UCLouvain

FACULTE DES SCIENCES

Faculté des sciences

Comparative analysis of a posteriori ratemaking in Motor Third-Party Liability Insurance in the Belgian market

Author: **Nathan Jacques**
Supervisor: **Michel Denuit**
Reader: **Cindy Courtois**
Academic year 2024–2025
Master [120] en sciences actuarielles

Comparative analysis of a posteriori ratemaking in Motor Third-Party Liability Insurance in the Belgian market

Nathan Jacques

Abstract

In a highly competitive Belgian market, comparing insurers' premium policies for MTPL Insurance is a key challenge. However, the variety of *a posteriori* ratemaking rules makes this assessment tricky. In this thesis, we develop a comprehensive framework to compare the optimality of the Bonus-Malus scales available on the market. Based on state-of-the-art optimal relativities theory, we apply this framework to some of the biggest players in Belgium. Finally, the numerical results are presented and discussed in order to establish a basis of comparison for the models encountered.

Key words : Insurance, MTPL insurance, Bonus-Malus Scale (BMS), Optimal relativities, Belgian MTPL insurance market

Acknowledgements

First, I would like to warmly thank Prof. M. Denuit, the academic supervisor of this master thesis. From his technical expertise and his ability to give me the right advice at the right time, to his availability, I am very grateful to have been given the opportunity to work under his supervision throughout the year. Anytime I was struggling, he had put me back to work with gentle words. Thank you.

Also, I wanted to thank Mrs. C. Courtois for the time dedicated to the attentive reading of this thesis. I hope you will enjoy reading it as much as I liked writing it.

Finally, I would like to thank all my relatives, family and friends. My dear colleagues, with whom I spent this last year working in addition to this thesis work, for helping me to sharpen my understanding of the topic by asking me simple but straightforward questions ; working with you has helped me gain significant experience in that short period of time. The monastic community of the Rochefort Trappist abbey, for the warm welcome during working periods, and especially to Fr. Gilbert for his kind attention and solicitude.

Contents

Introduction

The pricing process of MTPL insurance products can be decomposed in two parts:

1. *A priori* : a risk profile is established based on a limitative set of questions asked to the policyholder before signing the contract. An *a priori* tariff or *base premium* is then proposed in the contract.

2. *A posteriori* : as some important characteristics are unobservable or cannot be used for the pricing (e.g. gender), individuals with significantly different risk profiles can be assigned the same base premium. To balance this inadequacy, the claims experience over the life of the contract is used to adapt this base premium: we call it the *a posteriori* tariff.

Belgian law used to impose a fixed scheme, called a Bonus-Malus Scale (BMS), to all Belgian MTPL insurers until the early 2000's (a description of this scale can be found in [PDW03a]).
However, the Royal Decree of 16 January 2002 (applicable as of 01 January 2004) put an end to the historic Bonus-Malus scale used in the Belgian MTPL insurance market. Belgian insurers being now able to customize their own BMS, comparing their pricing policy became trickier.

The goal of this master thesis is to develop a clever strategy to model and analyze the BMS available on the Belgian market. Given the transition rules in the BMS, we will try to find the optimal *a posteriori* correction made to the base premium and compare it with the correction imposed contractually by the company. The first objective will be to formulate the problem rigorously and then, to suggest several approaches to address it.

This master thesis is composed of 3 chapters, described below:

Chapter 1 formally exposes the problem and defines the main mathematical tools that will be useful in the following. The Bonus-Malus Scales available on the Belgian market which will be analyzed are presented.

Chapter 2 details the framework for optimality of Bonus-Malus scales in steady-state regime.
Chapter 3 is focused on the adjustment of the previous framework in the context of transient regime.

Chapter 1

Problem statement

In this chapter, we shall first introduce the concept of a Bonus-Malus Scale (BMS) and present a set of BMS available on the Belgian market. Then, we will describe the *a priori* and *a posteriori* ratemaking frameworks used in this project. Finally, we will present the dataset used for our simulations.

1.1 Bonus-malus scales (BMS)

1.1.1 Introduction and definition

In a fairness objective, the cost of claims should be allocated equitably across the insured population. Indeed, more risky policyholders should pay a higher premium than the less risky ones. This typically involves grouping policyholders into risk classes, such that individuals within the same class are charged identical premiums. A common and effective approach to establishing this *a priori* classification is through the use of generalized linear models, such as Poisson regression for modeling claim frequencies (as discussed in section 1.2). Nevertheless, many relevant risk factors remain unobservable or unquantifiable when establishing this base premium, resulting in residual heterogeneity within each risk class.
To address this limitation, credibility theory has been a highly discussed topic, which allows to adjust premiums retrospectively based on individual claims history. This rating system penalizes policyholders who have caused one or more claims through maluses (premium surcharges), while rewarding claim-free insureds with bonuses (premium discounts). In addition to incentivizing prudent behavior and mitigating moral hazard, rating systems aim to refine the risk assessment by incorporating individual loss experience.
While credibility theory offers a theoretically sound framework for such retrospective adjustments, its practical implementation is often hindered by mathematical and

communicative complexity. In particular, the opacity of credibility formulas can present challenges in the context of customer relations, where policyholders may be reluctant to accept pricing mechanisms they do not fully understand.
As a pragmatic alternative, insurers have introduced Bonus-Malus Scales (BMS), which may be interpreted as commercially adapted implementations of credibility principles. These scales offer a simplified, rule-based approach that remains consistent with actuarial reasoning, yet is more accessible and transparent to the average policyholder, who can easily anticipate future premiums based on past claims history.
Belgium market makes no exception to these observations. As exposed in the introduction, a BMS was legally imposed to all insurance companies until the early 2000's. Nowadays, most of Belgian insurers kept a similar approach as the old Belgian standard (with more or less significant adjustments). We will first give a broad definition of a BMS, then present the BMS we will analyze through this project.

Definition 1.1.1 (Bonus-malus scale (BMS))**.** A BMS is a form of a posteriori rating (or experience rating) mainly used in motor insurance. It consists of three main building blocks:

- Levels : number of steps in the scale
- Transition rules : predefined rules to move up or down the scale over time
- Relativities : percentage applied to the base premium to get the commercial premium actually paid by the policyholder

1.1.2 BMS in the Belgian market

We decided to look at the biggest players in the Belgian market, for MTPL insurance (in terms of market shares). A 2024 Assuralia study [24] as ranked, based on 2023 figures:

1. AXA Belgium (market share of 19.04%)
2. AG Insurance (market share of 16.53%)
3. Baloise Belgium (market share of 12.83%)
4. Ethias (market share of 10.75%)
5. KBC Assurances (market share of 10.23%)
6. P&V Assurances (market share of 8.34%)

Note : All information about the parameters of the BMS in this section come from the terms and conditions of the MTPL insurance policies of each insurance company.

AXA Belgium has implemented a custom Bonus-Malus system, which consists in a combination of two BMS (see terms and conditions of their "Confort Auto" insurance policy [Bel24]). The applied relativity is then the product of the relativities from both the first and the second one. The first BMS is based on the number of years without a liable claim, while the second BMS is based on the number of liable claims within 5 years. As this architecture introduces more complexity in our developments, we decided to disregard AXA Belgium from this study. We shall now consider the following 5 biggest insurers in the Belgian market (at 12/2023).

1.1.2.1 AG Insurance (AG)

AG sells a MTPL insurance with a BMS called "Turbo Bonus" (see [Ins23]). On Figure 1.1, we see that it consists of 25 levels (from -2 to 22). The entry level is level 14, for which the relativity is fixed at 100% (i.e. base premium). An adapted entry level (at level 11) for restricted use of the vehicle is also foreseen in terms and conditions, but we shall ignore it for simplicity reasons. The policyholder moves unconditionally down the scale by 1 level each year. If one or more claim is reported, he moves 5 levels per claim up the scale (i.e. if one claim: 4 levels, with the 1-level down rule; if 2 claims: 9 levels up; etc).

Degrés	Niveau de primes par rapport au niveau de base 100
22	200
21	160
20	140
19	130
18	123
17	117
16	111
15	105
14	100
13	95
12	90
11	85
10	81
9	77
8	73
7	69
6	66
5	63
4	54
3	54
2	54
1	54
0	54
-1	54
-2	54

Figure 1.1: AG Turbo Bonus - Levels and relativities (Source : AG Insurance)

A specific rule introduced by AG (as well as other companies), is that when the bottom-level is reached (level -2), the policyholder will not go up the scale anymore, even if he reports a claim. Level -2 is considered as "frozen" since we cannot escape it once reached. The reader could wonder if this is a reasonable strategy: indeed, some policyholders at level -2 can suddenly start to exhibit very risky behavior and report a lot of claims while still being rewarded the lowest premium available. However, in accordance with Article 86 of 4/04/2014 Belgian law on insurance, insurers can terminate the contract when a claim is reported. This strategy of a "frozen" state is then acceptable from a risk management perspective, since a problematic contract in the portfolio can be removed.

1.1.2.2 Baloise Belgium (Baloise)

The BMS proposed by Baloise is very similar to the BMS of AG (see [Bel]). The entry level is determined based on the number of years since the policyholder holds a driving license and the number of claims in the last 5 years. When no "attestation de sinistralite" is provided to the company, the entry level is fixed at level 11. We will keep this assumption for simplicity reasons. The transition rule (-1/+5 per claim) and the frozen level -2 are the same as for AG.

Degrés	Pourcentage
22	200
21	160
20	140
19	130
18	123
17	117
16	111
15	105
14	100
13	95
12	90
11	85
10	81
9	77
8	73
7	69
6	66
5	63
4	60
3	57
2	54
1	54
0	54
-1	54
-2	54

Figure 1.2: Baloise BMS - Levels and relativities (Source : Baloise Belgium)

On top of that, Baloise decided to keep the special bonus rule in place in the former compulsory Belgian BMS. The rule is simple: if the policyholder is above level 14 with no claims reported in the last 4 years, he goes down to level 14 whatsoever.

This particularity makes the BMS non-markovian since we should keep track of the memory of reported claims to determine the level transition. This issue has been tackled in [PDW03a] and we will keep the exact same approach here. The remedy basically consists in creating additional fictive levels in the BMS to keep track of the reported claims in levels above 14 (see [PDW03a] for further details). Baloise also offers an optional clause, called "BM Safe", which consists in ignoring in the BMS the first claim reported each year. As this is an extension of the contract (which increases the premium), we don't consider it in our study.

1.1.2.3 Ethias

The BMS of Ethias is one of a kind (see [Eth]), its structure can be visualized in Figure 1.3. It starts with a usual scale from level 0 to level 22, with an entry level at level 13 (relativity of 100%). The transition rule is then as the other scales presented before (-1/+5 per claim). The special bonus rule described above is also included (i.e. back to level 14 if no claims in the last 4 years in higher levels).

a)	
Échelle bonus malus	
Niveau de prime par rapport à la prime commerciale de base exprimée à 100 %	Degrés
45	0
47	1
49	2
54	3
60	4
66	5
69	6
72	7
77	8
81	9
85	10
89	11
95	12
100	13
105	14
116	15
122	16
128	17
135	18
143	19
153	20
175	21
219	22

b)	
Nombre d'années au degré bonus malus 0	Réduction de prime
Entrée	
Après 1 année	0 %
Après 2 années	
Après 3 années	
Après 4 années	
Après 5 années	5 %
Après 6 années	
Après 7 années	
Après 8 années	
Après 9 années	10 %
Après 10 années	

c)
5 années consécutives sans sinistre en tort au degré bonus malus 0 vous donnent droit à un joker

Figure 1.3: Ethias BMS - Levels and relativities (Source : Ethias)

There is no frozen state as for AG and Baloise. However, the lowest level (level 0) offers an additional discount on the premium when the policyholder keeps level 0

for at least 3 years. A larger discount is granted if level 0 is maintained for at least 8 years. This additional discount (and the number of years maintained at level 0) are kept even if a claim is reported. However, if the policyholder goes above level 4 in the scale, the discount (and years memory) is lost. This architecture can be modeled using fictive states, as for the special bonus rule explained above.
On top of that, a joker is granted to the policyholder after 5 years maintained at level 0. It means that the next claim occurring will have no effect on the levels and the policyholder will stay at level 0. This feature can also be modeled using fictive states.
Unfortunately, modeling of the two last clauses together through fictive states is possible but will increase significantly the number of states in the BMS. As the additional discount after several years at level 0 doesn't affect directly the transition in the BMS, we decided to focus on the joker rule and discard the additional discount. This will be modeled by splitting level 0 in 6 different fictive states $0.i$ with $i = 0 \dots 5$ (this modeling technique is detailed in section 2.2.1.1).
The model used for Ethias can be seen in Table 1.1. Levels 1 to 22 are the same as for the special bonus rule detailed in [PDW03a]. As mentioned above, level 0 has been split in 6 fictive states to keep track of the number of years at level 0. The fictive state 0.5 corresponds to the joker-owner state. If a claim is reported at this stage, policyholder keeps level 0 but has no joker left (i.e. back to fictive state 0.0).

State	State after k accidents					
k	0	1	2	3	4	5
22	21.1	22	22	22	22	22
21.0	20.1	22	22	22	22	22
21.1	20.2	22	22	22	22	22
20.0	19.1	22	22	22	22	22
20.1	19.2	22	22	22	22	22
20.2	19.3	22	22	22	22	22
19.0	18.1	22	22	22	22	22
19.1	18.2	22	22	22	22	22
19.2	18.3	22	22	22	22	22
19.3	14	22	22	22	22	22
18.0	17	22	22	22	22	22
18.1	17.2	22	22	22	22	22
18.2	17.3	22	22	22	22	22
18.3	14	22	22	22	22	22
17	16	21.0	22	22	22	22
17.2	16.3	21.0	22	22	22	22
17.3	14	21.0	22	22	22	22
16	15	20.0	22	22	22	22
16.3	14	20.0	22	22	22	22
15	14	19.0	22	22	22	22
14	13	18.0	22	22	22	22
13	12	17	22	22	22	22
12	11	16	21.0	22	22	22
11	10	15	20.0	22	22	22
10	9	14	19.0	22	22	22
9	8	13	18.0	22	22	22
8	7	12	17	22	22	22
7	6	11	16	21.0	22	22
6	5	10	15	20.0	22	22
5	4	9	14	19.0	22	22
4	3	8	13	18.0	22	22
3	2	7	12	17	22	22
2	1	6	11	16	21.0	22
1	0.0	5	10	15	20.0	22
0.0	0.1	4	9	14	19.0	22
0.1	0.2	4	9	14	19.0	22
0.2	0.3	4	9	14	19.0	22
0.3	0.4	4	9	14	19.0	22
0.4	0.5	4	9	14	19.0	22
0.5	0.5	0.0	4	9	14	19.0

Table 1.1: Transition rules of Ethias BMS - excluding extra discount at level 0

1.1.2.4 KBC Assurances (KBC)

KBC has the most unique BMS in our selection (see [Assa]). Indeed, while most of Belgian insurers kept the former Belgian BMS (with minor or major adjustments), KBC decided to take a completely different approach. The *a posteriori* ratemaking has two components here : a discount on premium and a joker. After at least one year without liable claim, the policyholder gets a discount on premium[1]. If the policyholder does not report any liable claims for at least 5 consecutive years, he's granted a joker. Three situations are possible to transition from one state to another :

- **Joker** (State 0.4) - Policyholder has a joker : The premium discount is not lost but he will have to be claim-free for 5 new consecutive years to get the joker back.
- **Discount** (States 0.x) - Policyholder has a premium discount (but no joker) : The premium discount is lost and can only be earned again if no liable claim is reported in the following year.
- **No joker nor discount** (State 1) - There is no premium increase. However, a deductible of 250 euros will be applied to the claim and policyholder will have to wait for another year to get his premium discount back.

State	State after k accidents		
k	0	1	2
1	0.0	1	1
0.0	0.1	1	1
0.1	0.2	1	1
0.2	0.3	1	1
0.3	0.4	1	1
0.4	0.4	0.0	1

Table 1.2: Transition rules of KBC BMS

The main difference with the other BMS in our study is the introduction of a deductible. This could be treated as a relativity by solving an actuarial indifference principle (see [Den+07], section 7.3.2 for further detail). For a per claim deductible (as introduced in KBC BMS), this indifference principle states that, for a given level l, the level of deductible d_l should satisfy

$$\lambda r_l \mathbb{E}[C_1] = \lambda \mathbb{E}[C_1] + \lambda r_l \left(\mathbb{E}[C_1 | C_1 < d_l] \Pr[C_1 < d_l] + d_l \Pr[C_1 > d_l]\right)$$

where r_l is the relativity attached to level l, λ is the *a priori* expected claims frequency and C_1 is the amount of the claim reported. As, on one hand, this approach would require a model for the claims severity (which is not the focus of this study), and, on the other hand, KBC does not disclose the discount applied on premium (which would make it difficult to compare to our results), we decided to discard KBC Assurances from this study.

[1]KBC terms and conditions do not specify the level of discount applied

1.1.2.5 P&V Assurances (P&V)

P&V has the most simple BMS structure (see [Assb]). Indeed, this BMS has same levels and relativities as Baloise (see Figure 1.4). The entry level is level 11 (can be different based on specific conditions, but we will discard that in this study for simplicity).

Degrés	Niveau de prime par rapport au niveau de base 100
22	200
21	160
20	140
19	130
18	123
17	117
16	111
15	105
14	100
13	95
12	90
11	85
10	81
9	77
8	73
7	69
6	66
5	63
4	60
3	57
2	54
1	54
0	54
-1	54
-2	54

Figure 1.4: P&V BMS - Levels and relativities (Source : P&V Assurances)

The transition rule is also -1/+5 per claim. There is no other special rule (no joker, frozen state, etc).

The implementation in R of the different BMS characteristics (levels, transition rules and relativities) can be found in Appendix A.

Following all those observations, we will now detail the mathematical framework for the rest of this study and then analyze the behavior of the BMS selected above, that is : AG Insurance (AG), Baloise Belgium (Baloise), Ethias and P&V Assurances (P&V).

1.2 A priori ratemaking

When an individual asks an insurer for a new MTPL contract, the insurer will ask a limited number of questions in order to fix an *a priori* tariff. Let us denote $\mathbf{X}_i = (X_{i1}, \ldots, X_{iq})$ the vector of *a priori* characteristics of a policyholder i. The goal is then to find an estimation of the *a priori* claim frequency λ_i based on an observation of $\mathbf{X}_i$ for policyholder i. Each policyholder is then categorized in a specific risk class where all other insured present similar λ's. The base premium will then be attributed accordingly. In a Poisson setting (which is common use), the number Y_i of reported claims has

$$Y_i \sim \text{Poisson}\left(d_i \exp(\beta_0 + \sum_{m=1}^{q} \beta_m x_{im})\right)$$

for an exposure length d_i. After estimation of the regression parameters (e.g. using GLMs or other regression techniques), the *a priori* claim frequency λ_i can be estimated as

$$\hat{\lambda}_i = d_i \exp\left(\hat{\beta}_0 + \sum_{m=1}^{q} \hat{\beta}_m x_{im}\right)$$

However, some heterogeneity still remains in each risk class due to un-modeled/not observable variables (e.g. reflex swiftness, drugs consumption, etc). This residual heterogeneity is captured in the *a posteriori* ratemaking.

1.3 A posteriori ratemaking

In this study, we will focus on one specific form of *a posteriori* ratemaking, or experience rating : the BMS (cf. definition 1.1.1).
As previously explained, the goal is to capture the remaining heterogeneity in each risk class. Let us note Θ_i the random effect modeling the residual heterogeneity of policyholder i. Also, we assume that given $\Theta_i = \theta$, Y_i are independent Poisson variables with mean $\lambda_i\theta$, that is

$$\Pr\left[Y_i = k|\Theta_i = \theta\right] = \exp(-\lambda_i\theta)\frac{(\lambda_i\theta)^k}{k!}, \quad k = 0, 1, 2, \ldots \tag{1.1}$$

We will consider two different models for the distribution of Θ_i : a continuous model (Gamma distributed) and a discrete model (Good-driver/Bad-driver model).

Continuous model - Gamma

Let us also assume that the Θ_i's are iid Gamma distributed, that is

$$f(\theta) = \frac{1}{\Gamma(a)} a^a \theta^{a-1} \exp(-a\theta), \quad \theta \in \mathbb{R}^+ \tag{1.2}$$

We then have $\mathbb{E}[\Theta_i] = 1$ and it follows that $\mathbb{E}[Y_i] = \mathbb{E}[\mathbb{E}[Y_i|\Theta_i]] = \lambda_i$
To fit the Gamma distribution $f(\theta)$ defined in Equation (1.2), we proceed via maximum likelihood estimation by exploiting the marginal distribution of the number of claims Y_i. Conditionally on the residual heterogeneity factor $\Theta_i = \theta$, the claim count follows a Poisson distribution (as stated in (1.1)):

$$Y_i \mid \Theta_i = \theta \sim \text{Poisson}(\lambda_i \theta).$$

Assuming $\Theta_i \sim \text{Gamma}(a, a)$, the marginal distribution of Y_i is obtained by integrating the unobserved heterogeneity:

$$\begin{align} \Pr[Y_i = k] &= \int_0^\infty \Pr\left[Y_i = k | \Theta_i = \theta\right] \cdot f(\theta) d\theta \tag{1.3} \\ &= \int_0^\infty \frac{e^{-\lambda_i \theta} (\lambda_i \theta)^k}{k!} \cdot \frac{a^a}{\Gamma(a)} \theta^{a-1} e^{-a\theta} \, d\theta. \tag{1.4} \end{align}$$

This integral corresponds to the mixture of a Poisson distribution with a Gamma distribution, which is known to give a Negative Binomial distribution. Indeed, if we develop equation (1.4), we get

$$\Pr[Y_i = k] = \frac{a^a \lambda_i^k}{k! \Gamma(a)} \int_0^\infty \theta^{k+a-1} e^{-(\lambda_i + a)\theta} \tag{1.5}$$

The Gamma function is defined as

$$\Gamma(z) := \int_0^{+\infty} t^{z-1} e^{-t} dt \quad ; \quad \forall z \in \mathbb{C} \quad s.t. \quad \Re(z) > 0 \tag{1.6}$$

By putting equation (1.6) in equation (1.5), we get

$$\Pr[Y_i = k] = \frac{a^a \lambda_i^k}{k! \Gamma(a)} \cdot \frac{\Gamma(k + a)}{(\lambda + a)^{k+a}} \tag{1.7}$$

which is the pmf of a Negative Binomial variable.
So, we have, for our case:

$$Y_i \sim \text{NegBin}\left(r = a, \, p = \frac{a}{a + \lambda_i}\right),$$

where $r = a$ is the number of failures (or the shape parameter), and p is the success probability. The corresponding mean and variance of the marginal distribution are:

$$\mathbb{E}[Y_i] = \lambda_i, \quad \text{Var}[Y_i] = \lambda_i + \frac{\lambda_i^2}{a}.$$

This formulation reveals that the variance exceeds the mean by an additive term proportional to λ_i^2, which captures the overdispersion introduced by unobserved heterogeneity. To estimate the parameter a, we fit a Negative Binomial GLM to the observed number of claims Y_i, using the same covariates and log-exposure offset as in the Poisson model. The GLM estimates the dispersion parameter $\phi = 1/a$, from which the shape parameter is recovered as $a = 1/\phi$. The estimated value of a fully specifies the density $f(\theta)$ (which is then used in subsequent calculations involving the stationary distribution $\pi(\lambda\theta)$, see section 2.1) .

