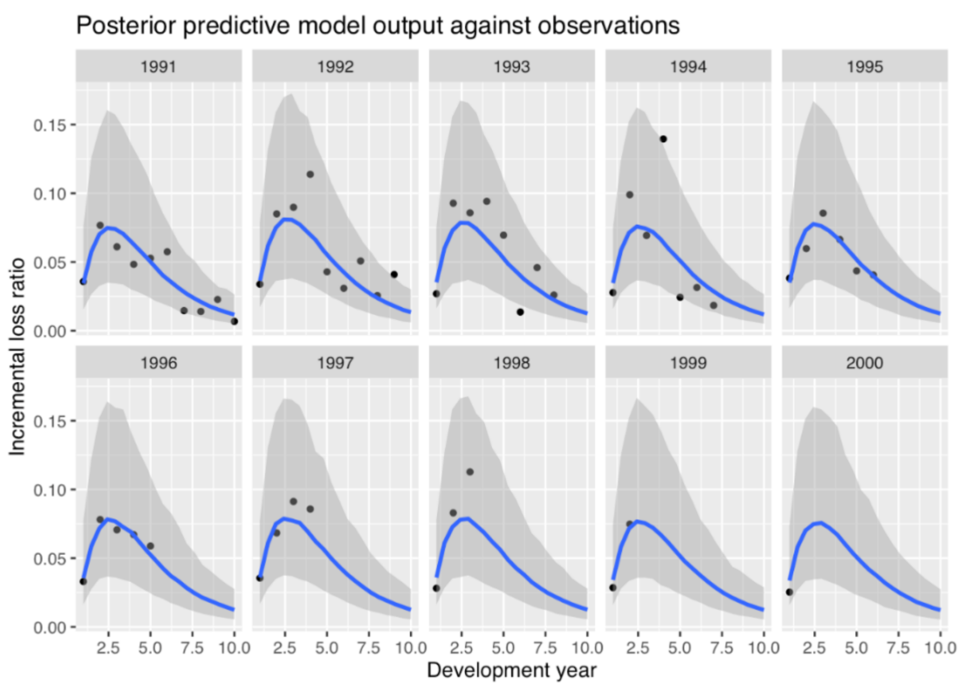
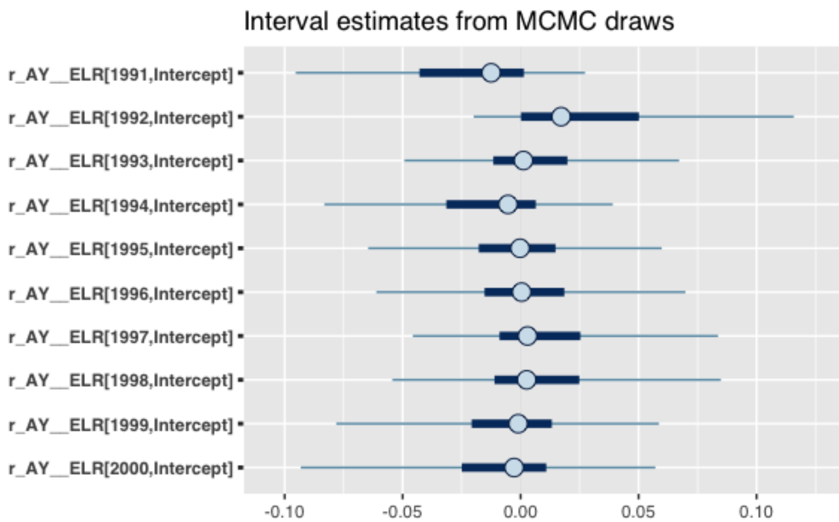


Figure 4.3 plots 50% and 90% posterior credible intervals for each accident year's estimated deviation from the population ELR on the log scale. This allows us to inspect how variable the

model estimates performance to be across accident years and the levels of uncertainty for each year.

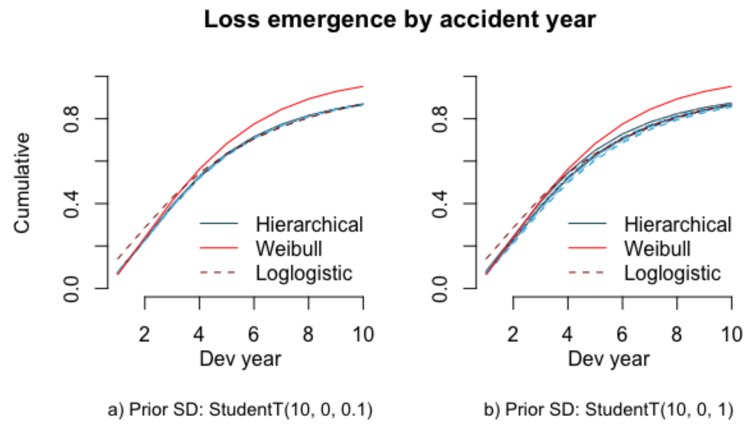
Figure 4.3. Posterior credible intervals from HMC draws of ELR by accident year, showing the expected performance variance across all years



Observe that all credible intervals contain 0, so we cannot be sure that any one year's ELR is different from the population average. However, there is some evidence of deviation across years, which, as observed in the cumulative paid developments, could be attributed to historical rate changes.

In addition, we compare the posterior mean loss emergence pattern by accident year against the Cape Cod method outlined in Clark (2003) with maximum age set to 20, as implemented in Gesmann et al. (2019). Figure 4.4, panel (a), shows that the selected compartmental model's loss emergence patterns do not vary much across accident years due to our relatively tight priors, mirroring initially the Weibull curve and, for later development years, the loglogistic growth curve.

Figure 4.4. Comparing hierarchical growth curves with different prior parameter distributions



If we increase the uncertainty of the hyperprior parameter distributions from $\text{StudentT}(10, 0, 0.1)$ to $\text{StudentT}(10, 0, 1)$, then the individual accident year development data gets more weight, and estimated loss emergence patterns start to exhibit some variance across accident years, shown in Figure 4.4, panel (b).

4.6 Expected versus ultimate loss

We parameterized the above model in terms of $ELRs$ (expected loss ratios) rather than $ULRs$ (ultimate loss ratios). This was deliberate, since our model aims to estimate the latent parameter of the underlying development process.

For a given accident year, the ELR parameter describes the underlying expected loss ratio in the statistical process that generated the year's loss emergence. Put another way, if an accident year were to play out repeatedly and infinitely from scratch, then the ELR is an estimate of the average ultimate loss ratio over all possible scenarios.

In reality, of course, we can observe only a single realization of the claims development for each accident year. This realization will generate the ULR (the actual ultimate loss ratio), which derives from the sum of payments to date plus estimated future incremental payments.

Hence, the ultimate loss (or ultimate loss ratio) is anchored to the latest cumulative payment, while the expected loss ratio treats the payments to date as one random series of payment realizations.

Table 4.3. Expected and ultimate losses (to age 20) with respective estimated standard errors of second model with wider hyperprior parameters

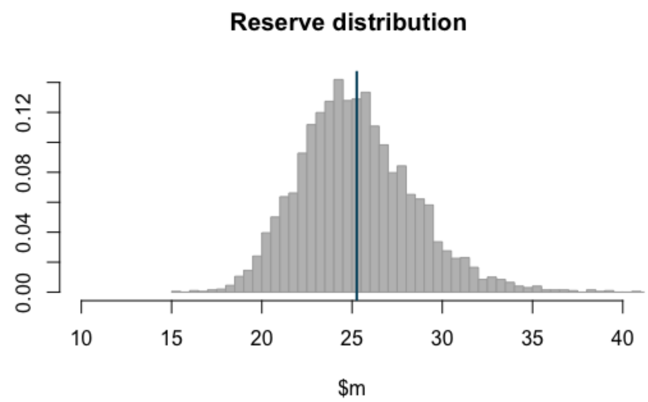
AY	ELR (%)	Est. error	ULR (%)	Est. error
1991	46.6	4.4	43.2	1.4
1992	52.7	5.4	58.7	2.2
1993	49.7	4.6	53.5	2.4
1994	47.4	4.8	50.2	2.9
1995	49.0	4.9	47.0	3.8
1996	49.5	5.4	49.1	4.7
1997	50.5	6.1	52.2	5.7
1998	50.4	6.0	53.8	6.7
1999	48.7	6.2	49.1	7.7
2000	48.3	6.7	48.5	8.6

The estimated ULR standard errors are driven only by the estimated errors for future payments; hence they increase significantly across newer accident years as ultimate uncertainty increases. The estimated errors around the ELR are more stable, as are the estimated mean ELR values. The differences between expected and ultimate loss ratios demonstrate the point made above: the ULR is anchored to the latest payment value and is therefore influenced by that particular random series of payments to date, as shown in Table 4.3.

Finally, we can review the distribution of the posterior predictive reserve, derived as the sum of extrapolated future payments to development year 20, or calendar year 2010 (see Figure 4.5).

The reserve distribution is not an add-on, but part of the model output.

Figure 4.5. Histogram of posterior predictive payments up to development year 20 of second model with wider hyperprior parameters. Mean model reserve highlighted in blue.



Note that reserve uncertainty can be analyzed in more detail, since the model returns the full posterior distribution for all parameters and hence, predictions, by accident year and development year.

4.7 Correlations across effects

As a further step we could test for correlations between parameters across accident years. For example, we might expect that lower loss ratios correlate with a faster payment speed.

Assuming a centered multivariate Gaussian distribution for the varying effects with an LKJ prior (Lewandowski, Kurowicka, and Joe 2009) for the correlations becomes somewhat cumbersome to write down in mathematical notation. However, if we define the growth curve to include $ELR_{[i]}$ as a parameter,

$$\tilde{G}(t; ELR_{[i]}, k_{e_{[i]}}, d_{r_{[i]}}, k_{p_{1[i]}}, k_{p_{2[i]}}) := ELR_{[i]} \cdot G(t; k_{e_{[i]}}, d_{r_{[i]}}, k_{p_{1[i]}}, k_{p_{2[i]}}),$$

then the notation for the varying-effects correlated model can be written as follows:

$$
\begin{aligned}
\ell_{ij} &\sim \mathrm{Lognormal}(\eta_i(t), \sigma) \\
\eta_i(t) &= \log\left(\tilde{G}(t; \Theta)\right) \\
\Theta &= \mu_\Theta + u_\Theta \\
\mu_\Theta &= \left(\mu_{ELR}, \mu_{k_e}, \mu_{d_r}, \mu_{k_{p_1}}, \mu_{k_{p_2}}\right) \\
u_\Theta &= \left(u_{ELR}, u_{k_e}, u_{d_r}, u_{k_{p_1}}, u_{k_{p_2}}\right) \\
\text{Priors} &: \\
\sigma &\sim \mathrm{StudentT}(3,0,0.1)^+ \\
\mu_{ELR} &\sim \mathrm{InvGamma}(4,2) \\
\mu_{k_e}, \mu_{d_r} &\sim \mathrm{Lognormal}(\log(3),0.2) \\
\mu_{k_{p_1}} &\sim \mathrm{Lognormal}(\log(0.5),0.1) \\
\mu_{k_{p_2}} &\sim \mathrm{Normal}(0.2,0.5) \\
u_\Theta &\sim \mathrm{MultivariateNormal}(\mathbf{0}, \mathbf{\Sigma}) \\
\mathbf{\Sigma} &= \mathbf{D}\,\mathbf{\Omega}\,\mathbf{D} \\
\mathbf{D} &= \mathrm{Diag}\left(\sigma_{ELR}, \sigma_{k_e}, \sigma_{d_r}, \sigma_{k_{p1}}, \sigma_{k_{p2}}\right) \\
(\sigma_{ELR}, \sigma_{k_e}, \sigma_{d_r}, \sigma_{k_{p1}}, \sigma_{k_{p2}}) &\sim \mathrm{StudentT}(3,0,0.1)^+ \\
\mathbf{\Omega} &\sim \mathrm{LkjCorr}(1)
\end{aligned}
\tag{6}
$$

Implementing this correlation structure in brms is straightforward; we simply add a unique character string to each varying effect term:

```
frml <- bf(incr_lr ~ log(ELR * lossemergence(dev, 1.0, ke, dr, kp1, kp2)),
   ELR ~ 1 + (1 | ID | AY),
   ke  ~ 1 + (1 | ID | AY), dr ~ 1 + (1 | ID | AY),
   kp1 ~ 1 + (1 | ID | AY), kp2 ~ 1 + (1 | ID | AY),
   nl = TRUE)
```

This notation naturally extends further. Suppose we have development data by company and accident year, as in Zhang, Dukic, and Guszcza (2012), and would like to model a structure that allows ELR to vary by accident year and company. With k_{er} and k_p constant by accident year but varying by company, and correlating ELR, k_{er}, and k_p by company, we can write the following:

```
(ELR ~ 1 + (1 | ID | company) + (1 | AY:company),
ker ~ 1 + (1 | ID | company),
kp  ~ 1 + (1 | ID | company))
```

An implementation of a similar multicompany model on a cumulative loss ratio basis in brms is given in Gesmann (2018).

These examples illustrate how brms provides a powerful and rich framework to build complex models using intuitive model notation. For more detail, please refer to the various brms and RStan vignettes. Note that the underlying Stan code can be extracted from any brms object using the stancode function.

5. Compartmental reserving case study

In this section, we demonstrate how to fit hierarchical compartmental models of varying complexity to paid and outstanding claims development data simultaneously. We also introduce models with parameter variation by both accident and development year, in addition to an application of the previously outlined approach for integrating pricing and reserving cycle information into the modeling process.

Our first model is based on the case study presented in Morris (2016), which models outstanding and cumulative paid claims using a Gaussian distribution. In Section 3 we proposed modeling incremental payments with a right-skewed process distribution (e.g., lognormal), which a second model demonstrates. Model 2 also introduces parameter variation by development year, showcasing the usability and flexibility of brms for specifying varying effects. Finally, we build on this work further with a third model, which incorporates pricing and reserving cycle trends into the modeling process. The purpose of this procedure is to capture performance drift across time and apply judgment for individual years—particularly for less mature years, where hierarchical growth curve approaches typically shrink parameters back to an “all-years” average.

The work flow in this case study involves six steps:

1. **Data preparation:** Create a training data set, up to the penultimate calendar year, and a test data set, based on the most recent calendar year.
2. **Model building:** Develop model structures including process distributions, prior parameter distributions, and hierarchical levels. We omit prior predictive distribution reviews in the text for brevity (see Section 3.4 for more detail).
3. **Training:** Fit the models on the training data and review in-sample posterior predictive checks.
4. **Testing and selection:** Review each model’s predictions against the latest calendar year’s paid loss ratios, and select the most appropriate model.

5. **Fitting:** Refit (train) the selected model on the combined training and test data, which include the latest calendar year.

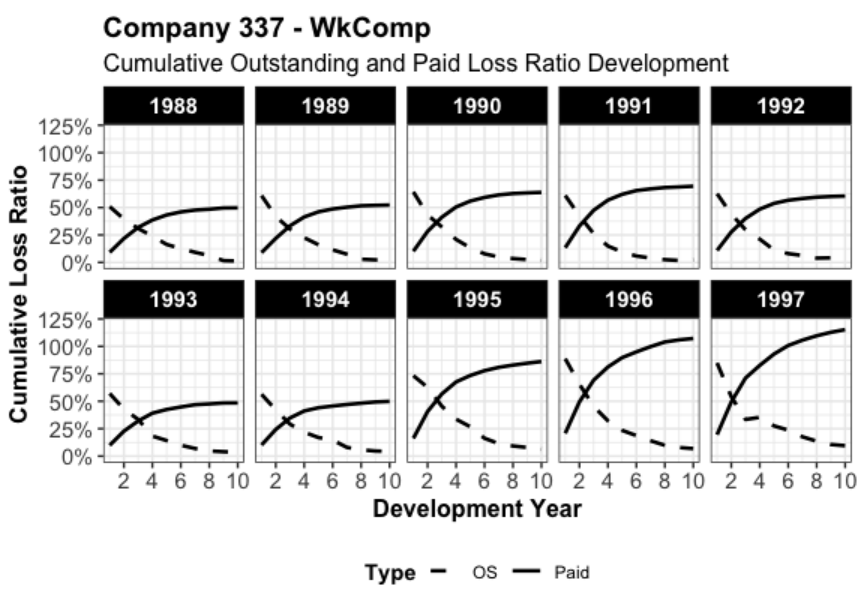
6. **Reserving:** Review the final model, predict future claims payments, and set the reserve.

