Université catholique de Louvain

Institut de Statistique, Biostatistique et Actuariat

Application of the artificial states method in individual reserving

Membres du jury:

Prof. Michel DENUIT, *Promoteur*
Prof. Jean-François WALHIN, *Lecteur*

Mémoire présenté en vue de l'obtention du master en sciences actuarielles (orientation sciences actuarielles) par:

Arnaud MARLIER

Louvain-La-Neuve
2021-2022

Contents

1 Introduction **1**
- 1.1 Contextualization of the subject . . . 1
- 1.2 Presentation of existing reserving models . . . 1
- 1.3 Presentation of the project and objectives . . . 2

2 Theoretical explanation **3**
- 2.1 Multistate approach . . . 3
- 2.2 Claim occurrence rate and reporting lag . . . 6
- 2.3 Cash-flows . . . 8
- 2.4 Switch from the Semi-Markov to the Markov model using artificial states . 9
- 2.5 Transition probability matrix . . . 11
- 2.6 Expected ultimate cost . . . 14
- 2.7 One-year risk view . . . 16

3 Numerical illustration **18**
- 3.1 Database . . . 18
  - 3.1.1 Individual Claims History Simulation Machine . . . 18
  - 3.1.2 Data analysis . . . 20
  - 3.1.3 Augmented database . . . 22
- 3.2 Estimation of transition probabilities and reporting lag . . . 24
  - 3.2.1 Claim occurrence rate and reporting lag . . . 24
  - 3.2.2 Transition probabilities and transition probability matrix . . . 25
- 3.3 Estimation of cash-flows . . . 26
  - 3.3.1 Discrete mixture model . . . 26
  - 3.3.2 Estimation of the first payment . . . 27
  - 3.3.3 Estimation of link ratios and tie with the first payment . . . 29
- 3.4 Expected ultimate cost and one-year risk view . . . 34
  - 3.4.1 $\tilde{M}$ matrix . . . 34
  - 3.4.2 Expected ultimate cost and reserve calculation . . . 35
  - 3.4.3 Simulations . . . 37
  - 3.4.4 Claim Development Result . . . 40

4 Conclusion **42**
4.1 Summary of the project 42
4.2 Further possible research 43

A R code **44**
A.1 No reopening of claims 44
A.2 Determination of the *State* variable 44
A.3 Determination of the *Duration* variable 46
A.4 Build of the augmented database 47
A.5 Reporting lag 48
A.6 Expected number of claims per year 49
A.7 Cash-flows estimation 49
A.8 Reserve calculation with the Chain-Ladder method 50
A.9 Simulation of transitions for the coming year 51
A.10 Simulation of payments for the coming year 52

B Transition probability matrix and $\tilde{M}$ matrix **54**
B.1 Transition probability matrix 55
B.2 $\tilde{M}$ matrix depending on the value of the first payment 56

Chapter 1

Introduction

1.1 Contextualization of the subject

As we know, modern non-life insurance aims to protect insured against the risk of a potential financial loss. The insured will therefore pay a certain amount to the insurer in order to be protected against this financial loss. Only the insurer will have to put money aside in order to be sure to be able to reimburse the insured if the claim ever occurs. This amount that has to be put aside is the reserve. In the Solvency II framework we will focus on the one-year risk view.

We will therefore focus in this work on a particular individual reserving method, the artificial states method in individual reserving. This discrete-time stochastic model aim is to modelize the trajectories as well as the cash-flows of each claim individually, that in order to calculate the reserve and to have a view on the one-year risk.

1.2 Presentation of existing reserving models

As we know it exists two different types of reserving methods in non-life insurance.

The first one, which is the most commonly used, consists of models which use aggregated data. These models allow us to determine the reserve for the entire portfolio only. Historically, the aggregated reserving methods are more used because they are easier to understand and to implement. The best-known aggregated reserving method is certainly the Chain Ladder method.

The second one consists of the use of individual data. These methods allow us to determine the reserve of each claim individually as well as the reserve for the entire portfolio. Individual reserving methods generally require much higher computing power. However, with the increasing computing power and the need to use as much data as possible, individual

reserving methods tend to be used more and more.
The artificial multistate method that will be used in this work is obviously an individual reserving method.

With regard to the literature, the mathematical basis of individual loss reserving models can be attributed to Arjas (1989) [1] and Norberg (1993) [2] , (1999) [3] with, among other things, the use of stochastic processes to categorize each claim individually.
We can also see in the paper *Individual claims reserving: a survey* (2017) [4] and in the paper *Macro vs. Micro Methods in Non-Life Claims Reserving (an Econometric Perspective)* (2016) [5] comparisons between different individual reserving models as well as with models using aggregated data.
It is also worth mentioning the paper *Stochastic projection for large individual losses* (2012) [6] which focuses on the estimated final value of large losses on an individual basis using a stochastic approach.
For more details about loss reserving in general we refer to Wüthrich & Merz (2008) [7] (2015) [8].

We can also see that currently new methods are being put in place, in particular with the use of neural networks such as in *Stochastic loss reserving with mixture density neural networks* (2022) [9] which is applying a Mixture Density Network to loss reserving.

1.3 Presentation of the project and objectives

The first objective of this work is to theoretically define the artificial states model in individual reserving as well as the methodology to follow in order to implement it. For that we will at first define the concept of the multistate approach in discrete time. After that we will determine the claim occurrence rate and the reporting lag. We will then define cash-flows as well as link-ratios associated with transition from one state to another. After that we will explain how to switch from the Semi-Markov to the Markov model using artificial states in order to ease the calculations and we will define the transition probability matrix. All this will allow us to retrieve the expected ultimate cost and therefore the reserve for all claims. We could then finally determine the one-year risk view with the help of the Claim Development Result.

The second objective of this work is to implement the presented method with the help of a numerical illustration. We will then at first present the used data and perform some data analysis. After that we will estimate the transition probabilities as well as the cash-flows. With that we could calculate the expected ultimate cost, which will allow us to retrieve the reserve for all claims. We could then finally have the one-year risk view. We will also compare these results with results obtained with the Chain Ladder method.

Chapter 2

Theoretical explanation

In this chapter we will establish the methodology to follow in order to implement the Semi-Markov loss reserving model and we will complete these explanations with the help of a simple example as we go through the different stages.

This model is based on the paper *Matrix calculation for ultimate and 1-year risk in the Semi-Markov individual loss reserving model* (2021) [10].

We will at first explain the principle of the multistate approach. Then we will define the claim occurrence rate and the reporting lag. After that we will determine cash-flows which are associated with transition between states.
We will then have a theoretical reminder on the Markov model, and explanation on how to switch from the Semi-Markov to the Markov model using artificial states, which will allow us to define a transition probability matrix.
Finally, we will define how to calculate the expected ultimate cost, which will allow us to determine the one-year risk view with the help of the Claim Development Result.

2.1 Multistate approach

As we know, claims can go through different states from its occurrence until final settlement :

- The first one, in which all claims go through, is the Incurred But Not Reported (IBNR) state. As its name says, claims are in this state from the time they took place until the time they are reported. The insurer is of course liable for these claims, even though he is still unaware of their existence.
- The second one is the Reported But Not Paid (RBNP) state. This is the state in which claims are reported but no payment has been made.
- The third one is the Reported But Not Settled (RBNS) state, which is the state in

where a payment has already been made but claims are not settled yet. This means that claims in this state are not closed, and there may still be payments in the future.

- The fourth one is the state where claims would be settled without any final payment being made. We will call this state Closed0.
- Finally, the last one is the state where claims would be settled with a final payment. We will call this state Closed+.

The three first possibilities are states where claims are not settled yet. These three cases will then transition into different states in continuous time :

- For the IBNR state, claims could transition to RBNP.
- For the RBNP state, claims could transition to RBNS, Closed0 or Closed+.
- Finally, if the claim is in the RBNS state, it could transition to Closed0 or Closed+.

The two last possibilities (Closed0 and Closed+) are states where claims are settled. As it is assumed that when a claim is closed it could not reopen, there is no possibility of transition from these two last states.

Discrete-time model

We will here use a discrete-time model with a time step of one year as it is the most commonly used time step in the insurance field.

The evolution of the state of a claim is then given by the stochastic process

$$S = \{S_0, S_1, S_2, ..., S_T\}$$

with T being the maximum number of years until all claims are settled.
As all claims are initially in the IBNR state, we have $S_0 =$ IBNR for all claims. We also have $S_T =$ Closed0 or $S_T =$ Closed+ as all claims are settled after a maximum of T years.

There will therefore possibly be one or more transitions in each period depending on the evolution of the claim. For example, a claim could go from the IBNR state to the RBNP state and then from this last state to the RBNS state and that only in one year. As we are working in a discrete time model, we will only take into account the transition from the initial state toward the state in which the claim is at the end of the period. That means that in our example, the transition will be seen as a transition from the IBNR state to the RBNS state.

Moreover, if several payments have been made during a single period, all these payments will be aggregated into a single payment linked to this period.

Each payment in a period of time will then be accompanied with a transition between states.

Finally, the RBNS state has to be subdivided into different sub-states RBNS_l with $l = 1, 2, ..., n$ and $n + 1$ being the maximum number of aggregated payments before the claim is considered to be settled[1]. We obviously have $T > n$ as claims could stay in the same state for multiple years.

The following figure illustrate the multistate discrete-time model and the different transition possibilities :

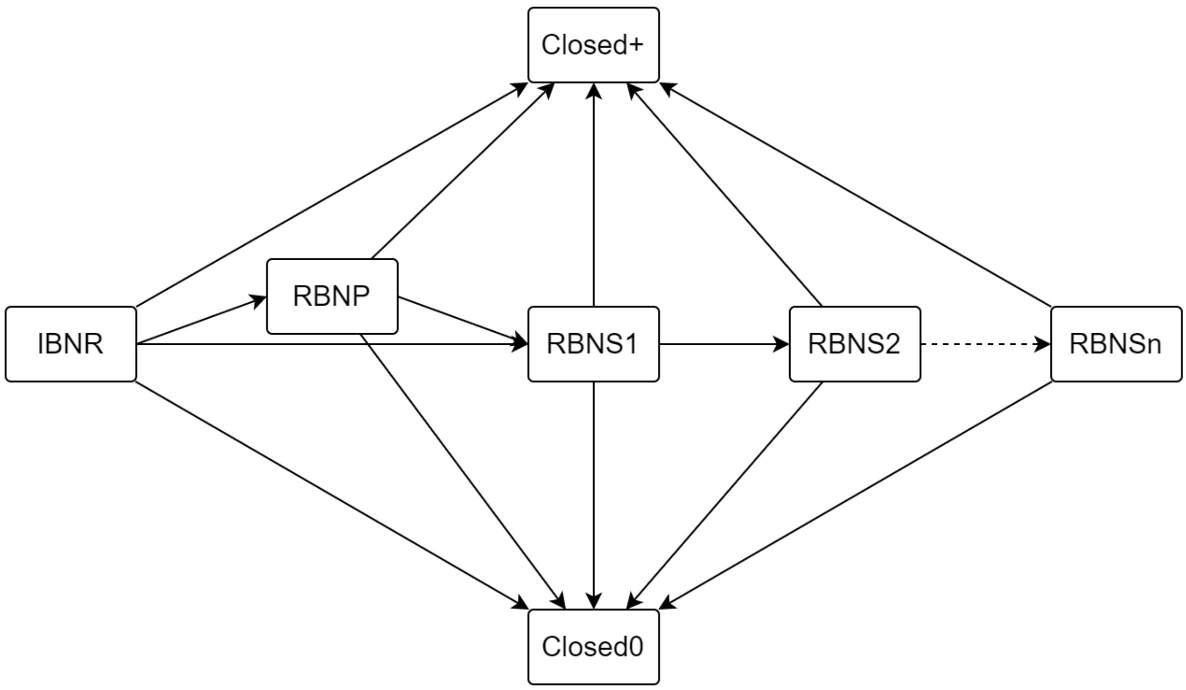

Figure 2.1.1: Illustration of the transition possibilities of the discrete-time multistate model.

Application to a simple example

As a simple example, we could have claims that could only go through four different states : the IBNR state, the RBNS state, the Closed+ state and the Closed0 state.
Additionally, we assume that these claims remains less than two years in the same state, except for the closed states which are the final states. We can see that the maximum

[1] It is indeed $n + 1$ and not n as there could still be a final payment when going to the Closed+ state.

number of years until all claims are settled is four years. We then have $T = 4$ and $n = 1$. We will also assume that there could not be claims where no payment are being made, which means that claims could not transition from the IBNR state directly to the Closed0 one.