Discrete model - Good-driver/Bad-driver

For illustration purposes, we also consider a simple model where Θ_i's take discrete values

$$\Theta_i = \begin{cases} \theta_1 \text{ if policyholder } i \text{ is a good driver} \\ \theta_2 \text{ if policyholder } i \text{ is a bad driver} \end{cases} . \tag{1.8}$$

It follows that $\theta_1 \leq \theta_2$, to obtain a smaller actual claims frequency $\lambda\theta$ for a good driver. Denoting g_1 the proportion of good drivers in the portfolio and $g_2 = 1 - g_1$ the proportion of bad drivers, (Let us assume that 60% of the portfolio are good drivers and 40% are bad drivers), then the pmf $g(\theta)$ of Θ_i is

$$g(\theta) = \begin{cases} g(\theta_1) = g_1 = 60\% \\ g(\theta_2) = g_2 = 40\% \end{cases} . \tag{1.9}$$

Moreover, we want to maintain the equilibrium $\mathbb{E}[\Theta_i] = 1$. A particular choice of (θ_1, θ_2) can be $(\theta_1, \theta_2) = (0.8, 1.3)$, thus satisfying

$$\mathbb{E}[\Theta_i] = g(\theta_1)\theta_1 + g(\theta_2)\theta_2 = 0.6 \times 0.8 + 0.4 \times 1.2 = 1$$

From now on, let's suppose we pick a policyholder at random from the insurance portfolio and we then drop the subscript i. The actual expected claims frequency of the policyholder can be expressed as $\Lambda\Theta$, where Λ is the unknown *a priori* expected claims frequency and Θ is the unknown residual heterogeneity effect. We will consider Λ and Θ independent, which is reasonable since Θ captures the residual heterogeneity effect that was not captured *a priori* by Λ in the risk classes.

1.4 Dataset and frequency model

The dataset used in our simulations comes from the R package `CASdatasets`. The dataset `beMTPL97` is chosen since it has the usual observable characteristics asked at the conclusion of a MTPL insurance contract. The dataset documentation states :

> The portfolio contains 163,212 unique policyholders, each observed during a period of exposure- to-risk expressed as the fraction of the year during which the policyholder was exposed to the risk of filing a claim. Claim information is known in the form of the number of claims filed and the total amount claimed (in euro) by a policyholder during the period of exposure.

Let us also assume that the portfolio is composed of h risk classes, where w_g is the proportion of policyholders in the g-th class where expected *a priori* claims frequency is λ_g, that is $\Pr[\Lambda = \lambda_g] = w_g$, for $g = 1, \ldots, h$.
To fit our claims frequency model (described in section 1.2), we will fit a GLM on a part of this dataset. The explanatory variables chosen are: the age of the policyholder, the age of the car and the power of the car, which are commonly asked when people ask for a MTPL insurance. As we make us of a GLM, we categorize those variables and hereby establish the *a priori* risk classes. Each variable is split in 3 categories, resulting in 27 risk classes:

- Age of policyholder : we consider "young" (age $<$ 30 years), "middle-aged" (30 years $<$ age $<$ 50 years) and "old" people (age $>$ 50 years)
- Age of the vehicle : we consider "new", "average" and "old" vehicles. The split is based on 33.33-percentiles, such that each category as a similar size
- Power of the vehicle : we consider "low", "regular" and "high" power vehicles. The split is also done quantile-wise.

We specify that other *a priori* ratemaking methods are of course suitable[2]. As this is not the focus of this study, we decided to choose a GLM as it is easy to implement and to interpret. It is also more convenient from a business point of view since the policyholder knows why his premium varies (e.g. higher premium because he is young, ...).

[2] To classify the policyholders into *a priori* risk classes with a GBM, a RF or a NN, we would partition the portfolio into h classes based on the estimated claim frequencies $\hat{\lambda}_i$, as determined by method used. The classification would then be performed by dividing the empirical distribution of $\hat{\lambda}_i$ into h quantile-based intervals, thereby assigning each policyholder to one of h disjoint risk classes. Moreover, class boundaries would be defined using the empirical quantiles of $\hat{\lambda}_i$ at probabilities $\{0, \frac{1}{h}, \frac{2}{h}, \ldots, 1\}$, and label the resulting intervals as classes $g = 1, \ldots, h$. Each class g is then assigned an *a priori* risk parameter λ_g, defined as the empirical mean of $\hat{\lambda}_i$ over all policyholders belonging to class g. That is,

$$\lambda_g = \frac{1}{n_g} \sum_{i \in \mathcal{C}_g} \hat{\lambda}_i,$$

where $\mathcal{C}_g$ denotes the set of policyholders assigned to class g, and $n_g = |\mathcal{C}_g|$.

This classification allows a discrete representation of the *a priori* risks, which is used in subsequent modeling of transitions across Bonus-Malus levels and in computing distribution of policyholders among levels (see section 2.2.1 and following).
The above development is implemented in R (see code in Appendix B) and the claims frequency for each risk class obtained is shown in Figure 1.5.

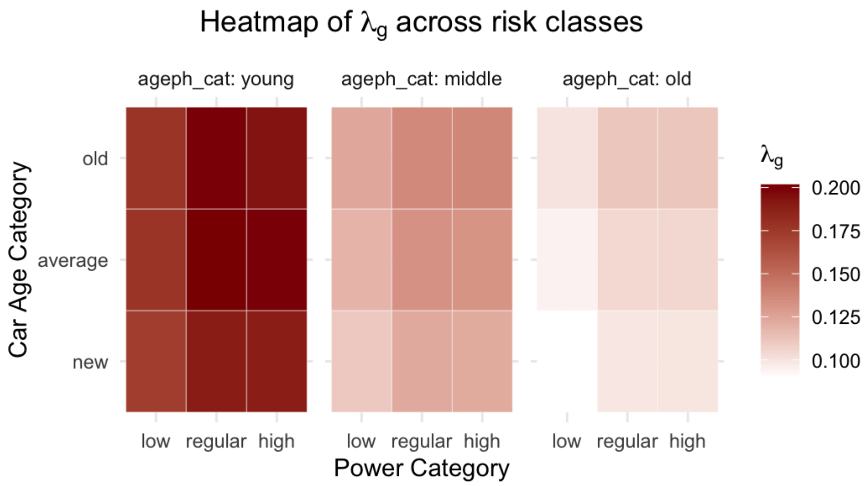


Figure 1.5: Visualization of λ_g across risk classes

In Figure 1.5, we can clearly see that the *a priori* claims frequency is decreasing with age (groups of boxes are lighter as we move to the right), which was expected. We also note that the more powerful the vehicle insured is, the higher the λ_g, which makes also sense. The impact of car age is less obvious to spot graphically since it's less significant. When looking at the results, new cars always exhibit the lowest *a priori* claims frequencies and average cars have the highest, while old cars fall in-between.

We once again insist that this *a priori* ratemaking is basic and is surely not as refined as the models found in market practice. However, those results will be very useful for latter development. We will now focus on the residual heterogeneity effect (captured by Θ), where the BMS will play its role.

Chapter 2

Stationary Regime

2.1 Stationary distribution

For the sake of later development, let's introduce the stationary distribution describing the stationary probability of a policyholder with expected claims frequency $\lambda\theta$ to belong to level l. Let $\mathbf{P}(\lambda\theta) = \{p_{l_1 l_2}(\lambda\theta)\}$ be the one-step transition matrix where $p_{l_1 l_2}(\lambda\theta)$ is the probability for a policyholder with expected claims frequency $\lambda\theta$ to move from level l_1 to level l_2, where $l_1, l_2 = 1, \ldots, s$. Then the stationary distribution $\pi(\lambda\theta) = (\pi_1(\lambda\theta), \ldots, \pi_s(\lambda\theta))$ is the solution of

$$\begin{cases} \pi(\lambda\theta) = \pi(\lambda\theta) \cdot \mathbf{P}(\lambda\theta) \\ \pi(\lambda\theta) \cdot \mathbf{1} = 1 \end{cases} . \tag{2.1}$$

where $\mathbf{1}$ is the column vector of 1's. We can then write

$$\Pr[L_{\lambda\theta} = l] = \pi_l(\lambda\theta) \tag{2.2}$$

where $L_{\lambda\theta}$ is the level occupied for a policyholder with expected claims frequency of $\lambda\theta$ once the steady state has been reached.
An important remark comes here into place : what if the steady state is never reached or after too many years? Some of the BMS described in chapter 1 may have a very slow convergence to this steady state[1]. The stationarity of the BMS is then a strong assumption we make here for the sake of future developments. It will be a stumbling block of our study and the convergence to this steady state is more thoroughly assessed in section 3.1.

[1]We shall see that the "frozen" state in some of the BMS will also introduce another challenge, since that, in the long run, every policyholder can reach this frozen state, creating a degenerate distribution at this state.

2.2 Optimality of BMS in stationary regime

Following the definition of the BMS, we easily identify 3 levels of action to achieve optimality of the ratemaking (optimality criterion to be defined) : the number of levels in the scale, the transition rules and the relativities. We will focus on the latter here, and briefly explain how transition rules could be optimally designed.

2.2.1 Optimal relativities

An educated guess would be to minimize the distance between the true premium $\Lambda\Theta$ and actual premium Λr_L of a policyholder, where r_L is the relative premium after steady state has been reached at level L. A particular choice of distance would be the expected squared difference. We would then like to find, given $L = l$, the optimal r_L such that

$$r_L = \operatorname{argmin} \mathbb{E}[(\Lambda\Theta - \Lambda r_L)^2] \tag{2.3}$$

This criterion has been suggested by [Tan+15]. Prior works rather only considered the *a posteriori* part of the premium, which translates in objective function

$$\mathbb{E}[(\Theta - r_L)^2]$$

However, this does not take into account the *a priori* expected claim frequency and consequently, policyholders with higher λ_g's are over-penalized when climbing up the BMS compared to less *a priori* risk profiles. This unfairness has been described in [Tan+15], refering also to the work of Pitrebois et al. in [PDW03b]. The optimality criterion defined in (2.3) remedies this unwanted behavior.
On top of that, the insurer has to maintain financial equilibrium, that is make sure that bonuses granted to good drivers balance the maluses granted to bad drivers. Mathematically, we get

$$\mathbb{E}[r_L] = 1 \tag{2.4}$$

Adding constraint (2.4) to problem (2.3), we get to solve this problem:

$$r_L = \operatorname{argmin} \mathbb{E}[(\Lambda\Theta - \Lambda r_L)^2] \quad s.t. \quad \mathbb{E}[r_L] = 1 \tag{2.5}$$

The Lagrangian of problem (2.5) is then

$$\begin{aligned}
\mathcal{L}(\mathbf{r}, \alpha) &= \mathbb{E}[(\Lambda\Theta - \Lambda r_L)^2] + \alpha(\mathbb{E}[r_L] - 1) && (2.6)\\
&= \sum_{l=1}^{s} \mathbb{E}[(\Lambda\Theta - \Lambda r_L)^2 | L = l] \Pr[L = l] + \alpha\left(\sum_{l=1}^{s} r_l \Pr[L = l] - 1\right) && (2.7)
\end{aligned}$$

with $\mathbf{r} = (r_1, \ldots, r_l)$. Let us develop the first-order conditions

$$\frac{\partial}{\partial \alpha}\mathcal{L}(\mathbf{r}, \alpha) = 1 - \sum_{l=1}^{s} r_l \Pr[L = l] = 0 \tag{2.8}$$

and

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{r}}\mathcal{L}(\mathbf{r}, \alpha) &= \frac{\partial}{\partial \mathbf{r}}\left(\sum_{l=1}^{s} \mathbb{E}[\Lambda^2\Theta^2 - 2\Lambda^2\Theta r_L + \Lambda^2 r_L^2 | L = l] \Pr[L = l]\right) + \alpha \Pr[L = l] \\
&= \mathbb{E}[-2\Lambda^2\Theta + 2\Lambda^2 r_L | L = l] \Pr[L = l] + \alpha \Pr[L = l] \\
&= \Pr[L = l]\left(2\mathbb{E}[\Lambda^2 r_L - \Lambda^2\Theta | L = l] + \alpha\right) = 0 \\
&\iff \Pr[L = l]\left(2\mathbb{E}[\Lambda^2\Theta - \Lambda^2 r_L | L = l] - \alpha\right) = 0
\end{aligned} \tag{2.9}$$

for $l = 1, \ldots, s$. Putting it together, we get for first-order conditions

$$\begin{cases} 1 - \sum_{l=1}^{s} r_l \Pr[L = l] = 0 \\ \Pr[L = l]\left(2\mathbb{E}[\Lambda^2\Theta - \Lambda^2 r_L | L = l] - \alpha\right) = 0 \quad \text{, for all } l = 1, \ldots, s \end{cases} . \tag{2.10}$$

Solving (2.10) for (α, r_l), we get

$$\alpha = \frac{\left(\sum_{l=1}^{s} \frac{\mathbb{E}[\Lambda^2\Theta | L = l] \Pr[L = l]}{\mathbb{E}[\Lambda^2 | L = l]}\right) - 1}{\sum_{l=1}^{s} \frac{\Pr[L = l]}{2\mathbb{E}[\Lambda^2 | L = l]}} \tag{2.11}$$

and

$$r_l = \frac{\mathbb{E}[\Lambda^2\Theta | L = l]}{\mathbb{E}[\Lambda^2 | L = l]} - \frac{\alpha}{2\mathbb{E}[\Lambda^2 | L = l]} \tag{2.12}$$

The different expressions found above can be derived as follows.

$$\Pr[L = l] = \sum_{g=1}^{h} \Pr[\Lambda = \lambda_g] \Pr[L = l | \Lambda = \lambda_g] \tag{2.13}$$

$$\begin{aligned}
&= \sum_{g=1}^{h} \Pr[\Lambda = \lambda_g] \int_0^{+\infty} \Pr[L = l | \Lambda = \lambda_g, \Theta = \theta] f(\theta) d\theta \\
&= \sum_{g=1}^{h} w_g \int_0^{+\infty} \pi_l(\lambda_g \theta) f(\theta) d\theta
\end{aligned} \tag{2.14}$$

$$\begin{aligned}
\mathbb{E}[\Lambda^2\Theta | L = l] &= \frac{\mathbb{E}[\Lambda^2\Theta \cdot \mathbb{I}_{L=l}]}{\Pr[L = l]} \quad (\mathbb{I} \text{ is the indicator function}) \\
&= \frac{\sum_{g=1}^{h} \Pr[\Lambda = \lambda_g] \int_0^{+\infty} \lambda_g^2 \theta \Pr[L = l | \Lambda = \lambda_g, \Theta = \theta] f(\theta) d\theta}{\Pr[L = l]} \\
&= \frac{\sum_{g=1}^{h} w_g \int_0^{+\infty} \lambda_g^2 \theta \pi_l(\lambda_g \theta) f(\theta) d\theta}{\sum_{g=1}^{h} w_g \int_0^{+\infty} \pi_l(\lambda_g \theta) f(\theta) d\theta}
\end{aligned} \tag{2.15}$$

and in a similar fashion

$$\mathbb{E}[\Lambda^2|L=l] = \frac{\sum_{g=1}^{h} w_g \int_0^{+\infty} \lambda_g^2 \pi_l(\lambda_g\theta) f(\theta) d\theta}{\sum_{g=1}^{h} w_g \int_0^{+\infty} \pi_l(\lambda_g\theta) f(\theta) d\theta} \tag{2.16}$$

2.2.1.1 Special bonus rule - fictive states

In [PDW03a], Pitrebois et al. fitted the optimal relativities for the former compulsory Belgian BMS. The main issue encountered was the non-markovian property of such a BMS. Indeed, the special bonus rule imposing a comeback to the entry level after 4 years without claims violates the memoryless property. In order to keep in memory the number of years without claims, fictive states are used. As several BMS available on the Belgian market nowadays have kept this special rule, we will use the same approach as [PDW03a]. However, [PDW03a] was not considering the *a priori* expected claims frequency in its objective function, which we will do, in accordance with the previous section.

For levels above the entry level, we will add states $i.j$, that is level i with j years without claims. Let n_j be the number of sub-states for each level j. We get the same result

$$r_l = \frac{\mathbb{E}[\Lambda^2\Theta|L=l]}{\mathbb{E}[\Lambda^2|L=l]} - \frac{\alpha}{2\mathbb{E}[\Lambda^2|L=l]} \tag{2.17}$$

but now

$$\Pr[L=j.i] = \sum_g w_g \int_0^{\infty} \pi_{j.i}(\lambda_g\theta) f(\theta) d\theta \tag{2.18}$$

By definition of the sub-states $j.i$, we then get

$$\begin{aligned}\Pr[L=j] &= \sum_{i=1}^{n_j} \Pr[L=j.i] \\ &= \sum_g w_g \int_0^{\infty} \pi_j(\lambda_g\theta) f(\theta) d\theta\end{aligned}$$

with $\pi_j = \sum_{i=1}^{n_j} \pi_{j.i}$.

In other words, the stationary distribution is computed on the whole space of states and sub-states and is then aggregated by the original levels in the BMS. The following is exactly the same as the case without fictive levels. More details can be found in [PDW03a].

The drawback of this approach is that it increases the number of states in the BMS, which would cause an increase of computational workload. This issue is addressed in the section 2.2.1.2.

2.2.1.2 Continuous model

The theoretical model is directly applicable by taking the fitted $f(\theta)$, as detailed in section 1.3. The derivation of the stationary distribution π_l has been tackled in section 2.1. The only possible issue at this step would be the numerical integration. Indeed, in this model, the stationary distribution of the BMS levels is computed by integrating over the unobserved heterogeneity parameter θ, which follows a Gamma distribution.

In R, the easiest way to compute this integral is using `integrate()` over a function that includes a matrix exponentiation of the transition matrix $P(\lambda\theta)$. This is fine for low-scale problems (see Appendix C, for implementation), but when the number of levels increases (i.e. especially when adding fictive states), the complexity of the computation increases drastically. This resulted in extremely long runtimes, especially when λ is large (slower Poisson convergence). Moreover, a high value of `max_k` is required in the BMS transition matrix, and on top of that, `integrate()` evaluates the integrand many times.

To reduce the computational cost, we replace the integral with a weighted sum over a discretized grid of θ values. That is, for a function $A(\theta)$,

$$\int_0^\infty A(\theta) f(\theta)\, d\theta \approx \sum_{i=1}^{N} A(\theta_i) \cdot u_i$$

where:

- $\{\theta_i\}_{i=1}^N$ is a fine grid of values from $[0, \theta_{\max}]$,
- $u_i = f(\theta_i) \cdot \Delta\theta$,
- $\theta_{\max}$ is chosen such that $\Pr(\Theta > \theta_{\max}) \ll \varepsilon$.

This approach is a form of integration quadrature with equispaced nodes and density-weighted summation.

Choice of Grid We use:

$$\theta_i = (i-1) \cdot \Delta\theta, \quad \text{for } i = 1, \ldots, N$$

with $\Delta\theta = 0.02$ and $N = 750$ (so $\theta_{\max} = 15$). This range covers essentially the full support of Gamma(a, a) for moderate a. In Figure 2.1, you can see that choosing $\theta_{\max} = 15$ insures that $\Pr(\Theta > \theta_{\max}) < 0.0001$.

The implementation in R can be found in Appendix D.

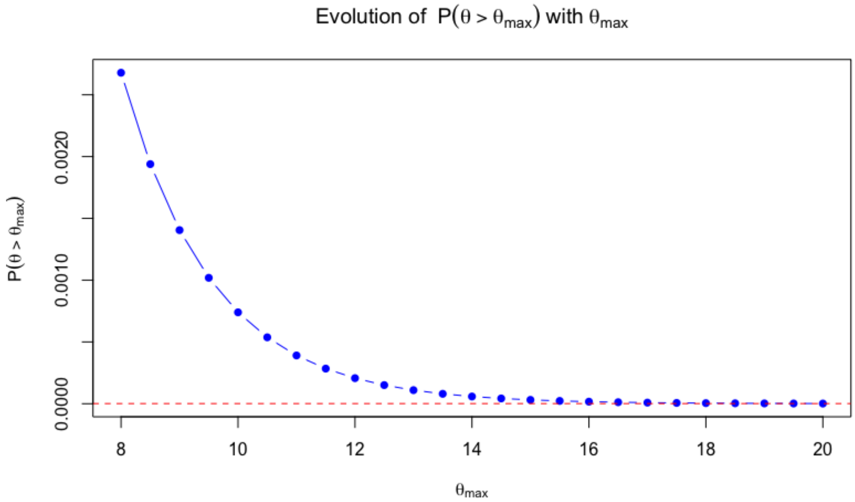


Figure 2.1: Evolution of the excess probability of Θ

2.2.1.3 Discrete model

In the discrete Good-driver/Bad-driver model, the unobserved heterogeneity Θ_i is assumed to take two possible values, θ_1 and θ_2, with corresponding probabilities g_1 and $g_2 = 1 - g_1$. This discretization simplifies the marginalization over Θ in the expressions for the stationary distribution and conditional expectations, replacing integrals by finite sums. In this setting, the marginal stationary distribution of the BMS level L is given by

$$\Pr[L = l] = \sum_{g=1}^{h} w_g \left[\pi_l(\lambda_g\theta_1) \cdot g_1 + \pi_l(\lambda_g\theta_2) \cdot g_2\right] \tag{2.19}$$

where $\pi_l(\lambda_g\theta_j)$ denotes the stationary probability of level l under transition intensity $\lambda_g\theta_j$, for $j \in \{1, 2\}$, and w_g is the proportion of policyholders in a priori class g. Similarly, the expected posterior premium $\Lambda^2\Theta$ given the BMS level $L = l$ is computed as

$$\mathbb{E}[\Lambda^2\Theta \mid L = l] = \frac{\sum_{g=1}^{h} w_g \left[\lambda_g^2\theta_1 \cdot \pi_l(\lambda_g\theta_1) \cdot g_1 + \lambda_g^2\theta_2 \cdot \pi_l(\lambda_g\theta_2) \cdot g_2\right]}{\Pr[L = l]} \tag{2.20}$$

and the conditional expectation of Λ^2 is

$$\mathbb{E}[\Lambda^2 \mid L = l] = \frac{\sum_{g=1}^{h} w_g \left[\lambda_g^2 \cdot \pi_l(\lambda_g\theta_1) \cdot g_1 + \lambda_g^2 \cdot \pi_l(\lambda_g\theta_2) \cdot g_2\right]}{\Pr[L = l]} \tag{2.21}$$

The use of discrete heterogeneity allows all components to be computed using closed-form stationary distributions and simple weighted sums, avoiding the need for numerical integration required in continuous models and its associated challenges.

2.2.1.4 Implementation and numerical results

Note : implementation of this section in R can be found in Appendix D.

As mentioned above, the BMS with a frozen state cannot be handled in this framework. Indeed, the stationary distribution π is degenerate for those BMS since every policyholder in the portfolio has a non-zero probability to reach this frozen state. Asymptotically, we'll end-up with all policyholders in that state. For those BMS, we then have

$$\pi_{\text{frozen state}}(\lambda\theta) = 1 \qquad \text{and} \qquad \pi_l(\lambda\theta) = 0, \quad \text{for } l \neq \text{frozen state}$$

Those BMS are then not eligible for our method. However, we'll develop another approach avoiding the use of the stationary distribution in chapter 3. The two eligible BMS are then the BMS of Ethias and P&V.

Ethias
The optimal relativities (as defined in (2.12)) computed on the BMS of Ethias can be found in Figure 2.2.