We also have the "lower triangle" of development, which allows us to critique our reserve estimates against actual values.

5.1 Data preparation

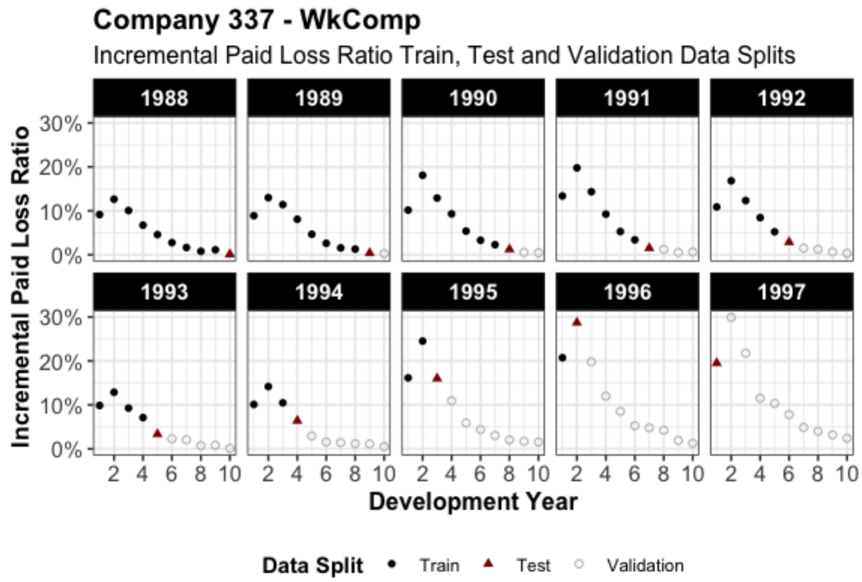
The data comprise Schedule P Workers' Compensation paid and incurred development data for company 337 (Fannin 2018), as in the Morris (2016) case study, depicted in Figure 5.1.

Figure 5.1. Paid and outstanding loss ratio development by accident year, shown for all 10 development years



We split the data into a training set, with data up to 1996; a test data set, which contains the 1997 calendar year movement; and finally, to review our reserve, a "validation" set, which contains development for all accident years to age 10. Figure 5.2 shows the validation scheme for incremental paid loss ratios.

Figure 5.2. Training, testing, and validation data splits



5.2 Model building

We build and train three hierarchical compartmental reserving models (see the appendix for the corresponding code):

- **Model 1:** A Gaussian distribution model, based on the original case study in Morris (2016), fitted to outstanding and cumulative paid amounts
- **Model 2:** A lognormal distribution model, fitted to outstanding and incremental paid loss ratios with additional parameter variation by development year
- **Model 3:** An enhancement of model 2 that incorporates market cycle data and judgment to influence forecasts for less mature accident years

5.2.1 Model 1: Gaussian distribution

Model 1 is analogous to the first model outlined the original case study in Morris (2016). We assume a Gaussian process distribution and model outstanding and cumulative paid amounts. For the hierarchical structure, we assume k_{er}, k_p are fixed by accident year and that $RLR_{[i]}, RRF_{[i]}$ have correlated varying effects by accident year, with a weak LKJ prior on the correlation between them.

$$
\begin{aligned}
y_{ij} &\sim \mathrm{Normal}\big(\mu(t_{ij};\Theta,\delta),\sigma_{[\delta]}\big)\\
\mu(t_{ij};\Theta,\delta) &= (1-\delta)OS_{ij} + \delta PD_{ij}\\
\delta &= \begin{cases} 0 \text{ if } y_{ij} \text{ is outstanding claim}\\ 1 \text{ if } y_{ij} \text{ is paid claim}\end{cases}\\
\Theta &= \big(k_{er}, RLR_{[i]}, k_p, RRF_{[i]}\big)\\
OS_{ij} &= \Pi_i\, RLR_{[i]} \frac{k_{er}}{k_{er}-k_p}\big(e^{-k_p t_j} - e^{-k_{er} t_j}\big)\\
PD_{ij} &= \Pi_i\, RLR_{[i]}\, RRF_{[i]} \frac{1}{k_{er}-k_p}\Big(k_{er}\big(1-e^{-k_p t_j}\big) - k_p\big(1-e^{-k_{er} t_j}\big)\Big)
\end{aligned}
\tag{7}
$$

Next, we specify priors for the parameters being estimated, based on judgment and intuition:

$$
\begin{aligned}
\log\,(\sigma_{[\delta]}) &\sim \mathrm{StudentT}(1,0,1000)\\
\\
k_{er} &\sim \mathrm{Lognormal}(\log(3),0.1)\\
k_p &\sim \mathrm{Lognormal}(\log(1),0.1)\\
RLR_{[i]} &\sim \mu_{RLR} + u_{RLR}\\
RRF_{[i]} &\sim \mu_{RLR} + u_{RRF}\\
\mu_{RLR} &\sim \mathrm{Lognormal}(\log(0.7),0.2)\\
\mu_{RRF} &\sim \mathrm{Lognormal}(\log(0.8),0.1)\\
(u_{RLR}, u_{RRF})' &\sim \mathrm{MultivariateNormal}(\mathbf{0}, \mathbf{D}\,\mathbf{\Omega}\,\mathbf{D})\\
\mathbf{\Omega} &\sim \mathrm{LKJCorr}(1)\\
\mathbf{D} &= \begin{pmatrix} \sigma_{RLR} & 0\\ 0 & \sigma_{RRF}\end{pmatrix}\\
\sigma_{RLR} &\sim \mathrm{StudentT}(10,0,0.2)^{+}\\
\sigma_{RRF} &\sim \mathrm{StudentT}(10,0,0.1)^{+}
\end{aligned}
$$

In summary, we anticipate the following:

- **A moderately high reported loss ratio**, reflected by a prior median *RLR* equal to 70%, with prior CoV around 20%
- **A relatively fast rate of reporting**, reflected by a prior median k_{er} equal to 3. This gives a value of claims reported in the first development year equal to $\Pi\ RLR\ (1-e^{-3}) = \Pi\ RLR$ 95%, where Π denotes ultimate earned premiums. The prior CoV of 10% covers the interval [2.5, 3.5].

- **Some degree of case overreserving**. A 0.8 prior median for RRF translates to 80% of outstanding losses becoming paid losses on average. The prior CoV of 10% covers the possibility of adequate reserving.

- **A rate of payment k_p, which is slower than the rate of earning and reporting**. The prior median of 1 and prior CoV of 10% covers a k_p between 0.85 and 1.2 with approximately 95% probability.

Note that each lognormal median is logged in the above specification, since $\exp(\mu)$ is the median of a lognormal distribution with parameters μ and σ, while σ is approximately the CoV.

5.2.2 Model 2: Lognormal distribution and additional structure

For models 2 and 3 we will assume

- a lognormal data-generating distribution with constant CoV across development and accident years;
- loss ratio dependent variables (rather than loss amounts), with incremental paid loss ratios being our principal target; and
- accident and development year varying effects for each compartmental parameter.

The implication of assuming a lognormal process distribution is that we estimate the median loss ratio development process. The number of parameters in the vector Θ will vary between models 2 and 3:

$$\begin{aligned} y_{ij} &\sim \text{Lognormal}(\log(\mu(t_j;\Theta,\delta)),\sigma_{[\delta]}^2) \\ \mu(t_j;\Theta,\delta) &= (1-\delta)OS_{ij} + \delta\left(PD_{ij} - PD_{i,j-1}\right) \\ \delta &= \begin{cases} 0 \text{ if } y \text{ is outstanding claims} \\ 1 \text{ if } y \text{ is paid claims} \end{cases} \end{aligned} \tag{8}$$

For model 2 we use similar prior assumptions to model 1, except for $\sigma_{[\delta]}$ (since we are modeling loss ratios rather than loss amounts). We set this prior to a lognormal with a median of 10%:

$$\sigma_{[\delta]} \sim \text{Lognormal}(\log(0.1),0.2)$$

Models 2 and 3 allow each compartmental model parameter to vary by both accident and development year. The approach is analogous to the “row” and “column” parameters defined in statistical models for the chain ladder, but with compartmental parameters varying rather than the expected outcome. As before, each parameter shrinks to a population estimate for sparse accident/development years:

$$\begin{aligned}
\Theta &= \mu_\Theta + u_\Theta \\
\mu_\Theta &= \left(\mu_{RLR}, \mu_{RRF}, \mu_{k_{er}}, \mu_{k_p}\right) \\
u_{\Theta_1} &= \left(u_{RLR_{[i,j]}}, u_{RRF_{[i,j]}}\right) \\
u_{\Theta_2} &= \left(u_{k_{er_{[i,j]}}}, u_{k_{p_{[i,j]}}}\right) \\
\\
u_{\Theta_1} &\sim \text{MultivariateNormal}(\mathbf{0}, \boldsymbol{\Sigma}) \\
\boldsymbol{\Sigma} &= \boldsymbol{D}\,\boldsymbol{\Omega}\,\boldsymbol{D} \\
\boldsymbol{\Omega}_{[\boldsymbol{i},\boldsymbol{j}]} &\sim \text{LKJCorr}(1) \\
\boldsymbol{D} &= \text{Diag}\left(\sigma_{RLR_{[i,j]}}, \sigma_{RRF_{[i,j]}}\right) \\
\\
u_{k_{er_{[i,j]}}} &\sim \text{Normal}\left(0, \sigma_{k_{er_{[i,j]}}}\right) \\
u_{k_{p_{[i,j]}}} &\sim \text{Normal}\left(0, \sigma_{k_{p_{[i,j]}}}\right) \\
\\
\sigma_{RLR_{[i,j]}} &\sim \text{StudentT}(10,0,0.7)^+ \\
\sigma_{RRF_{[i,j]}} &\sim \text{StudentT}(10,0,0.5)^+ \\
\sigma_{k_{er_{[i,j]}}}, \sigma_{k_{p_{[i,j]}}} &\sim \text{StudentT}(10,0,0.3)^+
\end{aligned}$$

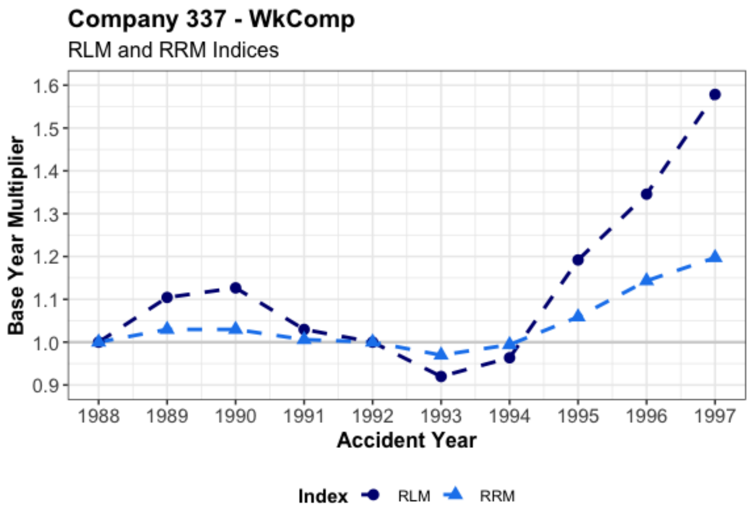
5.2.3 Model 3: Pricing and reserving cycle submodel

The final model in this case study builds pricing and reserving cycle information into the modeling process, as introduced in Section 4.4.

In lieu of market cycle information for the study, we compile an earned premium movement index and raise it to a judgmental power (0.6) to proxy a rate change index. This defines a set of reported loss ratio multipliers, RLM_i. We also select reserve robustness multipliers, RRF_i, which are set to correlate with RLM_i to reflect learnings from model 2 (shown later in this section).

These multipliers (Figure 5.3) are used to define prior values for RLR and RRF by accident year relative to the oldest year.

Figure 5.3. Proxy cycle indices



We model the extent to which RLR and RRF depend on the RLM and RRM indices with two additional parameters, $\lambda_{RLR}, \lambda_{RRF}$:

$$\begin{aligned}
RLR_{[i,j]} &= \mu_{RLR_{[i]}} + u_{RLR_{[i,j]}} \\
\mu_{RLR_{[i]}} &= \mu_{RLR_1} \cdot RLM_{[i]}^{\lambda_{RLR}} \\
\\
RRF_{[i,j]} &= \mu_{RRF_{[i]}} + u_{RRF_{[i,j]}} \\
\mu_{RRF_{[i]}} &= \mu_{RRF_1} \cdot RRM_{[i]}^{\lambda_{RRF}} \\
\\
\lambda_{RLR} &\sim \text{Normal}(1, 0.25) \\
\lambda_{RRF} &\sim \text{Normal}(1, 0.25)
\end{aligned}$$

The prior means for λ_{RLR} and λ_{RRF} are set to 1, which assumes that the expected loss ratio movements year-over-year directly correlate with the selected indices. This allows performance drift across those accident years in which market conditions are changing. However, the priors are weakly regularizing to allow inferences to pull away from our initial judgments—values less than 1, for example, would indicate a weaker correlation between the indices and loss ratio movements.

The varying effects $u_{RLR_{[i,j]}}$ and $u_{RRF_{[i,j]}}$ will override the index-driven accident year estimates if there is sufficient information in the data relative to the priors. The largest impact of the market cycle submodels should therefore be seen for less mature accident years, where we expect $u_{[i,j]}$ to shrink toward 0.

Note that in practice, we parameterize the model slightly differently to be able to estimate compartmental parameters on the standard normal scale before back-transforming them (see appendix for brms implementation).

For simplicity, we maintain the existing RLR and RRF population priors on μ_{RLR_1} and μ_{RRF_1}. All other assumptions from model 2 are carried forward.

5.3 Training

We train the models on loss and loss ratio development up to the 1996 calendar year. The review of model 1 is kept brief, with greater emphasis placed on models 2 and 3.

5.3.1 Training model 1

To review model 1 against the training data, we assess 100 outstanding and cumulative paid loss ratio posterior predictive samples by accident year and development year, shown in Figures 5.4 and 5.5.

