All Actuaries Summit 2026
25 – 27 May 2026, Melbourne

Actuaries Institute.

Life insurance applications of Bayesian models

Prepared by Colin Yellowlees and Tikiri Appuhamy

Presented to the Actuaries Institute
2026 All-Actuaries Summit
25-27 May 2026

Actuaries Institute
ABN 69 000 423 656
Level 34, Australia Square, 264 George Street, Sydney NSW 2000
P +61 (0) 2 9239 6100 | actuaries.asn.au

Abstract

The aim of this paper is to motivate life insurance actuaries to explore how Bayesian methods can become part of actuarial practice. We will introduce Bayesian approaches, highlighting how they differ from more traditional frequentist methods and why they can be advantageous. We will discuss their flexibility in handling complex models, their natural treatment of uncertainty through posterior distributions, and their ability to incorporate expert judgment or external information via priors.

Bayesian methods can allow actuaries to solve actuarial problems in ways that are often difficult under traditional approaches. For example, they allow actuaries to quantify the full distribution of outcomes rather than relying solely on point estimates or asymptotic approximations. They also adapt well to situations with sparse data, shifting business practices, or environments where existing knowledge can add meaningful insights. These strengths can improve decision-making in contexts such as pricing, reserving, and capital modelling, where understanding the range and drivers of uncertainty is critical.

Bayesian methods remain underused in life insurance practice. Barriers include the perception that Bayesian models are complex, the high computational burden, and a lack of familiarity among actuaries. However, with the growth of modern statistical software, advances in Markov Chain Monte Carlo and increasing exposure in other actuarial domains such as general insurance, these hurdles are diminishing. We believe that there is opportunity for life actuaries to adopt Bayesian methods more widely.

To illustrate this, the paper will present two case studies. The first will examine Bayesian approaches to experience investigations, showing how Bayesian modelling can address sparse data and credibility challenges. The second will explore prediction of incurred but not reported (IBNR) claims, highlighting how prior information and uncertainty quantification can improve insights.

Keywords: Bayesian modelling; IBNR; negative binomial; generalised linear models; regularisation; model diagnostics; experience studies; Bayesian updating; MCMC; Stan; R; Python

Table of contents

1. Introduction

The models and assumptions we use in actuarial work are never accurate. Actuaries seek ways to measure the uncertainty in model estimates by implementing a variety of tools and metrics such as scenario/sensitivity testing, statistical tests, cross-validation, measures of fit, and standard error confidence intervals.

In this paper we present a class of models that blend assumptions, data, and model uncertainty in a systematic framework known as Bayesian models.

The aim of this paper is to provide a motivation and an easy entry point for Life Actuaries to understand and start applying Bayesian modelling in their day-to-day work. Recognising that most Life Actuaries have limited familiarity with Bayesian modelling we have to start with the fundamentals and work up from there. This paper does not aim to be a textbook or technical introduction to Bayesian modelling, so parts lean more on providing a mental model of how they work and what they do rather than providing a robust treatise on Bayesian modelling.

While the standing and recognised value of Bayesian modelling has increased significantly in recent years, and there are many resources readily available across the internet for beginners, our objective is to bring essential concepts together in a single place specifically in the context of life insurance applications.

2. Why use Bayesian methods?

Bayesian modelling is a natural paradigm for actuaries. Historically, the application of Bayesian methods in actuarial settings was limited due to their complexity and computational intensity. These constraints have since been lifted and there are now strong motivations for considering them:

For an actuary, Bayesian modelling is particularly valuable because it provides a structured way to combine data, expert judgement, and prior actuarial assumptions in a transparent and clear framework. This is especially important where experience is sparse, volatile, or emerging (an example using termination rates is shown in a case study below). In these settings, relying only on raw internal experience can lead to over-reaction to noise, while relying only on existing assumptions can delay recognition of genuine changes in experience. Bayesian modelling provides a practical middle ground by allowing the actuary to start with a prior view and then update that view as evidence emerges. This can also improve documentation, challenge, and communication with boards, auditors, and regulators with a clear linkage from the inputs to the outputs which can be seen as more explicit and auditable.

Another important feature for the actuary is the natural quantification of uncertainty. Rather than relying only on point estimates, Bayesian methods quantify the full range of plausible outcomes and therefore support better assessment of reserve adequacy, assumption uncertainty, and the potential need for management action. This can help build trust that is hard to build with point estimates that are always wrong. For example, Bayesian models allow the actuary to change the message from "experience deviating from a point estimate" to whether "experience has been tracking within a reasonable prediction interval".

Bayesian modelling can also fit naturally into the actuarial control cycle. Assumptions do not need to be reset from scratch each time experience is reviewed. Instead, the posterior distribution from one review can become the prior distribution for the next review, allowing experience and judgement to be incorporated on an ongoing basis. This creates a disciplined process for updating assumptions over time and provides a clear link between past views, new evidence and revised conclusions.

From a governance perspective, Bayesian modelling also has advantages. Priors, likelihoods and model outputs are all explicit and can therefore be documented, challenged and reviewed. Sensitivity testing can be used to show how conclusions change under alternative prior assumptions or modelling choices. This can improve the defensibility of actuarial recommendations and make it easier to explain how professional judgement has been exercised.

For these reasons, Bayesian modelling should be viewed as a useful addition to the actuary's toolkit.

Some specific examples of how Bayesian modelling could be used would be:

- Updating pricing or valuation assumptions for a recently launched mortality or morbidity product where we still have limited experience. The initial pricing basis could be used as a prior and then produce an updated basis using a Bayesian model that naturally weights the prior knowledge and the recent experience. An example of how this can be done is contained in the termination rate case study below. This would be an ongoing process to continuously update the basis as required.

- Consider a risk appetite statement that says, "the chance of losing more than $20 million from mortality deterioration should be below 5%". A Bayesian model would combine historical experience, expert judgment, and the latest claims data to estimate the full distribution of next year's mortality outcome. Based on the updated posterior distribution, the probability of exceeding the $20 million threshold might be calculated to be 12%, so the portfolio is currently outside appetite. Management actions such as repricing or additional reinsurance could then be modelled.

3. Bayesian method basics

The core engine for all Bayesian models is Bayes Theorem (Andrew Gelman, 2015):

$$Pr(B) \cdot Pr(A|B) = Pr(A) * Pr(B|A)\text{; or.}$$

$$\Pr(\text{A}|\text{B}) = \frac{Pr(A) * Pr(B|A)}{Pr(B)}$$

A useful interpretation of the formula above is as follows:

$$\overbrace{\Pr(\theta \mid \text{Y})}^{\text{Posterior}} = \frac{\overbrace{\Pr(\theta)}^{\text{Prior}} \times \overbrace{\Pr(\text{Y} \mid \theta)}^{\text{Likelihood}}}{\Pr(\text{Y})}$$

Where:

- $\Pr(\text{Y}|\theta)$, the *likelihood function*, is the probability of observing some data Y for a given parameter value θ.
- $\Pr(\theta)$, the *prior distribution*, is the probability of the parameter taking a certain value based on our beliefs before the evidence or data Y has been considered.
- $\Pr(\text{Y})$ is the probability of observing Y and all possible values of θ. It acts as a normalisation factor to ensure that the posterior distribution is a proper distribution and can be integrated or summed to 1. To calculate $\Pr(\text{Y})$, we must take the sum of $\Pr(\Theta = \theta) \times \Pr(\text{Y}|\theta)$ for all possible values of θ i.e. $\Pr(\text{Y}) = \int_{\Theta} Pr(\theta) \times Pr(Y|\theta)\, d\theta$ for the continuous case.
- $\Pr(\theta \mid \text{Y})$ is the *posterior distribution*, is the updated probability distribution of θ after observing the data Y. It is the update to our prior beliefs about θ after we have observed Y.

Another way to express this formula is as follows:

$$\Pr(Assumption \mid Data) = \frac{Pr(Assumption) \times Pr(Data \mid Assumption)}{Pr(Data\ or\ Evidence)}$$

Once the posterior distribution has been estimated it is then easy to calculate quantities of interest, for example, expected values or distributions of future observations/unobserved data.

4. Prior selection

Selecting priors is as much art as it is science. There are many alternatives on how priors can be set with no single right way. In many actuarial applications priors somewhere between being weakly informative and informative is a good start.

Looking at the different classifications of priors:

Uninformative, noninformative or diffuse priors

Uninformative priors try to have minimal influence and contribute very little information to the posterior in an attempt to “allow the data to speak for itself” (Andrew Gelman, 2015).

The idea here is to select a prior that assigns similar probabilities to all possibilities, this causes the likelihood function to have more power in updating the posterior. In many circumstances uninformative priors are quite flat, for example, a uniform distribution or a prior of normal(0,100) for a normalised parameter would be considered uninformative.

Weakly-informative priors

Weakly-informative priors are designed to be intentionally less informative than the actual prior knowledge a practitioner may have (Andrew Gelman, 2015). Weakly-informative priors seek to regularise (i.e. keep the models estimates reasonable and stable) the posterior distribution given the observed data.

They achieve this by being wider than an informative prior and more concentrated than an uninformative prior. The idea is to avoid considering unrealistic parameter values while allowing the model to still consider potential parameter values that may be outside of the bounds of what is suggested by the data. They also give the model a little more structure which helps with computational efficiency (Andrew Gelman, 2015).

Informative priors

Informative priors reflect strong prior knowledge by incorporating judgement and other information into their setting; they are generally characterised by having more focused distributions than weakly-informative priors. By using an informative prior the model will balance the inference from the data with the information encoded into the prior.

The measurement of how informative a prior is best understood relative to the likelihood and observations used in the model. If the data outweighs the prior information, the prior will have less influence on the posterior inference. This is important when applying the same model to datasets of different sizes, since a prior that materially influences results for a small dataset may have limited effect when applied to a larger dataset.

Care should be taken in using informative priors to avoid the following issues:

- Prior information should ideally not be based on any of the information available in the observations used to fit the model as this may lead to predictive intervals being too narrow and overconfident in the results as the prior and the likelihood function in this circumstance are double counting the same information.
- The prior distribution may pull inferences towards predetermined outcome; this is not a great outcome if the purpose of model is to test a hypothesis (Andrew Gelman, 2015)

The advantages of using an informative prior are:

- They provide a baseline for the posterior to regularise in areas where there isn't much information to fit to.
- Priors that are well set can have a positive impact on being able to fit the model when the posterior distribution has a difficult shape to explore. They also tend to converge quicker. Convergence problems can often arise when priors are set poorly.

The disadvantages of using an informative prior:

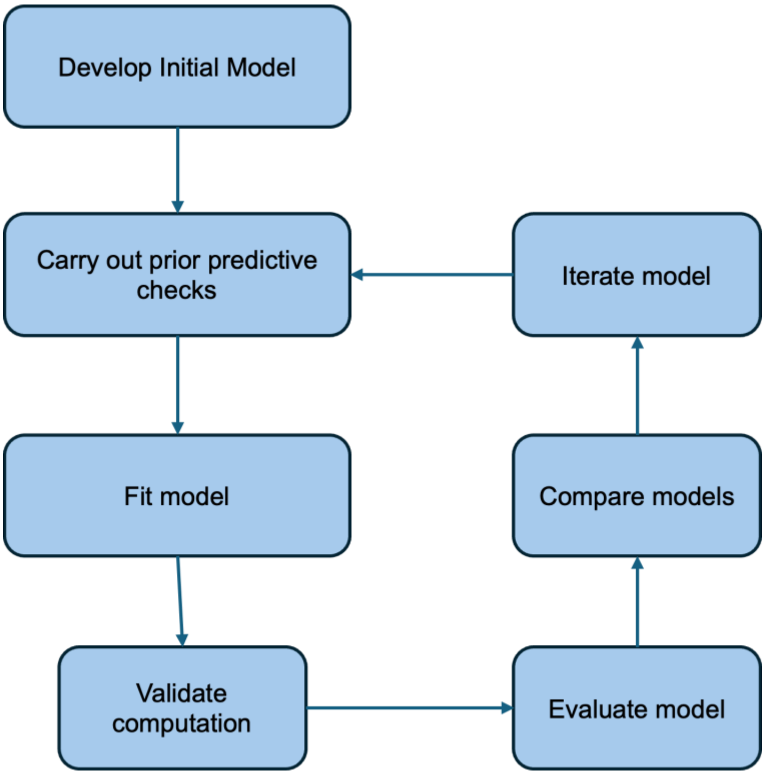
- If they are mis-specified, they will pull the posterior into the wrong direction which the data cannot correct. This feature is difficult to diagnose in practice.
- Developing an informative prior can be difficult in practice as this involves encoding knowledge into parameters and their prior distributions.
- They are more open to scrutiny than weakly informative priors. It is generally much easier for a group of experts to agree upon values in the tail than the median (Betancourt, Principled Bayesian Workflow, 2020).

An example of an informative prior would be to take an insurers pricing basis and add a distribution around it. The distribution represents the level of uncertainty in the pricing basis itself and its relevance to whatever is being modelled.

5. Workflow

When working with Bayesian models there is a general workflow that should be followed to help ensure that your models are well founded and robust. A high-level outline of the workflow, based upon (Gelman, et al., 2020), can be seen in Figure 1 below.

Figure 1: Simple workflow to use for Bayesian modelling



Develop initial model

The first step is to decide which probabilistic programming language (PPL) to use. The most common and easiest languages to start with are PyMC and Stan. PyMC is a Python library that allows you to run Bayesian models in Python. Stan is its own language, and the models written in Stan can be run and compiled from several languages including Python, R and Julia. For much of this paper we have used Stan and given the statistical background of actuaries we believe that it is likely to be the easier choice as the language more closely follows the statistical representation of the underlying statistical model.

Usually, you will have an existing model or previous analysis that you want to turn into a Bayesian model. In this case it is relatively uncomplicated to turn this into a Bayesian model. There are plenty of resources available to provide examples of all the common models and how they can be coded in Stan or PyMC.

If you are developing a model from scratch, then it is generally best to start with a simple model. If you know what non-Bayesian model you would start with then that is probably as good a start as any.

Something that is unique to Bayesian models is the need to specify priors for all the parameters. (Note that for most PPLs this is often not necessary, however, the algorithms will usually work much better if all parameters have priors and the model is not strictly a Bayesian model.). This paper includes a longer discussion on priors in the section above.

Prior predictive checks

Prior predictive checks are a very specific check that you do on a model before fitting the model on any data. A prior predictive check is running the model using the priors to look at simulations of data that the model produces. The aim is to see if the chosen priors and model are reasonable given our domain knowledge of the problem (Gabry, Simpson, Vehtari, Betancourt, & Gelman, 2019). Prior predictive checks are covered further in the group IBNR case study.

Fit model

Once we are happy with the prior predictive checks, we would run the model with real data. Most Bayesian models will be fitted using an MCMC algorithm. See MCMC section below for a description of how these models are fitted.

Validate computation

With MCMC, the aim is to explore the posterior distribution through a simulation. Unlike other parameter estimation techniques, we need to validate if the simulation has adequately explored the posterior distribution. With Bayesian models the output can look usable even when the diagnostics indicate the computation is unreliable. The main checks we would carry out are:

- Convergence checks. For this we look at trace and density plots and R-hat and ESS metrics. These are covered further in the Model Checking section below.

- Parameter recovery checks. This is where we use synthetic datasets where we know the underlying parameter to make sure that our model outputs sensible parameters close to those that generated the data. This is especially important for more complicated models.

Evaluate and compare models

Once we have validated the MCMC procedure, we need to evaluate the inference that is implied from the fitted model and compare this to the results of other models fitted. Procedures used to evaluate Bayesian models include:

- Posterior predictive checks: These checks involve generating many synthetic datasets using the fitted model and covariates from the observed data. This is equivalent to generating an expected value for each observed datapoint to perform actual vs expected style analysis. The aim of these checks is to compare the synthetic datasets to the observed for overall fit and to see if the model can reproduce important features of the observed data (Gabry, Simpson, Vehtari, Betancourt, & Gelman, 2019). For a good model we would expect, for example, that 95% of all the data points fall within the 95% posterior predictive interval. We can also look at a Q-Q plot of the posterior predictive draws vs observed data or a PIT histogram. (PIT stands for Probability Integral Transform. The histogram plots the CDF for each observation *i* (i.e. $F(y_i) = Pr(Y \leq y_i)$.)

- Out-of-sample model performance metrics: Like with other predictive models, we want to compare how iterations of models compare in terms of their out-of-sample predictive performance. For Bayesian models, we can generate metrics that estimate this such as Leave One Out Cross Validation (LOOCV), Watanabe-Akaike Information Criterion (WAIC) and Pareto Smoothed Importance Sampling (PSIS) (Vehtari, Gelman, & Gabry, Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC, 2017). These metrics are discussed in the Resources recommended for further reading in this paper.

Iterate model

Once we have a model that we consider is trustworthy we can then go about modifying and building on it. This might mean adding additional features, adjusting priors and sampling distributions etc. We would go through the above process all over again until we arrive at our final model/models.

6. Specific features and strengths of Bayesian modelling

Of interest to actuaries is that there are some modelling techniques and features that fit easily with Bayesian modelling. Below are some worth highlighting:

Hierarchical modelling

Hierarchical models are a way to share information across groups or features to produce more stable and reliable estimates, particularly where data is limited.

In actuarial work this is closely related to credibility. Instead of estimating segments independently, a hierarchical model assumes that related segments are connected through a common underlying distribution. For example, termination rates might be estimated separately by product, duration, age band or occupation class, while still allowing each segment to learn from the overall portfolio experience. Segments with limited experience are pulled more strongly towards the portfolio average, while segments with credible experience are allowed to differ more.

Gaussian Processes

Gaussian Processes are a method in Bayesian modelling to fit a non-parametric curve to data. The core of the idea is that observed data that is close to other observed data is correlated and you can use therefore the data at nearby points to get a better fit of the curve to the data.