We have then the following figure which illustrates the different transition possibilities based on our example :

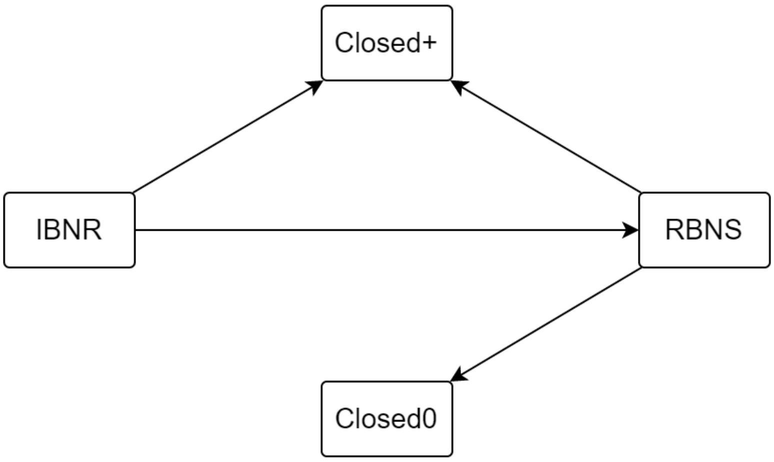

Figure 2.1.2: Illustration of the transition possibilities of a simplified example.

2.2 Claim occurrence rate and reporting lag

Let's denote N_i the total number of claims during the accident year i.
We assume the number of claims per year to be Poisson distributed and the independence of the number of claims between each year. That means that we have

$$N_i \sim Pois(\alpha_i)$$

with $\alpha_i = E[N_i]$ being the average number of claims for the accident year i.

But as we know, claims could stay in the IBNR state for years. This means that we don't necessarily know the total number of claims for each accident year at initial time. We then have to take into account a reporting lag, which is defined as the time that a claim stays in the IBNR state.
Let's now denote N_{ij} being the number of reported claims during the accident year i in the j^{th} development year. We also assume that these numbers follow a Poisson distribution

and that they are independent between each other. We have then

$$N_{ij} \sim Pois(\alpha_i \beta_j)$$

with α_i defined hereabove and β_j being the probability that a claim initially in the IBNR state is reported at lag j. As we can see, we have made the assumption that the probability of reporting depending on the development year is the same for each accident years. This means that the claim reporting mechanism does not change from accident year to accident year, or at least if it were to change, this would have no influence on the reporting probability.

Logically we have

$$\sum_{l=1}^{T} \beta_l = 1$$

as all claims should be reported after T years, with T being the maximum number of year until final settlement as described in the previous section.
The average number of claims per accident year i reported at lag j is then given by $E[N_{ij}] = \alpha_i \beta_j$. α_i and β_j could be estimated with the help of a Generalized Linear Model, and more precisely with a Poisson regression using the observed value of N_{ij}.

Finally, the probability that a claim is not in the IBNR state at lag j, which means that it is reported at lag j or before is defined by

$$P[S_j \neq \text{IBNR} | S_0 = \text{IBNR}] = \sum_{l=1}^{j} \beta_l$$

where j could take any values ranging from 1 to T.

Of course, as defined hereabove, we have

$$P[S_T \neq \text{IBNR} | S_0 = \text{IBNR}] = \sum_{l=1}^{T} \beta_l = 1$$

as all claims have to be reported at the horizon time T.

Application to a simple example

Let's retake the simple example describe in the previous section.

We recall that claims could not stay more than two years in a single state, which means that a claim could stay at most two years in the IBNR state. We then only have β_1 and

β_2 as the probability that a claim leaves the IBNR state at the corresponding development year, and all claims are reported at the latest after two years.
As

$$\sum_{j=1}^{2} \beta_j = \beta_1 + \beta_2 = 1,$$

we have $\beta_2 = 1 - \beta_1$.

Let's now assume that after having done a Poisson regression, we have found that the probability to stay in the IBNR state after the first year is equal to 10%. We have then $\beta_1 = 0.9$, which gives us $\beta_2 = 0.1$.
Mathematically, we will write

$$P[S_1 \neq \text{IBNR} | S_0 = \text{IBNR}] = \beta_1 = 0.9$$

and

$$P[S_2 \neq \text{IBNR} | S_0 = \text{IBNR}] = \beta_1 + \beta_2 = 0.9 + 0.1 = 1.$$

2.3 Cash-flows

Each transition from any state to a RBNS or a Closed+ state is associated with cash-flows such that every time a payment is being made, the claim will transition from one state to another until reaching one of the two final states.
Let's not forget that we are working in a discrete-time model, which means that payments made during the same period will be aggregated into a single payment associated with this period.

Let's write P_l the payment made when entering the RBNS_l state. This cash-flow could either be positive or negative. It is easy to understand that this cash-flow can be positive, as it is a payment related to the claim from the insurer towards the insured. There exists nevertheless negative cash-flows. These are explained by the fact that if the third party involved is recognized as being liable of the claim, the insurer will recover part if not all of the previously paid amount. It could be only a part because the third party does not necessarily reimburse the entire paid amount, as for example he does not reimburse the internal settlement expenses.

The cumulative payment in state RBNS_j is then given by

$$C_j = \sum_{l=1}^{j} P_l.$$

We can then define a link ratio associated with the transition from the state RBNS_j to the state RBNS_{j+1} :

$$\Lambda_j = \frac{C_{j+1}}{C_j}.$$

The advantage of this link ratio is that it is easier to see negative payments, as for negative payments, the link ratio will be smaller than one.

We can then redefine the cumulative payment with the help of this link ratio. We have

$$C_1 = P_1 \text{ and } C_j = P_1 \prod_{l=1}^{j-1} \Lambda_l \text{ for } j > 1.$$

Application to a simple example

Let's retake our previous example and further assume that $E[P_1] = 10$ and $E[P_2] = 15$.

We have then

$$E[C_1] = E[P_1] = 10$$

and

$$E[C_2] = \sum_{l=1}^{2} E[P_l] = E[P_1] + E[P_2] = 25.$$

Finally, if we assume in this example that $E\left[\frac{C_2}{C_1}\right] = 2.5$, we have for the link ratio

$$E[\Lambda_1] = E\left[\frac{C_2}{C_1}\right] = 2.5.$$

2.4 Switch from the Semi-Markov to the Markov model using artificial states

Let's first explain the principle of Semi-Markov and Markov models, and the difference between these two as explained by Vlad Stefan Barbu and N. Liminios (2008) [11].

A Markov process, and more precisely a Markov chain is a stochastic process where the transition probabilities from one state to another only depends on the current state, and not on the previous ones. This property is also known as the Markov memoryless property. For a Semi-Markov process, the transition probabilities also depend on the current state and not on the previous ones. But in addition to that, it also depends on the time spent in the current state. The transition probabilities will then differ according to the sojourn time spent in the current state. Still we can see that the Semi-Markov process respect the Markov memoryless property, but in a more flexible way.

As we can see, we have been working so far in a Semi-Markov model. Indeed, the transition probabilities depend on the current state and on the time spent in this state, but not on the previous states.
However, it is more appropriate to work in a Markov model, as such model eases computation.

Indeed, we can use matrix calculation in a Markov model in order to determine transition probabilities after several years and so on. That will allow us to determine the expected ultimate cost.

In order to switch from a Semi-Markov to a Markov model, we will have to make our model independent of duration effects.
We will then have to work with artificial states in order to no longer be dependent on the duration, and therefore to be in a Markov model instead of a Semi-Markov model.

An artificial state is a fictional state in which the claim is located. It is defined as the state in which the claim is and the number of years since it has been in this state.
However, there may no longer be a duration effect after a claim has spent several years in a certain state. This means that after several years, the probabilities of transition from one state to another no longer depend on the time spent in this initial state. We could then regroup claims that have stayed enough time in this state into a single artificial state.

Mathematically, we will describe these artificial states with the process

$$\{(S_t, D_t), t = 0, 1, 2, ...\}$$

where S_t is the current state in which the claim is and D_t is the duration, which means the number of years since the claim is in this state, or when no duration effect is remaining, it is the number of years from which there is no longer any duration effect.

As these artificial states are Markovian, we could write

$$p_{ij} = P[(S_{t+1}, D_{t+1}) = j \mid (S_t, D_t) = i]$$

which is the probability that the claim will be in the artificial state j knowing that it is currently in the artificial state i. As we can see, this probability only depends on the current state, and not on the previous ones.

Application to a simple example

If we retake our previous example, we would have six different artificial states :

- The (IBNR, 0) which is the initial state.
- The (IBNR, 1) which is the one-year IBNR state.
- The (RBNS, 0) which is the initial RBNS state.
- The (RBNS, 1) which is the one-year RBNS state.
- The (Closed0, 0) which is the Closed0 state.
- The (Closed+, 0) which is the Closed+ state.

2.5 Transition probability matrix

We could then define a transition probability matrix M which will contain the transition probabilities from one artificial state to another. We have

$$M = (p_{i,j})_{i,j=1,2,...}$$

where $p_{i,j}$ is the probability of transition from the artificial state i to the artificial state j.

In addition to that, we could define a vector ω which is filled with zero except for the artificial state which is currently occupied by the claim, which will have one as a value.
For a claim in the j^{th} artificial state, we would then have

$$\omega^\top = \begin{pmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

with the 1 placed on the j^{th} entry.

With this, we can now define the probability to be at any state at horizon k knowing that the initial state is given by the vector ω. These probabilities are given by $\omega^\top M^k$.
We could then retransform these results in order to have the probability transition from one state to another in the Semi-Markov model.

Let's now define the matrix $\tilde{M}$ as the transition probability matrix M multiplied either with the mean of the initial payment $E[P_1]$ or with the mean of the link ratio $E[\Lambda_j]$ where $j = 1, 2, ...$ depending on the transition that takes place. Indeed, if we have a transition from the IBNR or the RBNP state to the RBNS_1 or the Closed+ state, the corresponding entry in the matrix M will be multiplied with the mean of the initial payment $E[P_1]$, whereas if we have a transition from the RBNS_j state to the RBNS_{j+1} or the Closed+ state, we would multiply the corresponding entry in the matrix M with the mean of the link ratio $E[\Lambda_j]$.

This matrix allows us to retrieve the corresponding costs. Indeed, we have $\omega^\top \tilde{M}^k$ defining the corresponding cost after k years when being in a certain artificial state defined with the vector ω.

Application to a simple example

Let's once again retake our previous example. The transition probability matrix will then be of the size 6×6 as there are six different artificial states.
We also have the vector ω which is of length six, with the first entry being linked to the artificial state (IBNR, 0), the second one to the (IBNR, 1) state, the third one to the (RBNS, 0) state, the fourth one to the (RBNS, 1) state, the fifth one to the (Closed0, 0)

state and finally the sixth one to the (Closed+, 0) state.

Let's now write the different transition probabilities. We already know from the section 2.2 that

$$P[(S_t, D_t) \neq (\text{IBNR}, 1)|(S_{t-1}, D_{t-1}) = (\text{IBNR}, 0)] = \beta_1 = 0.9$$

which gives us

$$P[(S_t, D_t) = (\text{IBNR}, 1)|(S_{t-1}, D_{t-1}) = (\text{IBNR}, 0)] = p_{1,2} = 1 - \beta_1 = 0.1$$

which is the probability that a claim stays in the IBNR state after one year.

We also have

$$P[(S_t, D_t) = (S_{t-1}, D_{t-1})] = 0$$

as it is not possible to remain in the same artificial state after a year has passed in our example. This will lead us to have the diagonal of the transition matrix being equal to zero.

As previously said, a claim could not transition directly from the IBNR state to the Closed0 one, as there are no claims for which no payment is made. We have then

$$P[S_t = \text{Closed0}|S_{t-1} = \text{IBNR}] = p_{1,5} = p_{2,5} = 0.$$

We also have

$$P[D_t \geq 1|S_t \neq S_{t-1}] = 0$$

as if a claim has just moved into a new state, the time that it has spent in this new state is necessarily smaller than one year.

Lastly we have

$$P[S_t \neq \text{Closed} * |S_{t-1} = \text{Closed}*] = 0$$

and

$$P[S_t = \text{Closed} * |S_{t-1} = \text{Closed}*] = 1$$

with $* = +$ or $* = 0$. These results come from the fact that when a claim is closed, it could not reopen. It will then stays in it's final closed state.

Let's now assume that we have

$$P[S_t = \text{RBNS}|S_{t-1} = \text{IBNR}] = \begin{cases} p_{1,3} = 0.7 & \text{for } D_{t-1} = 0 \\ p_{2,3} = 0.6 & \text{for } D_{t-1} = 1. \end{cases}$$

which is the probability of a claim being in the RBNS state knowing that it's last state was the IBNR one and also knowing the time spent in this last state.
That gives us

$$P[S_t = \text{Closed+}|S_{t-1} = \text{IBNR}] = \begin{cases} p_{1,6} = 1 - 0.1 - 0.7 = 0.2 & \text{for } D_{t-1} = 0 \\ p_{2,6} = 1 - 0.6 = 0.4 & \text{for } D_{t-1} = 1. \end{cases}$$

which is the probability that a claim is closed with a final payment knowing that it was previously in the IBNR state and the time spent in this previous state.

Let's also assume that

$$P[S_t = \text{RBNS}|S_{t-1} = \text{RBNS}] = p_{3,4} = 0.5$$

which is the probability that a claim stays in the same RBNS state for an additional consecutive year.

Finally, let's assume that we have

$$P[S_t = \text{Closed} * |(S_{t-1}, D_{t-1}) = (\text{RBNS}, 0)] = \begin{cases} p_{3,5} = 0.3 & \text{for } * = 0 \\ p_{3,6} = 0.2 & \text{for } * = +. \end{cases}$$

and

$$P[S_t = \text{Closed} * |(S_{t-1}, D_{t-1}) = (\text{RBNS}, 1)] = \begin{cases} p_{4,5} = 0.2 & \text{for } * = 0 \\ p_{4,6} = 0.8 & \text{for } * = +. \end{cases}$$

which are the probability that a claim closes with or without a final payment knowing the time spent in the previous RBNS state.

