Trinity College Dublin

School of Computer Science and Statistics

M.Sc. by Research in Statistics

2020

Modelling the Distribution of Grouped Survival Data via Dependent Neutral-to-the-Right Priors

---

Author:
Fearghal Donaghy

Supervisor:
Dr. Bernardo Nipoti

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Signed: ______________________________

Fearghal Donaghy

Summary

The key methods used in this thesis are rooted in Bayesian nonparametric statistics. We construct a model for the analysis of right censored survival data. The data the model is built to analyse is discrete, separated into distinct groups, and features dependence between groups. In addition, the total number of censored observations in each group is unobservable.

We use neutral-to-the-right priors to specify our model and express the interdependence using superposition of completely random measures. In particular, the distribution of each group a priori is characterised by the linear combination of two completely random measures: one which is unique to that group and one which is common to all groups.

Our main theoretical results in Section 4 feature a characterisation of the posterior distribution and from there the derivation of each group's posterior mean distribution given a number of auxiliary variables. This derivation is carried out both for a general distribution and after we specify the completely random measures using a one-parameter Gamma base measure and a σ-stable density.

We construct a method for obtaining pointwise estimates of the posterior mean by marginalising with respect to the aforementioned auxiliary variables using a Gibbs sampler. Within this sampler, we use a Metropolis-Hastings algorithm to sample from the more complex full conditionals.

In order to test our model, we perform a simulation study using data generated from a number of combinations of Dirichlet processes. Then the model is applied to real data from our motivating illustration. These data comprise the report times of bugs in various releases of Mozilla Firefox's internet browser.

For both the simulated data and the real data, we calculate pointwise estimates for the posterior means and associated credible intervals for each group.

Our study demonstrates the advantages of borrowing of information across groups in the above described way. For instance, it allows us to model more effectively the distribution of groups for which there is a lack of exact observations because that group's censoring time is particularly early.

All computation was carried out using the R programming language.

Acknowledgements

Massive and sincere thanks are due to my supervisor, Dr. Nipoti, who gave hugely of his time, patience, and expertise throughout this project. His constant guidance has been invaluable to me. For going beyond the call of duty and affording me unique opportunities to learn and to better myself, I owe him hugely.

Without the unwavering support of my parents, I would never have come as far as I have. Each of my academic milestones serves as a testament to their belief in my potential. Their unbounded capacity for understanding and love has always sustained me. For this and for loads more besides, I will always be grateful.

Many kinds of support can often go overlooked or be taken for granted. I want to take this opportunity, therefore, to offer enormous thanks to my friends who gave freely of their company, their advice, their encouragement, and in some cases their apartments during this project. You know who you are.

It has been a pleasure to write this dissertation for many reasons. Two of those reasons are named Carly and Alan, who kept me in good cheer throughout the writing process. Thank you both kindly for your help.

Contents

1 Introduction

A typical framework for statistical inference assumes that the data y_i, $i = 1, 2, \ldots, n$ are drawn independently from an underlying distribution function G. A parametric approach further assumes that the probability density function g of G belongs to a family $\mathcal{G} = \{g_\theta : \theta \in \Theta\}$ labelled by a set of parameters θ belonging to an index set Θ.

When introducing the field of Bayesian nonparametric statistics, Müller et al. (2015) explain that constraining inference to a specific parametric form in this way can limit the scope of the resulting model. Ghosal and Van der Vaart (2017) agree, telling us that nonparametric models can allow us to avoid "the arbitrary and possibly unverifiable assumptions inherent in parametric models".

Our project will be carried out using Bayesian nonparametric analysis for this reason and because of the variety of useful tools available in this domain. For instance, we will make much use of neutral-to-the-right (NTR) processes, introduced by Doksum (1974). NTR priors are especially useful in survival analysis due to their structural conjugacy property in the case of right-censored data.

Completely random measures (CRMs) will feature heavily in our project, not least because of the useful mathematical relation they bear to NTR processes. First introduced by Kingman (1967), CRMs can readily characterise NTR processes and vice-versa. Their analytical tractibility and the mathematical convenience of their Lévy-Khintchine representation make them ideal for our purposes.

In our motivating illustration, the data comprise the report times of bugs across different Firefox software releases. Modelling interdependence between these releases is a fundamental aspect of our model. We will draw on the work of Lijoi and Nipoti (2014) when we establish this interdependence through superposition of CRMs.

Once our model is specified, we derive the results necessary to sample from the posterior and test these results using data simulated from known distributions.

1.1 Motivation

The data which motivates our model describes the discovery times of software failures or bugs in various releases of Mozilla Firefox's internet browser. This same dataset was analysed by Wilson and Ó'Ríordáin (2018), who also took a Bayesian approach, though not a nonparametric one. This data is publically available from the Bugzilla tracking system at bugzilla.mozilla.org, where newly discovered bugs in Firefox software are reported and their details are freely available.

Approximately every 42 days, Firefox release a new version of their browser. We work with the bug report times for each release between June 2011 and July 2013. The cumulative number of bugs reported for each version and their corresponding release dates are illustrated in Figure 1.

The data we analyse is discrete, with times of big reports known down to the hour. This discreteness allows for ties within and between releases, a feature of the data we will not only accommodate but use to our advantage.

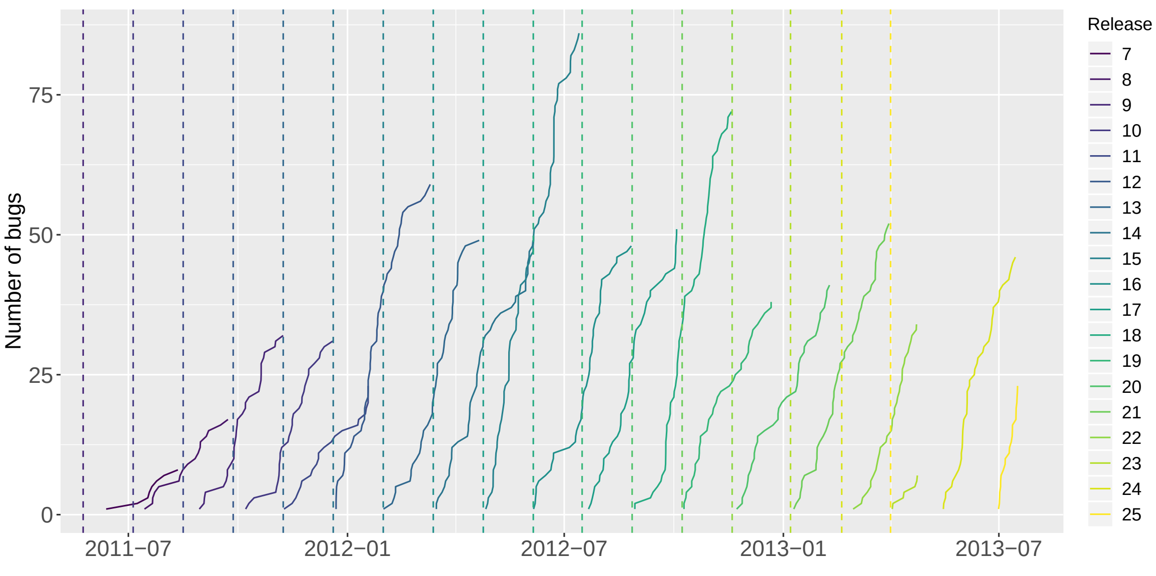


Figure 1: Mozilla Firefox bug report data

The problem facing us is how to describe data of this form, that is data with the following properties:

- Observations come from a discrete distribution and thus show ties with positive probability.
- The total number of censored obervations in the data is unobservable.
- The data is not easily described by existing parametric models.
- Data might be characterised by homogeneity within each release-specific sample but by heterogeneity across samples.
- There may exist dependence between releases of this data which we wish to exploit.

The second point above introduces an element of randomness which makes the issue of sampling from our posterior distributions non-trivial. We will need to marginalise with respect to a number of random variables in order to gain a posterior we can sample from.

We will construct our model using the tools and framework of Bayesian nonparametrics. In particular, we will make use of neutral-to-the-right priors and introduce prior dependence between distinct software releases.

Before we can achieve any of this, we will familiarise ourselves with the necessary tools.

2 Background

Bayesian nonparametrics is the statistical framework within which we build our model. In Bayesian statistics, we assume a prior for a certain model and incorporate the information gleaned from the data using the likelihood. We then derive the posterior distribution through the product of these objects.

In Bayesian nonparametrics, the principles of this practice remain in place but we dispense with the assumption that our model follows a parametric distribution. Instead, we allow the distributions, both prior and posterior, to exist outside the restrictions of, say, a Gamma or Normal distribution. Instead, the distribution may be composed of a potentially infinite combination of distributions.

More formally, we say that a model is nonparametric if the corresponding parameter space has infinite dimension. We call the field nonparametric but infinitely parametric may be a more appropriate description.

In this section we provide an overview of the various tools we borrow from Bayesian nonparametrics and how they relate to one another.

2.1 Completely Random Measures

We begin with the definition of Kingman's completely random measures (1967). Let $\mathbb{X}$ be a complete and separable metric space endowed with Borel σ-algebra $\mathscr{H}$. We denote by $\mathcal{M}_{\mathbb{X}}$ the space of boundedly finite measures on $(\mathbb{X}, \mathscr{H})$ and by $\mathscr{M}_{\mathbb{X}}$ the corresponding Borel σ-algebra on $\mathscr{M}_{\mathbb{X}}$.

Definition 1 (Completely random measure)**.** Let $\tilde{\mu}$ be a measurable mapping from probability space $(\Omega, \mathscr{F}, \mathbb{P})$ into $(\mathcal{M}_{\mathbb{X}}, \mathscr{M}_{\mathbb{X}})$ such that for any mutually exclusive $A_1, A_2, \ldots, A_n$ in $\mathscr{H}$, the random variables $\tilde{\mu}(A_1), \tilde{\mu}(A_2), \ldots, \tilde{\mu}(A_n)$ are mutually independent. Then $\tilde{\mu}$ is a *completely random measure* (CRM).

CRMs play an important role in Bayesian nonparametrics because they can be used to define nonparametric priors, as demonstrated by Lijoi and Prünster (2009). As detailed by

Griffin and Leisen (2014), we can always represent a CRM on $\mathbb{X}$ as the sum of two components. One is itself a completely random measure $\tilde{\mu}_c = \sum_{i=1}^{\infty} J_i \delta_{X_i}$, where both the positive jumps J_i and their locations X_i in $\mathbb{X}$ are random. The other is a measure whose masses $V_1, V_2, \ldots, V_M$ are again random but appear at fixed locations $x_1, x_2, \ldots, x_M \in \mathbb{X}$. Accordingly

$$\tilde{\mu} = \tilde{\mu}_c + \sum_{i=1}^{M} V_i \delta_{x_i}. \tag{1}$$

The random variables $V_1, V_2, \ldots, V_M$ are mutually independent and independent of $\tilde{\mu}_c$ and $M \in \{1, 2, \ldots\} \cup \{\infty\}$.

It is important to note that $\tilde{\mu}_c$ is characterised by the *Lévy-Khintchine* representation, which states that

$$\mathbb{E}\left[\mathrm{e}^{-\int_{\mathbb{X}} f(x)\tilde{\mu}_c(\mathrm{d}x)}\right] = \exp\left\{-\int_{\mathbb{R}^+ \times \mathbb{X}} \left[1 - \mathrm{e}^{-sf(x)}\right] \nu(\mathrm{d}s, \mathrm{d}x)\right\} \tag{2}$$

where $f : \mathbb{X} \to \mathbb{R}$ is a measurable function such that $\int |f| \mathrm{d}\tilde{\mu}_c < \infty$ with probability one and ν is a measure on $\mathbb{R}^+ \times \mathbb{X}$ such that

$$\int_B \int_{\mathbb{R}^+} \min\{s, 1\} \nu(\mathrm{d}s, \mathrm{d}x) < \infty$$

for any B in $\mathscr{H}$.

The measure ν is called the *Lévy intensity* of $\tilde{\mu}_c$ and fully characterises it. It will be useful too to express ν in terms of its two parts, which contain the information about $\tilde{\mu}_c$'s jumps and their locations respectively:

$$\nu(\mathrm{d}s, \mathrm{d}x) = \rho_x(\mathrm{d}s)\alpha(\mathrm{d}x).$$

For our purposes, it will be sufficient to consider the case where $\rho_x = \rho$ and so the distribution of the jumps of $\tilde{\mu}_c$ is independent of their location. In this case we say both ν and $\tilde{\mu}_c$ are *homogeneous*:

$$\nu(\mathrm{d}s, \mathrm{d}x) = \rho(s)\mathrm{d}s\alpha(\mathrm{d}x). \tag{3}$$

CRMs are useful to us as building blocks for our model largely because of their relation to neutral-to-the-right processes, which we explore in the following section.

2.2 Neutral-to-the-Right Priors

Neutral-to-the-right processes provide a method of reconciling CRMs with the distribution functions which directly describe the behaviour of our data.

Definition 2. A random distribution function $\tilde{F}$ on $\mathbb{R}^+$ is *neutral-to-the-right* (NTR) if, for any $0 \leq t_1 < t_2 < \cdots < t_k < \infty$ and any $k \geq 1$, the random variables

$$\tilde{F}(t_1), \frac{\tilde{F}(t_2) - \tilde{F}(t_1)}{1 - \tilde{F}(t_1)}, \ldots, \frac{\tilde{F}(t_k) - \tilde{F}(t_{k-1})}{1 - \tilde{F}(t_{k-1})}$$

are independent.

It is not immediately apparent from this definition how NTR processes might relate to CRMs. The following theorem by Doksum (1974) reveals a crucial connection.

Theorem 1. *A random distribution function* $\tilde{F} = \{\tilde{F}(t) : t \geq 0\}$ *is NTR if and only if it has the same probability density (p.d.) as the process* $\{1 - \mathrm{e}^{-\tilde{\mu}((0,t])} : t \geq 0\}$*, for some CRM* $\tilde{\mu}$ *on* $\mathbb{X} = \mathbb{R}^+$ *such that* $\mathbb{P}\left[\lim_{t\to\infty} \tilde{\mu}((0,t]) = \infty\right] = 1$.

This result means it is possible for us to characterise the prior distribution of $\tilde{F}$ in terms of the Lévy intensity ν corresponding to $\tilde{\mu}$. Here we note that the CRM $\tilde{\mu}$ we consider for the prior distribution of $\tilde{F}$ is without fixed points of discontinuity.

The corresponding posterior, however, will contain jumps at fixed points, as per the following theorem, again proved by Doksum (1974).

Theorem 2. *If* $\tilde{F}$ *is NTR(*$\tilde{\mu}$*), then the posterior distribution of* $\tilde{F}$*, given the data* $X_1, X_2, \ldots, X_n$*, is NTR(*$\tilde{\mu}^*$*) where* $\tilde{\mu}^*$ *is a CRM with fixed points of discontinuity.*

This useful and powerful property of NTR priors is known as *structural conjugacy.* Here it is important to note that this does not imply parametric conjugacy. Parametric conjugacy occurs when the posterior process has the same probability distribution as the prior process but with updated parameters.

The more general structural conjugacy describes the case where the posterior process belongs to the same class of random probability measures as its prior process. While parametric conjugacy implies structural conjugacy, the converse is not true in general. This distinction is made clear in Lijoi and Prünster (2009).

Putting these elements together, we can see already how this framework might be useful by simply considering the expected value of some NTR prior $\tilde{F}$, whose corresponding CRM $\tilde{\mu}$ is homogeneous.

$$\mathbb{E}\left[\tilde{F}(t)\right] = 1 - \mathbb{E}\left[\mathrm{e}^{-\tilde{\mu}((0,t])}\right] = 1 - \mathrm{e}^{-\int_{(0,t]}\int_{\mathbb{R}^+}\left[1-\mathrm{e}^{-s}\right]\rho(\mathrm{d}s)\alpha(\mathrm{d}x)}. \tag{4}$$

This kind of transformation will be used extensively when we derive our main results in section 4.

2.3 Survival Analysis

Now we depart momentarily from Bayesian nonparametrics and turn our attention to survival analysis. To understand the purpose of survival analysis, we first consider the properties of survival data.

2.3.1 Survival Data

Survival data takes values on the positive real line. Each data point usually defines some "time to failure". This may describe the onset of a disease, the death of a patient, or, in the case of our motivating example, the discovery of a bug in software. Consider a group of N entities, with failure times $x_1, x_2, \ldots, x_N$. We model these using random variables $X_1, X_2, \ldots, X_N$.

A key aspect of survival analysis is the concept of censoring. Experiments in survival analysis are often hindered by circumstance in how much data can be measured exactly. Typically, there is a certain window of time inside which a number $M \leq N$ of exact failure times are observed. Failure times which are not observed inside this window are *censored.*

The most common type of censoring in survival analysis, and the one we shall confine ourselves to here, is right censoring. In this case, each data point has an associated censoring time W_i. Censored data are those observations whose exact failure times are unknown because they have occured *after* their censoring times. We now must consider a new set of random variables $T_1, T_2, \ldots, T_N$ where

$$T_i = \min\{X_i, W_i\}, \quad i = 1, 2, \ldots, N.$$

The data itself may be continuous or discrete and usually comprises the total number of data points N, the exact failure times of M data points $T_1, T_2, \ldots, T_M$, and the censoring times W_i. We have that $W_i \geq T_i > 0,\ i = 1, 2, \ldots, M$.

The type of data we wish to describe is subtly different in two respects. Firstly, our data comes from a number $r \geq 1$ of distinct groups. Each group has a censoring time W_l. Secondly, the total number of data points N_l in the l^{th} group is unobservable and so must be treated by the model as a random variable.

In the case of the Firefox data, N_l is not observable because it represents the total number of bugs in an extensive software package which is only tested and used during a very limited period of time.

3 Model specification

We consider r releases of the same software and call respectively N_l and M_l the (unknown) total number of bugs and the observed number of discovered bugs for the lth release, with $l = 1, 2, \ldots, r$ and $N_l \geq M_l$. We suppose that the observation window for the lth release is $(0, W_l]$, meaning that the lth software release was tested starting from time 0 (e.g. this could be the moment the version was released) to some fixed time W_l.

The M_l observations for the lth release are $(T_{1,l}, T_{2,l}, \ldots, T_{M_l,l})$ where

$$T_{1,l} \leq T_{2,l} \leq \ldots \leq T_{M_l,l} \leq W_l.$$

We denote the N_l discovery times for the lth release by $X_{1,l} \leq X_{2,l} \leq \ldots \leq X_{N_l,l}$ and recall that all these observations are subject to deterministic right-censoring at W_l. Thus, in order to link the two sets of notation we introduced, we have

$$T_{j,l} = \min\{X_{j,l}, W_l\} \qquad \text{for } j = 1, 2, \ldots, N_l,$$

where the first M_l observations are exact and the remaining $N_l - M_l$ (unobserved) observations are right-censored and they all coincide to W_l, that is $T_{M_l+1,l} = T_{M_l+2,l} = \ldots = T_{N_l,l} = W_l$. For future convenience we also introduce the notation $N_l' = N_l - M_l$, to denote the unobserved number of censored bugs of the lth release.

The main idea is to replicate the model of Wilson and Samaniego (2007) in a multivariate setting by allowing borrowing of information across releases in order to better model the discovery distributions of subsequent releases. To this end we will make use of Griffith-Milne (GM) dependent neutral-to-the-right priors defined in Nipoti (2011) (see also Lijoi et al. (2014) for allied approach, and Epifani and Lijoi (2010) for an alternative definition of dependent NTR priors based on the use of Lévy copulas). Namely we define dependent nonparametric priors for distribution functions F_l, with $l = 1, 2, \ldots, r$, as

$$\tilde{F}_l(t) = 1 - \exp\{-\tilde{\mu}_l((0, t])\} = 1 - \exp\{-\mu_l((0, t]) - \mu_0((0, t])\}, \tag{5}$$

where the completely random measures $\mu_0, \mu_1, \ldots, \mu_r$ are independent and such that $\mu_1, \mu_2, \ldots, \mu_r$ are all identically distributed with Lévy intensity $z\nu(\mathrm{d}s, \mathrm{d}y)$, whereas the Lévy intensity of μ_0 is $(1-z)\nu(\mathrm{d}s, \mathrm{d}y)$, for some $z \in (0, 1)$. This leads to a sequence of r completely random measures $\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_r$ that are dependent and identically distributed.

The purpose of the parameter z is to provide further control over the model, specifically the degree of information sharing we wish to allow. Consider (5). As z approaches 1, the measure μ_0, which is common to each F_l, $l = 1, 2, \ldots, r$, loses significance and we approach independence between groups. Conversely, as z approaches 0, $\mu_1, \mu_2, \ldots, \mu_r$ lose significance and we grow closer to complete exchangeability of random variables within and between groups.

The joint independence of increments of the vector $(\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_r)$ guarantees analytical tractability. We will consider homogeneous completely random measures, that is we assume that ν admits the following factorisation

$$\nu(\mathrm{d}s, \mathrm{d}y) = \rho(s)\mathrm{d}s\alpha(\mathrm{d}y),$$

for some kernel ρ and some measure $\alpha = cP_0$, where P_0 is a probability distribution defined on $\mathbb{R}^+$.

The model can be written as

$$\begin{aligned} T_{j,1} \mid \tilde{F}_1 &\overset{iid}{\sim} \tilde{F}_1 \qquad \text{for } j = 1, 2, \ldots, N_1 \\ T_{j,2} \mid \tilde{F}_2 &\overset{iid}{\sim} \tilde{F}_2 \qquad \text{for } j = 1, 2, \ldots, N_2 \\ &\vdots \\ T_{j,r} \mid \tilde{F}_r &\overset{iid}{\sim} \tilde{F}_r \qquad \text{for } j = 1, 2, \ldots, N_r \\ (T_{j_1,1}, T_{j_2,2}, \ldots, T_{j_r,r}) \mid \tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_r &\sim \tilde{F}_1 \times \tilde{F}_2 \times \cdots \times \tilde{F}_r \\ &\text{for } j_l = 1, 2, \ldots, N_l \text{ and } l = 1, 2, \ldots, r. \end{aligned} \tag{6}$$

Assuming this is equivalent to assuming a partially exchangeable framework where the observations are exchangeable within but not across releases. The last line of model (6) says that, conditionally on the random distribution functions $\tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_r$, discovery times from different releases are independent. The dependency is introduced when defining the random distribution functions by using the common completely random measure μ_0. At the same time, the release-specific components $\mu_1, \mu_2, \ldots, \mu_r$ take into account the heterogeneity characterising different software releases.

For the sake of compactness we introduce the following notation:

$$\begin{aligned}
\tilde{\boldsymbol{\mu}} &= (\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_r) \\
\tilde{\mathbf{F}} &= (\tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_r) \\
\mathbf{N} &= (N_1, N_2, \ldots, N_r) \\
\mathbf{M} &= (M_1, M_2, \ldots, M_r) \\
\mathbf{W} &= (W_1, W_2, \ldots, W_r) \\
\mathbf{T}_l^{(M_l)} &= (T_{1,l}, T_{2,l}, \ldots, T_{M_l,l}) \qquad \text{for } l = 1, 2, \ldots, r \\
\mathbf{T}^{(\mathbf{M})} &= (\mathbf{T}_1^{(M_1)}, \mathbf{T}_2^{(M_2)}, \ldots, \mathbf{T}_r^{(M_r)}) \\
\mathbf{T}_l^{(N_l)} &= (T_{1,l}, T_{2,l}, \ldots, T_{M_l,l}, T_{M_l+1,l} = W_i, \ldots, T_{N_l,l} = W_l) \qquad \text{for } l = 1, 2, \ldots, r \\
\mathbf{T}^{(\mathbf{N})} &= (\mathbf{T}_1^{(N_1)}, \mathbf{T}_2^{(N_2)}, \ldots, \mathbf{T}_r^{(N_r)})
\end{aligned}$$

Notice that while $\mathbf{T}_l^{(M_l)}$ refers to the M_l-dimensional vector of observed discovery times, $\mathbf{T}_l^{(N_l)}$ is N_l-dimensional and composed by the observed M_l discovery times together with the unobserved N_l' right-censored discovery times, with N_l' being unobservable.