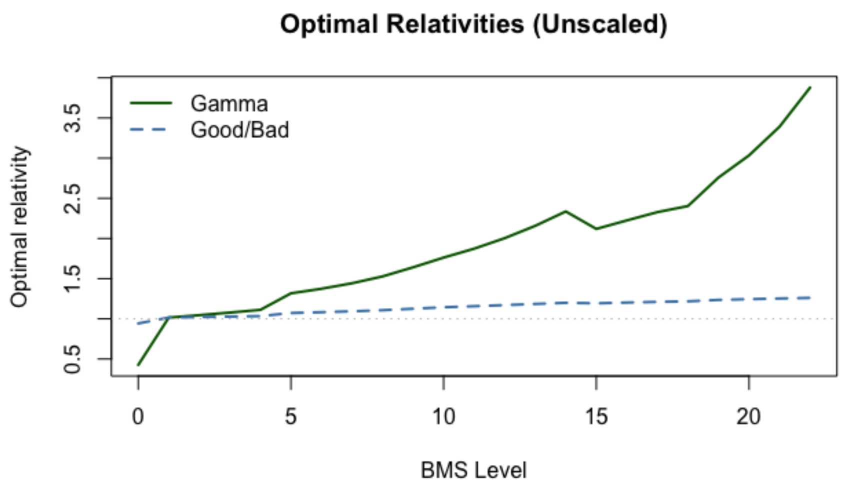


Figure 2.2: Ethias - Optimal relativities in stationary regime

In most of the BMS, inspired by the former compulsory Belgian BMS, the level with relativity of 100% is at (or around) the entry level of the BMS, which is more interesting from a business point of view since the relativities in the scale are more intuitive when you first have a 100% relativity. We can clearly see that this is not the case in Figure 2.2. To remedy this, we will re-scale the relativities so that the level at 100% in the relativities given by Ethias matches with our level at 100%. The associated premium is then scaled accordingly, and the scale becomes much clearer for the policyholder. The scaled relativities, besides the relativities provided by Ethias, can be found in Figure 2.3.

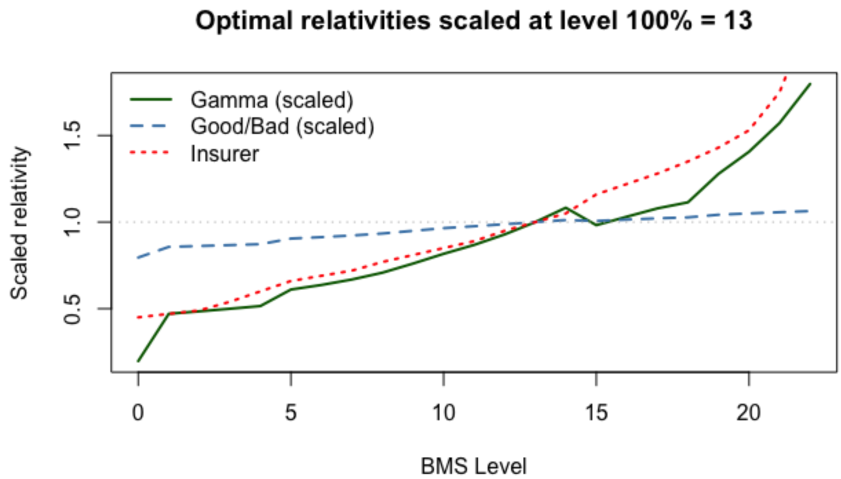
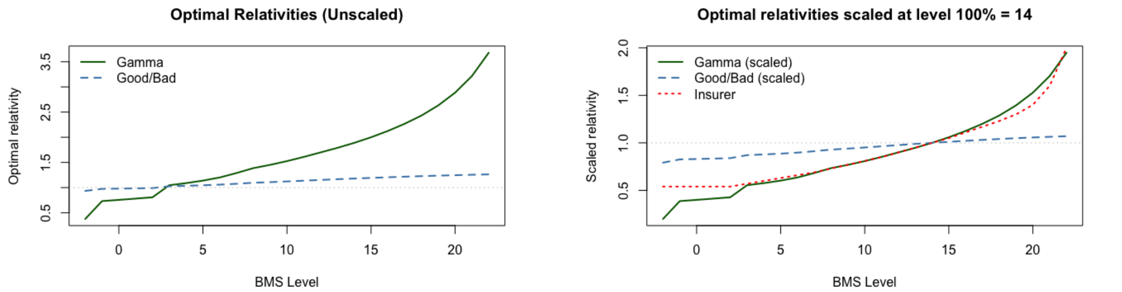


Figure 2.3: Ethias - Scaled optimal relativities in stationary regime

We first notice that the Gamma model fits better the relativities used by Ethias (while those of Ethias are slightly above the optimal values, which is conservative). The trend of the relativities is increasing, except at level 15, which was not guaranteed mathematically but matches the intuition, it's also desirable from a business point of view. The drop at level 15 can be explained by policyholders that used to report many claims and were in the upper levels, but at some point benefit from the special bonus rule and are sent back to level 14. The relativity at level 14 is then pushed up to incorporate these situations. The ordering of relativities is restored as of level 15. This matches the observation made in [PDW03a] on the former Belgian BMS. We also see that the discrete model seems to overestimate the relativities for lower levels while underestimating relativities for upper levels. This first illustration shows the limit of such a simple model in modeling a real-life BMS. However, the behavior of the model is coherent and has the advantage to give a simple framework to understand the mechanics of the *a posteriori* ratemaking.

P&V

The same scaling is applied to the optimal relativities computed for P&V, both plots can be found in Figure 2.4.



(a) Unscaled

(b) Scaled

Figure 2.4: P&V - Optimal relativities in stationary regime

We notice that the Gamma model is also a closer to the insurer-provided relativities. An underestimate of lower levels relativities compensates the overestimation of upper levels relativities, compared to P&V figures.

2.2.2 Linear relativities

The relativities obtained following the approach described above suffer one drawback : we have no guarantee of the trend they follow across levels. This could results in a erratic evolution of the relativities from one level to another. This situation is hardly tolerable from a commercial perspective. One way to remedy this behavior is to constraint the relativities, for example in a linear form. Tan has followed this approach in [Tan16a], following the same reasoning as above with $r_L^{linear} = a + bL$. Problem (2.5) writes as follows

$$(a, b) = \underset{a,b}{\operatorname{argmin}}\, \mathbb{E}[(\Lambda\Theta - \Lambda a - \Lambda bL)^2] \quad s.t. \quad a + b\mathbb{E}[L] \geq 1 \tag{2.22}$$

where the constraint comes from the financial equilibrium criterion $\mathbb{E}[r_L] \geq 1$ (cf. above).

The Lagrangian of (2.22) is

$$\mathcal{L}(a, b, \gamma) = \mathbb{E}[(\Lambda\Theta - \Lambda a - \Lambda bL)^2] + \gamma(1 - a - b\mathbb{E}[L]) \tag{2.23}$$

When trying to find solve a nonlinear optimization program

$$\min f(\mathbf{x})$$

such that $g(\mathbf{x}) \leq 0$ with lagrangian

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$$

then those conditions are necessary so that $\mathbf{x}^*$ is a minimum:
First order conditions

$$\frac{\partial}{\partial \mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda) = 0$$

Primal feasibility

$$g(\mathbf{x}^*) \leq 0$$

Dual feasibility

$$\lambda \geq 0$$

Complementary slackness

$$\lambda g(\mathbf{x}^*) = 0$$

This set of conditions is known as the Karush - Kuhn -Tucker (KKT) conditions. The KKT conditions associated to problem (2.22) are

KKT 1 :

$$\begin{aligned} \frac{\partial}{\partial a} \mathcal{L}(a, b, \gamma) &= 0 \\ \mathbb{E}\left[-2\Lambda^2\Theta + 2a\Lambda^2 + 2\Lambda^2 bL\right] - \gamma &= 0 \\ -2\left(\mathbb{E}[\Lambda^2\Theta] - a\mathbb{E}[\Lambda^2] - b\mathbb{E}[\Lambda^2 L]\right) - \gamma &= 0 \end{aligned} \tag{2.24}$$

KKT 2:

$$\begin{aligned} \frac{\partial}{\partial b} \mathcal{L}(a, b, \gamma) &= 0 \\ \mathbb{E}\left[-2\Lambda^2(\Theta - a)L + 2\Lambda^2 bL^2\right] - \gamma\mathbb{E}[L] &= 0 \\ -2\left(\mathbb{E}[\Lambda^2\Theta L] - a\mathbb{E}[\Lambda^2 L] - b\mathbb{E}[\Lambda^2 L^2]\right) - \gamma\mathbb{E}[L] &= 0 \end{aligned} \tag{2.25}$$

KKT 3:

$$\frac{\partial}{\partial \gamma} \mathcal{L}(a, b, \gamma) = 1 - a - b\mathbb{E}[L] \leq 0 \tag{2.26}$$

KKT 4:

$$\gamma(1 - a - b\mathbb{E}[L]) = 0 \tag{2.27}$$

KKT 5:

$$\gamma \geq 0 \tag{2.28}$$

For latter development, let us define

$$
\begin{aligned}
A = \mathbb{E}[\Lambda^2] &= \sum_{g=1}^{h} \lambda_g^2 \Pr[\Lambda = \lambda_g] && (2.29)\\
&= \sum_{g=1}^{h} \lambda_g^2 \sum_{l=1}^{s} \Pr[\Lambda = \lambda_g | L = l] \Pr[L = l] && (2.30)\\
&= \sum_{g=1}^{h} \lambda_g^2 \sum_{l=1}^{s} \Pr[L = l | \Lambda = \lambda_g] \Pr[\Lambda = \lambda_g] && (2.31)\\
&= \sum_{g=1}^{h} w_g \lambda_g^2 \sum_{l=1}^{s} \int_0^{+\infty} \Pr[L = l | \Lambda = \lambda_g, \Theta = \theta] f(\theta) d\theta && (2.32)\\
&= \sum_{g=1}^{h} w_g \lambda_g^2 \sum_{l=1}^{s} \int_0^{+\infty} \pi_l(\lambda_g \theta) f(\theta) d\theta && (2.33)\\
&= \sum_{l=1}^{s} \sum_{g=1}^{h} w_g \lambda_g^2 \int_0^{+\infty} \pi_l(\lambda_g \theta) f(\theta) d\theta && (2.34)
\end{aligned}
$$

And in a similar fashion

$$B = \mathbb{E}[\Lambda^2 \Theta] = \sum_{l=1}^{s} \sum_{g=1}^{h} w_g \lambda_g^2 \int_0^{+\infty} \theta \pi_l(\lambda_g \theta) f(\theta) d\theta \tag{2.35}$$

$$C = \mathbb{E}[\Lambda^2 L] = \sum_{l=1}^{s} l \sum_{g=1}^{h} w_g \lambda_g^2 \int_0^{+\infty} \pi_l(\lambda_g \theta) f(\theta) d\theta \tag{2.36}$$

$$D = \mathbb{E}[\Lambda^2 L^2] = \sum_{l=1}^{s} l^2 \sum_{g=1}^{h} w_g \lambda_g^2 \int_0^{+\infty} \pi_l(\lambda_g \theta) f(\theta) d\theta \tag{2.37}$$

$$E = \mathbb{E}[\Lambda^2 \Theta L] = \sum_{l=1}^{s} l \sum_{g=1}^{h} w_g \lambda_g^2 \int_0^{+\infty} \theta \pi_l(\lambda_g \theta) f(\theta) d\theta \tag{2.38}$$

$$F = \mathbb{E}[L] = \sum_{l=1}^{s} l \sum_{g=1}^{h} w_g \int_0^{+\infty} \pi_l(\lambda_g \theta) f(\theta) d\theta \tag{2.39}$$

The KKT conditions become

$$
\begin{aligned}
-2(B - aA - bC) - \gamma = 0 && (2.40)\\
-2(E - aC - bD) - \gamma F = 0 && (2.41)\\
1 - a - bF \leq 0 && (2.42)\\
\gamma(1 - a - bF) = 0 && (2.43)\\
\gamma \geq 0 && (2.44)
\end{aligned}
$$

2.2.2.1 Unconstrained

If we release the financial equilibrium constraint by setting $\gamma_{unconstrained} = 0$, conditions (2.43) and (2.44) are satisfied. With conditions (2.40) and (2.41), the problem becomes

$$\begin{bmatrix} A & C \\ C & D \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} B \\ E \end{bmatrix}$$

which solves for (a, b) as

$$a = \frac{DB - CE}{AD - C^2} \quad \text{and} \quad b = \frac{AE - CB}{AD - C^2} \tag{2.45}$$

However, condition (2.42) may not be satisfied, making the insurer lose money in the long run. If condition (2.42) is fulfilled, it means that the solution of the unconstrained problem is the same as the solution of the constrained problem. The condition doesn't play any role in the optimization problem : the constraint is not binding.

2.2.2.2 Financial equilibrium constraint

If $\gamma > 0$, (2.43) reduces to

$$(1 - a - bF) = 0 \tag{2.46}$$

By incorporating this extra condition in the unconstrained model, we get to solve (2.40), (2.41) and (2.46) simultaneously, that is

$$\begin{bmatrix} 2A & 2C & -1 \\ 2C & 2D & -F \\ 1 & F & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ \gamma \end{bmatrix} = \begin{bmatrix} 2B \\ 2E \\ 1 \end{bmatrix}$$

which solves for (a, b, γ) as

$$\gamma = \frac{(A - \frac{C^2}{D})(D - \frac{C^2}{A}) - (A - \frac{C^2}{D})(EF - \frac{BCF}{A}) - (D - \frac{C^2}{A})(B - \frac{EC}{D})}{(A - \frac{C^2}{D})(\frac{F^2}{2} - \frac{CF}{2A}) + (D - \frac{C^2}{A})(\frac{1}{2} - \frac{CF}{2D})} \tag{2.47}$$

$$a = \frac{B - \frac{C(E + \frac{\gamma}{2}F)}{D} + \frac{\gamma}{2}}{A - \frac{C^2}{D}} \tag{2.48}$$

$$b = \frac{E - \frac{C(B + \frac{\gamma}{2})}{A} + \frac{\gamma}{2}F}{D - \frac{C^2}{A}} \tag{2.49}$$

The same remark as above applies to condition (2.42).

2.2.2.3 Implementation and numerical results

Note : implementation of this section in R can be found in Appendix E.

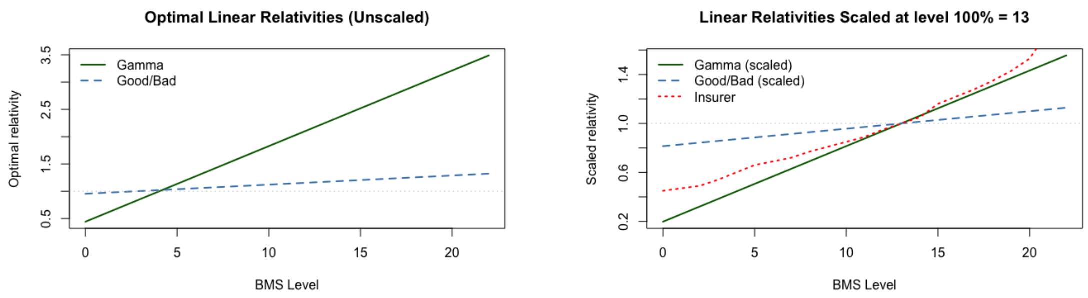
For the same reason as in the non-linear case, we are only going to look at the BMS of Ethias and P&V in this section. The optimal linear relativities are computed taking the financial equilibrium constraint into account, that is

$$r_l = a + bl$$

for each level l, with a and b defined as in (2.48) and (2.49).

Ethias

The optimal linear relativities computed on the BMS of Ethias can be found in Figure 2.5.



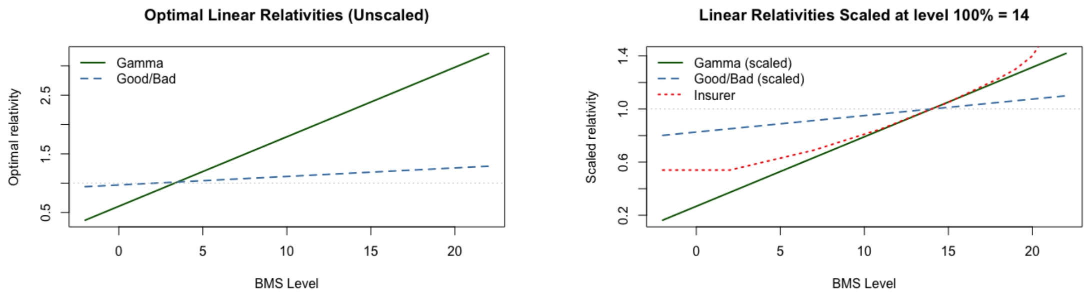
(a) Unscaled (b) Scaled

Figure 2.5: Ethias - Optimal linear relativities in stationary regime

We can once again see that the Good-driver/Bad-driver model is providing a good trend but that the Gamma model is closer to Ethias relativities. We clearly observe that the linear relativities of the Gamma model are close to Ethias ones in the neighborhood of the entry level (level 13). In the more extreme levels, Ethias is above our estimation, which is conservative. However, we remind the reader that we took an "edulcorated" version of Ethias BMS, which allows for larger discounts if several years are spent in level 0. The integration of the full Ethias model would bring the curve of Ethias closer to our Gamma model in level 0.

P&V

The same computation is done to obtain the optimal linear relativities computed for P&V, both plots can be found in Figure 2.6.



(a) Unscaled (b) Scaled

Figure 2.6: P&V - Optimal linear relativities in stationary regime

We observe the same situation as for Ethias. Indeed, the linear relativities of the Gamma model are close to P&V ones in the neighborhood of the entry level (level 14). In the more extreme levels, P&V is above our estimation, which is conservative.

2.2.3 Optimal transition rules

As explained by Tan in [Tan+15], one of the pitfalls of fixed transition rules is the clustering effect in the higher BMS levels of policyholders with a higher *a priori* ratemaking. The goal is to define a metric assessing the ability of the scale to let the policyholders move across the scale, making his *a priori* characteristics undistinguishable.
The average *a priori* expected claim frequency in level l is

$$\mathbb{E}[\Lambda|L=l] = \frac{\sum_{g=1}^{h} w_g \int_0^{+\infty} \lambda_g \pi_l(\lambda_g\theta) f(\theta) d\theta}{\sum_{g=1}^{h} w_g \int_0^{+\infty} \pi_l(\lambda_g\theta) f(\theta) d\theta} \tag{2.50}$$

As we want to reduce the volatility in each level l, we are seeking for transition rules with stable $\mathbb{E}[\Lambda|L=l]$ across the different l's, that is we want $\mathbb{V}[\mathbb{E}[\Lambda|L]]$ to be low. From Eve's law, we get that

$$\mathbb{V}[\Lambda] = \mathbb{E}[\mathbb{V}[\Lambda|L]] + \mathbb{V}[\mathbb{E}[\Lambda|L]] \tag{2.51}$$

As we want the second component of the sum ($\mathbb{V}[\mathbb{E}[\Lambda|L]]$) to be small, for a fixed $\mathbb{V}[\Lambda]$, the first component ($\mathbb{E}[\mathbb{V}[\Lambda|L]]$) will be large.
Thus, [Tan+15] suggested to take

$$\tau = \frac{\mathbb{E}[\mathbb{V}[\Lambda|L]]}{\mathbb{V}[\Lambda]} \tag{2.52}$$

as a measure of the effectiveness of the transition rules. The higher τ, the higher the flexibility of the scale, allowing policyholders from any *a priori* risk class to attain any BMS level in the steady state.
The clustering effect we want to avoid can be approached using a set of varying transition rules. That is, for high levels policyholders, to set a higher bonus and a lower malus transitions, allowing them to escape more easily the higher levels of the scale.
For a policyholder at level l who has reported k claims in the year, let's denote $t_{l,k}$ the effective transition rule for $l = 1, \ldots, s$ and $k \geq 0$. The idea is to have that, for $l, l_1, l_2 = 1, \ldots, s$:

- $t_{l,0} \leq 0$: bonus transition rule makes the policyholder move down the scale
- $t_{l,k} \geq 0$, for $k \geq 1$: malus transition rule makes the policyholder move up the scale
- $t_{l_2,k} \leq t_{l_1,k}$, for $k \geq 1$ and $l_2 > l_1$: malus transition rule has larger magnitude for lower scale levels
- $t_{l_2,0} \geq t_{l_1,0}$, for $l_2 > l_1$: bonus transition rule has higher magnitude for higher scale levels

In [Tan+15], this simple level-varying transition rule is considered:

$$t_{l,0} = \begin{cases} 0, \text{ for } l = 1 \\ -1, \text{ for } 2 \leq l \leq \left\lceil \frac{s}{2} \right\rceil + 1 \\ -2, \text{ for } l > \left\lceil \frac{s}{2} \right\rceil + 1 \end{cases}$$

$$t_{l,k} = \begin{cases} \min \left[s - l, \max \left[k, \left\lceil \frac{s-l}{p} \times k \right\rceil \right] \right], \text{ for } k \geq 1, l < s \\ 0, \text{ for } k \geq 1, l = s \end{cases}$$

It allows 2 different bonus transitions: -1 for the lower part of the scale and -2 for the higher levels[2]. For the malus transition, p is selected by the insurer and reflects the smallest number of claims required for the drivers to move from level l to level s. Taking the maximum with k ensures that there is at least one level penalty per claim reported and the minimum with $s - l$ is a safety condition to not go beyond highest level s.

[2] The ceiling function $\lceil s/2 \rceil$ is only there to ensure that there are more levels with -1 bonus than -2

We will not go further into details since this study focuses on relativities. However, the interested reader could find more about varying transition rules in [Tan+15] and [Tan16b].

Chapter 3

Transient Regime

In the previous chapter, we introduced a comprehensive framework to analyze BMS based on optimal relativities. This approach is centered around the core concept of stationary distribution across levels. However, two main concerns arise:

1. We don't know at which pace this stationary distribution is attained. If the period needed to reach this situation is significantly larger than the usual time policies are in force in the portfolio, it is no longer relevant.

2. BMS featuring a "frozen" state (as detailed in chapter 1) have a degenerate stationary distribution, with unit probability mass at the frozen state. Indeed, every policyholder in the portfolio has a non-zero probability to reach this frozen state, which leads asymptotically to an undesired scenario[1].

We will first analyze the convergence rate of the studied BMS in section 3.1, to illustrate our first concern. A new framework, based on the transient behavior of the BMS, is then presented in section 3.2. Finally, the linear adaptation of this model is detailed in section 3.3.