We can then fill the transition probability matrix using all the data that we have :

$$M = \begin{pmatrix} 0 & 0.1 & 0.7 & 0 & 0 & 0.2 \\ 0 & 0 & 0.6 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 0.5 & 0.3 & 0.2 \\ 0 & 0 & 0 & 0 & 0.2 & 0.8 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Based on this matrix, on the average initial payment $E[P_1]$ as well as on the average link ratio $E[\Lambda_1]$, we can find the matrix $\tilde{M}$. We have

$$\tilde{M} = \begin{pmatrix} 0 & 0.1 & 0.7 \times 10 & 0 & 0 & 0.2 \times 10 \\ 0 & 0 & 0.6 \times 10 & 0 & 0 & 0.4 \times 10 \\ 0 & 0 & 0 & 0.5 & 0.3 & 0.2 \times 2.5 \\ 0 & 0 & 0 & 0 & 0.2 & 0.8 \times 2.5 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0.1 & 7 & 0 & 0 & 2 \\ 0 & 0 & 6 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0.5 & 0.3 & 0.5 \\ 0 & 0 & 0 & 0 & 0.2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

2.6 Expected ultimate cost

We can now determine the expected ultimate cost depending on the initial state of each claim using everything that we have seen so far.

Let's retake T as the maximum number of years until the settlement of all claims.

We have to calculate the expected ultimate cost in different ways as we have different cases. We have three different cases :

- The first case is for claims that are in the IBNR state. For all claims that have occurred in the accident year i and which are in the IBNR state, we have

$$UC^{expected} = \alpha_i \sum_{j \geq j_0+1} \beta_j (\omega_{IBNR_j}^{\top} \tilde{M}^T \mathbf{1})$$

  with $j_0 \geq 1$ being the number of years that have passed since the accident took place and ω_{IBNR_j} the vector indicating the current artificial state of the claim.

- The second case is for claims that are in the RBNP state. The expected ultimate cost for a claim in the RBNP state is given by

$$UC^{expected} = \omega_{RBNP}^{\top} \tilde{M}^T \mathbf{1}.$$

- Finally the last case is for claims that are in the RBNS_j state. We have

$$UC^{expected} = P_1 \left(\prod_{l=1}^{j-1} \Lambda_l \right) (\omega_{RBNS_j}^{\top} \tilde{M}^T \mathbf{1}).$$

The difference between these two last cases is that for a claim in the RBNS_j state, we will use the values that we already know, which are the value of the first payment and of the $j-1$ first link ratios whereas for a claim in the RBNP state we don't know any of these values. We have then to work instead with their expected values.

Application to a simple example

We will here present two different examples.

In the first example, we would like to calculate the expected ultimate cost for claims relating to a specific accident year that are still in the IBNR state after the first development year.

Let's imagine that the expected number of claims for this specific accident year is equal to 10. We have then

$$E[N] = \alpha = 10.$$

As said earlier, the maximum number of years until final settlement is 4 years, which gives us $T = 4$.

As one year has already past, claims are in the artificial state (S_t, D_t) =(IBNR, 1), which gives us

$$\omega_{IBNR}^{\top} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

We can then calculate the total expected ultimate cost for claims that are still in the IBNR state. We have then

$$\begin{aligned} UC^{expected} &= \alpha \times \beta_2(\omega_{IBNR}^{\top}\tilde{M}^T\mathbf{1}) \\ &= 10 \times 0.1(\omega_{IBNR}^{\top}\tilde{M}^4\mathbf{1}) \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 & 2.4 & 13 \end{pmatrix}\mathbf{1} \\ &= 15.4. \end{aligned}$$

The expected ultimate cost for claims in the IBNR state is then equal to 15.4.

For the other example we will take a claim that is in the (RBNS, 0) artificial state with a first known payment $P_1 = 15$.
We still have $T = 4$ as the maximum number of years until final settlement hasn't changed.
We also have

$$w_{RBNS}^{\top} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

We will then have for the ultimate cost

$$\begin{aligned} UC^{expected} &= P_1\omega_{RBNS_j}^{\top}\tilde{M}^T\mathbf{1} \\ &= 15\begin{pmatrix} 0 & 0 & 0 & 0 & 0.4 & 1.5 \end{pmatrix}\mathbf{1} \\ &= 28.5. \end{aligned}$$

The expected ultimate cost for the claim in the (RBNS, 0) artificial state is then equal to 28.5.

2.7 One-year risk view

As we are working in the Solvency II framework, we are only interested in the one-year risk view. Indeed, in the Solvency II framework, reserve risk is no longer defined as the risk that the reserve will not cover all future payments of claims that have already occurred, but is rather defined using the Claim Development Result. This Claim Development Result is defined as

$$\begin{aligned} CDR = & \text{ Reserve at the beginning of the year} \\ & - \text{ Reserve at the end of the year} \\ & - \text{ Payments occurred during the year.} \end{aligned}$$

We can therefore see that, in the Solvency II framework, the time horizon is only one year. This will then prevent us from having simulation-within-simulation errors as we will only have to determine the reserve at the beginning of the year, the one at the end of the year and the payment that occurred between these two moments.

The reserve at the beginning of the year is given by the analytical matrix method whereas the payment that occurred during the year will be simulated. After that simulation, we could again calculate the reserve with the help of the analytical matrix method, but this time the reserve is calculated at the end of the year, which means after that the simulated payment occurred.

By repeating theses steps a large number of time, we could determine both the best estimate, which is defined as the mean of the Claim Development Result for all these simulations, and the required capital in order to remain solvent at a certain percentile, which is defined as the Value-at-Risk.

Application to a simple example

Let's once again retake our previous example. If we take the two examples from the previous section, the reserve at the beginning of the year is equal to the sum of the expected ultimate cost for the two different cases from which we subtract the payment that have already been made. We have then

$$\begin{aligned} \text{Reserve at the beginning of the year} &= 15.4 + 28.5 - 15 \\ &= 28.9. \end{aligned}$$

Let's now imagine that at the end of the year there is only one claim that came from the IBNR state and that this claim transitioned to the RBNS state with an associated payment equal to 25. Let's also imagine that the claim that was in the RBNS state transitioned to

the Closed+ state, with a payment equal to 20.
The payment that have occurred during the year is then equal to

$$\begin{aligned}\text{Payments occurred during the year} &= 25 + 20 \\ &= 45.\end{aligned}$$

As one claim is still open, we have to calculate the reserve at the end of the year for this claim.
For that we need at first to calculate the ultimate cost for this claim. As the claim is now in the $(RBNS, 0)$ state, we have

$$w_{RBNS}^{\top} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

The ultimate cost is then given by

$$\begin{aligned}UC^{expected} &= P_1 \omega^{\top} \tilde{M}^T \mathbf{1} \\ &= 25 \begin{pmatrix} 0 & 0 & 0 & 0 & 0.4 & 1.5 \end{pmatrix} \mathbf{1} \\ &= 47.5.\end{aligned}$$

We have then for the reserve at the end of the year

$$\begin{aligned}\text{Reserve at the end of the year} &= 47.5 - 25 \\ &= 22.5.\end{aligned}$$

We have then for the Claim Development Result

$$\begin{aligned}CDR &= 28.9 - 22.5 - 45 \\ &= -38.6.\end{aligned}$$

We can see that the Claim Development Result is negative in our example. If the capital of the insurance company is lower than 38.6, it would mean that the company is not solvent anymore. In order to prevent this from happening, we have to determine a required capital to remain solvent, with the help of the Value-at-Risk, after having done multiple simulations.

Chapter 3

Numerical illustration

In this chapter we will implement the Semi-Markov individual loss reserving model on simulated data claims. We will then at first present and analyse the database. After that we will estimate the transition probabilities as well as the cash-flows linked to some of these transitions. With these estimations we can calculate the expected ultimate cost and the reserve which will finally allow us to have the one-year risk view with the help of the Claim Development Result. We will then compare our results with the initial database and with results obtained with the Chain Ladder method.

3.1 Database

3.1.1 Individual Claims History Simulation Machine

The used data for this numerical illustration are stochastically simulated data coming from the *Individual Claims History Simulation Machine* by Andrea Gabrielli and Mario V. Wüthrich (2018) [12].
This simulation machine is calibrated to real individual insurance claims using a neural network architecture. It then allows to simulate with a portfolio generating algorithm individual non-life claims history from the time it occurred until the time it is settled or after a maximum of 12 accounting years.
This tool allows us to take into account both payments and recoveries. These generated cash-flows are simulated using a log-normal distribution, with a standard deviation parameter which could differ depending on the nature of the cash-flow. Obviously, recoveries will never have a value greater than the sum of the previous cash-flows so that the cumulative payments are always positive or equal to zero.
This simulation machine is programmed in the R language. It can be downloaded from the following website : `https://people.math.ethz.ch/~wmario/simulation.html` .

We have next the input parameters for this tool as well as the used input parameters for our numerical illustration :

- V is the total expected number of claims. This number is equal to 50 000 for our simulation.
- *LoB.dist* gives us the distribution for the allocation of the claims to the 4 lines of business. We have decided to take only one line of business into account. All the weight has then been put on the first line of business, which gives us $LoB.dist = (1, 0, 0, 0)$.
- *inflation* define a growth for the number of claims in the 12 accident years. It has been decided that there will be no inflation for the number of claims. We have then taken this parameter equal to 0.
- *seed1* is the used seed for the simulation. We have $seed1 = 100$ for our simulation.
- *std1* is the standard deviation parameter in the log-normal distribution for the total claim amount simulation. We have taken $std1 = 1.85$.
- *std2* is the standard deviation parameter in the log-normal distribution for the recovery payments simulation. This parameter for our simulation is $std2 = 1.5$.

The output on the other hand are the following :

- *ClNr* is the claim ID and is unique for each individual claim.
- *LoB* is a number ranging from 1 to 4 in order to define the line of business of the claim. Obviously all claims have their LoB equal to 1 here as all the weight has been put on the first line of business.
- *cc* is a claim code and may take as value a number between 1 and 53.
- AY is the accident year, ranging from 1994 to 2005.
- AQ is the quarter of claims occurrence which can take value between 1 to 4. It defines the quarter in which the accident took place.
- *age* is the age of the insured and is ranging from 15 years old to 70 years old.
- *inj_part* is the part of the body that is injured. It could take value between 1 and 99 depending on the injured part.
- *RepDel* is the reporting delay of the claim and could take value between 0 and 11. This output allows us to find the number of years that each claim has spent in the IBNR state.
- Pay_j gives us the cash-flows paid in the development year j with $j = 0, 1, ..., 11$.

- *Open$_j$* indicate if the claim is open or closed at the end of the development year j with $j = 0, 1, ..., 11$. It takes 1 as a value if the claim is open and 0 if it is closed.

As for our simulation, the only outputs that we are interested in are the claim ID, the accident year, the reporting delay, the payments and finally the open parameter. We could then remove the LoB parameter, the claim code, the quarter of claims occurrence, the age of the insured and the injured part from the output as we don't use these parameters.

3.1.2 Data analysis

We will now analyse our data in order to see if some adjustments need to be made.

After a first glance at the output data, we can see that some closed claims could reopen as it can be seen in the following example :

ID	Open00	Open01	Open02	Open03	Open04	Open05	Open06	Open07	Open08	Open09	Open10	Open11
55	1	0	1	1	1	1	1	1	1	1	1	0

But as we have previously said, this should not be the case. We will then ensure that each claim stays open from the first development year and that until the final settlement. We can see in the A.1 appendix the code that allowed us to take this into account.

If we retake the previous example, we would then have :

ID	Open00	Open01	Open02	Open03	Open04	Open05	Open06	Open07	Open08	Open09	Open10	Open11
55	1	1	1	1	1	1	1	1	1	1	1	0

Another analysis shows us that some claims are not closed at the end of the development period.
Indeed, after the 12 development years, there is still 1187 claims that are open out of a total of 50313 claims. That is 2.36% of claims that are still open at the end of the development period.