Similarly to Wilson and Samaniego (2007), given $\mathbf{N}$ and $\tilde{\mathbf{F}}$, for any $l = 1, 2, \ldots, r$, the number of discovery times M_l observed not later than W_l is assumed as being binomial with probability parameter equal to $\tilde{F}_l(W_l)$, where $\tilde{F}_l(W_l)$ is the probability that an event distributed according to $\tilde{F}_l$ occurs before W_l, that is the probability for a bug of the l^{th} release to be actually observed. Given this, we can write the likelihood for the data $\mathbf{T}^{(\mathbf{M})}$ as

$$L(\tilde{\mathbf{F}}, \mathbf{N} \mid \mathbf{T}^{(\mathbf{M})}) = \prod_{l=1}^{r} \frac{N_l!}{N_l'!}(1 - \tilde{F}_l(W_l))^{N_l'} \prod_{j=1}^{M_l}(\tilde{F}_l(T_{j,l}) - \tilde{F}_l(T_{j,l}^-)),$$

where $\tilde{F}_l(T^-) = \lim_{\varepsilon \to 0^+} \tilde{F}_l(T - \varepsilon)$.

We complete the model specification by assigning a prior distribution to the vector $\mathbf{N}$. As in Wilson and Samaniego (2007), for the sake of simplicity we assume that such vector is independent of $\tilde{\mathbf{F}}$. We assume that the total numbers of bugs of each release are independent and identically distributed Poisson random variables.

$$N_l \overset{iid}{\sim} \text{Pois}(\lambda) \qquad \text{for } l = 1, 2, \ldots, r.$$

It is straightforward to extend the model to a case where different priors (that is different λ) are specified for different releases.

4 Main Results

The final goal of our inference is the pointwise estimation of $\mathbb{E}[\tilde{F}_l(t) \mid \mathbf{T}^{(\mathbf{M})}]$, for every $l = 1, 2, \ldots, r$. The main result that will allow us to achieve this goal is a characterisation of the posterior distribution of $\tilde{F}_l$. Namely, after introducing some set of suitable auxiliary random variables $\mathbf{V}$, we will derive a closed form expression for $\mathbb{E}[\tilde{F}_l(t) \mid \mathbf{T}^{(\mathbf{M})}, \mathbf{N}, \mathbf{V}]$. In order to marginalise with respect to the (unobservable) random vectors $\mathbf{N}$ and $\mathbf{V}$ and thus evaluate $\mathbb{E}[\tilde{F}_l(t) \mid \mathbf{T}^{(\mathbf{M})}]$, we will devise a Gibbs sampling algorithm exploiting the availability of the full conditional distributions of $\mathbf{N}$ and $\mathbf{V}$.

4.1 Characterisation of the posterior distribution of $\tilde{\mathbf{F}}$

We derive here a characterisation of the posterior distribution of $\tilde{\mathbf{F}}$. To this end we start by observing that the almost sure discreteness of the elements of $\tilde{\mu}$ implies that the corresponding random distributions in $\tilde{\mathbf{F}}$ are almost surely discrete and, thus, for $i \neq j$, $P[T_{i,l} = T_{j,l}] > 0$ for every $l = 1, 2, \ldots, r$ and $i, j = 1, 2, \ldots, M_l$. Moreover, by definition, the elements of the vector $\tilde{\boldsymbol{\mu}}$ share a common component μ_0 and, therefore, for $l_1 \neq l_2$, $P[T_{i,l_1} = T_{j,l_2}] > 0$, for any $i = 1, 2, \ldots, M_{l_1}$ and $j = 1, 2, \ldots, M_{l_2}$.

Possible coincidences among the observations induce a partition structure within the vector $\mathbf{T}^{(\mathbf{M})}$ (and thus within the vector of observed and unobserved data $\mathbf{T}^{(\mathbf{N})}$): let us assume that there are k distinct observations, including censoring times W_l, among $\mathbf{T}^{(\mathbf{N})}$ and record them in an increasing order as $T_1^* < T_2^* < \ldots < T_k^*$. We call $n_{i,l} \geq 0$ the number of exact observations referring to the l^{th} release, with $l = 1, 2, \ldots, r$, that coincide with T_i^*. We introduce the notation $\bar{n}_i := \sum_{l=1}^{r} n_{i,l}$ and observe that $\bar{n}_i > 0$ for every $i = 1, 2, \ldots, k$, that $\sum_{i=1}^{k} n_{i,l} = M_l$. For future convenience we set $T_0^* := 0$ and we introduce the notation $\bar{n}_{j,l} := \sum_{i=j}^{k-1} n_{i,l}$.

Finally we partition the set of indices $\{1, 2, \ldots, k\}$ into $r + 1$ subsets, namely

1. $\mathcal{I}_0 = \{j \,:\, \exists\, l_1, l_2 \text{ s.t. } l_1 \neq l_2, n_{j,l_1} n_{j,l_2} > 0\}$
2. $\mathcal{I}_l = \{j \,:\, n_{j,l} > 0, \sum_{g=1,\, g\neq l}^{r} n_{j,g} = 0\}$, for $l = 1, 2, \ldots, r$.

3. $\mathcal{I}_W = \{j : \sum_{g=1}^r n_{j,g} = 0\}$.

That is we have one subset $\mathcal{I}_0$ consisting of all the distinct values that are shared by at least two releases, a subset $\mathcal{I}_l$ consisting of all the distinct values that appear only in the l^{th} release for every $l = 1, 2, \ldots, r$, and a subset of distinct values coinciding with no exact observations but only with censoring times $W_l,\ l = 1, 2, \ldots, r$.

We introduce r vectors of sequences of Bernoulli auxiliary random variables $\mathbf{V}_l = (V_{j_l,l})_{j_l \geq 1}$, with $l = 1, 2, \ldots, r$, such that

$$P[V_{j_l,l} = 0] = 1 - P[V_{j_l,l} = 1] = z \qquad \text{for } l = 1, 2, \ldots, r; \quad j_l = 1, 2, \ldots$$

We further introduce the notation $\mathbf{V} = (\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_r)$ for the vector of sequences of auxiliary random variables. The next result provides a characterisation of the posterior distribution of $\tilde{\mathbf{F}}$, specifically it provides the distribution of $\tilde{\mathbf{F}}$ conditionally on $\mathbf{T}^{(\mathbf{M})}$, $\mathbf{N}$ and $\mathbf{V}$.

Theorem 3. *Let $\tilde{\mathbf{F}}$ be a vector of dependent neutral-to-the-right random distributions defined as in* (5). *Given $\mathbf{T}^{(\mathbf{M})}$, $\mathbf{N}$, and $\mathbf{V}$, the posterior distribution of $\tilde{\mathbf{F}}$ coincides with the distribution of a vector of neutral-to-the-right random distribution functions $\tilde{\mathbf{F}}' = (\tilde{F}_1', \tilde{F}_2', \ldots, \tilde{F}_r')$ such that, for $l = 1, 2, \ldots, r$, $\tilde{F}_l'(t) = 1 - \exp\{-\tilde{\mu}_l'((0,t])\}$, where*

$$(\tilde{\mu}_1', \tilde{\mu}_2', \ldots, \tilde{\mu}_r') = (\tilde{\mu}_1^*, \tilde{\mu}_2^*, \ldots, \tilde{\mu}_r^*) + \sum_{l=1}^r \sum_{i \in \mathcal{I}_l} \Big(\underbrace{V_{i,l}, \ldots, V_{i,l}}_{l-1}, 1, \underbrace{V_{i,l}, \ldots, V_{i,l}}_{r-l} \Big) J_{i,l} \delta_{T_i^*} + \sum_{i \in \mathcal{I}_0} \Big(\underbrace{1, 1, \ldots, 1}_{r} \Big) J_{i,0} \delta_{T_i^*}, \qquad (7)$$

where $(\tilde{\mu}_1^*, \tilde{\mu}_2^*, \ldots, \tilde{\mu}_r^*)$ are CRMs without fixed points of discontinuity and such that

$$\tilde{\mu}_l^* = \mu_l^* + \mu_0^*, \qquad \text{for } l = 1, 2, \ldots, r,$$

with $\mu_0^*, \mu_1^*, \ldots, \mu_r^*$ being independent and such that each μ_l^*, for $l = 0, 1, \ldots, r$, has Lévy intensity ν_l^* defined as

$$\nu_0^*(\mathrm{d}s, \mathrm{d}y) = (1 - z) \mathrm{e}^{-s \sum_{i=1}^k \sum_{l=1}^r \left(\bar{n}_{i,l} + N_l' \mathbb{1}_{\{i \leq k_l\}} \right) \mathbb{1}_{(t_{i-1}^*, t_i^*]}(y)} \nu(\mathrm{d}s, \mathrm{d}y)$$

$$\nu_l^*(\mathrm{d}s, \mathrm{d}y) = z\mathrm{e}^{-s\sum_{i=1}^{k}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\mathbb{1}_{(t_{i-1}^*, t_i^*]}(y)}\nu(\mathrm{d}s, \mathrm{d}y), \qquad l = 1, 2, \dots, r.$$

Moreover the jumps $J_{i,l}$, with $l = 0, 1, \dots, r$ and $i = 1, 2, \dots, k_l$ are mutually independent and independent of $(\tilde{\mu}_1^*, \tilde{\mu}_2^*, \dots, \tilde{\mu}_r^*)$. For every $l = 1, 2, \dots, r$ and $i = 1, 2, \dots, k_l$, the jump $J_{i,l}$ has density $f_{J_{i,l}}(s)$ proportional to

$$\mathrm{e}^{-s\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}+V_{i,l}\sum_{j=1,\, j\neq l}^{r}\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\right)}\left(\mathrm{e}^s - 1\right)^{n_{i,l}}\rho(s),$$

while, for $i = 1, 2, \dots, k_0$, the jump $J_{i,0}$ has density $f_{J_{i,0}}(s)$ proportional to

$$\mathrm{e}^{-s\sum_{l=1}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s - 1\right)^{\bar{n}_i}\rho(s),$$

where we recall that $\bar{n}_i := \sum_{l=1}^{r} n_{i,l}$.

Proof. The joint distribution of $(\tilde{\mu}_l, T_{i,l})$ is given, for every $l = 1, 2, \dots, r$, by

$$\begin{aligned}
\mathrm{p}(\tilde{\mu}_l, T_{i,l}) &= \mathrm{p}(T_{i,l} \mid \tilde{\mu}_l)\mathrm{p}(\tilde{\mu}_l) \\
&= \exp\{-\tilde{\mu}_l(0, T_{i,l}]\}\left(\mathbb{1}_{\{i\le M_l\}}(\exp\{\tilde{\mu}_l(T_{i,l}^-, T_{i,l}]\} - 1) + \mathbb{1}_{\{i>M_l\}}\right)\mathrm{p}(\tilde{\mu}_l).
\end{aligned}$$

where we define $T_{i,l}^- := T_{i,l} - \varepsilon$, for ε arbitrarily small.

The joint distribution of $(\tilde{\boldsymbol{\mu}}, \mathbf{N}, \mathbf{T}^{(\mathbf{M})})$ is given, for every $l = 1, 2, \dots, r$, by

$$\begin{aligned}
\mathrm{p}(\tilde{\boldsymbol{\mu}}, \mathbf{N}, \mathbf{T}^{(\mathbf{M})}) &= L(\tilde{\boldsymbol{\mu}}, \mathbf{N} \mid \mathbf{T}^{(\mathbf{M})})\mathrm{p}(\tilde{\boldsymbol{\mu}})\mathrm{p}(\mathbf{N}) \\
&= \prod_{l=1}^{r}\frac{(N_l' + M_l)!}{N_l'!}\exp\{-N_l'\tilde{\mu}_l(0, W_l]\} \\
&\times \prod_{i=1}^{k-1}\exp\{-n_{i,l}\tilde{\mu}_l(0, T_i^*]\}\left(\exp\{\tilde{\mu}_l(T_i^{*-}, T_i^*]\} - 1\right)^{n_{i,l}}\mathrm{p}(\tilde{\boldsymbol{\mu}})\mathrm{p}(\mathbf{N}).
\end{aligned}$$

We want to characterize the distribution of $\tilde{\boldsymbol{\mu}}$ given $\mathbf{T}^{(\mathbf{M})}$ and $\mathbf{N}$, via its Laplace functional transform. This is obtained if one can determine

$$\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(f_l)\} \mid \mathbf{N}, \mathbf{T}^{(\mathbf{M})}\right] = \frac{\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(f_l)\}\mathrm{p}(\mathbf{T}^{(\mathbf{M})} \mid \tilde{\boldsymbol{\mu}}, \mathbf{N})\right]}{\mathbb{E}\left[\mathrm{p}(\mathbf{T}^{(\mathbf{M})} \mid \tilde{\boldsymbol{\mu}}, \mathbf{N})\right]},$$

for every set of measurable functions $(f_1, f_2, \dots, f_r)$ from $\mathbb{R}^+$ to $\mathbb{R}$, where the expectations are taken with respect to $\tilde{\boldsymbol{\mu}}$. We confine ourselves to consider $f_l(t) = \lambda_l\mathbb{1}_{(0,t]}$, that is

$$\frac{\mathbb{E}\left[\exp\left\{-\sum_{l=1}^{r}\lambda_l\tilde{\mu}_l(0, t]\right\}\mathrm{p}(\mathbf{T}^{(\mathbf{M})} \mid \tilde{\boldsymbol{\mu}}, \mathbf{N})\right]}{\mathbb{E}\left[\mathrm{p}(\mathbf{T}^{(\mathbf{M})} \mid \tilde{\boldsymbol{\mu}}, \mathbf{N})\right]}, \tag{8}$$

and observe that this characterises the conditional distribution of $\tilde{\boldsymbol{\mu}}$ given $\mathbf{N}$ and $\mathbf{T}^{(\mathbf{M})}$ since $\tilde{\boldsymbol{\mu}}$ has jointly independent increments.

In the following we set $T_0^* := 0$ and, for any $i = 1, \dots, k$, define

$$I_i := (T_{i-1}^*, T_i^{*-}] \quad \text{and} \quad L_i := (T_i^{*-}, T_i^*].$$

Moreover, we set $I_{k+1} := (T_k^*, \infty)$. Since ε is arbitrarily small, time t may either coincide with some observation T_i^*, for $i = 1, \dots, k$, or be in some interval I_i, with $i = 1, \dots, k+1$. Both situations may be tackled with the same technique. We focus on the second case as it is the one that requires more care, giving rise to a finer partition of the interval $(0, \infty)$. Assume that $t \in I_{i_0}$ for some $i_0 \in \{1, \dots, k+1\}$, that is $T_{i_0-1}^* < t \leq T_{i_0}^{*-}$, and consider the partition of $(0, \infty)$ defined as

$$(0, \infty) = \left(\cup_{i=1,\, i \neq i_0}^{k+1} I_i\right) \cup \left(\cup_{i=1}^{k} L_i\right) \cup (T_{i_0-1}^*, t] \cup (t, T_{i_0}^{*-}].$$

Then, for every $l = 1, 2, \dots, r$, we can write $\tilde{\mu}_l(0, t]$ as

$$\tilde{\mu}_l(0, t] = \sum_{i=1}^{i_0-1} (\tilde{\mu}_l(I_i) + \tilde{\mu}_l(L_i)) + \tilde{\mu}_l(T_{i_0-1}^*, t]. \tag{9}$$

Similarly for any $j = 1, 2, \dots, k$ we can write $\tilde{\mu}_l(0, T_j^*]$ as

$$\tilde{\mu}_l(0, T_j^*] = \sum_{i=1,\, i \neq i_0}^{j} \tilde{\mu}_l(I_i) + \sum_{i=1}^{j} \tilde{\mu}_l(L_i) + \left(\tilde{\mu}_l(T_{i_0-1}^*, t] + \tilde{\mu}_l(t, T_{i_0}^{*-}]\right) \mathbb{1}_{\{i_0 \leq j\}}. \tag{10}$$

We now define k_l such that $T_{k_l}^* = W_l$. Using (9) and (10) we can rewrite the numerator of (8) as

$$\prod_{l=1}^{r} \frac{(N_l' + M_l)!}{N_l'!} \mathbb{E}\left[\exp\{-\mathcal{G}(\tilde{\boldsymbol{\mu}}, \mathbf{N}, \mathbf{T}^{(\mathbf{M})})\} \prod_{i=1}^{k-1} \prod_{l=1}^{r} (\exp\{\tilde{\mu}_l(L_i)\} - 1)^{n_{i,l}} \mid \tilde{\mu}, \mathbf{T}^{(\mathbf{M})}\right], \tag{11}$$

where $\mathcal{G}(\tilde{\boldsymbol{\mu}}, \mathbf{N}, \mathbf{T}^{(\mathbf{M})})$ is defined as

$$\mathcal{G}(\tilde{\boldsymbol{\mu}}, \mathbf{N}, \mathbf{T}^{(\mathbf{M})}) = \sum_{l=1}^{r} \left[\lambda_l \left(\sum_{i=1}^{i_0-1} (\tilde{\mu}_l(I_i) + \tilde{\mu}_l(L_i)) + \tilde{\mu}_l(T_{i_0-1}^*, t]\right)\right.$$

$$+\sum_{i=1}^{k-1} n_{i,l}\left(\sum_{j=1,\, j\neq i_0}^{i} \tilde{\mu}_l(I_j) + \sum_{j=1}^{i} \tilde{\mu}_l(L_j) + \left(\tilde{\mu}_l(T^*_{i_0-1}, t] + \tilde{\mu}_l(t, T^{*-}_{i_0}]\right) \mathbb{1}_{\{i_0\leq i\}}\right)$$

$$+N'_l\left(\sum_{i=1,\, i\neq i_0}^{k_l} \tilde{\mu}_l(I_i) + \sum_{i=1}^{k_l} \tilde{\mu}_l(L_i) + \left(\tilde{\mu}_l(T^*_{i_0-1}, t] + \tilde{\mu}_l(t, T^{*-}_{i_0}]\right) \mathbb{1}_{\{i_0\leq k_l\}}\right)\Bigg]$$

$$= \sum_{l=1}^{r}\Bigg[\lambda_l\left(\sum_{i=1,\, i\neq i_0}^{k-1} \mathbb{1}_{\{i\leq i_0-1\}}\tilde{\mu}_l(I_i) + \sum_{i=1}^{k-1} \mathbb{1}_{\{i\leq i_0-1\}}\tilde{\mu}_l(L_i) + \tilde{\mu}_l(T^*_{i_0-1}, t]\right)$$

$$+ \sum_{i=1,\, i\neq i_0}^{k-1} \sum_{j=i}^{k-1} n_{j,l}\tilde{\mu}_l(I_i) + \sum_{i=1}^{k-1}\sum_{j=i}^{k-1} n_{j,l}\tilde{\mu}_l(L_i) + \sum_{j=i_0}^{k-1} n_{j,l}\tilde{\mu}_l(T^*_{i_0-1}, t] + \sum_{j=i_0}^{k-1} n_{j,l}\tilde{\mu}_l(t, T^{*-}_{i_0}]$$

$$+N'_l\left(\sum_{i=1,\, i\neq i_0}^{k_l} \tilde{\mu}_l(I_i) + \sum_{i=1}^{k_l} \tilde{\mu}_l(L_i) + \mathbb{1}_{\{i_0\leq k_l\}}\tilde{\mu}_l(T^*_{i_0-1}, t] + \mathbb{1}_{\{i_0\leq k_l\}}\tilde{\mu}_l(t, T^{*-}_{i_0}]\right)\Bigg].$$

We recall the notation $\bar{n}_{j,l} = \sum_{i=j}^{k-1} n_{i,l}$, with the proviso that $\bar{n}_{k,l} = \bar{n}_{k+1,l} = 0$, and we rewrite $\mathcal{G}(\tilde{\boldsymbol{\mu}}, \mathbf{N}, \mathbf{T}^{(\mathbf{M})})$ as

$$\sum_{l=1}^{r}\Bigg[\sum_{i=1,\, i\neq i_0}^{k} \tilde{\mu}_l(I_i)\left(\lambda_l \mathbb{1}_{\{i\leq i_0-1\}} + \bar{n}_{i,l} + N'_l\mathbb{1}_{\{i\leq k_l\}}\right)$$

$$+\sum_{i=1}^{k} \tilde{\mu}_l(L_i)\left(\lambda_l \mathbb{1}_{\{i\leq i_0-1\}} + \bar{n}_{i,l} + N'_l\mathbb{1}_{\{i\leq k_l\}}\right)$$

$$+ \tilde{\mu}_l(T^*_{i_0-1}, t]\left(\lambda_l + \bar{n}_{i_0,l} + N'_l\mathbb{1}_{\{i_0\leq k_l\}}\right)$$

$$+\tilde{\mu}_l(t, T^{*-}_{i_0}]\left(\bar{n}_{i_0,l} + N'_l\mathbb{1}_{\{i_0\leq k_l\}}\right)\Bigg]$$

Thus, by exploiting the independence of increments of $\tilde{\mu}_l$, we can write (11) as

$$\prod_{l=1}^{r}\frac{(N'_l+M_l)!}{N'_l!}\prod_{i=1,\, i\neq i_0}^{k} \mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(I_i)\left(\lambda_l \mathbb{1}_{\{i\leq i_0-1\}} + \bar{n}_{i,l} + N'_l\mathbb{1}_{\{i\leq k_l\}}\right)\}\right]$$

$$\times\prod_{i=1}^{k}\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(L_i)\left(\lambda_l \mathbb{1}_{\{i\leq i_0-1\}} + \bar{n}_{i,l} + N'_l\mathbb{1}_{\{i\leq k_l\}}\right)\}\left(\exp\{\tilde{\mu}_l(L_i)\}-1\right)^{n_{i,l}}\right]$$

$$\times\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(T^*_{i_0-1}, t]\left(\lambda_l + \bar{n}_{i_0,l} + N'_l\mathbb{1}_{\{i_0\leq k_l\}}\right)\}\right]$$

$$\times\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(t, T^{*-}_{i_0}]\left(\bar{n}_{i_0,l} + N'_l\mathbb{1}_{\{i_0\leq k_l\}}\right)\}\right]. \quad (12)$$

Next we proceed by considering separately all the expected values appearing in (12) that, as a matter of fact, is composed by two types of quantity, namely

$$\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(I)a_l\}\right] \quad \text{and} \quad \mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(I)a_l\}\left(\exp\{\tilde{\mu}_l(I)\}-1\right)^{m_l}\right], \tag{13}$$

with a_l being real numbers, I a real interval and m_l nonnegative integers. The first type coincides with the joint Laplace functional transform of $(\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_r)$ evaluated at $(a_1\mathbb{1}_I, a_2\mathbb{1}_I, \ldots, a_r\mathbb{1}_I)$. The vector of GM-dependent completely random measures $(\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_r)$ has known closed form joint Laplace transform (see Lijoi et al. (2014)) that leads to

$$\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(I)a_l\}\right] = \exp\left\{-z\alpha(I)\sum_{l=1}^{r}\int_0^\infty \left(1-\exp\{-sa_l\}\right)\rho(s)\mathrm{d}s \right.$$
$$\left. -(1-z)\alpha(I)\int_0^\infty \left(1-\exp\left\{-s\sum_{l=1}^{r}a_l\right\}\right)\rho(s)\mathrm{d}s\right\}.$$

Similarly, the second expected value in (13) can be thought of as the sum of joint Laplace functional transforms:

$$\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(I)a_l\}\left(\exp\{\tilde{\mu}_l(I)\}-1\right)^{m_l}\right]$$
$$= \sum_{i_1=0}^{m_1}\cdots\sum_{i_r=0}^{m_r}(-1)^{\sum_{l=1}^{r} i_l}\binom{m_1}{i_1}\cdots\binom{m_r}{i_r}\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-(a_l-m_l+i_l)\tilde{\mu}_l(I)\}\right].$$

Moreover, for every $i = 1, \ldots, k$, the infinitesimal nature of L_i allows us to use a first order approximation of the exponential function and write

$$\mathbb{E}\left[\prod_{l=1}^{r}\exp\{-\tilde{\mu}_l(L_i)\left(\lambda_l\mathbb{1}_{\{i\le i_0-1\}}+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\}\left(\exp\{\tilde{\mu}_l(L_i)\}-1\right)^{n_{i,l}}\right]$$
$$= \sum_{i_1=0}^{n_{i,1}}\cdots\sum_{i_r=0}^{n_{i,r}}(-1)^{\sum_{l=1}^{r} i_l}\binom{n_{i,1}}{i_1}\cdots\binom{n_{i,r}}{i_r}$$
$$\times\left[1-z\alpha(L_i)\sum_{l=1}^{r}\int_0^\infty\left(1-\mathrm{e}^{-s\left(\lambda_l\mathbb{1}_{\{i\le i_0-1\}}+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}-n_{i,l}+i_l\right)}\right)\rho(s)\mathrm{d}s\right.$$
$$\left.-(1-z)\alpha(L_i)\int_0^\infty\left(1-\mathrm{e}^{-s\sum_{l=1}^{r}\left(\lambda_l\mathbb{1}_{\{i\le i_0-1\}}+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}-n_{i,l}+i_l\right)}\right)\rho(s)\mathrm{d}s\right]+\mathrm{o}\left(\alpha(L_i)\right).$$

Here we stop and consider the possibility that $\sum_{l=1}^r n_{i,l} = 0$ for some i. This case corresponds to an absence of exact observations at T_i^*. In other words, only censoring times occur at this point.