3.1 Convergence to the steady-state

The convergence of the BMS to the steady-state can be defined in multiple ways. In a broad sense, we want to assess the convergence to zero of the distance between the distribution of policyholders across levels at time $t = n$ (i.e. the transient distribution) and the stationary distribution. Let $L_{\lambda\theta}^{(n)}$ be the level occupied by a policyholder with expected claims frequency $\lambda\theta$ after n years. The transient distribution $p^{(n)}(\lambda\theta) = (p_1^{(n)}(\lambda\theta), \dots, p_s^{(n)}(\lambda\theta))$ is defined as

$$p_l^{(n)}(\lambda\theta) = \Pr[L_{\lambda\theta}^{(n)} = l], \qquad \text{for} \quad l = 1 \dots s$$

[1]Of course, the termination of the contract is always an option on the insurer's side

A particular choice of distance made in [Den+07] is the total variation distance d_{TV} defined as

$$d_{TV}(p^{(n)}(\lambda\theta), \pi(\lambda\theta)) = \sum_{l=1}^{s} |p_l^{(n)}(\lambda\theta) - \pi_l(\lambda\theta)|, \qquad \text{for} \quad n \in \mathbb{N} \tag{3.1}$$

Picking a policyholder at random in the portfolio, we can drop $\lambda\theta$ and write $L^{(n)}$ the level occupied by this policyholder after n years in the BMS. Equation (3.1) can then be written

$$\begin{aligned} d_{TV}(p^{(n)}, \pi) &= \sum_{l=1}^{s} |p_l^{(n)} - \pi_l| \\ &= \sum_{l=1}^{s} \Big| \Pr[L^{(n)} = l] - \Pr[L = l] \Big| \end{aligned} \tag{3.2}$$

The expression $\Pr[L = l] = \sum_{g=1}^{h} w_g \int_0^{+\infty} \pi_l(\lambda_g\theta) f(\theta) d\theta$ has already been developed in equation (2.14). The expression of obtained the exact same way, thus

$$\Pr[L^{(n)} = l] = \sum_{g=1}^{h} w_g \int_0^{+\infty} p_l^{(n)}(\lambda_g\theta) f(\theta) d\theta \tag{3.3}$$

Putting it together in (3.2), we have

$$d_{TV}(p^{(n)}, \pi) = \sum_{l=1}^{s} \left| \sum_{g=1}^{h} w_g \int_0^{+\infty} \left(p_l^{(n)}(\lambda_g\theta) - \pi_l(\lambda_g\theta) \right) f(\theta) d\theta \right| \tag{3.4}$$

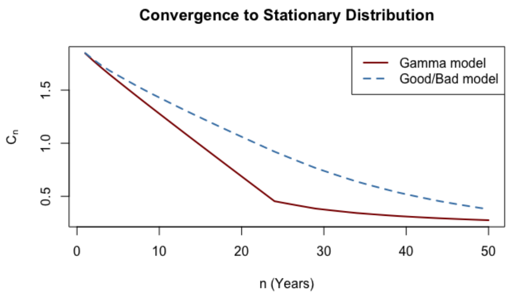
for $n \in \mathbb{N}$.
We can now assess the convergence of a BMS by computing (3.4) until $d_{TV}(p^{(n)}, \pi)$ falls below a defined threshold. Indeed, let $C_n = d_{TV}(p^{(n)}, \pi)$, we can assume the stationary state have been reached after n_0 years if we find a small $\epsilon > 0$ such that

$$C_n \leq \epsilon, \qquad \forall n \geq n_0$$

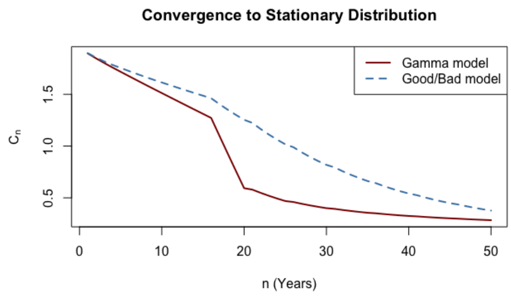
3.1.1 Implementation and results

Note : implementation of this section in R can be found in Appendix F.

We shall now analyze the convergence of the four BMS selected for our study. For each insurer, the evolution of C_n is represented as a function of the number of years $n = 0, \ldots, 50$.
In Figure 3.1a, we see that AG's BMS is converging slowly to $C_n = 0$, the Gamma model only passing $C_n = 0.1$ after 50 years. The Good-driver/Bad-driver model is converging even slower.



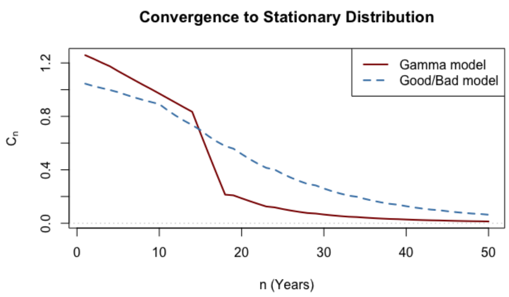
(a) AG - Convergence of the transient distribution to the stationary distribution



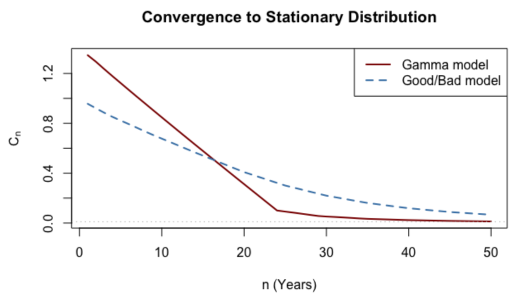
(b) Baloise - Convergence of the transient distribution to the stationary distribution

Figure 3.1: Convergence of the transient distribution to the stationary distribution

The convergence of Baloise's BMS is shown in Figure 3.1b. With the Gamma model, C_n converges faster in the first 20 years but then slows down to reach the same convergence performance as AG after 50 years. Once again, the discrete model performs less than the continuous model. With a quick look at the scales of Figures 3.2, we notice that Ethias and P&V converge faster to their steady-state. However, we should be cautious about the interpretation to give since stationary distributions of both AG and Baloise are degenerated (with a probability mass at frozen level). Indeed, our particular choice of distance might penalize more a sparse stationary distribution (as for AG and Baloise) than the stationary distribution of Ethias and P&V.



(a) Ethias - Convergence of the transient distribution to the stationary distribution



(b) P&V - Convergence of the transient distribution to the stationary distribution

Figure 3.2: Convergence of the transient distribution to the stationary distribution

On the left, Figure 3.2a shows a better convergence than the first two BMS analyzed. After a fast convergence phase, C_n decreases more slowly to reach $C_n < 0.05$ for $n \geq 39$ and finally $C_{50} = 0.021$. On the right, Figure 3.2b displays a fast convergence of C_n for P&V until 23 years. It then slows down to reach $C_n < 0.05$ for $n \geq 37$ and reach $C_{50} = 0.02$. Once again, the Good-driver/Bad-driver model exhibits worse convergence to the steady-state than the Gamma model.

As we saw in the four studied BMS, the convergence to the stationary distribution can be considered as slow. Indeed, after 50 years, 2 BMS of out 4 didn't eventually reach a reasonable threshold of $C_n = 0.05$.

3.2 Transient optimality

As we showed in the previous section, convergence to steady state can be very slow. We should then establish a new approach so that the optimal relativities found will rely on the transient behavior of the BMS. [Den+07] suggested an optimality criterion taking into account the transient distribution presented above.

3.2.1 State of the art

In the stationary regime framework, a policyholder was described by two variables: Λ, the *a priori* expected claims frequency and Θ, the residual heterogeneity in each risk class. In a transient setting, the evolution of the policyholder in the BMS will also depend on a third variable, the age of the policy in the portfolio, A. The distribution of A can be described as

$$\Pr[A = n] = a_n, \quad n \in \mathbb{N} \tag{3.5}$$

with a_n the proportion of policies in the portfolio in force for n years. We have explained in section 1.3 why we can consider Λ and Θ to be independent. Since the *a priori* ratemaking captured by Λ takes into account the age of the policyholder, it is most likely that Λ and A, the age of the policy in the portfolio, are correlated. However, we will assume independence between Λ, Θ and A to avoid cumbersome algebra.[2]

For a randomly picked policyholder in the portfolio, let us denote L_A the level occupied by that policyholder after A years in the BMS, and $r_{L_A}^{(A)}$ the relativity applied to that policyholder. [Den+07] suggested that, for each group of policyholders with age of policy n, the objective function to minimize is

$$Q_n = \mathbb{E}\left[(\Theta - r_{L_A}^{(A)})^2 | A = n\right] = \mathbb{E}\left[(\Theta - r_{L_n}^{(n)})^2\right] \tag{3.6}$$

[2] A dependence model between those variables could be foreseen. We will not go further in that direction but it could be interesting for future analysis.

As a portfolio might have very different ages of policies, [Den+07] then suggested to weight this criterion based on the proportion of each group of age in the portfolio, that is

$$\bar{Q} = \mathbb{E}\left[(\Theta - r^{(A)}_{L_A})^2\right] = \sum_{n=1}^{\infty} a_n Q_n \tag{3.7}$$

3.2.2 Transient model

For consistency reasons, we would want, as detailed in section 2.2.1, to incorporate the *a priori* expected claims frequency in our objective function. Let us redefine Q_n as

$$Q_n = \mathbb{E}\left[(\Lambda\Theta - \Lambda r^{(A)}_{L_A})^2 | A = n\right] = \mathbb{E}\left[(\Lambda\Theta - \Lambda r^{(n)}_{L_n})^2\right] \tag{3.8}$$

The financial equilibrium condition (2.4) applied in the stationary scenario insured that the BMS is financially balanced once the steady-state has been reached. Maintaining this equilibrium during the transient phase of the system would require adjusting the relativities applied to policyholders in each period n n so that they coincide with $r^{(n)}_l$ for that period. From a commercial perspective, however, the implementation of such a strategy appears impractical. However, the financial balance of the BMS is of most importance for the actuary. We'll show how to keep track of that balance in section 3.2.3.
We then get to solve this problem:

$$r^{(n)}_l = \operatorname{argmin} \mathbb{E}\left[(\Lambda\Theta - \Lambda r^{(n)}_{L_n})^2\right] \tag{3.9}$$

Let us develop the first-order conditions

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{r}} Q_n &= \frac{\partial}{\partial \mathbf{r}} \left(\mathbb{E}\left[(\Lambda\Theta - \Lambda r^{(n)}_{L_n})^2\right]\right) \\
&= \frac{\partial}{\partial \mathbf{r}} \left(\sum_{l=1}^{s} \mathbb{E}[(\Lambda\Theta - \Lambda r^{(n)}_{L_n})^2 | L_n = l] \Pr[L_n = l]\right) \\
&= \frac{\partial}{\partial \mathbf{r}} \left(\sum_{l=1}^{s} \mathbb{E}[\Lambda^2\Theta^2 - 2\Lambda^2\Theta r^{(n)}_{L_n} + \Lambda^2 (r^{(n)}_{L_n})^2 | L_n = l] \Pr[L_n = l]\right) \\
&= \mathbb{E}[-2\Lambda^2\Theta + 2\Lambda^2 r^{(n)}_{L_n} | L_n = l] \Pr[L_n = l] \\
&= \Pr[L_n = l] \left(2\mathbb{E}[\Lambda^2 r^{(n)}_{L_n} - \Lambda^2\Theta | L_n = l]\right) = 0 \\
\iff& \Pr[L_n = l] \left(2\mathbb{E}[\Lambda^2\Theta - \Lambda^2 r^{(n)}_{L_n} | L_n = l]\right) = 0 \\
\iff& \mathbb{E}[\Lambda^2 r^{(n)}_{L_n} | L_n = l] - \mathbb{E}[\Lambda^2\Theta | L_n = l] = 0
\end{aligned} \tag{3.10}$$

for $l = 1, \dots, s$. Isolating $r_l^{(n)}$ on the left-hand side, we get

$$r_l^{(n)} = \frac{\mathbb{E}[\Lambda^2 \Theta | L_n = l]}{\mathbb{E}[\Lambda^2 | L_n = l]} \tag{3.11}$$

with

$$\begin{aligned}
\mathbb{E}[\Lambda^2\Theta|L_n = l] &= \frac{\mathbb{E}[\Lambda^2\Theta \cdot \mathbb{I}_{L_n=l}]}{\Pr[L_n = l]} \quad (\mathbb{I} \text{ is the indicator function}) \\
&= \frac{\sum_{g=1}^h \Pr[\Lambda = \lambda_g] \int_0^{+\infty} \lambda_g^2 \theta \Pr[L_n = l | \Lambda = \lambda_g, \Theta = \theta] f(\theta) d\theta}{\Pr[L_n = l]} \\
&= \frac{\sum_{g=1}^h w_g \int_0^{+\infty} \lambda_g^2 \theta p_l^{(n)}(\lambda_g \theta) f(\theta) d\theta}{\sum_{g=1}^h w_g \int_0^{+\infty} p_l^{(n)}(\lambda_g \theta) f(\theta) d\theta}
\end{aligned} \tag{3.12}$$

and in a similar fashion

$$\mathbb{E}[\Lambda^2|L_n = l] = \frac{\sum_{g=1}^h w_g \int_0^{+\infty} \lambda_g^2 p_l^{(n)}(\lambda_g \theta) f(\theta) d\theta}{\sum_{g=1}^h w_g \int_0^{+\infty} p_l^{(n)}(\lambda_g \theta) f(\theta) d\theta} \tag{3.13}$$

The application to both Gamma and Good-driver/Bad-driver models is done as described in sections 2.2.1.2 and 2.2.1.3, replacing π_l by $p_l^{(n)}$.
Since (3.8) is defined conditionally to $A = n$, we would also want an unconditional criterion, not depending on the age of the policies. We thus define $\bar{Q}$ following the approach of [Den+07], but including once again the *a priori* expected claims frequency Λ ,that is

$$\bar{Q} = \sum_{n=1}^{\infty} a_n Q_n \tag{3.14}$$

with Q_n as defined in (3.8). The first-order conditions of the associated minimization problem are

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{r}} \bar{Q} &= \frac{\partial}{\partial \mathbf{r}} \sum_{n=1}^{\infty} a_n Q_n \\
&= \sum_{n=1}^{\infty} \frac{\partial}{\partial \mathbf{r}} a_n Q_n \\
&= \sum_{n=1}^{\infty} a_n \frac{\partial}{\partial \mathbf{r}} Q_n = 0
\end{aligned}$$

By injecting (3.10) in this last equation, we get:

$$\sum_{n=1}^{\infty} a_n \left(\mathbb{E}[\Lambda^2 r_{L_n}^{(n)} | L_n = l] - \mathbb{E}[\Lambda^2 \Theta | L_n = l] \right) = 0$$

$$\sum_{n=1}^{\infty} a_n \mathbb{E}[\Lambda^2 r_{L_n}^{(n)} | L_n = l] = \sum_{n=1}^{\infty} a_n \mathbb{E}[\Lambda^2 \Theta | L_n = l]$$

When then isolate

$$\bar{r}_l = \frac{\sum_{n=1}^{\infty} a_n \mathbb{E}[\Lambda^2\Theta | L_n = l]}{\sum_{n=1}^{\infty} a_n \mathbb{E}[\Lambda^2 | L_n = l]} \tag{3.15}$$

3.2.3 Financial equilibrium in transient regime

The BMS is financially balanced at year n if

$$\sum_{l=1}^{s} r_l^{(n)} \Pr[L_n = l] = 1$$

Asking for the BMS financial balance at all time would then require to change the relativities each year. It's of course hardly acceptable from a business point of view. Since the actuary is interested in keeping the financial equilibrium, an important indicator to be monitored is then

$$I_n = \sum_{l=1}^{s} r_l \Pr[L_n = l] = \sum_{l=1}^{s}\sum_{g=1}^{h} w_g \int_0^{+\infty} r_l p_l^{(n)}(\lambda_g \theta) f(\theta) d\theta \tag{3.16}$$

with relativities r_l fixed for each level over time.

3.2.4 Implementation and numerical results

Note : implementation of this section in R can be found in Appendix H.

In chapter 2, we didn't care so much about the initial distribution of policyholders in the BMS because in the long-run, the initial distribution does not play any role in the stationary distribution. However, the transient regime is very sensitive to the initial distribution. In order to assess our computations, we shall then perform a sensitivity analysis of the relativities to this initial distribution, L_0. Our basecase being a uniformly distributed L_0, we also assess the impact of a "top" initial distribution and a "bottom" initial distribution defined as such:

- Top distribution : 1% of policyholders for $l = 1, \ldots, s-1$ and the rest of the probability mass at the highest level $l = s$
- Bottom distribution : 1% of policyholders for $l = 2, \ldots, s$ and the rest of the probability mass at the lowest level $l = 1$

The same applies for the distribution of ages A, which has a direct influence on $\bar{r}_l$ (cf. (3.15)). The span of ages considered will be $n = 0, \ldots, 50$, following our analysis of convergence in the above section. A sensitivity test is also performed with three cases. The basecase is A uniformly distributed over $n = 0, \ldots, 50$, and we also define a "young" distribution of ages and an "old" distribution of ages, such that:

- Young distribution: More probability is put on small n and less on n close to 50
- Old distribution : More probability is put on n close to 50 and less on n close to 0.

The definition of A in R can be found in the script below (for ages $n = 0, \ldots, N$):

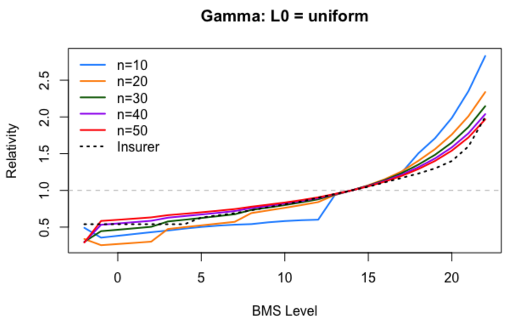
```r
build_An <- function(N, type = c("uniform","young","old")){
  if (type == "uniform") {
    An <- rep(1 / N, N)
  } else if (type == "young") {
    An <- c(rep(0.1,2),rep(0.05,5),rep(0.025,10),rep(0.3/(N-17),N
    -17))
  } else { # "old"
    An_young <- c(rep(0.1,2),rep(0.05,5),rep(0.025,10),rep(0.3/(N
    -17),N-17))
    An <- rev(An_young)
  }
  An
}
```

All relativities reported below are scaled following the same procedure as in chapter 2. Also, as already noticed in chapter 2, the Good-driver/Bad-driver model is interesting as a limiting case (e.g. for educational purpose) but is too restricted to provide meaningful results in our real-world BMS. We also saw in section 3.1.1 that the Gamma model has a higher convergence rate than the Good-driver/Bad-driver model. Therefore, we will only focus on the Gamma model in the rest of this chapter [3].

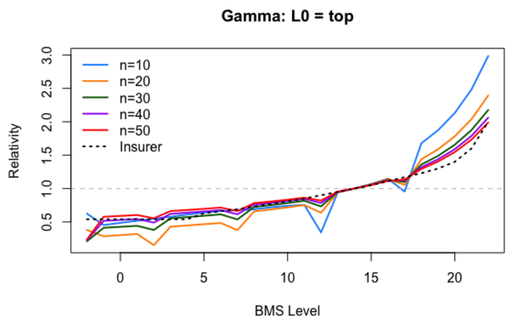
3.2.4.1 AG

The Figures 3.3 show the optimal transient relativities for the 3 initial distributions described above. The first observation is that the lower the number of years n considered, the more erratic the relativities. This matches intuition since not enough years are observed in the model to grasp the dynamics of the BMS, especially for the top initial distribution. This non-monotonic behavior is of course undesirable from a commercial perspective, however, the same remedy as in the stationary regime can be applied (i.e. imposing a fixed pattern to the relativities, see section 3.3 for linear relativities in transient regime).

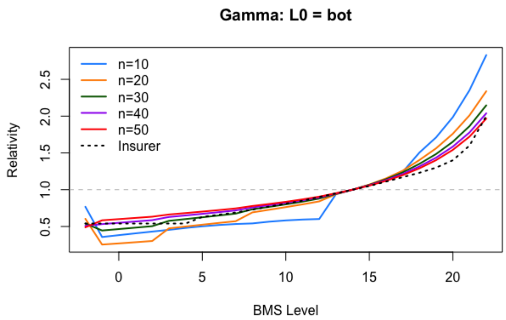
[3] If the reader is interested, the implementation of the Good-driver/Bad-driver model can be found in Appendix G



(a) with uniform initial distribution



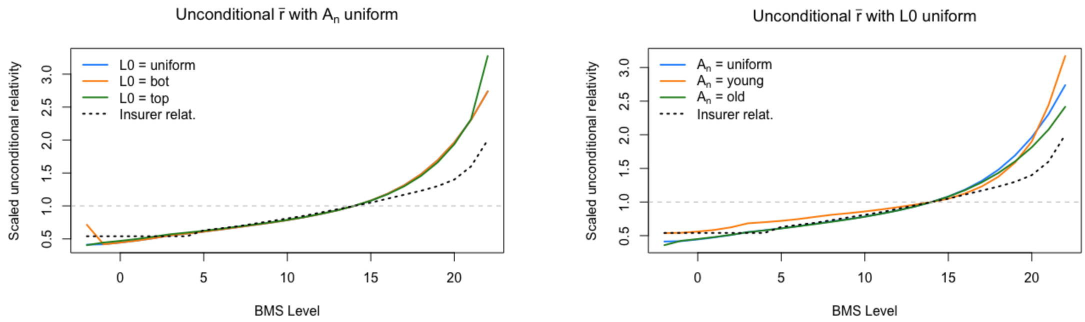
(b) with top initial distribution



(c) with bottom initial distribution

Figure 3.3: AG - $r_l^{(n)}$ for $n = 10, 20, 30, 50$ with a different initial distributions

As expected, we notice that the curves converge as n increases. It can also be noted that the curves are getting closer to AG relativities as n increases. We can also see on three figures 3.3 that the relativities are decreasing with n for the higher levels in the scale, while they increase with n for the lower levels. In the three cases, AG relativities is a good fit of the optimal relativities for n sufficiently large, except for higher levels, where AG relativities seem to underestimate the optimal relativities (while the gap shrinks with n increasing).

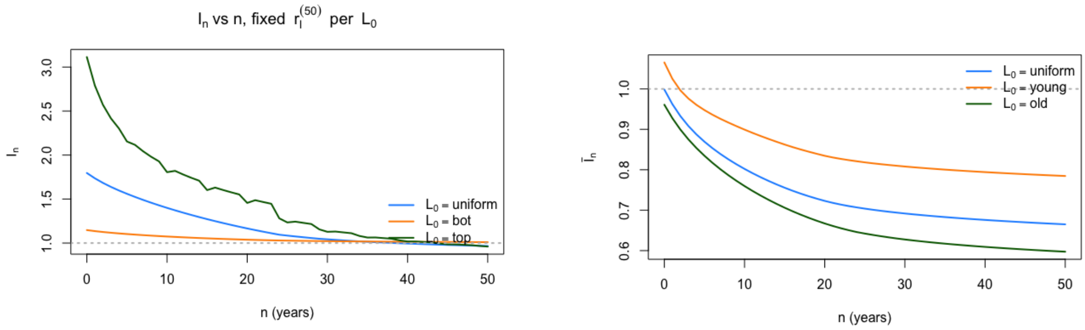


(a) Impact of the initial distribution on $\bar{r}_l$ (b) Impact of the age distribution on $\bar{r}_l$

Figure 3.4: AG - Evolution of $\bar{r}_l$ with the levels

In Figure 3.4a, we see the evolution of the unconditional relativities as defined in (3.15), for the three initial distributions described earlier (with uniformly distributed A). The trend of the optimal relativities seems to match the trend of AG relativities. However, the optimal criterion allows higher malus for higher levels. The uniform initial distribution has similar relativites as the top distribution for the lower levels and then follows the relativities given by the bottom distribution for higher levels. The higher relativity provided by the bottom distribution at level -2 may be explained by the fact the this level is frozen, making policyholder stick to this level even if they report a lot of claims.

In Figure 3.4b, the sensitivity is done on the distribution of the policies ages A. We see that the "youngest" distribution of ages provides higher relativities. As the portfolio becomes more mature, we are getting closer to the relativities of AG. However, optimal relativities fall below AG relativities for lower levels, while giving higher malus for higher levels.



(a) Evolution of financial balance with $r_l = r_l^{(n=50)}$

(b) Evolution of financial balance with $r_l = \bar{r}_l$

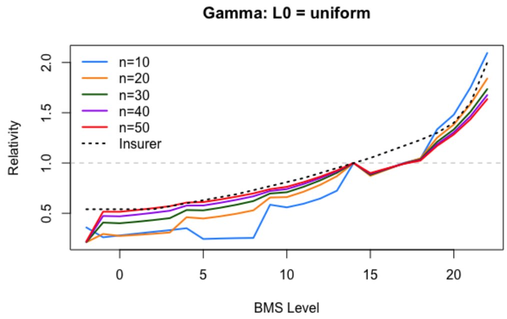
Figure 3.5: AG - Evolution of financial balance with the age of policies

As mentioned in section 3.2.3, the evolution of the BMS financial balance is an important indicator to follow. Figures 3.5 show the evolution of I_n with the number of years n, considering each of the three initial distributions. In the left figure 3.5a, the financial balance is computed using (3.16) with $r_l = r_l^{(n=50)}$, while in the right figure 3.5b, $r_l = \bar{r}_l$ with uniform distribution of ages $a_n = 1/50$ for $n = 1, \dots, 50$. In Figure 3.5a, we see that whatever the initial distribution, we have a financial profit in the first years but we eventually fall below the 100% threshold in the last observed years. This may be due to having too many policyholders in bonus levels in the long-run. Indeed, frozen level -2 gradually absorbs policyholders making the portfolio unavoidably loss making if the insurer does not terminate the contract. We also notice that the more policyholders in the higher levels initially, the more profit is generated in the first years. Indeed, the bottom initial distribution allocates more people in the bonus levels and thus generates less profitability.