In the next figure we can see the state in which the claims are going through as a percentage of the total number of claims :

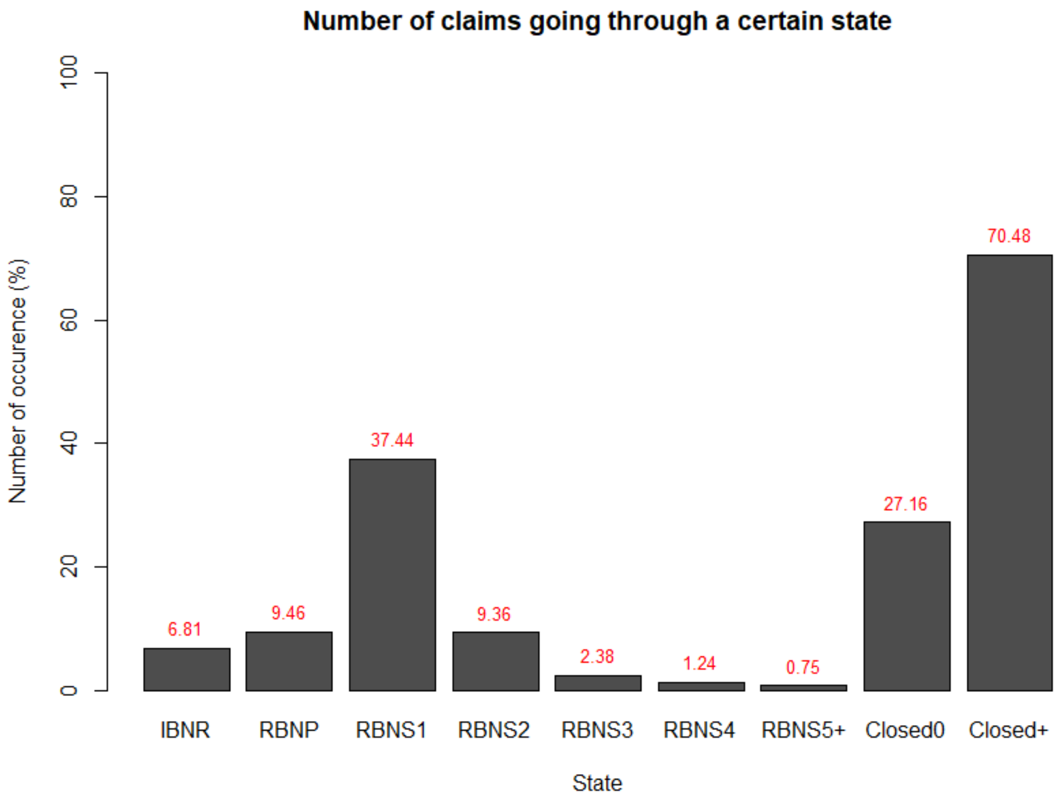

Figure 3.1.1: Number of claims going through a certain state.

At first, we can see that the sum of all these percentage is higher than 100%. That is because a claim could go through different states during it's lifetime.

We can also see that the more we advance in the RBNS states, the less there are claims that go through these states. We have then decided to regroup all claims that have five payments or more before final settlement in the RBNS5+ state as we already have only 0.75% of claims that have been through the RBNS5 state. We have then decided to limit ourselves to five RBNS states as the data for more would be insufficient in order to construct the transition probabilities for these states.

It can be seen that the final state of most claims is the Closed+ state, as 70.48% of claims end in this state. That could be explained by the fact that a lot of claims are closing after only one development year and, as few claims are closing without payment, the majority of these claims are then going to the Closed+ state.

Finally we can see that the sum of the Closed0 state percentage and of the Closed+ state percentage is equal to 97.64%. Indeed, as we have previously said, we have 2.36% of claims that are still open at the end of the development period, which means that the remaining 97.64% is settled at the end of the development period.

After some analyses, it has been decided to also aggregate data for which the time spent in some states is to high. It has then been decided for the RBNP state to aggregate the data for which the duration is greater than or equal to one year, whereas for the IBNR state it is for a duration greater than or equal to two years, and finally, for the RBNS states, the data have been aggregated for a duration greater than or equal to five years.

3.1.3 Augmented database

We will at first define two new variables :

- The *State* variable that correspond to the state in which the claim is at the end of the accounting year. We can retrieve the state of each claim at any time with the help of the R code available in the A.2 appendix. We can see in the code that we have indeed grouped together all claims with five payments or more before final settlement in the RBNS5+ state.
- The *Duration* variable which represents in years the time that the claim has spent on the current state. We can also retrieve the code to find the duration at any time and for each claim in the A.3 appendix. We can see in the code that we have taken into account that the duration for a claim in a RBNS state could not be greater than five years, whereas for the IBNR state it is two years and for the RBNP state it is only one year.

As a reminder, an artificial state has been defined with the help of two variables : the state in which the claim is and the time since when the claim is in this state. We can then determine the artificial state of each claim at any time with the help of these new variables. We therefore switch from a Semi-Markov model to a Markov model using these artificial states.

Finally we could build an augmented database with each line corresponding to a specific claim associated with a development year. We will add to this augmented database the *State* and the *Duration* variables defined hereabove as well as the *Transition* variable which is equal to 1 if there is a transition from one state to another and 0 otherwise. The construction of this database can be found in the A.4 appendix.

As an example we can see in the next table some lines of this augmented database :

ID	AY	DY	Payment	State	Duration	Transition
1	1994	1	1112	RBNS1	0	1
1	1994	2	0	Closed0	0	1
2	1994	1	914	Closed+	0	1
3	1994	1	904	Closed+	0	1
4	1994	1	0	RBNP	0	1
4	1994	2	273	RBNS1	0	1
4	1994	3	0	RBNS1	1	0
4	1994	4	0	RBNS1	2	0
4	1994	5	0	RBNS1	3	0
4	1994	6	0	Closed0	0	1

Table 3.1.1: Some lines of the augmented database.

We can see that the two first line of this augmented database are related to the same claim as they have the same claim ID. Indeed in it's first development year, the claim will transition from the IBNR state to the RBNS1 state. This transition will be accompanied with a payment equal to 1112. It will then transition from the RBNS1 state to the Closed0 state in it's second development year. As the claim is in the Closed0 state, there is no payment associated with this transition.

3.2 Estimation of transition probabilities and reporting lag

3.2.1 Claim occurrence rate and reporting lag

β calculation

At first let's calculate β_j which is the probability that a claim initially in the IBNR state is reported at lag j. For that we will use a Generalized Linear Model (GLM) with a Poisson distribution. This will be done using the *glm* function in the software R. The code for this calculation is available in the A.5 appendix.

We can see in the following table the results for a reporting delay of up to 5 years :

Reporting delay	Coefficient	β	Standard Deviation	p-value
0	-0.070694	0,931747	0,004619	$< 10^{-6}$
1	0.026943	0,065331	0,017440	0.012171
2	-0.488922	0,001670	0,109109	$< 10^{-6}$
3	-0.705145	0,000576	0,185700	0.000147
4	-0.683078	0,000318	0,249999	0.006298
5	-1.099378	0,000358	0,408242	$< 10^{-6}$

Table 3.2.1: Transition probabilities from the IBNR state depending on the reporting delay.

The β_j have been calculated with a log link function that we had to multiply to the number of claims having a reporting delay of j years divided by the total number of claims. As an example, for a reporting delay of 2 years, we have

$$\begin{aligned}\beta_2 &= \exp(-0.0707 - 0.4889) \times \frac{\text{Number of claims with } RepDel = 2}{\text{Number of recorded claims}} \\ &= 0.5714 \times 0.0029 \\ &= 0.001670.\end{aligned}$$

We can see from the table that the majority of claims leaves the IBNR state quite quickly, as 93.17% of claims are reported before their first development year and 99.71% are reported before the second development year.

α calculation

We also need to determine the α_i parameter which has been defined above as the expected number of claims for the accident year i. As the inflation for the number of claims has been taken as equal to zero, we expect that the number of claims is on average the same each year. We will still perform a Poisson regression in order to be sure. We can see the code of this Poisson regression in the A.6 appendix.

The results of this Poisson regression are available in this table :

Accident year	Coefficient	α	Standard Deviation	p-value
1994	8.337588	4178	0.015471	$< 10^{-6}$
1995	0.009291	4217	0.021829	0.670363
1996	-0.023981	4079	0.022012	0.275948
1997	-0.00962	4138	0.021932	0.660928
1998	0.015674	4244	0.021794	0.472036
1999	0.01095	4224	0.02182	0.615781
2000	0.001196	4183	0.021873	0.956392
2001	0.005728	4202	0.021848	0.793188
2002	0.026221	4289	0.021737	0.227712
2003	0.019202	4259	0.021775	0.377869
2004	-0.000958	4174	0.021884	0.965089
2005	-0.012524	4126	0.021948	0.568248

Table 3.2.2: Expected number of claims per accident year.

As expected, we can see that the expected numbers of claims per accident year are very close to each other. It has therefore been decided to take the same parameter α_i for each accident year. This parameter will be determined as the average number of claims from each accident year. We have then

$$\begin{aligned}\alpha_i &= \alpha \\ &= \frac{\text{Number of recorded claims}}{\text{Number of accident years}} \\ &= \frac{50313}{12} \\ &= 4192.75.\end{aligned}$$

We are expecting that there will be on average 4192.75 claims each year.

3.2.2 Transition probabilities and transition probability matrix

We will now focus on the transition probabilities. Transition probabilities from one artificial state to another are calculated using multiple Binomial GLM depending on the previous state. The used link function for these models is the canonical link function. We are only using one explanatory variable for this model which is the time spent in the previous state, or the *Duration* variable. This variable will be treated as a categorical feature in order to allow for more flexibility.

As we are using a Binomial GLM with a canonical link function, the transition probabilities will be calculated as

$$p = \frac{\exp(s)}{1 + \exp(s)}$$

with s being the regression coefficient associated with the transition.

We can then build the transition probability matrix as we have the transition probabilities from and to each artificial state. This transition probability matrix is available in the B.1 appendix.

We can see with the help of this transition probability matrix that we are indeed in a Semi-Markov model as the next state of a claim really depends on the current state and on the time that the claim has stayed in this state. As an example we could take the transition probabilities from the IBNR state to the RBNS1 state. We have

$$P(S_t = \text{RBNS1}|S_{t-1} = \text{IBNR}) = \begin{cases} 0.3203 & \text{for } t = 1 \\ 0.1517 & \text{for } t = 2 \\ 0.2781 & \text{for } t = 3. \end{cases}$$

We can then see that these probabilities are different depending on the time that the claim has stayed in the current state. Artificial states therefore remedy to this problem as the transition probabilities do not depend anymore on the time that the claim has stayed in the artificial state.

3.3 Estimation of cash-flows

We will now estimate cash-flows, and more precisely the first payment and the following link ratios. We will then need an appropriate distribution in order to simulate each process.

3.3.1 Discrete mixture model

As the payments simulated in the Simulation Machine (2018) [12] follow a log-normal distribution, it would have been logical to modelize the payment process also with a log-normal distribution. However this distribution does not give sufficiently accurate results.

It has then been decided to rather use a discrete mixture model combining two log-normal distribution as these distributions give more accurate results. We will then use the *gamlss* and the *gamlss.mx* packages, and more precisely the *gamlssMXfits* function in the software R in order to modelize the payment pattern with the help of these two log-normal distribution.
The *gamlssMXfits* function will fit a mixture of K distributions according to the database. In our case it has been decided that we will combine two log-normal distributions.

3.3.2 Estimation of the first payment

The first payment P_1 is the payment that occurs when a claim is leaving the IBNR or the RBNP state in order to go to the RBNS1 or to the Closed+ state.
As previously said, the first payment will be modelized with a mixture of two log-normal distributions. It has been decided to take the previous state as well as the duration as explanatory variables.

The parameters of these two distributions are displayed in the next table. These parameters can be found using the code available in the A.7 appendix.

	First log-normal distribution	**Second log-normal distribution**
Probability	0.502	0.498
	$\boldsymbol{\mu_1}$	$\boldsymbol{\mu_2}$
Intercept	5.0930	6.1825
Previous state = RBNP	0.6209	0.6504
$Duration = 1$	-0.0562	0.0029
$Duration \geq 2$	1.1496	-0.4090
	$\boldsymbol{\log(\sigma_1)}$	$\boldsymbol{\log(\sigma_2)}$
Intercept	0.5695	0.6903

Table 3.3.1: Estimated coefficients for the first payment.

As we can see, the first payment tends to be larger on average when claims goes through the RBNP state. It also tends to be larger for a duration being equal or larger than two. We could then compute the mean of the first payment with the help of these parameters.

We now want to assess the goodness of fit of the combined distribution for the first payment. We will then use two different methods in order to do that, one is a visual method whereas the other one is a formal uniformity test.

For the first one we will use a PP-plot. We will then plot the empirical cumulative distribution function of the observed first payments to see if these values follow an uniform distribution on the unit interval. Indeed, the cumulative distribution function of the unit uniform distribution follows a 45-degree line. The cumulative distribution function of the observed payments should then follows the same 45-degree line.