We consider separately two cases. In the first, $n_{i,l} = 0$ for each l. The above expression then reduces to

$$1 - z\alpha(L_i)\sum_{l=1}^r \int_0^\infty \left(1 - \mathrm{e}^{-s\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s$$
$$- (1-z)\alpha(L_i)\int_0^\infty \left(1 - \mathrm{e}^{-s\sum_{l=1}^r\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s + \mathrm{o}\left(\alpha(L_i)\right)$$

In the second case, $\sum_{l=1}^r n_{i,l} > 0$ and we get

$$\alpha(L_i)\left[z\sum_{l=1}^r \int_0^\infty \mathrm{e}^{-s\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s - 1\right)^{n_{i,l}}\rho(s)\mathrm{d}s \mathbb{1}_{\{n_{i,l}>0\,\wedge\, n_{i,j}=0,\, j\ne l\}}\right.$$
$$\left. + (1-z)\int_0^\infty \mathrm{e}^{-s\sum_{l=1}^r\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s - 1\right)^{\sum_{l=1}^r n_{i,l}}\rho(s)\mathrm{d}s\right] + \mathrm{o}\left(\alpha(L_i)\right)$$

Thus (12) can be written as

$$\prod_{l=1}^r \frac{(N_l'+M_l)!}{N_l'!}\prod_{i=1,\, i\ne i_0}^k \exp\left\{-z\alpha(I_i)\sum_{l=1}^r\int_0^\infty\left(1 - \mathrm{e}^{-s\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.$$
$$\left. -(1-z)\alpha(I_i)\int_0^\infty\left(1 - \mathrm{e}^{-s\sum_{l=1}^r\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right\}$$
$$\times\prod_{i=1}^k\left(\left[1 - z\alpha(L_i)\sum_{l=1}^r\int_0^\infty\left(1 - \mathrm{e}^{-s\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.\right.$$
$$\left. -(1-z)\alpha(L_i)\int_0^\infty\left(1 - \mathrm{e}^{-s\sum_{l=1}^r\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right]\mathbb{1}_{\{\bar{n}_i=0\}}$$
$$+\alpha(L_i)\left[z\sum_{l=1}^r\int_0^\infty \mathrm{e}^{-s\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s - 1\right)^{n_{i,l}}\rho(s)\mathrm{d}s \mathbb{1}_{\{n_{i,l}>0\,\wedge\, n_{i,j}=0,\, j\ne l\}}\right.$$
$$\left.\left. +(1-z)\int_0^\infty \mathrm{e}^{-s\sum_{l=1}^r\left(\lambda_l \mathbb{1}_{\{i\le i_0-1\}} + \bar{n}_{i,l} + N_l' \mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s - 1\right)^{\sum_{l=1}^r n_{i,l}}\rho(s)\mathrm{d}s\right]\mathbb{1}_{\{\bar{n}_i>0\}}\right)$$
$$\times\exp\left\{-z\alpha(T_{i_0-1}^*, t]\sum_{l=1}^r\int_0^\infty\left(1 - \mathrm{e}^{-s\left(\lambda_l + \bar{n}_{i_0,l} + N_l' \mathbb{1}_{\{i_0\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.$$
$$\left. -(1-z)\alpha(T_{i_0-1}^*, t]\int_0^\infty\left(1 - \mathrm{e}^{-s\sum_{l=1}^r\left(\lambda_l + \bar{n}_{i_0,l} + N_l' \mathbb{1}_{\{i_0\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right\}$$

$$\times \exp\left\{-z\alpha(t,T_{i_0}^{*-}]\sum_{l=1}^{r}\int_0^\infty \left(1-\mathrm{e}^{-s\left(\bar n_{i_0,l}+N_l'\mathbb{1}_{\{i_0\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.$$

$$\left.-(1-z)\alpha(t,T_{i_0}^{*-}]\int_0^\infty \left(1-\mathrm{e}^{-s\sum_{l=1}^{r}\left(\bar n_{i_0,l}+N_l'\mathbb{1}_{\{i_0\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right\}+\mathrm{o}\left(\prod_{i=1}^{k}\alpha(L_i)\right). \quad (14)$$

Similarly we can show that the denominator of (8) coincides with

$$\prod_{l=1}^{r}\frac{(N_l'+M_l)!}{N_l'!}\prod_{i=1}^{k}\exp\left\{-z\alpha(I_i)\sum_{l=1}^{r}\int_0^\infty \left(1-\mathrm{e}^{-s\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.$$

$$\left.-(1-z)\alpha(I_i)\int_0^\infty \left(1-\mathrm{e}^{-s\sum_{l=1}^{r}\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right\}$$

$$\times\prod_{i=1}^{k}\left(\left[1-z\alpha(L_i)\sum_{l=1}^{r}\int_0^\infty \left(1-\mathrm{e}^{-s\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.\right.$$

$$\left.-(1-z)\alpha(L_i)\int_0^\infty \left(1-\mathrm{e}^{-s\sum_{l=1}^{r}\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right]\mathbb{1}_{\{\bar n_i=0\}}$$

$$+\alpha(L_i)\left[z\sum_{l=1}^{r}\int_0^\infty \mathrm{e}^{-s\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^{s}-1\right)^{n_{i,l}}\rho(s)\mathrm{d}s\mathbb{1}_{\{n_{i,l}>0\,\wedge\, n_{i,j}=0,\,j\neq l\}}\right.$$

$$\left.\left.+(1-z)\int_0^\infty \mathrm{e}^{-s\sum_{l=1}^{r}\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^{s}-1\right)^{\sum_{l=1}^{r}n_{i,l}}\rho(s)\mathrm{d}s\right]\mathbb{1}_{\{\bar n_i>0\}}\right)$$

$$+\mathrm{o}\left(\prod_{i=1}^{k}\alpha(L_i)\right). \quad (15)$$

By combining (14) and (15) we can rewrite (8) as

$$\prod_{i=1}^{i_0-1}\exp\left\{-z\alpha(I_i)\sum_{l=1}^{r}\left[\int_0^\infty \left(1-\mathrm{e}^{-s\left(\lambda_l+\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.\right.$$

$$\left.-\int_0^\infty \left(1-\mathrm{e}^{-s\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right]$$

$$-(1-z)\alpha(I_i)\left[\int_0^\infty \left(1-\mathrm{e}^{-s\sum_{l=1}^{r}\left(\lambda_l+\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.$$

$$\left.\left.-\int_0^\infty \left(1-\mathrm{e}^{-s\sum_{l=1}^{r}\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right]\right\}$$

$$\times\exp\left\{-z\alpha(T_{i_0-1}^{*},t]\sum_{l=1}^{r}\left[\int_0^\infty \left(1-\mathrm{e}^{-s\left(\lambda_l+\bar n_{i_0,l}+N_l'\mathbb{1}_{\{i_0\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.\right.$$

$$\left.-\int_0^\infty \left(1-\mathrm{e}^{-s\left(\bar n_{i_0,l}+N_l'\mathbb{1}_{\{i_0\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right]$$

$$-(1-z)\alpha(T_{i_0-1}^{*},t]\left[\int_0^\infty \left(1-\mathrm{e}^{-s\sum_{l=1}^{r}\left(\lambda_l+\bar n_{i_0,l}+N_l'\mathbb{1}_{\{i_0\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\right.$$

$$
\begin{aligned}
&-\int_0^\infty \left(1-\mathrm{e}^{-s\sum_{l=1}^r\left(\bar n_{i_0,l}+N_l'\mathbb{1}_{\{i_0\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\Big]\Big\}\\
&\times\prod_{i=1}^{i_0-1}\Bigg\{\Bigg(\Bigg[1-z\alpha(L_i)\sum_{l=1}^r\int_0^\infty\left(1-\mathrm{e}^{-s\left(\lambda_l+\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\\
&-(1-z)\alpha(L_i)\int_0^\infty\left(1-\mathrm{e}^{-s\sum_{l=1}^r\left(\lambda_l+\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\Bigg]\mathbb{1}_{\{\bar n_i=0\}}\\
&+\Bigg[z\sum_{l=1}^r\int_0^\infty \mathrm{e}^{-s\left(\lambda_l+\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s-1\right)^{n_{i,l}}\rho(s)\mathrm{d}s\mathbb{1}_{\{n_{i,l}>0\,\wedge\,n_{i,j}=0,\,j\neq l\}}\\
&+(1-z)\int_0^\infty \mathrm{e}^{-s\sum_{l=1}^r\left(\lambda_l+\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s-1\right)^{\sum_{l=1}^r n_{i,l}}\rho(s)\mathrm{d}s\Bigg]\mathbb{1}_{\{\bar n_i>0\}}\Bigg)\\
&\times\Bigg(\Bigg[1-z\alpha(L_i)\sum_{l=1}^r\int_0^\infty\left(1-\mathrm{e}^{-s\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\\
&-(1-z)\alpha(L_i)\int_0^\infty\left(1-\mathrm{e}^{-s\sum_{l=1}^r\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\right)\rho(s)\mathrm{d}s\Bigg]\mathbb{1}_{\{\bar n_i=0\}}\\
&+\alpha(L_i)\Bigg[z\sum_{l=1}^r\int_0^\infty \mathrm{e}^{-s\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s-1\right)^{n_{i,l}}\rho(s)\mathrm{d}s\mathbb{1}_{\{n_{i,l}>0\,\wedge\,n_{i,j}=0,\,j\neq l\}}\\
&+(1-z)\int_0^\infty \mathrm{e}^{-s\sum_{l=1}^r\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\left(\mathrm{e}^s-1\right)^{\sum_{l=1}^r n_{i,l}}\rho(s)\mathrm{d}s\Bigg]\mathbb{1}_{\{\bar n_i>0\}}\Bigg)^{-1}\Bigg\}+\mathrm{o}\left(\prod_{i=1}^k\alpha(L_i)\right).
\end{aligned}
\tag{16}
$$

We now can let $\varepsilon\to 0$, so that each $\alpha(L_i)$ goes to 0 and $(I_1, I_2, \ldots, I_{k+1})$ becomes a partition of $(0,\infty)$. So we obtain an exact expression for (8).

It is important to note that now the terms in the second product of the above expression which appear only when $\bar n_i = 0$ are now each equal to 1. We now consider (16) as the product of two parts. In the first one we can recognise the Laplace functional transform of a vector of completely random measures $(\tilde\mu_1^*, \tilde\mu_2^*, \ldots, \tilde\mu_r^*)$ such that the lth component is defined as

$$\tilde\mu_l^* = \mu_l^* + \mu_0^*,$$

for $l = 1, 2, \ldots, r$, such that the completely random measures $(\mu_0^*, \mu_1^*, \mu_2^*, \ldots, \mu_r^*)$ are independent with Lévy intensities respectively equal to $(\nu_0^*, \nu_1^*, \ldots, \nu_r^*)$, where

$$
\begin{aligned}
\nu_0^*(\mathrm{d}s,\mathrm{d}y) &= (1-z)\mathrm{e}^{-s\sum_{i=1}^k\sum_{l=1}^r\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\mathbb{1}_{(t_{i-1}^*,t_i^*]}(y)}\nu(\mathrm{d}s,\mathrm{d}y)\\
\nu_l^*(\mathrm{d}s,\mathrm{d}y) &= z\mathrm{e}^{-s\sum_{i=1}^k\left(\bar n_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\mathbb{1}_{(t_{i-1}^*,t_i^*]}(y)}\nu(\mathrm{d}s,\mathrm{d}y).
\end{aligned}
$$

As for the second part of (16) we consider two distinct cases, namely

1. $n_{i,l} > 0$ only for one index $l = l_0$ for some $l_0 = 1, 2, \dots, r$.

2. There exist at least two indexes l_1, l_2 such that $l_1 \neq l_2$ and $n_{i,l_1} n_{i,l_2} > 0$.

Thus,

1. If $n_{i,l_0} > 0$ while $n_{i,l} = 0$ for every $l \neq l_0$, then

$$z \int_0^\infty \mathrm{e}^{-s\left(\lambda_{l_0} + \bar{n}_{i,l_0} + N'_{l_0} \mathbb{1}_{\{i \leq k_{l_0}\}}\right)} (\mathrm{e}^s - 1)^{n_{i,l_0}} \rho(s) \mathrm{d}s \mathbb{1}_{\{n_{i,l_0} > 0 \wedge n_{i,l} = 0, l \neq l_0\}} + (1 - z) \int_0^\infty \mathrm{e}^{-s \sum_{l=1}^r \left(\lambda_l + \bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \leq k_l\}}\right)} (\mathrm{e}^s - 1)^{\sum_{l=1}^r n_{i,l}} \rho(s) \mathrm{d}s$$

is the expected value of

$$\int_0^\infty \mathrm{e}^{-s\left(\lambda_{l_0} + \bar{n}_{i,l_0} + N'_{l_0} \mathbb{1}_{\{i \leq k_{l_0}\}}\right) - sV_{i,l_0} \sum_{l=1,\, l \neq l_0}^r \left(\lambda_l + \bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \leq k_l\}}\right)} (\mathrm{e}^s - 1)^{n_{i,l_0}} \rho(s) \mathrm{d}s$$

and therefore we recognise in the last part of (16) the Laplace transform of the random vector

$$\sum_{i \in \mathcal{I}_{l_0}} \left(J_1^{(i,l_0)} \delta_{T_i^*}, \dots, J_{l_0-1}^{(i,l_0)} \delta_{T_i^*}, J_{l_0}^{(i,l_0)} \delta_{T_i^*}, J_{l_0+1}^{(i,l_0)} \delta_{T_i^*}, \dots, J_r^{(i,l_0)} \delta_{T_i^*}\right),$$

where, for each $i = 1, \dots, k_{l_0}$, the random jumps $\left(J_1^{(i,l_0)}, \dots, J_{l_0-1}^{(i,l_0)}, J_{l_0}^{(i,l_0)}, J_{l_0+1}^{(i,l_0)}, \dots, J_r^{(i,l_0)}\right)$ are such that

$$\left(J_1^{(i,l_0)}, \dots, J_{l_0-1}^{(i,l_0)}, J_{l_0}^{(i,l_0)}, J_{l_0+1}^{(i,l_0)}, \dots, J_r^{(i,l_0)}\right) \stackrel{d}{=} (V_{i,l_0} J_{i,l_0}, \dots, V_{i,l_0} J_{i,l_0}, J_{i,l_0}, V_{i,l_0} J_{i,l_0}, \dots, V_{i,l_0} J_{i,l_0})$$

and every J_{i,l_0} has density proportional to

$$\mathrm{e}^{-s\left(\bar{n}_{i,l_0} + N'_{l_0} \mathbb{1}_{\{i \leq k_{l_0}\}}\right) - sV_{i,l_0} \sum_{l=1,\, l \neq l_0}^r \left(\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \leq k_l\}}\right)} (\mathrm{e}^s - 1)^{n_{i,l_0}} \rho(s).$$

2. There exist two indexes l_1, l_2 such that $l_1 \neq l_2$ and $n_{i,l_1} n_{i,l_2} > 0$. In this case

$$z \int_0^\infty \mathrm{e}^{-s\left(\lambda_l + \bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \leq k_l\}}\right)} (\mathrm{e}^s - 1)^{n_{i,l}} \rho(s) \mathrm{d}s \mathbb{1}_{\{n_{i,l} > 0 \wedge n_{i,j} = 0, j \neq l\}}$$

$$+ (1-z)\int_0^\infty \mathrm{e}^{-s\sum_{j=1}^r\left(\lambda_j+\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\leq k_j\}}\right)}\,(\mathrm{e}^s-1)^{\sum_{j=1}^r n_{i,j}}\,\rho(s)\mathrm{d}s$$

$$=(1-z)\int_0^\infty \mathrm{e}^{-s\sum_{j=1}^r\left(\lambda_j+\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\leq k_j\}}\right)}\,(\mathrm{e}^s-1)^{\sum_{j=1}^r n_{i,j}}\,\rho(s)\mathrm{d}s$$

coincides, up to a proportionality constant, to the Laplace transform of the random vector

$$\sum_{i\in\mathcal{I}_0}\left(J_1^{(i,0)}\delta_{T_i^*},J_2^{(i,0)}\delta_{T_i^*},\ldots,J_r^{(i)}\delta_{T_i^*}\right),$$

where, for each $i=1,\ldots,k_0$, the random jumps $\left(J_1^{(i,0)},J_2^{(i,0)},\ldots,J_r^{(i,0)}\right)$ are such that

$$\left(J_1^{(i,0)},J_2^{(i,0)},\ldots,J_r^{(i,0)}\right)\overset{d}{=}(J_{i,0},J_{i,0},\ldots,J_{i,0})$$

and every $J_{i,0}$ has density proportional to

$$\mathrm{e}^{-s\sum_{l=1}^r\left(\bar{n}_{i,l}+N'_l\mathbb{1}_{\{i\leq k_l\}}\right)}\,(\mathrm{e}^s-1)^{\sum_{l=1}^r n_{i,l}}\,\rho(s).$$

□

Now we use this result to obtain an expression for the posterior mean of $\tilde{F}_l(t)$.

Corollary 1. *Assuming a square loss function, the estimator of $\tilde{F}_l(t)$, for any $l=1,2,\ldots,r$ and $t>0$, conditionally on $\mathbf{T}^{(\mathbf{M})}$, $\mathbf{N}$ and $\mathbf{V}$ coincides with*

$$1-\exp\left\{-z\int_{\mathbb{R}^+\times(0,t]}\left(1-\mathrm{e}^{-s}\right)\mathrm{e}^{-s\sum_{i=1}^k\left(\bar{n}_{i,l}+N'_l\mathbb{1}_{\{i\leq k_l\}}\right)\mathbb{1}_{(t^*_{i-1},t^*_i]}(y)}\nu(\mathrm{d}s,\mathrm{d}y)\right\}$$
$$\times\exp\left\{-(1-z)\int_{\mathbb{R}^+\times(0,t]}\left(1-\mathrm{e}^{-s}\right)\mathrm{e}^{-s\sum_{i=1}^k\sum_{j=1}^r\left(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\leq k_j\}}\right)\mathbb{1}_{(t^*_{i-1},t^*_i]}(y)}\nu(\mathrm{d}s,\mathrm{d}y)\right\}$$
$$\times\prod_{i\in\mathcal{I}_l}\int_0^\infty \mathrm{e}^{-s\left(\mathbb{1}_{[T_i^*,\infty)}(t)+\bar{n}_{i,l}+N'_l\mathbb{1}_{\{i\leq k_l\}}+V_{i,l}\sum_{j=1,\,j\neq l}^r\left(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\leq k_j\}}\right)\right)}\,(\mathrm{e}^s-1)^{n_{i,l}}\,\rho(s)\mathrm{d}s$$
$$\times\prod_{j\neq l}\prod_{i\in\mathcal{I}_j}\int_0^\infty \mathrm{e}^{-s\left(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\leq k_j\}}+V_{i,j}\left[\mathbb{1}_{[T_i^*,\infty)}(t)+\sum_{J=1,\,J\neq j}^r\left(\bar{n}_{i,J}+N'_J\mathbb{1}_{\{i\leq k_J\}}\right)\right]\right)}\,(\mathrm{e}^s-1)^{n_{i,j}}\,\rho(s)\mathrm{d}s$$
$$\times\prod_{i\in\mathcal{I}_0}\int_0^\infty \mathrm{e}^{-s\left(\mathbb{1}_{[T_i^*,\infty)}(t)+\sum_{j=1}^r\left(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\leq k_j\}}\right)\right)}\,(\mathrm{e}^s-1)^{\bar{n}_i}\,\rho(s)\mathrm{d}s. \quad (17)$$

Proof. We exploit Theorem 3, equation (4), and the mutual independence of μ_l^* and $J_{i,l}$, with $l = 0, 1, \ldots, r$ and $i = 1, 2, \ldots, k_l$ to rewrite $\mathbb{E}[\tilde{F}_l(t) \mid \mathbf{T}^{(\mathbf{M})}, \mathbf{N}, \mathbf{V}]$ as follows.

$$\begin{aligned}\mathbb{E}[\tilde{F}_l(t) \mid \mathbf{T}^{(\mathbf{M})}, \mathbf{N}, \mathbf{V}] = \mathbb{E}[\tilde{F}_l'(t)] = 1 - \mathbb{E}[e^{-\tilde{\mu}_l'(0,t]}] \\ = 1 - \mathbb{E}\left[\mathrm{e}^{-\mu_l^*(0,t]}\right] \mathbb{E}\left[\mathrm{e}^{-\mu_0^*(0,t]}\right] \prod_{i\in\mathcal{I}_l} \mathbb{E}\left[\mathrm{e}^{-J_{i,l}\mathbb{1}_{[T_i^*,\infty)}(t)}\right] \\ \times \prod_{j\neq l}\prod_{i\in\mathcal{I}_j} \mathbb{E}\left[\mathrm{e}^{-V_{i,j}J_{i,j}\mathbb{1}_{[T_i^*,\infty)}(t)}\right] \prod_{i\in\mathcal{I}_0} \mathbb{E}\left[\mathrm{e}^{-J_{i,0}\mathbb{1}_{[T_i^*,\infty)}(t)}\right]. \quad (18)\end{aligned}$$

The proof then boils down to the evaluation of the expected values appearing in (18).