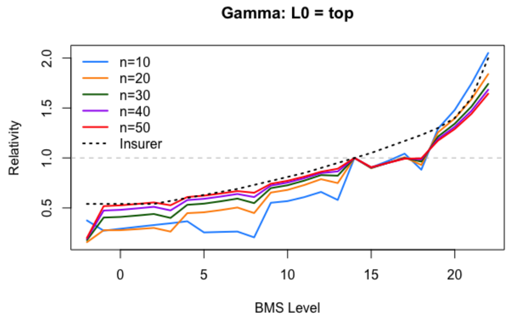
In Figure 3.5b, using $\bar{r}_l$ as relativities shows that the BMS rapidly goes below the profitability threshold. This matches the observation made by [Den+07].

3.2.4.2 Baloise

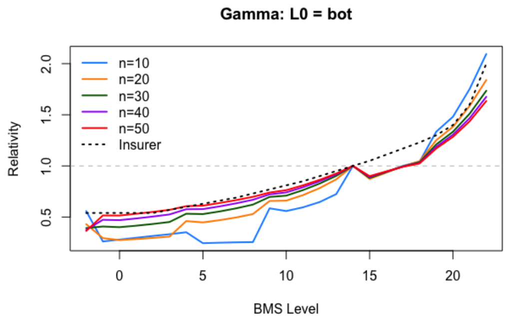
The same analysis as for AG is performed below. The Figures 3.6 show the optimal transient relativities for the 3 initial distributions described above. The same observation is made about the erratic behavior of relativities for low n.



(a) with uniform initial distribution



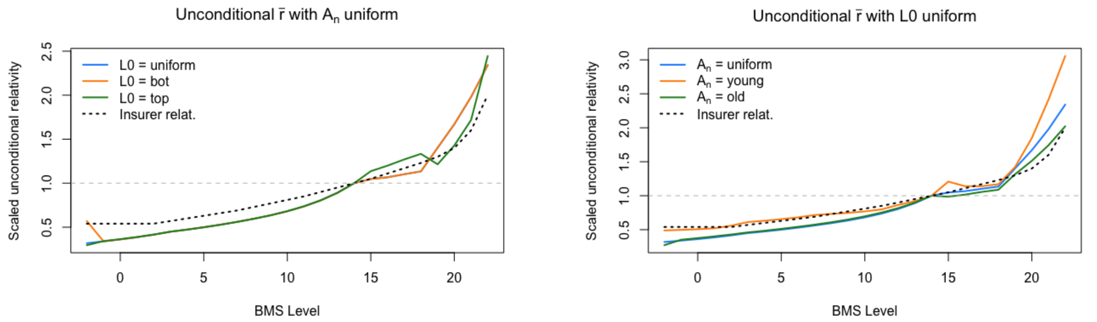
(b) with top initial distribution



(c) with bottom initial distribution

Figure 3.6: Baloise - $r_l^{(n)}$ for $n = 10, 20, 30, 50$ with a different initial distributions

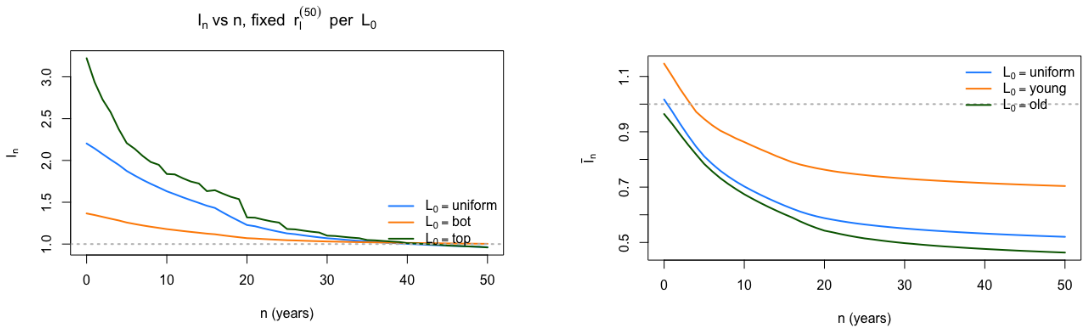
As expected, we notice that the curves converge as n increases. It can also be noted that, for lower levels, the curves are getting closer to Baloise relativities as n increases. Conversely, the curves converge to lower relativities than Baloise for levels above level 14. This may be caused by the special bonus rule incorporated in Baloise transition rules. Indeed, we scaled our relativities based on the level at 100% relativity (level 14). However, the special bonus rule introduces a spike in optimal relativities at the "return" level (here level 14), as explained in section 2.2.1.4 for Ethias BMS (which has this special bonus rule too). We can also see on three figures 3.6 that the relativities are decreasing with n for the higher levels in the scale, while they increase with n for the lower levels. In the three cases, Baloise relativities is a good fit of the optimal relativities for n sufficiently large, except for higher levels, where Baloise relativities seem to overestimate the optimal relativities.



(a) Impact of the initial distribution on $\bar{r}_l$ (b) Impact of the age distribution on $\bar{r}_l$

Figure 3.7: Baloise - Evolution of $\bar{r}_l$ with the levels

In Figure 3.7a, the optimal criterion allows higher malus for higher levels, and higher bonus in the lower levels too. The uniform initial distribution has similar relativites as the top distribution for the lower levels and then follows the relativities given by the bottom distribution for higher levels. The higher relativity provided by the bottom distribution at level -2 may be explained by the fact the this level is frozen, making policyholder stick to this level even if they report a lot of claims.
In Figure 3.7b, the sensitivity is done on the distribution of the policies ages A. The results are comparable to those of AG. Indeed, we see that the "youngest" distribution of ages provides higher relativities, as already observed for AG. As the portfolio becomes more mature, we are getting closer to the relativities of Baloise. However, optimal relativities fall below Baloise relativities for lower levels, while giving higher malus for higher levels.



(a) Evolution of financial balance with $r_l = r_l^{(n=50)}$ (b) Evolution of financial balance with $r_l = \bar{r}_l$

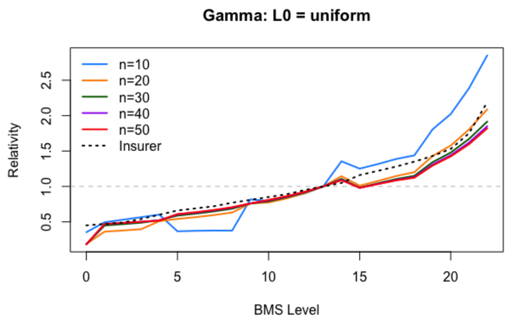
Figure 3.8: Baloise - Evolution of financial balance with the age of policies

In Figure 3.8a, we see that we obtain the same results as for AG, while it takes 3 years more to fall below the profitability threshold of 100% in Baloise's case.

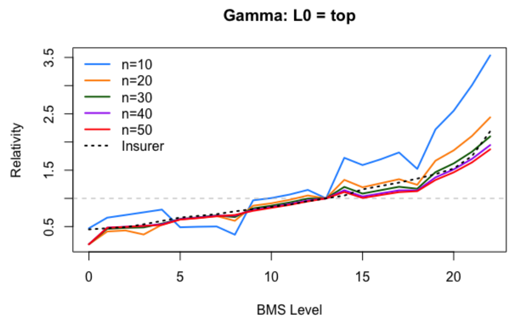
In Figure 3.8b, using $\bar{r}_l$ as relativities shows that the BMS rapidly goes below the profitability threshold as for AG. However, the uniform initial distribution is also generating profit for the 2 first years here.

3.2.4.3 Ethias

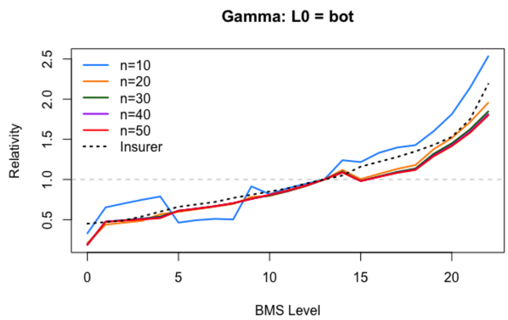
Figures 3.9 show the optimal transient relativities for the 3 initial distributions described above. The same observation as above is made about the erratic behavior of relativities for low n.



(a) with uniform initial distribution



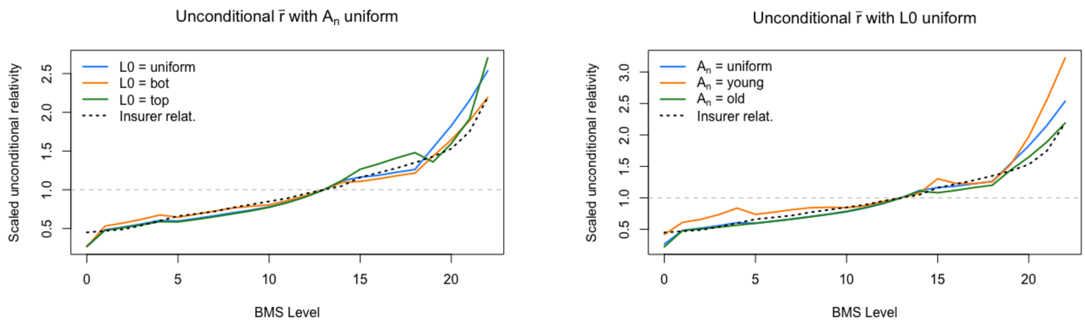
(b) with top initial distribution



(c) with bottom initial distribution

Figure 3.9: Ethias - $r_l^{(n)}$ for $n = 10, 20, 30, 50$ with a different initial distributions

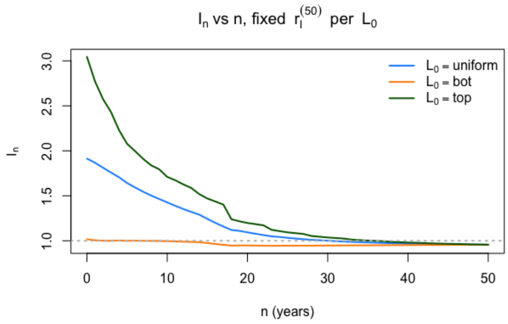
As expected, we notice that the curves converge as n increases. It can also be noted that, for lower levels, the curves are getting closer to Ethias relativities as n increases. Conversely, the curves converge to lower relativities than Ethias for levels above level 14. This may be caused by the special bonus rule incorporated in Ethias transition rules, as it was for Baloise (cf. Baloise section for further details). We can also see on three figures 3.9 that the relativities are decreasing with n for the higher levels in the scale, while they increase with n for the lower levels. In the three cases, Ethias relativities is a good fit of the optimal relativities for n sufficiently large, except for higher levels, where Ethias relativities seem to overestimate the optimal relativities. Furthermore, we also point out that the curves of $r_l^{(n)}$ converge to the steady-state relativities computed in section 2.2.1.4, whatever is the initial distribution of policyholders.



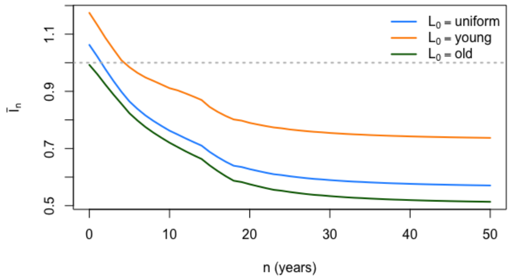
(a) Impact of the initial distribution on $\bar{r}_l$ (b) Impact of the age distribution on $\bar{r}_l$

Figure 3.10: Ethias - Evolution of $\bar{r}_l$ with the levels

In Figure 3.10a, the optimal criterion allows higher malus for higher levels, while fitting the curve of Ethias more closely for lower levels. The uniform initial distribution has similar relativities as the top distribution for the lower levels but splits apart for higher levels. As we don't have a frozen state at level 0 (as for the two previous BMS studied), we don't experience the higher relativity provided by the bottom distribution at level 0, which is in line with expectations.
In Figure 3.10b, the sensitivity is done on the distribution of the policies ages A. The results are comparable to those of the other BMS. Indeed, we see that the "youngest" distribution of ages provides higher relativities, as already observed above. As the portfolio becomes more mature, we are getting closer to the relativities of Ethias, ending up with a god fit for the old distribution of ages. We notice a drop in estimated relativities at level 0. However, we recall that Ethias gives an additional discount for consecutive years spent at level 0, which is not captured here.



(a) Evolution of financial balance with $r_l = r_l^{(n=50)}$



(b) Evolution of financial balance with $r_l = \bar{r}_l$

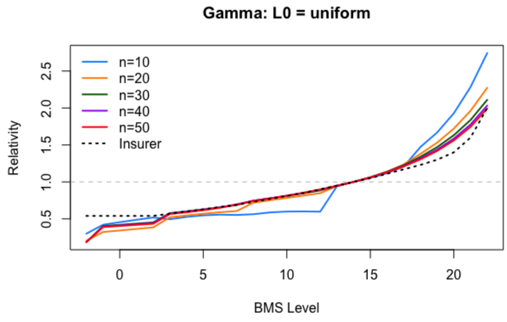
Figure 3.11: Ethias - Evolution of financial balance with the age of policies

In Figure 3.11a, we observe a similar behavior of the financial balance. However, the situation becomes loss making in a fewer years than for the BMS studied above, especially for the bottom initial distribution.

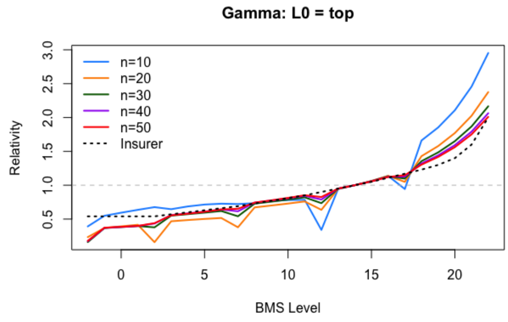
In Figure 3.11b, using $\bar{r}_l$ as relativities shows that the BMS rapidly goes below the profitability threshold, as for other BMS. However, the uniform and young initial distributions are generating profit for more years than for Baloise and AG.

3.2.4.4 P&V

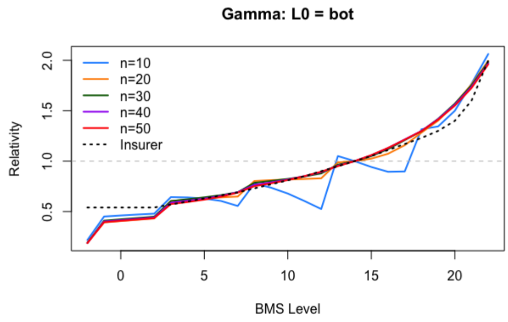
Figures 3.12 show the optimal transient relativities for the 3 initial distributions described above. The same observation as above is made about the erratic behavior of relativities for low n.



(a) with uniform initial distribution



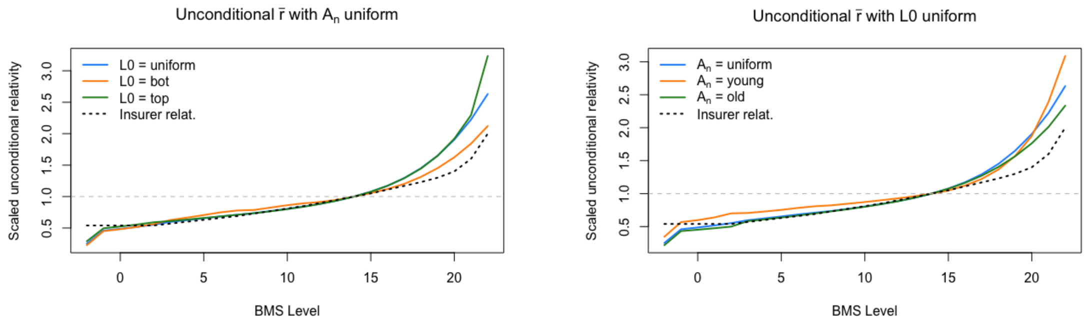
(b) with top initial distribution



(c) with bottom initial distribution

Figure 3.12: P&V - $r_l^{(n)}$ for $n = 10, 20, 30, 50$ with a different initial distributions

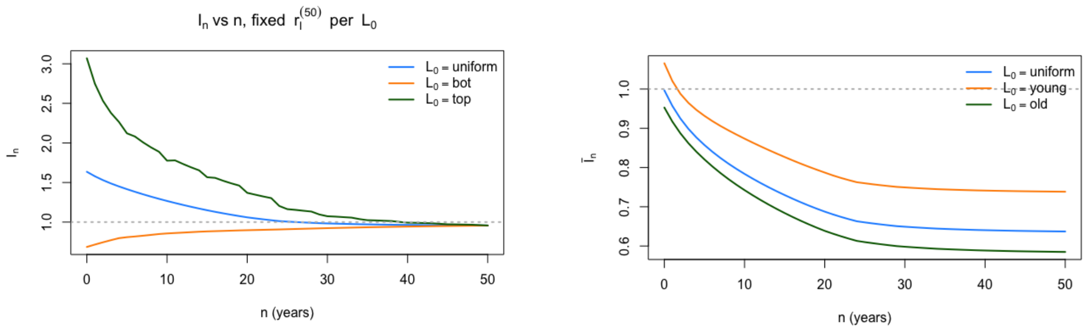
We once again observe the convergence of the curves as n increases. It can also be noted that the curves are getting closer to P&V relativities as n increases. We can also see on three figures 3.12 that the relativities are decreasing with n for the higher levels in the scale, while their evolution with n is more erratic for the lower levels. In the three cases, P&V relativities is a good fit of the optimal relativities for n sufficiently large. However, as in the stationary regime, an underestimate of lower levels relativities compensates the overestimation of upper levels relativities, compared to P&V figures. Indeed, independently of the initial distribution, the relativities converge to the optimal stationary relativities, as expected.



(a) Impact of the initial distribution on $\bar{r}_l$ (b) Impact of the age distribution on $\bar{r}_l$

Figure 3.13: P&V - Evolution of $\bar{r}_l$ with the levels

In Figure 3.13a, the optimal criterion allows higher malus for higher levels, while fitting the curve of P&V more closely for lower levels. The uniform initial distribution has similar relativities as the top distribution for the lower levels but splits apart for higher levels.
In Figure 3.13b, the sensitivity is done on the distribution of the policies ages A. The results are comparable to those of the other BMS. Indeed, we see that the "youngest" distribution of ages provides higher relativities, as already observed above. As the portfolio becomes more mature, we are getting closer to the relativities of P&V, ending up with a god fit for the old distribution of ages, except for higher levels.



(a) Evolution of financial balance with $r_l = r_l^{(n=50)}$ (b) Evolution of financial balance with $r_l = \bar{r}_l$

Figure 3.14: P&V - Evolution of financial balance with the age of policies

In Figure 3.14a, we observe a similar behavior of the financial balance as for the BMS studied above, except for the bottom initial distribution. Indeed, with a bottom initial distribution, the portfolio is loss making in the first years but is slowly going up to reach 95.6%. In the top and uniform initial distributions, too many policyholders are in the malus levels to have a perfect equilibrium, resulting in a profit making situation. On the other hand, with the bottom initial distribution, too many policyholders are in the bonus levels, hence leading to a loss making situation.

In Figure 3.14b, using $\bar{r}_l$ as relativities shows that the BMS rapidly goes below the profitability threshold, as for the other BMS studied.

3.3 Linear relativities

For the same reasons as in the steady-state regime, one would want to have monotonic increasing relativities as we move up the levels in the BMS. We saw that the transient relativities developed in the previous section would not always follow this trend. A clever alternative is then to force the relativities in a linear form. We will follow the same reasoning as above with $\bar{r}_l^{linear} = a + bl$. We are then looking for (a, b) such that

$$(a, b) = \underset{a,b}{\operatorname{argmin}}\, \mathbb{E}[(\Lambda\Theta - \Lambda a - \Lambda b L_A)^2] \tag{3.17}$$

The first-order conditions are

$$\begin{aligned} &\frac{\partial}{\partial a}\mathbb{E}[(\Lambda\Theta - \Lambda a - \Lambda b L_A)^2] = 0 \\ \iff\ &\mathbb{E}\left[-2\Lambda^2\Theta + 2a\Lambda^2 + 2\Lambda^2 b L_A\right] = 0 \\ \iff\ &\mathbb{E}[\Lambda^2\Theta] - a\mathbb{E}[\Lambda^2] - b\mathbb{E}[\Lambda^2 L_A] = 0 \end{aligned} \tag{3.18}$$

and

$$\begin{aligned} &\frac{\partial}{\partial b}\mathbb{E}[(\Lambda\Theta - \Lambda a - \Lambda b L_A)^2] = 0 \\ \iff\ &\mathbb{E}\left[-2\Lambda^2(\Theta - a)L_A + 2\Lambda^2 b L_A^2\right] = 0 \\ \iff\ &\mathbb{E}[\Lambda^2\Theta L_A] - a\mathbb{E}[\Lambda^2 L_A] - b\mathbb{E}[\Lambda^2 L_A^2] = 0 \end{aligned} \tag{3.19}$$

Putting it together, we get to solve, for (a, b):

$$\begin{bmatrix} \mathbb{E}[\Lambda^2] & \mathbb{E}[\Lambda^2 L_A] \\ \mathbb{E}[\Lambda^2 L_A] & \mathbb{E}[\Lambda^2 L_A^2] \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \mathbb{E}[\Lambda^2\Theta] \\ \mathbb{E}[\Lambda^2\Theta L_A] \end{bmatrix}$$

$$\iff \begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\mathbb{E}[\Lambda^2]\mathbb{E}[\Lambda^2 L_A^2] - \mathbb{E}[\Lambda^2 L_A]^2} \begin{bmatrix} \mathbb{E}[\Lambda^2 L_A^2] & -\mathbb{E}[\Lambda^2 L_A] \\ -\mathbb{E}[\Lambda^2 L_A] & \mathbb{E}[\Lambda^2] \end{bmatrix} \begin{bmatrix} \mathbb{E}[\Lambda^2 \Theta] \\ \mathbb{E}[\Lambda^2 \Theta L_A] \end{bmatrix}$$

$$\iff \begin{cases} a = \dfrac{\mathbb{E}[\Lambda^2 L_A^2]\mathbb{E}[\Lambda^2\Theta] - \mathbb{E}[\Lambda^2 L_A]\mathbb{E}[\Lambda^2\Theta L_A]}{\mathbb{E}[\Lambda^2]\mathbb{E}[\Lambda^2 L_A^2] - \mathbb{E}[\Lambda^2 L_A]^2} \\ b = \dfrac{\mathbb{E}[\Lambda^2]\mathbb{E}[\Lambda^2\Theta L_A] - \mathbb{E}[\Lambda^2 L_A]\mathbb{E}[\Lambda^2\Theta]}{\mathbb{E}[\Lambda^2]\mathbb{E}[\Lambda^2 L_A^2] - \mathbb{E}[\Lambda^2 L_A]^2} \end{cases} \tag{3.20}$$

where the expectation operator is

$$\mathbb{E}[\cdot] = \sum_{n=1}^{\infty} a_n \sum_{l=1}^{s} \sum_{g=1}^{h} w_g \int_0^{+\infty} \cdot \quad p_l^{(n)}(\lambda_g\theta) f(\theta) d\theta$$

The optimal transient linear relativities are then obtained as

$$\bar{r}_l^{linear} = a + bl$$

with (a, b) computed as (3.20).