Figure 5.4. Model 1: Outstanding loss ratio versus 100 simulations from the posterior predictive distribution

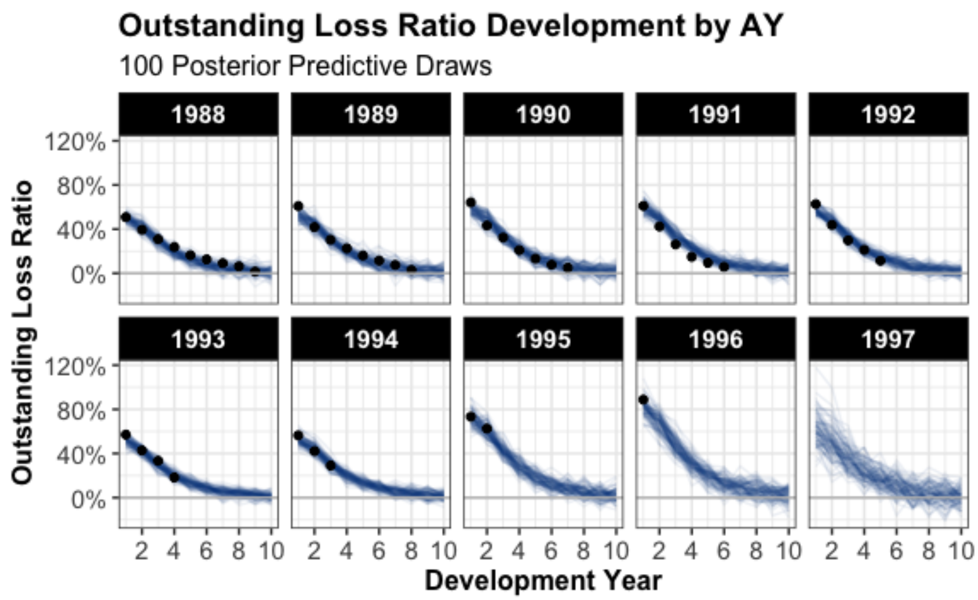
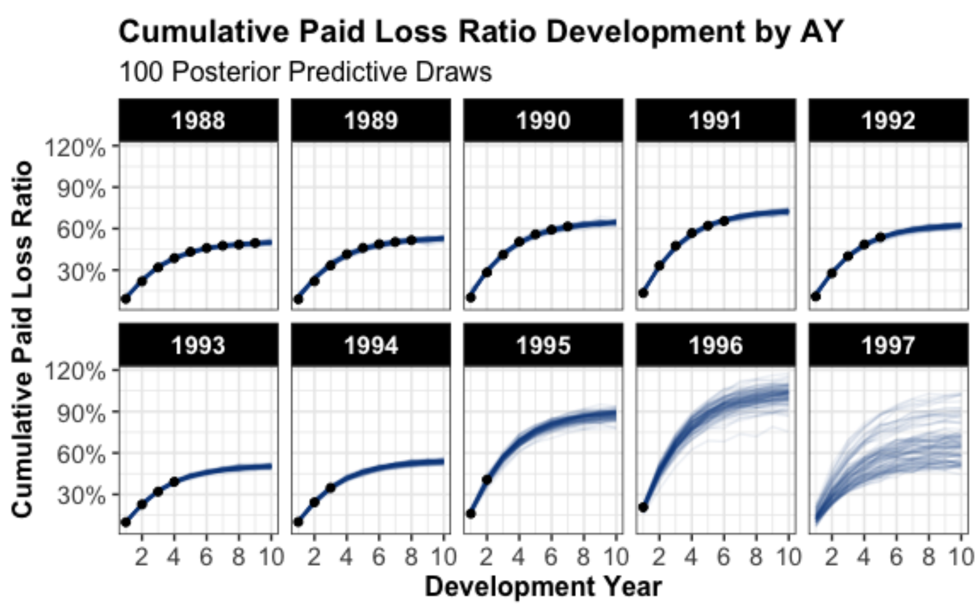


Figure 5.5. Model 1: Cumulative paid loss ratio versus 100 simulations from the posterior predictive distribution

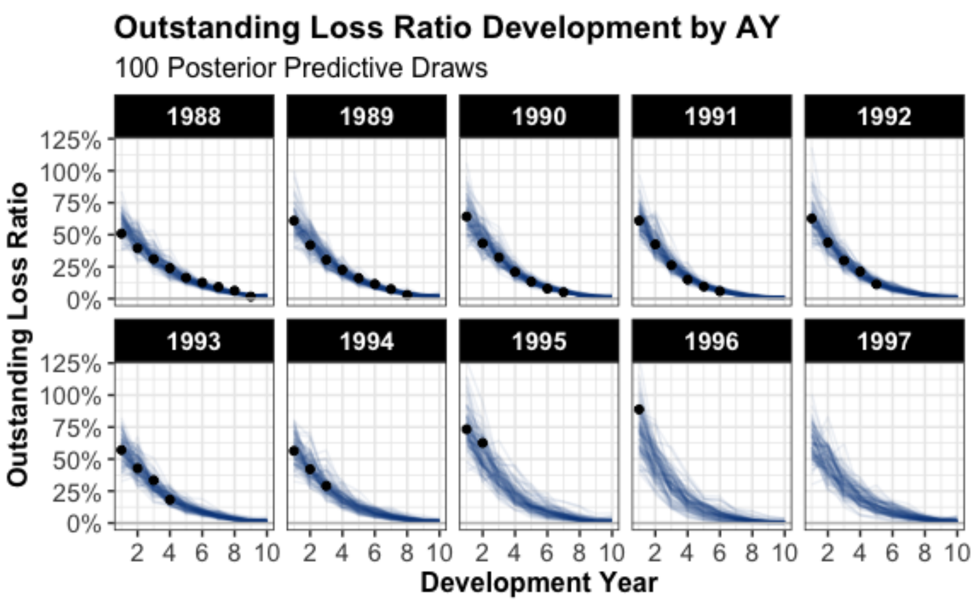


At a glance, the model appears to provide reasonable in-sample coverage of the data points themselves. However, the spaghetti plots illustrate an incompatibility of the constant-variance Gaussian process distribution with our intuition of the claims development process: in particular, we do not usually expect negative outstanding amounts or reductions in cumulative payments over time. Modeling cumulative payments with a constant process variance allows the model to generate negative posterior paid increments. Furthermore, the Gaussian assumption does not prevent negative outstanding posterior realizations.

5.3.2 Training model 2

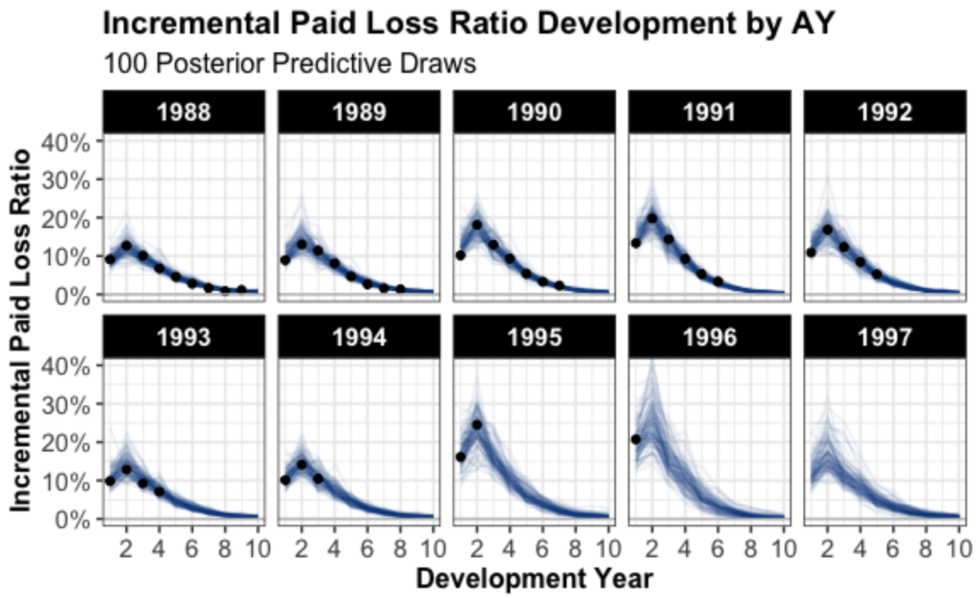
For model 2, we target outstanding and incremental paid loss ratios, and replace the Gaussian process distribution assumption with a lognormal. Each compartmental model parameter is able to vary by accident and development year.

Figure 5.6. Model 2: Outstanding loss ratio versus 100 simulations from the posterior predictive distribution



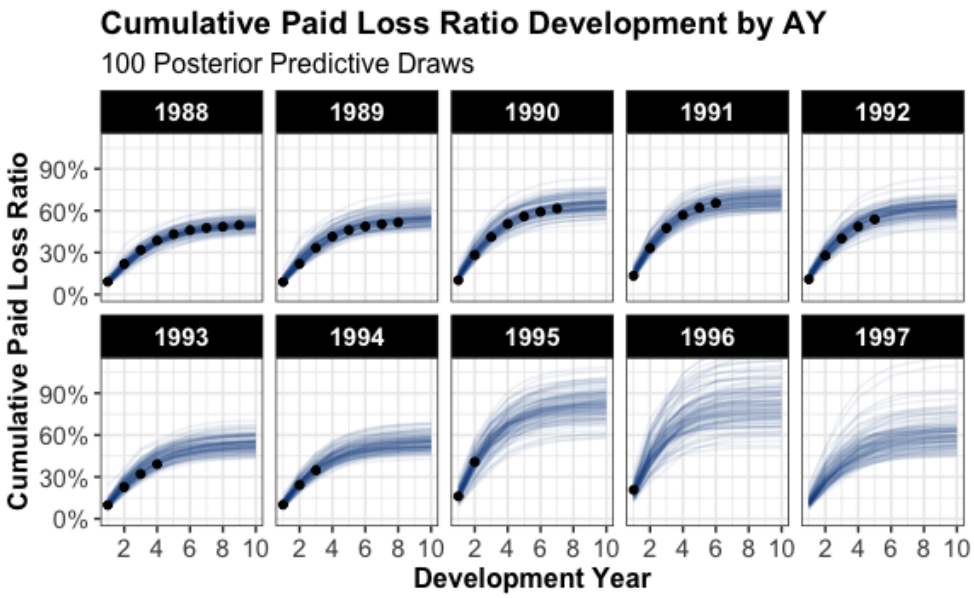
The posterior realizations for the constant CoV lognormal distribution now better reflect our understanding of typical development data: outstanding loss ratio variance shrinks by development year, and outstandings do not fall below 0, as shown in Figure 5.6.

Figure 5.7. Model 2: Incremental paid loss ratio versus 100 simulations from the posterior predictive distribution



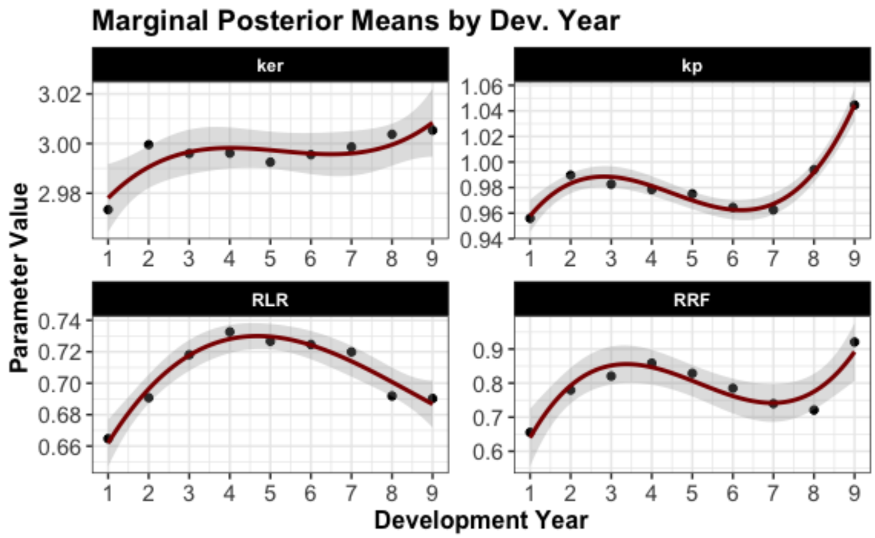
The incremental paid loss ratio samples also appear reasonable (Figure 5.7). As with the outstandings, we observe a reduction in variance over development time, together with strictly positive realizations. Consequently, when we cumulate the paid loss ratios, the process behavior aligns with expectations (Figure 5.8).

Figure 5.8. Model 2: Cumulative paid loss ratio versus 100 simulations from the posterior predictive distribution



To assess the impact of the inclusion of additional varying effects compared with model 1, we inspect marginal posterior mean parameter estimates by development year. If these have any systematic trends, we may consider incorporating them into the ODEs (analytical solutions) to carry through into extrapolation and, hence, reserve setting. Alternatively, we could model certain of the compartmental parameters to be functions of development year in our statistical model.

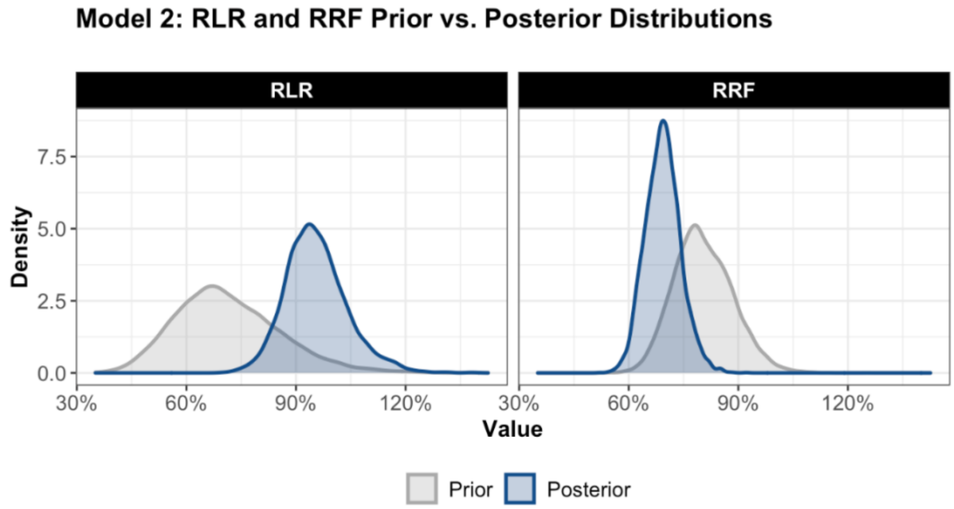
Figure 5.9. Model 3: Marginal compartmental posterior parameter estimates by development year, with 3-degree-of-freedom B-spline smoothers



The model estimates significant variation for RLR and RRF by development year, supporting the decision to allow them to vary across this dimension (see Figure 5.9). However, the trends appear somewhat cyclical, with uncertain direction beyond development year 9 in most cases. Therefore, we opt not to change the compartmental/ODE structure to account for directional trends into unseen development years.

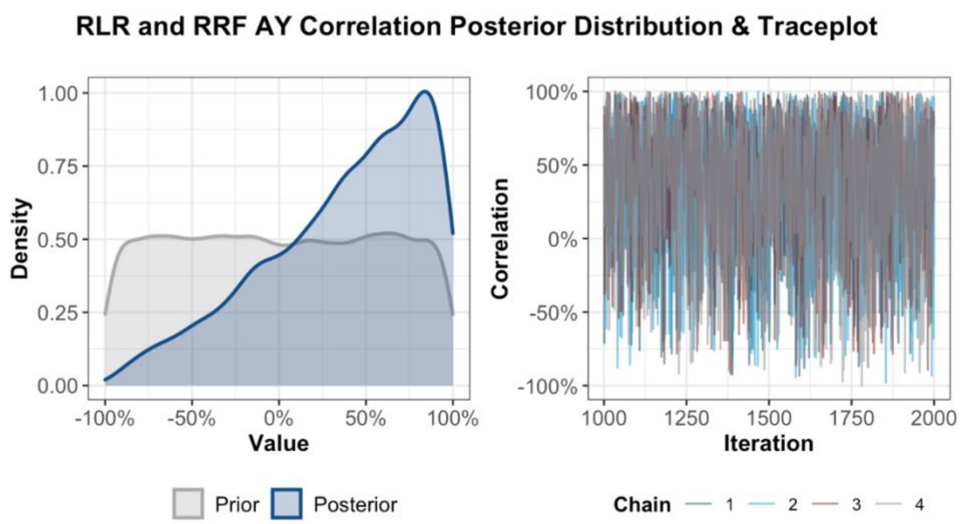
In Figures 5.10 and 5.11 we review posterior parameter distributions for RLR and RRF, in addition to the correlation between them by accident year. A traceplot is shown for the latter to diagnose convergence (inspected for all models and parameters but not shown).

Figure 5.10. Model 3: Prior versus posterior parameter densities



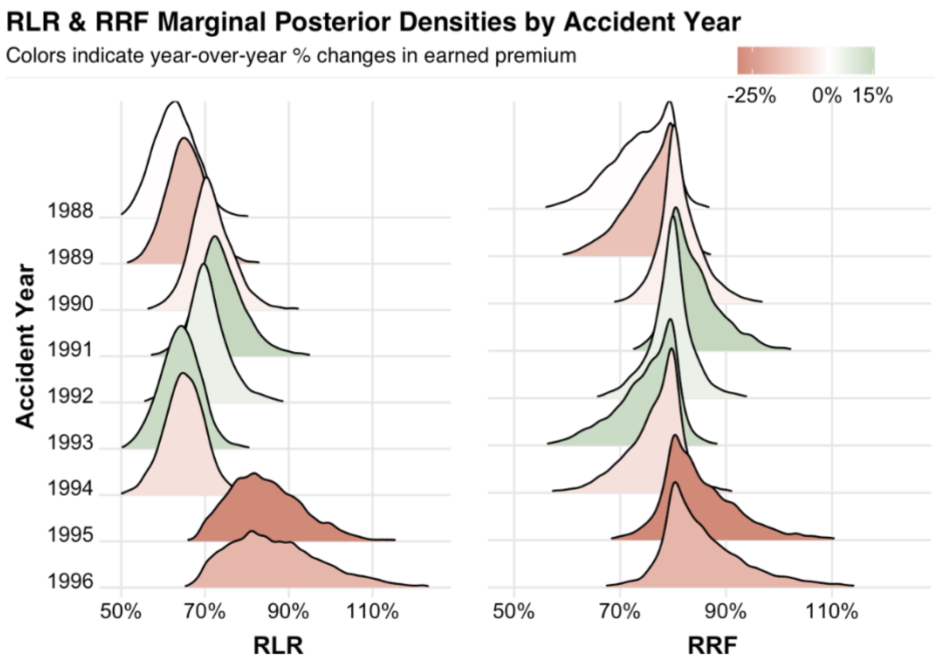
The moderate positive posterior correlation between the RLR and RRF varying effects by accident year mirrors the original compartmental reserving paper (Morris 2016), and is suggestive of a reserving cycle effect where prudent case reserves are set in a hard market and vice versa.