The PP-plot for the first payment is given by :

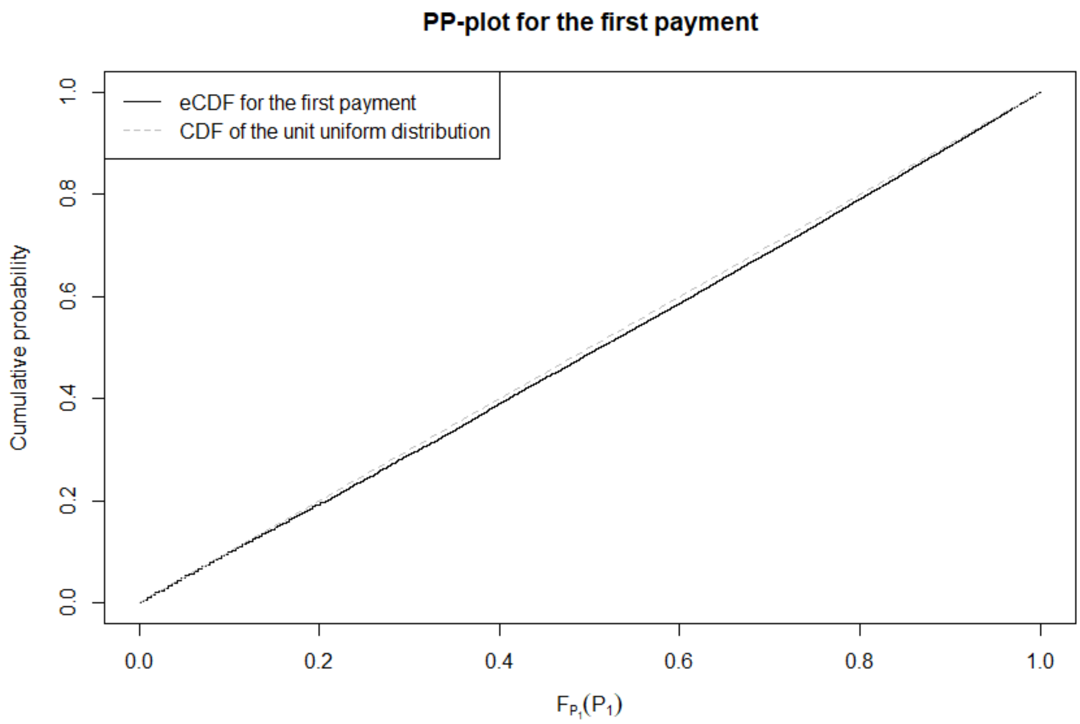

Figure 3.3.1: Empirical distribution function of the log-normal mixture model for the first payment P_1.

We can see in this plot that the empirical distribution function of the log-normal mixture model for the first payment practically merges with the cumulative distribution function of the unit uniform distribution, which means that the mixture model with the mix of the two log-normal distributions performs well.

We have also performed a Kolmogorov-Smirnov uniformity test with the help of the *kolmogorov.unif.test* function available in the *uniftest* package in the R software.
This test gave us a p-value equal to 9.30% which is above the usual 5% level. This result confirms that the used model performs well.

3.3.3 Estimation of link ratios and tie with the first payment

We have next histograms of the observed link-ratios :

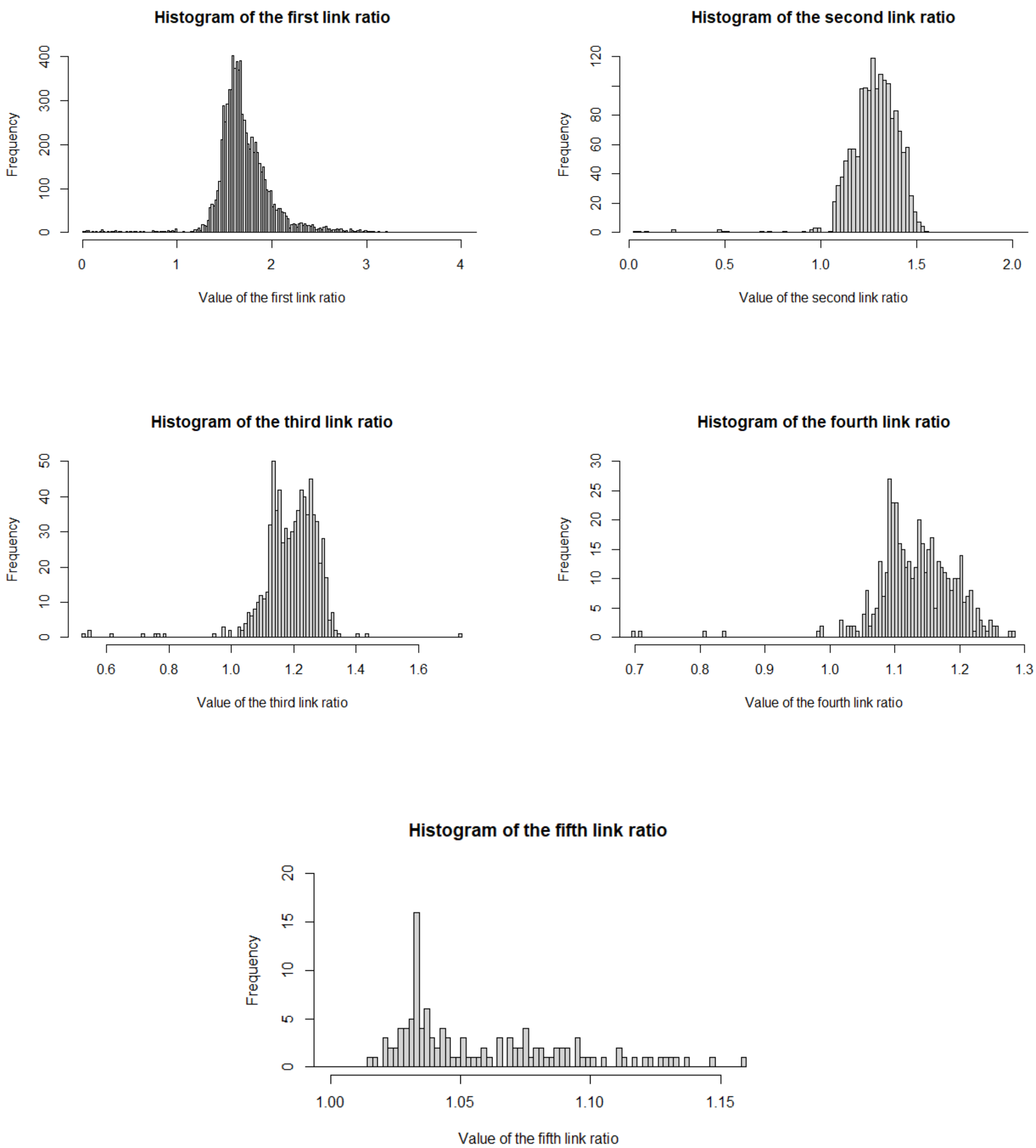

Figure 3.3.2: Histograms of observed link ratios.

As previously mentioned, the link ratios will also be modelized with a discrete mixture of two log-normal distributions.

However, a filter on the data will have to be made in order to avoid unrealistically large payments. Indeed, we could expect that a large first payment will be followed with smaller link ratios, and vice versa as shown in the figure below :

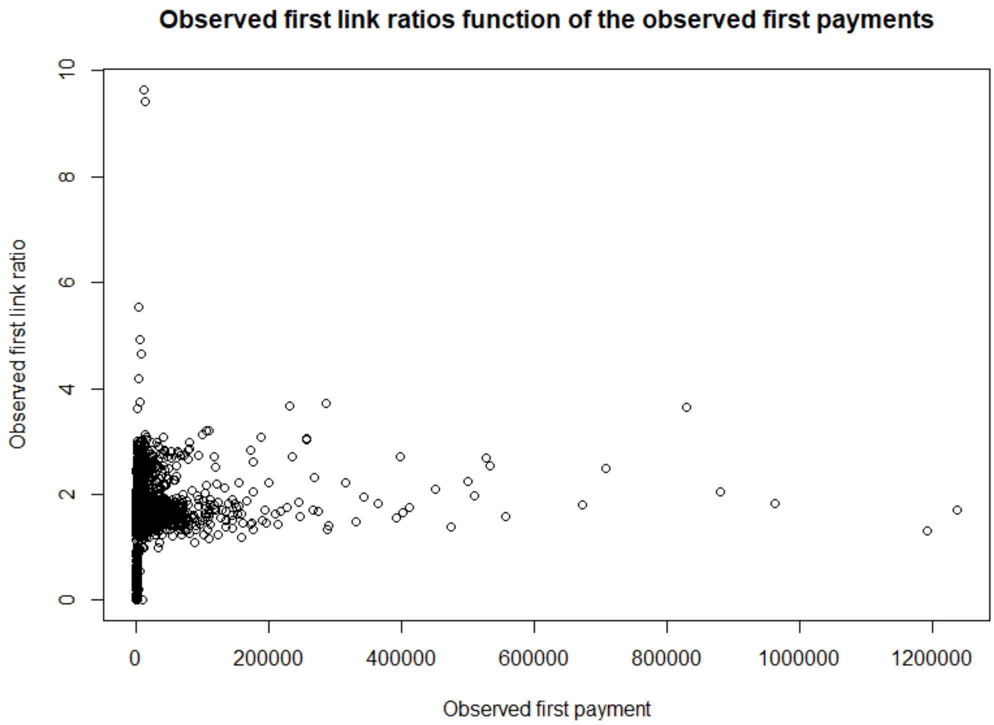

Figure 3.3.3: Observed first link ratios function of observed first payments.

The data will then be divided into two groups depending on the amount of the first payment. The first group corresponds to claims for which the first payment is lower than 100 000 whereas the second one will be for claims whose first payment is greater than 100 000.

In the following tables we can then see the parameters of the mixture model for the first link ratios and the following ones. These parameters will depend on the duration and on the value of the first payment. These parameters can be found once again with the help of the code in the A.7 appendix.

	First log-normal distribution	**Second log-normal distribution**
Probability	0.0508	0.9492
	$\boldsymbol{\mu_1}$	$\boldsymbol{\mu_2}$
Intercept	0.3200	0.5262
$Duration = 1$	-0.5701	-0.0242
$Duration = 2$	-2.8058	-0.0151
$Duration = 3$	-0.4910	-0.0392
$Duration = 4$	0.2155	-0.0242
$Duration \geq 5$	0.3566	-0.0101
$P1 \geq 100000$	0.6877	-0.0013
	$\boldsymbol{\log(\sigma_1)}$	$\boldsymbol{\log(\sigma_2)}$
Intercept	-0.0098	-2.1293

Table 3.3.2: Estimated coefficients for the first link ratio.

	First log-normal distribution	**Second log-normal distribution**
Probability	0.9890	0.0110
	$\boldsymbol{\mu_1}$	$\boldsymbol{\mu_2}$
Intercept	0.2521	-0.5413
$Duration = 1$	0.0001	-0.4917
$Duration = 2$	0.0002	-0.8701
$Duration = 3$	-0.0083	0.5424
$Duration = 4$	-0.0042	-0.3388
$Duration \geq 5$	-0.0340	0.6660
$P1 \geq 100000$	-0.0626	0.7020
	$\boldsymbol{\log(\sigma_1)}$	$\boldsymbol{\log(\sigma_2)}$
Intercept	-2.5170	0.1208

Table 3.3.3: Estimated coefficients for the second link ratio.

	First log-normal distribution	**Second log-normal distribution**
Probability	0.9738	0.0262
	$\boldsymbol{\mu_1}$	$\boldsymbol{\mu_2}$
Intercept	0.1840	-0.0680
$Duration = 1$	0.0180	-0.2946
$Duration = 2$	0.0132	0.1774
$Duration = 3$	0.0208	0.2399
$Duration = 4$	-0.0237	0.2104
$Duration \geq 5$	-0.0191	0.2075
$P1 \geq 100000$	-0.0762	0.1533
	$\boldsymbol{\log(\sigma_1)}$	$\boldsymbol{\log(\sigma_2)}$
Intercept	-2.9569	-1.2191

Table 3.3.4: Estimated coefficients for the third link ratio.

	First log-normal distribution	**Second log-normal distribution**
Probability	0.8238	0.1762
	$\boldsymbol{\mu_1}$	$\boldsymbol{\mu_2}$
Intercept	0.5038	0.1250
$Duration = 1$	0.0195	0.0527
$Duration = 2$	0.0365	-0.0962
$Duration = 3$	0.0105	0.0277
$Duration = 4$	-0.3729	0.1429
$Duration \geq 5$	0.0329	-0.2281
$P1 \geq 100000$	-0.0571	-0.0710
	$\mathbf{log}\,(\boldsymbol{\sigma_1})$	$\mathbf{log}\,(\boldsymbol{\sigma_2})$
Intercept	-3.2521	-1.5309

Table 3.3.5: Estimated coefficients for the fourth link ratio.

	First log-normal distribution	**Second log-normal distribution**
Probability	0.9281	0.0719
	$\boldsymbol{\mu_1}$	$\boldsymbol{\mu_2}$
Intercept	0.0827	-0.2413
$Duration = 1$	-0.0082	0.3134
$Duration = 2$	-0.0154	0.3076
$Duration = 3$	-0.0220	0.2332
$Duration = 4$	-0.0229	0.2902
$Duration \geq 5$	-0.0376	-1.5499
$P1 \geq 100000$	-0.0308	1.8231
	$\mathbf{log}\,(\boldsymbol{\sigma_1})$	$\mathbf{log}\,(\boldsymbol{\sigma_2})$
Intercept	-3.6881	-10.0680

Table 3.3.6: Estimated coefficients for the fifth link ratio.

We could then compute the average of the different link ratios according to these parameters.

We will also use PP-plots and the Kolmogorov-Smirnov test in order to see if each of these models gives us accurate results.