Conclusion

The objective set in the introduction is met. We formulated the problem in a formal way and then proposed multiple ways to address it: optimal stationary relativities (general and linear case) and optimal transient relativities, using two different *a posteriori* structures (Good-driver/Bad-driver and Gamma). Some of them were convincing, some other less appropriate.

In order to carry out this project, the first task was to really understand what we were looking for, do an inventory of BMS available on the Belgian market, look for a methodological approach to address the problem and investigate state-of-the-art. After that, a considerable amount of time was spent in implementing our framework in R to obtain meaningful results. In the end, given the specifics of most BMS found, the transient regime is the most suitable approach considered in this study. It allowed us to drill down the dynamics of the BMS encountered, even if their complexity makes them very slow to converge to their steady-state.

Some interesting questions were left open for further research. For example, since some BMS include a per-claim deductible, it would be interesting to develop the severity model to analyse the model suggested in the first chapter. Also, the linar relativities in transient regime (as presented in the last section) can be a good alternative to the general results, which sometimes exhibit undesirable behavior (e.g. non-monotonic evolution with levels).

Finally, we would like to insist on the fact that many assumptions were intentionally made throughout this thesis. While they were made on purpose and after due diligence, other assumptions would sometimes make perfect sense. For instance, the *a priori* frequency model used is basic as we didn't want to further elaborate this part. However, as the *a priori* expected claims frequency is an essential part of our optimality criteria, the relativities obtained are sensitive to this assumption. Furthermore, the assumption of independence between Λ and A is strong and the development of a dependence model could be an interesting way to enrich our analysis.

Appendix A

BMS modeling

Implementation of AG BMS characteristics in R

```
# Definition of AG BMS
BMS <- list(
  levels = -2:22,
  entry_level = 14,

  transition = function(l, k) {
    if (l == "-2") return("-2")  # Absorbing state
    if (k == 0)  return(as.character(max(-2, as.integer(l) - 1)))
    else         return(as.character(min(22, as.integer(l) + 4 + (
   k-1)*5)))
  },

  level_100 = "14",

  insurer_relat = c(rep(54,7)
   ,63,66,69,73,77,81,85,90,95,100,105,111,117,123,130,140,160,200)
   /100

)
```

Implementation of Baloise BMS characteristics in R

```
# Definition of Baloise BMS
# ==== BMS Levels ====

true_levels <- c(-2:22)

first_levels <- c(-2:15)

# Fictive levels
high_levels <- c(
  "16","16.3",
  "17", "17.2", "17.3",
```

```r
  paste0("18.", 0:3),
  paste0("19.", 0:3),
  paste0("20.", 0:2),
  paste0("21.", 0:1),
  "22"
)

# Full state space
all_levels <- c(as.character(first_levels), high_levels)

# Entry level
entry_level <- 11

# Transition function
transition <- function(level, k) {
  level <- as.character(level)

  if (level == "-2") return("-2")  # absorbing state

  # === From base levels ===
  if (level %in% as.character(-2:17)) {
    l <- as.integer(level)
    if (k == 0) {
      return(as.character(max(-2, l - 1)))
    } else {
      next_level <- min(22, l + 4 + (k-1)*5)
      if (next_level < 18 || next_level==22){return(as.character(
 next_level))}
      else{
        return(paste0(next_level,".",0))
      }
    }
  }

  # === From fictive states ===
  if (grepl("\\.", level)) {
    parts <- strsplit(level, "\\.")[[1]]
    j <- as.integer(parts[1])
    i <- as.integer(parts[2])

    if (k > 0) { # if at least one claim
      if (level=="16.3"){
        if (k==1){return("20.0")}
        else{return("22")}
        }
      else if (j=="17"){
        if (k==1){return("21.0")}
        else{return("22")}
        }
```

```r
      else{
        return("22")
      }
    } else { # if no claims
      if (i == 3) {
        return("14")  # reset after 4th year (i=3)
      } else {
        if (level=="18.0"){return("17")}
        next_level <- paste0(j - 1, ".", i + 1)
        return(next_level)
      }
    }
  }

  if (level == "22"){
    if (k==0) {return("21.1")}else{
      return("22")
    }
  }

  stop("Unknown state: ", level)
}


level_100 <- "14"

insurer_relat <- c(rep(54,5)
    ,57,60,63,66,69,73,77,81,85,90,95,100,105,111,117,123,130,140,160,200)
    /100


# ==== Export ====
BMS <- list(
  levels = all_levels,
  true_levels = true_levels,
  entry_level = as.character(entry_level),
  transition = transition,
  level_100 = level_100,
  insurer_relat = insurer_relat
)
```

Implementation of Ethias BMS characteristics in R

```r
# Definition of Ethias BMS
# ==== BMS Levels ====

true_levels <- c(0:22)

first_levels <- c(1:15)

```

```r
# Fictive levels
high_levels <- c(
  "16","16.3",
  "17", "17.2", "17.3",
  paste0("18.", 0:3),
  paste0("19.", 0:3),
  paste0("20.", 0:2),
  paste0("21.", 0:1),
  "22"
)

joker_levels <- c(
  paste0("0.", 5:0)
)

# Full state space
all_levels <- c(joker_levels,as.character(first_levels), high_levels)

# Entry level
entry_level <- 13

# Transition function
transition <- function(level, k) {
  level <- as.character(level)

  # === From base levels ===
  if (level %in% as.character(1:17)) {
    l <- as.integer(level)
    if (k == 0) {
      if (l==1){return("0.0")}
      else{return(as.character(max(0, l - 1)))}
    } else {
      next_level <- min(22, l + 4 + (k-1)*5)
      if (0<next_level && next_level < 18 || next_level==22){ return(as.character(next_level))}
      else{
        return(paste0(next_level,".",0))
      }
    }
  }

  # === From fictive states ===
  if (grepl("\\.", level)) {
    parts <- strsplit(level, "\\.")[[1]]
    j <- as.integer(parts[1])
    i <- as.integer(parts[2])

    if (k > 0) { # if at least one claim
```

```
      if (level=="16.3"){
        if (k==1){return("20.0")}
        else{return("22")}
      }
      else if (j==17){
        if (k==1){return("21.0")}
        else{return("22")}
      }
      else if (j==0){
        if (i==5 && k==1){return("0.0")}
        next_level <- min(j+4+(k-1)*5 - max(i-4,0)*5,22)
        if (next_level==19){return("19.0")}
        else{return(as.character(next_level))}
      }
      else{
        return("22")
      }
    } else { # if no claims
      if (i == 3 && j!=0) {
        return("14")  # reset after 4th year (i=3)
      }
      else if(j==0){
        if (i==5){return("0.5")}
        else{return(paste0("0.",i+1))}
      }
      else {
        if (level=="18.0"){return("17")}
        next_level <- paste0(j - 1, ".", i + 1)
        return(next_level)
      }
    }
  }

  if (level == "22"){
    if (k==0) {return("21.1")}else{
      return("22")
    }
  }

  stop("Unknown state: ", level)
}

level_100 <- "13"

insurer_relat <- c
   (45,47,49,54,60,66,69,72,77,81,85,89,95,100,105,116,122,128,135,143,153,175,2
   /100

# ==== Export ====
```

```r
BMS <- list(
  levels = all_levels,
  true_levels = true_levels,
  entry_level = as.character(entry_level),
  transition = transition,
  level_100 = level_100,
  insurer_relat = insurer_relat
)
```

Implementation of P&V BMS characteristics in R

```r
# Definition of P&V BMS

BMS <- list(
  levels = -2:22,
  entry_level = 11,

  transition = function(l, k) {
    if (k == 0)  return(as.character(max(-2, as.integer(l) - 1)))
    else         return(as.character(min(22, as.integer(l) + 4 + (
  k-1)*5)))
  },

  level_100 = "14",

  insurer_relat = c(rep(54,5)
    ,57,60,63,66,69,73,77,81,85,90,95,100,105,111,117,123,130,140,160,200)
    /100

)
```

Appendix B

A priori ratemaking

A priori ratemaking implementation in R.

```
# Data Preparation Script
# =======================

# ==== Configuration ====
SEED <- 42
set.seed(SEED)

# ==== Required Libraries ====
library(CASdatasets)
library(MASS)
library(dplyr)
library(caret)

# ==== Load Dataset ====
data("beMTPL97")
dataset <- beMTPL97

# ==== Step 1: Categorize Variables ====

# Categorize policyholder age
dataset$ageph_cat <- cut(dataset$ageph,
                         breaks = c(-Inf, 30, 50, Inf),
                         labels = c("young", "middle", "old"),
                         right = FALSE)

# Create evenly distributed bins (tertiles) based on quantiles
dataset$agec_cat <- cut(dataset$agec,
                        breaks = quantile(dataset$agec, probs =
    seq(0, 1, length.out = 4), na.rm = TRUE),
                        labels = c("new", "average", "old"),
                        include.lowest = TRUE)

```

```
dataset$power_cat <- cut(dataset$power,
                         breaks = quantile(dataset$power, probs = seq(0, 1, length.out = 4), na.rm = TRUE),
                         labels = c("low", "regular", "high"),
                         include.lowest = TRUE)

# ==== Step 2: Split Data ====
in_training <- createDataPartition(dataset$nclaims, p = 0.8, list = FALSE)
training_set <- dataset[in_training, ]
testing_set <- dataset[-in_training, ]

# ==== Step 3: Fit GLM (Poisson) using categorical variables ====
m0 <- glm(nclaims ~ ageph_cat + agec_cat + power_cat + offset(log(expo)),
          data = training_set,
          family = poisson(link = "log"))

# Predict lambda_hat on testing set
testing_set$lambda_hat <- predict(m0, newdata = testing_set, type = "response")

# ==== Step 4: Assign A Priori Risk Class ====

# Define a unique risk class per combination (3ÃŮ3ÃŮ3 = 27 levels)
testing_set$risk_class <- interaction(testing_set$ageph_cat,
                                      testing_set$agec_cat,
                                      testing_set$power_cat,
                                      sep = "_",
                                      drop = TRUE)

# Assign integer index to each risk class
testing_set$g <- as.integer(as.factor(testing_set$risk_class))

# ==== Step 5: Compute lambda_g for each risk class ====
lambda_g <- testing_set %>%
  group_by(g) %>%
  summarise(lambda = mean(lambda_hat), .groups = "drop") %>%
  arrange(g)

# Create a mapping from g to risk_class
risk_class_labels <- testing_set %>%
  select(g, risk_class) %>%
  distinct() %>%
  arrange(g)

# Merge with lambda_g to show readable labels
lambda_g_labeled <- left_join(lambda_g, risk_class_labels, by = "g")
```

```r

# Print with readable labels
print(lambda_g_labeled)

# Visualization of lambda_g's
risk_map <- testing_set %>%
  select(g, ageph_cat, agec_cat, power_cat) %>%
  distinct()

lambda_g_labeled <- lambda_g %>%
  left_join(risk_map, by = "g")

library(ggplot2)

ggplot(lambda_g_labeled, aes(x = power_cat, y = agec_cat, fill = lambda)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "white", high = "darkred") +
  facet_grid(. ~ ageph_cat, labeller = label_both) +
  labs(
    title = expression(paste("Heatmap of ",lambda[g]," across risk classes")), #Îż_g across Risk classes")),
    x = "Power Category", y = "Car Age Category",
    fill = expression(lambda[g])
  ) +
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))


# ==== Output ====
# The following objects are now available to use in other scripts:
# - testing_set: contains lambda_hat and g
# - training_set
# - lambda_g: mean lambda per risk class
```

Appendix C

Gamma model - Stationary setting (function integrate())

Implementation of stationary Gamma model, using `integrate()` R function

```
# Bonus-Malus System with Gamma Heterogeneity - Steady-state
# ==============================================

rm(list = ls())

# ==== Load BMS and A Priori Preparation ====
source("BMS1_AG.R")         # <-- switch BMS here
source("a_priori_2.R")    # <-- shared a priori classification

levels <- BMS$levels
entry_level <- BMS$entry_level
transition_BMS <- BMS$transition

# ==== Configuration ====
h <- nrow(lambda_g)  # Number of a priori risk classes
w_g <- as.numeric(prop.table(table(testing_set$g)))  # Empirical
    weights

# ==== Required Libraries ====
library(expm)

# ==== Step 1: Estimate Gamma Parameter (a_hat) ====
m_nb <- glm.nb(nclaims ~ ageph_cat + agec_cat + power_cat + offset
    (log(expo)), data = training_set)
a_hat <- 1 / m_nb$theta

# ==== Step 2: Transition Matrix ====
build_transition_matrix <- function(lambda_theta, max_k = 50) {
  level_names <- as.character(levels)
  P <- matrix(0, nrow = length(levels), ncol = length(levels),
```

```
                dimnames = list(from = level_names, to = level_names
    ))

  for (l1 in levels) {
    for (k in 0:max_k) {
      prob_k <- dpois(k, lambda_theta)
      l2 <- transition_BMS(l1, k)
      i <- as.character(l1)
      j <- as.character(l2)
      P[i, j] <- P[i, j] + prob_k
    }
  }
  if (any(abs(rowSums(P) - 1) > 1e-6)) {
    warning("Transition matrix rows do not sum to 1: increase max_
    k if needed.")
  }
  return(P)
}

# ==== Step 3: Stationary Distribution ====
pi <- function(lambda, theta) {
  P <- build_transition_matrix(lambda * theta)
  pi_vec <- (P %^% 100)[1, ]
  names(pi_vec) <- colnames(P)
  return(pi_vec)
}

# ==== Step 4: Marginal Distribution of BMS Level ====
gamma_density <- function(theta, a) dgamma(theta, shape = a, rate
    = a)

P_L <- function(l) {
  l_name <- as.character(l)
  sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    integrand <- function(theta_vec) {
      sapply(theta_vec, function(theta) {
        pi(lambda, theta)[l_name] * gamma_density(theta, a_hat)
      })
    }
    integrate(integrand, lower = 0, upper = Inf, rel.tol = 1e-6)$
    value * w_g[g]
  }))
}

P_L_vec <- sapply(levels, P_L)
names(P_L_vec) <- levels

# ==== Step 5: Expectations ====
```

```
E_L2Theta <- function(l) {
  l_name <- as.character(l)
  num <- sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    integrand <- function(theta_vec) {
      sapply(theta_vec, function(theta) {
        lambda^2 * theta * pi(lambda, theta)[l_name] * gamma_density(theta, a_hat)
      })
    }
    integrate(integrand, lower = 0, upper = Inf, rel.tol = 1e-6)$value * w_g[g]
  }))
  num / P_L_vec[l_name]
}

E_L2 <- function(l) {
  l_name <- as.character(l)
  num <- sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    integrand <- function(theta_vec) {
      sapply(theta_vec, function(theta) {
        lambda^2 * pi(lambda, theta)[l_name] * gamma_density(theta, a_hat)
      })
    }
    integrate(integrand, lower = 0, upper = Inf, rel.tol = 1e-6)$value * w_g[g]
  }))
  num / P_L_vec[l_name]
}

E_L2Theta_vec <- sapply(levels, E_L2Theta)
E_L2_vec <- sapply(levels, E_L2)
names(E_L2Theta_vec) <- names(E_L2_vec) <- levels

# ==== Step 6: Compute Optimal Relativities ====
num <- sum(E_L2Theta_vec * P_L_vec / E_L2_vec)
den <- sum(P_L_vec / (2 * E_L2_vec))
alpha <- (num - 1) / den

opt_relat <- E_L2Theta_vec / E_L2_vec - alpha / (2 * E_L2_vec)
names(opt_relat) <- levels

# ==== Step 7: Plot Relativities ====
plot(levels, opt_relat, type = "b", pch = 16, col = "darkgreen",
     xlab = "BMS Level", ylab = "Optimal Relativity",
     main = "Optimal Relativities under Gamma Heterogeneity")
abline(h = 1, col = "gray", lty = 2)
```

```r

# ==== Step 8: Scale Optimal Relativities ====
scaled_relat <- opt_relat / opt_relat[as.character(entry_level)]

plot(levels, scaled_relat, type = "b", pch = 16, col = "blue",
     xlab = "BMS Level", ylab = "Optimal Relativity",
     main = "Scaled Optimal Relativities under Gamma Heterogeneity
  ")
abline(h = 1, col = "gray", lty = 2)
```

Appendix D

Stationary setting - Good driver/Bad driver model and Gamma model (integration approximation scheme)

Implementation of stationary optimal relativities for both *a posteriori* models, using integration approximation scheme.

```
# ================================================================
# Compare Optimal Relativities: Gamma vs Good/Bad - Steady-state
# ================================================================

rm(list = ls())
library(expm)
library(MASS)

# ---- Load BMS + a_priori -------------------------
source("../BMS/BMS_Ethias.R")   # <-- pick BMS here
source("../a_priori_2.R")

levels          <- as.character(BMS$levels)
entry_level     <- as.character(BMS$entry_level)
transition_BMS  <- BMS$transition
level_100 <- BMS$level_100
insurer_relat <- BMS$insurer_relat

# Base levels (handle fictive/non-fictive)
if (!is.null(BMS$true_levels)) {
  true_levels <- as.character(BMS$true_levels)
} else {
  true_levels <- as.character(levels)
```

```
}

# ---- Portfolio weights ---------------------------
h   <- nrow(lambda_g)
w_g <- as.numeric(prop.table(table(testing_set$g)))


# ==== Transition matrix P(ÎżÎÿ) ====
build_transition_matrix <- function(lambda_theta, max_k = 50) {
  lev <- as.character(BMS$levels)
  P <- matrix(0, length(lev), length(lev), dimnames = list(from = lev, to = lev))
  for (l1 in lev) {
    for (k in 0:max_k) {
      pk <- dpois(k, lambda_theta)
      j  <- BMS$transition(l1, k)
      P[l1, j] <- P[l1, j] + pk
    }
    # put the truncated tail mass onto the "largest" jump (k = max_k + 1)
    tail <- 1 - sum(dpois(0:max_k, lambda_theta))
    if (tail > 0) {
      j_tail <- BMS$transition(l1, max_k + 1)
      P[l1, j_tail] <- P[l1, j_tail] + tail
    }
  }
  stopifnot(all(abs(rowSums(P) - 1) < 1e-12))
  P
}

pi_stationary <- function(lambda_theta) {
  P <- build_transition_matrix(lambda_theta)
  out <- (P %^% 100)[1, ]
  names(out) <- colnames(P)
  out
}

aggregate_dist <- function(vec_named) {
  # vec_named is a named vector over full state space (including fictive)
  if (is.null(names(vec_named))) stop("aggregate_dist: vector must have names.")
  base <- sub("\\..*$", "", names(vec_named))
  agg  <- tapply(vec_named, base, sum)
  out  <- setNames(numeric(length(true_levels)), true_levels)
  out[names(agg)] <- agg
  out
}

```

```
# ================================================================
# 1) Gamma model
# ================================================================
message("== Gamma model ==")

# Estimate a_hat from NB GLM
m_nb  <- glm.nb(nclaims ~ ageph_cat + agec_cat + power_cat + offset(log(expo)),
                data = training_set)
a_hat <- 1 / m_nb$theta

# Theta grid + weights
theta_grid    <- seq(0.02, 15, length.out = 750)
theta_weights <- dgamma(theta_grid, shape = a_hat, rate = a_hat)
theta_weights <- theta_weights / sum(theta_weights)

# Precompute aggregated stationary distributions for all (g, theta)
# pi_map[i, g, l] = aggregated pi at base level l for g-th risk class and theta_grid[i]
pi_map <- array(NA_real_, dim = c(length(theta_grid), h, length(true_levels)),
                dimnames = list(theta = NULL, g = NULL, l = true_levels))

for (g in 1:h) {
  lambda <- lambda_g$lambda[g]
  message(sprintf("  Gamma: precomputing pi_map for g=%d / %d", g, h))
  for (i in seq_along(theta_grid)) {
    pi_full <- pi_stationary(lambda * theta_grid[i])
    pi_map[i, g, ] <- aggregate_dist(pi_full)
  }
}

# P[L = l]
P_L_gamma <- sapply(true_levels, function(l_name) {
  sum(sapply(1:h, function(g) {
    w_g[g] * sum(theta_weights * pi_map[, g, l_name])
  }))
})
names(P_L_gamma) <- true_levels

# E[Lambda^2 * Theta | L=l] / E[Lambda^2 | L=l] pieces
E_L2Theta_gamma <- sapply(true_levels, function(l_name) {
  num <- sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    w_g[g] * sum(lambda^2 * theta_grid * theta_weights * pi_map[, g, l_name])
```

```r
  }))
  num / P_L_gamma[l_name]
})

E_L2_gamma <- sapply(true_levels, function(l_name) {
  num <- sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    w_g[g] * sum(lambda^2 * theta_weights * pi_map[, g, l_name])
  }))
  num / P_L_gamma[l_name]
})

# Lagrange term alpha and optimal relativities
num_gamma <- sum(E_L2Theta_gamma * P_L_gamma / E_L2_gamma)
den_gamma <- sum(P_L_gamma / (2 * E_L2_gamma))
alpha_gamma <- (num_gamma - 1) / den_gamma

opt_relat_gamma <- E_L2Theta_gamma / E_L2_gamma - alpha_gamma / (2
    * E_L2_gamma)
names(opt_relat_gamma) <- true_levels

# Scaled
scaled_gamma <- opt_relat_gamma / opt_relat_gamma[level_100]

# ==================================================================
# 2) Good/Bad model
# ==================================================================
message("== Good/Bad model ==")

theta_vals <- c(0.8, 1.3)  # good, bad
g_weights  <- c(0.6, 0.4)  # P(good), P(bad)

# Precompute aggregated stationary distributions for (g, good/bad)
# gb_map[m, g, l] where m=1 (good), m=2 (bad)
gb_map <- array(NA_real_, dim = c(2, h, length(true_levels)),
                dimnames = list(driver = c("good","bad"), g = NULL
    , l = true_levels))

for (g in 1:h) {
  lambda <- lambda_g$lambda[g]
  message(sprintf("  Good/Bad: precomputing for g=%d / %d", g, h))
  for (m in 1:2) {
    pi_full <- pi_stationary(lambda * theta_vals[m])
    gb_map[m, g, ] <- aggregate_dist(pi_full)
  }
}

# P[L=l]
P_L_gb <- sapply(true_levels, function(l_name) {
```

```
  sum(sapply(1:h, function(g) {
    w_g[g] * ( g_weights[1] * gb_map[1, g, l_name] +
                g_weights[2] * gb_map[2, g, l_name] )
  }))
})
names(P_L_gb) <- true_levels

# Expectations
E_L2Theta_gb <- sapply(true_levels, function(l_name) {
  num <- sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    w_g[g] * lambda^2 * ( g_weights[1] * theta_vals[1] * gb_map[1, g, l_name] +
                          g_weights[2] * theta_vals[2] * gb_map[2, g, l_name] )
  }))
  num / P_L_gb[l_name]
})

E_L2_gb <- sapply(true_levels, function(l_name) {
  num <- sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    w_g[g] * lambda^2 * ( g_weights[1] * gb_map[1, g, l_name] +
                          g_weights[2] * gb_map[2, g, l_name] )
  }))
  num / P_L_gb[l_name]
})

num_gb <- sum(E_L2Theta_gb * P_L_gb / E_L2_gb)
den_gb <- sum(P_L_gb / (2 * E_L2_gb))
alpha_gb <- (num_gb - 1) / den_gb

opt_relat_gb <- E_L2Theta_gb / E_L2_gb - alpha_gb / (2 * E_L2_gb)
names(opt_relat_gb) <- true_levels

# Scaled
scaled_gb <- opt_relat_gb / opt_relat_gb[level_100]

# ================================================================
# Plot both on the same graph
# ================================================================

# Unscaled
plot(as.numeric(true_levels), opt_relat_gamma, type = "l", lwd = 2, col = "darkgreen",
     xlab = "BMS Level", ylab = "Optimal relativity",
     main = "Optimal Relativities (Unscaled)")
lines(as.numeric(true_levels), opt_relat_gb, lwd = 2, col = "steelblue", lty = 2)
```

```
abline(h = 1, col = "gray", lty = 3)
legend("topleft",
       legend = c("Gamma", "Good/Bad"),
       col = c("darkgreen", "steelblue"), lty = c(1,2), lwd = 2,
    bty = "n")

# Scaled
plot(as.numeric(true_levels), scaled_gamma, type = "l", lwd = 2,
    col = "darkgreen",
     xlab = "BMS Level", ylab = "Scaled relativity",
     main = paste0("Optimal relativities scaled at level 100% = ",
     level_100))
lines(as.numeric(true_levels), scaled_gb, lwd = 2, col = "
    steelblue", lty = 2)
lines(as.numeric(true_levels), insurer_relat, lwd = 2, col = "red"
    , lty = 3)  # <-- added line
abline(h = 1, col = "gray", lty = 3)
legend("topleft",
       legend = c("Gamma (scaled)", "Good/Bad (scaled)", "Insurer"
    ),
       col = c("darkgreen", "steelblue", "red"),
       lty = c(1, 2, 3), lwd = 2, bty = "n")
```