Figure 5.11. Model 3. Prior versus posterior parameter densities



We can see the correlation between RLR and RRF more clearly by visualizing the marginal posterior parameter distributions by accident year. Overlaying the year-over-year percentage changes in ultimate earned premiums reveals performance improvements for increases in premium and deteriorations for reductions in premium (Figure 5.12). This suggests that the movements in premium may be partially driven by rate changes.

Figure 5.12. Model 3: RLR and RRF posterior distributions by accident year



Note that this correlation breaks down between 1995 and 1996, where a premium reduction is not mirrored by a deterioration in expected performance.

With only two data points available for 1996, this could be a consequence of regularization. More specifically, we expect the model to credibility weight between the 1996 data and an all-years average. However, the prior years have been relatively favorable up until 1995, where a significant deterioration in performance is estimated.

If we intend to carry forward the prior years' correlation between premium movements and performance, then regularization back to an all-years average loss ratio is not desirable.

5.3.3 Training model 3

We train our third model on the data, which integrates market cycle indices and judgment into the model to capture trends in RLR and RRF by accident year, and proceed to review the posterior predictive checks (Figures 5.13–5.15).

Figure 5.13. Model 3: Posterior predictive checks for outstanding loss ratio development

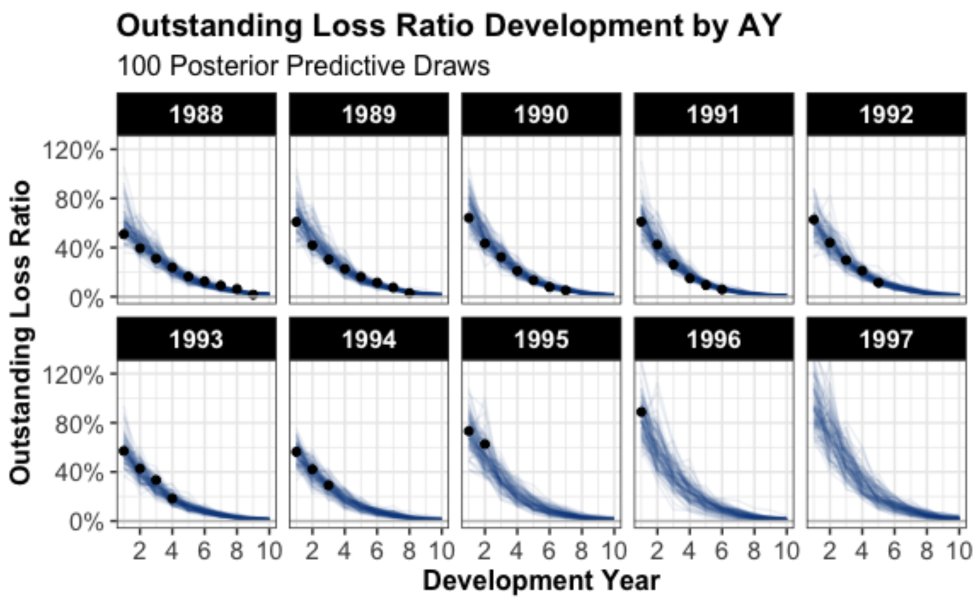


Figure 5.14. Model 3: Posterior predictive checks for incremental paid loss ratio development

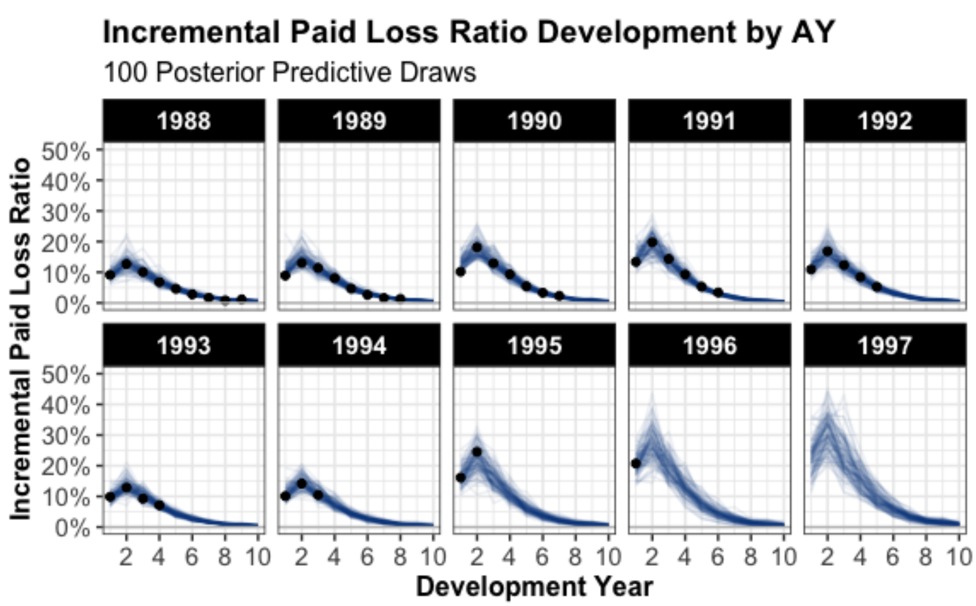
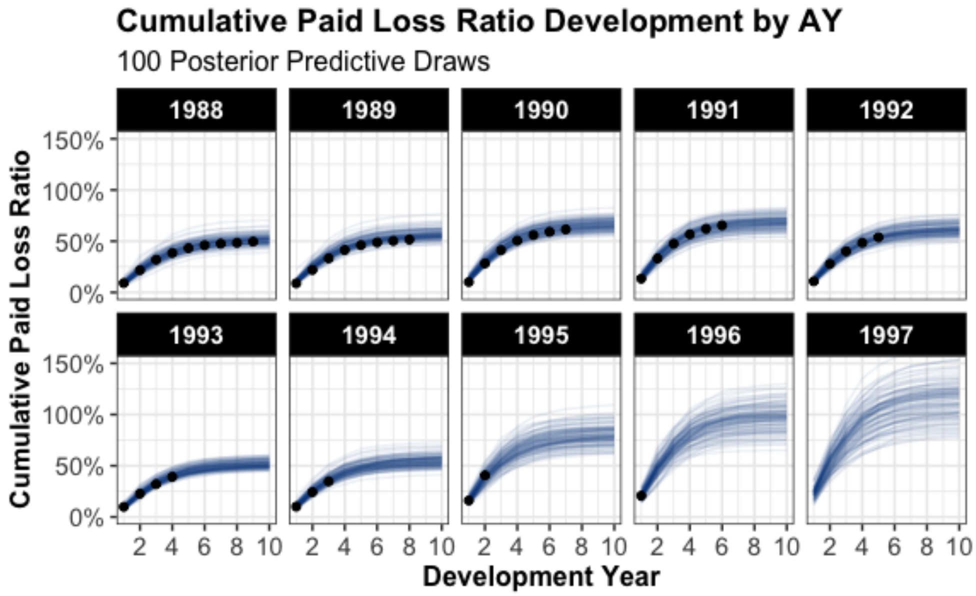
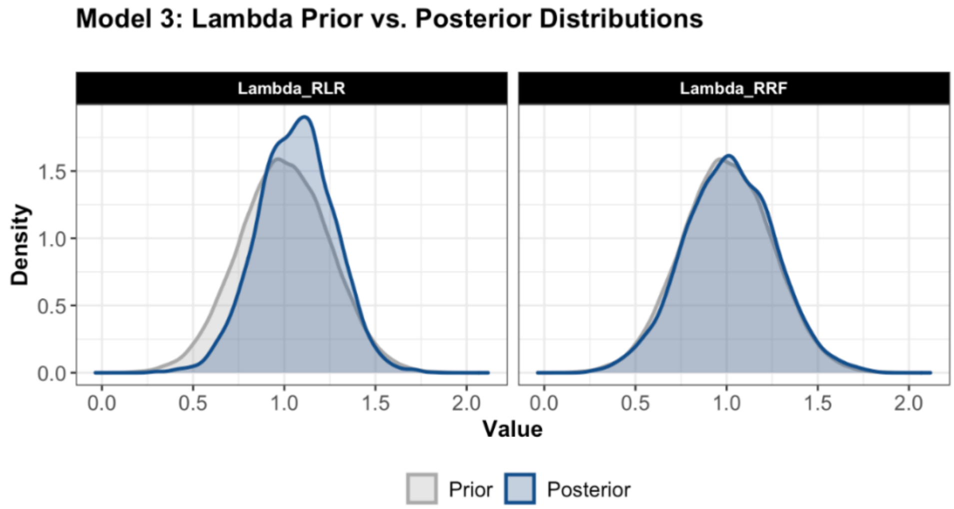


Figure 5.15. Model 3: Posterior predictive checks for cumulative paid loss ratio development



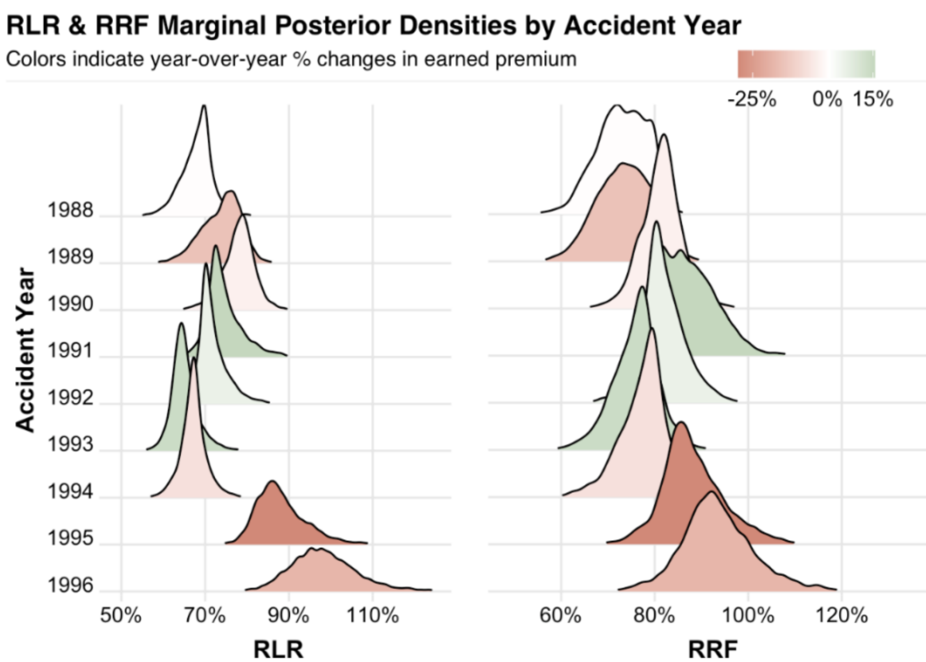
The in-sample fits once again appear reasonable, but observe that in contrast to model 2, this model projects a performance deterioration across the more recent accident years, in which premiums have been reducing. This can be attributed to the use of the RLM and RRM indices, which trend upward for more recent years. We also see that the λ_{RLR} mean posterior has increased slightly, to 1.06 from our prior of 1.00, whereas the λ_{RRF} posterior is materially unchanged from the prior (Figure 5.16).

Figure 5.16. Model 3: Prior versus posterior lambda parameter densities



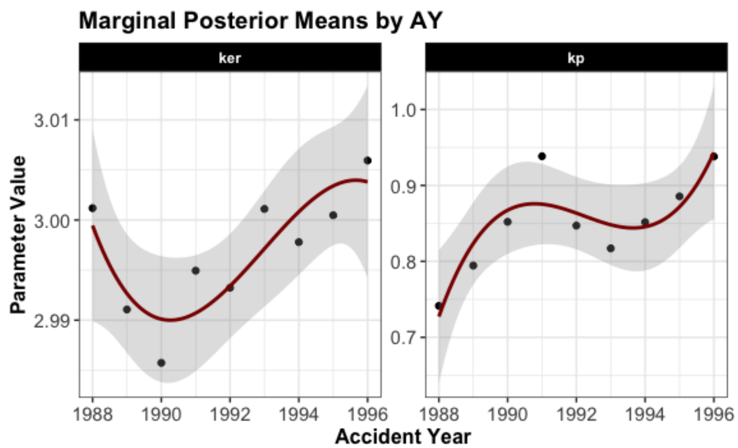
We visualize RLR and RRF posterior distributions by accident year once more (Figure 5.17) and observe a stronger correlation between their year-over-year changes and corresponding premium movements up to 1996. The model estimates that the 1996 accident year has a modest probability of inadequate case reserving (RRF > 1).

Figure 5.17. Model 3: RLR and RRF posterior distributions by accident year



Finally, we review marginal k_{er} and k_p estimates by accident year to investigate trends for additional consideration within the model structure (Figure 5.18).

Figure 5.18. Model 3: Marginal compartmental posterior parameter estimates by accident year, with 3-degree-of-freedom B-spline smoothers



Observe that k_p is estimated to trend upward by accident year, suggestive of a faster claims settlement process in more recent years. The model's accident year parameter variation takes care of this pattern, but if we expected a smoother trend we could model k_p to increase by accident year monotonically. This would be analogous to the changing settlement rate model outlined in Meyers (2015), and is left as an exercise for the reader.

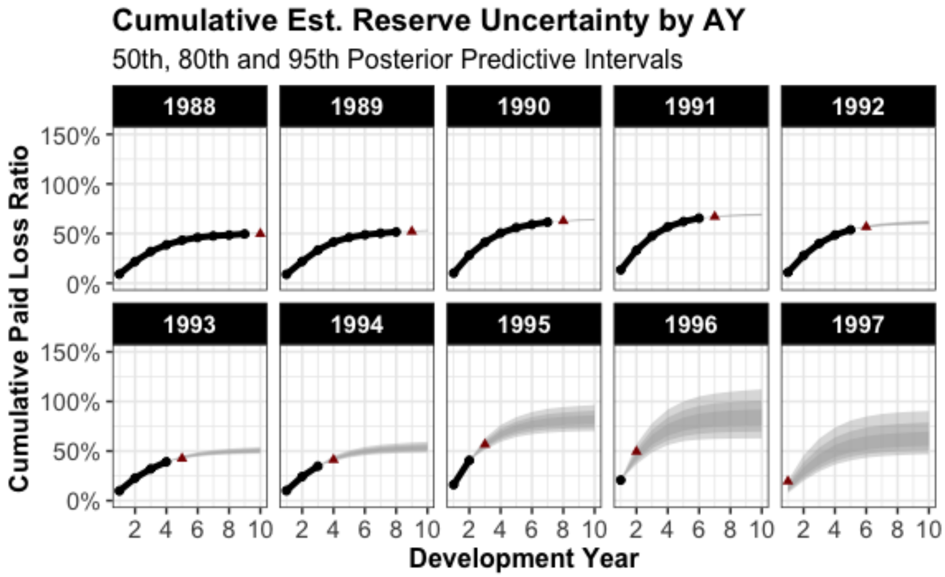
5.4 Testing and Selection

We exclude Model 1 from the selection process due to the incompatibilities of a Gaussian process distribution. Next, we predict incremental paid loss ratios using models 2 and 3 and compare these against actual loss ratios for the 1997 calendar year.

5.4.1 Testing model 2

We first inspect the model 2 future paid loss ratio development distributions by accident year, and overlay the actual one-year-ahead cumulative paid loss ratios (Figure 5.19).