We evaluate the first two expected values using the Lévy-Khintchine representation we used in equation (4):

$$\mathbb{E}\left[\mathrm{e}^{-\mu_l^*(0,t]}\right] = \exp\left\{-z\int_{\mathbb{R}^+\times(0,t]} \left(1-\mathrm{e}^{-s}\right)\mathrm{e}^{-s\sum_{i=1}^k\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\mathbb{1}_{(t_{i-1}^*,t_i^*]}(y)}\nu(\mathrm{d}s,\mathrm{d}y)\right\}. \quad (19)$$

$$\mathbb{E}\left[\mathrm{e}^{-\mu_0^*(0,t]}\right] = \exp\left\{-(1-z)\int_{\mathbb{R}^+\times(0,t]} \left(1-\mathrm{e}^{-s}\right)\mathrm{e}^{-s\sum_{i=1}^k\sum_{j=1}^r\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\mathbb{1}_{(t_{i-1}^*,t_i^*]}(y)}\nu(\mathrm{d}s,\mathrm{d}y)\right\} \quad (20)$$

The other expected values can be evaluated using the densities of the jumps $J_{i,l}$ derived in Theorem 3.

$$\mathbb{E}\left[\mathrm{e}^{-J_{i,l}\mathbb{1}_{[T_i^*,\infty)}(t)}\right] = \int_0^\infty \mathrm{e}^{-s\left(\mathbb{1}_{[T_i^*,\infty)}(t)+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}+V_{i,l}\sum_{j=1,\,j\neq l}^r\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\right)}\left(\mathrm{e}^s-1\right)^{n_{i,l}}\rho(s)\mathrm{d}s \quad (21)$$

$$\mathbb{E}\left[\mathrm{e}^{-V_{i,j}J_{i,j}\mathbb{1}_{[T_i^*,\infty)}(t)}\right] = \int_0^\infty \mathrm{e}^{-s\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}+V_{i,j}\left[\mathbb{1}_{[T_i^*,\infty)}(t)+\sum_{J=1,\,J\neq j}^r\left(\bar{n}_{i,J}+N_J'\mathbb{1}_{\{i\le k_J\}}\right)\right]\right)}\left(\mathrm{e}^s-1\right)^{n_{i,j}}\rho(s)\mathrm{d}s \quad (22)$$

$$\mathbb{E}\left[\mathrm{e}^{-J_{i,0}\mathbb{1}_{[T_i^*,\infty)}(t)}\right] = \int_0^\infty \mathrm{e}^{-s\left(\mathbb{1}_{[T_i^*,\infty)}(t)+\sum_{j=1}^r\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\right)}\left(\mathrm{e}^s-1\right)^{\bar{n}_i}\rho(s)\mathrm{d}s. \quad (23)$$

Combining (19), (20), (21), (22) and (23) with (18) completes the proof. □

4.2 Specifying a Lévy Intensity

In the application we will further assume that the underlying CRMs have σ-stable distribution, with $0 < \sigma < 1$, that is

$$\nu(\mathrm{d}s, \mathrm{d}y) = \frac{\sigma}{\Gamma(1-\sigma)} s^{-1-\sigma} \mathrm{d}s\, \alpha(\mathrm{d}y). \tag{24}$$

Plugging the specified form for ν into Theorem 3 shows how the characterisation of the posterior distribution $\tilde{F}_l$ simplifies under the additional assumptions.

Under the assumption (24), the density $f_{J_{i,l}}(s)$ of $J_{i,l}$ coincides with

$$\frac{\sigma}{\Gamma(1-\sigma)} \frac{\mathrm{e}^{-s\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}+V_{i,l}\sum_{j=1,\,j\neq l}^{r}\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\right)}\,(\mathrm{e}^s-1)^{n_{i,l}}\,s^{-1-\sigma}}{\sum_{m=0}^{n_{i,l}}(-1)^{n_{i,l}-m+1}\binom{n_{i,l}}{m}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}+V_{i,l}\sum_{j=1,\,j\neq l}^{r}\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)-m\right)^{\sigma}}$$

and the density $f_{J_{i,0}}(s)$ of $J_{i,0}$ coincides with

$$\frac{\sigma}{\Gamma(1-\sigma)} \frac{\mathrm{e}^{-s\sum_{l=1}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)}\,(\mathrm{e}^s-1)^{\bar{n}_i}\,s^{-1-\sigma}}{\sum_{m=0}^{\bar{n}_i}(-1)^{\bar{n}_i-m+1}\binom{\bar{n}_i}{m}\left(\sum_{l=1}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)-m\right)^{\sigma}}.$$

These expressions are obtained by letting $\rho(s)\mathrm{d}s = \frac{\sigma}{\Gamma(1-\sigma)}s^{-1-\sigma}\mathrm{d}s$ as in (24) and calculating the normalising constants of $f_{J_{i,l}}(s)$ and $f_{J_{i,0}}(s)$.

Next, under the assumption (24) we derive a closed-form expression for the expected value of $\tilde{F}_l(t)$ for any $l = 1, 2, \ldots, r$ and $t > 0$, conditionally on $\mathbf{T^{(M)}}$, $\mathbf{N}$ and $\mathbf{V}$.

4.3 Specifying a Base Measure

We can further specify the intensity ν by imposing that $\alpha = cP_0$, with P_0 a one-parameter Gamma distribution with parameter γ, that is

$$\nu(\mathrm{d}s, \mathrm{d}y) = \frac{\sigma}{\Gamma(1-\sigma)} s^{-1-\sigma} \mathrm{d}s\, c\,\frac{\gamma^\gamma}{\Gamma(\gamma)} y^{\gamma-1} e^{-\gamma y}\mathrm{d}y. \tag{25}$$

Proposition 1. *Assuming a Lévy intensity as in* (25)*, the estimator of $\tilde{F}_l(t)$, for any $l = 1, 2, \ldots, r$ and $t > 0$, conditionally on $\mathbf{T^{(M)}}$, $\mathbf{N}$ and $\mathbf{V}$ coincides with*

$$1 - \exp\left\{-c\sum_{i=1}^{i_0-1}\left(F_\gamma(T_i^*) - F_\gamma(T_{i-1}^*)\right)\left(z[(1+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}})^{\sigma} - (\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}})^{\sigma}]\right.\right.$$

$$
+(1-z)\left[\left(1+\sum_{j=1}^{r}(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\le k_j\}})\right)^{\sigma}-\left(\sum_{j=1}^{r}(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\le k_j\}})\right)^{\sigma}\right]\Bigg)
$$

$$
-c\mathbb{1}_{\{1,2,\dots,k\}}(i_0)(F_\gamma(t)-F_\gamma(T^*_{i_0-1}))\left(z[(1+\bar{n}_{i_0,l}+N'_l\mathbb{1}_{\{i_0\le k_l\}})^{\sigma}-(\bar{n}_{i_0,l}+N'_l\mathbb{1}_{\{i_0\le k_l\}})^{\sigma}]\right.
$$

$$
\left.+(1-z)\left[\left(1+\sum_{j=1}^{r}(\bar{n}_{i_0,j}+N'_j\mathbb{1}_{\{i_0\le k_j\}})\right)^{\sigma}-\left(\sum_{j=1}^{r}(\bar{n}_{i_0,l}+N'_j\mathbb{1}_{\{i_0\le k_j\}})\right)^{\sigma}\right]\right)
$$

$$
-c\mathbb{1}_{\{k+1\}}(i_0)(F_\gamma(t)-F_\gamma(T^*_k))\Big\}
$$

$$
\times\prod_{i\in\mathcal{I}_l}\left((1-V_{i,l})\frac{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(\mathbb{1}_{[T^*_i,\infty)}(t)+\bar{n}_{i,l}+N'_l\mathbb{1}_{\{i\le k_l\}}-m\right)^{\sigma}}{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(\bar{n}_{i,l}+N'_l\mathbb{1}_{\{i\le k_l\}}-m\right)^{\sigma}}\right.
$$

$$
\left.+V_{i,l}\frac{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(\mathbb{1}_{[T^*_i,\infty)}(t)+\sum_{j=1}^{r}\left(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\le k_j\}}\right)-m\right)^{\sigma}}{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(\sum_{j=1}^{r}\left(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\le k_j\}}\right)-m\right)^{\sigma}}\right)
$$

$$
\times\prod_{j\ne l}\prod_{i\in\mathcal{I}_j}\left((1-V_{i,j})+V_{i,j}\frac{\sum_{m=0}^{n_{i,j}}(-1)^m\binom{n_{i,j}}{m}\left(\mathbb{1}_{[T^*_i,\infty)}(t)+\sum_{J=1}^{r}\left(\bar{n}_{i,J}+N'_J\mathbb{1}_{\{i\le k_j\}}\right)-m\right)^{\sigma}}{\sum_{m=0}^{n_{i,j}}(-1)^m\binom{n_{i,j}}{m}\left(\sum_{J=1}^{r}\left(\bar{n}_{i,J}+N'_J\mathbb{1}_{\{i\le k_j\}}\right)-m\right)^{\sigma}}\right)
$$

$$
\times\prod_{i\in\mathcal{I}_0}\left(\frac{\sum_{m=0}^{\bar{n}_i}(-1)^m\binom{\bar{n}_i}{m}\left(\mathbb{1}_{[T^*_i,\infty)}(t)+\sum_{j=1}^{r}\left(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\le k_j\}}\right)-m\right)^{\sigma}}{\sum_{m=0}^{\bar{n}_i}(-1)^m\binom{\bar{n}_i}{m}\left(\sum_{j=1}^{r}\left(\bar{n}_{i,j}+N'_j\mathbb{1}_{\{i\le k_j\}}\right)-m\right)^{\sigma}}\right). \quad (26)
$$

where F_γ is the distribution function of a one-parameter Gamma distribution with shape and rate equal to γ.

Proof. We begin with our specified Lévy intensity from (25) and evaluate the expected values as in Corollary 1.

$$
\nu(\mathrm{d}s,\mathrm{d}y)=\rho(s)\mathrm{d}s\alpha(\mathrm{d}y)=\frac{\sigma}{\Gamma(1-\sigma)}s^{-1-\sigma}\mathrm{d}s\,c\,\frac{\gamma^\gamma}{\Gamma(\gamma)}y^{\gamma-1}e^{-\gamma y}\mathrm{d}y.
$$

We obtain

$$
\mathbb{E}\left[\mathrm{e}^{-\mu^*_l(0,t]}\right]
$$

$$
=\exp\left\{-cz\sum_{i=1}^{i_0-1}\left[\left(1+\bar{n}_{i,l}+N'_l\mathbb{1}_{\{i\le k_l\}}\right)^{\sigma}-\left(\bar{n}_{i,l}+N'_l\mathbb{1}_{\{i\le k_l\}}\right)^{\sigma}\right]\right.
$$

$$
\times\left(F_\gamma(T^*_i)-F_\gamma(T^*_{i-1})\right)
$$

$$
-c\mathbb{1}_{\{1,2,\dots,k\}}(i_0)\left(z[(1+\bar{n}_{i_0,l}+N'_l\mathbb{1}_{\{i\le k_l\}})^{\sigma}-(\bar{n}_{i_0,l}+N'_l\mathbb{1}_{\{i\le k_l\}})^{\sigma}]\right)(F_\gamma(t)-F_\gamma(T^*_{i_0-1}))
$$

$$-cz\mathbb{1}_{\{k+1\}}(i_0)(F_\gamma(t)-F_\gamma(T_k^*))\Big\}\,. \tag{27}$$

Similarly,

$$\begin{aligned}
&\mathbb{E}\left[\mathrm{e}^{-\mu_0^*(0,t]}\right]\\
&= \exp\left\{-c(1-z)\sum_{i=1}^{i_0-1}\left[\left(1+\sum_{j=1}^{r}\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)^{\sigma}-\left(\sum_{j=1}^{r}\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)^{\sigma}\right]\right.\\
&\qquad\times(F_\gamma(T_i^*)-F_\gamma(T_{i-1}^*))\\
&-c\mathbb{1}_{\{1,2,\dots,k\}}(i_0)(1-z)\left[\left(1+\sum_{j=1}^{r}(\bar{n}_{i_0,j}+N_j'\mathbb{1}_{\{i_0\le k_j\}})\right)^{\sigma}-\left(\sum_{j=1}^{r}(\bar{n}_{i_0,l}+N_j'\mathbb{1}_{\{i_0\le k_j\}})\right)^{\sigma}\right]\\
&\qquad\times(F_\gamma(t)-F_\gamma(T_{i_0-1}^*))\\
&\qquad\left.-c\mathbb{1}_{\{k+1\}}(i_0)(1-z)(F_\gamma(t)-F_\gamma(T_k^*))\right\}
\end{aligned} \tag{28}$$

$$\begin{aligned}
&\mathbb{E}\left[\mathrm{e}^{-J_{i,l}\mathbb{1}_{[T_i^*,\infty)}(t)}\right]\\
&=\frac{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(\mathbb{1}_{[T_i^*,\infty)}(t)+m+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}+V_{i,l}\sum_{j=1,j\ne l}^{r}\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\right)^{\sigma}}{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(m+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}+V_{i,l}\sum_{j=1,j\ne l}^{r}\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\right)^{\sigma}}\\
&=(1-V_{i,l})\frac{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(\mathbb{1}_{[T_i^*,\infty)}(t)+m+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)^{\sigma}}{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(m+\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)^{\sigma}}\\
&\quad+V_{i,l}\frac{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(\mathbb{1}_{[T_i^*,\infty)}(t)+m+\sum_{j=1}^{r}\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\right)^{\sigma}}{\sum_{m=0}^{n_{i,l}}(-1)^m\binom{n_{i,l}}{m}\left(m+\sum_{j=1}^{r}\left(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}\right)\right)^{\sigma}}
\end{aligned} \tag{29}$$

$$\begin{aligned}
&\mathbb{E}\left[\mathrm{e}^{-V_{i,j}J_{i,j}\mathbb{1}_{[T_i^*,\infty)}(t)}\right]\\
&=\frac{\sum_{m=0}^{n_{i,j}}(-1)^m\binom{n_{i,j}}{m}\left(m+\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}+V_{i,j}\left(\mathbb{1}_{[T_i^*,\infty)}(t)+\sum_{l=1,\,l\ne j}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\right)\right)^{\sigma}}{\sum_{m=0}^{n_{i,j}}(-1)^m\binom{n_{i,j}}{m}\left(m+\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}}+V_{i,j}\sum_{l=1,\,l\ne j}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\right)^{\sigma}}\\
&=(1-V_{i,j})+V_{i,j}\frac{\sum_{m=0}^{n_{i,j}}(-1)^m\binom{n_{i,j}}{m}\left(\mathbb{1}_{[T_i^*,\infty)}(t)+m+\sum_{l=1}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\right)^{\sigma}}{\sum_{m=0}^{n_{i,j}}(-1)^m\binom{n_{i,j}}{m}\left(m+\sum_{l=1}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\right)^{\sigma}}
\end{aligned} \tag{30}$$

$$\mathbb{E}\left[\mathrm{e}^{-J_{i,0}\mathbb{1}_{[T_i^*,\infty)}(t)}\right]=\frac{\sum_{m=0}^{\bar{n}_i}(-1)^m\binom{\bar{n}_i}{m}\left(\mathbb{1}_{[T_i^*,\infty)}(t)+m+\sum_{l=1}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\right)^{\sigma}}{\sum_{m=0}^{\bar{n}_i}(-1)^m\binom{\bar{n}_i}{m}\left(m+\sum_{l=1}^{r}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}\right)\right)^{\sigma}}. \tag{31}$$

□

The estimator provided in Proposition 1 can be evaluated exactly, conditionally on the realisation of (unobservable) latent variables $\mathbf{N}$ and $\mathbf{V}$. In order to marginalise with respect to these variables we will devise a Gibbs sampler in section 5.

5 Posterior Sampling

We proceed using the Lévy intensity specified in (25) and the corresponding expression for $\tilde{F}_l(t \mid \mathbf{T^{(N)}}, \mathbf{N}, \mathbf{V})$ derived in the previous section. Our goal is to use this expression to estimate the posterior mean $\tilde{F}_l(t \mid \mathbf{T^{(M)}})$. We achieve this by marginalising with respect to sequences of unobservable variables by means of a Gibbs sampler.

5.1 Gibbs sampling

Chapter 9 of Robert and Casella (2004) offers a thorough explanation of Gibbs sampling. The purpose of our Gibbs sampler is to marginalise (26) with respect to $\mathbf{N}$ and $\mathbf{V}$. We do this by first deriving the full conditional distributions of these random variables and of the relevant parameters from the joint distribution of $(\mathbf{N}, \mathbf{T^{(M)}}, \mathbf{V})$, which will be derived in the next section.

Once we have the full conditionals and can sample from them, we can begin Gibbs sampling. We consider these variables and parameters, apart from $\mathbf{T^{(M)}}$, as entries in one vector $\mathbf{Y}$. We have $\mathbf{Y}^{(0)}$, the entries' initial values, and we sample further values as follows.

$$\begin{aligned} Y_1^{(1)} &\sim \; p(Y_1 \mid \mathbf{T}^{(M)}, Y_2^{(0)}, Y_3^{(0)}, \dots) \\ Y_2^{(1)} &\sim \; p(Y_2 \mid \mathbf{T}^{(M)}, Y_1^{(1)}, Y_3^{(0)}, Y_4^{(0)}, \dots) \\ &\vdots \end{aligned}$$

and repeat the process for $\mathbf{Y}^{(2)}$ and so on for a large number of iterations. After a suitably chosen burn-in period, we can record these realisations and use each $Y^{(j)}$ to calculate $F_l(t \mid \mathbf{N}, \mathbf{T^{(M)}}, \mathbf{V})^{(j)}$. Then we average the values produced over a grid of t values to obtain pointwise estimations and credible intervals of $\tilde{F}_l(t \mid \mathbf{T^{(M)}})$.

5.2 Full conditionals

In order to derive full conditional distributions for $\mathbf{N}$, $\mathbf{V}$, and the hyperparameters involved, we start from the joint distribution of $(\mathbf{N}, \mathbf{T^{(M)}}, \mathbf{V})$. This can be obtained starting from (15). Specifically, we write

$$
\begin{aligned}
\mathrm{p}(\mathbf{T^{(M)}}, \mathbf{N}, \mathbf{V}) &= \mathrm{p}(\mathbf{T^{(M)}} \mid \mathbf{V}, \mathbf{N})\mathrm{p}(\mathbf{V})\mathrm{p}(\mathbf{N}) \\
&= \prod_{i=1}^{k} \exp\left\{ -z\alpha(I_i) \sum_{l=1}^{r} \int_0^\infty \left(1 - \mathrm{e}^{-s\left(\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}}\right)}\right) \rho(s)\mathrm{d}s \right. \\
&\quad \left. - (1-z)\alpha(I_i) \int_0^\infty \left(1 - \mathrm{e}^{-s \sum_{l=1}^{r}\left(\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}}\right)}\right) \rho(s)\mathrm{d}s \right\} \\
&\times \prod_{l=1}^{r} \prod_{i \in \mathcal{I}_l} \alpha(L_i) \int_0^\infty e^{-s(\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}}) - sV_{i,l} \sum_{j=1, j\neq l}^{r} (\bar{n}_{i,j} + N'_j \mathbb{1}_{\{i \le k_j\}})} (e^s - 1)^{n_{i,l}} \rho(s)\mathrm{d}s \\
&\times \prod_{i \in \mathcal{I}_0} \alpha(L_i)(1-z) \int_0^\infty e^{-s \sum_{l=1}^{r} (\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}})} (e^s - 1)^{\bar{n}_i} \rho(s)\mathrm{d}s \\
&\times z^{\sum_{i=1}^{k} \sum_{l=1}^{r} (1 - V_{i,l})} (1-z)^{\sum_{i=1}^{k} \sum_{l=1}^{r} V_{i,l}} \prod_{l=1}^{r} \frac{(N'_l + M_l)!}{N'_l!} \mathrm{p}(\mathbf{N}).
\end{aligned}
$$

Now we attribute the σ-stable distribution to ρ and specify α as a one-parameter Gamma base measure with parameter γ. We thus obtain

$$
\begin{aligned}
&\prod_{i=1}^{k} \alpha(\mathrm{d}t_i^*)(1-z)^{\#\mathcal{I}_0} \prod_{i=1}^{k} \exp\left\{ -z(F_\gamma(T_i^*) - F_\gamma(T_{i-1}^*)) \sum_{l=1}^{r} (\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}})^\sigma \right. \\
&\quad \left. - (1-z)(F_\gamma(T_i^*) - F_\gamma(T_{i-1}^*)) \left(\sum_{l=1}^{r} (\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}}) \right)^\sigma \right\} \\
&\times \prod_{l=1}^{r} \prod_{i \in \mathcal{I}_l} \left(\sum_{J=0}^{n_{i,l}} (-1)^{n_{i,l} - J + 1} \binom{n_{i,l}}{J} \left(\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}} + V_{i,l} \sum_{j=1, j\neq l}^{r} (\bar{n}_{i,j} + N'_j \mathbb{1}_{\{i \le k_j\}}) - J \right)^\sigma \right) \\
&\quad \times \prod_{i \in \mathcal{I}_0} \left(\sum_{J=0}^{\bar{n}_i} (-1)^{\bar{n}_i - J + 1} \binom{\bar{n}_i}{J} \left(\sum_{l=1}^{r} (\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}}) - J \right)^\sigma \right) \\
&\quad \times z^{\sum_{i=1}^{k} \sum_{l=1}^{r} (1 - V_{i,l})} (1-z)^{\sum_{i=1}^{k} \sum_{l=1}^{r} V_{i,l}} p \prod_{l=1}^{r} \frac{(N'_l + M_l)!}{N'_l!} \mathrm{p}(\mathbf{N}), \qquad (32)
\end{aligned}
$$

where $\#\mathcal{I}_0$ is the number of indices in $\mathcal{I}_0$.