Appendix E

Stationary setting - Linear form - Good driver/Bad driver model and Gamma model (integration approximation scheme)

Implementation of stationary linear optimal relativities for both *a posteriori* models, using integration approximation scheme.

```
# ===============================================================
# Compare Optimal Linear Relativities: Gamma vs Good/Bad - Steady-
    state
# ===============================================================

rm(list = ls())
library(expm)

# ==== Load BMS and A Priori Preparation ====
source("../BMS/BMS_Ethias.R")   # <-- switch BMS here
source("../a_priori_2.R")

# Imports
levels_raw     <- BMS$levels
levels         <- as.character(levels_raw)
entry_level    <- as.character(BMS$entry_level)
transition_BMS  <- BMS$transition
level_100 <- BMS$level_100
insurer_relat <- BMS$insurer_relat

# Handle both with or without fictive levels
if (!is.null(BMS$true_levels)) {
  true_levels <- as.character(BMS$true_levels)
```

```r
} else {
  true_levels <- as.character(levels)
}

# ==== Portfolio weights ====
h   <- nrow(lambda_g)
w_g <- as.numeric(prop.table(table(testing_set$g)))  # empirical weights (a priori classes)

# Good/Bad model params
theta_vals <- c(0.8, 1.3)     # good, bad
g_weights  <- c(0.6, 0.4)

# ==== Estimate Gamma parameter (a_hat) ====
m_nb  <- glm.nb(nclaims ~ ageph_cat + agec_cat + power_cat + offset(log(expo)),
                data = training_set)
a_hat <- 1 / m_nb$theta

# ==== Îÿ-grid for Gamma  ====
theta_grid    <- seq(0.02, 15, length.out = 750)
theta_weights <- dgamma(theta_grid, shape = a_hat, rate = a_hat)
theta_weights <- theta_weights / sum(theta_weights)

# ==== Transition matrix P(ÎżÎÿ) ====
build_transition_matrix <- function(lambda_theta, max_k = 50) {
  lev <- as.character(BMS$levels)
  P <- matrix(0, length(lev), length(lev), dimnames = list(from = lev, to = lev))
  for (l1 in lev) {
    for (k in 0:max_k) {
      pk <- dpois(k, lambda_theta)
      j  <- BMS$transition(l1, k)
      P[l1, j] <- P[l1, j] + pk
    }
    # put the truncated tail mass onto the "largest" jump (k = max_k + 1)
    tail <- 1 - sum(dpois(0:max_k, lambda_theta))
    if (tail > 0) {
      j_tail <- BMS$transition(l1, max_k + 1)
      P[l1, j_tail] <- P[l1, j_tail] + tail
    }
  }
  stopifnot(all(abs(rowSums(P) - 1) < 1e-12))
  P
}

pi_full <- function(lambda, theta) {
  P <- build_transition_matrix(lambda * theta)
```

```
  (P %^% 100)[1, ]
}

aggregate_to_true <- function(vec) {
  # vec: named over 'levels' (full state space)
  # returns length true_levels vector aggregated by base level
  agg <- numeric(length(true_levels))
  names(agg) <- as.character(true_levels)
  names(vec) <- levels
  for (j in names(agg)) {
    idx <- grep(paste0("^", j, "(\\.|$)"), names(vec), value = TRUE)
    agg[j] <- sum(vec[idx], na.rm = TRUE)
  }
  agg
}

# Precompute dot products helpers over true_levels
L  <- as.numeric(true_levels)
L1 <- L                                 # l
L2 <- L^2                                # l^2
names(L1) <- names(L2) <- as.character(true_levels)

# ====================================================================
#                           GAMMA (linear)
# ====================================================================

# Precompute aggregated stationary distributions for Gamma grid
# pi_gamma_map[theta_index, g, base_level]
pi_gamma_map <- array(NA_real_,
                      dim = c(length(theta_grid), h, length(true_levels)),
                      dimnames = list(theta = NULL,
                                      g = paste0("g", 1:h),
                                      l = as.character(true_levels)))

message("Precomputing stationary distributions for Gamma grid...")
for (g in 1:h) {
  lambda <- lambda_g$lambda[g]
  message(sprintf("  Gamma: precomputing pi_map for g=%d / %d", g, h))
  for (i in seq_along(theta_grid)) {
    pi_gamma_map[i, g, ] <- aggregate_to_true(pi_full(lambda, theta_grid[i]))
```

```
  }
}

# Compute A..F for Gamma (using moment sums to avoid looping over l)
A_g <- B_g <- C_g <- D_g <- E_g <- F_g <- 0
for (g in 1:h) {
  lambda <- lambda_g$lambda[g]
  wg     <- w_g[g]
  for (i in seq_along(theta_grid)) {
    p  <- pi_gamma_map[i, g, ]               # vector over true_levels
    wt <- theta_weights[i]
    m0 <- sum(p)                             # = 1
    m1 <- sum(L1 * p)
    m2 <- sum(L2 * p)
    th <- theta_grid[i]

    A_g <- A_g + wg * wt * lambda^2 * m0
    B_g <- B_g + wg * wt * lambda^2 * th * m0
    C_g <- C_g + wg * wt * lambda^2 * m1
    D_g <- D_g + wg * wt * lambda^2 * m2
    E_g <- E_g + wg * wt * lambda^2 * th * m1
    F_g <- F_g + wg * wt * m1
  }
}

# Solve (Gamma linear)
gamma_num_g <- (A_g - C_g^2 / D_g) * (D_g - C_g^2 / A_g) -
  (A_g - C_g^2 / D_g) * (E_g * F_g - B_g * C_g * F_g / A_g) -
  (D_g - C_g^2 / A_g) * (B_g - E_g * C_g / D_g)
gamma_den_g <- (A_g - C_g^2 / D_g) * (F_g^2 / 2 - C_g * F_g / (2 * A_g)) +
  (D_g - C_g^2 / A_g) * (1 / 2 - C_g * F_g / (2 * D_g))
gamma_g <- gamma_num_g / gamma_den_g

a_g <- (B_g - C_g * (E_g + gamma_g / 2 * F_g) / D_g + gamma_g / 2) / (A_g - C_g^2 / D_g)
b_g <- (E_g - C_g * (B_g + gamma_g / 2) / A_g + gamma_g / 2 * F_g) / (D_g - C_g^2 / A_g)

opt_linear_relat_gamma <- a_g + b_g * L
names(opt_linear_relat_gamma) <- as.character(true_levels)

# ====================================================================
#                          GOOD/BAD (linear)
```

```
# =====================================================================

# Precompute aggregated stationary distributions for Good/Bad
pi_gb_map <- array(NA_real_,
                   dim = c(2, h, length(true_levels)),
                   dimnames = list(theta_idx = c("good","bad"),
                                   g = paste0("g", 1:h),
                                   l = as.character(true_levels)))

message("Precomputing stationary distributions for Good/Bad...")
for (g in 1:h) {
  lambda <- lambda_g$lambda[g]
  message(sprintf("  Good/Bad: precomputing for g=%d / %d", g, h))
  pi_gb_map["good", g, ] <- aggregate_to_true(pi_full(lambda, theta_vals[1]))
  pi_gb_map["bad",  g, ] <- aggregate_to_true(pi_full(lambda, theta_vals[2]))
}

# Compute A..F for Good/Bad (linear)
A_b <- B_b <- C_b <- D_b <- E_b <- F_b <- 0
for (g in 1:h) {
  lambda <- lambda_g$lambda[g]
  wg     <- w_g[g]

  for (j in 1:2) {
    th  <- theta_vals[j]
    pg  <- pi_gb_map[j, g, ]                   # vector over true_levels
    wgb <- g_weights[j]
    m0  <- sum(pg)
    m1  <- sum(L1 * pg)
    m2  <- sum(L2 * pg)

    A_b <- A_b + wg * wgb * lambda^2 * m0
    B_b <- B_b + wg * wgb * lambda^2 * th * m0
    C_b <- C_b + wg * wgb * lambda^2 * m1
    D_b <- D_b + wg * wgb * lambda^2 * m2
    E_b <- E_b + wg * wgb * lambda^2 * th * m1
    F_b <- F_b + wg * wgb * m1
  }
}

# Solve (Good/Bad linear)
gamma_num_b <- (A_b - C_b^2 / D_b) * (D_b - C_b^2 / A_b) -
  (A_b - C_b^2 / D_b) * (E_b * F_b - B_b * C_b * F_b / A_b) -
  (D_b - C_b^2 / A_b) * (B_b - E_b * C_b / D_b)
```

```r
gamma_den_b <- (A_b - C_b^2 / D_b) * (F_b^2 / 2 - C_b * F_b / (2 * A_b)) +
  (D_b - C_b^2 / A_b) * (1 / 2 - C_b * F_b / (2 * D_b))
gamma_b <- gamma_num_b / gamma_den_b

a_b <- (B_b - C_b * (E_b + gamma_b / 2 * F_b) / D_b + gamma_b / 2) / (A_b - C_b^2 / D_b)
b_b <- (E_b - C_b * (B_b + gamma_b / 2) / A_b + gamma_b / 2 * F_b) / (D_b - C_b^2 / A_b)

opt_linear_relat_gb <- a_b + b_b * L
names(opt_linear_relat_gb) <- as.character(true_levels)

# ======================================================================
#                              PLOTS
# ======================================================================

# Unscaled
plot(as.numeric(true_levels), opt_linear_relat_gamma, type = "l", lwd = 2, col = "darkgreen",
     xlab = "BMS Level", ylab = "Optimal relativity",
     main = "Optimal Linear Relativities (Unscaled)")
lines(as.numeric(true_levels), opt_linear_relat_gb, lwd = 2, col = "steelblue", lty = 2)
abline(h = 1, col = "gray", lty = 3)
legend("topleft",
       legend = c("Gamma", "Good/Bad"),
       col = c("darkgreen", "steelblue"), lty = c(1,2), lwd = 2, bty = "n")

# --- Scaled ---
scaled_gamma_linear <- opt_linear_relat_gamma / opt_linear_relat_gamma[as.character(level_100)]
scaled_gb_linear    <- opt_linear_relat_gb    / opt_linear_relat_gb[as.character(level_100)]

plot(as.numeric(true_levels), scaled_gamma_linear, type = "l", lwd = 2, col = "darkgreen",
     xlab = "BMS Level", ylab = "Scaled relativity",
     main = paste0("Linear Relativities Scaled at level 100% = ", level_100))
lines(as.numeric(true_levels), scaled_gb_linear, lwd = 2, col = "steelblue", lty = 2)
lines(as.numeric(true_levels), insurer_relat, lwd = 2, col = "red", lty = 3)  # <-- added line
```

```r
abline(h = 1, col = "gray", lty = 3)
legend("topleft",
       legend = c("Gamma (scaled)", "Good/Bad (scaled)", "Insurer"
   ),
       col = c("darkgreen", "steelblue", "red"),
       lty = c(1, 2, 3), lwd = 2, bty = "n")
```

Appendix F

Convergence analysis

Implementation of the convergence analysis in R.

```
# === Convergence analysis: Gamma & Good/Bad ===
rm(list = ls())
library(expm)
library(MASS)

# --------------------------------------------------------------------
# Load BMS and a priori classification
source("./BMS/BMS_Ethias.R")
source("a_priori_2.R")

# --- Pull BMS pieces ---
levels_raw     <- BMS$levels
levels         <- as.character(levels_raw)
entry_level    <- as.character(BMS$entry_level)
transition_BMS <- BMS$transition

# Handle fictive states
if (!is.null(BMS$true_levels)) {
  true_levels <- as.character(BMS$true_levels)
} else {
  true_levels <- as.character(levels)
}

# === Shared config ===
h      <- nrow(lambda_g)
w_g    <- as.numeric(prop.table(table(testing_set$g)))
n_max  <- 50
max_k  <- 100

# ==== Transition matrix P(ÎżÎÿ) ====
```

```r
build_transition_matrix <- function(lambda_theta, max_k = 50) {
  lev <- as.character(BMS$levels)
  P <- matrix(0, length(lev), length(lev), dimnames = list(from = lev, to = lev))
  for (l1 in lev) {
    for (k in 0:max_k) {
      pk <- dpois(k, lambda_theta)
      j  <- BMS$transition(l1, k)
      P[l1, j] <- P[l1, j] + pk
    }
    # put the truncated tail mass onto the "largest" jump (k = max_k + 1)
    tail <- 1 - sum(dpois(0:max_k, lambda_theta))
    if (tail > 0) {
      j_tail <- BMS$transition(l1, max_k + 1)
      P[l1, j_tail] <- P[l1, j_tail] + tail
    }
  }
  stopifnot(all(abs(rowSums(P) - 1) < 1e-12))
  P
}

# === Aggregate distribution to base levels ===
aggregate_dist <- function(vec) {
  if (is.null(names(vec))) {
    stop("Vector passed to aggregate_dist() must have names.")
  }
  base <- sub("\\..*$", "", names(vec))
  agg  <- tapply(vec, base, sum)
  out  <- setNames(numeric(length(true_levels)), true_levels)
  out[names(agg)] <- agg
  out
}

# === Function to compute C_n for a given theta grid & weights ===
compute_Cn <- function(theta_grid, w_theta) {
  # Precompute P and stationary dist
  P_list  <- list()
  pi_list <- list()
  for (g in 1:h) {
    cat("Precomputing for risk class", g, "of", h, "...\n")
    lambda <- lambda_g$lambda[g]
    for (i in seq_along(theta_grid)) {
      theta <- theta_grid[i]
      key <- paste0("g", g, "_i", i)
      P <- build_transition_matrix(lambda * theta)
      P_list[[key]]  <- P
      pi_list[[key]] <- aggregate_dist((P %^% 100)[1, ])
    }
```

```r
  }

  # Initial distribution
  L0 <- rep(1 / length(levels), length(levels))
  names(L0) <- levels

  # Loop over n
  C_vec <- numeric(n_max)
  for (n in 1:n_max) {
    total_diff <- setNames(numeric(length(true_levels)), true_
levels)
    for (g in 1:h) {
      wg <- w_g[g]
      for (i in seq_along(theta_grid)) {
        wt  <- w_theta[i]
        key <- paste0("g", g, "_i", i)
        Pn   <- P_list[[key]] %^% n
        pn   <- aggregate_dist((L0 %*% Pn)[1, ])  # keep names!
        pinf <- pi_list[[key]]
        total_diff <- total_diff + wg * wt * (pn - pinf)
      }
    }
    C_vec[n] <- sum(abs(total_diff))
    cat("n =", n, "| C_n =", C_vec[n], "\n")
  }
  C_vec
}

# === 1) Gamma model ===
m_nb  <- glm.nb(nclaims ~ ageph_cat + agec_cat + power_cat +
    offset(log(expo)),
                      data = training_set)
a_hat <- 1 / m_nb$theta
theta_grid_gamma <- seq(0.02, 15, length.out = 750)
w_theta_gamma    <- dgamma(theta_grid_gamma, shape = a_hat, rate =
    a_hat)
w_theta_gamma    <- w_theta_gamma / sum(w_theta_gamma)
C_gamma <- compute_Cn(theta_grid_gamma, w_theta_gamma)

# === 2) Good/Bad model ===
theta_grid_gb <- c(0.8, 1.3)
w_theta_gb    <- c(0.6, 0.4)
C_goodbad <- compute_Cn(theta_grid_gb, w_theta_gb)

# === Plot both ===
plot(1:n_max, C_gamma, type = "l", col = "darkred", lwd = 2,
     xlab = "n (Years)", ylab = expression(C[n]),
     main = "Convergence to Stationary Distribution")
lines(1:n_max, C_goodbad, col = "steelblue", lwd = 2, lty = 2)
```

```r
abline(h = 0.0, col = "gray", lty = 3)
legend("topright",
       legend = c("Gamma model", "Good/Bad model"),
       col = c("darkred", "steelblue"),
       lty = c(1, 2), lwd = 2)
```

Appendix G

Transient setting - Good-driver/Bad-driver model

Implementation of the Good-driver/Bad-driver model for transient optimal relativities.

```
# Good-driver/Bad-driver model âĂŤ Transient conditional &
    unconditional relativities
#
    ======================================================================

rm(list = ls())

# ==== Load BMS Definition ====
source("../BMS/BMS_AG.R")   # <--- switch BMS here
source("../a_priori_2.R")

levels          <- BMS$levels

# Base levels (handle fictive/non-fictive)
if (!is.null(BMS$true_levels)) {
  true_levels <- as.character(BMS$true_levels)
} else {
  true_levels <- as.character(levels)
}

entry_level    <- BMS$entry_level
transition_BMS  <- BMS$transition
level_100 <- BMS$level_100
insurer_relat <- BMS$insurer_relat
names(insurer_relat) <- true_levels
lvl_names <- as.character(true_levels)

```

```r
# ==== Configuration Parameters ====
theta_vals <- c(0.8, 1.3)
g_weights  <- c(0.6, 0.4)
h          <- nrow(lambda_g)
w_g        <- as.numeric(prop.table(table(testing_set$g)))

# Time horizon for the transient behaviour
n <- 50

# ==== Initial Distribution L0 ====
L0_uniform <- rep(1 / length(levels), length(levels))
concentrate_level <- 1 - (length(levels)-1)/100
L0_top <- c(rep(0.01, length(levels)-1),concentrate_level)
L0_bot <- c(concentrate_level ,rep(0.01, length(levels)-1))

L0 <- L0_uniform          # <---  change type of initial distribution here
names(L0) <- as.character(levels)

# ==== Distribution of A (weights over horizons) ====
An_uniform <- rep(1 / n, n)   # uniform weights over n=1..n
An_young <- c(rep(0.1,2),rep(0.05,5),rep(0.025,10),rep(0.3/(n-17),n-17))
An_old <- rev(An_young)

An <- An_young

# ==== Required Libraries ====
library(expm)


# ==== Transition matrix P(ÎżÎÿ) ====
build_transition_matrix <- function(lambda_theta, max_k = 50) {
  lev <- as.character(BMS$levels)
  P <- matrix(0, length(lev), length(lev), dimnames = list(from = lev, to = lev))
  for (l1 in lev) {
    for (k in 0:max_k) {
      pk <- dpois(k, lambda_theta)
      j  <- BMS$transition(l1, k)
      P[l1, j] <- P[l1, j] + pk
    }
    # put the truncated tail mass onto the "largest" jump (k = max_k + 1)
    tail <- 1 - sum(dpois(0:max_k, lambda_theta))
    if (tail > 0) {
      j_tail <- BMS$transition(l1, max_k + 1)
      P[l1, j_tail] <- P[l1, j_tail] + tail
    }
```

```r
  }
  stopifnot(all(abs(rowSums(P) - 1) < 1e-12))
  P
}

# ==== Step 2: Stationary Distribution (not used below, kept for
    completeness) ====
pi <- function(lambda, theta) {
  P <- build_transition_matrix(lambda * theta)
  pi_vec <- (P %^% 100)[1, ]
  names(pi_vec) <- colnames(P)
  pi_vec
}

# ==== Step 3: n-step distribution p_n(lambda, theta, n, L0) ====
p_n <- function(lambda, theta, n, L0) {
  if (length(L0) != length(levels)) stop("L0 must have one entry
    per BMS level.")
  if (abs(sum(L0) - 1) > 1e-8)       stop("L0 must sum to 1 (
    probability vector).")
  if (n < 0 || n %% 1 != 0)          stop("n must be a non-negative
     integer.")

  P   <- build_transition_matrix(lambda * theta)
  Pn  <- P %^% n
  dist_n <- as.numeric(L0 %*% Pn)
  names(dist_n) <- as.character(levels)
  dist_n
}

# ==== Step 4: Aggregate distribution over true BMS levels ====
aggregate_dist <- function(p_vec) {
  out <- numeric(length(true_levels))
  names(out) <- as.character(true_levels)

  state_names <- names(p_vec)
  for (j in names(out)) {
    matching <- grep(paste0("^", j, "(\\.|$)"), state_names, value
    = TRUE)
    out[j] <- sum(p_vec[matching], na.rm = TRUE)
  }
  out
}

# ==== Step 5: P[L(n) = l] at a given n (kept for reference/plots)
    ====
P_Ln <- function(l, n) {
  l_name <- as.character(l)
  sum(sapply(1:h, function(g) {
```

```
    lambda <- lambda_g$lambda[g]
    wg     <- w_g[g]
    good   <- aggregate_dist(p_n(lambda, theta_vals[1], n, L0))[l_name] * g_weights[1]
    bad    <- aggregate_dist(p_n(lambda, theta_vals[2], n, L0))[l_name] * g_weights[2]
    wg * (good + bad)
  }))
}

P_Ln_vec <- sapply(true_levels, function(l) P_Ln(l, n))
names(P_Ln_vec) <- true_levels

# ==== Step 6: Conditional relativities at horizon n (r_l^(n)) ====
# (Refactor to accept explicit n)
Num_cond_at <- function(l, n) {
  l_name <- as.character(l)
  sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    w_g[g] * lambda^2 * (
      theta_vals[1] * aggregate_dist(p_n(lambda, theta_vals[1], n, L0))[l_name] * g_weights[1] +
        theta_vals[2] * aggregate_dist(p_n(lambda, theta_vals[2], n, L0))[l_name] * g_weights[2]
    )
  }))
}

Den_cond_at <- function(l, n) {
  l_name <- as.character(l)
  sum(sapply(1:h, function(g) {
    lambda <- lambda_g$lambda[g]
    w_g[g] * lambda^2 * (
      aggregate_dist(p_n(lambda, theta_vals[1], n, L0))[l_name] * g_weights[1] +
        aggregate_dist(p_n(lambda, theta_vals[2], n, L0))[l_name] * g_weights[2]
    )
  }))
}

# r^(n)
num_vec <- sapply(true_levels, function(l) Num_cond_at(l, n))
den_vec <- sapply(true_levels, function(l) Den_cond_at(l, n))
names(num_vec) <- names(den_vec) <- true_levels

cond_relat <- num_vec / den_vec
names(cond_relat) <- true_levels
```

```
# ==== Step 7: Unconditional relativities (time-averaged with A_n) ====
N <- length(An)

num_sum <- setNames(numeric(length(lvl_names)), lvl_names)
den_sum <- setNames(numeric(length(lvl_names)), lvl_names)

for (t in seq_len(N)) {
  message(sprintf("  Computing unconditional relativities for n=%d / %d", t, N))
  num_sum <- num_sum + An[t] * sapply(lvl_names, function(l) Num_cond_at(l, t))
  den_sum <- den_sum + An[t] * sapply(lvl_names, function(l) Den_cond_at(l, t))
}

r_bar <- num_sum / den_sum
names(r_bar) <- lvl_names

# ==== Step 8: Check financial balance ====
In <- sum(cond_relat*P_Ln_vec)
I_bar <- sum(r_bar*P_Ln_vec)

# Scaling
scaled_cond <- cond_relat / cond_relat[level_100]
scaled_r_bar <- r_bar / r_bar[level_100]

# ==== Plot: Conditional, Unconditional, and Insurer Relativities ====
plot(true_levels, scaled_cond, type = "l", pch = 16, col = "blue", lwd = 2,
     xlab = "BMS Level", ylab = "Relativity",
     main = "Relativities: Conditional vs Unconditional vs Insurer",
     ylim = range(c(cond_relat, r_bar, insurer_relat)))

lines(true_levels, scaled_r_bar, type = "l", pch = 17, col = "darkgreen", lwd = 2)
lines(true_levels, insurer_relat, type = "l", pch = 15, col = "red", lwd = 2)

abline(h = 1, col = "gray", lty = 2)

legend("topleft",
       legend = c("Conditional (r^(n))", "Unconditional (r_bar)", "Insurer Relativity"),
       col = c("blue", "darkgreen", "red"),
       pch = c(16, 17, 15), lwd = 2)
```