Figure 5.19. Model 2: Cumulative paid loss ratio one-step-ahead holdout performance

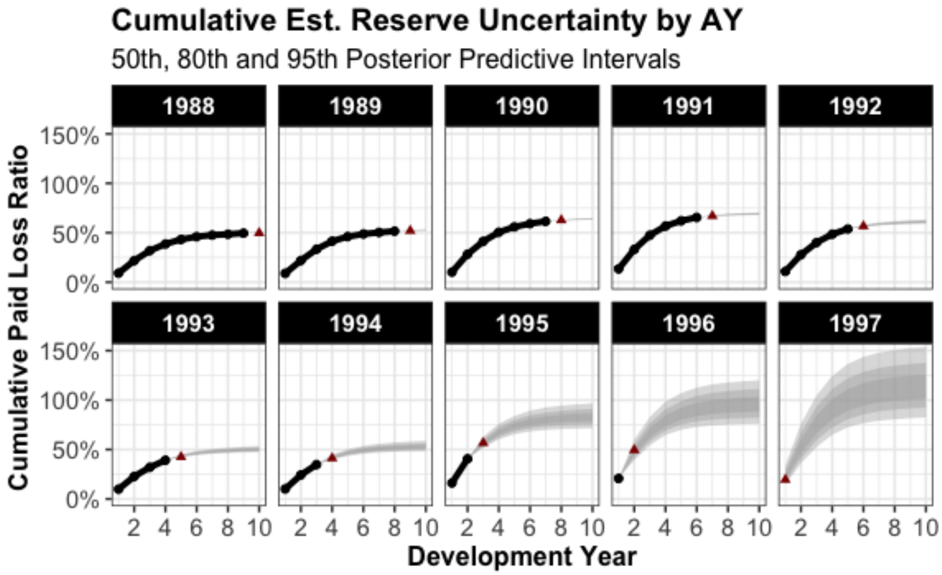


The one-step-ahead predictions are within the reserve uncertainty bands for most years; however, the model does not appear to perform as well at the mean level for 1996 and 1997. For 1997 in particular, the projections could be considered optimistic against projected 1995 and 1996 performance.

5.4.2 Testing model 3

Compared with model 2, model 3 perhaps does a better job on the one-step-ahead 1996 and 1997 accident year predictions (Figure 5.20).

Figure 5.20. Model 3: Cumulative paid loss ratio one-step-ahead holdout performance

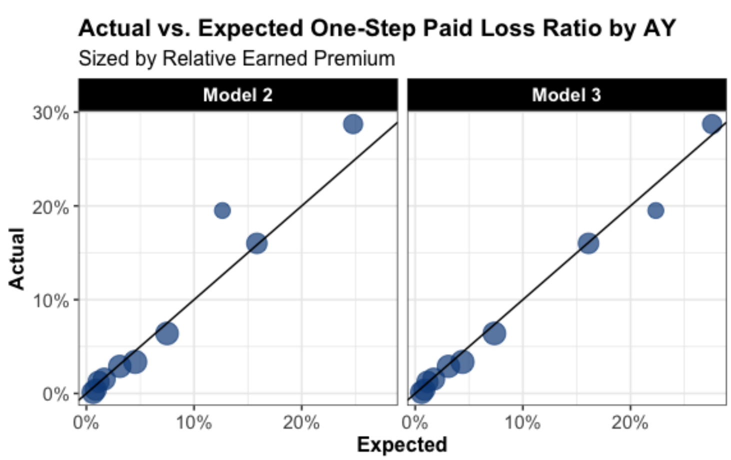


By tracking proxy market cycle information, the model is able to better account for the increasing loss ratio trend between 1994 and 1996, and into the unseen 1997 accident year.

5.4.3 Model selection

To compare each of the models in detail, one-step-ahead incremental paid loss ratio actual-versus-expected plots and root-mean-square errors (RMSEs) are reviewed in Figure 5.21 and Table 5.1, respectively.

Figure 5.21. Actual versus expected one-step-ahead paid loss ratio. Each circle represents a different accident year, with size relative to its earned premium.



5.1. One-step-ahead paid loss ratio RMSE comparisons

Model	RMSE	Change
Model 2	2.56%	-
Model 3	1.13%	-56.1%

Model 3 offers a 56% one-step-ahead RMSE improvement on model 2. The out-of-sample actual-versus-expected plots corroborate this and suggest that model 3 may be more predictive than model 2 at the mean level. We therefore select model 3 as the preferred structure.

In practice, a wide range of models could be tested, comprising different structural and prior assumptions, with a view to maximizing one-step-ahead predictive performance. This approach is not taken here primarily for brevity, but also because favorable one-step-ahead performance may not translate to favorable 10-step-ahead performance. A more robust approach would perhaps be to predict n-step-ahead performance based on fitting each model to ever-smaller triangles and optimizing the trade-off between n-step-ahead performance estimates and the quantity of data used to derive the model parameters and performance estimates.

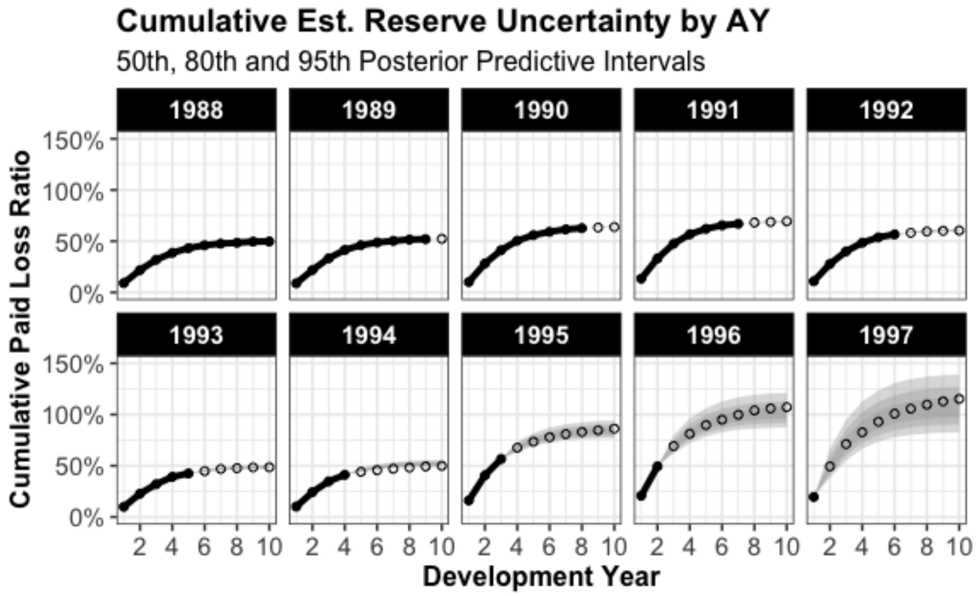
5.5 Fitting

Having selected model 3 for our reserving exercise, we fit it to the training and test data (i.e., triangles up to and including calendar year 1997) and compare actual reserves against model-estimated reserve uncertainty as a final validation step.

5.5.1 Fitting the selected model

We retrain our selected model on all information known to date (in the context of this reserving exercise), and proceed to review reserve posterior intervals for cumulative paid loss ratios.

Figure 5.22. Final model: Cumulative paid loss ratio full holdout performance

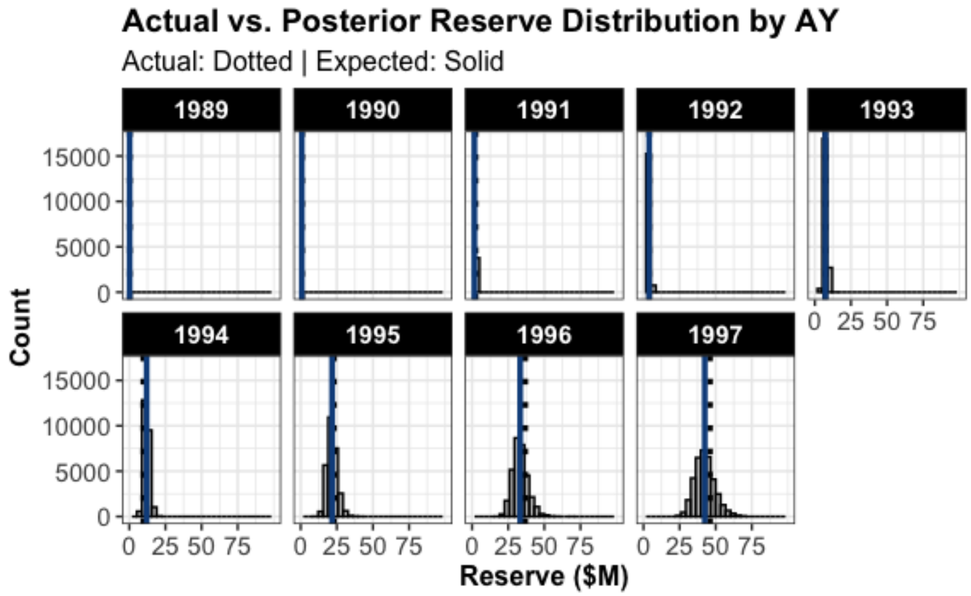


Based on Figure 5.22, the model has done a reasonable job of predicting reserve uncertainty for 1996 and prior. The 1997 year had a deteriorating development profile and longer tail relative to prior years, which the model was able to anticipate to some extent by leveraging the RLM and RRM indices.

5.6 Reserving

We take a closer look at the actual-versus-predicted reserve by accident year in Figure 5.23. The same scale is adopted for all years to highlight the insignificance of earlier-year reserves relative to the later years.

Figure 5.23. Final model: Age 10 reserve versus predicted reserve distribution by accident year

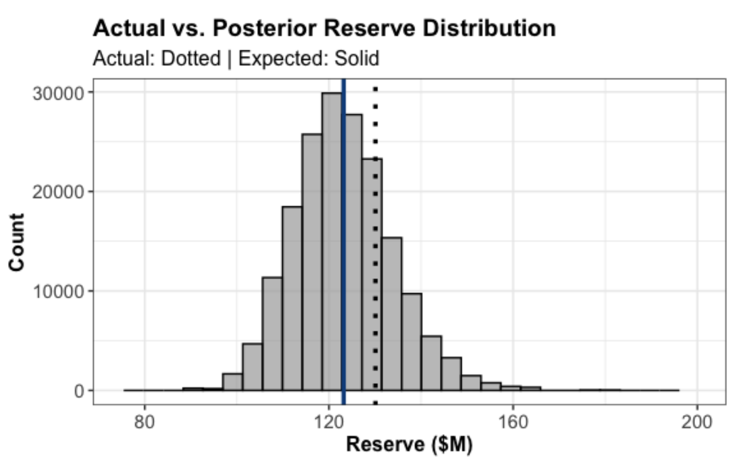


5.7 Discussion

The compartmental model predictions are reasonably accurate at the mean level, with some under- and overprediction for individual years. Across the more recent years, there is posterior mean overprediction for 1994, and underprediction for 1996 and 1997. However, the actual reserves fall within the estimated distributions.

The total estimated reserve distribution at age 10 in aggregate is depicted in Figure 5.24.

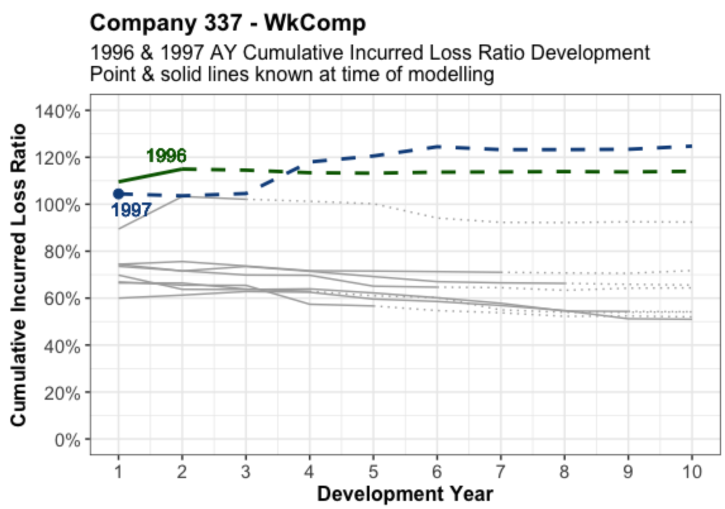
Figure 5.24. Final model: Age 10 reserve versus predicted reserve distribution



The actual reserve was $130M against an estimated mean reserve of $123M (a 5% underprediction). This is at the 75th percentile of the estimated reserve distribution.

The underprediction can be attributed to the 1996 and 1997 accident years—reviewing the upper and lower triangles, we observe that both of these years exhibited underreserving in contrast to the overreserving observed in prior years (Figure 5.25).

Figure 5.25. Accident years 1996 and 1997 incurred loss ratio development (green and blue, respectively) exhibit underreserving in contrast to prior years



Model 3 was able to forecast a deterioration through the market cycle submodel and estimated positive correlation between RLR and RRF. However, with a marked shift in performance and just two data points at the time of fitting, it is perhaps unsurprising that the model's posterior mean reserve falls short.

We conclude that the incurred data are somewhat misleading in this study due to deteriorating performance and case reserve robustness for less mature years. However, the incorporation of market cycle information (and judgment), together with a separation of portfolio performance and reserve robustness assumptions, can facilitate challenge, scenario analysis, and communication of key uncertainties during the reserving process.

6. Summary and future developments

In this paper we have presented a fully Bayesian modeling framework for the aggregated claims process to capture trends observed in paid and outstanding claims development data.

In Section 2 we outlined how to map the claims process to a system of differential equations from first principles to describe key dynamics. Using the basic building blocks of compartmental models, readers can extend and adjust the presented models to their own individual requirements.

In Section 3 we developed stochastic models for the claims process, describing the random nature of claims and latent underlying process parameters.

We showed how practitioners can utilize their expertise to describe the structure of underlying risk exposure profiles and corresponding parameter uncertainties. In addition, we highlighted the subtle but important difference between modeling incremental and cumulative claims payments.

This discussion culminated in a stochastic compartmental model, developed without reference to any particular data set, which was used to generate artificial prior predictive samples. These were used to test whether underlying model assumptions could produce data that bear a resemblance to actual observations. This is a critical aspect of the modeling process to understand model behavior. Note that the CAS Loss Simulator (CAS Loss Simulation Model Working Party 2018), based on Parodi (2014), uses similar ideas for individual claims simulation.

In Section 4, the model was further extended to allow for fixed and varying parameters across grouping dimensions. Thanks to regularization we can incorporate many modeling parameters, while at the same time mitigating the risk of overfitting. Having fitted a model, we discussed the difference between the expected loss for a given accident year (i.e., the underlying latent mean loss) and the ultimate loss (i.e., actual cumulative claim payments to date, plus the sum of future claim payments). While the expected loss provides a means for us to challenge our model and has applications in pricing, the actual reserve is the key metric for financial reporting and capital setting.

The case study in Section 5 provided a practical guide to hierarchical compartmental model building. A work flow based on training and test data sets was outlined, which included model checking and improvement, and selection criteria. We introduced the concept of parameter variation by both accident year and development year, together with a method for incorporating market cycle information and explicit judgments into the modeling process.

Code snippets were shown throughout the document to illustrate how this modeling framework can be applied in practice using the brms interface to Stan from R. The succinct model notation language used by brms allows the user to test different models and structures quickly, including across several companies and/or lines of business, with or without explicit correlations.

Those familiar with probabilistic programming languages can write hierarchical compartmental reserving models directly in Stan (Carpenter et al. 2017), PyMC (Salvatier, Wiecki, and Fonnesbeck 2016), TensorFlow Probability (Abadi et al. 2015), or other software.

Well-specified models with appropriate priors run within minutes on modern computers, and therefore hierarchical compartmental reserving models can be a part of the modern actuary's reserving toolbox. The transparency of model assumptions and ability to simulate claims process behavior provides a means of testing and challenging models in a visually intuitive manner.

Finally, as new data are collected, the Bayesian framework allows us to update our model and challenge its existing assumptions. If the posterior output changes significantly, then this should raise a call for action, either to investigate the model further or to challenge business assumptions.

6.1 Extensions

The framework and tools provided in this paper accommodate a wide range of modeling extensions, which may target ODE structures, statistical modeling assumptions, or both.

Examples of extensions that may warrant further investigation include the following:

- Double compartmental modeling of claim counts (IBNR—“incurred but not reported”) and claims severity (IBNER—“incurred but not enough reported”). An approach to developing severity using a growth curve approach is given in McNulty (2017).
- Using Gaussian processes in conjunction with compartmental models to model the stochastic loss ratio development process
- Mixture models that combine the compartmental approach with other parametric models, such as growth curves. Mixing proportions that vary by development age would provide greater flexibility to describe “nonstandard” average claims development patterns.

About the authors

Markus Gesmann is an analyst and data scientist with over 15 years' experience in the London market. He is the maintainer of the ChainLadder (Gesmann et al. 2019) and googleVis (Gesmann and de Castillo 2011) R packages. Gesmann is co-founder of the Insurance Data Science conference series and the Bayesian Mixer Meetups in London. On his blog he has published various implementations of different hierarchical loss reserving models in Stan and brms, including Jake Morris's hierarchical compartmental reserving models (Morris 2016).