From the joint distribution above, we derive the following expression, which defines the full conditional distribution of the l^{th} element of $\mathbf{N}$ up to a multiplicative constant. We recall our prior assumption that $N_l \sim \mathrm{Pois}(\lambda)$ and obtain

$$
\mathrm{p}(N_l | \mathbf{N}_{-l}, \mathbf{T^{(N)}}, \mathbf{V}) \propto \frac{(N'_l + M_l)!}{N'_l!} \prod_{i=1}^{k} \exp\left\{ -(F_\gamma(T_i^*) - F_\gamma(T_{i-1}^*)) \right.
$$

$$\times \left(z(\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}})^\sigma + (1-z) \left(\sum_{j=1}^{r} (\bar{n}_{i,j} + N'_j \mathbb{1}_{\{i \le k_j\}}) \right)^\sigma \right) \Bigg\}$$

$$\times \prod_{j=1}^{r} \prod_{i \in \mathcal{I}_j} \left(\sum_{J=0}^{n_{i,j}} (-1)^{n_{i,j}-J+1} \binom{n_{i,j}}{J} \left(\bar{n}_{i,j} + N'_j \mathbb{1}_{\{i \le k_j\}} + \right. \right.$$

$$\left. \left. V_{i,j} \sum_{m=1, m \ne l}^{r} (\bar{n}_{i,m} + N'_m \mathbb{1}_{\{i \le k_m\}}) - J \right)^\sigma \right)$$

$$\times \prod_{i \in \mathcal{I}_0} \left(\sum_{J=0}^{\bar{n}_i} (-1)^{\bar{n}_i - J+1} \binom{\bar{n}_i}{J} \left(\sum_{m=1}^{r} (\bar{n}_{i,m} + N'_m \mathbb{1}_{\{i \le k_m\}}) - J \right)^\sigma \right) \frac{\lambda^{N_l}}{N_l!} \quad (33)$$

where $\mathbf{N}_{-p}$ is the vector $\mathbf{N}$ with the pth value deleted.

The full conditional distribution of the (p,q)th element of $\mathbf{V}$ is given by

$$\mathrm{p}(V_{i,l} | \mathbf{N}, \mathbf{T}^{(\mathbf{N})}, \mathbf{V}_{-(i,l)}) \propto \sum_{J=0}^{n_{i,l}} (-1)^{n_{i,l}-J+1} \binom{n_{i,l}}{J}$$

$$\times \left(\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}} + V_{i,l} \sum_{j=1, j \ne l}^{r} (\bar{n}_{i,j} + N'_j \mathbb{1}_{\{i \le k_j\}}) - J \right)^\sigma z^{1-V_{i,l}} (1-z)^{V_{i,l}}$$

for $i \in \mathcal{I}_l$, some $l = 1, 2, \ldots, r$.

We denote by $\pi = (c, \sigma, \gamma, ...)$ the vector of hyperparameters of the model. The full conditional for the total mass c is given by

$$\begin{aligned} \mathrm{p}(c \mid \ldots) &\propto \mathrm{p}(c) \mathrm{p}(\mathbf{T}^{(\mathbf{N})}, \mathbf{N}, \mathbf{V} \mid \pi) \\ &\propto \mathrm{p}(c) c^{k-1} \exp \left\{ -c \sum_{i=1}^{k} (F_\gamma(T_i^*) - F_\gamma(T_{i-1}^*)) \right. \\ &\left. \times \left[z \sum_{l=1}^{r} (\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}})^\sigma + (1-z) \left(\sum_{l=1}^{r} (\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}}) \right)^\sigma \right] \right\}. \end{aligned}$$

This implies that, if $\mathrm{p}(c)$ is gamma, that is a priori $c \sim \mathrm{Gamma}(a, b)$, where a and b are, respectively, shape and rate parameter, then

$$c \mid \ldots \sim \mathrm{Gamma} \left(a + k - 1, b + \sum_{i=1}^{k} (F_\gamma(T_i^*) - F_\gamma(T_{i-1}^*)) \right.$$
$$\left. \times \left[z \sum_{l=1}^{r} (\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}})^\sigma + (1-z) \left(\sum_{l=1}^{r} (\bar{n}_{i,l} + N'_l \mathbb{1}_{\{i \le k_l\}}) \right)^\sigma \right] \right).$$

The full conditional distribution of σ with a uniform prior is given by

$$\begin{aligned}
\mathrm{p}(\sigma|\mathbf{N},\mathbf{T}^{(\mathbf{N})},\mathbf{V}) \propto \prod_{i=1}^{k} \exp\Bigg\{ & -z(F_\gamma(T_i^*)-F_\gamma(T_{i-1}^*))\sum_{l=1}^{r}(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}})^\sigma \\
& -(1-z)(F_\gamma(T_i^*)-F_\gamma(T_{i-1}^*))\left(\sum_{l=1}^{r}(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}})\right)^\sigma \Bigg\} \\
\times \prod_{l=1}^{r}\prod_{i\in\mathcal{I}_l}\left(\sum_{J=0}^{n_{i,l}}(-1)^{n_{i,l}-J+1}\binom{n_{i,l}}{J}\left(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}}+V_{i,l}\sum_{j=1,j\neq l}^{r}(\bar{n}_{i,j}+N_j'\mathbb{1}_{\{i\le k_j\}})-J\right)^\sigma\right) & \\
\times \prod_{i\in\mathcal{I}_0}\left(\sum_{J=0}^{\bar{n}_i}(-1)^{\bar{n}_i-J+1}\binom{\bar{n}_i}{J}\left(\sum_{l=1}^{r}(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}})-J\right)^\sigma\right). &
\end{aligned}$$

We attriute a Gamma prior to γ and obtain the full conditional

$$\begin{aligned}
\mathrm{p}(\gamma|\mathbf{N},\mathbf{T}^{(\mathbf{N})},\mathbf{V}) \propto \prod_{i=1}^{k} \exp\Bigg\{ & -z(F_\gamma(T_i^*)-F_\gamma(T_{i-1}^*))\sum_{l=1}^{r}(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}})^\sigma \\
& -(1-z)(F_\gamma(T_i^*)-F_\gamma(T_{i-1}^*))\left(\sum_{l=1}^{r}(\bar{n}_{i,l}+N_l'\mathbb{1}_{\{i\le k_l\}})\right)^\sigma \Bigg\} p_\gamma(\gamma).
\end{aligned}$$

5.3 Sampling from the full conditionals

Some of the full conditional distributions we have derived require delicate treatment during sampling. In this section we go into the detail of how we approached various issues we encountered during the sampling process.

5.3.1 Metropolis-Hastings for N_p

We propose a Metropolis-Hastings strategy, as detailed in chapter 7 of Robert and Casella (2004), to sample from the full conditional for N_p, given in (33). It will be numerically convenient to consider the logarithm of such full conditional, that is

$$\begin{aligned}
\log \mathrm{p}(N_p \mid \mathbf{N}_p,\mathbf{T}^{(\mathbf{M})},\mathbf{V}) = \text{const } & + \log \mathrm{p}(N_p) + \log\Gamma(N_p+1) - \log\Gamma(N_p-M_p+1) \\
& - \sum_{i=1}^{k}(F_\gamma(T_i^*)-F_\gamma(T_{i-1}^*))\big(z(\bar{n}_{i,p}+(N_p-M_p)\mathbb{1}_{\{i\le k_p\}})^\sigma \\
& +(1-z)\left(\sum_{l=1}^{r}(\bar{n}_{i,l}+(N_l-M_l)\mathbb{1}_{\{i\le k_l\}})\right)\Bigg)
\end{aligned}$$

$$+\sum_{l=1}^{r}\sum_{i\in\mathcal{I}_l}\log\left(\sum_{J=0}^{n_{i,l}}(-1)^{n_{i,l}-J+1}\binom{n_{i,l}}{J}\left(\bar{n}_{i,l}+(N_l-M_l)\mathbb{1}_{\{i\leq k_l\}}\right.\right.$$
$$\left.\left.+V_{i,l}\sum_{j=1,j\neq l}^{r}(\bar{n}_{i,j}+(N_j-M_j)\mathbb{1}_{\{i\leq k_j\}})-J\right)^{\sigma}\right)$$
$$+\sum_{i\in\mathcal{I}_0}\log\left(\sum_{J=0}^{\bar{n}_i}(-1)^{\bar{n}_i-J+1}\binom{\bar{n}_i}{J}\left(\sum_{l=1}^{r}(\bar{n}_{i,l}+(N_l-M_l)\mathbb{1}_{\{i\leq k_l\}})-J\right)^{\sigma}\right). \quad (34)$$

Suppose that at the rth iteration N_p takes value N. Given N, we propose a new value N^* distributed as $N^* = N + X$ where $X = Y - \varphi$ and $Y \sim \text{Pois}(\varphi)$, for some positive integer φ. That is the random variable X has support on $\{-\varphi, -\varphi + 1, \ldots\}$, has mean 0 and variance φ and has probability mass function g given by

$$g(x) = \begin{cases} \mathrm{e}^{-\varphi}\frac{\varphi^{\varphi+x}}{\Gamma(\varphi+x+1)} & \text{if } x \in \{-\varphi, -\varphi+1, \ldots\}, \\ 0 & \text{otherwise.} \end{cases}$$

Equivalently, we say that N^* is generated from $q(N^* \mid N) = g(N^* - N)$. Once we generate N^* we accept it with probability

$$\alpha(N, N^*) = \min\left\{1, \frac{\mathrm{p}(N^* \mid \ldots)g(N - N^*)}{\mathrm{p}(N \mid \ldots)g(N^* - N)}\right\}. \quad (35)$$

Observe that, by definition of g, the proposed value N^* takes values in $\{N-\varphi, N-\varphi+1, \ldots\}$, which implies that (N^*-N) takes values in $\{-\varphi, -\varphi+1, \ldots\}$ and that $(N-N^*)$ takes values in $\{\ldots, \varphi-1, \varphi\}$. This means that, whatever the proposed value N^* is, $g(N^*-N) > 0$, that is the acceptance probability in (35) is well defined. Notice that, if $N^* < M_p$ or if $N^* > N+\varphi$, then $\alpha(N, N^*) = 0$.

The resulting algorithm is composed of the following steps:

1. sample $Y \sim \text{Pois}(\varphi)$,
2. if $Y < \varphi - N + M_p$ or $Y > 2\varphi$ set $N_p^{(r+1)} = N$ and go to step 7,
3. set $N^* = N + Y - \varphi$,

4. set $\alpha = \mathrm{p}(N^* \mid \ldots) g(N - N^*)/(\mathrm{p}(N \mid \ldots) g(N^* - N))$,

5. sample $U \sim \mathrm{Unif}(0, 1)$,

6. if $U < \alpha$ set $N_p^{(r+1)} = N^*$, otherwise set $N_p^{(r+1)} = N$,

7. return $N_p^{(r+1)}$.

In order to work only with the logarithm of the full conditional of N_p, given in (34), one can replace points 4 and 6 in the previous algorithm by

4′. set $\log \alpha = \log \mathrm{p}(N^* \mid \ldots) - \log \mathrm{p}(N \mid \ldots) + \log g(N - N^*) - \log g(N^* - N)$,

6′. if $\log U < \log \alpha$ set $N_p^{(r+1)} = N^*$, otherwise set $N_p^{(r+1)} = N$,

where

$$\log g(x) = -\varphi + (\varphi + x) \log(\varphi) - \log \Gamma(\varphi + x + 1).$$

5.3.2 Monte Carlo evaluation

In Proposition 1, we derive (26), an expression for $\tilde{F}_l(t \mid \mathbf{T}^{(\mathbf{M})}, \mathbf{N}, \mathbf{V})$. Evaluating this expression numerically is non-trivial because it contains ratios of the form

$$\frac{\sum_{m=0}^{n} (-1)^{m-n} \binom{n}{m} (1 + A - m)^\sigma}{\sum_{m=0}^{n} (-1)^{m-n} \binom{n}{m} (A - m)^\sigma}. \tag{36}$$

with $A \geq n + 1$. The problem is that both parts of the ratio go to 0 quickly and this poses numerical instability.

For convenience, let

$$I(n, A) = \sum_{m=0}^{n} (-1)^{m-n} \binom{n}{m} (A - m)^\sigma.$$

The goal is to compute the ratio

$$\frac{I(n, A + 1)}{I(n, A)}.$$

We claim that $I(n, A)$ can be expressed as the solution of an integral, specifically

$$I(n, A) = -\frac{\sigma}{\Gamma(1 - \sigma)} \int_0^\infty \mathrm{e}^{-sA} (\mathrm{e}^s - 1)^n s^{-1-\sigma} \mathrm{d}s. \tag{37}$$

This can be proved by induction on n.

1. First we consider the case where $n = 1$. We write the Gamma function $\Gamma(1-\sigma)$ in its integral form $\int_0^\infty s^{-\sigma}e^{-s}\mathrm{d}s$ and obtain

$$I(1, A) = \sigma \int_0^\infty \mathrm{e}^{-s(A-1)}(\mathrm{e}^s - 1)s^{-1}.$$

   This reduces to the difference of exponential integrals which yield the required result.

2. Now suppose the relation holds for $n = n_0$

3. We prove it for the $n = n_0 + 1$ case.

$$\begin{aligned}
I(n_0+1, A) &= -\frac{\sigma}{\Gamma(1-\sigma)} \int_0^\infty \mathrm{e}^{-sA}(\mathrm{e}^s - 1)^{n_0+1}s^{-1-\sigma}\mathrm{d}s \\
&= -\frac{\sigma}{\Gamma(1-\sigma)} \left[\int_0^\infty \mathrm{e}^{-s(A-1)}(\mathrm{e}^s - 1)^{n_0}s^{-1-\sigma}\mathrm{d}s - \int_0^\infty \mathrm{e}^{-sA}(\mathrm{e}^s - 1)^{n_0}s^{-1-\sigma}\mathrm{d}s \right] \\
&= I(n_0, A-1) - I(n_0, A) \\
&= \sum_{m=0}^{n_0} (-1)^{m-n_0} \binom{n_0}{m} (A - (m+1))^\sigma - \sum_{m=0}^{n_0} (-1)^{m-n_0} \binom{n_0}{m} (A-m)^\sigma.
\end{aligned}$$

   Here we shift the indices in the first sum by 1 and factor out the $(A-m)^\sigma$ term, which after some manipulation yields

$$\begin{aligned}
&\sum_{m=1}^{n_0} (-1)^{m-(n_0+1)}(A-m)^\sigma \binom{n_0+1}{m} + (-1)^{n_0+1-(n_0+1)}(A-(n_0+1))^\sigma + (-1)^{-(n_0+1)}A^\sigma \\
&\qquad = \sum_{m=0}^{n_0+1} (-1)^{m-(n_0+1)} \binom{n_0+1}{m} (A-m)^\sigma,
\end{aligned}$$

   as required.

The integral (37) can be rewritten as

$$\begin{aligned}
&-\frac{\sigma}{\Gamma(1-\sigma)} \int_0^\infty \mathrm{e}^{-s(A-n)}(1-\mathrm{e}^{-s})^n s^{-1-\sigma}\mathrm{d}s \\
&= -\frac{\sigma}{\Gamma(1-\sigma)} \int_1^0 y^{(A-n)}(1-y)^n(-\log(y))^{-1-\sigma}(-\frac{1}{y})\mathrm{d}y \\
&= -\frac{\sigma}{\Gamma(1-\sigma)} \int_0^1 y^{(A-n)-1}(1-y)^n(-\log(y))^{-1-\sigma}\mathrm{d}y,
\end{aligned}$$

where we performed the change of variable $y = \exp\{-s\}$, which imples $-\log(y) = s$ and $\mathrm{d}s = -\frac{1}{y}\mathrm{d}y$.

Our proposed solution is to devise an importance sampling approach whereby we evaluate the ratio (36) by sampling from a Beta proposal distribution. Let $Y \sim \text{Beta}(A-n, n+1)$ with density $f(y; A-n, n+1)$ and observe that

$$\begin{aligned} I(n, A) &= -\frac{\sigma}{\Gamma(1-\sigma)} \int_0^1 y^{(A-n)-1}(1-y)^n(-\log(y))^{-1-\sigma} \mathrm{d}y \\ &= -\frac{\sigma}{\Gamma(1-\sigma)} \mathrm{B}(A-n, n+1) \int_0^1 f(y; A-n, n+1)(-\log(y))^{-1-\sigma} \mathrm{d}y \\ &= -\frac{\sigma}{\Gamma(1-\sigma)} \mathrm{B}(A-n, n+1) \mathbb{E}[(-\log(Y))^{-1-\sigma}]. \end{aligned}$$

Moreover, keeping the distribution of Y fixed, we have

$$\begin{aligned} I(n, A+1) &= -\frac{\sigma}{\Gamma(1-\sigma)} \int_0^1 y^{A-n}(1-y)^n(-\log(y))^{-1-\sigma} \mathrm{d}y \\ &= -\frac{\sigma}{\Gamma(1-\sigma)} \mathrm{B}(A-n+1, n+1) \int_0^1 f(y; A-n+1, n+1)(-\log(y))^{-1-\sigma} \mathrm{d}y \\ &= -\frac{\sigma}{\Gamma(1-\sigma)} \mathrm{B}(A-n+1, n+1) \\ &\times \int_0^1 f(y; A-n, n+1) \frac{f(y; A-n+1, n+1)}{f(y; A-n, n+1)} (-\log(y))^{-1-\sigma} \mathrm{d}y \\ &= -\frac{\sigma}{\Gamma(1-\sigma)} \mathrm{B}(A-n, n+1) \mathbb{E}[Y(-\log(Y))^{-1-\sigma}]. \end{aligned}$$

If $y_1, \ldots, y_N$ is a sample from Y, then

$$I(n, A) \approx -\frac{\sigma}{\Gamma(1-\sigma)} \mathrm{B}(A-n, n+1) \frac{1}{N} \sum_{j=-1}^{N} (-\log(y_j))^{-1-\sigma}$$

and

$$I(n, A+1) \approx -\frac{\sigma}{\Gamma(1-\sigma)} \mathrm{B}(A-n, n+1) \frac{1}{N} \sum_{j=1}^{N} y_j(-\log(y_j))^{-1-\sigma}.$$

Thus

$$\frac{I(n, A+1)}{I(n, A)} \approx \frac{\sum_{j=1}^N y_j(-\log(y_j))^{-1-\sigma}}{\sum_{j=1}^N (-\log(y_j))^{-1-\sigma}}.$$

While not strictly necessary, we noticed that the use of the same sample for both numerator and denominator guarantees numerical stability. The intuition is that the Monte Carlo error goes in the same direction on both parts of the ratio.

6 Data Analysis

In this section, we apply our theoretical results to data and thus gauge the performance of the model. First we generate data from a known distribution so that we can compare the results to the true distribution. Then we will turn our attention towards the real data that motivated our project.

In both cases, we will calculate pointwise estimates of the posterior mean derived in Proposition 1 as well as credible intervals for those estimates.

6.1 Simulation study

6.1.1 How the Data Are Generated

Our goal is to generate multiple groups of discrete survival data where there exists some dependence between groups that our model might infer using sharing of information.

We construct the necessary dependent processes using combinations of independent Dirichlet processes. Consider four such processes:

$$Q_0 \sim \mathrm{DP}(G_0, c)$$
$$Q_1 \sim \mathrm{DP}(G_1, c)$$
$$Q_2 \sim \mathrm{DP}(G_2, c)$$
$$Q_3 \sim \mathrm{DP}(G_3, c).$$

Now we construct and sample from three dependent processes given by

$$P_1 = 0.5 \times Q_0 + 0.5 \times Q_1$$
$$P_2 = 0.5 \times Q_0 + 0.5 \times Q_2$$
$$P_3 = 0.5 \times Q_0 + 0.5 \times Q_3.$$

From linearity and the properties of the Dirichlet process, we write down the expected values of these processes:

$$\mathbb{E}[P_l] = 0.5 \times \mathbb{E}[Q_0] + 0.5 \times \mathbb{E}[Q_l] \tag{38}$$

$$= 0.5 \times G_0 + 0.5 \times G_l \;\; l = 1, 2, 3. \tag{39}$$

This is useful to us because we can now judge the performance of the model by properly comparing its results to the objects it is estimating.

It remains to specify a value for the total mass c and appropriate distributions for the base measures $G_j,\ j = 0, 1, 2, 3$. For each measure, we choose a Weibull distribution.

Even though the Weibull is a continuous distribution, our simulated data will be effectively discrete. This is because of a property of the Dirichlet process, whereby samples which have already been drawn can be redrawn with positive probability.

There are a number of variables whose values we need either to specify or to provide a prior distribution for. As for this section the model is specified as follows:

$$\begin{aligned} p(\sigma) &\sim \text{ Unif}(0,1) \\ p(\gamma) &\sim \text{ Gamma}(a_\gamma, b_\gamma);\;\; a_\gamma = 1;\; b_\gamma = 0.2 \\ p(N_l) &\sim \text{ Pois}(\lambda);\quad \lambda = \max_l(M_l) \\ p(c) &\sim \text{ Gamma}(a_c, b_c);\quad a_c = b_c = 1 \\ z &= 0.5, \end{aligned}$$

where the value of z was here chosen to reflect the degree of dependence we want the model to infer. We will return to this point in the discussion section.

ϕ, the parameter of the Poisson distribution used in the proposal stage of N_l's Metropolis-Hastings algorithm, needs to be tuned on the fly so as to ensure adequate mixing of the algorithm, as per Roberts et al. (1997).

Another important quantity we need to consider is the number of samples we draw from each process P_l, $l = 1, 2, 3$; in other words, we need to set N_l, $l = 1, 2, 3$. Part of our approach is seeing how the model's behaviour changes with the number of observations.

Censoring times for each group also have to be specified but first we specify the parameters of the Weibull base measures of our Dirichlet processes:

$$G_0 \sim \text{Weibull}(k_0 = 5, \lambda_0 = 2.5)$$

$$G_1 \sim \text{Weibull}(k_1 = 2, \lambda_1 = 1)$$
$$G_2 \sim \text{Weibull}(k_2 = 3, \lambda_2 = 1.5)$$
$$G_3 \sim \text{Weibull}(k_3 = 4, \lambda_3 = 2)$$

We want to test the capacity of the model to use sharing of information to calculate estimates. To this end, we simulate our data in a way that allows us to censor many observations from the shared component for group 1, say. Ideally our model would infer the behaviour of group 1 nonetheless by borrowing information from groups 2 and 3.

This idea is illustrated in Figure 2. The censoring times for groups 2 and 3 are then chosen so as to censor approximately 5-10% of data.

6.1.2 Results of Analysis

The three values of n we consider are $30, 100$, and 200. We will examine how varying the sample size affects the degree to which information is shared and the sizes of the estimates' credible intervals.

An important consideration is that our model's estimation of N_l is very sensitive to the prior distribution we attribute to it.

For instance, if the mean λ of a Poisson prior is too low, N_l is generally underestimated for those releases with shorter censoring times. This leads to the corresponding posterior mean estimates sharply increasing and departing from the true distribution.

Conversely, a high value for λ results in overestimation of N_l for the releases with longer censoring times, leading to the opposite problem. We set $\lambda = \max_l(M_l)$ as this seems to be the best compromise.

Under ideal circumstances, the prior placed on N_l is informative, reliable, and based on expert opinion. Outside of such circumstances however, one risks encountering problems of identifiability.