For actuaries, Gaussian Processes can be thought of as a flexible way to smooth rates or trends without imposing a fixed functional form. For example, they could be used to smooth mortality, morbidity or termination rates by age, duration or calendar time. Rather than selecting a specific curve shape in advance, the model places a prior over possible smooth curves and then updates this using the observed experience.

Handling censored and truncated data

Censoring is where an observation is included in your data set, but the value of interest is not known exactly. Truncation is where the observation is not in your data set unless it falls within a specified range. In practice, with truncated data, the excluded observations are not seen at all, and usually the number excluded is unknown.

A classic example of censored data in life insurance are disability termination rates where claims are still in payment, so the eventual termination time has not been observed. This specific example is right censoring.

An example of truncation in life insurance is where there are claim reporting delays. Claims not yet reported at the valuation date (reserved for in an IBNR) are absent from the data set entirely and are truncated due to the reporting delay.

There is also interval censoring where the event time is only known to be between two values or dates. An example of this is where claims are placed into reporting delays in claim development triangles for IBNR calculations.

Bayesian modelling can handle censored or truncated data easily by adjusting the likelihood function to allow for the observation mechanism. For censored observations we use a likelihood based on probability that the censored observation was above or below the censoring point. For truncation we divide the likelihood for the observed data by the probability that the data was observed.

Modelling latent variables and missing structure

Bayesian regression can be used to try and find hidden structures to the data. An example related to life insurance might be where you have operational impacts over time to the management of disability claims. This can have an impact on the observed termination rate. By including a latent variable to account for these operational impacts we can get a better understanding of the underlying termination rates. We can also see if the hidden structure aligns with our understanding of how operational processes have changed.

Regularisation

In traditional machine learning we are familiar with GLM models being extended to the likes of LASSO, Ridge regression or elastic net regression. Bayesian regression also has the ability to allow for regularisation or shrinkage of parameters. In Bayesian regression:

- The equivalent to Ridge regression (or L2 regularisation) is to use a normal prior centred at 0 on each of the parameters.
- The equivalent of LASSO regression (or L1 regularisation) is to use a Laplace distribution (or double exponential distribution).

There are also additional ways to regularise Bayesian regression via:

- A horseshoe prior. The idea here is to apply an aggressive shrinkage of noise and a minimal shrinkage of signal by aiming to only shrink small parameters (Carvalho, Polson, & Scott, 2010).
- A spike-and-slab prior can be more effective at making parameters 0 than a LASSO regression (Mitchell & Beauchamp, 1988).

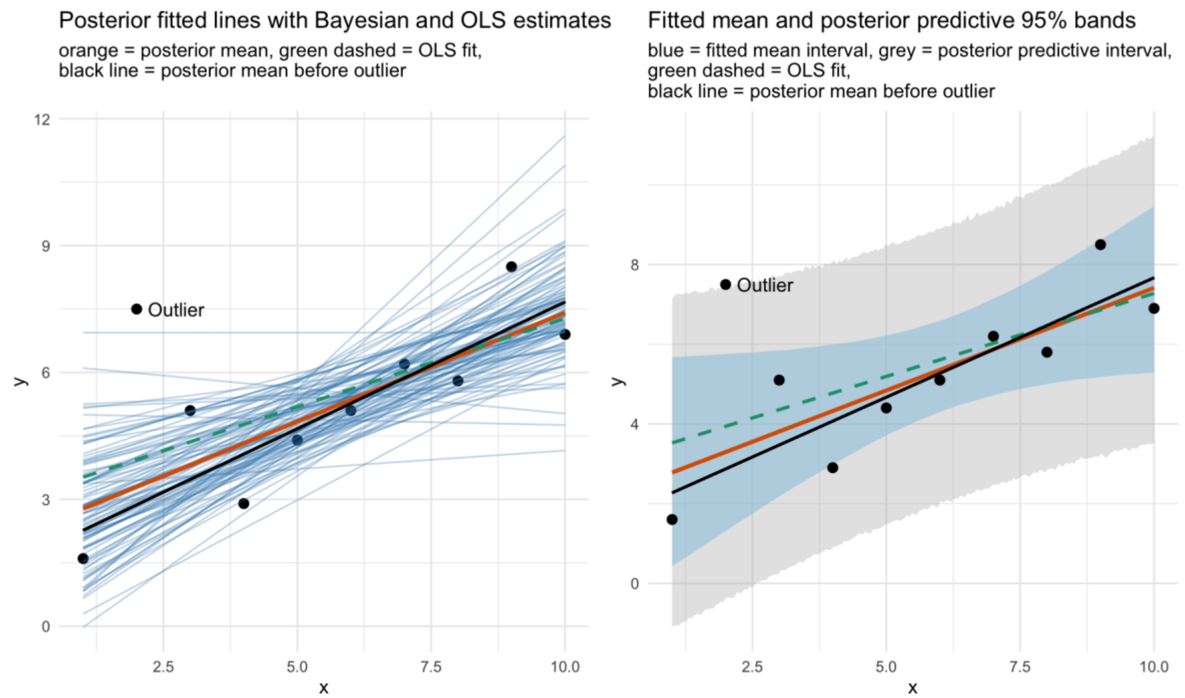
Over-dispersion

Often, the data we work with is generated by a messy process with many unknown influences. This can lead to over-dispersion. For example, count data that would ideally follow a Poisson distribution, for which the mean equals the variance, may instead be over-dispersed, with variance greater than the mean. In such cases, we can easily use a likelihood function, like a negative binomial distribution, that can accommodate over-dispersion.

Robust Regression

From simple linear regression through to Generalised Linear Models a common feature of the data we deal with is outliers. Outliers can sometimes be difficult to identify and handle especially in higher dimensional data. In more traditional regression you would normally go through a process of identifying outliers and then treating them in a couple of ways: either by removing them from the data or normalising them in some manner so that they can still be included. In Bayesian modelling there is a natural way to handle outliers by changing the likelihood function to one that anticipates the existence of outliers and reduces the sensitivity to them. For example, in a linear regression you would change the normal distribution to a student-t distribution. The following shows what happens in the above example if one of the data points was an outlier.

Figure 2: Line of best fit and posterior predictive intervals from an OLS and Bayesian linear regression.



The orange line is the updated Bayesian regression using the data including the outlier, the dotted line is the updated ordinary least squares regression, and the thin line is the original Bayesian/OLS regression line.

Building complex or compound models

With Bayesian models it is relatively easy to join or blend two different models. For example, you might have models that:

- Fit a morbidity curve to your TPD experience split by age and gender.
- Calculate the IBNR for TPD claims.

With Bayesian modelling you could combine these into a single model so that you can calculate IBNR and a morbidity curve in a single process.

7. Example 1 – Bayesian Updating

The example discussed below walks through how a Bayesian model can be updated as new empirical data is added to the model. In the model we use synthetic data where we know the underlying parameters to show that as we add more data it converges to the true underlying parameter used to generate the data. The example demonstrates Bayesian updating analytically by constraining the domain of the prior distribution. When this constraint is loosened, estimation approaches are required such as MCMC to estimate the posterior distribution.

In this example, we have a new critical illness product where we need decline rates to calculate a Reported But Not Admitted reserve (RBNA). The assumed true decline rate is 10%, however, this is unknown to us at product launch. We observe a steady rate of 30 reported claims per month. We can develop a Bayesian model to estimate the frequency of declined claims and update the posterior of this model each period as we observe new claims.

The data generated for this example was generated using a binomial distribution assuming N = 30 and p = 10% for each of the 12 months. The claims data generated is summarised below:

Table 1: Generate Claim Data from binomial distribution

Month	Reported Claims	Declines	Observed Decline Rate
1	30	3	10%
2	30	6	20%
3	30	4	13%
4	30	2	7%
5	30	2	7%
6	30	2	7%
7	30	2	7%
8	30	6	20%
9	30	6	20%
10	30	4	13%
11	30	2	7%
12	30	5	17%

Our model assumes that the frequency of declines is binomially distributed – this is the likelihood function Pr(Y|p). The binomial distribution only has one parameter, p, which is the decline rate. The likelihood function in this example is as follows:

$$Pr(Y|p) = \binom{N}{Y} p^Y (1-p)^{N-Y}$$

A Bayesian model will typically have a prior distribution on each of the parameters implicit in the likelihood function. To analytically calculate the posterior distribution, assume that p can only take the values 0.05, 0.10, 0.15, 0.20 and 0.25 and that each value is equally likely i.e. probability of any value of p = 0.20.

$$\Pr(P = p) = 0.20 \quad \text{where } p \in \{0.05,\ 0.10,\ 0.15,\ 0.20,\ 0.25\}$$

Or more compactly

$$P \sim Uniform(0.05,\ 0.10,\ 0.15,\ 0.20,\ 0.25)$$

The posterior can then be calculated using Bayes theorem as follows:

$$Pr(p\,|\,Y) = \frac{Pr(Y|\,p) \times Pr(P = p)}{\int_P Pr(Y|\theta) \times Pr(P = \theta)\,d\theta} = \frac{\binom{N}{Y} p^Y (1-p)^{N-Y} \times Pr(P = p)}{\sum_P \binom{N}{Y} \theta^Y (1-\theta)^{N-Y} \times Pr(P = \theta)}$$

Given that the prior given is a uniform distribution the formula simplifies to:

$$Pr(p\,|\,Y) = \frac{\binom{N}{Y} p^Y (1-p)^{N-Y} \times 0.2}{\sum_P \binom{N}{Y} \theta^Y (1-\theta)^{N-Y} \times 0.2}$$

Valuation at start

In the first valuation period since we do not have any data yet we can use the prior distribution to calculate the expected decline rate:

$$E[p] = 0.20 \times (0.05 + 0.10 + 0.15 + 0.20 + 0.25) = 15\%$$

Valuation after month 1

In the first valuation period, 3 out of the 30 claims reported are declined. We can update the posterior distribution using this data point and formulas specified above, the detailed calculations are found below, given that there were 3 declined claims out of 30 claims reported Y = 3 and N = 30 and our prior distribution's domain only has 5 values so $p \in$ $0.05,\ 0.1,\ 0.15,\ 0.2, 0.25$:

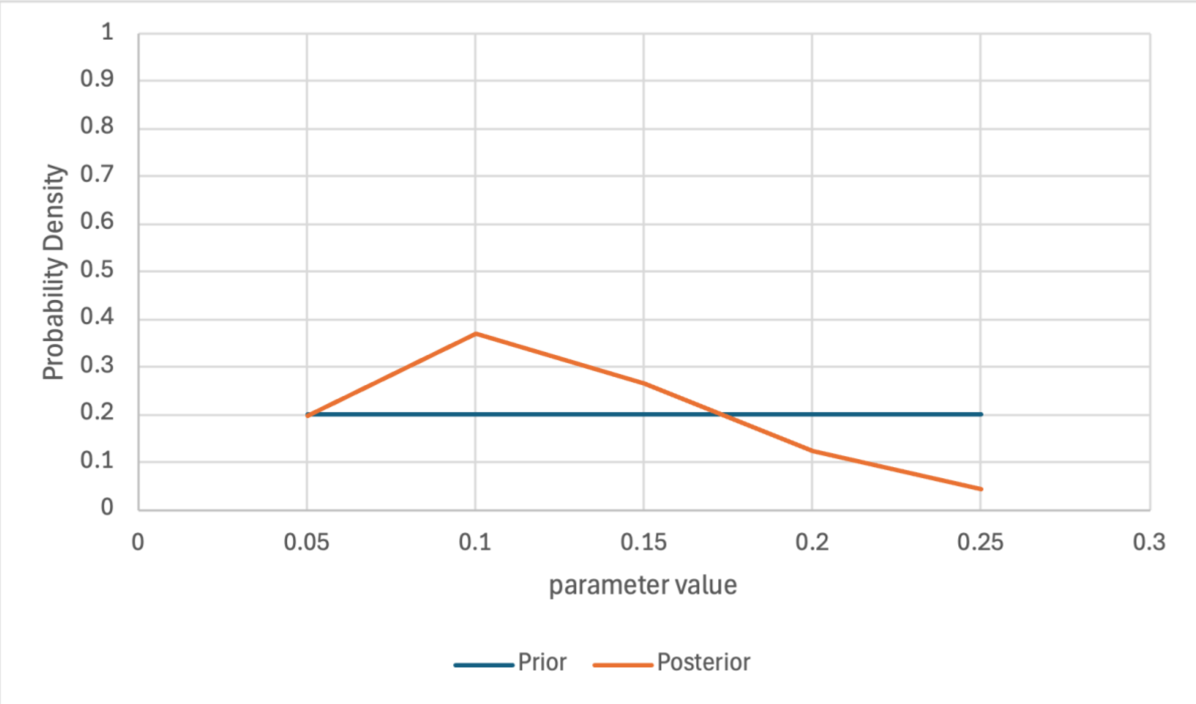
Table 2: Calculation of posterior after 1 month

Decline Rate: p	Prior: Pr(p)	Likelihood: Pr(Y| p)	Prior * Likelihood: Pr(p) * Pr(Y|p)	Posterior: Pr(p|Y)
0.05	0.20	$\binom{30}{3} 0.05^3 (0.95)^{27} = 0.127$	$0.2 \times 0.127 = 0.0254$	$\frac{0.0254}{0.1278} = 0.199$
0.10	0.20	$\binom{30}{3} 0.1^3 (0.9)^{27} = 0.236$	$0.2 \times 0.236 = 0.0472$	$\frac{0.0472}{0.1278} = 0.37$

0.15	0.20	$\binom{30}{3}\ 0.15^3(0.85)^{27} = 0.17$	$0.2 \times 0.17 = 0.0341$	$\frac{0.0341}{0.1278} = 0.267$
0.20	0.20	$\binom{30}{3}\ 0.2^3(0.8)^{27} = 0.079$	$0.2 \times 0.079 = 0.0157$	$\frac{0.0157}{0.1278} = 0.123$
0.25	0.20	$\binom{30}{3}\ 0.25^3(0.75)^{27} = 0.027$	$0.2 \times 0.027 = 0.0054$	$\frac{0.0054}{0.1278} = 0.042$
Total	**1.00**	**0.6388**	**0.1278**	**1.000**

The graph below shows the prior (dotted blue line), posterior (black line) and actual decline rate for the period (red line) by the expected decline rate. The key thing to notice is that the posterior's density has shifted towards the actual decline rate. This concentration towards the actual decline rate will grow stronger as more data is added to the model:

Figure 3: Posterior after 1 month (N = 30, declines = 3)



We now can use the posterior distribution to perform the valuation; the expected decline rate is now calculated as:

$$E[p] = (0.05 \times 0.199) + (0.1 \times 0.370) + (0.15 \times 0.267) + (0.2 \times 0.123) + (0.25 \times 0.042) = 12.2\%$$

Valuation after month 2

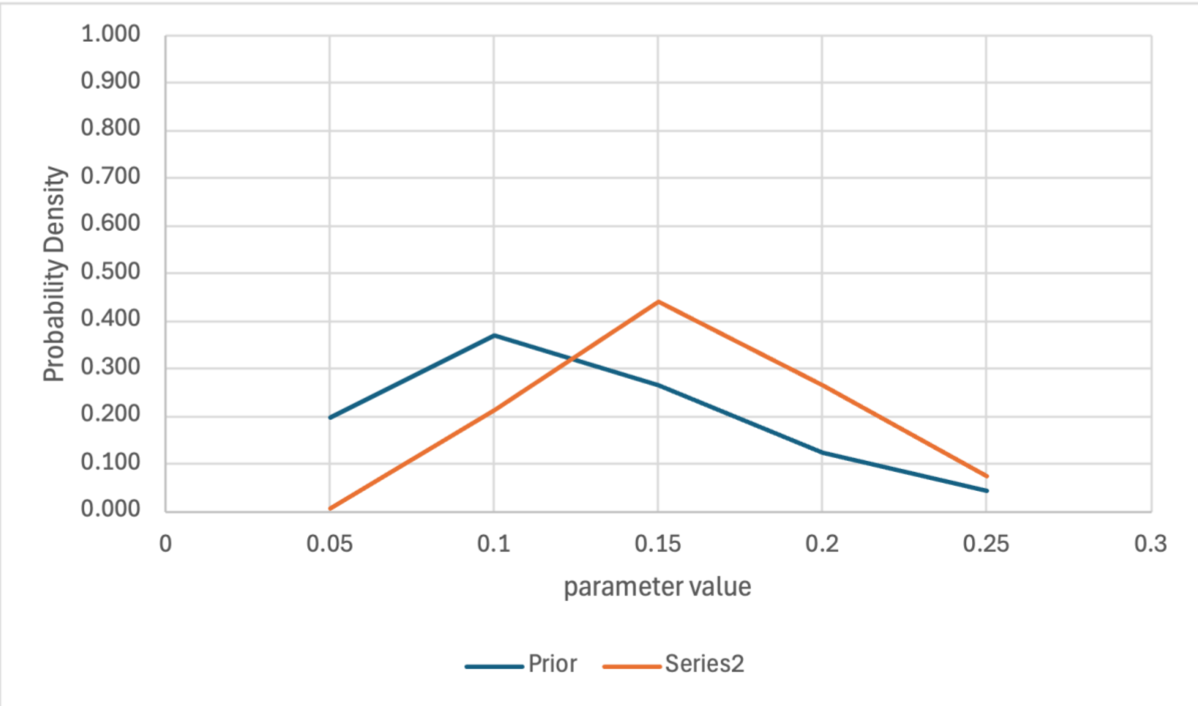
In the second valuation period, 6 out of the 30 claims reported are declined also we update our prior with the previously calculated posterior to use the information we learnt in the previous valuation. As we proceed to later valuation dates the prior will be updated to the previously calculated posterior, this is Bayesian updating. The calculations are similar to the previous valuation:

Table 3: Calculation of posterior after 2 months

Decline Rate: p	Prior: Pr(p)	Likelihood: Pr(Y| p)	Prior * Likelihood: Pr(p) * Pr(Y|p)	Posterior: Pr(p|Y)
0.05	0.199	$\binom{30}{6}\ 0.05^6(0.95)^{24}\ =\ 0.003$	$0.199 \times 0.003\ = 0.0005$	$\frac{0.0005}{0.0828} = 0.007$
0.10	0.370	$\binom{30}{6}\ 0.1^6(0.9)^{24}\ =\ 0.047$	$0.37 \times 0.047\ = 0.0175$	$\frac{0.0175}{0.0828} = 0.212$
0.15	0.267	$\binom{30}{6}\ 0.15^6(0.85)^{24}\ =\ 0.137$	$0.267 \times 0.137\ = 0.0365$	$\frac{0.0365}{0.0828} = 0.441$
0.20	0.123	$\binom{30}{6}\ 0.2^6(0.8)^{24}\ =\ 0.179$	$0.123 \times 0.179\ = 0.0221$	$\frac{0.0221}{0.0828} = 0.267$
0.25	0.042	$\binom{30}{6}\ 0.25^6(0.75)^{24}\ =\ 0.145$	$0.042 \times 0.145\ = 0.0061$	$\frac{0.0061}{0.0828} = 0.074$
Total	1.000	0.512	0.0828	1.000

The graph below shows the prior (dotted blue line), posterior (black line) and actual decline rate for the period (red line) by the expected decline rate. We can see again the posterior's density has shifted again towards the actual decline rate:

Figure 4: Posterior after 2 months (Y = 6, N = 30)

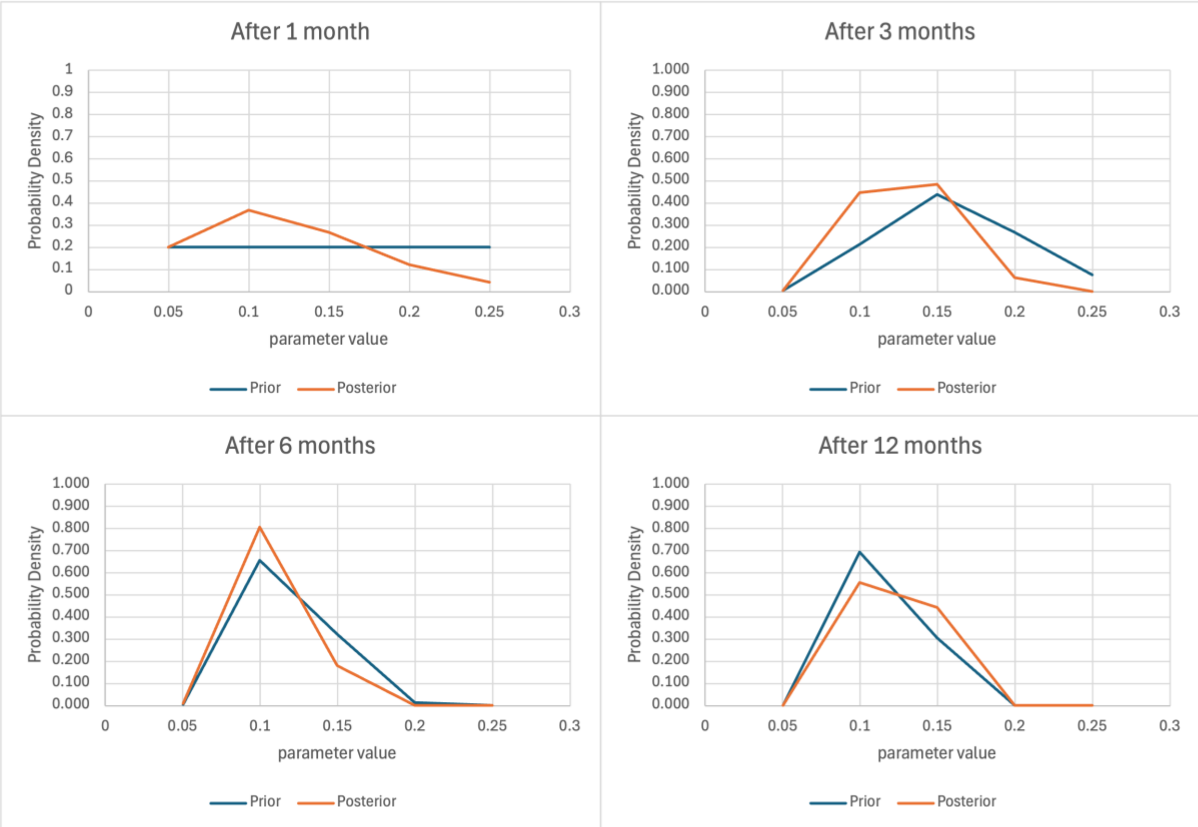


For this valuation period, the expected decline rate is now **16.0%.**

Valuation up to month 12

Using the same process we can keep updating our prior and posterior distributions for month 3 to 12. The following figure shows the impact of Bayesian updating after 12 months.

Figure 5: Bayesian updating outcomes after 1,3,6 and 12 months



The figure above shows that as more data is fitted to the model, the initial prior will be overwhelmed by the data. In this example, this results in the posterior converging to the true parameter value 10% that was used to generate the data. This has a very different shape to the prior distribution that the model started with. The other key observation is the posterior becomes less sensitive to new observations in each period. We can start to see this with iterations 6 and 12 where the posterior only changes slightly with the new data point.

The procedure presented in this example above is simple and analytically tractable for two reasons:

1. The only parameter used in the model is p, the probability of declining a claim. The calculation becomes more complicated when more parameters are introduced as the joint probability of the likelihood and prior distribution of all parameter values need to be integrated over.

2. The prior distribution used was discrete and bounded, and as a result the full posterior distribution could be easily calculated. When distributions are continuous this requires integrating over complex functions to find the normalising constant which is difficult to do analytically.

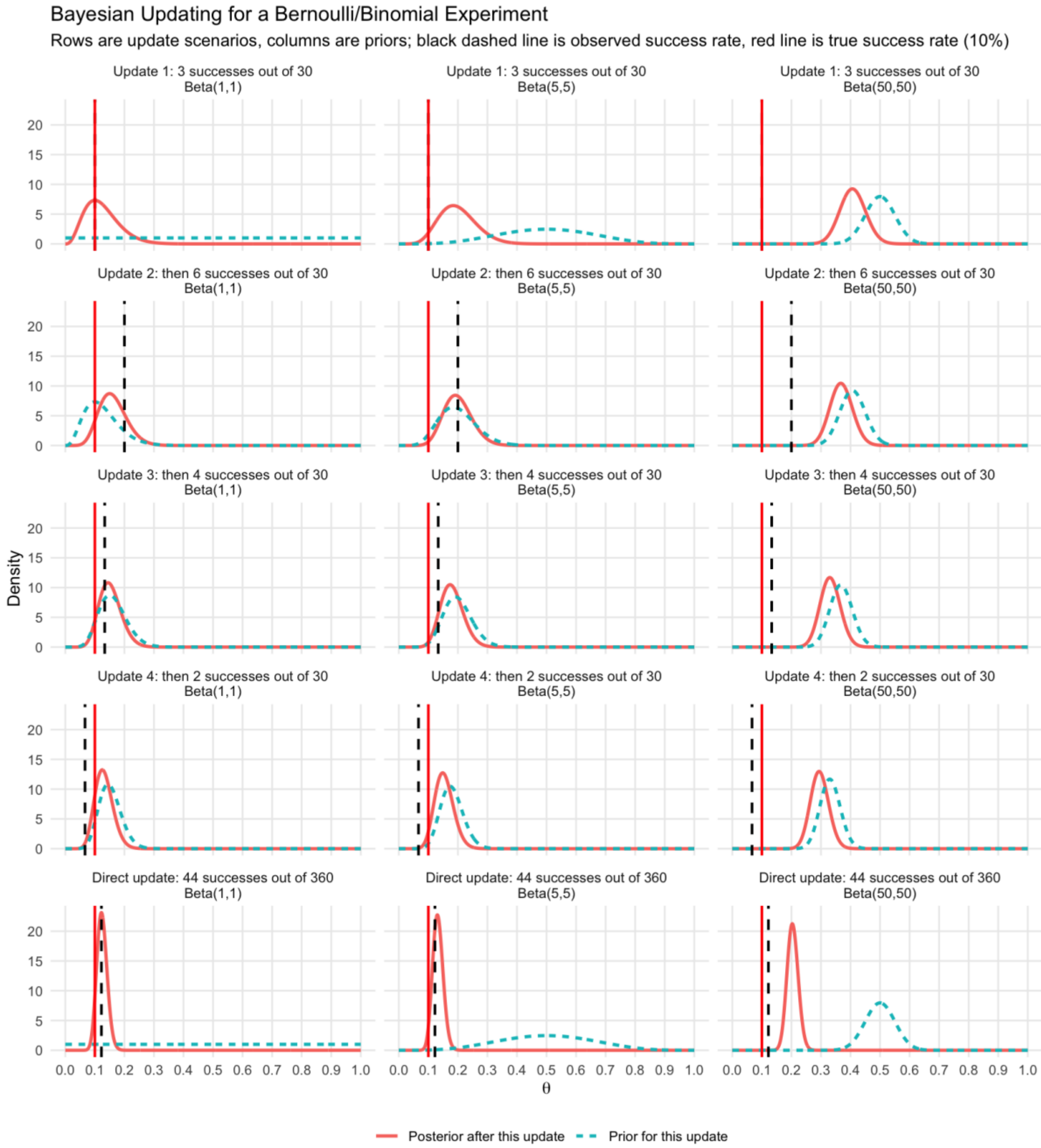
For continuous distributions we usually need to rely on estimation methods to retrieve the posterior distribution such as grid approximation, MCMC, Laplace approximation, or INLA. Although these methods produce estimates we can get arbitrarily close to the exact distribution by running them longer or using a finer grid.

The above example showed how Bayesian updating can work with discrete distributions. The following chart shows the results for the case where we are using a continuous distribution for the prior. We also use a prior centred around 0.5 rather than 0.15 to better show the mixing between the prior and the likelihood. Below we contrast using 3 different priors: a uniform prior (or a Beta(1,1) distribution), a Beta(5,5) distribution and a Beta(50,50) distribution. The figure below, using the same data as above, shows how the posterior distribution responds to the three different priors.

Uniform prior: This prior exerts relatively little influence over the posterior compared with the likelihood. As a result, when the mean posterior is calculated it is almost solely calculated from the data

Beta(5,5): This prior still supports a wide range for the parameter, p that we are estimating. Although it has a mean of 0.5 it is still a fairly wide distribution and heavily influenced by the data

Beta(50,50): This prior has a concentrated distribution around a mean of 0.5. This prior has a strong influence on the posterior and requires a lot more data before the posterior distribution starts getting close to the true value of p.



There is a mixing between data and the prior within the Bayesian model. Generally, posteriors with priors that are less dispersed respond less to data - we can take advantage of this feature to regularise parameter estimates to prevent overfitting. The above figures demonstrate this mixing; the more dispersed prior, in the left-hand column, responds to the earlier and later updates similarly, however, the more concentrated prior, in the right-hand column, responds less to the earlier updates and more the later updates.

8. Markov Chain Monte Carlo (MCMC)

The way that most Bayesian models are fitted today is via Markov Chain Monte Carlo (MCMC) methods. These methods can be broken down into three main steps:

- Specifying the joint distribution which combines the likelihood and all of the prior distributions.
- Applying an MCMC algorithm to generate samples from the posterior distribution.
- Extracting those samples to estimate items of interest.

You can think of MCMC a guided random walk designed to visit different parts of the posterior distribution with the right frequency to represent the distribution. If we run the MCMC algorithm long enough then we can build up a good picture of what the distribution looks like.

Most of what happens in the MCMC algorithm will be hidden away by the Probabilistic Programming language that is used. The most important part of the process that you will need to know are:

- The algorithm being used: some algorithms can be more efficient and reliable than other algorithms depending on the situation. Most algorithms can work with continuous distributions, while some that can also handle discrete distributions.
- The number of chains: algorithms are usually run more than once to be able to test for consistency across chains and convergence. The aim is that if we run the algorithm multiple times the results should have similar outcomes.
- Initial parameter values: all the algorithms need starting values for the parameters. In most MCMC implementations starting values are chosen randomly unless the modeller enters them themselves.
- Warmup period: Depending on how the algorithm is initiated it may take some time for the algorithm to find the area of highest posterior density. Samples generated during this period are usually discarded so that they do not distort the results. The warmup period needs to be long enough for the algorithm to settle into the main region of the posterior distribution.
- Sampling period: this is the period after warmup and contains the samples that are used to represent the posterior distribution. As this is a random process the longer the sampling period the more reliably it will represent the posterior distributions.

These are the main setting for MCMC process. Each algorithm will also have additional tuning parameters, however, most MCMC implementations will have reasonable defaults set that can usually be left as they are unless there are issues with running the algorithm.

9. Model Checking

Checking a Bayesian model can be a complex undertaking depending on the complexity of the model itself. The following sets out some basic checks that should be made and provides references for more detailed checks. Suggestions are also included on how to handle situations where you find the model is not working. The following are based on our experience and also (Gelman, et al., 2020).

The most commonly used method for estimating the posterior distribution in a Bayesian model is MCMC. As discussed in a previous section, this method involves running a series of simulations to efficiently explore a posterior distribution. MCMC simulations are initialised at random points in the distribution, as a result multiple iterations of the simulation, known as chains, are advised to ensure that sampler's results are not biased by the initialisation. There are two main issues to look out for when running this estimation method:

1. Convergence and chains mixing: Convergence means that the MCMC algorithm has been running long enough for the samples to be a reliable representation of the posterior distribution. After the warmup phase, it is expected that all chains converge to the same posterior distribution. If chains do not agree on the posterior distribution, i.e. the chains are not mixed, it suggests that one or more chains are getting stuck in certain areas of the posterior distribution.

2. Divergent transitions: This is only relevant to some MCMC algorithms (HMC and NUTS being the main ones). Too many divergent transitions suggest that the posterior distribution has a difficult shape to be easy to explore.

Most PPLs you use will have available a series of outputs to be able to check for convergence. Most PPLs will provide warnings if any of the diagnostics indicate there may be a problem with convergence. Below we cover some of the main checks. There are other checks that have not been covered (e.g. divergences, tree-depth, energy etc.).

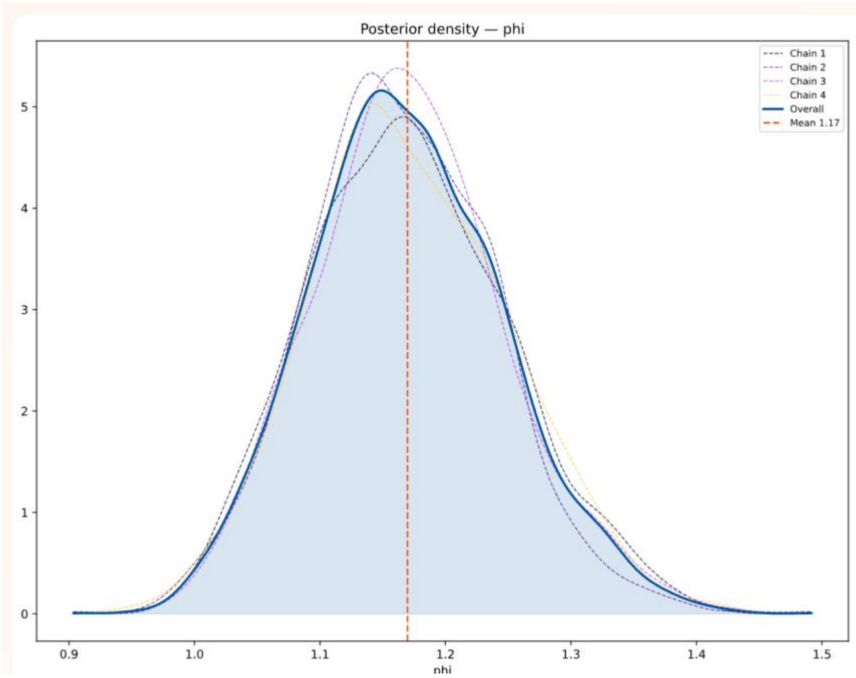
When fitting the model, the sampler in Stan and PyMC will automatically notify the user that something has gone wrong.

Two easy visual checks for convergence are to look at:
- density plots for each chain.
- trace plots for each chain.

Density plots

A density plot shows the marginal distribution for each parameter in the model. For diagnostic purposes we are interested in the plots that show the marginal density for each chain that has been run. An example of this is in Figure 6 below.

Figure 6: Example of a posterior density for a single parameter



If the model has converged, you are expecting to that the distribution for all the chains look similar. If the chains look different then that is a sign that the model has not converged.

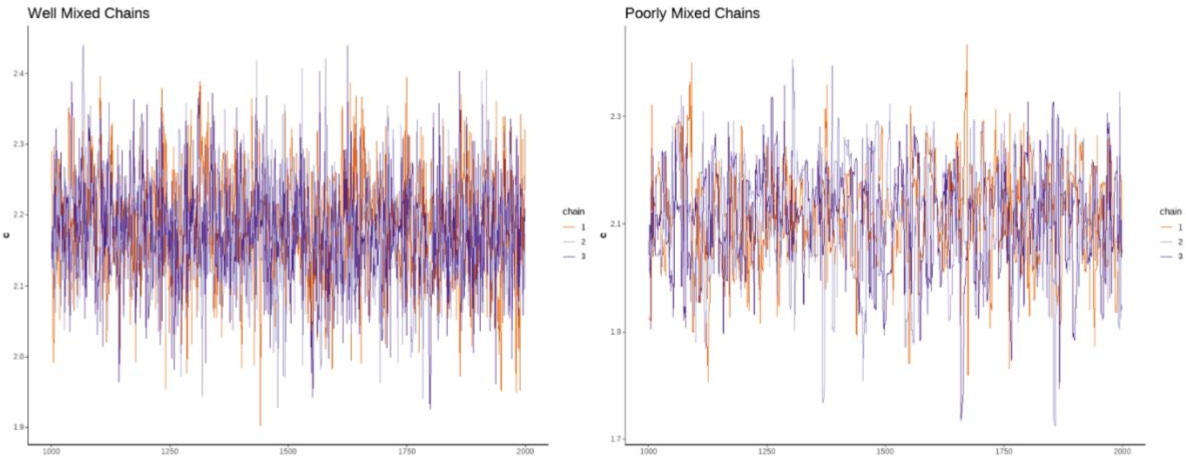
Trace plots

The most straightforward way to check for any degenerate chains is to plot the parameter values that each chain explores on a single plot, this is known as a traceplot. For unimodal distributions they should all be centred on the mean/mode with no persistent drifts away from the centre of the distribution. The left-hand side of Figure 7. below shows a well-mixed trace plot. Traceplots are useful in identifying sampling inefficiencies visually as highly autocorrelated samples can be spotted easily, however this becomes impractical when dealing with models with many parameters which is quite common in Bayesian modelling. Typically relying on sampling metrics such as R-hat and effective sample size is more efficient.