Jake Morris is an actuarial data scientist with 10 years' experience in predictive modeling and commercial insurance analytics. He has presented on Bayesian and hierarchical techniques at various international actuarial and data science conferences, and he is the author of "Hierarchical Compartmental Models for Loss Reserving" (2016). Morris is a Fellow of the Institute and Faculty of Actuaries (FIA) and a Certified Specialist in Predictive Analytics (CSPA).

Acknowledgments

The authors would like to thank the following individuals for taking the time to review this work and provide valuable feedback: Michael Betancourt, Paul-Christian Bürkner, Dave Clark, Mick Cooney, Adam M. Gerdes, Michael Henk, Robert Hooley, Dan Murphy, Roland Ramsahai, and Mario Wüthrich.

They would also like to thank the Casualty Actuarial Society for sponsoring the research and providing periodic feedback throughout the process.

7. Appendix

The appendix presents the R code used in Sections 4 and 5. The code can be copied and pasted into an R session. At the time of writing, R version 3.6.1 (2019-07-05) was used, with brms 2.10.0 and RStan 2.19.2.

```
library(rstan)
library(brms)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

7.1 R code from Section 4

The GenIns triangle is part of the ChainLadder package. The triangle is transformed into a long table format, with premiums, incremental paid, and loss ratio columns added:

```
library(ChainLadder)
library(data.table)
data(GenIns)
lossDat <- data.table(
  AY = rep(1991:2000, 10),
  dev = sort(rep(1:10, 10)),
  premium = rep(10000000+400000*0:9, 10),
  cum_loss = as.vector(GenIns),
  incr_loss = as.vector(cum2incr(GenIns))
)[order(AY, dev),
  `:=`(cum_lr = cum_loss/premium,
       incr_lr = incr_loss/premium)]
```

The next code chunk shows how the loss emergence pattern is modeled using differential equations in Stan. The Stan code is stored as a character string in R, and later passed on into brm.

```
myFuns <- "
real[] ode_lossemergence(real t, real [] y,  real [] theta,
                         real [] x_r, int[] x_i){
  real dydt[3];
  real ke = theta[1];
  real dr = theta[2];
  real kp1 = theta[3];
  real kp2 = theta[4];

  dydt[1] = pow(ke, dr) * pow(t, dr - 1) * exp(-t * ke)/tgamma(dr)
          - (kp1 + kp2) * y[1];
  dydt[2] = kp2 * (y[1] - y[2]);
  dydt[3] = (kp1 *  y[1] + kp2 * y[2]);

  return dydt;
}
real int_lossemergence(real t, real ke, real dr,
                        real kp1, real kp2){
  real y0[3]; real y[1, 3]; real theta[4];

  y0[1] = 0; y0[2] = 0; y0[3] = 0;

  theta[1] = ke;
  theta[2] = dr;
  theta[3] = kp1;
  theta[4] = kp2;

  y = integrate_ode_rk45(ode_lossemergence,
```

```
                    y0, 0, rep_array(t, 1), theta,
                    rep_array(0.0, 0), rep_array(1, 1),
                    0.0001, 0.0001, 500); // tolerances, steps
  return (y[1, 3]);
}
real lossemergence(real t, real devfreq, real ke, real dr,
                real kp1, real kp2){
   real out = int_lossemergence(t, ke, dr, kp1, kp2);
   if(t > devfreq){ // paid greater dev period 1
   // incremental paid
    out = out - int_lossemergence(t - devfreq, ke, dr, kp1, kp2);
   }
   return(out);
}
"
```

The following code defines the hierarchical structure using the formula interface in brms:

```
frml <- bf(incr_lr ~ eta,
     nlf(eta ~ log(ELR * lossemergence(dev, 1.0, ke, dr, kp1, kp2))),
     nlf(ke ~ exp(oke * 0.5)),
     nlf(dr ~ 1 + 0.1 * exp(odr * 0.5)),
     nlf(kp1 ~ 0.5 * exp(okp1 * 0.5)),
     nlf(kp2 ~ 0.1 * exp(okp2 * 0.5)),
     ELR ~ 1 + (1 | AY),
     oke  ~ 1 + (1 | AY), odr ~ 1 + (1 | AY),
     okp1 ~ 1 + (1 | AY), okp2 ~ 1 + (1 | AY),
     nl = TRUE)
```

7.1.1 Multilevel effects with narrow priors

Model run with narrow priors for the multilevel effects:

```
mypriors <- c(prior(inv_gamma(4, 2), nlpar = "ELR", lb=0),
        prior(normal(0, 1), nlpar = "oke"),
        prior(normal(0, 1), nlpar = "odr"),
        prior(normal(0, 1), nlpar = "okp1"),
        prior(normal(0, 1), nlpar = "okp2"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "ELR"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "oke"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "odr"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "okp1"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "okp2"),
        prior(student_t(10, 0, 1), class = "sigma"))

fit_loss <- brm(frml, prior = mypriors,
         data = lossDat, family = lognormal(), seed = 12345,
         stanvars = stanvar(scode = myFuns, block = "functions"),
         file="models/section_4/GenInsIncModelLog")

fit_loss
#>  Family: lognormal
#>   Links: mu = identity; sigma = identity
#> Formula: incr_lr ~ eta
#>          eta ~ log(ELR * lossemergence(dev, 1, ke, dr, kp1, kp2))
#>          ke ~ exp(oke * 0.5)
#>          dr ~ 1 + 0.1 * exp(odr * 0.5)
#>          kp1 ~ 0.5 * exp(okp1 * 0.5)
#>          kp2 ~ 0.1 * exp(okp2 * 0.5)
#>          ELR ~ 1 + (1 | AY)
```

```
#>         oke ~ 1 + (1 | AY)
#>         odr ~ 1 + (1 | AY)
#>         okp1 ~ 1 + (1 | AY)
#>         okp2 ~ 1 + (1 | AY)
#>    Data: lossDat (Number of observations: 55)
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>          total post-warmup samples = 4000
#>
#> Group-Level Effects:
#> ~AY (Number of levels: 10)
#>                    Estimate Est.Error l-95% CI
#> sd(ELR_Intercept)      0.04      0.03     0.00
#> sd(oke_Intercept)      0.08      0.06     0.00
#> sd(odr_Intercept)      0.09      0.07     0.00
#> sd(okp1_Intercept)     0.08      0.06     0.00
#> sd(okp2_Intercept)     0.09      0.07     0.00
#>                    u-95% CI Rhat Bulk_ESS Tail_ESS
#> sd(ELR_Intercept)      0.11 1.00     1784     1690
#> sd(oke_Intercept)      0.24 1.00     3352     2226
#> sd(odr_Intercept)      0.26 1.00     4252     2019
#> sd(okp1_Intercept)     0.23 1.00     3068     2082
#> sd(okp2_Intercept)     0.26 1.00     3572     1614
#>
#> Population-Level Effects:
#>               Estimate Est.Error l-95% CI u-95% CI
#> ELR_Intercept     0.49      0.03     0.43     0.56
#> oke_Intercept    -0.90      0.54    -1.92     0.13
#> odr_Intercept     0.49      1.04    -1.62     2.47
```

```
#> okp1_Intercept    -0.40      0.58    -1.40     0.82
#> okp2_Intercept     0.01      0.97    -1.87     2.00
#>                Rhat Bulk_ESS Tail_ESS
#> ELR_Intercept  1.00     4459     3290
#> oke_Intercept  1.00     3769     2887
#> odr_Intercept  1.00     6278     3278
#> okp1_Intercept 1.00     3746     2407
#> okp2_Intercept 1.00     5210     3193
#>
#> Family Specific Parameters:
#>       Estimate Est.Error l-95% CI u-95% CI Rhat
#> sigma     0.37      0.04     0.30     0.45 1.00
#>       Bulk_ESS Tail_ESS
#> sigma     6018     2946
#>
#> Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
#> is a crude measure of effective sample size, and Rhat is the potential
#> scale reduction factor on split chains (at convergence, Rhat = 1).
```

Population-level posterior parameters on original scale:

```
x <- posterior_samples(fit_loss, "^b")
mySummary <- function(x){
  c(Estimate = mean(x), Est.Error = sd(x),
    `l-95% CI` = as.numeric(quantile(x, probs = 0.025)),
    `u-95% CI` = as.numeric(quantile(x, probs = 0.975)))
}
rbind(
  ELR = mySummary(x[, 'b_ELR_Intercept']),
```

```
 ke = mySummary(exp(x[, 'b_oke_Intercept'] * 0.5)),
 dr = mySummary(1 + 0.1 * exp(x[, 'b_odr_Intercept'] * 0.5)),
 kp1 = mySummary(0.5 * exp(x[, 'b_okp1_Intercept'] * 0.5)),
 kp2 = mySummary(0.1 * exp(x[, 'b_okp2_Intercept'] * 0.5))
 )
#>     Estimate Est.Error l-95% CI u-95% CI
#> ELR   0.4912   0.03330  0.42824   0.5595
#> ke    0.6595   0.17935  0.38199   1.0696
#> dr    1.1454   0.07655  1.04457   1.3441
#> kp1   0.4281   0.13434  0.24851   0.7521
#> kp2   0.1134   0.05934  0.03931   0.2723
```

7.1.2 Multilevel effects with wider priors

Model run with wider priors for the multilevel effects:

```
mypriors2 <- c(prior(inv_gamma(4, 2), nlpar = "ELR", lb=0),
        prior(normal(0, 1), nlpar = "oke"),
        prior(normal(0, 1), nlpar = "odr"),
        prior(normal(0, 1), nlpar = "okp1"),
        prior(normal(0, 1), nlpar = "okp2"),
        prior(student_t(10, 0, 1), class = "sd", nlpar = "ELR"),
        prior(student_t(10, 0, 1), class = "sd", nlpar = "oke"),
        prior(student_t(10, 0, 1), class = "sd", nlpar = "odr"),
        prior(student_t(10, 0, 1), class = "sd", nlpar = "okp1"),
        prior(student_t(10, 0, 1), class = "sd", nlpar = "okp2"),
        prior(student_t(10, 0, 1), class = "sigma"))

fit_loss2 <- brm(frml, prior = mypriors2,
         data = lossDat, family = lognormal(), seed = 12345,
         control = list(adapt_delta = 0.9, max_treedepth=15),
```

```
          stanvars = stanvar(scode = myFuns, block = "functions"),
          file="models/section_4/GenInsIncModelLog2")

fit_loss2
#>  Family: lognormal
#>   Links: mu = identity; sigma = identity
#> Formula: incr_lr ~ eta
#>          eta ~ log(ELR * lossemergence(dev, 1, ke, dr, kp1, kp2))
#>          ke ~ exp(oke * 0.5)
#>          dr ~ 1 + 0.1 * exp(odr * 0.5)
#>          kp1 ~ 0.5 * exp(okp1 * 0.5)
#>          kp2 ~ 0.1 * exp(okp2 * 0.5)
#>          ELR ~ 1 + (1 | AY)
#>          oke ~ 1 + (1 | AY)
#>          odr ~ 1 + (1 | AY)
#>          okp1 ~ 1 + (1 | AY)
#>          okp2 ~ 1 + (1 | AY)
#>    Data: lossDat (Number of observations: 55)
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>          total post-warmup samples = 4000
#>
#> Group-Level Effects:
#> ~AY (Number of levels: 10)
#>                    Estimate Est.Error l-95% CI
#> sd(ELR_Intercept)      0.05      0.04     0.00
#> sd(oke_Intercept)      0.24      0.20     0.01
#> sd(odr_Intercept)      0.67      0.51     0.03
#> sd(okp1_Intercept)     0.24      0.20     0.01
```

```
#> sd(okp2_Intercept)     0.81      0.61     0.04
#>                    u-95% CI Rhat Bulk_ESS Tail_ESS
#> sd(ELR_Intercept)      0.13 1.00     1764     1934
#> sd(oke_Intercept)      0.73 1.00     2545     1999
#> sd(odr_Intercept)      1.93 1.00     2642     1788
#> sd(okp1_Intercept)     0.75 1.00     2603     2620
#> sd(okp2_Intercept)     2.26 1.00     3638     2070
#>
#> Population-Level Effects:
#>                Estimate Est.Error l-95% CI u-95% CI
#> ELR_Intercept      0.49      0.04     0.42     0.57
#> oke_Intercept     -0.88      0.56    -1.94     0.18
#> odr_Intercept      0.29      1.00    -1.70     2.20
#> okp1_Intercept    -0.44      0.60    -1.50     0.84
#> okp2_Intercept     0.00      0.98    -1.92     2.00
#>                Rhat Bulk_ESS Tail_ESS
#> ELR_Intercept  1.00     3294     2689
#> oke_Intercept  1.00     2795     2949
#> odr_Intercept  1.00     5223     2804
#> okp1_Intercept 1.00     2771     3067
#> okp2_Intercept 1.00     3941     2730
#>
#> Family Specific Parameters:
#>       Estimate Est.Error l-95% CI u-95% CI Rhat
#> sigma     0.37      0.04     0.30     0.45 1.00
#>       Bulk_ESS Tail_ESS
#> sigma     5560     2997
#>
```

```
#> Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
#> is a crude measure of effective sample size, and Rhat is the potential
#> scale reduction factor on split chains (at convergence, Rhat = 1).
```

Population-level posterior parameters on original scale:

```
x <- posterior_samples(fit_loss2, "^b")
rbind(
 ELR = mySummary(x[, 'b_ELR_Intercept']),
 ke = mySummary(exp(x[, 'b_oke_Intercept'] * 0.5)),
 dr = mySummary(1 + 0.1 * exp(x[, 'b_odr_Intercept'] * 0.5)),
 kp1 = mySummary(0.5 * exp(x[, 'b_okp1_Intercept'] * 0.5)),
 kp2 = mySummary(0.1 * exp(x[, 'b_okp2_Intercept'] * 0.5))
 )
#>     Estimate Est.Error l-95% CI u-95% CI
#> ELR   0.4921   0.03795  0.42340   0.5741
#> ke    0.6698   0.19095  0.37871   1.0928
#> dr    1.1310   0.06812  1.04282   1.3004
#> kp1   0.4213   0.13688  0.23628   0.7600
#> kp2   0.1130   0.05958  0.03821   0.2717
```

Note that in order to predict the models, the user-defined Stan functions have to be exported to R via the following:

```
expose_functions(fit_loss, vectorize = TRUE)
```

7.2 R code from case study in Section 5

7.2.1 Data

The data used for the case study is a subset of the wkcomp data set from the raw R package (Fannin 2018):

```
library(raw)
data(wkcomp)
library(data.table)
library(tidyverse)
# Convert to tibble, rename cols, add calendar year and loss ratio columns
wkcomp <- wkcomp %>%
 as_tibble() %>%
 rename(accident_year = AccidentYear, dev_year = Lag,
       entity_id = GroupCode) %>%
 mutate(cal_year = accident_year + dev_year - 1,
       paid_loss_ratio = CumulativePaid/DirectEP,
       os_loss_ratio = (CumulativeIncurred - CumulativePaid)/DirectEP)

# Add incremental paid loss ratio column
wkcomp <- wkcomp %>%
 group_by(entity_id, accident_year) %>%
 arrange(dev_year) %>%
 mutate(incr_paid_loss_ratio = paid_loss_ratio -
         shift(paid_loss_ratio, n=1, fill=0,
              type="lag")) %>%
 ungroup() %>%
 arrange(entity_id, accident_year, dev_year)

# Stack paid and os into one column + define train and test
wkcomp2 <- wkcomp %>%
 transmute(
  entity_id, accident_year, dev_year, cal_year,
  premium = DirectEP, delta = 1, deltaf = "paid",
```

```
  loss_ratio_train = ifelse(cal_year < max(accident_year),
                            incr_paid_loss_ratio,
                            NA),
  loss_ratio_test = ifelse(cal_year >= max(accident_year),
                           incr_paid_loss_ratio,
                           NA),
  loss_amount_train = ifelse(cal_year < max(accident_year),
                             CumulativePaid,
                             NA),
  loss_amount_test = ifelse(cal_year >= max(accident_year),
                            CumulativePaid,
                            NA)
) %>%
bind_rows(
  wkcomp %>%
    transmute(
      entity_id, accident_year, dev_year, cal_year,
      premium = DirectEP, delta = 0, deltaf = "os",
      loss_ratio_train = ifelse(cal_year < max(accident_year),
                                os_loss_ratio,
                                NA),
      loss_ratio_test = ifelse(cal_year >= max(accident_year),
                               os_loss_ratio,
                               NA),
      loss_amount_train = ifelse(cal_year < max(accident_year),
                                 CumulativeIncurred - CumulativePaid,
                                 NA),
      loss_amount_test = ifelse(cal_year >= max(accident_year),
```