Appendix H

Transient setting - Gamma model

Implementation of the Gamma model for transient optimal relativities.

```
# Gamma model âĂŤ Transient conditional & unconditional relativities
# ======================================================================

rm(list = ls())

# ==== Load BMS & data prep ====
source("../BMS/BMS_PV.R")
source("../a_priori_2.R")

levels          <- BMS$levels
true_levels     <- if (!is.null(BMS$true_levels)) as.character(BMS$true_levels) else as.character(levels)
entry_level     <- BMS$entry_level
transition_BMS  <- BMS$transition
level_100       <- as.character(BMS$level_100)
insurer_relat   <- BMS$insurer_relat; names(insurer_relat) <- true_levels

# ==== User options ====
N        <- 50           # horizon for transient computations; also length of A_n
n_set    <- c(10,20,30,40,50)   # conditional curves to draw
cols     <- c("dodgerblue","darkorange","darkgreen","purple","red")

# Îÿ-grid settings
theta_min <- 0.01
theta_max <- 12
I_theta   <- 750
max_k     <- 75
```

```r

# ==== Libraries ====
library(expm)

# ==== A-priori classes & weights ====
h   <- nrow(lambda_g)
w_g <- as.numeric(prop.table(table(testing_set$g)))

# ==== Estimate Gamma shape ====
m_nb  <- glm.nb(nclaims ~ ageph_cat + agec_cat + power_cat + offset(log(expo)), data = training_set)
a_hat <- 1 / m_nb$theta

# ==== Îÿ-grid and weights ====
theta_grid    <- seq(theta_min, theta_max, length.out = I_theta)
theta_weights <- dgamma(theta_grid, shape = a_hat, rate = a_hat)
theta_weights <- theta_weights / sum(theta_weights)

# ==== Builders: L0 & A_n ====
build_L0 <- function(type = c("uniform","bot","top")) {
  if (type == "uniform") {
    L0 <- rep(1 / length(levels), length(levels))
  } else if (type == "bot") {
    concentrate_level <- 1 - (length(levels)-1)/100
    L0 <- c(concentrate_level ,rep(0.01, length(levels)-1))
  } else if (type == "top") {
    concentrate_level <- 1 - (length(levels)-1)/100
    L0 <- c(rep(0.01, length(levels)-1),concentrate_level)
  }
  L0
}

build_An <- function(N, type = c("uniform","young","old")){
  if (type == "uniform") {
    An <- rep(1 / N, N)
  } else if (type == "young") {
    An <- c(rep(0.1,2),rep(0.05,5),rep(0.025,10),rep(0.3/(N-17),N-17))
  } else { # "old"
    An_young <- c(rep(0.1,2),rep(0.05,5),rep(0.025,10),rep(0.3/(N-17),N-17))
    An <- rev(An_young)
  }
  An
}

# ==== Transition matrix P(ÎżÎÿ) ====
build_transition_matrix <- function(lambda_theta, max_k = 50) {
  lev <- as.character(BMS$levels)
```

```r
  P <- matrix(0, length(lev), length(lev), dimnames = list(from = lev, to = lev))
  for (l1 in lev) {
    for (k in 0:max_k) {
      pk <- dpois(k, lambda_theta)
      j  <- BMS$transition(l1, k)
      P[l1, j] <- P[l1, j] + pk
    }
    # put the truncated tail mass onto the "largest" jump (k = max_k + 1)
    tail <- 1 - sum(dpois(0:max_k, lambda_theta))
    if (tail > 0) {
      j_tail <- BMS$transition(l1, max_k + 1)
      P[l1, j_tail] <- P[l1, j_tail] + tail
    }
  }
  stopifnot(all(abs(rowSums(P) - 1) < 1e-12))
  P
}


# ==== Aggregate distribution over base levels ====
aggregate_dist <- function(p_vec) {
  out <- numeric(length(true_levels)); names(out) <- as.character(true_levels)
  state_names <- names(p_vec)
  for (j in names(out)) {
    idx <- grep(paste0("^", j, "(\\.|$)"), state_names, value = TRUE)
    out[j] <- sum(p_vec[idx], na.rm = TRUE)
  }
  out
}

# ==== Precompute all P matrices (reused everywhere) ====
P_map <- vector("list", h)
for (g in 1:h) {
  message(sprintf("  Precomputing matrix for g=%d / %d", g, h))
  lambda <- lambda_g$lambda[g]
  Pg <- vector("list", length(theta_grid))
  for (i in seq_along(theta_grid)) {
    Pg[[i]] <- build_transition_matrix(lambda * theta_grid[i])
  }
  P_map[[g]] <- Pg
}

# ==== A) Conditional relativities: 5 curves per L0 (scaled by level_100) ====

```

```r
cond_relativities_for_L0 <- function(L0_type_here) {
  L0_here <- build_L0(L0_type_here)

  # collect pn at horizons n_set
  pn_slices <- array(0, dim = c(length(n_set), h, length(theta_grid), length(true_levels)),
                     dimnames = list(n = as.character(n_set),
                                     g = paste0("g", 1:h),
                                     theta = NULL,
                                     l = as.character(true_levels)))
  maxN <- max(n_set)

  for (g in 1:h) {
    message(sprintf("  Computing conditional relativities for g=%d / %d", g, h))
    for (i in seq_along(theta_grid)) {
      P <- P_map[[g]][[i]]
      v <- L0_here
      k_idx <- 1
      for (t in 1:maxN) {
        v <- as.numeric(v %*% P); names(v) <- levels
        if (t == n_set[k_idx]) {
          pn_slices[k_idx, g, i, ] <- aggregate_dist(v)
          if (k_idx < length(n_set)) k_idx <- k_idx + 1
        }
      }
    }
  }

  # compute conditional relativity for each n in n_set, then scale
  cond_by_n <- list()
  for (k in seq_along(n_set)) {
    Num_vec <- sapply(as.character(true_levels), function(l_name) {
      sum(sapply(1:h, function(g) {
        lambda <- lambda_g$lambda[g]
        w_g[g] * sum(lambda^2 * theta_grid * theta_weights * pn_slices[k, g, , l_name])
      }))
    })
    Den_vec <- sapply(as.character(true_levels), function(l_name) {
      sum(sapply(1:h, function(g) {
        lambda <- lambda_g$lambda[g]
        w_g[g] * sum(lambda^2 *              theta_weights * pn_slices[k, g, , l_name])
      }))
    })
```

```
    cond_vec <- Num_vec / Den_vec
    cond_vec <- cond_vec / cond_vec[level_100]
    names(cond_vec) <- as.character(true_levels)
    cond_by_n[[as.character(n_set[k])]] <- cond_vec
  }
  cond_by_n
}

# ---- Plot 1,2,3: Conditional curves for L0 = uniform | bot | top ----
for (L0_type_here in c("uniform", "bot", "top")) {
  message(sprintf("  Computing for L0 = %s", L0_type_here))
  cond_by_n <- cond_relativities_for_L0(L0_type_here)
  all_vals  <- unlist(cond_by_n, use.names = FALSE)
  if (!is.null(insurer_relat)) all_vals <- c(all_vals, insurer_relat)
  ylim_all <- range(all_vals, na.rm = TRUE)

  # base plot with the first n
  first_n <- n_set[1]
  plot(as.numeric(true_levels), cond_by_n[[as.character(first_n)]],
       type = "l", lwd = 2, col = cols[1],
       xlab = "BMS Level", ylab = "Relativity",
       main = sprintf("Gamma: L0 = %s", L0_type_here),
       ylim = ylim_all)
  abline(h = 1, col = "gray", lty = 2)

  # add other horizons
  if (length(n_set) > 1) {
    for (k in 2:length(n_set)) {
      nk <- n_set[k]
      lines(as.numeric(true_levels), cond_by_n[[as.character(nk)]],
            lwd = 2, col = cols[k])
    }
  }

  # insurer reference (unscaled)
  if (!is.null(insurer_relat)) {
    lines(as.numeric(true_levels), insurer_relat, lwd = 2, col = "black", lty = 3)
    legend("topleft", legend = c(paste0("n=", n_set), "Insurer"),
           col = c(cols, "black"), lwd = 2, lty = c(rep(1, length(n_set)), 3), bty = "n")
  } else {
    legend("topleft", legend = paste0("n=", n_set),
           col = cols, lwd = 2, lty = 1, bty = "n")
  }
```

```r
}

# ==== B) Unconditional relativities (scaled) ====

compute_rbar_for <- function(L0_type_here, An_type_here = "uniform
  ") {
  L0_here <- build_L0(L0_type_here)
  An_here <- build_An(N, An_type_here)

  Num <- setNames(numeric(length(true_levels)), as.character(true_
   levels))
  Den <- setNames(numeric(length(true_levels)), as.character(true_
   levels))

  for (g in 1:h) {
    message(sprintf("  Computing unconditional relativities for g
   =%d / %d", g, h))
    lambda <- lambda_g$lambda[g]
    wg     <- w_g[g]
    for (i in seq_along(theta_grid)) {
      P    <- P_map[[g]][[i]]
      v    <- L0_here
      wt   <- theta_weights[i]
      lam2 <- lambda^2
      th   <- theta_grid[i]

      for (t in 1:N) {
        v <- as.numeric(v %*% P); names(v) <- levels
        agg <- aggregate_dist(v)
        Num <- Num + An_here[t] * wg * (lam2 * th * wt) * agg
        Den <- Den + An_here[t] * wg * (lam2      * wt) * agg
      }
    }
  }
  r_bar <- Num / Den
  r_bar / r_bar[level_100]  # scale
}

# ---- Plot 4: Unconditional âĂŤ varying L0, A_n fixed = uniform
  ----
r_bar_L0_uniform <- compute_rbar_for("uniform", "uniform")
r_bar_L0_bot     <- compute_rbar_for("bot",     "uniform")
r_bar_L0_top     <- compute_rbar_for("top",     "uniform")

ylim1 <- range(c(r_bar_L0_uniform, r_bar_L0_bot, r_bar_L0_top,
   insurer_relat), na.rm = TRUE)
plot(as.numeric(true_levels), r_bar_L0_uniform, type = "l", lwd =
   2, col = "dodgerblue",
     xlab = "BMS Level", ylab = "Scaled unconditional relativity",
```

```
    main = expression(paste("Unconditional ", bar(r), " with ", A[n], " uniform")),
    ylim = ylim1)
lines(as.numeric(true_levels), r_bar_L0_bot, lwd = 2, col = "darkorange")
lines(as.numeric(true_levels), r_bar_L0_top, lwd = 2, col = "forestgreen")
lines(as.numeric(true_levels), insurer_relat, lwd = 2, col = "black", lty = 3)
abline(h = 1, col = "gray", lty = 2)
legend("topleft",
       legend = c("L0 = uniform", "L0 = bot", "L0 = top", "Insurer relat."),
       col = c("dodgerblue", "darkorange", "forestgreen", "black"),
       lwd = 2, lty = c(1,1,1,3), bty = "n")

# ---- Plot 5: Unconditional âĂŤ varying A_n, L0 fixed = uniform ----
r_bar_A_uniform <- r_bar_L0_uniform
r_bar_A_young   <- compute_rbar_for("uniform", "young")
r_bar_A_old     <- compute_rbar_for("uniform", "old")

ylim2 <- range(c(r_bar_A_uniform, r_bar_A_young, r_bar_A_old, insurer_relat), na.rm = TRUE)
plot(as.numeric(true_levels), r_bar_A_uniform, type = "l", lwd = 2, col = "dodgerblue",
    xlab = "BMS Level", ylab = "Scaled unconditional relativity",
    main = expression(paste("Unconditional ", bar(r), " with L0 uniform")),
    ylim = ylim2)
lines(as.numeric(true_levels), r_bar_A_young, lwd = 2, col = "darkorange")
lines(as.numeric(true_levels), r_bar_A_old,   lwd = 2, col = "forestgreen")
lines(as.numeric(true_levels), insurer_relat, lwd = 2, col = "black", lty = 3)
abline(h = 1, col = "gray", lty = 2)
legend("topleft",
       legend = c(expression(A[n]*" = uniform"),
                  expression(A[n]*" = young"),
                  expression(A[n]*" = old"),
                  "Insurer relat."),
       col = c("dodgerblue", "darkorange", "forestgreen", "black"),
       lwd = 2, lty = c(1,1,1,3), bty = "n")

# ==== C) Financial balance I_n for conditional relativities (per L0) ====
```

```
cond_relat_for_L0 <- function(L0_type_here, n_fixed) {
  # compute & return unscaled conditional vector at horizon n_fixed
  L0_here <- build_L0(L0_type_here)
  names(L0_here) <- levels

  lvl_names <- as.character(true_levels)
  Num_vec <- setNames(numeric(length(lvl_names)), lvl_names)
  Den_vec <- setNames(numeric(length(lvl_names)), lvl_names)

  for (g in 1:h) {
    message(sprintf("  Computing conditional relativities for g=%d / %d", g, h))
    lambda <- lambda_g$lambda[g]
    wg     <- w_g[g]
    for (i in seq_along(theta_grid)) {
      P  <- P_map[[g]][[i]]
      wt <- theta_weights[i]
      th <- theta_grid[i]

      v <- L0_here
      for (t in 1:n_fixed) v <- as.numeric(v %*% P)
      names(v) <- levels
      aggn <- aggregate_dist(v)

      Den_vec <- Den_vec + wg * (lambda^2)      * wt * aggn
      Num_vec <- Num_vec + wg * (lambda^2 * th) * wt * aggn
    }
  }
  Num_vec / Den_vec  # unscaled
}

In_curve_for_L0 <- function(L0_type_here, cond_relat_fixed) {
  L0_here <- build_L0(L0_type_here)
  names(L0_here) <- levels

  lvl_names <- as.character(true_levels)
  PL_acc <- matrix(0, nrow = N + 1, ncol = length(lvl_names),
                   dimnames = list(n = 0:N, l = lvl_names))

  for (g in 1:h) {
    message(sprintf("  Computing financial balance for g=%d / %d", g, h))
    wg <- w_g[g]
    for (i in seq_along(theta_grid)) {
      wt <- theta_weights[i]
      P  <- P_map[[g]][[i]]

```

```
      # n=0
      v <- L0_here
      aggn0 <- aggregate_dist(v)
      PL_acc[1, ] <- PL_acc[1, ] + wg * wt * aggn0

      # n=1..N
      for (t in 1:N) {
        v <- as.numeric(v %*% P); names(v) <- levels
        aggn <- aggregate_dist(v)
        PL_acc[t + 1, ] <- PL_acc[t + 1, ] + wg * wt * aggn
      }
    }
  }

  as.numeric(PL_acc %*% cond_relat_fixed[lvl_names])
}

# fix n for conditional relativities used in I_n
n_fixed_for_I <- 50
cond_uniform <- cond_relat_for_L0("uniform", n_fixed_for_I)
cond_bot     <- cond_relat_for_L0("bot",     n_fixed_for_I)
cond_top     <- cond_relat_for_L0("top",     n_fixed_for_I)

In_uniform <- In_curve_for_L0("uniform", cond_uniform)
In_bot     <- In_curve_for_L0("bot",     cond_bot)
In_top     <- In_curve_for_L0("top",     cond_top)

ylim_I <- range(c(In_uniform, In_bot, In_top), na.rm = TRUE)
plot(0:N, In_uniform, type = "l", lwd = 2, col = "dodgerblue",
     xlab = "n (years)", ylab = expression(I[n]),
     main = bquote(I[n]~"vs n, fixed"~~r[1]^{(.(n_fixed_for_I))}~~
    "per"~~L[0]),
     ylim = ylim_I)
lines(0:N, In_bot, lwd = 2, col = "darkorange")
lines(0:N, In_top, lwd = 2, col = "darkgreen")
abline(h = 1, col = "gray", lty = 3, lwd = 2)
legend("topright",
       legend = c(expression(L[0] == "uniform"),
                  expression(L[0] == "bot"),
                  expression(L[0] == "top")),
       col = c("dodgerblue", "darkorange", "darkgreen"),
       lwd = 2, bty = "n")

# ==== C) Financial balance I_n for unconditional relativities (
    per An) ====
In_curve_for_An <- function(An_type_here, uncond_relat_fixed) {
  An_here <- build_An(n_fixed_for_I,An_type_here)
  L0_here <- build_L0("uniform")
  names(An_here) <- names(L0_here) <- levels
```

```
  lvl_names <- as.character(true_levels)
  PL_acc <- matrix(0, nrow = N + 1, ncol = length(lvl_names),
                   dimnames = list(n = 0:N, l = lvl_names))

  for (g in 1:h) {
    message(sprintf("  Computing financial balance for g=%d / %d",
    g, h))
    wg <- w_g[g]
    for (i in seq_along(theta_grid)) {
      wt <- theta_weights[i]
      P  <- P_map[[g]][[i]]

      # n=0
      v <- L0_here
      aggn0 <- aggregate_dist(v)
      PL_acc[1, ] <- PL_acc[1, ] + wg * wt * aggn0

      # n=1..N
      for (t in 1:N) {
        v <- as.numeric(v %*% P); names(v) <- levels
        aggn <- aggregate_dist(v)
        PL_acc[t + 1, ] <- PL_acc[t + 1, ] + wg * wt * aggn
      }
    }
  }

  as.numeric(PL_acc %*% uncond_relat_fixed[lvl_names])
}

# fix n for unconditional relativities used in I_n
n_fixed_for_I <- N

In_uniform <- In_curve_for_An("uniform", r_bar_A_uniform)
In_young     <- In_curve_for_An("young",     r_bar_A_young)
In_old     <- In_curve_for_An("old",     r_bar_A_old)

ylim_I <- range(c(In_uniform, In_young, In_old), na.rm = TRUE)
plot(0:N, In_uniform, type = "l", lwd = 2, col = "dodgerblue",
     xlab = "n (years)", ylab = expression(bar(I)[n]),
     #main = bquote(I[n]~"vs n, fixed"~~r[l]^{(.(n_fixed_for_I))}~
    ~"per"~~L[0]),
     ylim = ylim_I)
lines(0:N, In_young, lwd = 2, col = "darkorange")
lines(0:N, In_old, lwd = 2, col = "darkgreen")
abline(h = 1, col = "gray", lty = 3, lwd = 2)
legend("topright",
       legend = c(expression(L[0] == "uniform"),
                  expression(L[0] == "young"),
```

```
                    expression(L[0] == "old")),
        col = c("dodgerblue", "darkorange", "darkgreen"),
        lwd = 2, bty = "n")
```

Bibliography

[PDW03a] Sandra Pitrebois, M. Denuit, and J. F. Walhin. “Fitting the Belgian bonus-malus system”. In: *Belgian Actuarial Bulletin* (2003), pp. 58–62.

[PDW03b] Sandra Pitrebois, Michel Denuit, and Jean-François Walhin. “Setting a Bonus-Malus Scale in the Presence of Other Rating Factors: Taylor’s Work Revisited”. In: *ASTIN Bulletin* 33.2 (2003). Edition: 2015/04/17 Publisher: Cambridge University Press, pp. 419–436. ISSN: 0515-0361. DOI: 10.2143/AST.33.2.503701. URL: https://www.cambridge.org/core/product/C27ED6CAEBF891B2B1FAB92957308CAA.

[Den+07] Michel Denuit et al. *Actuarial modelling of claim counts: Risk classification, credibility and bonus-malus systems*. John Wiley & Sons, 2007. ISBN: 0-470-51741-7.

[Tan+15] Chong It Tan et al. “Optimal relativities and transition rules of a bonus–malus system”. en. In: *Insurance: Mathematics and Economics* 61 (Mar. 2015), pp. 255–263. ISSN: 01676687. DOI: 10.1016/j.insmatheco.2015.02.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167668715000153 (visited on 09/08/2024).

[Tan16a] Chong It Tan. “Optimal design of a bonus-malus system: linear relativities revisited”. en. In: *Annals of Actuarial Science* 10.1 (Mar. 2016), pp. 52–64. ISSN: 1748-4995, 1748-5002. DOI: 10.1017/S1748499515000111. URL: https://www.cambridge.org/core/product/identifier/S1748499515000111/type/journal_article (visited on 09/08/2024).

[Tan16b] Chong It Tan. “Varying transition rules in bonus–malus systems: From rules specification to determination of optimal relativities”. en. In: *Insurance: Mathematics and Economics* 68 (May 2016), pp. 134–140. ISSN: 01676687. DOI: 10.1016/j.insmatheco.2016.03.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167668715301815 (visited on 09/08/2024).

[Ins23] AG Insurance. *Assurance RC Auto - Conditions générales*. Nov. 2023. URL: https://ag.be/dam/transversal/retail/documents/product-documents/fr/vehicles/20004F.pdf (visited on 08/15/2025).

[Bel24] AXA Belgium. *Confort Auto - Assurance et assistance Auto - Conditions générales*. Oct. 2024. URL: https://cdn.website.axa.be/fr-be/-/media/axa-documents/library-public/cg/c/d/4/f/a/418556010201302/CONFORT%20AUTO%20-%20Assurance%20et%20assistance%20Auto%20-%2020241001.pdf (visited on 08/15/2025).

[24] *Top 15 RC automobile avec parts de marché*. Tech. rep. Assuralia, Sept. 2024. URL: https://files.assuralia.be/stats/FR/04_composition-du-marche/04_03_top15-rc-automobile.htm (visited on 08/02/2025).

[Assa] KBC Assurances. *Conditions générales - Police KBC pour votre véhicule*. URL: https://multimediafiles.kbcgroup.eu/ng/published/KBC/PDF/KBC_algemene_voorwaarden_personenauto_F.pdf (visited on 08/15/2025).

[Assb] P&V Assurances. *Conditions générales - P&V Auto*. URL: https://www.pv.be/documents/1465818/28359821/PV+-+version+novembre+2019+-+version+finale.pdf/ (visited on 08/15/2025).

[Bel] Baloise Belgium. *Mobility Safe 1 - Conditions Générales*. URL: https://www.baloise.be/dam/mybaloise-baloise-be/fr/non-vie-particuliers/Vehicules-automoteurs/conditions-generales/B3311.MOT.01.17-CG-Mobility-Safe-1.pdf (visited on 08/15/2025).

[Eth] Ethias. *Conditions générales - Assurance Auto*. URL: https://www.ethias.be/content/dam/B2C/general/pdf/general-conditions/cg-auto-fr.pdf (visited on 08/15/2025).

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences
Place des Sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/sc