There are two key features of ideal traceplots:

1. Chains should look quite random and move between two extremes of the posterior distribution.
2. The chains should overlap, if they do not, it means that the chains have not mixed.

Figure 7 – Traceplot comparison between well mixed chains vs poorly mixed chains



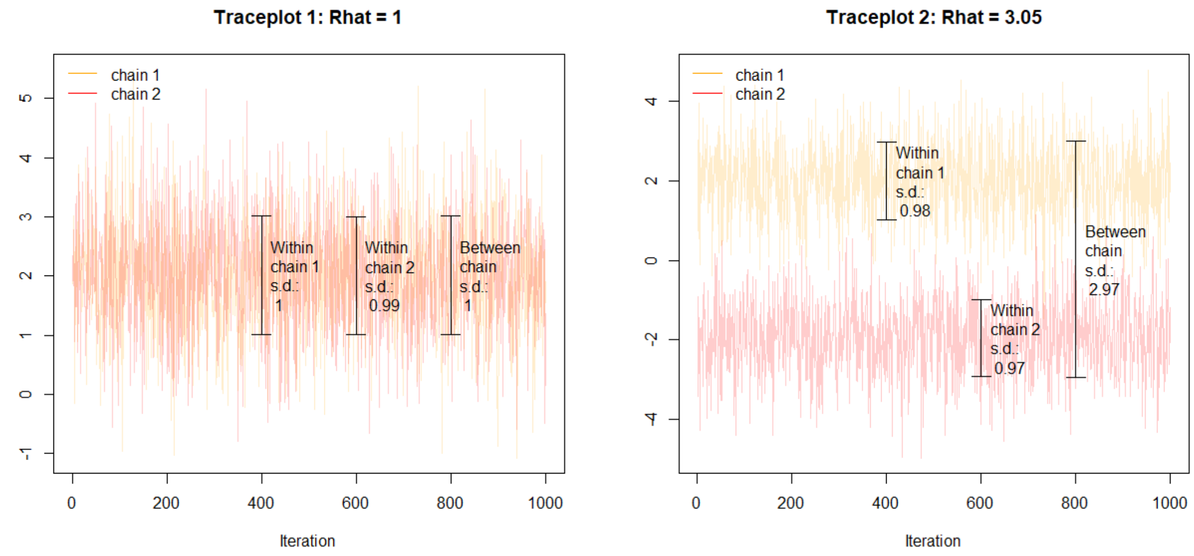
R-hat

R-hat is a measurement of convergence, its purpose is similar to both the trace and density plots and is a statistical check that the chains come from the same distribution and are well mixed. There will be an R-hat for each parameter and ideally all of these will be < 1.01. The calculation of R-hat is a statistically sound way to check that the posterior distribution of all the chains is the same.

R-hat compares the variance of all the simulated values between chains to the variance of the simulated values within each chain. The idea is that variance across all chains should be similar to the variance of individual chains if the simulation has converged. Ideally R-hat should be below 1.01 (Vehtari, Gelman, Simpson, Bob, & Bürkner, 2021).

The figure below shows two simulations, one that converged and one that did not, and their respective R-hat metrics:

Figure 8 - Traceplot comparison between well mixed chains vs poorly mixed chains with R-hat measures



The description of R-hat above is simplified to convey the intention of the measure; in practice there are several ways that R-hat can be calculated (Vehtari, Gelman, Simpson, Bob, & Bürkner, 2021). A short summary of each method is found below:

- Split R-hat: This measure is calculated using the same formulation as R-hat except each chain is split in half and treated a new chain effectively doubling the number of chains and halving the number of iterations using the calculation. This measurement fails to detect issues when chains have the same mean but different variances or chains have infinite variances and different scales. The other two R-hat measures were designed to overcome these issues.
- Rank normalized split R-hat: This measure first ranks all iterations across all chains and normalises these ranks, the measure then calculates split R-hat with these normalised values. This measurement fails to detect when chains have the same location but different scales - this can occur if one chain gets stuck sampling the central portion of the posterior distribution.
- Rank normalized folded split R-hat: This measure transforms each iteration into “folded” iterations calculated as the absolute deviation from the median of all draws. These folded iterations are then passed through the same procedure as the rank normalised split R-hat procedure.

The current recommendation is to calculate both the rank normalised split R-hat and rank normalised folded split R-hat metrics and take the maximum.

Effective Sample Size (ESS)

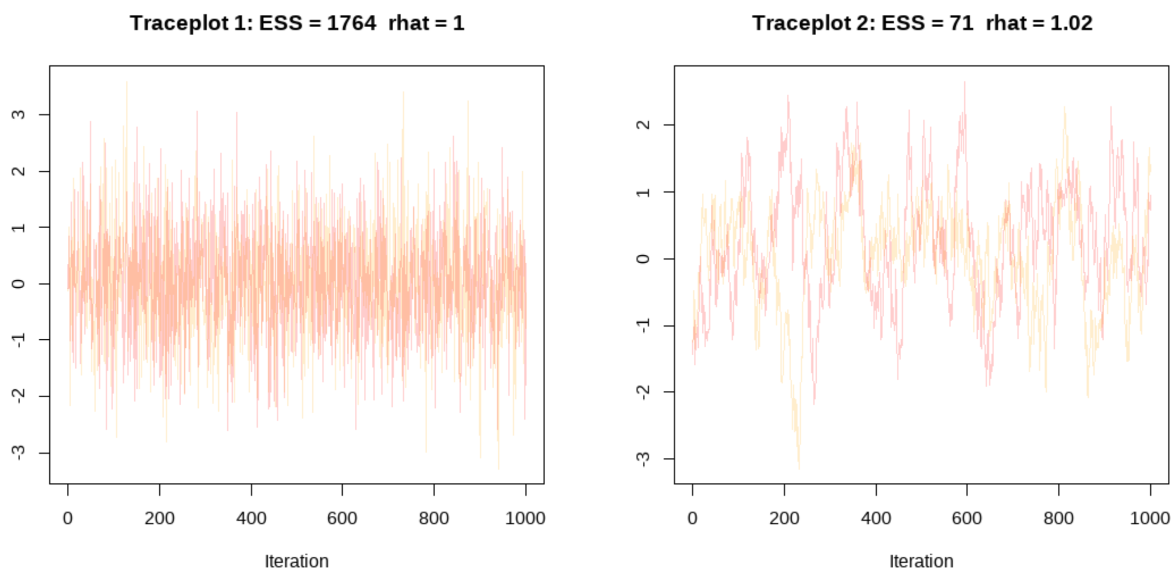
As discussed above, MCMC can be described as a guided random walk, this means that each iteration of this random walk contains some information that it learnt in the previous step – this is autocorrelation. If there is too much overlapping information in results our parameter estimates are less reliable.

Given that each step will contain some level of information from the previous step means that the overall procedure has learnt less information in total compared to if this information was independently drawn. We capture this as a measurement called Effective Sample Size (ESS).

ESS is trying to measure the reliability of the simulation. There are two ESS measurements that are commonly used: ESS-Bulk and ESS-Tail, these measure the reliability of the sampled central portion and tails of the posterior respectively. The rule of thumb is ESS bulk and ESS tail values should be greater than 100 times the number of chains run in the simulation (Vehtari, Gelman, Simpson, Bob, & Bürkner, 2021), for example if 4 chains are run, ESS-bulk and ESS-tail should be greater than 400.

The figure below shows an example of two traceplots of two model runs, one with a high ESS (high reliability) and once with low ESS (low reliability). The figure shows that the R-hat is within a reasonable range however the ESS reveals that there is a severe issue for the right-hand side plot. This shows that ESS and R-hat measure two different things and must be used in combination when running diagnostics on MCMC results.

Figure 9 - Traceplot comparison between well sampled and poorly sampled chains with ESS and R-hat measures



There are further diagnostics not covered due to their more technical nature, for example, divergences, tree depth, BFMI/energy. Most PPLs will alert you if these diagnostics are an issue so that you can investigate and check the model further.

Troubleshooting

Where you have concerns that your model has not converged the following are worth considering:

1. Increase the number of samples. Sometimes the problem is that you have not run your model long enough. This is often the case if there are strong correlations between parameters.

2. Run your model with a range of synthetic data that you know is consistent with the formulation of your model. Your model should then be able to re-discover the parameters you used to generate the synthetic data.

3. If you have a complex model, it is usually best to start with a simplified version and work up from there. Issues should be easier to identify and resolve with a simplified model.

4. Revisit your priors. Using tighter priors initially may help with convergence. You can then look at weakening them.

5. Use better initial values for your parameters. If you have used initial parameters for your model that are not near the centre of the posterior distribution the model may be having difficulty finding its way to the centre of the posterior distribution.

6. Look at the parameterisation of your model. There could be several issues with the parameterisation of your model often related to the geometry of the posterior distribution. A few are:
   - Identifiability: For some models identifiability could be an issue. Identifiability is where there is more than one set of parameters that is exactly equivalent. To fix this you need to add an identifiability constraint to define a unique solution.
   - Geometry of the posterior distribution: Some geometries can cause difficulty with the algorithm and calculations. There can be any number of reasons for this including having highly correlated parameters and parameters close to 0. Solutions often relate to finding different parameterisations of your model, for example: normalising the parameter, moving to a log scale or using centred or non-centred parameterisations.

7. Look at normalising your data. Like looking at the parameterisation, you should also look at your data. The solution here might be to normalising your data or moving it to a log scale.

More detailed guidance on validating the model and addressing computational issues it is recommended that the reader start with Gelman (2020)

10. Example 2 – Simple linear regression

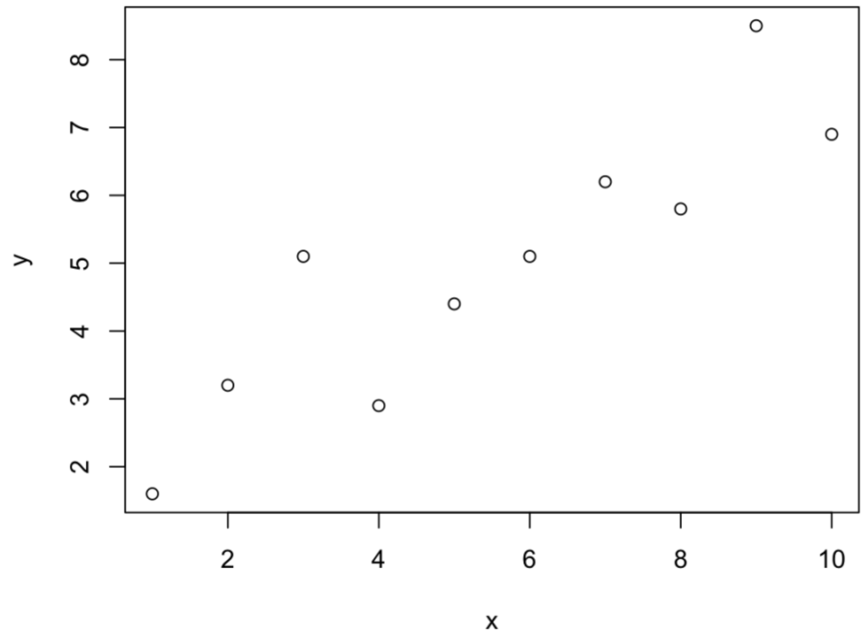
To illustrate the basics, we will start with a simple linear regression (many introductions to Bayesian statistics start with a coin toss example i.e. a series of Bernoulli trials, however, I have found that a linear regression is equally intuitive). We will follow these steps:

1. Specify the model.
2. Write the model in Stan language.
3. Run the Stan model.
4. Evaluate the model output.

Data

We have generated 10 data points to fit a linear model to:

Figure 10: Plot of data for linear regression



Step 1: Specify the model

Consider a simple linear regression with a single independent variable with a normally distributed error term as follows:

$$y_i = a + b \cdot x_i + \epsilon_i$$

Where the error term is distributed as: $\epsilon_i \sim normal(0, \sigma)$

This can also be written as:

$$Pr(y_i | x_i, a, b, \sigma) \sim normal(a + b \cdot x_i, \sigma)$$

We need two components for a Bayesian model, the likelihood function and prior distributions for each parameter. We have a likelihood function:

$$Likelihood \ = \ Pr(y_i | x_i, a, b, \sigma) = \frac{1}{\sqrt{2\pi} \cdot \sigma} exp(-\frac{1}{2}\left(\frac{y_i - (a + b \cdot x_i)}{\sigma}\right)^2)$$

Where the right-hand side of the equation is the equation for a normal or gaussian distribution

For the priors, for the sake of this example we specify wide priors. Given that a and b are parameters of a linear regression, they can be either positive or negative, as such a normal distribution was selected as the prior for these parameters. The σ parameter can only be strictly positive, as such a half-normal distribution was selected as σ prior distribution.

The priors selected are summarised below:

$$a \sim normal(0, 100)$$
$$b \sim normal(0, 100)$$
$$\sigma \sim halfnormal(0, 5)$$

The Bayesian model is specified as:

Priors

$$a \sim normal(0, 100)$$
$$b \sim normal(0, 100)$$
$$\sigma \sim halfnormal(0, 5)$$

Likelihood

$$y_i \sim normal(a + b \cdot x_i, \sigma)$$

Step 2: Write the model in Stan language

Using a PPL like Stan we can implement the above model as follows:

```
data {
  int<lower=0> N; // Number of data points
  vector[N] x;  // Covariate xi
  vector[N] y; // Dependant variable yi
}

parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;  // restricted to > 0 because it is a standard deviation
}

model {
  // priors
  alpha ~ normal(0, 100);
  beta ~ normal(0, 100);
  sigma ~ normal(0, 5); // because sigma is restricted to >0 sigma is effectively half-normal

  // likelihood
  y ~ normal(alpha + beta * x, sigma);
}
```

The structure of a Stan program logically separates out the data, the parameters and the model. A more detailed description of these code blocks can be found in the section Probabilistic Programming and Stan.

Step 3: Run the Stan model

While Stan is the PPL we are using it is implemented to be run within more familiar environments like R, Python, Julia etc. Tasks such as data preparation, exploratory data analysis, model evaluation are usually performed in these other environments while Stan is the engine that performs the Bayesian inference and sampling.

The following code block prepares the data and runs the Stan model created in the previous step from R. The code below has been kept as simple as possible with minimal optional arguments:

```
library(cmdstanr)

# Define the data
stan_data  <-  list(
 N = 10,
 x = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
 y = c(1.6, 3.2, 5.1, 2.9, 4.4, 5.1, 6.2, 5.8, 8.5, 6.9))

# Compile the Stan model
compiled_model <- cmdstan_model(stan_file = “linear-regression.stan“)
```

```
# Run the Stan model using the data
fitted_model <- compiled_model$sample(
            data = stan_data,
            iter_warmup = 1000,
            iter_sampling = 10000,
            chains = 4)
```

Step 4: Evaluate the model output

Once the model has been fitted, we can extract various elements from the fitted model such as posterior distributions and diagnostics.

The overall model diagnostics show that convergence has occurred as:

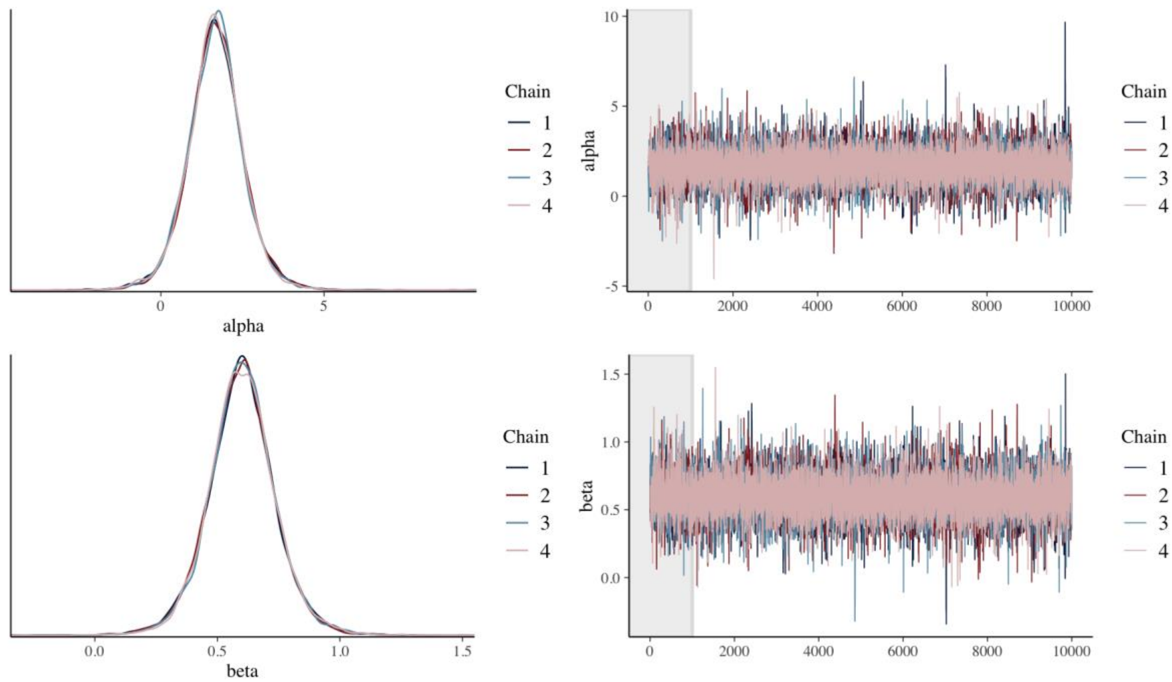
- R-hat for all parameters below 1.01, this means that the 4 chains run in the model have converged.
- ess_bulk and ess_tail are above 1,000, this means the quantities such as mean, median and tail values are trustworthy.