```
                    CumulativeIncurred - CumulativePaid,
                    NA)
   )
 )
```

Filter for company “337”:

```
dat337 <- wkcomp2 %>% filter(entity_id ==337)
```

7.2.2 Model 1

```
myFunsCumPaid <- "
real paid(real t, real ker, real kp, real RLR, real RRF){
 return(
  RLR*RRF/(ker - kp) * (ker *(1 - exp(-kp*t)) -
  kp*(1 - exp(-ker*t)))
 );
}
real os(real t, real ker, real kp, real RLR){
 return(
  (RLR*ker/(ker - kp) * (exp(-kp*t) - exp(-ker*t)))
 );
}
real claimsprocess(real t, real ker, real kp,
           real RLR, real RRF, real delta){
   real out;
   out = os(t, ker, kp, RLR) * (1 - delta) +
      paid(t, ker, kp, RLR, RRF) * delta;

   return(out);
```

```
}
"

  frml1 <- bf(loss_amount_train ~ premium * claimsprocess(dev_year, ker, kp,
                                            RLR, RRF, delta),
      nlf(ker ~ 3 * exp(oker * 0.1)),
      nlf(kp ~ 1 * exp(okp * 0.1)),
      nlf(RLR ~ 0.7 * exp(oRLR * 0.2)),
      nlf(RRF ~ 0.8 * exp(oRRF * 0.1)),
      oRLR ~ 1 + (1 | ID | accident_year),
      oRRF ~ 1 + (1 | ID | accident_year),
      oker ~ 1,  okp ~ 1,
      sigma ~ 0 + deltaf,
      nl = TRUE)

  mypriors1 <- c(prior(normal(0, 1), nlpar = "oRLR"),
        prior(normal(0, 1), nlpar = "oRRF"),
        prior(normal(0, 1), nlpar = "oker"),
        prior(normal(0, 1), nlpar = "okp"),
        prior(student_t(1, 0, 1000), class = "b",
             coef="deltafpaid", dpar= "sigma"),
        prior(student_t(1, 0, 1000), class = "b",
             coef="deltafos", dpar= "sigma"),
        prior(student_t(10, 0, 0.2), class = "sd", nlpar = "oRLR"),
        prior(student_t(10, 0, 0.1), class = "sd", nlpar = "oRRF"),
        prior(lkj(1), class="cor"))

  m1fit <- brm(frml1, data = dat337[!is.na(loss_ratio_train)],
       family = gaussian(),
```

```
        prior = mypriors1,
        stanvars = stanvar(scode = myFunsCumPaid, block = "functions"),
        control = list(adapt_delta = 0.99, max_treedepth=15),
        file="models/section_5/CaseStudy_Model_1",
        seed = 123, iter = 2000, chains = 4)

    m1fit
#>  Family: gaussian
#>   Links: mu = identity; sigma = log
#> Formula: loss_amount_train ~ premium * claimsprocess(dev_year, ker, kp, RLR, RRF, delta)
#>          ker ~ 3 * exp(oker * 0.1)
#>          kp ~ 1 * exp(okp * 0.1)
#>          RLR ~ 0.7 * exp(oRLR * 0.2)
#>          RRF ~ 0.8 * exp(oRRF * 0.1)
#>          oRLR ~ 1 + (1 | ID | accident_year)
#>          oRRF ~ 1 + (1 | ID | accident_year)
#>          oker ~ 1
#>          okp ~ 1
#>          sigma ~ 0 + deltaf
#>    Data: dat337[!is.na(loss_ratio_train)] (Number of observations: 90)
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>          total post-warmup samples = 4000
#>
#> Group-Level Effects:
#> ~accident_year (Number of levels: 9)
#>                          Estimate Est.Error
#> sd(oRLR_Intercept)           0.64      0.15
#> sd(oRRF_Intercept)           0.74      0.22
```

```
#> cor(oRLR_Intercept,oRRF_Intercept)     0.54      0.26
#>                               l-95% CI u-95% CI
#> sd(oRLR_Intercept)                 0.40     1.00
#> sd(oRRF_Intercept)                 0.33     1.22
#> cor(oRLR_Intercept,oRRF_Intercept)    -0.01     0.97
#>                               Rhat Bulk_ESS
#> sd(oRLR_Intercept)              1.00     1779
#> sd(oRRF_Intercept)              1.00     1763
#> cor(oRLR_Intercept,oRRF_Intercept) 1.00     1300
#>                               Tail_ESS
#> sd(oRLR_Intercept)                 2221
#> sd(oRRF_Intercept)                 1536
#> cor(oRLR_Intercept,oRRF_Intercept)     1532
#>
#> Population-Level Effects:
#>                 Estimate Est.Error l-95% CI u-95% CI
#> oRLR_Intercept       1.44      0.26     0.92     1.95
#> oRRF_Intercept      -1.36      0.42    -2.17    -0.50
#> oker_Intercept      -5.40      0.81    -6.87    -3.71
#> okp_Intercept       -8.49      0.36    -9.17    -7.79
#> sigma_deltafos       8.25      0.16     7.96     8.57
#> sigma_deltafpaid     6.66      0.20     6.31     7.08
#>                 Rhat Bulk_ESS Tail_ESS
#> oRLR_Intercept   1.00     1137     2014
#> oRRF_Intercept   1.00     2518     2684
#> oker_Intercept   1.00     1821     2368
#> okp_Intercept    1.00     2491     2826
#> sigma_deltafos   1.00     1485     2306
```

```
#> sigma_deltafpaid 1.00     1825     2109
#>
#> Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
#> is a crude measure of effective sample size, and Rhat is the potential
#> scale reduction factor on split chains (at convergence, Rhat = 1).
```

7.2.3 Model 2

```
myFunsIncrPaid <- "
real paid(real t, real ker, real kp, real RLR, real RRF){
 return(
  RLR*RRF/(ker - kp) * (ker *(1 - exp(-kp*t)) -
  kp*(1 - exp(-ker*t)))
 );
}
real os(real t, real ker, real kp, real RLR){
 return(
  (RLR*ker/(ker - kp) * (exp(-kp*t) - exp(-ker*t)))
 );
}
real claimsprocess(real t, real devfreq, real ker, real kp,
                   real RLR, real RRF, real delta){
   real out;
   out = os(t, ker, kp, RLR) * (1 - delta) +
         paid(t, ker, kp, RLR, RRF) * delta;

   if( (delta > 0) && (t > devfreq) ){ // paid greater dev period 1
   // incremental paid
    out = out - paid(t - devfreq, ker, kp, RLR, RRF)*delta;
```

```
  }
  return(out);
}
"

  frml2 <- bf(loss_ratio_train ~ eta,
        nlf(eta ~ log(claimsprocess(dev_year, 1.0, ker, kp,
                                    RLR, RRF, delta))),
        nlf(ker ~ 3 * exp(oker * 0.1)),
        nlf(kp ~ 1 * exp(okp * 0.1)),
        nlf(RLR ~ 0.7 * exp(oRLR * 0.2)),
        nlf(RRF ~ 0.8 * exp(oRRF * 0.1)),
        oRLR ~ 1 + (1 | ID | accident_year) + (1 | dev_year),
        oRRF ~ 1 + (1 | ID | accident_year) + (1 | dev_year),
        oker ~ 1  + (1 | accident_year) + (1 | dev_year),
        okp ~ 1 + (1 | accident_year) + (1 | dev_year),
        sigma ~ 0 + deltaf, nl = TRUE)

  mypriors2 <- c(prior(normal(0, 1), nlpar = "oRLR"),
          prior(normal(0, 1), nlpar = "oRRF"),
          prior(normal(0, 1), nlpar = "oker"),
          prior(normal(0, 1), nlpar = "okp"),
          prior(normal(log(0.2), 0.2), class = "b",
             coef="deltafpaid", dpar= "sigma"),
          prior(normal(log(0.2), 0.2), class = "b",
             coef="deltafos", dpar= "sigma"),
          prior(student_t(10, 0, 0.3), class = "sd", nlpar = "oker"),
          prior(student_t(10, 0, 0.3), class = "sd", nlpar = "okp"),
          prior(student_t(10, 0, 0.7), class = "sd", nlpar = "oRLR"),
```

```
          prior(student_t(10, 0, 0.5), class = "sd", nlpar = "oRRF"),
          prior(lkj(1), class="cor"))

m2fit <- brm(frml2, data = dat337[!is.na(loss_ratio_train)],
       family = brmsfamily("lognormal", link_sigma = "log"),
       prior = mypriors2,
       stanvars = stanvar(scode = myFunsIncrPaid, block = "functions"),
       control = list(adapt_delta = 0.99, max_treedepth=15),
       file="models/section_5/CaseStudy_Model_2",
       seed = 123, iter = 2000, chains = 4)

m2fit
#>  Family: lognormal
#>   Links: mu = identity; sigma = log
#> Formula: loss_ratio_train ~ eta
#>          eta ~ log(claimsprocess(dev_year, 1, ker, kp, RLR, RRF, delta))
#>          ker ~ 3 * exp(oker * 0.1)
#>          kp ~ 1 * exp(okp * 0.1)
#>          RLR ~ 0.7 * exp(oRLR * 0.2)
#>          RRF ~ 0.8 * exp(oRRF * 0.1)
#>          oRLR ~ 1 + (1 | ID | accident_year) + (1 | dev_year)
#>          oRRF ~ 1 + (1 | ID | accident_year) + (1 | dev_year)
#>          oker ~ 1 + (1 | accident_year) + (1 | dev_year)
#>          okp ~ 1 + (1 | accident_year) + (1 | dev_year)
#>          sigma ~ 0 + deltaf
#>    Data: dat337[!is.na(loss_ratio_train)] (Number of observations: 90)
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>          total post-warmup samples = 4000
#>
```

```
#> Group-Level Effects:
#> ~accident_year (Number of levels: 9)
#>                               Estimate Est.Error
#> sd(oRLR_Intercept)                0.69      0.31
#> sd(oRRF_Intercept)                0.73      0.47
#> sd(oker_Intercept)                0.26      0.22
#> sd(okp_Intercept)                 2.23      1.35
#> cor(oRLR_Intercept,oRRF_Intercept)    0.37      0.47
#>                               l-95% CI u-95% CI
#> sd(oRLR_Intercept)                0.15     1.37
#> sd(oRRF_Intercept)                0.04     1.79
#> sd(oker_Intercept)                0.01     0.80
#> sd(okp_Intercept)                 0.57     5.44
#> cor(oRLR_Intercept,oRRF_Intercept)   -0.69     0.98
#>                               Rhat Bulk_ESS
#> sd(oRLR_Intercept)              1.00     1032
#> sd(oRRF_Intercept)              1.00     1042
#> sd(oker_Intercept)              1.00     3430
#> sd(okp_Intercept)               1.00      470
#> cor(oRLR_Intercept,oRRF_Intercept) 1.00     2682
#>                               Tail_ESS
#> sd(oRLR_Intercept)                1089
#> sd(oRRF_Intercept)                1527
#> sd(oker_Intercept)                1638
#> sd(okp_Intercept)                 1269
#> cor(oRLR_Intercept,oRRF_Intercept)     2614
#>
#> ~dev_year (Number of levels: 9)
```

```
#>                    Estimate Est.Error l-95% CI
#> sd(oRLR_Intercept)     0.35      0.26     0.01
#> sd(oRRF_Intercept)     1.11      0.49     0.13
#> sd(oker_Intercept)     0.27      0.23     0.01
#> sd(okp_Intercept)      0.45      0.43     0.02
#>                    u-95% CI Rhat Bulk_ESS Tail_ESS
#> sd(oRLR_Intercept)     0.98 1.00     1072     1669
#> sd(oRRF_Intercept)     2.12 1.00      949      921
#> sd(oker_Intercept)     0.86 1.00     3410     1546
#> sd(okp_Intercept)      1.32 1.01      552      903
#>
#> Population-Level Effects:
#>                  Estimate Est.Error l-95% CI u-95% CI
#> oRLR_Intercept       1.54      0.44     0.70     2.47
#> oRRF_Intercept      -1.45      0.69    -2.78    -0.07
#> oker_Intercept      -1.31      1.08    -3.40     0.84
#> okp_Intercept       -5.07      1.75    -7.71    -1.50
#> sigma_deltafos      -1.78      0.16    -2.13    -1.49
#> sigma_deltafpaid    -1.89      0.16    -2.20    -1.57
#>                  Rhat Bulk_ESS Tail_ESS
#> oRLR_Intercept   1.00     1167     1897
#> oRRF_Intercept   1.00     1689     2552
#> oker_Intercept   1.00     4150     2685
#> okp_Intercept    1.00      499     1205
#> sigma_deltafos   1.00     1109     1860
#> sigma_deltafpaid 1.00     1198     2632
#>
#> Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
```

```
#> is a crude measure of effective sample size, and Rhat is the potential
#> scale reduction factor on split chains (at convergence, Rhat = 1).
```

7.2.4 Model 3

```
CycleIndex <- data.table(
  accident_year = 1988:1997,
  RLM = c(1, 1.18, 1.22, 1.05, 1, 0.87, 0.94, 1.34, 1.64, 2.14)^0.6,
  RRM = c(1, 1.05, 1.05, 1.01, 1.0, 0.95, 0.99, 1.1, 1.25, 1.35)^0.6
)
setkey(dat337, accident_year)
setkey(CycleIndex, accident_year)
dat337 <- CycleIndex[dat337]

frml3 <- bf(loss_ratio_train ~ eta,
      nlf(eta ~ log(claimsprocess(dev_year, 1.0, ker, kp,
                          RLR, RRF, delta))),
      nlf(ker ~ 3 * exp(oker * 0.1)),
      nlf(kp ~ 1 * exp(okp * 0.1)),
      nlf(RLR ~ 0.7 * exp(oRLR * 0.2) * (RLM^lambda1)),
      nlf(RRF ~ 0.8 * exp(oRRF * 0.1) * (RRM^lambda2)),
      oRLR ~ 1 + (1 | ID | accident_year) + (1 | dev_year),
      oRRF ~ 1 + (1 | ID | accident_year) + (1 | dev_year),
      lambda1 ~  1,
      lambda2 ~ 1,
      oker ~ 1  + (1 | accident_year) + (1 | dev_year),
      okp ~ 1 + (1 | accident_year) + (1 | dev_year),
      sigma ~ 0 + deltaf, nl = TRUE)
```

```
mypriors3 <- c(prior(normal(0, 1), nlpar = "oRLR"),
          prior(normal(0, 1), nlpar = "oRRF"),
          prior(normal(0, 1), nlpar = "oker"),
          prior(normal(0, 1), nlpar = "okp"),
          prior(normal(log(0.2), 0.2), class = "b",
               coef="deltafpaid", dpar= "sigma"),
          prior(normal(log(0.2), 0.2), class = "b",
               coef="deltafos", dpar= "sigma"),
          prior(student_t(10, 0, 0.3), class = "sd", nlpar = "oker"),
          prior(student_t(10, 0, 0.3), class = "sd", nlpar = "okp"),
          prior(student_t(10, 0, 0.7), class = "sd", nlpar = "oRLR"),
          prior(student_t(10, 0, 0.5), class = "sd", nlpar = "oRRF"),
          prior(normal(1, 0.25), nlpar = "lambda1"),
          prior(normal(1, 0.25), nlpar = "lambda2"),
          prior(lkj(1), class="cor"))

m3fit <- brm(frml3, data = dat337[!is.na(loss_ratio_train)],
        family = brmsfamily("lognormal", link_sigma = "log"),
        prior = mypriors3,
        stanvars = stanvar(scode = myFunsIncrPaid, block = "functions"),
        control = list(adapt_delta = 0.99, max_treedepth=15),
        file="models/section_5/CaseStudy_Model_3",
        seed = 123, iter = 2000, chains = 4)

m3fit
#>  Family: lognormal
#>   Links: mu = identity; sigma = log
#> Formula: loss_ratio_train ~ eta
#>          eta ~ log(claimsprocess(dev_year, 1, ker, kp, RLR, RRF, delta))
```