At the same time it is worth stressing that the goal of our model is not to extrapolate beyond the maximum censoring time $\max_l(M_l)$ but rather to exploit borrowing of information across releases in order to get more accurate estimates of the survival curves in the interval

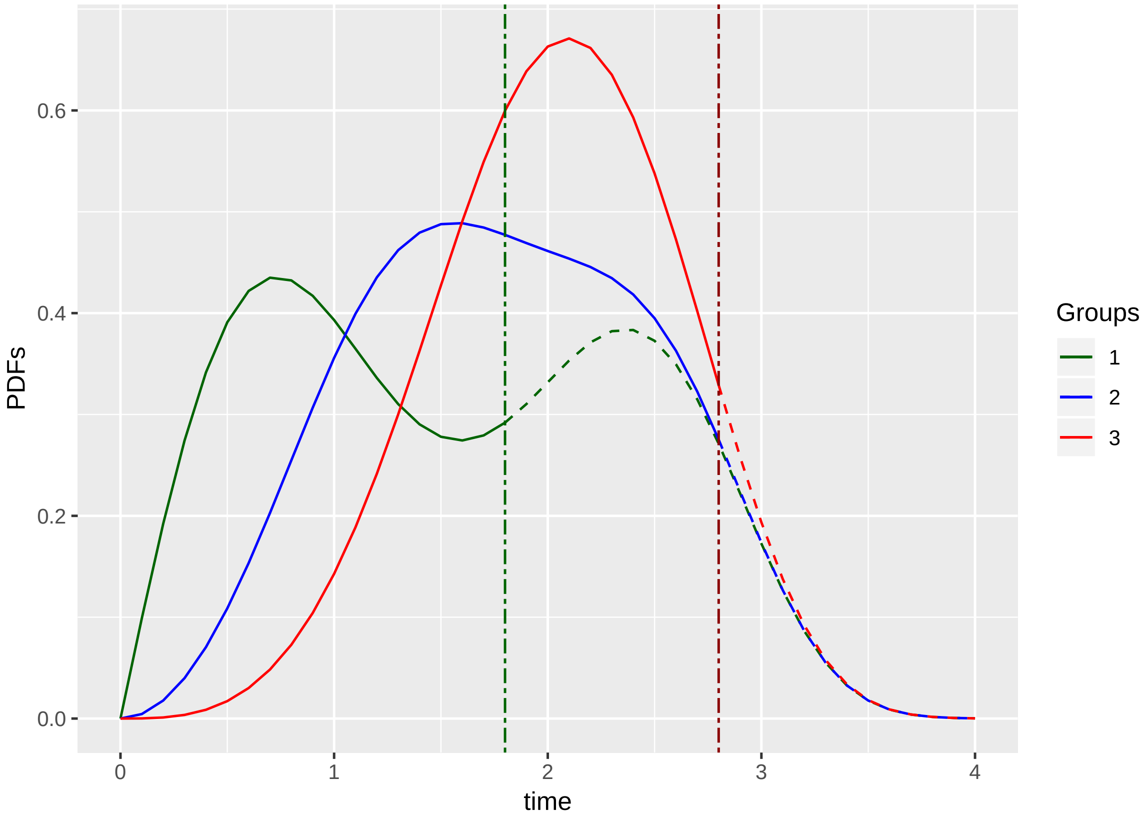


Figure 2: Mean probability densities and censoring times of generated data

$(0, \max_l(M_l))$.

We consider two cases. First, in order to eliminate any erroneous effects arising from identifiability issues with N_l, we specify an "oracle" prior, fixing N_l, $l = 1, 2, 3$ to their true values. Then we consider the Poisson prior for N_l specified in section 3.

Figure 2 depicts the densities specified in (38). These densities correspond to the distributions we attempt to estimate.

Analysis with oracle prior

First we generate 30 values from each $P_l, l = 1, 2, 3$. In Figure 3, the estimates of each of our three groups' posterior mean distributions are plotted along with the corresponding Kaplan-Meier estimates and true distributions. Figures 4 and 5 show the same plots for the cases where we draw 100 and 200 values respectively.

It seems clear that the estimates and the empirical distribution grow closer when more information is available. This makes sense, especially given that the empirical distribution will more closely approximate the true distribution, and therefore a smooth distribution, with more samples.

We note also that credible bounds are smaller when more information is available. Indeed this is how we would expect a statistical model to react when it is given more data.

When there is a dearth of data (see Figure 3), the estimates seem to drift towards each other. That is the estimates for $\tilde{F}_1$ err on the low side of its empirical distribution and the estimates for $\tilde{F}_2$ and $\tilde{F}_3$ err on the high side of theirs, as though they are approaching one another.

We might speculate that this is because of an increased reliance on sharing of information in the absence of exact observations. However, there is insufficient evidence to confirm this.

Of course it is important that we consider the behaviour of the model in the case of $\tilde{F}_1$, where the earlier censoring time imposed that less than half of that group's original sample was available to inform the posterior. We observe that the shape of the $\tilde{F}_1$ posterior estimate is very similar to those of $\tilde{F}_2$ and $\tilde{F}_3$ in Figures 3, 4, and 5. In fact it lies quite close to the true distribution in each case. We can appreciate here that the borrowing of information between groups has benefitted the model's performance.

Analysis with Poisson prior

Now we examine the case where the N_l are considered random. These results will be more reflective of the model's performance in practice.

We notice immedately in Figures 6, 7, and 8 that the credible intervals for these results are much larger than in the previous case. This makes sense because the N_l are no longer constant and so there is greater fluctuation in the estimates produced by the Gibbs sampler. Nonetheless, the intervals shrink, as before, given more information.

Another similarity to the oracle prior case is that there is "drifting" between the groups in the case where fewer data are available, that is, in Figure 6, for instance. In addition, we

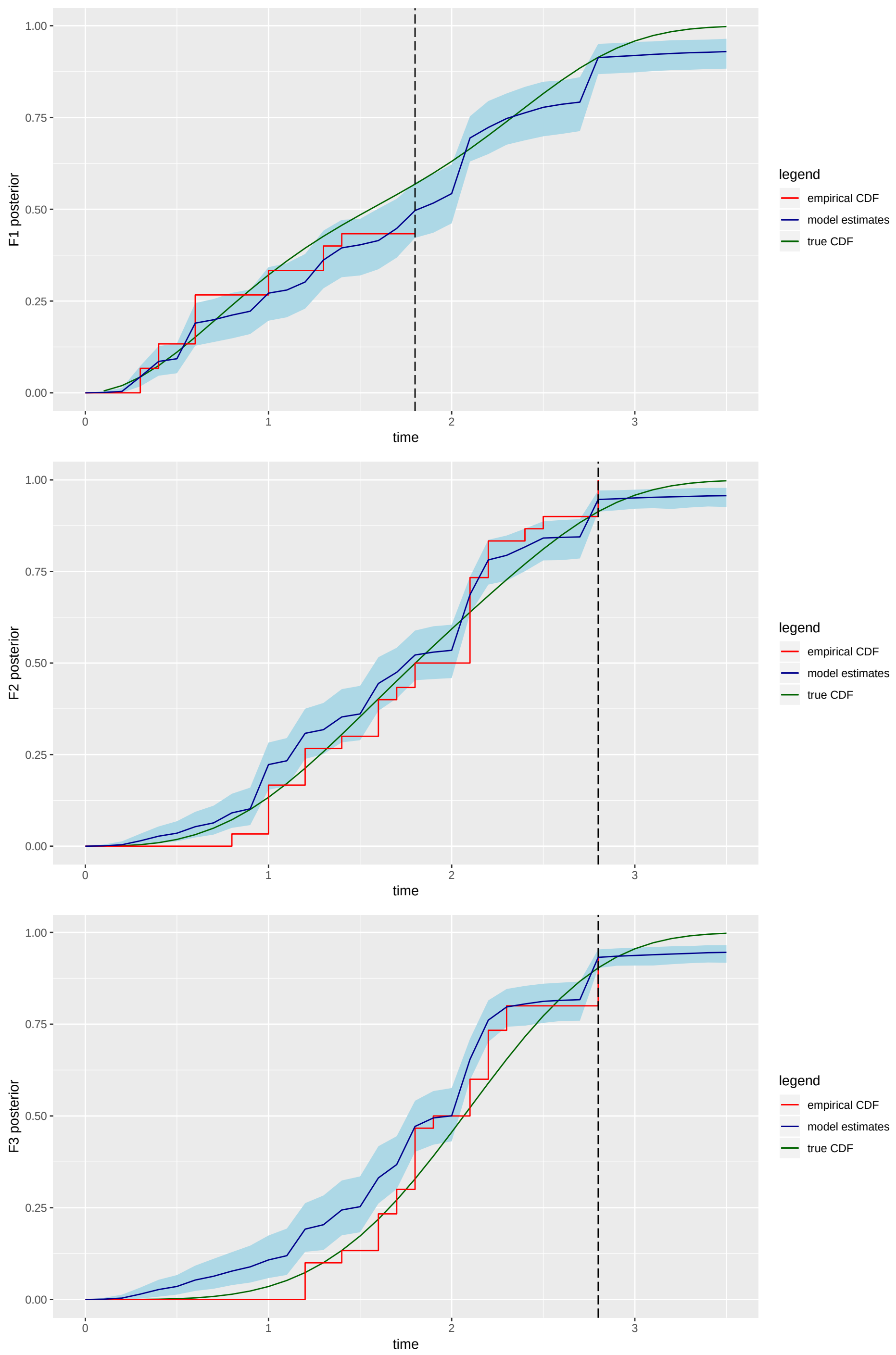


Figure 3: Model estimates for $n = 30$ and true N_l

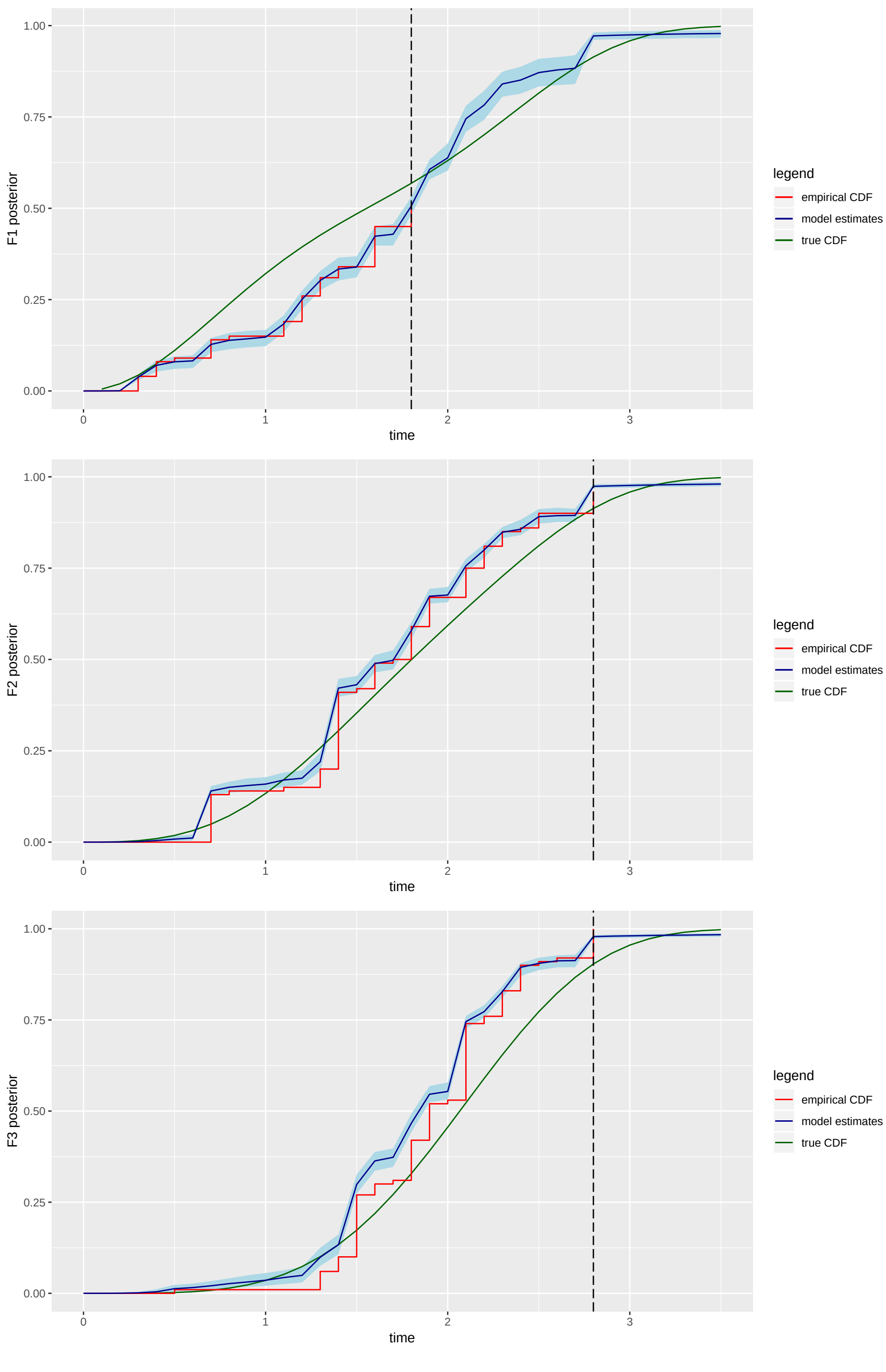


Figure 4: Model estimates for $n = 100$ and true N_l

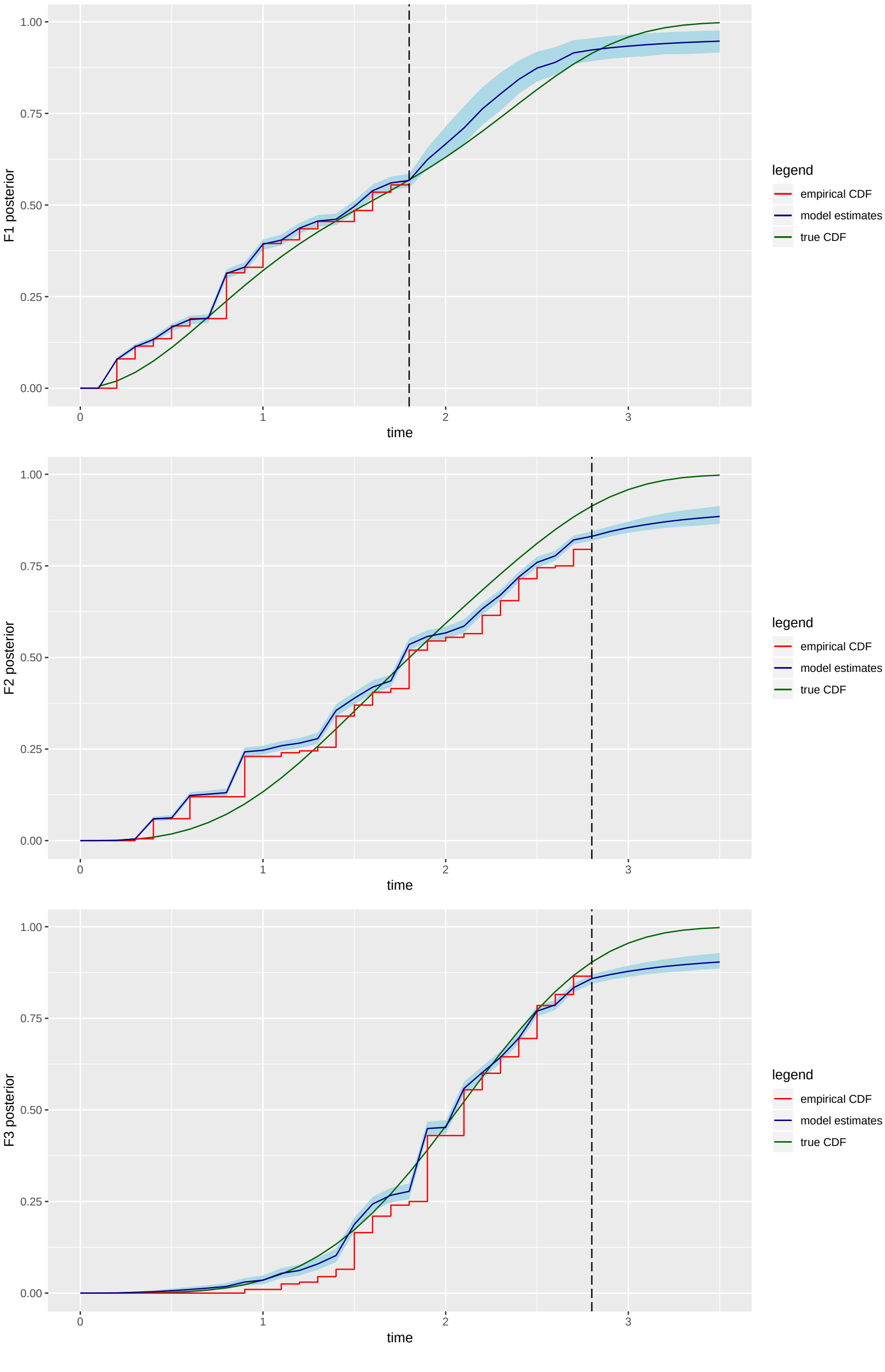


Figure 5: Model estimates for $n = 200$ and true N_l

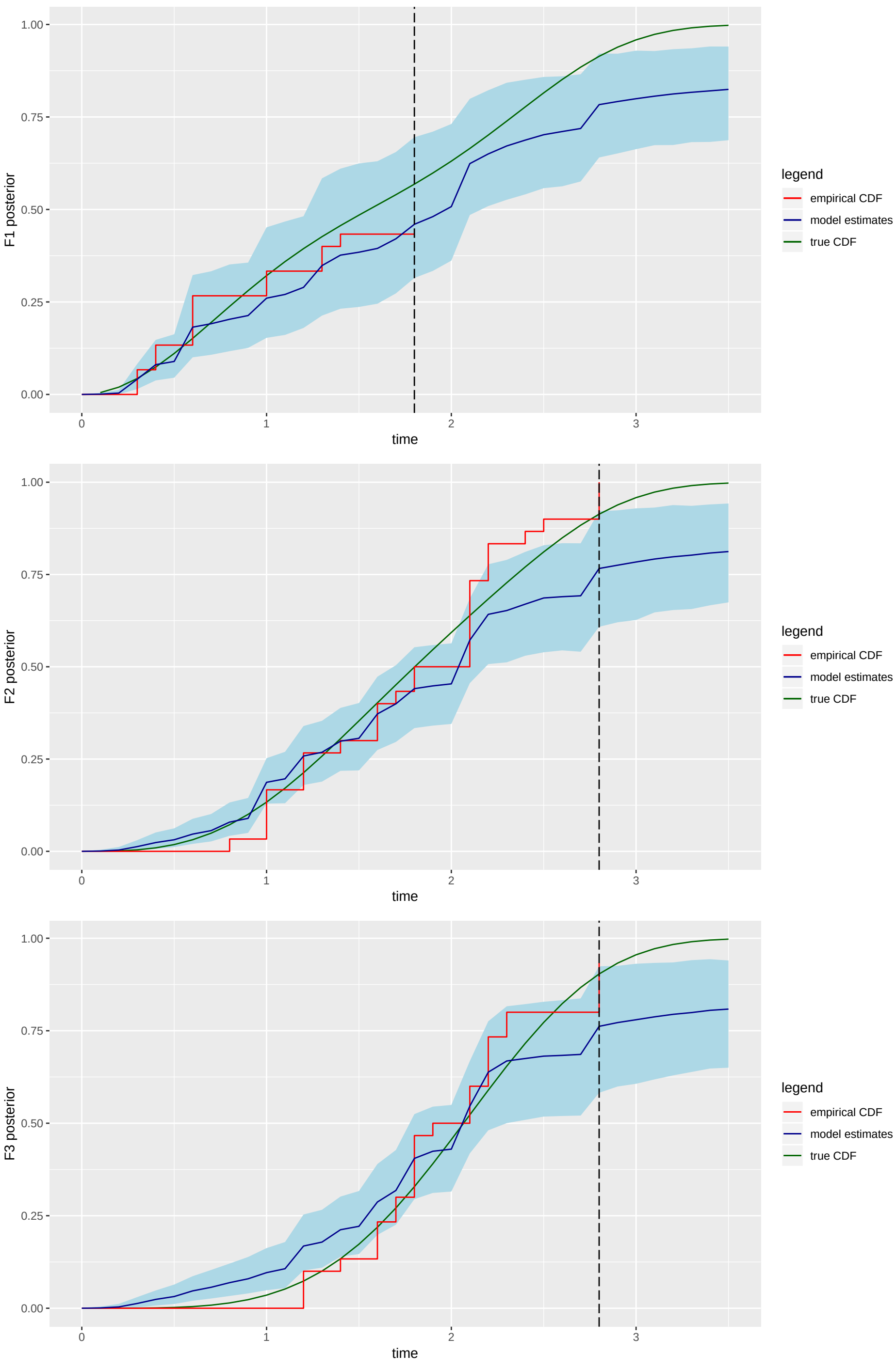


Figure 6: Model estimates for $n = 30$ and random N_l

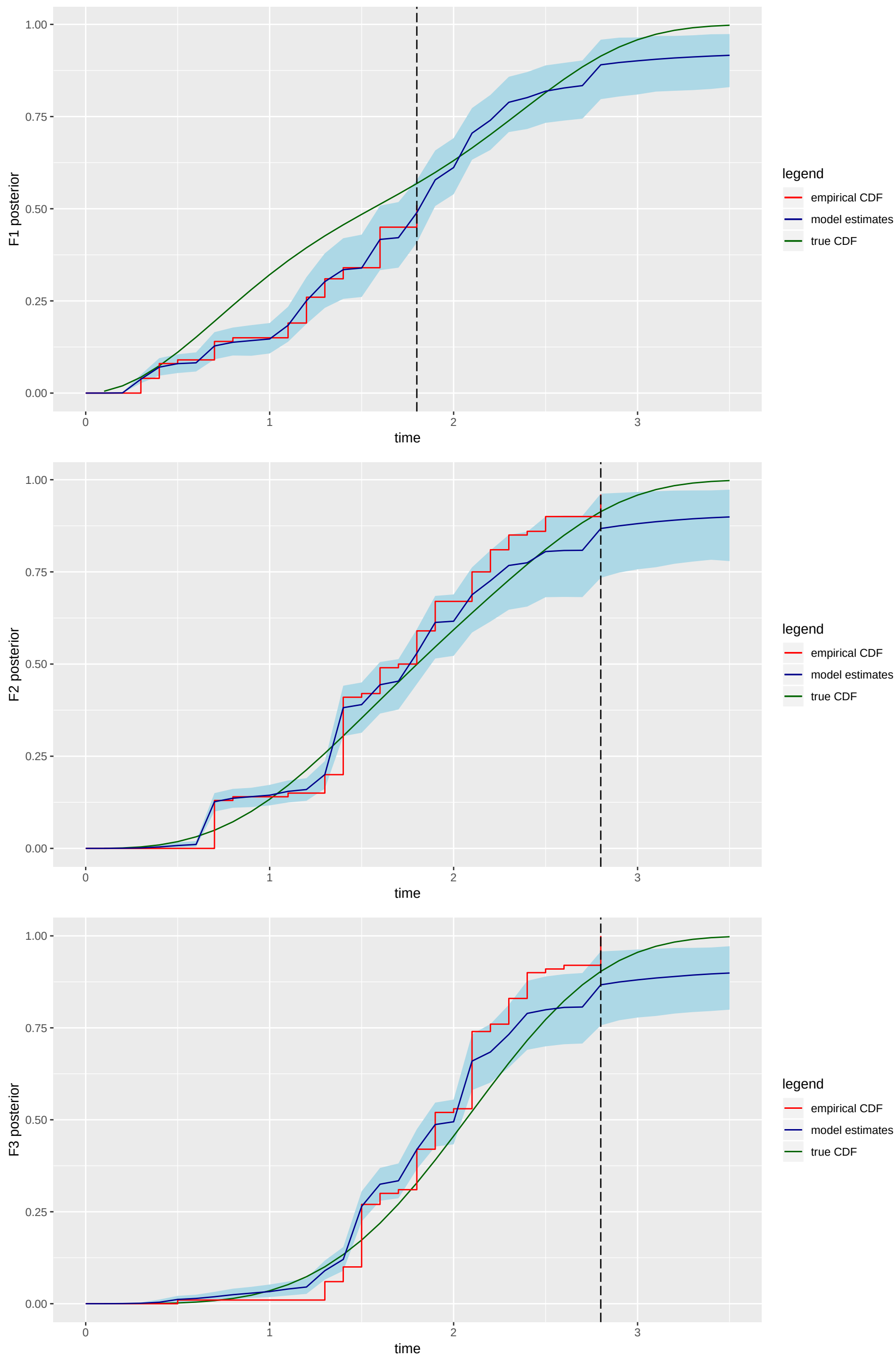
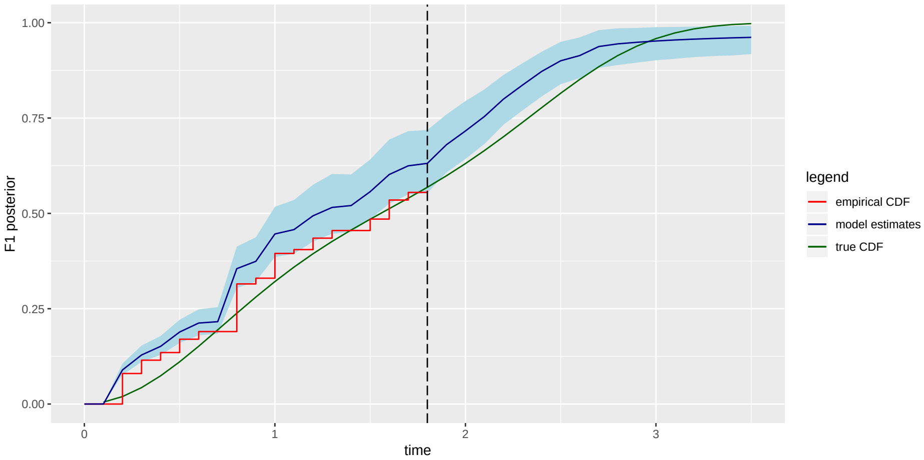
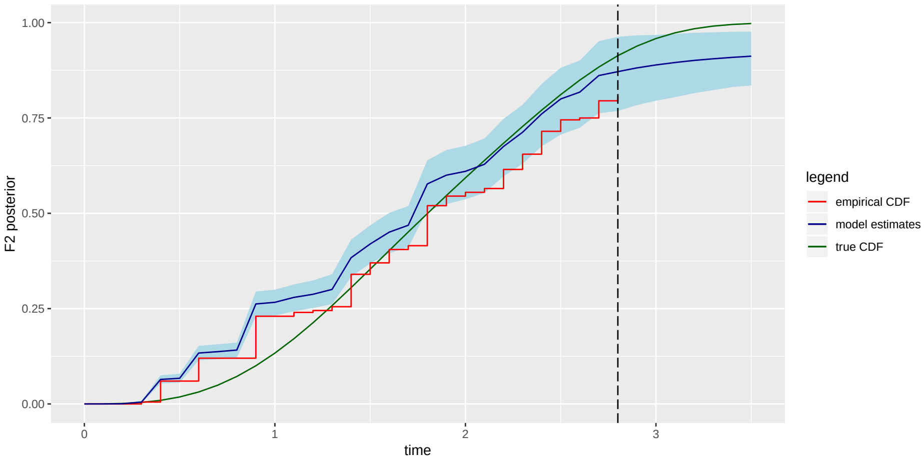