Table 4: Results from fitted Bayesian model

variable	mean	median	sd	mad	q5	q95	r-hat	ess_bulk	ess_tail
lp_	-5.73	-5.33	1.51	1.21	-8.68	-4.12	1.00	9820	10753
alpha	1.67	1.67	0.87	0.76	0.26	3.07	1.00	11259	11125
beta	0.60	0.60	0.14	0.12	0.38	0.82	1.00	11560	11540
sigma	1.20	1.12	0.39	0.30	0.75	1.92	1.00	11243	12158

Plotting density and traceplots

Figure 11: Density and trace plots for slope and intercept of fitted linear regression



The R-hat, ESS, density plot and trace plot above all show that the results have converged well. As this is a simple example we would expect to have no worries about the model converging.

Plotting the fitted line

Figure 12: Line of best fit, 95% credible interval and 95% posterior predictive interval

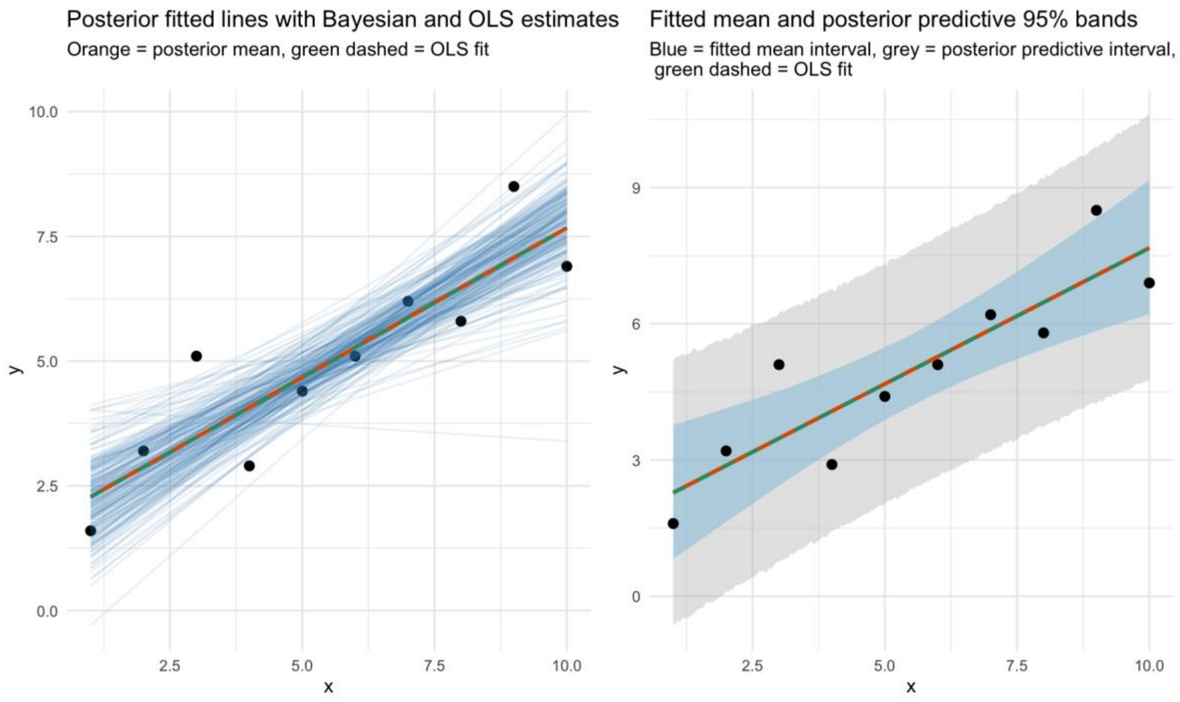


Figure 12 above also includes the fitted line from using Ordinary Least Squares to show that the fit is materially the same.

11. Case Study 1 – Termination rates

This case study goes through a simplified scenario of setting a termination rate basis for a disability income product where we have limited experience. This case study shows:

- How a termination rate (or other basis) can be set by blending a starting assumption basis (used as a prior) and experience.
- An example of prior predictive checks.
- An example of posterior predictive checks.

Updating an assumption basis with limited experience can be one of the more challenging tasks for a pricing/valuation actuary as it often relies upon significant judgement or rules of thumb to know how much credibility to give the experience of the portfolio. The aim of using Bayesian modelling is to make this process as explicit as possible and being clearer about when judgement is applied.

The scenario below has been simplified to only consider a single termination curve for terminations up to twelve months in duration.

Consider the following disability income termination experience:

Table 5

Duration (months)	Starting Assumption Basis	Exposure in months	Terminations	Crude Rate	A/E
1	0.120	220	32	0.159	133%
2	0.105	210	25	0.129	122%
3	0.092	195	18	0.103	111%
4	0.081	180	13	0.078	96%
5	0.072	160	10	0.069	95%
6	0.064	140	8	0.057	89%
7	0.057	110	4	0.045	80%
8	0.051	85	4	0.047	92%
9	0.046	60	1	0.017	36%
10	0.042	40	2	0.050	119%
11	0.038	25	0	0.000	0%
12	0.035	15	1	0.067	190%

When a basis is set the starting assumption basis would often be one of, or a combination of:

- A standard industry table with potentially adjustments for the portfolio.
- The insurers own experience for similar portfolios.
- A basis provided by reinsurers or created with input by the reinsurer.

The pattern above is typical where there is little exposure in later durations with some cells (e.g. month 11) having 0 terminations.

Given the data above we want to set a basis for our disability product. We can see early on that the experience is higher or better than the basis, however, after the first few months the experience is generally lower or worse. Given this it can be hard to know how to set the basis.

Model

In this scenario we will use a simple Bayesian model to blend the starting basis with the observed experience, with greater credibility being assigned where experience is more credible.

The Bayesian model is specified as:

Priors

$$\alpha \sim normal(0, 0.3)$$
$$\sigma \sim halfnormal(0, 0.1)$$
$$g_1 \sim normal(0, 0.3)$$
$$g_2 \sim normal(g_1, 0.1)$$
$$g_d \sim normal(2g_{d-1} - g_{d-2}, \sigma)\ for\ d \geq 3$$

Where d is the claim duration.

The priors have been set to control how smooth the updated basis is and how close or far away it can be from the starting pricing basis. Priors are discussed further below.

Likelihood

$$\eta_d = logit\left(\mu_d^{Basis}\right) + \alpha + g_d$$

$$terminations_d \sim binomial(E_d, logit^{-1}(\eta_d))$$

Where μ_d^{Basis} is the starting assumption basis

The Stan program below implements one method to easily create a new basis. The model assumes that the new basis is equal to the starting basis times an adjustment *g[d]* for each duration. The model then restricts the adjustments *g[d]* to be a second order random walk from one duration to the next – this is how the model imposes smoothness on the adjustments and also pools weight across nearby durations as well.

The input into the model is straightforward where we input exposure, number of observed terminations and the termination rates from our basis.

The parameters we are going to find are parameters for smoothness (σ), an overall shift up or down to the termination basis (α) and the adjustments to get to the new basis (g_i)

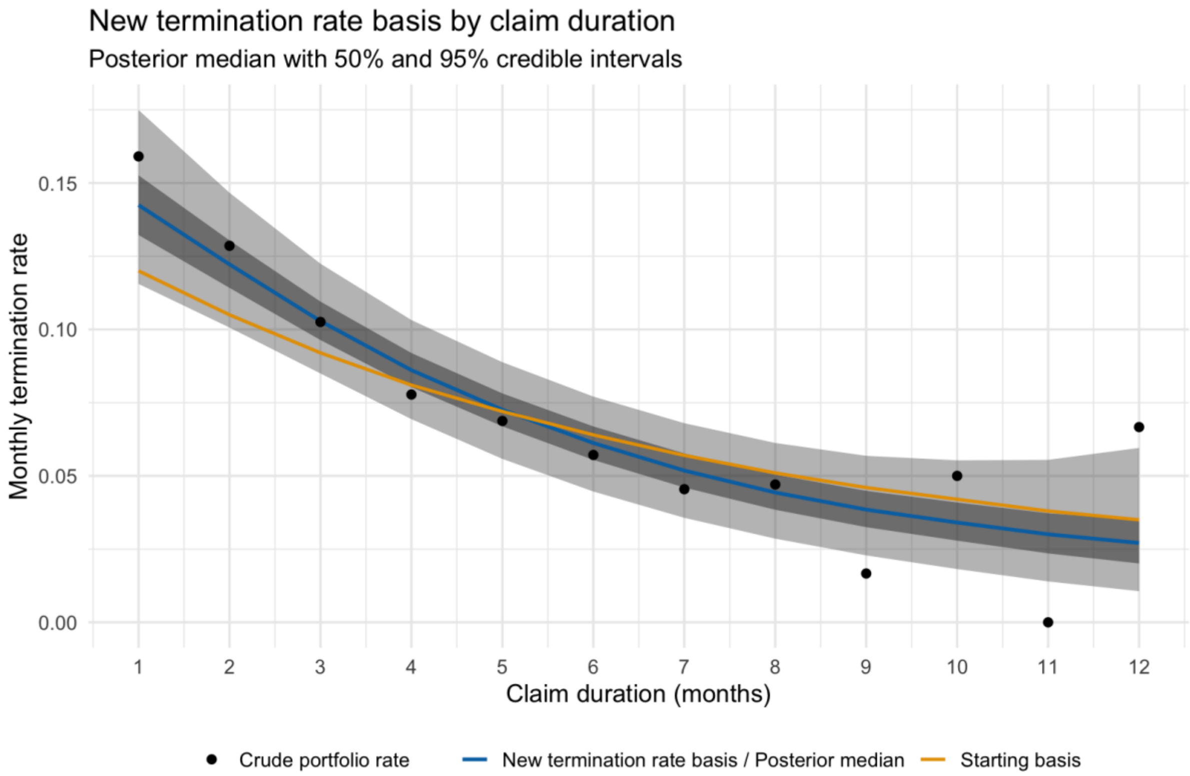
To keep the curve smooth the sequence of adjustments to the termination curve (g_i) are restricted to follow a second order random walk via the term $g_d \sim normal(2g_{d-1} - g_{d-2}, \sigma)$.

The last term in the model codes the likelihood with the number of terminations in each duration band being drawn from a binomial distribution with mean equal to exposure x new updated basis. A logit transformation has been used to keep values between 0 and 1.

Results

Running the data through the model we get the output in Figure 13:

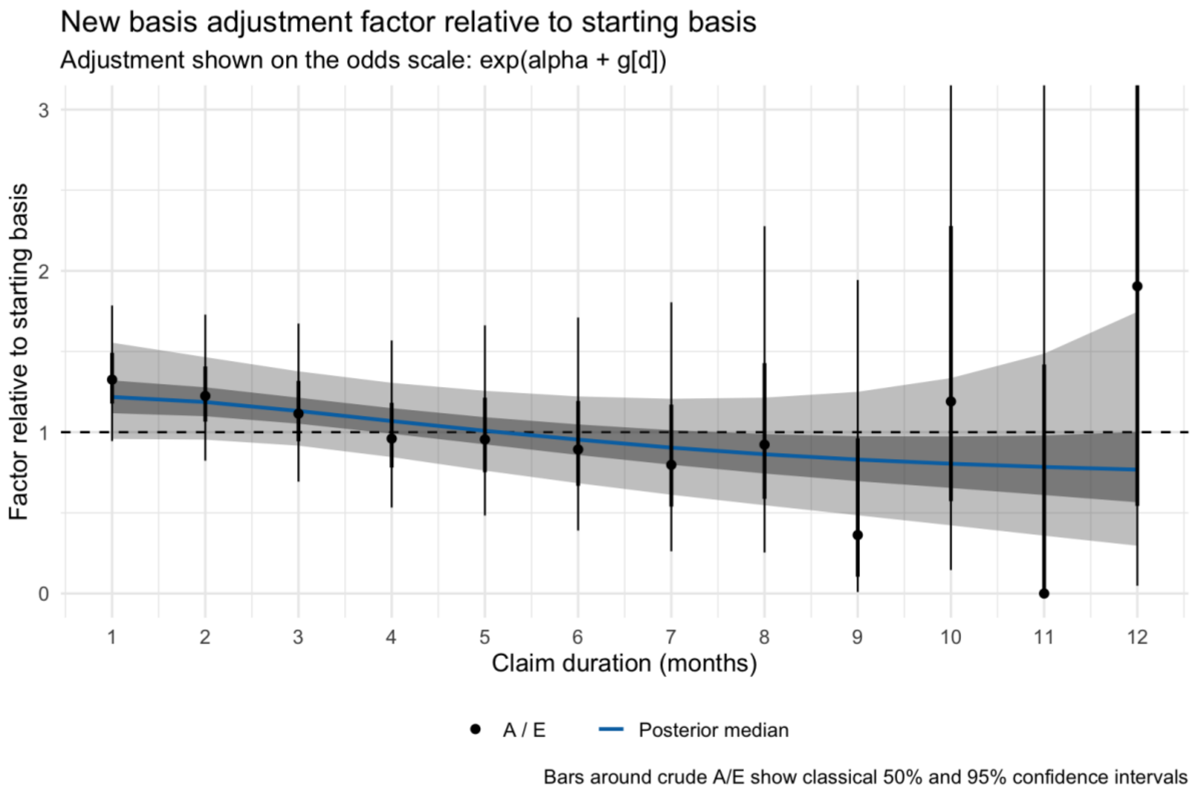
Figure 13: Fitted termination rate basis



In Figure 13 above:

- The orange line shows the starting termination rate basis.
- The blue line shows the new termination rate basis.
- The points show the actual crude termination rate.
- The two different ribbons show the credible intervals of our estimated termination rate basis. These are called credible intervals in Bayesian statistics (which are different to confidence intervals from frequentist statistics).

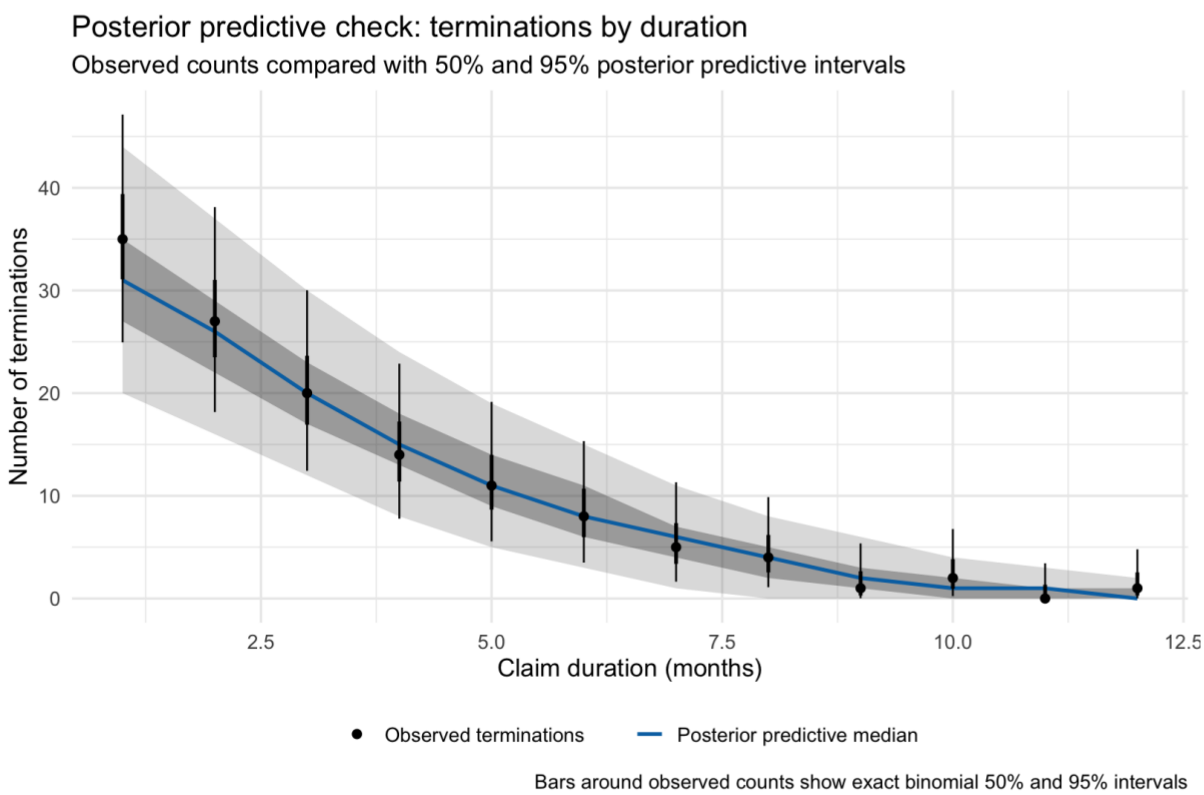
Figure 14: Fitted adjustment factors to starting basis



In Figure 14 above:

- The dashed line is at 100% of the starting termination rate basis.
- The blue line shows the adjustment to the starting basis to get the new termination rate basis.
- The two different ribbons show the 50% and 95% credible intervals for the new termination rate basis.
- The points show the actual versus expected on the starting termination rate basis.
- The vertical bars show the classical 50% and 95% confidence intervals around the actual experience.