We can see in the next figures the PP-plots of the empirical distribution function of the observed five link-ratios :

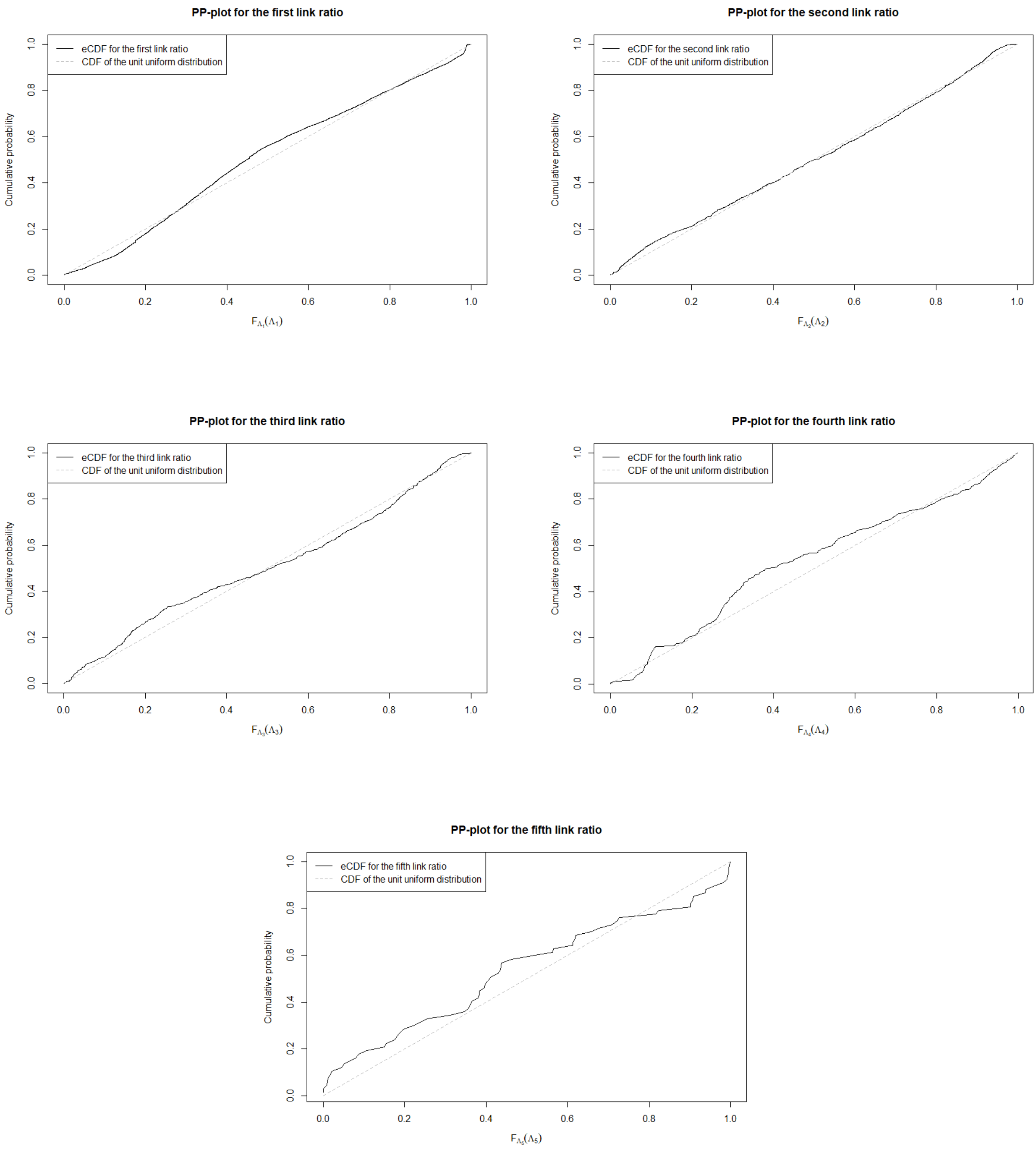

Figure 3.3.4: Empirical distribution functions of the log-normal mixture models for the link ratios Λ_j with $j = 1, 2, ..., 5$.

We can see once again that the empirical distribution function for these models is close to the 45-degree line, which means that our models perform well with the data.
However, we can see that the empirical distribution function tends to move away from the 45-degree line for higher link ratios. That could be explained by the fact that we have less data for the last link ratios, which means that the performance of these models is slightly worse than for the previous ones, but it remains nevertheless acceptable.

We have also performed a Kolmogorov-Smirnov uniformity test on these five models. We can see the p-value of these tests in the following table :

	Λ_1	Λ_2	Λ_3	Λ_4	Λ_5
p-value	87.20%	70.85%	69.05%	51.10%	64.70%

Table 3.3.7: p-value of the Kolmogorov-Smirnov uniformity test for the five link ratios.

We can see once again that these values are well above the 5% level, which means that the different models fit well.

3.4 Expected ultimate cost and one-year risk view

We will here calculate the expected ultimate cost which will allow us to retrieve the reserve for all claims and then we will calculate the Claim Development Result in order to have the one-year risk view. We will also compare our result with a Chain-Ladder model.

3.4.1 $\tilde{M}$ matrix

Now that we have the transition probability matrix as well as the mean of the first payment and of the following link ratios, we can build the $\tilde{M}$ matrix which has been previously defined as the transition probability matrix M multiplied with the mean of the first payment or with the mean of one of the following link ratio depending on the transition that takes place.

We will here define two different $\tilde{M}$ matrices depending on the value of the first payment. The first one $\tilde{M}_{P_1<100000}$ is defined as the transition probability matrix M multiplied with the mean of the first payment or with the mean of the following link ratios knowing that the first payment is lower than 100 000. As its name suggests, this matrix is used for claims where the first payment is lower than 100 000.
The second one $\tilde{M}_{P_1\geq 100000}$ on the other hand is also defined as the transition probability matrix but this time it is multiplied with the mean of the first payment or with the mean of the following link ratios knowing that the first payment is greater than 100 000. This matrix is for claims where the first payment is greater than 100 000.
The used matrix will then depends on the value of the first payment. We can see these

two matrices in the B.2 appendix.

However we have a problem for claims that are in the IBNR or in the RBNP state. Indeed, if a claim is in one of these two states, we do not know the value of the first payment as the first payment has not yet occurred. The used matrix will then depend on the probability that the first payment is lower or greater than 100 000 knowing the previous state. We can see in the following table these probabilities depending on the previous state of the claim :

	IBNR 0	IBNR 1	IBNR 2	RBNP 0	RBNP 1
$P[P_1 < 100000]$	0.997334	0.998092	1	0.995085	0.98
$P[P_1 \geq 100000]$	0.002666	0.001908	0	0.004915	0.02

Table 3.4.1: Probabilities that the first payment P_1 is lower or greater than 100 000 depending on the previous state of the claim.

As we could expect, we can see that the majority of claims are using the matrix where the first payment is lower than 100 000 as there are far more first payment lower than 100 000 than first payment greater than 100 000.

3.4.2 Expected ultimate cost and reserve calculation

As previously said, the reserve will be calculated analytically in order to avoid simulation within simulation errors.

We need then at first to calculate the ultimate cost in order to retrieve the reserve.
The expected ultimate cost will be calculated in the same way as it has been described in the 2.6 section.

The expected ultimate cost for all claims that are in the IBNR state in the i^{th} accident year is given by

$$UC^{expected} = \alpha_i \Big(P[P_1 < 100000] \times \sum_{j \geq j_0+1} \beta_j (\omega_{IBNR}^{\top} \tilde{M}^T_{P_1<100000} \mathbf{1}) + P[P_1 \geq 100000] \times \sum_{j \geq j_0+1} \beta_j (\omega_{IBNR}^{\top} \tilde{M}^T_{P_1 \geq 100000} \mathbf{1}) \Big)$$

with the different variables also defined in the 2.6 section. We can obviously see here that the $\tilde{M}$ matrix will differ depending on the probability to have a first payment lower or greater than 100 000.

For a claim in the RBNP state, we have

$$\begin{aligned} UC^{expected} = P[P_1 < 100000] \times \omega_{RBNP}^{\top} \tilde{M}_{P_1<100000}^{T} \mathbf{1} \\ + P[P_1 \geq 100000] \times \omega_{RBNP}^{\top} \tilde{M}_{P_1 \geq 100000}^{T} \mathbf{1}. \end{aligned}$$

We can see once again that the $\tilde{M}$ matrix will differ depending on the probability to have a first payment lower or greater than 100 000.

Finally we have two different cases for claims in the RBNS state depending once again on the value of the first payment. For a claim in the RBNS$_j$ state with a first payment lower than 100 000, we have

$$UC^{expected} = P_1 \left(\prod_{l=1}^{j-1} \Lambda_l \right) (\omega_{RBNS_j}^{\top} \tilde{M}_{P_1<100000}^{T} \mathbf{1})$$

whereas for a claim in the RBNS$_j$ state with a first payment greater than 100 000, we have

$$UC^{expected} = P_1 \left(\prod_{l=1}^{j-1} \Lambda_l \right) (\omega_{RBNS_j}^{\top} \tilde{M}_{P_1 \geq 100000}^{T} \mathbf{1}).$$

We can now calculate the reserve as the reserve is defined as the expected ultimate cost for all claims at which we will withdraw payments that have already been made.

We can see in the following table the database reserve, the reserve calculated with the analytical method described previously and the reserve calculated with a classical Chain Ladder method. We recall that some claims are still open after the 12 development years, the reserve for the database as well as the reserve calculated using the Chain-Ladder method will therefore be slightly underestimated because of this. The reserve calculated with the Chain-Ladder method is calculated using the *ChainLadder* package in R. We can see in the A.8 appendix the code that allowed us to find this reserve.

	Database	Analytical method	Chain Ladder
Reserve for open claims	23 745 212	22 388 684	-
Reserve for IBNR claims	756 151	879 776	-
Total reserve	24 501 363	23 261 932	24 939 749

We can see from this table that the total reserve for the analytical method is a bit smaller than the actual reserve whereas for the Chain Ladder method, the reserve is a bit larger.

We can still see that the reserves calculated with these two methods are very close to the database reserve. We can therefore consider that these two models provide consistent results, even though they tend to slightly underestimate the actual reserve.

3.4.3 Simulations

In order to calculate the Claim Development Result, we have to simulate the transition as well as the payments associated with some of these transitions for the coming year.

Transitions simulation

We will at first simulate the transition of open claims from an artificial state to another one. In order to do that, we will create a cumulative distribution matrix based on the transition probability matrix.
After that we will take a random number between 0 and 1. Depending on the value of this number and of the cumulative distribution matrix, a certain transition will take place.

We will take the transition from the RBNP 0 artificial state as an example. As a reminder, the transition probabilities for this state are given by

Next artificial state	RBNS1 0	Closed+	Closed0	RBNP 1
Transition probability	40.08%	55.82%	2.70%	1.39%
Cumulative transition probability	40.08%	95.90%	98.61%	100.00%

Table 3.4.2: Transition probabilities from the RBNP 0 artificial state.

We can of course see that only the non-zero transition probabilities have been put in this table in order to have a better visibility of it.

We have in the next figure the cumulative distribution function of the transition probabilities from the RBNP 0 artificial state :

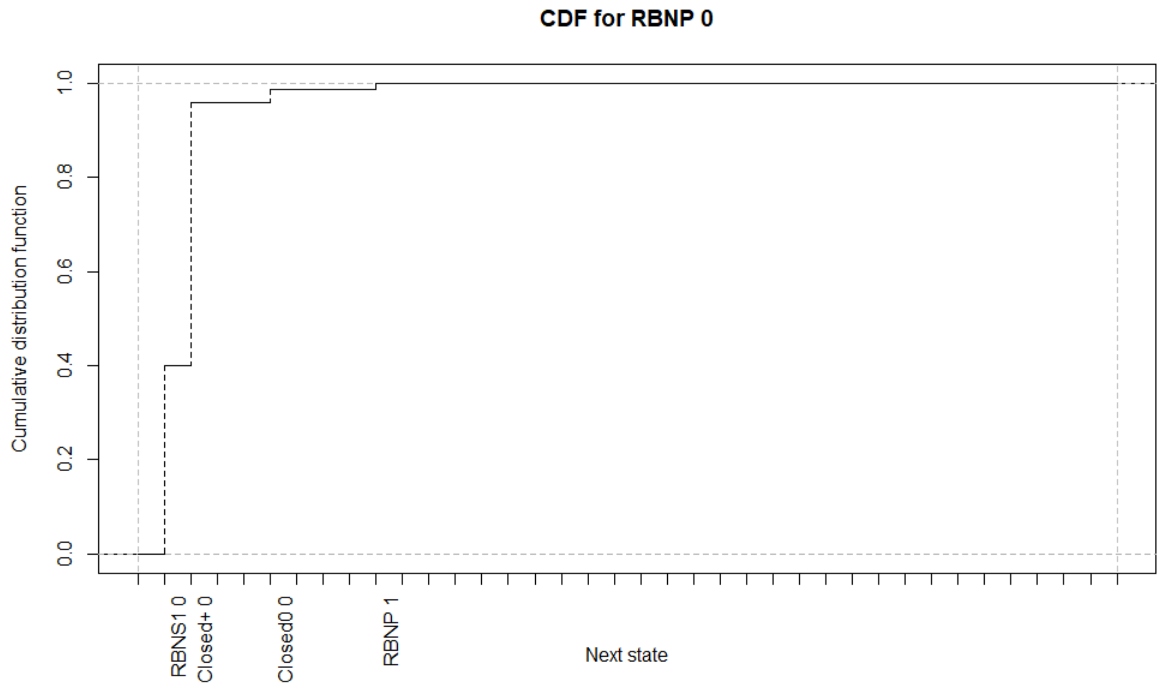

Figure 3.4.1: Cumulative distribution function of the transition probabilities for the RBNP 0 artificial state.