Figure 7: Model estimates for $n = 100$ and random N_l





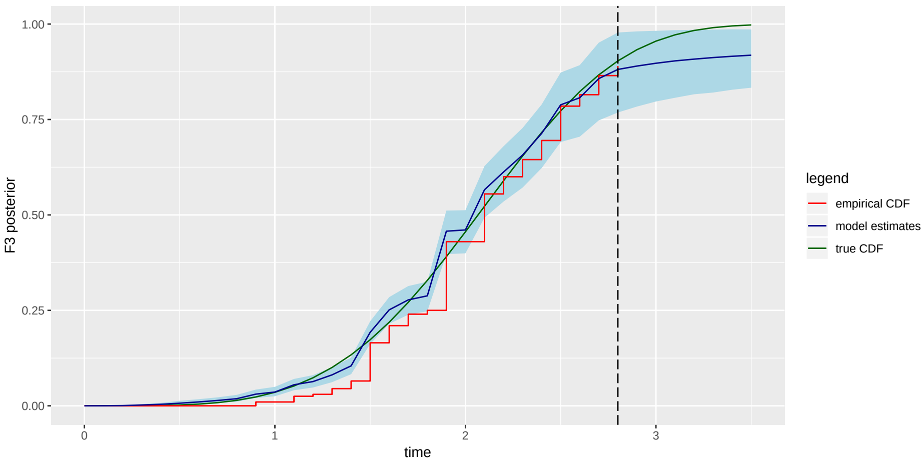


Figure 8: Model estimates for $n = 200$ and random N_l

again see the borrowing of information at work in Figures 6, 7, 8, informing the posterior estimates of $\tilde{F}_1$ after its early censoring time in each case.

It should be noted that in both of the above cases the Kaplan-Meier estimates are "cheating" in that these estimates are informed by the pre-censored data whereas, in this second part of our analysis, our model only has the censored data to work with.

6.2 Real data

Shown in Figure 9 are examples of the estimates we obtained for three of the 19 Firefox releases we analysed. While we cannot compare our modelling of the Firefox bug data with any "true" distributions, there are comments we can make.

For example, we look at how the model seems to react when confronted with shorter censoring times, as we did in the simulation case. The data in releases 24 and 25, the latest two in our dataset, are cut short by earlier censoring times than others because the dataset was composed before further observations could be recorded.

The third plot in Figure 9 depicts the estimated posterior mean distribution for release 24 and the model continues smoothly past the censoring time, seemingly relying on information borrowed from its peers, as expected.

In all three plots in Figure 9, the credible bounds widen with time, as we saw in the simulation study results.

Figure 10 depicts the estimated posterior distribution functions of all 19 releases that were analysed. The vertical dotted lines highlight their release dates and each release's censoring time is shown by the dashed lines of the estimates after those points.

A notable feature of Figure 10 is that many of the estimated distribution functions appear to plateau before getting close to 1. This is not necessarily an issue. We have seen credible bounds grow wider with time and so it is possible that the upper bounds grow towards 1. Alternatively this behaviour could be a feature of the Lévy intensity specifying the model in this case.

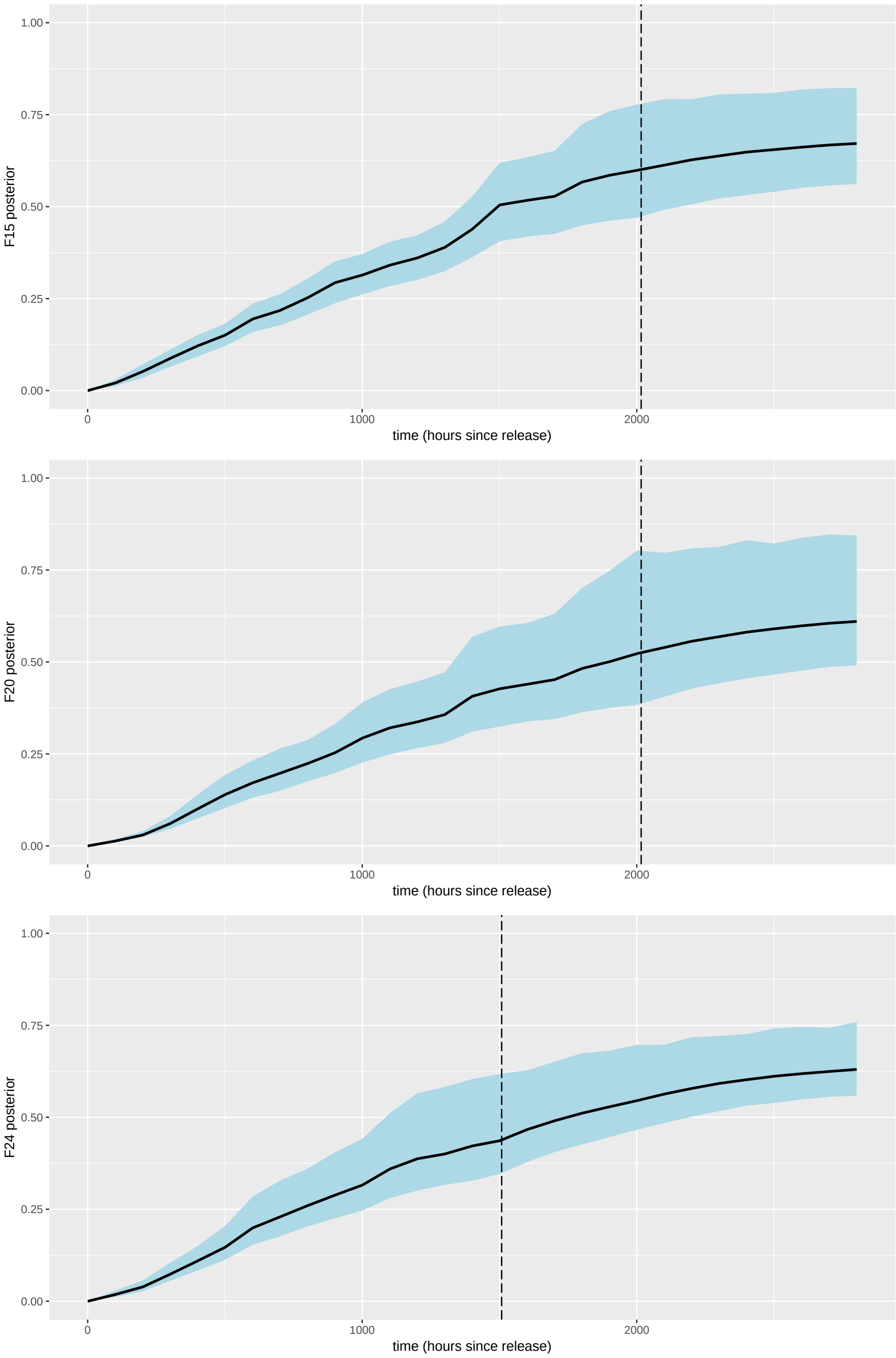


Figure 9: Model estimates and credible intervals of Firefox data distributions for the 15th, 20th, and 24th releases

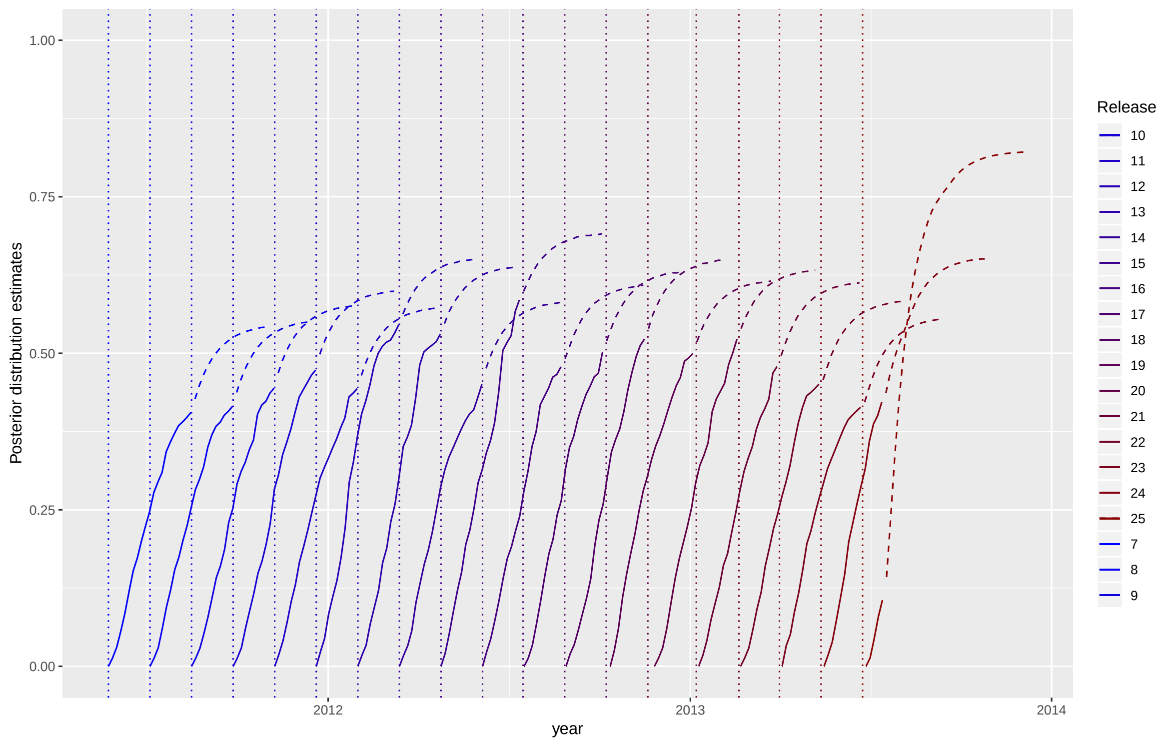


Figure 10: Model estimates pre and post censoring for releases 7-25

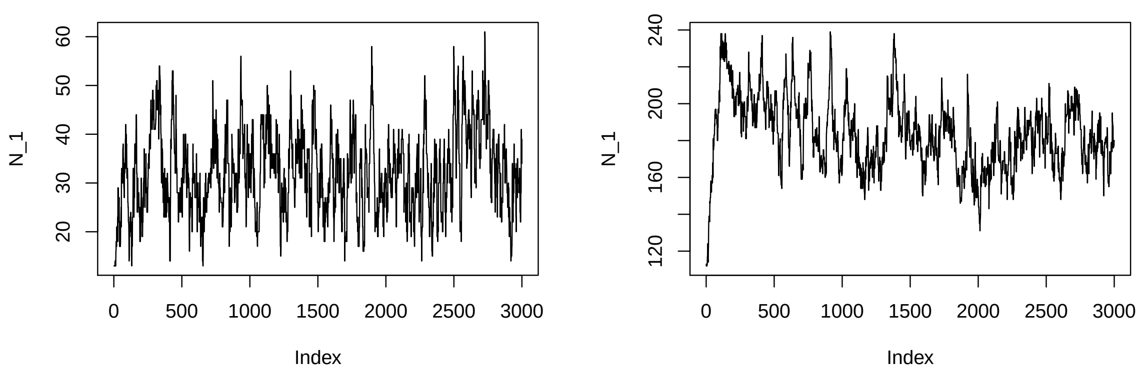


Figure 11: Trace plots of N_1 for sample sizes $n = 30$ (left) and $n = 200$ (right)

6.3 Sampler diagnostics

It is important to note that the results illustrated in this section are specious if our Gibbs sampler is for some reason unreliable or malfunctioning. Therefore we examine the estimates of all variables produced in order to ensure convergence is achieved.

N_l, $l = 1, \ldots, r$ are key quantities in the sampling process. Traceplots produced for N_1 in the simulation study can be seen in Figure 11. On visual inspection, convergence and adequate mixing of the Metropolis-Hastings algorithm are apparent.

Only values realised after a burn-in period of 2000 iterations were used to calculate estimates. This measure is taken to give the sampler time to reach equilibrium.

We noted that the time required to run the sampler and to compute the results increased with the size of the dataset, as expected. It also rose sharply with the number of groups of data. For the simulated data (3 groups), the process took 5 minutes for the $N = 30$ case and 10 minutes for the $N = 200$ case. For the real data (19 groups and 766 observations), the time taken was over 3 hours.

The results were obtained using RStudio on a machine with 8 GB RAM and an i5 dual-core processor.

7 Discussion

Our model as per the above results has been shown to perform on our chosen simulation study and on our real data. It remains to discuss its advantages, its drawbacks, and its potential improvements going forward.

7.1 Censoring times

One of the main advantages of the model was that it is able to predict the behaviour of releases with earlier censoring times more accurately because it was allowed to borrow information from those releases' late censored counterparts. We see this feature of the model in Figure 7, for instance.

7.2 Choice of Lévy intensity

Our choice of prior Lévy intensity $\nu(\mathrm{d}s, \mathrm{d}y)$ was a homogeneous one with σ-stable density ρ and one-parameter Gamma measure α. These decisions were chiefly for numerical convenience. Theorem 3 and Corollary 1, however, hold for any homogeneous Lévy intensity. This offers the model a great degree of flexibility.

The only potential difficulty is the evaluation of the integrals in (17). Indeed our choice of Lévy intensity required delicate treatment when obtaining an expression for the posterior estimator in Proposition 1.

7.3 Using all the information

One aspect of our motivating illustration which did not inform our model was the fact that the releases in the Firefox data are time-ordered. One could undertake to construct a model which accounted for time-ordering of the releases but this was not our goal in this project.

Our goal was rather to construct a model under the assumption that the releases are exchangeable and identically distributed a priori. As such, our model can be applied to data whose "releases" may simply be distinct groups with some underlying interdependence which we wish to describe.

7.4 Discrete versus continuous data

Our model is built for discrete data. Hence, as highlighted in Theorem 3, the occurence of ties in the data is given positive probability and used to inform the posterior. The data we analysed comprised bug report times per hour. There was a positive probability of finding ties in the data and indeed, we discovered ties which helped to inform the model.

We believe the model could work with data which is continuous but in order for the model to work well, this data may require discretisation. Here, the question arises of how wide or narrow to make the intervals into which the data is discretised. If they are too wide, we lose too much information and the model will be poorly informed. If they are too narrow, we encounter difficulty sharing information. So a balance must be struck.

References

DOKSUM, K. (1974). Tailfree and neutral random probabilities and their posterior distributions. *The Annals of Probability*, 183–201.

EPIFANI, I., and LIJOI, A. (2010). Nonparametric priors for vectors of survival functions. *Statistica Sinica*, 1455–1484.

GHOSAL, S. and VAN DER VAART, A. (2017). Fundamentals of nonparametric Bayesian inference. *Cambridge University Press*, **44**.

GRIFFIN, J. E. and LEISEN, F. (2014). Compound random measures and their use in Bayesian nonparametrics. *arXiv:1410.0611v3 [stat.ME] 2 Sep 2015.*

KINGMAN, J. F. C. (1967). Completely random measures. *Pacific Journal of Mathematics* **21**(1), 59–78.

LIJOI, A. and NIPOTI, B. (2014). A class of hazard rate mixtures for combining survival data from different experiments. *Journal of the American Statistical Association*, **109**(506), 802–814.

LIJOI, A., NIPOTI, B., and PRÜNSTER, I. (2014). Bayesian inference with dependent normalized completely random measures. *Bernoulli*, **20**(3), 1260–1291.

LIJOI, A. and PRÜNSTER, I. (2009). Models beyond the Dirichlet process. *Carlo Alberto Notebooks* **129**.

MÜLLER, P., QUINTANA, F. A., JARA, A., and HANSON, T. (2015). Bayesian nonparametric data analysis. *New York: Springer*, **18**.

NIPOTI, B. (2011). Dependent completely random measures and statistical applications. *Doctoral Thesis, Department of Mathematics, University of Pavia.*

ROBERT, C. P. and CASELLA, G. (2004). Monte Carlo statistical methods. *New York: Springer.*

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Probability*, **7**(1), 110–120.

Wilson, S. P. and Ó'Ríordáin, S. (2018). Optimal software testing across version releases. *Analytic Methods in Systems and Software Testing*, 65–80.

Wilson, S. P. and Samaniego, F. J. (2007). Nonparametric analysis of the order-statistic model in software reliability. *Software Engineering, IEEE Transactions*, **33**(3), 198–208.

A Code for data simulation

```
set.seed(5)
#Total mass DP (larger values of cDP induce less ties and
#therefore an empirical CDF which is closer to the base
#measure)
cDP<-6


#Function to generate n exchangeable observations from
# a DP with base measure weibull with parameters shapeG
#and scaleG, and total mass c


rDPwbl <- function(n,a,shapeG,scaleG){
  genvec<-rep(0,n)
  gennew<-rweibull(n, shape=shapeG, scale=scaleG)
  genvec[1]<-gennew[1]
  j<-2
  while (j<=n){
    pick<-rmultinom(1,1,c(a/(a+j-1),rep(1/(a+j-1),j-1)))
    i<-1
    while (i<=j){
      if (pick[i]==1){
        ind<-i
        i<-j+1
        }
      else{i<-i+1}
    }
    if (ind==1){
      genvec[j]<-gennew[j]
```

```
    }else{
      genvec[j]<-genvec[ind-1]
    }
    j<-j+1
  }
  return(genvec)
}

#Generating data (these need to be set)
shape0 <- 5
scale0 <- 2.5
shape1 <- 2
scale1 <- 1
shape2 <- 3
scale2 <- 1.5
shape3 <- 4
scale3 <- 2

#This is half of the total sample size
sstotal <- 15

r0 <- rDPwbl(3*sstotal,cDP,shape=shape0, scale=scale0)
r1temp <- rDPwbl(sstotal,cDP,shape=shape1, scale=scale1)
r2temp <- rDPwbl(sstotal,cDP,shape=shape2, scale=scale2)
r3temp <- rDPwbl(sstotal,cDP,shape=shape3, scale=scale3)

r1 <- c(r1temp,r0[1:sstotal])
r2 <- c(r2temp,r0[(sstotal+1):(2*sstotal)])
r3 <- c(r3temp,r0[(2*sstotal+1):(3*sstotal)])
```

```
#ECDFs
plot(ecdf(r2),  col="green", main="Empirical CDFs",ylim=c(0,1))
lines(ecdf(r1), col="blue")
lines(ecdf(r3), col="red")

#PDFs
s <- seq(from=0.1, to=3.5, by=0.1)

pdf1 <- 0.5*dweibull(s, shape=shape1, scale=scale1)+
0.5*dweibull(s, shape=shape0, scale=scale0)
pdf2 <- 0.5*dweibull(s, shape=shape2, scale=scale2)+
0.5*dweibull(s, shape=shape0, scale=scale0)
pdf3 <- 0.5*dweibull(s, shape=shape3, scale=scale3)+
0.5*dweibull(s, shape=shape0, scale=scale0)

#For plots

sim <- data.frame(s, pdf1, pdf2, pdf3)
sim<-setNames(sim, c("time", "PDFs", "pdf2", "pdf3"))

cols <- c("1"="darkgreen", "2"="blue", "3"="red")

simplot <- ggplot(sim, xlab="PDFs",
colour=c(PDFs, pdf2, pdf3))
simplot_out<-simplot+
  geom_line(data=sim[s<=cen[1]+0.001,],
aes(x=time, y=PDFs, colour="1"))+
  geom_line(data=sim[s>=cen[1]-0.001,],
```

```
aes(x=time, y=PDFs, colour="1"),
            linetype="dashed")+
  geom_line(data=sim[s<=cen[2]+0.001,],
aes(x=time, y=pdf2, colour="2"))+
  geom_line(data=sim[s>=cen[2]-0.001,],
aes(x=time, y=pdf2, colour="2"),
            linetype="dashed")+
  geom_line(data=sim[s<=cen[3]+0.001,],
aes(x=time, y=pdf3, colour="3"))+
  geom_line(data=sim[s>=cen[3]-0.001,],
aes(x=time, y=pdf3, colour="3"),
            linetype="dashed")+
  geom_vline(xintercept=cen[1], linetype="twodash",
color="darkgreen")+
  geom_vline(xintercept=cen[2], linetype="twodash",
 color="darkred")+
  scale_colour_manual(name="Groups", values=cols)+
ylab("PDFs")

pdf(file="simdata.pdf", width=7, height=5)
simplot_out
dev.off()

#CDFs
cdf1 <- 0.5*pweibull(s, shape=shape1, scale=scale1)+
0.5*pweibull(s, shape=shape0, scale=scale0)
cdf2 <- 0.5*pweibull(s, shape=shape2, scale=scale2)+
0.5*pweibull(s, shape=shape0, scale=scale0)
cdf3 <- 0.5*pweibull(s, shape=shape3, scale=scale3)+
```

```
0.5*pweibull(s, shape=shape0, scale=scale0)

#for the ecdf at the end
r1t <- r1
r2t <- r2
r3t <- r3

#censoring
cen <- c(1.8, 2.8, 2.8)
r1<-r1[r1<cen[1]]
r2<-r2[r2<cen[2]]
r3<-r3[r3<cen[3]]

meen <- mean(c(r1, r2, r3))

r1 <- r1/meen
r2 <- r2/meen
r3 <- r3/meen

R <- data.frame(t=c(r1, r2, r3),
release=c(rep(1, length.out=length(r1)),
         rep(2, length.out=length(r2)),
         rep(3, length.out=length(r3))))
```