Figure 15: Posterior predictive check



In Figure 15 above:

- The solid line shows how many claims we would expect on the new termination rate basis.
- The points show the actual claims terminated.
- The two different ribbons show the 50% and 95% Posterior Predictive intervals. These show, given our new termination rate basis what is the range of possible observations within a given probability. For a 95% Posterior Predictive interval we would expect that 95% of all observations would be with that interval.

Some advantages of the above process are that:

- Once it has been set up it can be easily updated in a mechanical manner with new data as it comes.
- We don't need to aggregate duration periods and end up with odd looking adjustments and sudden changes in termination rates.

The final numerical output looks like the following:

Table 6

Duration (months)	Starting Basis	Crude Rate	A/E	Fitted Adj	Lower bound of 50% credible interval	Upper bound of 50% credible interval
1	0.120	0.159	133%	122%	112%	132%
2	0.105	0.129	122%	119%	110%	128%
3	0.092	0.103	111%	113%	105%	121%
4	0.081	0.078	96%	107%	99%	115%
5	0.072	0.069	95%	101%	92%	109%
6	0.064	0.057	89%	95%	86%	105%
7	0.057	0.045	80%	90%	80%	101%
8	0.051	0.047	92%	86%	74%	99%
9	0.046	0.017	36%	83%	70%	97%
10	0.042	0.050	119%	80%	65%	97%
11	0.038	0.000	0%	78%	61%	98%
12	0.035	0.067	190%	77%	57%	101%

Where the fitted adjustment is used to derive the newly updated rates as follows:

New termination rates = starting basis x fitted adjustment.

Notes on setting priors:

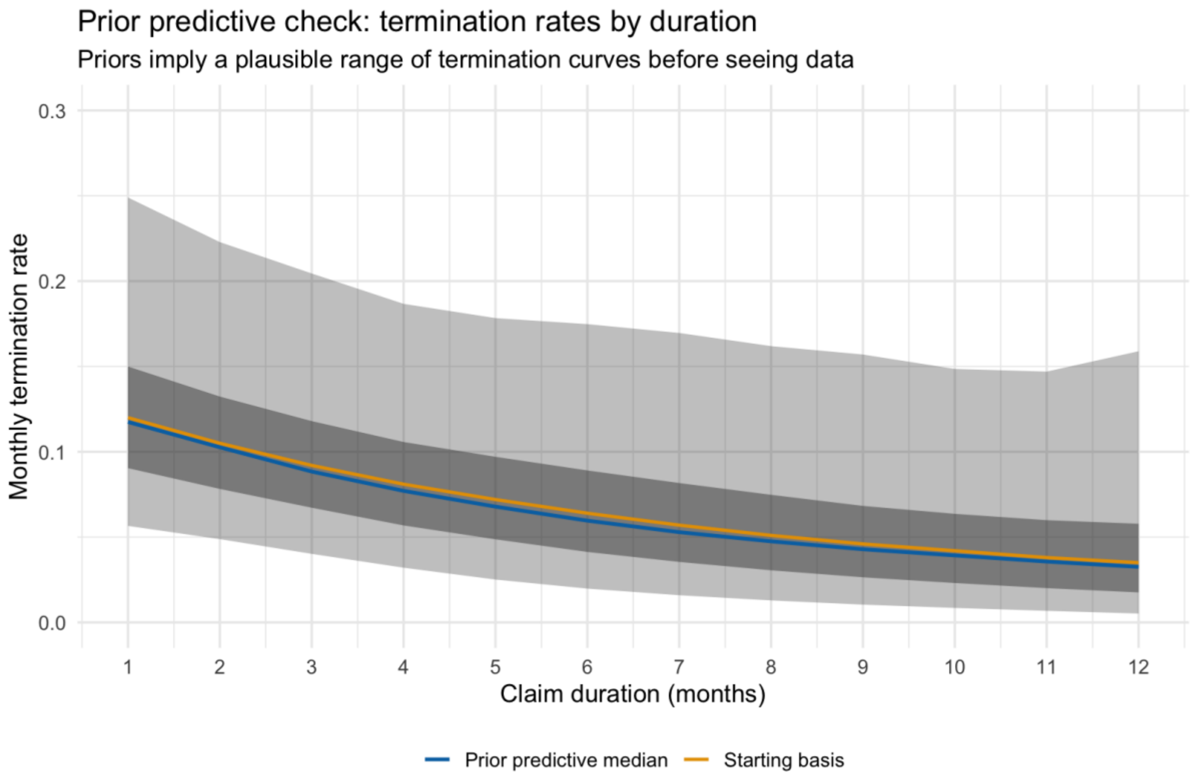
In deriving the new termination rates decisions still need to be made in setting what priors to use. Priors are unique to Bayesian modelling and are a way of providing additional information to the model in addition to the data itself. This information can be gleaned from other sets of data, wider sources of knowledge or even professional judgement.

The parameter alpha controls the overall shift up or down of the updated termination rate basis. The prior for alpha should be based upon our view of how far away we think the underlying experience of the portfolio could reasonably move away from our starting termination rate basis. The prior used above of normal(0, 0.3) implies we think the true underlying basis is reasonably likely to be within say ±35% (recalling that alpha is on a log scale). In practice we would use our knowledge of how different other portfolios experience can be; input from reinsurers combined with professional judgement.

The parameter sigma_g controls how smooth the curve should be. We should expect that final termination rates would be relatively smooth and therefore have a reasonable small sigma_g to ensure that they are smooth. A small sigma_g also means that durations close together pool their info and strongly inform each other.

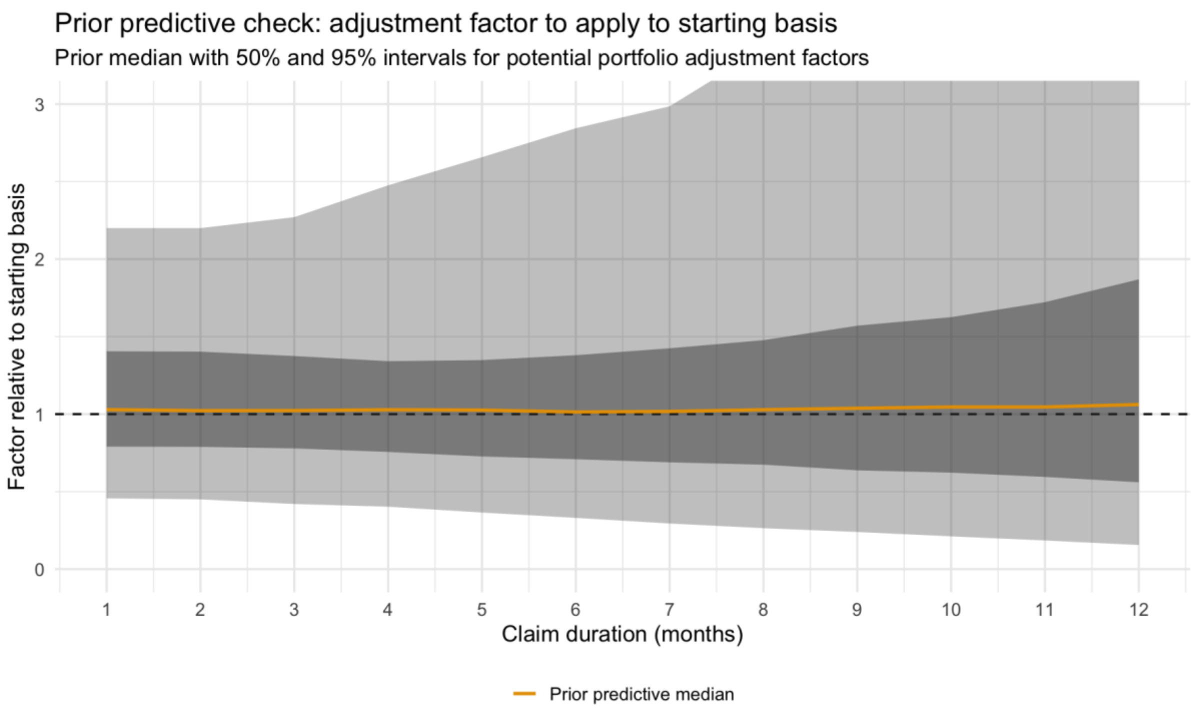
One way to check the suitability before running the model is to conduct prior predictive checks. The aim of this is to look at the range of possible outcomes just from the priors without any data. With this check we want to see that the outcomes are broad enough to cover the full range of possible values for the fitted model. Figure 16 and Figure 17 below show the range of values our prior is likely to take.

Figure 16: Prior predictive check on termination rates



The bands above show where the outcome would be either 50% or 95% of the time.

Figure 17: Prior predictive check on adjustment factors



The above charts show that the priors allow the model to cover a wide range of outcomes. If we thought that the outcome could be outside the ranges shown then we should widen or loosen our priors.

Once we have run our model other ways for ensuring suitable priors would be:

- Sensitivity testing different priors and judging visually which looks most appropriate.
- Usually, a real-world termination study will have more durations, features etc. In this situation it would make sense to use cross-validation (e.g. LOOCV) to optimise sigma_g.
- Use other diagnostics like residual plots, Q-Q plots, PIT histograms, etc.

Governance

As with any model it should have a robust process around it. A robust framework around this basis-setting process might include the following elements:

Model Validation

- A peer and technical review should be conducted on the model code to verify it correctly implements the intended methodology, including the second-order random walk, the binomial likelihood, and the prior specifications.

- Confirm that the model converges properly (e.g., R-hat statistics close to 1.0, sufficient effective sample sizes) and that diagnostics are reviewed each time it is run.

Experience Study Controls

- Define a clear, repeatable process for extracting the underlying exposure and termination data, with documented data definitions (e.g. what constitutes a termination event, ensuring consistent exposure measures are used between the starting basis, the experience study data and subsequent pricing/valuation models).

- Maintain an audit trail linking the data inputs used in each model run to the source systems.

- Document any adjustments or exclusions made to the raw data before it enters the model.

Review and Sign-Off

- Establish a formal review frequency, for example, annually or when a material volume of new experience has emerged (e.g. a 20% increase in exposure).

- The basis should be reviewed and signed off by a peer reviewer.

- The review should explicitly assess whether the credible intervals have widened or narrowed since the last update, and what that implies for uncertainty in the basis.

Change Management

- Upon the introduction of this new methodology there should be testing of using the previous methodology compared to the new methodology.

- Define a materiality threshold for when a change in fitted basis triggers a formal basis change versus simply being noted for monitoring. For example, a change of more than 5 percentage points in any duration's fitted adjustment might require a formal update.

- Any basis change should be accompanied by an impact assessment quantifying the effect on reserves and/or pricing.

Documentation Standards

- Each model run should produce a self-contained report including: input data, prior specifications and rationale, convergence diagnostics, fitted adjustments with credible intervals, comparison to the prior basis, and a recommendation.

- Version control should be applied to both the Stan model code and the basis output, so that any historical basis can be reproduced exactly.

Limitations Register

- Maintain an explicit list of known limitations, for example, the assumption that the starting basis is appropriate as a starting point, or the limited data in later durations.

12. Case Study 2 – Group IBNR

The following Group IBNR example will walk through the development of a Bayesian model for IBNR and shows how this model compares to a chain ladder model. The aim is to demonstrate a typical Bayesian workflow and show how we can produce quantities of interest using prior and posterior simulations. For actuaries, we are most interested in the loss distribution. To build the Bayesian model we have chosen to use a Negative Binomial (NB) model.

In the Australian group life insurance market, IBNR is a material assumption for both pricing and valuation. This is driven by long reporting delays for total and permanent disability products (TPD). The average reporting delay for TPD claims in Australia can be over 2 years (Actuaries Institute of Australia, 2017). This means that TPD claims experience in the most recent 2 years is underdeveloped creating uncertainty for pricing and valuation.

Typically, the Australian life insurance industry relies on non-parametric methods to estimate IBNR, with the chain ladder method being the most commonly used. There are two key issues with these approaches:

- The methods usually only calculate single point estimates that are then used for the best estimate assumptions;
- The experience for large group risk insurance arrangements is often treated as highly credible in practice with over reliance on historical experience to inform future reporting delays.

Bayesian models can address this by:
- Capturing the full loss distribution reflecting both the process risk and the misestimation of assumptions.
- Introducing priors and credibility weighting the actual experience against other evidence.

Data

In 2014, the Actuaries institute published an information note on IBNR discussing various methods (Actuaries Institute of Australia, 2014). This information note was accompanied with a claims dataset. This example uses this claims dataset to fit a TPD IBNR triangle.

To keep the example simple, it is assumed that the average claim size is known so that the chain ladder and NB models focus on predicting number of future reported claims.

Like the information note, a decline rate of 10% was assumed to weight pending claims count. This is something that should be sensitivity tested in practice.

The following claims triangles are presented on a half-yearly basis:

Table 7: Claim Counts (Paid + Pending)

	Reporting Delay (HY)																	
Event HY	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
2005H1	1	8	10	12	11	8	3	5	4	2	2	7	5	4	2	3	1	1.8
2005H2	4	14	24	11	9	11	3	5	3	5	5	2	3	3	4	3	2.9	
2006H1	5	16	21	16	6	7	8	7	11	10	3	3	3	5	1.9	3.8		
2006H2	4	16	19	6	12	12	13	7	7	5	7	8.9	5	2.9	1.8			
2007H1	4	13	20	13	11	12	10	10	5	10	8	1.9	1.9	3.9				
2007H2	2	19	20	12	20	10	7	7	3	6	6	8	5.9					
2008H1	10	32	28	23	16	14	14	10	12	6	7.9	8.5						
2008H2	5	32	34	19	14	4	20	17.9	9	6	14.2							
2009H1	15	31	31	22	10	13.9	13.8	9	12.9	11.4								
2009H2	12	43	44	11	16	15	18	15.9	14.3									
2010H1	2	43	20	18	15	16	27.9	21										
2010H2	6	26	49	31	21.9	17.9	23											
2011H1	1	20	30.9	22.8	32.6	26												
2011H2	4	34.9	37.8	28.7	44.9													
2012H1	1	21.9	49.4	38.7														
2012H2	1	31.6	49.9															
2013H1	1	36.2																
2013H2	2																	

Table 8: Claim Amounts ($'000)

	Reporting Delay (HY)																	
Event HY	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
2005H1	27	197	496	422	562	351	178	311	246	140	70	471	202	225	149	80	98	163
2005H2	143	709	1173	476	372	553	186	223	171	370	378	132	167	268	306	174	84	
2006H1	144	736	853	593	400	403	550	585	810	516	139	281	230	408	74	282		
2006H2	145	627	1148	222	368	794	641	444	472	149	366	581	255	75	169			
2007H1	113	786	852	806	483	604	634	480	215	489	344	141	117	250				
2007H2	108	834	1036	616	1108	533	374	418	227	370	372	560	441					
2008H1	321	1110	1321	1092	755	696	738	610	626	207	466	556						
2008H2	156	1441	1663	1058	407	202	1094	941	475	382	891							
2009H1	723	1014	1489	1173	432	774	684	471	497	680								
2009H2	469	1812	1764	669	883	476	1019	890	731									
2010H1	81	1766	859	754	820	724	1183	1294										
2010H2	347	855	2086	1458	741	710	1103											
2011H1	20	1013	1497	815	1611	1241												
2011H2	132	1190	1479	1200	2560													
2012H1	96	935	1932	1917														
2012H2	34	1209	2050															
2013H1	22	1228																
2013H2	152																	

Table 9: Average Claim Size ($'000)

	Reporting Delay (HY)																	
Event HY	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
2005H1	36	38	39	41	43	44	46	48	49	51	52	54	56	57	59	61	62	64
2005H2	37	38	40	41	43	45	46	48	50	51	53	55	56	58	60	61	63	64
2006H1	37	39	40	42	44	45	47	49	50	52	53	55	57	58	60	62	63	65
2006H2	38	39	41	42	44	46	47	49	51	52	54	56	57	59	60	62	64	65
2007H1	38	40	41	43	45	46	48	49	51	53	54	56	58	59	61	63	64	66
2007H2	38	40	42	43	45	47	48	50	52	53	55	57	58	60	61	63	65	66
2008H1	39	41	42	44	46	47	49	50	52	54	55	57	59	60	62	64	65	67
2008H2	39	41	43	44	46	48	49	51	53	54	56	58	59	61	62	64	66	67
2009H1	40	42	43	45	47	48	50	51	53	55	56	58	60	61	63	65	66	68
2009H2	40	42	44	45	47	49	50	52	54	55	57	59	60	62	63	65	67	68
2010H1	41	43	44	46	48	49	51	52	54	56	57	59	61	62	64	66	67	69
2010H2	41	43	45	46	48	50	51	53	55	56	58	59	61	63	64	66	68	69
2011H1	42	44	45	47	48	50	52	53	55	57	58	60	62	63	65	67	68	70
2011H2	42	44	46	47	49	51	52	54	56	57	59	60	62	64	65	67	69	70
2012H1	43	45	46	48	49	51	53	54	56	58	59	61	63	64	66	68	69	71
2012H2	43	45	47	48	50	52	53	55	57	58	60	61	63	65	66	68	70	71
2013H1	44	46	47	49	50	52	54	55	57	59	60	62	64	65	67	69	70	72
2013H2	44	46	48	49	51	53	54	56	58	59	61	62	64	66	67	69	71	72

Chain ladder model

The basic chain ladder model is a widely used approach for developing IBNR assumptions. Its ubiquitousness is likely due to its simplicity which makes it easy to communicate and update. In this section we will show the formulation used to generate the chain ladder model that are used to compare against the Bayesian model in the later sections.

Let $X_{i,j}$ be the incremental claim count respectively for event period i and development period j such the that incremental claims triangle with I event periods and J development period is as follows:

$$\begin{bmatrix} X_{1,1} & \cdots & X_{1,J} \\ \vdots & \ddots & \vdots \\ X_{I,1} & \cdots & X_{I,J} \end{bmatrix}$$

Let $C_{i,j}$ be the cumulative claim count respectively for event period i and development period j such that:

$$C_{i,n} = \sum_{j=1}^{j=n} X_{i,j}$$

The chain ladder model assumes the following:

$$C_{i,j} = C_{i,j-1} \times f_j$$

Where f_j is the claims development factor for development period j. The usual chain ladder method calculates these factors as:

$$\hat{f}_j = \frac{\sum_{i=1}^{n-j} C_{i,j+1}}{\sum_{i=1}^{n-j} C_{i,j}}$$

Often the development factors are calculated in a variety of ways including:

- Calculating individual age-to-age factors and calculating the average age-to-age factor.
- Assuming that development periods in the tail follow an exponential or hyperbolic decay.
- Using data from more recent reporting periods to inform the development factors.