```
#>        ker ~ 3 * exp(oker * 0.1)
#>        kp ~ 1 * exp(okp * 0.1)
#>        RLR ~ 0.7 * exp(oRLR * 0.2) * (RLM^lambda1)
#>        RRF ~ 0.8 * exp(oRRF * 0.1) * (RRM^lambda2)
#>        oRLR ~ 1 + (1 | ID | accident_year) + (1 | dev_year)
#>        oRRF ~ 1 + (1 | ID | accident_year) + (1 | dev_year)
#>        lambda1 ~ 1
#>        lambda2 ~ 1
#>        oker ~ 1 + (1 | accident_year) + (1 | dev_year)
#>        okp ~ 1 + (1 | accident_year) + (1 | dev_year)
#>        sigma ~ 0 + deltaf
#>    Data: dat337[!is.na(loss_ratio_train)] (Number of observations: 90)
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>        total post-warmup samples = 4000
#>
#> Group-Level Effects:
#> ~accident_year (Number of levels: 9)
#>                               Estimate Est.Error
#> sd(oRLR_Intercept)                0.29      0.21
#> sd(oRRF_Intercept)                0.80      0.40
#> sd(oker_Intercept)                0.26      0.21
#> sd(okp_Intercept)                 1.69      1.28
#> cor(oRLR_Intercept,oRRF_Intercept)     0.17      0.52
#>                               l-95% CI u-95% CI
#> sd(oRLR_Intercept)                0.01     0.80
#> sd(oRRF_Intercept)                0.08     1.62
#> sd(oker_Intercept)                0.01     0.78
#> sd(okp_Intercept)                 0.40     5.02
```

```
#> cor(oRLR_Intercept,oRRF_Intercept)    -0.85     0.96
#>                                  Rhat Bulk_ESS
#> sd(oRLR_Intercept)               1.00     1435
#> sd(oRRF_Intercept)               1.00     1383
#> sd(oker_Intercept)               1.00     4237
#> sd(okp_Intercept)                1.00      392
#> cor(oRLR_Intercept,oRRF_Intercept) 1.00     1747
#>                                  Tail_ESS
#> sd(oRLR_Intercept)                   1803
#> sd(oRRF_Intercept)                   1811
#> sd(oker_Intercept)                   2390
#> sd(okp_Intercept)                    1217
#> cor(oRLR_Intercept,oRRF_Intercept)     2039
#>
#> ~dev_year (Number of levels: 9)
#>                    Estimate Est.Error l-95% CI
#> sd(oRLR_Intercept)     0.37      0.27     0.02
#> sd(oRRF_Intercept)     1.16      0.52     0.17
#> sd(oker_Intercept)     0.27      0.23     0.01
#> sd(okp_Intercept)      0.48      0.53     0.02
#>                    u-95% CI Rhat Bulk_ESS Tail_ESS
#> sd(oRLR_Intercept)     1.02 1.00     1269     1897
#> sd(oRRF_Intercept)     2.24 1.00     1020      905
#> sd(oker_Intercept)     0.83 1.00     4717     2299
#> sd(okp_Intercept)      1.61 1.00      924      867
#>
#> Population-Level Effects:
#>                   Estimate Est.Error l-95% CI u-95% CI
```

```
#> oRLR_Intercept       1.33      0.42     0.58     2.25
#> oRRF_Intercept      -1.58      0.70    -2.91    -0.14
#> lambda1_Intercept    1.06      0.21     0.65     1.46
#> lambda2_Intercept    1.02      0.25     0.53     1.50
#> oker_Intercept      -1.35      1.09    -3.46     0.75
#> okp_Intercept       -5.78      1.69    -7.95    -1.85
#> sigma_deltafos      -1.80      0.15    -2.11    -1.52
#> sigma_deltafpaid    -1.91      0.16    -2.22    -1.60
#>                   Rhat Bulk_ESS Tail_ESS
#> oRLR_Intercept    1.00     1434     2624
#> oRRF_Intercept    1.00     2272     2852
#> lambda1_Intercept 1.00     6838     3540
#> lambda2_Intercept 1.00     8468     2802
#> oker_Intercept    1.00     4601     2525
#> okp_Intercept     1.01      450     1590
#> sigma_deltafos    1.00     1567     2758
#> sigma_deltafpaid  1.00     1535     2518
#>
#> Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
#> is a crude measure of effective sample size, and Rhat is the potential
#> scale reduction factor on split chains (at convergence, Rhat = 1).
```

Note that in order to predict from the models, the user-defined Stan functions have to be exported to R via the following:

```
expose_functions(m1, vectorize = TRUE)
expose_functions(m2, vectorize = TRUE)
expose_functions(m3, vectorize = TRUE)
```

7.3 Session information

```
utils:::print.sessionInfo(session_info, local=FALSE)
#> R version 3.6.1 (2019-07-05)
#> Platform: x86_64-apple-darwin15.6.0 (64-bit)
#> Running under: macOS Mojave 10.14.6
#>
#> Matrix products: default
#> BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
#> attached base packages:
#> [1] stats graphics  grDevices utils   data sets  methods   base
#>
#> other attached packages:
#>  [1] cowplot_1.0.0      ggridges_0.5.1
#>  [3] raw_0.1.6          MASS_7.3-51.4
#>  [5] knitr_1.25         modelr_0.1.5
#>  [7] forcats_0.4.0      stringr_1.4.0
#>  [9] dplyr_0.8.3        purrr_0.3.2
#> [11] readr_1.3.1        tidyr_1.0.0
#> [13] tibble_2.1.3       tidyverse_1.2.1
#> [15] tidybayes_1.1.0     latticeExtra_0.6-28
#> [17] RColorBrewer_1.1-2  lattice_0.20-38
#> [19] ChainLadder_0.2.10  data.table_1.12.2
#> [21] bayesplot_1.7.0     brms_2.10.0
#> [23] Rcpp_1.0.2          rstan_2.19.2
#> [25] ggplot2_3.2.1       StanHeaders_2.19.0
#> [27] deSolve_1.24
```

8. References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, et al., "Large-Scale Machine Learning on Heterogeneous Distributed Systems," TensorFlow white paper, 2015, https://www.tensorflow.org/about/bib.

Antonio, K., J. Beirlant, T. Hoedemakers, and R. Verlaak, "Lognormal Mixed Models for Reported Claims Reserves," North American Actuarial Journal 10:1, 2006, pp. 30–48.

Bååth, R., "Diagrams of Distributions in the Style of Kruschke," unpublished, 2012, https://github.com/rasmusab/distribution_diagrams.

Bates, D., M. Mächler, B. Bolker, and S. Walker, "Fitting Linear Mixed-Effects Models Using lme4," Journal of Statistical Software 67:1, 2015, pp. 1–48, https://doi.org/10.18637/jss.v067.i01.

Betancourt, M., "A Conceptual Introduction to Hamiltonian Monte Carlo," arXiv preprint, 2017, https://arxiv.org/abs/1701.02434.

Betancourt, M., "Towards a Principled Bayesian Workflow (RStan)," unpublished, 2018, https://betanalpha.github.io/assets/case_studies/principled_bayesian_workflow.html.

Bornhuetter, R. L., and R. E. Ferguson, "The Actuary and IBNR," Proceedings of the Casualty Actuarial Society 59:112, 1972, pp. 181–195.

Bühlmann, H., and E. Straub, "Glaubgwürdigkeit Für Schadensätze," Bulletin of the Swiss Association of Actuaries 70, 1970, pp. 111–133.

Bürkner, P.-C., "brms: An R Package for Bayesian Multilevel Models Using Stan," Journal of Statistical Software 80:1, 2017, pp. 1–28, https://doi.org/10.18637/jss.v080.i01.

Bürkner, P., "Define Custom Response Distributions with brms," The Comprehensive R Archive Network, 2020, https://cran.r-project.org/web/packages/brms/vignettes/brms_customfamilies.html.

Carpenter, B., A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A Probabilistic Programming Language," Journal of Statistical Software 76:1, 2017, pp. 1–32, https://doi.org/10.18637/jss.v076.i01.

Carrato, A., and M. Visintin, "From the Chain Ladder to Individual Claims Reserving Using Machine Learning Techniques," paper presented at ASTIN Colloquium, Cape Town, South Africa, April 2–5, 2019, https://www.colloquium2019.org.za/wp-content/uploads/2019/04/Alessandro-Carrato-From-Chain-Ladder-to-Individual-Claims-Reserving-using-Machine-Learning-ASTIN-Colloquium.pdf.

CAS Loss Simulation Model Working Party, CAS Loss Simulator 2.0, R package, 2018, https://www.casact.org/research/lsmwp/index.cfm?fa=software.

Clark, D. R., LDF Curve-Fitting and Stochastic Reserving: A Maximum Likelihood Approach, Casualty Actuarial Society Forum, Fall 2003, pp. 41-92, http://www.casact.org/pubs/forum/03fforum/03ff041.pdf.

Cooney, M., "Modelling Loss Curves in Insurance with RStan," Stan Case Studies 4, 2017, https://mc-stan.org/users/documentation/case-studies/losscurves_casestudy.html.

England, P. D., and R. J. Verrall, "A Flexible Framework for Stochastic Claims Reserving," Proceedings of the Casualty Actuarial Society 88:168, 2001, pp. 1–38, https://www.casact.org/pubs/proceed/proceed01/01001.pdf.

Fannin, B. A., Raw: R Actuarial Workshops, R package version 1.0.6, 2018, https://CRAN.R-project.org/package=raw.

Gabrielli, A., R. Richman, and M. V. Wüthrich, "Neural Network Embedding of the Over-Dispersed Poisson Reserving Model," Social Science Research Network preprint, 2018, https://Ssrn.com/Abstract=3288454.

Gao, G., Bayesian Claims Reserving Methods in Non–Life Insurance with Stan: An Introduction, Singapore: Springer Nature, 2018, https://doi.org/10.1007/978-981-13-3609-6.

Gelman, A., B. Carlin, H. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, Bayesian Data Analysis (3rd ed.), Chapman & Hall/CRC Texts in Statistical Science, Boca Raton, FL: Chapman & Hall/CRC, 2014.

Gelman, A., and J. Hill, Data Analysis Using Regression and Multilevel/Hierarchical Models, Analytical Methods for Social Research, Cambridge, UK: Cambridge University Press, 2007.

Gesmann, M., "Modellierung Und Analyse Neuronaler Dynamiken (Modeling and Analysing Neuronal Dynamics)," master's thesis, University of Cologne, Germany, 2002.

Gesmann, M., "Hierarchical Loss Reserving with Growth Curves Using brms," Mages' Blog, July 15, 2018, https://magesblog.com/post/2018-07-15-hierarchical-loss-reserving-with-growth-cruves-using-brms/.

Gesmann, M., and D. de Castillo, "Using the Google Visualisation API with R," The R Journal 3:2, 2011, pp. 40–44, https://journal.r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf.

Gesmann, M., D. Murphy, W. Zhang, A. Carrato, M. Wüthrich, and F. Concina, ChainLadder, R package, 2019, https://github.com/mages/ChainLadder.

Guszcza, J., "Hierarchical Growth Curve Models for Loss Reserving," Casualty Actuarial Society E-Forum, Fall 2008, pp. 146–173, https://www.casact.org/pubs/forum/08fforum/7Guszcza.pdf.

Kruschke, J., Doing Bayesian Data Analysis: A Tutorial with R, Jags, and Stan, London: Academic Press, 2014, https://books.google.co.uk/books?id=FzvLAwAAQBAJ.

Kuo, K., "DeepTriangle: A Deep Learning Approach to Loss Reserving," arXiv preprint, 2018, https://arxiv.org/abs/1804.09253.

Lally, N., and B. Hartman, "Estimating Loss Reserves Using Hierarchical Bayesian Gaussian Process Regression with Input Warping," Insurance: Mathematics and Economics 82, 2018, pp. 124–140, https://doi.org/https://doi.org/10.1016/j.insmatheco.2018.06.008.

Lewandowski, D., D. Kurowicka, and H. Joe, "Generating Random Correlation Matrices Based on Vines and Extended Onion Method," Journal of Multivariate Analysis 100:9, 2009, pp. 1989–2001, https://doi.org/https://doi.org/10.1016/j.jmva.2009.04.008.

Lunn, D. J., A. Thomas, N. Best, and D. Spiegelhalter, "WinBUGS—a Bayesian Modeling Framework: Concepts, Structure, and Extensibility," Statistics and Computing 10:4, 2000, pp. 325–37.

Mack, T., "Distribution-Free Calculation of the Standard Error of Chain Ladder Reserve Estimates," ASTIN Bulletin 23, 1993, pp. 213–25.

McElreath, R., Statistical Rethinking: A Bayesian Course with Examples in R and Stan, Chapman & Hall/CRC Texts in Statistical Science, Boca Raton, FL: CRC Press, 2015.

McNulty, G., "Severity Curve Fitting for Long Tailed Lines: An Application of Stochastic Processes and Bayesian Models," Variance 11:1, 2017, pp. 118–132.

Meyers, G., "Stochastic Loss Reserving Using Bayesian MCMC Models," monograph no. 1, Arlington, VA: Casualty Actuarial Society, 2015, http://www.casact.org/pubs/monographs/papers/01-Meyers.PDF.

Morris, J., "Hierarchical Compartmental Models for Loss Reserving," Casualty Actuarial Society E-Forum, Summer 2016, pp. 1–72, https://www.casact.org/pubs/forum/16sforum/Morris.pdf.

Parodi, P., "Triangle-Free Reserving: A Non-Traditional Framework for Estimating Reserves and Reserve Uncertainty," British Actuarial Journal 19:1, 2014, pp. 219–233, https://doi.org/10.1017/S1357321713000354.

Plummer, M., "JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling," presented at international workshop on distributed statistical computing, Vienna, Austria, March 20–22, 2003.

R Core Team, R: A Language and Environment for Statistical Computing, Vienna, Austria: R Foundation for Statistical Computing, 2019, https://www.R-project.org/.

Salvatier, J., T. V. Wiecki, and C. Fonnesbeck, "Probabilistic Programming in Python Using PyMC3," PeerJ Computer Science 2, April 2016, e55, https://doi.org/10.7717/peerj-cs.55.

Schmid, F. A., "Robust Loss Development Using MCMC," SSRN Electronic Journal, February 2010, https://doi.org/10.2139/ssrn.1501706.

Schmidt, K. D., "Methods and Models of Loss Reserving Based on Run-Off Triangles: A Unifying Survey," Casualty Actuarial Society Forum, Fall 2006, pp. 269–317, https://www.math.tu-dresden.de/sto/schmidt/publications_online/2006-cas-method.pdf.

Scollnik, D. P. M. "Actuarial Modeling with MCMC and BUGS," North American Actuarial Journal 5:2, 2001, pp. 96–124.

Sherman, R. E., "Extrapolating, Smoothing, and Interpolating Development Factors," Proceedings of the Casualty Actuarial Society 71:136, 1984, pp. 122–155, https://www.casact.org/pubs/proceed/proceed84/84122.pdf.

Stan Development Team, RStan: The R Interface to Stan, R package, 2019, http://mc-stan.org/.

Taylor, G. C., and F. R. Ashe, "Second Moments of Estimates of Outstanding Claims," Journal of Econometrics 23:1, 1983, pp. 37–61.

Verrall, R., "A Bayesian Generalized Linear Model for the Bornhuetter-Ferguson Method of Claims Reserving," North American Actuarial Journal 8:3, 2004, pp. 67–89, https://doi.org/10.1080/10920277.2004.10596152.

Winkel, B. J., "Modelling Mixing Problems with Differential Equations Gives Rise to Interesting Questions," International Journal of Mathematical Education in Science and Technology 25:1, 1994, pp. 55–60, https://doi.org/10.1080/0020739940250107.

Wüthrich, M. V., "Machine Learning in Individual Claims Reserving," Scandinavian Actuarial Journal 2018:6, 2018, pp. 465–80.

Zehnwirth, B., and G. Barnett, "Best Estimates for Reserves," Proceedings of the Casualty Actuarial Society 87:167, 2000, pp. 245–321.

Zhang, Y., V. Dukic, and J. Guszcza, "A Bayesian Nonlinear Model for Forecasting Insurance Loss Payments," Journal of the Royal Statistical Society, Series A 175, 2012, pp. 637–656.