We can then see that if the random number is between 0 and 0.4008, the transition that takes place is the transition to the RBNS1 0 state, whereas if it is between 0.4008 and 0.9590 the transition that takes place is to the Closed+ state. After that, if the number is between 0.9590 and 0.9861, there will be a transition to the Closed0 state and finally, if the number is between 0.9861 and 1, the transition will be to the RBNP 1 state.

We will now focus on the transition of claims that are in the IBNR state. For that we have previously run a Poisson regression in order to determine the β_j coefficients and we have also determined the α coefficient previously. These coefficient will allow us to determine the number of claims which will be reported on the next year for each previous accident year.
After that, these claims will transition from their IBNR artificial state to another open artificial state. These transitions will be determined in the same way as it has been done for the open claims with the cumulative distribution function of the transition probabilities and with the random number between 0 and 1.

We can see in the A.9 appendix the R code that will simulate the transitions for claims that are reported as well as for claims that are in the IBNR state.

Payments simulations

We will now simulate the first payment or simulate link ratios depending on the transition that takes place. If there is a transition to a ($RBNS_i$, 0) or to a Closed+ state then there will be a payment associated with this transition. We could simulate this payment as previously with the help of the cumulative distribution function of the discrete mixture model of the first payment or of link ratios described previously.

As an example we have the cumulative distribution function for the first payment where the previous artificial state is the (IBNR, 0) state :

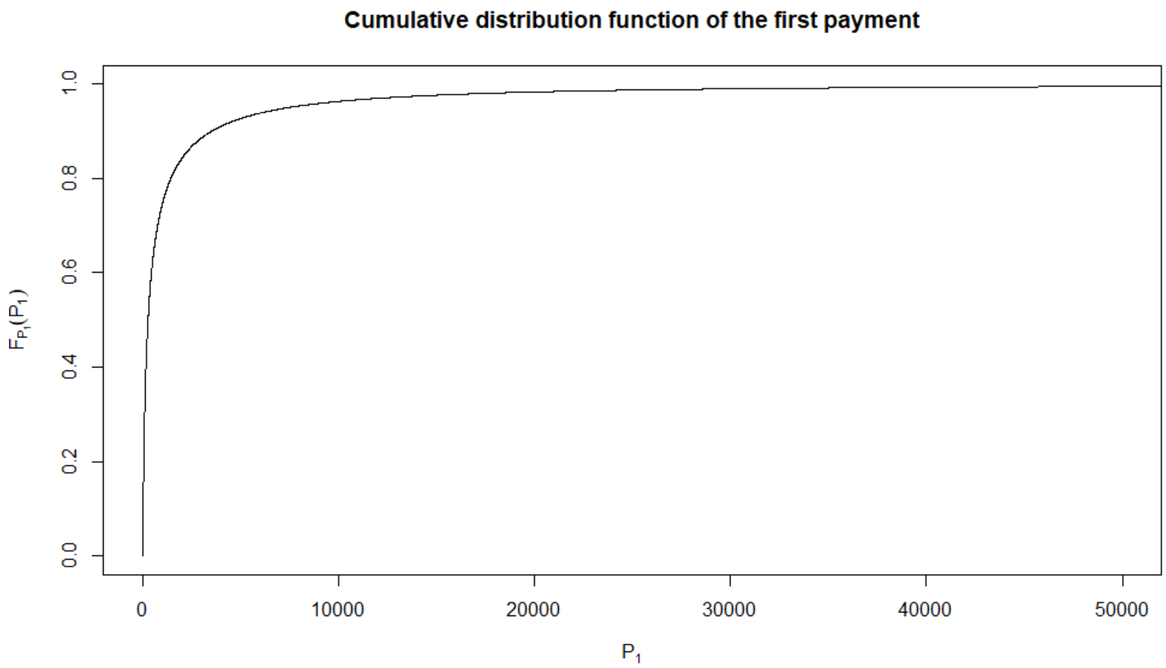

Figure 3.4.2: Cumulative distribution function of the first payment from the (IBNR, 0) artificial state.

For instance, if the random number is equal to 0.9, then the simulation of the first payment will be equal to 3524.76.

We can once again see the R code in the A.10 appendix that simulates payments and link ratios for the following year. We have only taken as an example the simulation for the first link ratio knowing that claims are in the ($RBNS_1$, 2) artificial state at the beginning of the year with a first payment smaller than 100 000 and the simulation for the first payment knowing that claims are in the (IBNR, 0) artificial state at the beginning of the year.

3.4.4 Claim Development Result

Within the Solvency II framework we are interested in the one-year risk view. We then want to calculate the Claim Development Result. In order to do that, we need at first to calculate the reserve. This calculation was done analytically as described previously. We will then need to simulate transitions as well as payments that are associated with some of these transitions for the following year. This has been done as described previously. After these transitions we could once again calculate the reserve at the end of the year.

Finally, we can calculate the Claim Development Result as it is defined as

$$\begin{aligned} CDR = &\ \text{Reserve at the beginning of the year} \\ &- \text{Reserve at the end of the year} \\ &- \text{Payments occurred during the year.} \end{aligned}$$

We will here perform 2000 simulations. We could then retrieve the average CDR as well as the Value-at-Risk for some quantile. We will compare these results with the one obtained with the Chain Ladder method using the *CDR.BootChainLadder* function in the *ChainLadder* package in the software R.

	Artificial states model	Chain Ladder
$E[CDR]$	-496 973	-628 621
$\text{VaR}_{75\%}(CDR)$	-1 950 810	-1 302 628
$\text{VaR}_{95\%}(CDR)$	-4 098 606	-3 967 957
$\text{VaR}_{99.5\%}(CDR)$	-6 000 484	-6 940 663

We also have in the following plot the empirical distribution of the Claim Development Result for 2000 simulations :

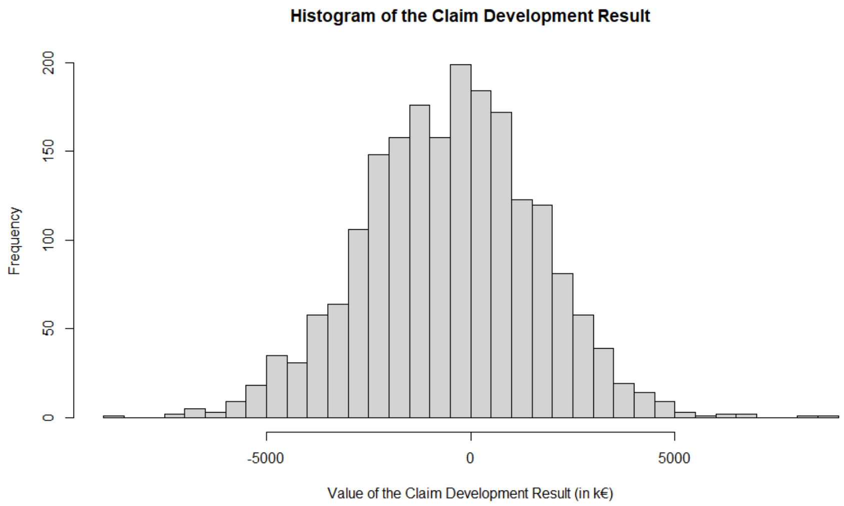

Figure 3.4.3: Empirical distribution of the Claim Development Result for 2000 simulations.

As we can see, the average Claim Development Result is slightly lower than zero for these two methods. This result should theoretically be equal to zero but we can see in the empirical distribution function that this value is quite close to zero. We can also see that the different Value-at-Risks for the two methods are close to each other. We can therefore estimate that our model provides us with consistent results.

The insurance company will therefore have to put 6 000 484 aside in order to be sure of being 99.5% solvent for the coming year if we refer to the Semi-Markov model, whereas for the Chain Ladder model the company will have to put aside 6 940 663 in order to remain solvent at 99.5%.

Chapter 4

Conclusion

4.1 Summary of the project

In this work we have developed a Semi-Markov model in order to model the claim settlement process. We have at first developed the theoretical part of this model which has been illustrated with a simple example. It has then been decided to implement the Semi-Markov model on more than 50 000 claims in order to illustrate the model numerically. Finally a comparison has been made between the initial database, the results obtained with this model and the results obtained with the Chain-Ladder method.

We can see that there are advantages and disadvantages to the use of this kind of model. The main advantage is that it can take most of data characteristics into account, as for example the state in which the claim is or the time spent in this state and so on. We can also follow the settlement process of each claim individually. It is not possible for the Chain-Ladder method as well as for aggregated methods to take into account such things as all claims are aggregated.
On the other hand there is also some negative points that need to be mentioned. The first one is that this kind of models is way more complicated. It then needs someone with a real actuary qualification in order to understand and more specifically to implement this model compared to the Chain-Ladder method. There is also a need of a lot of computing power, as for only 2000 simulations it has taken more than 15 hours in order to have results for this model. It also requires a large database in order to train the model initially as there is a need to modelize all the different possible transitions as well as the cash-flows associated with some of these transitions.

To conclude, we can say that this model is interesting for some cases were for example it is needed to know the claim settlement process for each claim or for very specific claims. Nevertheless it is still easier to use aggregate methods for more general cases where we do not have to necessarily know the claim settlement process of each claim individually.

4.2 Further possible research

However, as we can imagine, it is always possible to improve a model. As possible additions to this model we could then

- introduce an inflation parameter, whether it is an inflation on the number of claims according to the accident year or a monetary inflation.
- take into account the fact that a claim could reopen even after it has been closed.
- use more explanatory variables in order to determine the cash-flows, as for example the injured body part or the age of the insured. However, there is a problem with taking into account more explanatory variables in our case as the initial database must be much larger.

Appendix A

R code

A.1 No reopening of claims

```
# We create an Open variable which takes the open parameter
# from the simulation machine
Open <- input[,21:32]
# We check for the greatest development year were the claim is still open
# and we ensure that the claim is open all the years before that year
for (i in 1:N){
  for (j in 1:11){
    if (Open[i,(13-j)] == 1){
      Open[i,(12-j)] <- 1
    }
  }
}
```

A.2 Determination of the *State* variable

```
### State determination
State <- matrix(nrow = N, ncol = 12)
for(i in 1:N){
  # IBNR and Closed claims
  if(input$RepDel[i] != 0){
    State[i,1:input$RepDel[i]] <- "IBNR"
  }
  if(YOpen[i] != 12){
    if(Pay[i,(YOpen[i]+1)] == 0){
      State[i,(YOpen[i]+1):12] <- "Closed0"
    }
    if(Pay[i,(YOpen[i]+1)] != 0){
```

```
    State[i,(YOpen[i]+1):12] <- "Closed+"
  }
}
# RBNP claims
for (j in 1:12){
  if(is.na(State[i,j]) == TRUE & CumPay[i,j] == 0){
    State[i,j] <- "RBNP"
  }
}
# RBNS1 claims for the first development year
if(is.na(State[i,1]) == TRUE & Pay[i,1] != 0){
  State[i,1] <- "RBNS1"
}
# RBNS1 to RBNS5+ claims for the following years
for (j in 1:11){
  if(is.na(State[i,j+1]) == TRUE & is.na(State[i,j]) == FALSE
      & ((Pay[i,j+1] != 0 & (State[i,j] == "IBNR" | State[i,j] == "RBNP"))
      | (Pay[i,j+1] == 0 &  State[i,j] == "RBNS1"))){
        State[i,j+1] <- "RBNS1"
  }
  if(is.na(State[i,j+1]) == TRUE & is.na(State[i,j]) == FALSE
      & ((Pay[i,j+1] != 0 & State[i,j] == "RBNS1")
      | (Pay[i,j+1] == 0 & State[i,j] == "RBNS2"))){
        State[i,j+1] <- "RBNS2"
  }
  if(is.na(State[i,j+1]) == TRUE & is.na(State[i,j]) == FALSE
      & ((Pay[i,j+1] != 0 & State[i,j] == "RBNS2")
      | (Pay[i,j+1] == 0 & State[i,j] == "RBNS3"))){
        State[i,j+1] <- "RBNS3"
  }
  if(is.na(State[i,j+1]) == TRUE & is.na(State[i,j]) == FALSE
      & ((Pay[i,j+1] != 0 & State[i,j] == "RBNS3")
      | (Pay[i,j+1] == 0 & State[i,j] == "RBNS4"))){
        State[i,j+1] <- "RBNS4"
  }
  if(is.na(State[i,j+1]) == TRUE & is.na(State[i,j]) == FALSE
      & ((Pay[i,j+1] != 0 & State[i,j] == "RBNS4")
      | (Pay[i,j+1] == 0 & State[i,j] == "RBNS5"))){
        State[i,j+1] <- "RBNS5+"
  }
  # We regroup the last payments in the RBNS5+ state
  if(is.na(State[i,j+1]) == TRUE){
    State[i,j+1] <- "RBNS5+"
  }
}
```

```
}
```

$YOpen[i]$ is the number of year that the claim i stays open and $CumPay[i,j]$ is the cumulative payment for the claim i at development year j.