B Code for data analysis

```
library("plyr")
library("ggplot2")

r=3 #number of releases

M <- sapply(1:r, FUN=function(ll){
return(sum(R$release==ll))})
#numbers of uncensored observations in each release

C <- count(R, vars=c("t", "release")) #tallies of data

dis_T <- c(0, C$t, cen/meen)
dis_T <- dis_T[!duplicated(dis_T)] #distinct observations
dis_T <- dis_T[order(dis_T)] #dis_T order failsafe
k <- length(dis_T[-1]) #number of distinct observations

#indices of group censoring times
k_l <- sapply(1:r, function(l_){
return(which(dis_T%in%(cen[l_]/meen)))})

n_f <- function(i, l){
return(sum(C$freq[C$t==dis_T[i+1] & C$release==l]))}
n_mat <- outer(1:k, 1:r, FUN=Vectorize(n_f))

#Boolean variables indicating whether distinct observations are
#unique to release l or shared between multiple releases
sort_I<- function(rel){
```

```
  u <- C$t[C$release==rel & !C$t%in%
intersect(C$t[C$release==rel], C$t[C$release!=rel])]
  return(dis_T[-1] %in% u)
}
I_mat <- sapply(X=1:r, FUN=sort_I)
I_mat <- cbind(I_mat, rowSums(I_mat)==0)

n_bar <- rowSums(n_mat)
n_bar_mat <- function(j, l){
  if(j==k){return(0)}else{return(sum(n_mat[j:(k-1), l]))}
}
n_bar_mat <- Vectorize(n_bar_mat)
n_bar_mat <- outer(1:k, 1:r, FUN = n_bar_mat)
#for convenience

#Full conditional functions

MC_ratio <- function(A_, n_, N_=N_MC, sig){
  y_vec <- rbeta(N_, A_-n_+1, n_+1)
  num_vec <- (-log(y_vec))^(-1-sig)
  den_vec <- (1/y_vec)*num_vec
  return(sum(num_vec)/sum(den_vec))
}#Monte Carlo approximation of tricky ratio

MC_term <- function(A_, n_, N_=N_MC, sig){
  y_vec <- rbeta(N_, A_-n_+1, n_+1)
  num_vec <- (-log(y_vec))^(-1-sig)
  lconst <- log(sig) -lgamma(1-sig) + lbeta(A_-n_+1, n_+1) -
 log(N_) + log(sum(num_vec))
```

```
  ans <- exp(lconst)
  return(ans)
}

log_lambda_post <- function(lamb, N_vec, V_mat, sigm, c_){
  arg <- function(i){
    return(dgamma(dis_T[i+1],rate=lamb,shape=lamb,log = TRUE)-
             (c_*(pgamma(dis_T[i+1],rate=lamb,shape=lamb)-
                   pgamma(dis_T[i],rate=lamb,shape=lamb)))*
             ((z*sum((n_bar_mat[i, ]+N_vec-M)^sigm))
              +(1-z)*(sum(n_bar_mat[i,]+N_vec-M))^sigm))
  }
  sumterm <- sapply(X=1:k, FUN=arg)
  sumterm <- sumterm+dgamma(lamb, shape=a_l, rate=b_l)
#gamma prior term
  return(sum(sumterm))
}
draw_l_post <- function(grd, Nd, Vd, sw, cw){
  log_weights <- sapply(X=grd, FUN=log_lambda_post,
                        N_vec=Nd, V_mat=Vd, sigm=sw, c_=cw)
  log_weights <- log_weights-mean(log_weights)
  weights <- exp(log_weights)
  return(sample(grd, size=1, prob=weights))
}

log_sigma_post <- function(sgm, lmd, cc, N_ve, V_ma){
  llp <- log_lambda_post(lamb=lmd, N_vec=N_ve,
V_mat=V_ma, sigm=sgm, c_=cc)
  return(llp + A_u(Nloc=N_ve, Vloc = V_ma, sg=sgm) +
```

```
A_s(N_ve, sg=sgm))
}
draw_s_post <- function(grd, Nd, Vd, lw, cw){
  log_weights <- sapply(X=grd, FUN=log_sigma_post,
N_ve=Nd, V_ma=Vd, lmd=lw, cc=cw)
  log_weights <- log_weights-max(log_weights)
  weights <- exp(log_weights)
  return(sample(grd, size=1, prob=weights))
}

A_u <- function(Nloc, Vloc, sg){
  prod_term_u <- function(i_prod){
    l_prod <- which(n_mat[i_prod,]!=0)
    AuA <- n_bar_mat[i_prod, l_prod]+
      (Nloc[l_prod]-M[l_prod])*(i_prod<=k_l[l_prod])+
Vloc[i_prod,l_prod]*
      (sum(n_bar_mat[i_prod, -l_prod])+
         sum((Nloc[-l_prod]-M[-l_prod])*(i_prod<=k_l[-l_prod])))
    Aun <- n_mat[i_prod, l_prod]
    Au_term <- MC_term(A_ = AuA, n_ = Aun, sig=sg)
    return(Au_term)
  }
  tes <- sapply(X=which(!I_mat[,r+1]), FUN=prod_term_u)
  return(sum(log(tes)))
}

A_s <- function(Nvec, sg){
  prod_term_s <- function(i_prod){
    As_term <- MC_term(A_=sum(n_bar_mat[i_prod,]+(Nvec-M)*
```

```
(i_prod<=k_l)),
                       n_=n_bar[i_prod], sig=sg)
    return(As_term)
  }
  if(length(which(I_mat[,r+1]))==1)return(0)
  this <- sapply(X=which(I_mat[,r+1])[-sum(I_mat[,r+1])],
 FUN=prod_term_s)
  return(sum(log(this)))
}

N_fc_2 <- function(N_arg, N_vec, L, V_mat, c_, si, la){
  N_vec[L] <- N_arg
  arg <- function(i){
    l1 <- n_bar_mat[i, L]+(N_arg-M[L])*(i<=k_l[L])
    l2 <- sum(n_bar_mat[i,]+(N_vec-M)*(i<=k_l))
    if(l1<0)stop(paste("l1", l1))
    if(l2<0)stop(matrix(c(N_vec, M)))
    return(-(c_*(pgamma(dis_T[i+1],rate=la,shape=la)-
                   pgamma(dis_T[i],rate=la,shape=la)))*
             ((z*(n_bar_mat[i, L]+(N_arg-M[L])*(i<=k_l[L]))^si)
              +(1-z)*(sum(n_bar_mat[i,]+(N_vec-M)*(i<=k_l)))^si))
  }

  presum <- sapply(1:k, FUN=arg)
  term1 <- sum(presum)
  term2 <- A_u(N_vec, V_mat, sg=si)
  term3 <- A_s(N_vec, sg=si)
  term4 <- -plambda+(N_arg)*log(plambda)-lgamma(N_arg+1)
```

```
  log_Nfc <- lgamma(N_arg+1)-lgamma(N_arg+1-M[L]) +
 term1+term2+term3+term4
  if(is.nan(log_Nfc))stop(paste("up here", N_arg, lgamma(N_arg+1),
lgamma(N_arg+1-M[L]),
                                term1, term2, term3, term4))
  return(log_Nfc)
}
#log full conditional of N times Poisson prior


draw_N_2 <- function(N_v, L_, V_v, c_d, s_d, l_d){
  Y <- rpois(1, phi)
  log_g <- function(x){
    if(phi+x+1<1)stop(paste("lgamma argument less than 1:",
 phi+x+1))
    return(-phi+(phi+x)*log(phi)-lgamma(phi+x+1))
  }
  if(Y<phi-N_v[L_]+M[L_] | Y >2*phi)return(N_v[L_])else{
    Nstar <- N_v[L_]+Y-phi
    if(Nstar<M[L_])return(N_v[L_])
    log_alpha <- N_fc_2(Nstar, N_vec = N_v, L=L_, V_mat=V_v,
c_=c_d, si=s_d, la=l_d) -
      N_fc_2(N_v[L_], N_vec = N_v, L=L_, V_mat=V_v, c_=c_d,
si=s_d, la=l_d) +
      log_g(N_v[L_]-Nstar) - log_g(Nstar-N_v[L_])
    if(log(runif(1))<log_alpha){
      arc[L_] <- arc[L_]+1
      return(Nstar)
      }else{return(N_v[L_])}
  }
```

```
}

V_fc_u <- function(Nloc, Vloc, ploc, s_v){
  qloc <- which(n_mat[ploc,]!=0)
  Vfc_term <- MC_term(A_ <- n_bar_mat[ploc, qloc]+
(Nloc[qloc]-M[qloc])*
                        (ploc<=k_l[qloc])+
                        Vloc[ploc, qloc]*
(sum(n_bar_mat[ploc, -qloc])+
                  sum((Nloc[-qloc]-M[-qloc])*
                            (ploc<=k_l[-qloc]))),
                      n_ <- n_mat[ploc, qloc], sig=s_v)
  return(Vfc_term)
}
#full conditional of V

draw_V <- function(p_, q_, Ns, Vs, sd){
  if (p_ %in% which(I_mat[,r+1])){
    return(rbinom(1, size=1, prob=1-z))
  }
  else{
    Vs[p_,q_] <- 0
    V_f1 <- V_fc_u(Ns, Vs, p_, s_v=sd)
    if(V_f1<0 | is.na(V_f1))stop(paste("pointA", V_f1))
    lower <- V_f1*z
    Vs[p_,q_] <- 1
    V_f2 <- V_fc_u(Ns, Vs, p_, s_v=sd)
    if(V_f2<0 | is.na(V_f2))stop(paste("pointB", V_f2))
    upper <- V_f2*(1-z)
```

```
    prob_V<-upper/(lower+upper)
    return(rbinom(1, size=1, prob=prob_V))
  }
}
draw_V <- Vectorize(draw_V, vectorize.args = c("p_", "q_"))
#draws V; weights determined by V_fc_u

#Estimator for F_l posterior

post_dist <- function(t, N, V, l, c_, spd, lpd){
  i_0 <- max(which(dis_T<t))
  I <- I_mat[,l]
  f1 <- function(i_f1){
    t1 <- c_*(pgamma(dis_T[i_f1+1],rate=lpd,shape=lpd)-
                pgamma(dis_T[i_f1],rate=lpd,shape=lpd))
    t2 <- z*((1+n_bar_mat[i_f1, l]+(N[l]-M[l])*
(i_f1<=k_l[l]))^spd -
               (n_bar_mat[i_f1, l]+(N[l]-M[l])*
(i_f1<=k_l[l]))^spd)+
      (1-z)*((1+sum(n_bar_mat[i_f1,]+(N-M)*(i_f1<=k_l)))^spd -
               (sum(n_bar_mat[i_f1,]+(N-M)*(i_f1<=k_l)))^spd)
    ans <- t1*t2
    return(ans)
  }

  f2 <- function(i_f2){
    if(i_f2==1)return(0)else{
      return(sum(sapply(X=1:(i_f2-1), FUN=f1)))
    }
```

```
  }

  f3 <- function(i_f3){
    t1 <- c_*(pgamma(t,rate=lpd,shape=lpd)-
                pgamma(dis_T[i_f3],rate=lpd,shape=lpd))
    t2 <- z*((1+n_bar_mat[i_f3, l]+(N[l]-M[l])*
(i_f3<=k_l[l]))^spd -
               (n_bar_mat[i_f3, l]+(N[l]-M[l])*
(i_f3<=k_l[l]))^spd) +
      (1-z)*((1+sum(n_bar_mat[i_f3,]+(N-M)*
(i_f3<=k_l)))^spd -
               (sum(n_bar_mat[i_f3,]+(N-M)*
(i_f3<=k_l)))^spd)
    return(t1*t2)
  }

  if(i_0<=k){
    arg1 <- f2(i_0) + f3(i_0)
  } else {
    arg1 <- f2(i_0) + (c_*(pgamma(t,rate=lpd,shape=lpd)-
    pgamma(dis_T[k+1],rate=lpd,shape=lpd)))
  }

  g1 <- function(i_){
    return(MC_ratio(A_=n_bar_mat[i_, l]+
(N[l]-M[l])*(i_<=k_l[l]),
                   n_=n_mat[i_, l], sig=spd))
  }
```

```
  g2 <- function(i_, j){
    return(MC_ratio(A_=sum(n_bar_mat[i_,]+(N-M)*(i_<=k_l)),
n_=n_mat[i_, j], sig=spd))
  }

  g3 <- function(i_){
    return(MC_ratio(A_=sum(n_bar_mat[i_,]+(N-M)*(i_<=k_l)),
 n_=n_bar[i_], sig=spd))
  }

  h1 <- function(i){
    if(t<dis_T[i+1]){return(1)} else {
      if(i %in% which(I)){
        if(i==k & t<dis_T[k+1]){print(1)}
        if(V[i, l]==0){
          if(g1(i_=i)<0){stop("point1")}
          return(g1(i_=i))
        } else if(V[i, l]==1) {
if(g2(i_=i, j=l)<0){stop(paste("point2", g2(i_=i, j=l)))}
          return(g2(i_=i, j=l))
        }
      } else if (i %in% which(I_mat[,r+1])){
        if(i==k & t<dis_T[k+1]){print(2)}
        return(g3(i_=i))
      } else if (V[i, which(I_mat[i,])]==1){
        if(i==k & t<dis_T[k+1]){print(3)}
        if(g2(i_=i, j=which(I_mat[i,]))<0){stop("point3")}
        return(g2(i_=i, j=which(I_mat[i,])))
      } else {
```

```
        return(1)
      }
    }
  }
  ter <- sapply(X=1:(k-1), FUN=h1)

  #print(str(ter))
  arg2 <- prod(ter)
  if(is.nan(arg2))stop("hmm")
  if(arg2>exp(arg1)){print(paste(which(ter>1),
  "<- index of term making arg2>exp(arg1)"))}
  if(arg2<0){print(paste(which(ter<0) ,
 "<- index of term making arg2<0"))}
  answer <- 1-exp(-arg1)*arg2

  return(answer)
}
#posterior distribution of F_l given N, V, and data

#Gibbs sampler

burnin <-2000
sample_size <- 1000
end <- burnin+sample_size

#Metropolis-Hastings parameter
phi <- 20

#Markov chain sample size
```

```
N_MC=50

#alpha base measure scale parameter c (we use a Gamma prior)
c_init <- 1 # initial value for c
a <- 1
b_init <- 1
#prior values

s_init <- 0.5 # initial value for sigma
l_init <- 0.5 # initial value for gamma

a_post <- a+k-1
#posterior a value (b is updated
# iteratively as it depends on N)

z <- 0.5 #parameter determining weight given to "shared" data

plambda<-max(M) # parameter for Poisson prior on N

N <- matrix(rep(NA, end*r), ncol = r)
V <- array(rep(NA, end*r*k), dim = c(k, r, end))
c_vec <- rep(NA, end)
sig_vec <- rep(NA, end)
lam_vec <- rep(NA, end)
N[1,] <- M
V[,,1] <- matrix(rbinom(r*k, 1, 0.5), nrow=r)
c_vec[1] <- c_init
sig_vec[1] <- s_init
lam_vec[1] <- l_init
```

```
N_slice <- N[1,]
V_slice <- V[,,1]
c_slice <- c_vec[1]
b_slice <- b_init
s_slice <- sig_vec[1]
l_slice <- lam_vec[1]

#values over which to evaluate the posteriors
#of sigma and gamma
griddy_s <- seq(from=0.1, to=0.9, by=0.1)
griddy_l <- seq(from=0.1, to=10, by=0.5)
a_l <- 1   #shape and rate for prior on lambda (phi)
b_l <- 0.2

s1 <- function(i_s){
  t1 <- (pgamma(dis_T[i_s+1],rate=l_slice,shape=l_slice)-
           pgamma(dis_T[i_s],rate=l_slice,shape=l_slice))
  t2 <- z*sum((n_bar_mat[i_s,]+
(N_slice-M)*(i_s<=k_l))^s_slice) +
    (1-z)*(sum(n_bar_mat[i_s,]+
(N_slice-M)*(i_s<=k_l)))^s_slice
  return(t1*t2)
}#function used in Gibbs sampling

for(iter in 2:end){
  N_slice <- sapply(1:r, FUN=draw_N_2, N_v=N_slice,
V_v=V_slice, c_d=c_slice,
                    s_d=s_slice, l_d=l_slice)
  V_slice <- base::outer(1:k, 1:r, FUN=draw_V, Ns=N_slice,
```

```
Vs=V_slice, sd=s_slice)

  sum <- sapply(X=1:k, FUN=s1)
  b_slice <- b_init + sum(sum)
  c_slice <- rgamma(1, shape=a_post, rate=b_slice)
  s_slice <- draw_s_post(grd=griddy_s, Nd=N_slice,
Vd=V_slice, lw=l_slice, cw=c_slice)
  l_slice <- draw_l_post(grd=griddy_l, Nd=N_slice,
 Vd=V_slice, sw=s_slice, cw=c_slice)

  N[iter,] <- N_slice
  V[,,iter] <- V_slice
  c_vec[iter] <- c_slice
  sig_vec[iter] <- s_slice
  lam_vec[iter] <- l_slice
}

hist(lam_vec[(burnin+1):end], breaks=20)
hist(sig_vec[(burnin+1):end], breaks=20)
plot(N[,1], main="N_1")
plot(N[,2], main="N_2")
plot(N[,3], main="N_3")

#Estimator of F_l given N and V results from Gibbs sampler
Gibbs_t <- function(t_){
  f_it <- function(ite, l_){
    if(true_N==TRUE){
prob <- post_dist(t_, rep(sstotal*2, r), V[,,ite], l_,
c_=c_vec[ite], spd=sig_vec[ite], lpd=lam_vec[ite])
```

```
 }else {prob <- post_dist(t_, N[ite,], V[,,ite], l_, c_=c_vec[ite],
  spd=sig_vec[ite], lpd=lam_vec[ite])}
    return(as.numeric(prob))
  }
  F_it <- Vectorize(f_it)
  #returns F_l posterior for N, V from
# particular Gibbs sampler iteration
  F_mat <- outer((burnin+1):end, 1:r, FUN=F_it)

  CI <- function(l_){return(quantile(F_mat[,l_],
 c(0.025, 0.975)))}
  #credible interval for F_l posterior mean

  CIs <- sapply(1:r, FUN=CI)

  df <- data.frame(t_, t(colMeans(F_mat)), t(c(CIs)))

  names_f <- sapply(1:r,
FUN=function(i){return(paste0("F", i))})
  names_ci <- sapply(1:r, FUN=function(i){
    return(append(paste0("F", i, "_lower"),
 paste0("F", i, "_upper")))})
  names_ci <- c(names_ci)
  names <- c("time", names_f, names_ci)
  df <- setNames(df, names)
  return(df)
}

true_N <- FALSE #Do we use the true N values?
```

```
#Otherwise the Gibbs sampler random values are used.

result <- ldply(s/meen, .fun = Gibbs_t)
result <- rbind(rep(0,7),result)
#data frame containing F_l mean and
#credible intervals for each t in t_grid

s_cen <- list()
for(ll in 1:r){s_cen[[ll]] <- s[round(s, 1)<=cen[ll]]}

#plots
pl1 <- ggplot(result, aes(x=time*meen, y=F1)) +
  geom_point()  +  geom_errorbar(aes(ymin=F1_lower,
ymax=F1_upper))+
  xlab("time")+ylab("F1 posterior")
p1<-pl1 + geom_point(data=data.frame(tr=cdf1),
aes(x=s,  y=tr, color="true CDF")) +
  geom_step(mapping=aes(x=c(0,s_cen[[1]]), y=ecd,
color="empirical CDF"),
     data=data.frame(ecd=c(0,
ecdf(r1t)(s)[1:length(s_cen[[1]])])))+
  geom_vline(xintercept=cen[1], linetype="longdash")
pdf(file="F130False.pdf", width=10, height=5)
p1
dev.off()

pl2 <- ggplot(result, aes(x=time*meen, y=F2)) +
  geom_point()  +  geom_errorbar(aes(ymin=F2_lower,
 ymax=F2_upper))+
```

```
  xlab("time")+ylab("F2 posterior")
p2<-pl2 + geom_point(data=data.frame(tr=cdf2),
 aes(x=s,  y=tr, color="true CDF")) +
  geom_step(mapping=aes(x=c(0,s_cen[[2]]), y=ecd,
color="empirical CDF"),
data=data.frame(ecd=c(0,ecdf(r2t)(s)[1:length(s_cen[[2]])])))+
  geom_vline(xintercept=cen[2], linetype="longdash")
pdf(file="F230False.pdf", width=10, height=5)
p2
dev.off()

pl3 <- ggplot(result, aes(x=time*meen, y=F3)) +
  geom_point()  +  geom_errorbar(aes(ymin=F3_lower,
ymax=F3_upper))+
  xlab("time")+ylab("F3 posterior")
p3<-pl3 + geom_point(data=data.frame(tr=cdf3),
 aes(x=s,  y=tr, color="true CDF"))+
  geom_step(mapping=aes(x=c(0,s_cen[[3]]), y=ecd,
color="empirical CDF"),
data=data.frame(ecd=c(0,ecdf(r3t)(s)[1:length(s_cen[[3]])])))+
  geom_vline(xintercept=cen[3], linetype="longdash")
pdf(file="F330False.pdf", width=10, height=5)
p3
dev.off()

#traceplots
pdf(file="gamma30False.pdf", width=5, height=4)
plot(lam_vec, ylab="gamma")
dev.off()
```

```
pdf(file="sigma30False.pdf", width=5, height=4)
plot(sig_vec, ylab="sigma")
dev.off()
pdf(file="c30False.pdf", width=5, height=4)
plot(c_vec, ylab="c")
dev.off()
pdf(file="N130False.pdf", width=5, height=4)
plot(N[,1], ylab="N_1")
dev.off()
pdf(file="N230False.pdf", width=5, height=4)
plot(N[,2], ylab="N_2")
dev.off()
pdf(file="N330False.pdf", width=5, height=4)
plot(N[,3], ylab="N_3")
dev.off()
```