The implicit assumption regardless of how these claim development factors are determined is that each event period will have the same development factors.

Negative Binomial model

The over-dispersed Poisson model for IBNR is a well-documented model for loss reserving first appearing in (Renshaw & Verrall, 1998). It assumes that the claims count or amount in each cell of the claim triangle can be expressed as a generalised linear model (GLM) with a Poisson distribution with a quasi-likelihood function to allow for over-dispersion. Here we have chosen to use a negative binomial distribution over the Poisson distribution with a quasi-likelihood for convenience as the negative binomial distribution is supported in Stan and produces very similar outcomes. This model will work on either cumulative triangles or incremental triangles. We will work with the incremental triangle in this example.

Here we parameterise the NB model by assuming that the lambda parameter (i.e. the mean of the NB model) varies by event period and development period. This is achieved through a GLM structure assuming a log link function and log of lambda varying by a linear combination of factors: alpha (event period factors), betas (development period factors) and a flat constant c. The constant c captures the global scale of the triangle, this is achieved through applying sum-to-zero constraints on both alpha and beta.

For each element (X_{ij}) of an incremental count triangle with event period i and development period j, it is assumed that:

$$X_{i,j} \sim NB(\lambda_{i,j}, \phi)$$

$$\log(\lambda_{i,j}) = \alpha_i + \beta_j + c$$

The above is not sufficient for this model to be identifiable; to make it identifiable we will assume the following constraints:

$$\sum_i \alpha_i = 0$$

$$\sum_j \beta_j = 0$$

Note that there are different ways to parameterise the negative binomial distribution. Here we have parameterised it with λ, the mean of the distribution and ϕ, such that the variance equals $\lambda + \lambda^2/\phi$. Here ϕ is the overdispersion parameter.

Connection of the NB model to the chain ladder model

The NB model can produce very similar results to the chain ladder model. The beta parameters can be transformed into link ratios to compare with the link ratio factors in the chain ladder model.

The key difference between the NB and the chain ladder model is that the NB model is a fully parametric model hence doesn't rely on actual data to produce estimates once fitted. The chain ladder model requires actual reported claims to gross up to calculate IBNR estimates. We will see that the maximum likelihood estimate of the NB model produces similar estimates of IBNR to the chain ladder model.

Motivation for using Bayesian estimation

Existing maximum likelihood estimation techniques are fully capable of fitting this model, so why go to the effort of trying to estimate this model using a Bayesian model?

There are several reasons:

- There is a lot of uncertainty in the parameterisation of these models that is lost when generating point estimates for parameters.
- We are typically interested in more than the best estimate assumption, and want to know the full range of estimates for reserving, management information and capital purposes.
- We can test to see if the underlying assumptions of our model hold true.
- We may be interested in subdividing this data into other groups such as age bands, divisions, occupations etc. Bayesian estimation can regularise through the choice of priors or through hierarchical modelling.

It should be noted that Bayesian methods are not the only stochastics methods available to estimate the uncertainty in the IBNR. Other methods include those based on fitting GLMs and Mack's method (England & Verrall, 2002), (Taylor & McGuire, 2016).

Prior Selection and Prior Predictive Simulations

For priors we use the normal distribution for the parameters α, β, and c. The main motivations for selecting the normal distribution is:

- Sampling convenience.
- Regularisation properties that normal priors have.
- Supported by the shape of the empirical distribution of these parameters.

The exponential distribution for phi was selected as this distribution is continuous, positive, and has a single parameter.

The priors that have been selected for the TPD model are as follows:

Table 10 – Prior selection for NB IBNR model

Parameter	TPD Model
α	normal(0, 1)
β	normal(0, 1)
c	normal(1.5, 0.25)
phi	exponential(1)

The parameters of the prior distributions have been set so that it covers a wide range of outcomes for all the X_{ij} in the triangle and the IBNR for each event period. The priors have been calibrated such that the median loss ratio is close to a reasonable figure. Prior predictive checks have been to confirm that this is the case.

Prior predictive simulations are useful in checking the selection of priors. The goal of this check is to use the prior to simulate data to assess if the data produced is sufficiently spread across the range of plausible outcomes including extreme scenarios.

Figure 20 below shows the prior predictive checks which have sampled the prior to produce metrics on the implied loss ratio, annual claims count and overdispersion parameter phi.

To produce the prior predictive simulation in this IBNR example, the following procedure is used:

1. k samples were randomly generated for each parameter for each event period i and development period j using the specified prior distribution:
    a. $\alpha_{k,i} \sim \mathrm{normal}(0, 1)$
    b. $\beta_{k,j} \sim \mathrm{normal}(0, 1)$
    c. $c_k \sim \mathrm{normal}(1.5, 0.25)$
    d. $\phi_k \sim \mathrm{exponential}(1)$

2. Samples of lambda are calculated for each element of the triangle using the samples from the previous step:

$$\lambda_{k,i,j} = \exp(\alpha_{k,i} + \beta_{k,j} + c_k)$$

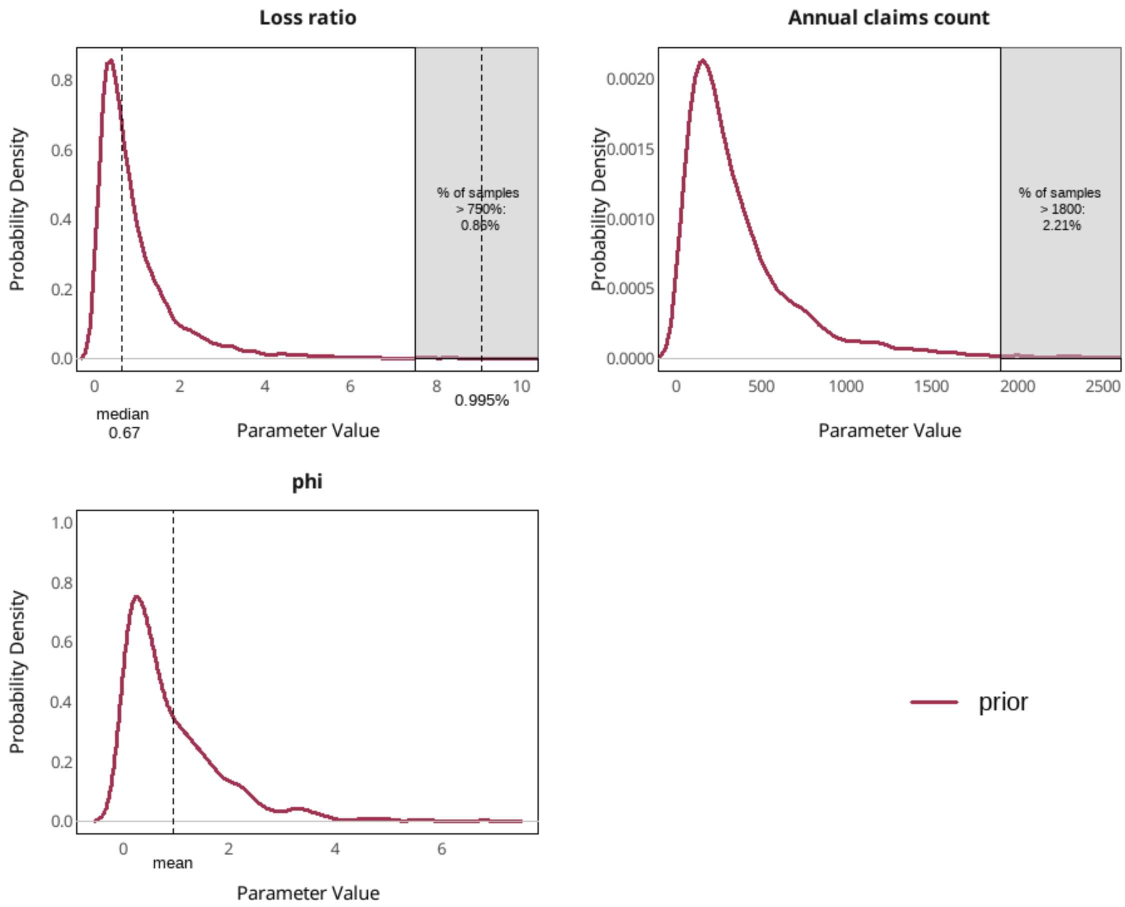
3. Using the lambda and phi samples generated above, a claim count for each element of the triangle was randomly generated using a negative binomial distribution for each element of the triangle:

$$y_{k,i,j}^{rep} \sim NB(\lambda_{k,i,j}, \phi_k)$$

The output from this step are k iterations of the full incremental claims count triangle.

The prior predictive checks below show the loss ratios predicted by the prior distribution have a probability density mass at 67%, which is quite reasonable and less than 1% of loss ratios sampled exceed an extreme value of 750%.

Figure 18: Prior predictive check on the TPD IBNR



Model

The Bayesian model selected based on the prior predictive checks is summarised below:

Priors

For each event period i and development period j:

$$\alpha_{\mathrm{i}} \sim \mathrm{normal}(0, 1)$$
$$\beta_{\mathrm{j}} \sim \mathrm{normal}(0, 1)$$
$$c \sim \mathrm{normal}(1.5, 0.25)$$
$$\phi \sim \mathrm{exponential}(1)$$

$$\lambda_{\mathrm{i}, j} = \exp(\alpha_i + \beta_j + \mathrm{c})$$

Likelihood

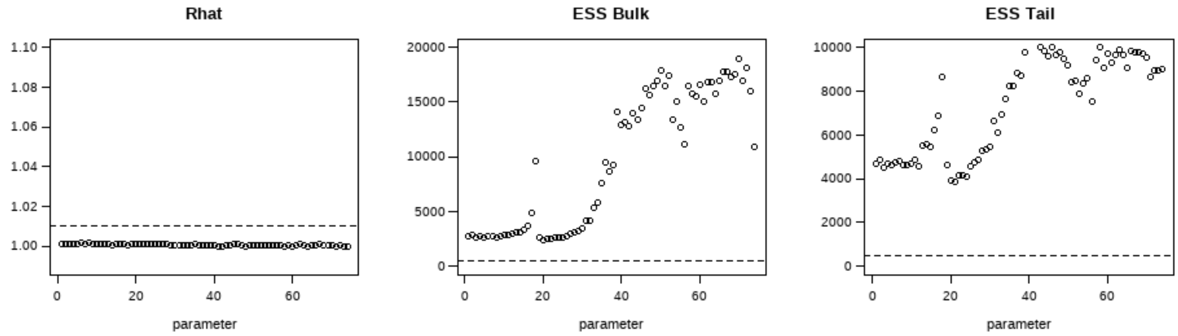
$$X_{i,j} \sim NB(\lambda_{i,j}, \phi)$$

The Stan code for this model can be found in the *Appendix - Code for Case Study 2* below.

Model diagnostics

The model above was fitted using 5 chains and 5000 iterations. To check if the model has converged, we will rely on the model diagnostic metrics R-hat, ESS Bulk, and ESS Tail. The figure below shows that each model parameter has an R-hat below the 1.01 threshold confirming that the chains have converged; And both ESS Bulk and Tail were above the 500 ESS threshold confirming the estimate of the posterior distribution is reliable.

Figure 19 - IBNR model diagnostics



Posterior predictive checks on distribution

Posterior predictive checks work in a similar fashion to the prior predictive checks with the exception that we compare the posterior distribution against empirical data and observe if the model reflects features present in the empirical data. This is similar to an actual vs expected analysis.

To produce the posterior predictive simulation in this IBNR example, the following procedure is used:

For each sample in the posterior distribution i.e. each sample of the parameter α, β, c and phi:

- A sample of lambda was generated for each element of the triangle i.e. each event period i and development period j: $\lambda_{k,i,j} = \exp(\alpha_{k,i} + \beta_{k,j} + c_k)$.
- Using the lambda generated above and phi, a claim count for each element of the triangle was randomly generated using a negative binomial distribution for each element of the triangle: $y_{k,i,j}^{rep} \sim NB(\lambda_{k,i,j}, phi_k)$.

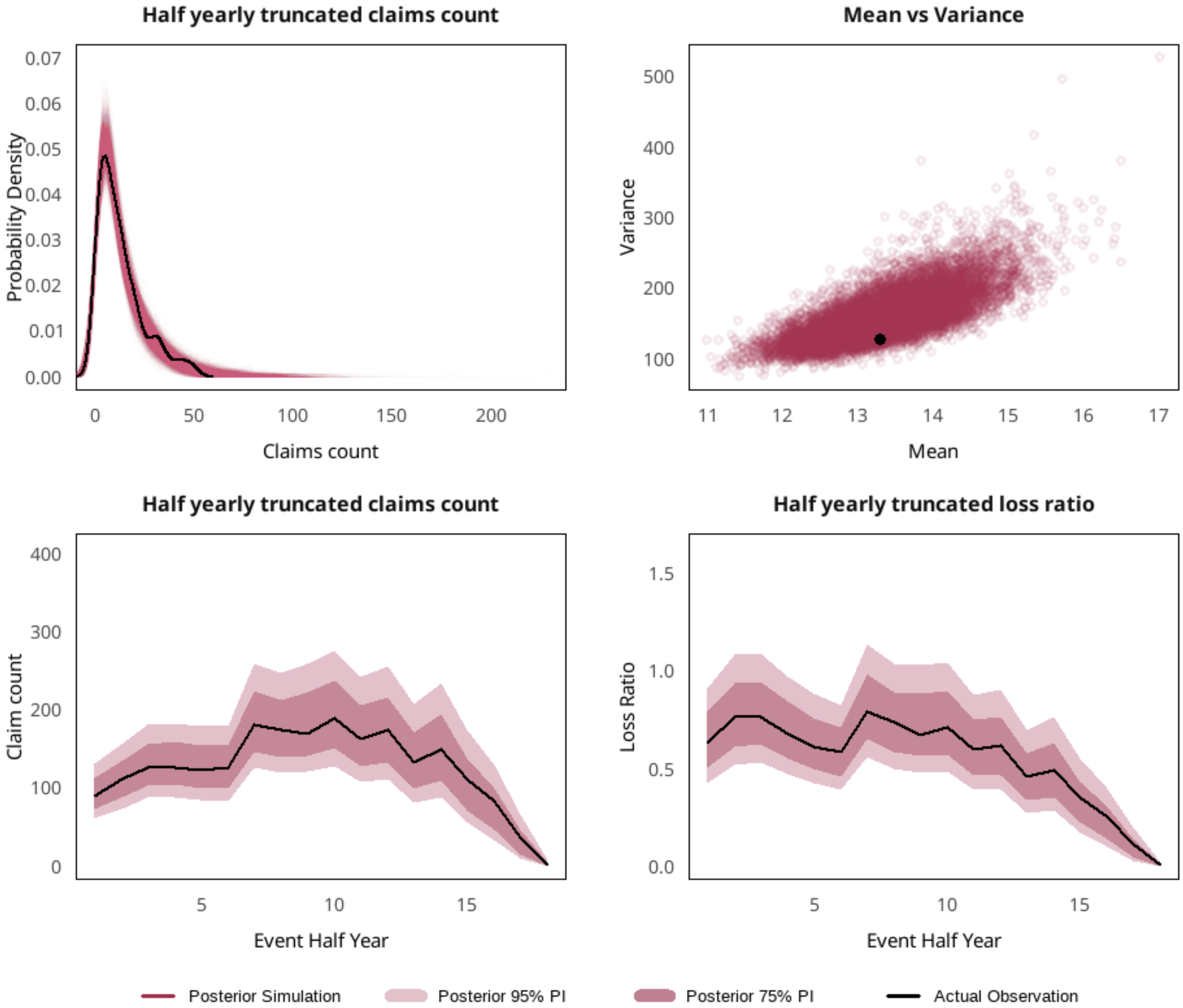
The output from this step are k iterations of the full incremental claims count triangle.

Various transformations of this data are used to generate other metrics such as:

- The truncated claims count triangle: This is the upper half of the claims triangle representing the claims reported to the valuation date. This compares the actual claims reported against the expected predicted by the model.
- Mean and variance: This is the mean and variance of each triangle. This is used to compare the mean/variance relationship of the actual claims.
- The truncated loss ratio: This the loss ratio implied for each event period using the truncated claims count triangle multiplied by the average claim size and divided by the premium for each event period. This compares the actual loss ratio reported against the expected predicted by the model.

The model fitted has appropriately fitted claims count, the mean/variance relationship and loss ratios in line with the empirical distribution (the black line/dot):

Figure 20: Posterior Predictive Simulation



Presentation of results

Distribution of IBNR and ultimate claims

The following graph shows the IBNR predicted by the NB model using the posterior distribution (grey bars) in comparison to the chain ladder estimate. The graph shows that the chain ladder result is closely aligned to the mode of the NB estimate of IBNR.

The key practical advantage of the Bayesian model is in its ability to produce a full distribution of results based on the uncertainty in parameter estimates implied by the posterior distribution. We can see that the IBNR estimate has a wide distribution that is hidden in the chain ladder estimate. The other feature that we can observe in the loss ratio by event year is that there is large fan of outcomes in the latest event periods well above the chain ladder estimate showing the uncertainty in these underdeveloped periods.

For pricing actuaries, the loss distribution generated can be utilised for:

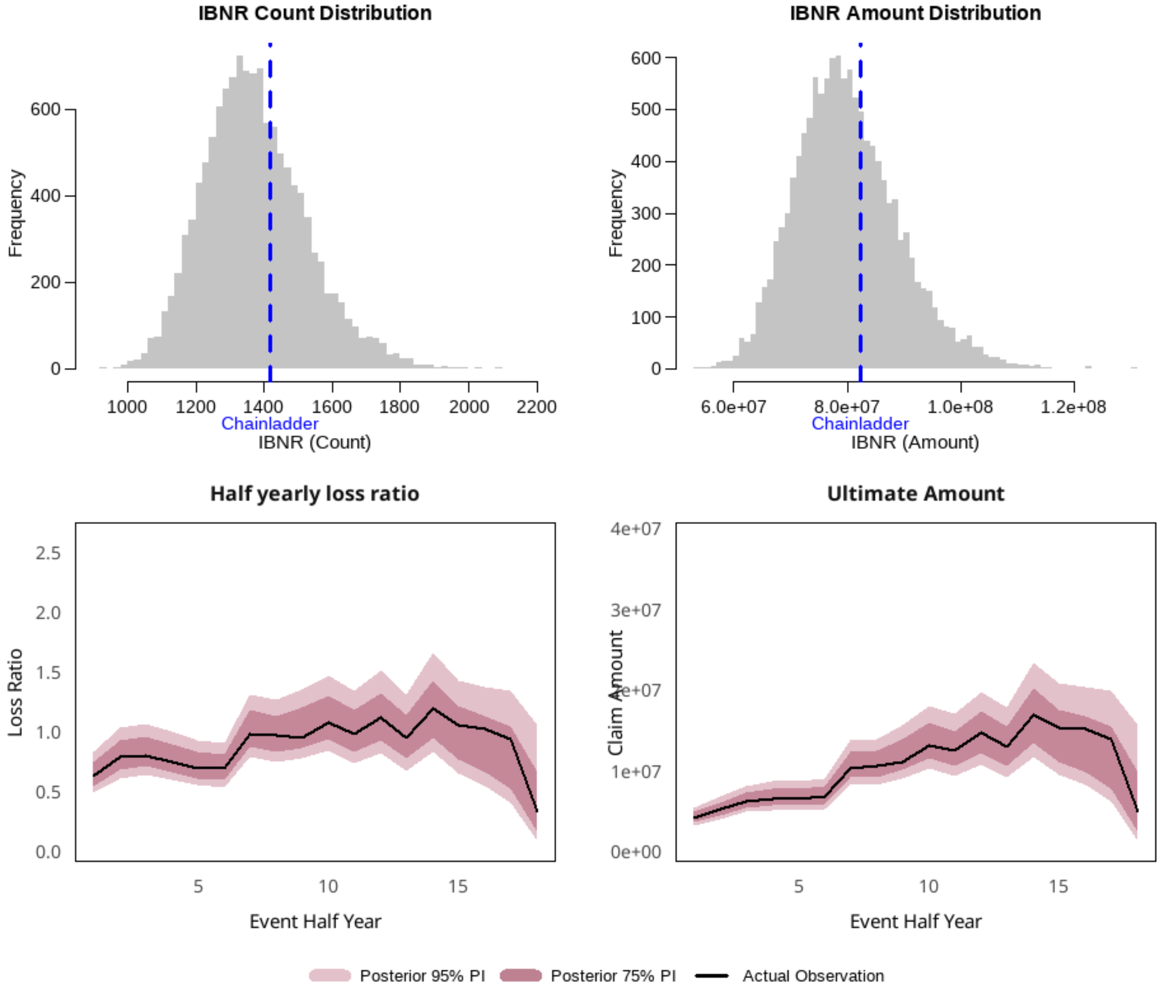
- Pricing experience sharing mechanisms.
- Measuring the level of uncertainty in ultimate claim amounts in underdeveloped event periods. This information can be used to convey levels of confidence over various pricing outcomes.

For valuation actuaries, the loss distribution generated can be utilised for:

- Develop reserves and capital estimates that are designed to be held at various quantiles of the loss distribution. The loss distribution below can be used to develop quantities such as value at risk (VAR) and conditional tail expectation (CTE).

It should be noted that the distribution is not a full distribution of risks; there are unmeasured quantities such as model, catastrophe, regime changes etc that are not captured in the loss distribution. The key risks captured here are process and misestimation risk.

Figure 21 – Projected IBNR distributions, ultimate loss ratios and claim amounts



Conclusion

This case study has demonstrated how practitioners can develop a Bayesian model using realistic data using a Bayesian workflow. We have shown that the Bayesian approach produces results broadly consistent with the chain ladder method at the best estimate level, while delivering substantially more information about the full distribution of possible outcomes.

The key advantages demonstrated in this example are:

1. The full posterior distribution of IBNR quantifies both process risk and parameter uncertainty in a single quantitative framework, rather than using a qualitative framework.
2. The Bayesian workflow — prior predictive checks, MCMC diagnostics, and posterior predictive checks — provides a structured and transparent basis for model validation that is auditable and communicable to stakeholders.

As with any model, the results should be interpreted with an understanding of its limitations. The loss distribution produced captures process and parameter uncertainty but does not account for model risk, catastrophic events, or structural regime changes in claims experience. These limitations do not diminish the value of the approach but should be kept in mind when using these models.

13. Probabilistic Programming and Stan

In this paper we have mentioned the use of PPLs frequently. PPLs can be either a stand-alone language like Stan or libraries built on top of an existing language like PyMC on Python that allow users to specify a Bayesian model. For this paper we have mostly used Stan as we find it allows us to represent our statistical models close to their mathematical form and can be run in many other environments. Below is a brief description of the main components of a Stan model.

The Stan language breaks down the model into several components:

```
data {
        This section defines the data that is being inputted into our model plus anything else
        that we want to specify from outside the model e.g. any fixed parameters we want to
        use, user defined values or dimensions etc
}

transformed data {
        This section allows us to do any initial calculations on our data e.g. normalising data.
        These calculations are only performed once before the model is run properly.
}

parameters {
        This section defines the parameters of our model. Every parameter will be included
        in the posterior distribution. We are not defining any distributions at this stage.
}

transformed parameters {
        This section defines any calculations or transformations that we want to do on our
        parameters.
}

model {
        This section is where we define our priors and likelihood function. Anything to do
        with the distributions of the data or the parameters is specified within this section.
        Distributions or probabilities can either be defined directly by specifying the
        probability distribution (e.g. x ~ normal(mu, sigma)) or by adding on the probability
        density function onto the log likelihood/posterior (e.g. target += normal_lpdf( x | mu,
        sigma) where normal_lpdf() is the Normal probability density function for x).
}

generated quantities {
        This calculates any further values we want calculated from the samples from the
        posterior distribution.
}
```

14. Glossary

Core Bayesian Concepts

Term	Description
Bayes' Theorem	The core updating rule of Bayesian statistics: posterior is proportional to prior multiplied by likelihood, with evidence acting as a normalising term.
Bayesian modelling	A modelling framework that combines prior beliefs, observed data, and uncertainty to estimate parameters and outcomes probabilistically.
Bayesian workflow	A disciplined process for building, fitting, checking, and refining Bayesian models, including prior checks, fitting, diagnostics, and predictive checks.
Prior distribution	The probability distribution expressing beliefs or external knowledge about a parameter before observing the current data.
Likelihood	The part of the model that describes how probable the observed data are for different parameter values.
Posterior distribution	The updated distribution of parameter values after combining the prior with the observed data through the likelihood.
Evidence / marginal likelihood	The normalising constant in Bayes' theorem obtained by integrating over all possible parameter values so the posterior sums or integrates to 1.
Bayesian updating	The process of using the posterior from one stage as the prior for the next stage as new data arrive.
Credibility	The weight given to observed experience relative to prior assumptions or external information; in Bayesian models this arises naturally through the model structure.
Credible interval	A Bayesian interval giving a range of parameter values with a stated posterior probability, such as a 95% credible interval.

MCMC and model diagnostics

Term	
Probabilistic programming language (PPL)	Software such as Stan or PyMC that lets the modeller specify priors and likelihoods while the software handles posterior computation.
Markov Chain Monte Carlo (MCMC)	A family of algorithms used to generate samples from complex posterior distributions when analytic solutions are not available.

Term	
Chain	One independent run of an MCMC algorithm; multiple chains are compared to assess whether the sampler has converged.
Warmup / burn-in	The initial part of an MCMC run that is discarded because the sampler is still settling into the high-density region of the posterior.
Sampling period	The post-warmup part of an MCMC run whose draws are used to summarise the posterior.
Convergence	The point at which MCMC samples are considered a reliable representation of the posterior distribution.
Trace plot	A diagnostic plot showing sampled parameter values across iterations; well-mixed, stable traces suggest convergence.
Density plot	A plot of the sampled posterior distribution; similar densities across chains support convergence.
R-hat	A convergence diagnostic comparing variation within and between chains; values close to 1 indicate good convergence.
Effective sample size (ESS)	The number of independent draws that the autocorrelated MCMC sample is effectively worth.
Prior predictive check	A check that simulates data from the priors before fitting, to see whether the implied outcomes are reasonable for the problem.
Posterior predictive check	A check that simulates data from the fitted posterior to assess whether the model can reproduce important features of the observed data.
Leave-one-out cross-validation (LOOCV)	A model-checking and comparison approach in which observations are left out one at a time to assess predictive performance.

Model Structure and Data Issues

Term	
Hierarchical model	A model that shares information across related groups or features so estimates for smaller or noisier groups are stabilised by the broader population.
Regularisation / shrinkage	A modelling feature that pulls noisy parameter estimates toward simpler or more plausible values, often through choosing a prior centred on 0.

Term	
Latent variable	An unobserved quantity introduced into the model to represent hidden structure or influences in the data.
Censoring	A situation where an observation is present, but the exact value of interest is only partially observed, such as a claim still in payment.
Truncation	A situation where observations outside a specified range are not included in the dataset at all.
Over-dispersion	A condition where the variance in the data exceeds what, for example, a standard Binomial, Poisson or Normal model would imply, requiring a more flexible likelihood.
Robust regression	Regression designed to reduce sensitivity to outliers, for example by using a Student-t rather than Normal likelihood.
Gaussian process (GP)	A non-parametric Bayesian method for fitting smooth functions by specifying how nearby points are correlated.

15. Resources

There are many great resources available to learn about Bayesian modelling. Below is a short list of recommended resources for actuaries who want to learn more:

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A. and Rubin, D.B. 2013. *Bayesian Data Analysis*, Third Edition. Chapman-Hall/CRC
(or anything else written by Andrew Gelman)

Richard McElreath's Statistical Rethinking: A Bayesian Course with Stan and R, Chapman Hall/CRC.
(also check out his excellent lecture series on Youtube: www.youtube.com/@rmcelreath)

Stan language website: https://mc-stan.org/.

PyMC language website: https://www.pymc.io/welcome.html

Both the Stan and PyMC websites have further references to some excellent learning resources.

16. References

Actuaries Institute of Australia. (2014). *Information Note: IBNR.* Actuaries Institute of Australia. Retrieved from https://content.actuaries.asn.au/resources/resource-ce6yyqn64sx3-2093352434-59316

Actuaries Institute of Australia. (2017). *Efficiency and competiveness in the superannuation system.* Actuaries Institute of Australia. Retrieved from https://content.actuaries.asn.au/resources/resource-ce6yyqn64sx3-2093352434-18352

Andrew Gelman, J. C. (2015). *Bayesian Data Analysis Third Edition.* CRC Press.

Betancourt, M. (2017). A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv*.

Betancourt, M. (2020, April). *Principled Bayesian Workflow*. Retrieved from Michael Betancourt: https://betanalpha.github.io/assets/case_studies/principled_bayesian_workflow.html

Carvalho, C. M., Polson, N. G., & Scott, G. J. (2010). The horseshoe estimator for sparse signals. *Biometrika, 97*(2), 465-480.

England, P., & Verrall, R. (2002). Stochastic Claims Reserving in General Insurance. *British Actuarial Journal*, 443-544.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society Series A: Statistics in Society, 182*(2), 389-402.

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., . . . Modrák, M. (2020). *Bayesian Workflow.* ArviZ.

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research, 15*, 1593-1623.

Mitchell, T. J., & Beauchamp, J. J. (1988). Bayesian Variable Selection in Linear Regression. *Journal of the American Statistical Association, 83*(404), 1023-1032.

Renshaw, A. E., & Verrall, R. J. (1998). A STOCHASTIC MODEL UNDERLYING THE CHAIN-LADDER TECHNIQUE. *British Actuarial Journal*, 4(4), 903–923.

Taylor, G., & McGuire, G. (2016). *Stochastic Loss Reserving Using Generalized Linear Models.* Casualty Actuarial Society.

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing, 27*(5), 1413-1432.

Vehtari, A., Gelman, A., Simpson, D., Bob, C., & Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved Rhat for assessing convergence of MCMC*. *Bayesian analysis, 16*(2), 667-718.

Appendix – Code for Example 2

```
data {
  int<lower=0> N;
  vector[N] x;
  vector[N] y;
}

parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;  // restricted to > 0 because it is a standard deviation
}

model {
  // priors
  alpha ~ normal(0, 100);
  beta ~ normal(0, 100);
  sigma ~ normal(0, 5); // because sigma is restricted to >0 sigma is effectively half-normal

  // likelihood
  y ~ normal(alpha + beta * x, sigma);
}
```

```
library(cmdstanr)

stan_data = list(
 N = 10,
 x = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
 y = c(1.6, 3.2, 5.1, 2.9, 4.4, 5.1, 6.2, 5.8, 8.5, 6.9))

compiled_model = cmdstan_model(stan_file = “linear-regression.stan“)

fitted_model = compiled_model$sample(
            data = stan_data,
            iter_warmup = 1000,
            iter_sampling = 10000,
            chains = 4)
```

```
library(bayesplot)
draws = fitted_model(variables = c("alpha", "beta", "sigma"), inc_warmup = TRUE)
mcmc_dens_overlay(draws, pars = variable)
mcmc_trace(draws, pars = variable, n_warmup = 1000)
```

Appendix – Code for Case Study 1

```
data {
int<lower=1> D;                    // number of durations
array[D] int<lower=0> terminations;  // observed terminations
vector<lower=0>[D] exposure;        // exposure by duration
vector<lower=0>[D] mu_std;          // starting termination rates
}

parameters {
real alpha;                        // overall portfolio level shift vs starting basis
vector[D] g;                       // smooth duration-specific deviation
real<lower=0> sigma_g;    // smoothness scale for second differences
}

transformed parameters {
vector[D] logit_mu;
vector<lower=0>[D] mu;
logit_mu = logit(mu_std) + alpha + g;
mu = inv_logit(log_mu);
}

model {
// prior distributions
alpha ~ normal(0, 0.3);
sigma_g ~ normal(0, 0.1);
g[1] ~ normal(0, 0.3);
g[2] ~ normal(g[1], sigma_g);
g[D] ~ normal(0, 0.5);

// Second-order random walk prior for smooth graduation by duration.
for (d in 3:D)    g[d] ~ normal(2 * g[d - 1] - g[d - 2], sigma_g);

// Binomial likelihood function
for (d in 1:D)   terminations[d] ~ binomial(exposure[d] , mu[d]);
}
```

Appendix - Code for Case Study 2

```
/*
The following model is a negative binomial model for estimating the incremental
claims in a triangle.

The model has the following structure for each i event period, and j event period:

claims[i,j] ~ negative_binomial(lambda[i,j], phi)

log(lambda[i,j]) = a[i] + b[j] + c

sum(a) = 0

sum(b) = 0

Claims data must include reported periods with no claims as well! This means
any reported period before the valuation date with zero claims reported must be
included in the data

*/

functions{

  // custom log pdf for negative binomial type 2 distribution

  real neg_binomial_2_real(real x, real mu, real phi){
  return lgamma(x + phi) - lgamma(x + 1) - lgamma(phi) + x * log(mu) + phi * log(phi) - (x + phi) * log(mu + phi);

}

}

data{

  int n_obs; // Number of observations
  int n_event; // number of event periods in triangle
  int n_dev; // number of development periods in triangle

  real a_mu; // mu for the a parameter
  real b_mu; // mu for the b parameter
  real c_mu; // mu for the c parameter
  real phi_rate; // rate for the phi parameter



  real a_sigma; // sigma for the a parameter
  real b_sigma; // sigma for the b parameter
  real c_sigma; // sigma for the c parameter



  array[n_obs] int event_per; // Event period index value
  array[n_obs] int dev_per; // Development period index value
  array[n_obs] real count; // Count of claims
}
```

```
parameters{

  vector[n_event] a_raw;
  vector[n_dev] b_free;
  real c;
  real<lower=0> phi;

}

transformed parameters{

  vector[n_dev] b;
  vector[n_event] a;

  // Adding a contraint of b such that the sum(b) = 0 to make the model identifiable
  b = b_free - mean(b_free);

  // Adding a contraint of a such that the sum(a) = 0 to make the model identifiable
  a = a_raw - mean(a_raw);


}

model{

  vector[n_obs] lambda;

  // Priors
  a_raw ~ normal(a_mu, a_sigma);
  b_free ~ normal(b_mu, b_sigma);
  c ~ normal(c_mu, c_sigma);
  phi ~ exponential(phi_rate);

  //Likelihood
  for(i in 1:n_obs){

   target += neg_binomial_2_real(count[i], exp(a[event_per[i]] + b[dev_per[i]] + c), phi);

  }

}

generated quantities{

  //posterior predictive simulation - observed data
  vector[n_obs] yrep;

  for(i in 1:n_obs){
      yrep[i] = neg_binomial_2_rng(exp(a[event_per[i]] + b[dev_per[i]] + c), phi);

  }


  //posterior predictive simulation - full triangle projection
  matrix[n_event, n_dev] exp_claims_sim;

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

```
    for(j in 1:n_dev){
      exp_claims_sim[i,j] = neg_binomial_2_rng(exp(a[i] + b[j] + c), phi);
    }
  }

    matrix[n_event, n_dev] exp_claims;

  for(i in 1:n_event){
    for(j in 1:n_dev){
      exp_claims[i,j] = exp(a[i] + b[j] + c);
    }
  }

}
```