A.3 Determination of the *Duration* variable

The duration is determined according to the previous state of the claim.

```
### Duration determination
Duration <- matrix(0, nrow = N, ncol=12)
for (i in 1:N){
  if(State[i,1] != "IBNR"){
    Duration[i,1] <- 0
  }
  if(State[i,1] == "IBNR"){
    Duration[i,1] <- 1
  }
  for (j in 1:11){
    if (State[i,j+1] != State[i,j]){
      Duration[i,j+1] <- 0
      }
    if (State[i,j+1] == State[i,j] & Duration[i,j] < 5){
      Duration[i,j+1] <- Duration[i,j]+1
    }
    # Maximum duration of 5 years
    if (State[i,j+1] == State[i,j] & Duration[i,j] >= 5){
      Duration[i,j+1] <- Duration[i,j]
    }
  }
}
# Maximum duration for the IBNR and the RBNP states
for(i in 1:N){
  for(j in 1:12){
    if(State[i,j] == "IBNR" & Duration[i,j]>2){
      Duration[i,j] <- 2
    }
    if(State[i,j] == "RBNP" & Duration[i,j]>1){
      Duration[i,j] <- 1
    }
  }
}
```

A.4 Build of the augmented database

Some columns in this augmented database are here to ease future calculation. The 3.1.1 table only takes into account the "ID", the "AY", the "DY", the "Pay", the "Current_State", the "Duration" and the "Transition" columns into account.

```
### Augmented database
DATA <- matrix(nrow = N*12, ncol = 13)
DATA <- data.frame(DATA)
colnames(DATA) <- c("ID", "AY", "DY", "CumPay", "Pay", "Previous_State",
                    "Current_State", "Previous_Duration", "Duration",
                    "Transition", "FirstPayment", "LR", "FirstPaymentValue")

# Value of the first payment
DATA$FirstPaymentValue <- rep(FirstPaymentValue, each=12)
# Variable that tells us if the first payment is
# greater or smaller than 100k
DATA$FirstPayment <- rep(FirstPayment, each=12)
DATA$ID <- rep(1:N, each=12)
DATA$AY <- rep(input[,4], each=12)
DATA$DY <- rep(1:12,N)
# Cumulative payment
CumPay <- as.matrix(CumPay)
DATA$CumPay <- c(t(CumPay))
DATA$Pay <- c(t(Pay))
DATA$Current_State <- c(t(State))
DATA$Duration <- c(t(Duration))

# Transition for the first dvp year
for (i in 1:N){
  if (DATA$Current_State[1+12*(i-1)] != "IBNR"){
    DATA$Transition[1+12*(i-1)] <- 1
  }
  if(DATA$Current_State[1+12*(i-1)] == "IBNR"){
    DATA$Transition[1+12*(i-1)] <- 0
  }
}

# Transition for the following years
for(i in 1:(12*N-1)){
  if(DATA$Current_State[i+1] != DATA$Current_State[i] &
     is.na(DATA$Transition[i+1]) == TRUE){
      DATA$Transition[i+1] <- 1
  }
  if(DATA$Current_State[i+1] == DATA$Current_State[i] &
     is.na(DATA$Transition[i+1]) == TRUE){
```

```
      DATA$Transition[i+1] <- 0
  }
}
# Determination of the previous state and the previous duration
DATA$Previous_State[2:(12*N)] <- DATA$Current_State[1:(12*N-1)]
DATA$Previous_Duration[2:(12*N)] <- DATA$Duration[1:(12*N-1)]
for (i in 1:N){
  DATA$Previous_State[1+12*(i-1)] <- "IBNR"
  DATA$Previous_Duration[1+12*(i-1)] <- 0
}

# We remove the closed claims for more than a year
# in order to reduce the size of the database
DATA <- subset(DATA,(Current_State != "Closed+" &
                     Current_State != "Closed0") | Duration == 0)
rownames(DATA) <- c(1:length(DATA[,1]))
```

A.5 Reporting lag

```
### Calculation of the beta_j
# Poisson regression
glm.IBNR <- glm(subset(DATA,Previous_State == "IBNR")$Transition~
                subset(DATA,Previous_State == "IBNR")$Previous_Duration,
                family = poisson())

# Calculation of the Beta
Beta0 <- exp(coef(glm.IBNR)[1])
Beta1 <- exp(coef(glm.IBNR)[1]+coef(glm.IBNR)[2])*
         length(subset(DATA,Previous_State == "IBNR" &
                Previous_Duration == 1)[,1])/length(input[,1])
Beta2 <- exp(coef(glm.IBNR)[1]+coef(glm.IBNR)[3])*
         length(subset(DATA,Previous_State == "IBNR" &
                Previous_Duration == 2)[,1])/length(input[,1])
Beta3 <- exp(coef(glm.IBNR)[1]+coef(glm.IBNR)[4])*
         length(subset(DATA,Previous_State == "IBNR" &
                Previous_Duration == 3)[,1])/length(input[,1])
Beta4 <- exp(coef(glm.IBNR)[1]+coef(glm.IBNR)[5])*
         length(subset(DATA,Previous_State == "IBNR" &
                Previous_Duration == 4)[,1])/length(input[,1])
Beta5 <- exp(coef(glm.IBNR)[1]+coef(glm.IBNR)[6])*
         length(subset(DATA,Previous_State == "IBNR" &
                Previous_Duration == 5)[,1])/length(input[,1])
```

A.6 Expected number of claims per year

```
### Determination of the alpha
N1994 <- length(subset(input, AY == 1994)[,1])
N1995 <- length(subset(input, AY == 1995)[,1])
N1996 <- length(subset(input, AY == 1996)[,1])
N1997 <- length(subset(input, AY == 1997)[,1])
N1998 <- length(subset(input, AY == 1998)[,1])
N1999 <- length(subset(input, AY == 1999)[,1])
N2000 <- length(subset(input, AY == 2000)[,1])
N2001 <- length(subset(input, AY == 2001)[,1])
N2002 <- length(subset(input, AY == 2002)[,1])
N2003 <- length(subset(input, AY == 2003)[,1])
N2004 <- length(subset(input, AY == 2004)[,1])
N2005 <- length(subset(input, AY == 2005)[,1])
N_AY <- as.data.frame(cbind(c(N1994,N1995,N1996,N1997,N1998,N1999,N2000,
                              N2001,N2002,N2003,N2004,N2005),1994:2005))
colnames(N_AY) <- c("N", "AY")
N_AY$AY <- as.character(N_AY$AY)
# Poisson regression
glm.alpha <- glm(data = N_AY, N_AY$N~N_AY$AY, family = poisson())
```

A.7 Cash-flows estimation

In order to estimate the cash-flows we will at first take lines of the augmented database which are linked to the first payment or to the link ratio that we want to estimate. After that we will use the *gamlssMXfits* function in order to estimate the probability and the coefficients of the two log-normal distribution.

```
### Payment process

# First payment
P1DATA <- subset(DATA,(Previous_State == "IBNR" | Previous_State == "RBNP")
                 & Transition == 1 & (Current_State == "RBNS1" |
                 Current_State == "Closed+"))
P1 <- gamlssMXfits(n=10,
        P1DATA$CumPay~P1DATA$Previous_State+P1DATA$Previous_Duration,
        data = P1DATA, family = LOGNO, K=2)

# First link ratio
Lambda1DATA <- subset(DATA, Previous_State == "RBNS1" &
                      Transition == 1 & Current_State != "Closed0")
Lambda1 <- gamlssMXfits(n=10,
                Lambda1DATA$LR~Lambda1DATA$Previous_Duration+
```

```
                Lambda1DATA$FirstPayment,
                data = Lambda1DATA, family = LOGNO, K=2)

# Second link ratio
Lambda2DATA <- subset(DATA,Previous_State == "RBNS2" &
                           Transition == 1 & Current_State != "Closed0")
Lambda2 <- gamlssMXfits(n=10,
                Lambda2DATA$LR~Lambda2DATA$Previous_Duration+
                Lambda2DATA$FirstPayment,
                data = Lambda2DATA, family = LOGNO, K=2)

# Third link ratio
Lambda3DATA <- subset(DATA,Previous_State == "RBNS3" &
                           Transition == 1 & Current_State != "Closed0")
Lambda3 <- gamlssMXfits(n=10,
                Lambda3DATA$LR~Lambda3DATA$Previous_Duration+
                Lambda3DATA$FirstPayment,
                data = Lambda3DATA, family = LOGNO, K=2)

# Fourth link ratio
Lambda4DATA <- subset(DATA,Previous_State == "RBNS4" &
                           Transition == 1 & Current_State != "Closed0")
Lambda4 <- gamlssMXfits(n=10,
                Lambda4DATA$LR~Lambda4DATA$Previous_Duration+
                Lambda4DATA$FirstPayment,
                data = Lambda4DATA, family = LOGNO, K=2)

# Fifth link ratio
Lambda5DATA <- subset(DATA,Previous_State == "RBNS5+" &
                           Transition == 1 & Current_State != "Closed0")
Lambda5 <- gamlssMXfits(n=10,
                Lambda5DATA$LR~Lambda5DATA$Previous_Duration+
                Lambda5DATA$FirstPayment,
                data = Lambda5DATA, family = LOGNO, K=2)
```

A.8 Reserve calculation with the Chain-Ladder method

```
### Mack chain-ladder analysis
# tri_dat is the upper part of the Chain Ladder triangle
for (i in 0:11){
  for (j in 0:(11-i)){
    tri_dat[i+1,j+1] <- cum_CF[i+1,j+1]
  }
}
```

```
tri_dat <- as.triangle(as.matrix(tri_dat))
Mack <- MackChainLadder(tri_dat, est.sigma="Mack")
# Calculation of the reserve for each accident year
for (i in 0:11){
 CL.reserves[i+1] <- Mack$FullTriangle[i+1,12]-Mack$FullTriangle[i+1,12-i]
}
# Reserve for the entire portfolio
CL.reserves.Total <- sum(CL.reserves[1:12])
```

A.9 Simulation of transitions for the coming year

For claims that are reported :

```
### Simulation of the transitions for the coming year
NSimu <- 2000
# Augmented database with data from the current year
CYData <- subset(DATA, AY + DY == 2006 & Current_State != "Closed0" &
                 Current_State != "Closed+" & Current_State != "IBNR")
# Matrix containing realisations of a U(0,1) law
Vunif <- matrix(runif(length(CYData$ID)*NSimu),
                nrow = NSimu, ncol = length(CYData$ID))

ddd <- array(data = 100, c(length(CYData$ID),
             length(CumTransition[1,]), NSimu))
for(k in 1:NSimu){
  for(i in 1:length(CYData$ID)){
    for(j in 1:length(CumTransition[1,])){
      if(CumTransition[CurrentAS[i],j] > Vunif[k,i]){
        ddd[i,j,k] <- j
      }
    }
  }
}
# We take the first state where the CDF is bigger than
# the realisation of the U(0,1) law
SimulatedAS <- matrix(nrow = length(CYData$ID), ncol = NSimu)
for(k in 1:NSimu){
  SimulatedAS[,k] <- colnames(Transition)[apply(ddd[,,k], 1, which.min)]
}
```

For claims that are hypothetically in the IBNR state :

```
### Simulation of the transitions for claims in the IBNR state
# Transition when leaving an IBNR state
```

```
Transition.IBNR <- Transition[c("IBNR␣0", "IBNR␣1", "IBNR␣2"),
                            c("RBNP␣0", "RBNS1␣0", "Closed0␣0", "Closed+␣0")]
Transition.IBNR <- Transition.IBNR/rowSums(Transition.IBNR)
# Theoretical number of claims that will leave the IBNR state for each year
CurrentAS.IBNR <- c(rep("IBNR␣0", round(alpha*Beta[2],0)),
                    rep("IBNR␣1", round(alpha*Beta[3],0)),
                    rep("IBNR␣2", round(alpha*(1-sum(Beta[1:3])),0)))
CumTransition.IBNR <- t(apply(Transition.IBNR,1,cumsum))
Vunif.IBNR <- matrix(runif(length(CurrentAS.IBNR)*NSimu),
                     nrow = NSimu, ncol = length(CurrentAS.IBNR))
ddd.IBNR <- array(data = 100, c(length(CurrentAS.IBNR),
                                length(CumTransition.IBNR[1,]), NSimu))
for(k in 1:NSimu){
  for(i in 1:length(CurrentAS.IBNR)){
    for(j in 1:length(CumTransition.IBNR[1,])){
      if(CumTransition.IBNR[CurrentAS.IBNR[i],j] > Vunif.IBNR[k,i]){
        ddd.IBNR[i,j,k] <- j
      }
    }
  }
}
SimulatedAS.IBNR <- matrix(nrow = length(CurrentAS.IBNR), ncol = NSimu)
for(k in 1:NSimu){
  SimulatedAS.IBNR[,k] <- colnames(Transition.IBNR)[apply(ddd.IBNR[,,k],
                                                         1, which.min)]
}
```

A.10 Simulation of payments for the coming year

For claims in the RBNS1 2 artificial state with a first payment smaller than 100 000 :

```
# Creation of functions to optimise in order to retrieve
# the value of the payment depending on the quantile
Pred.Lambda1.2.inf <- function(x,a){
  abs(pMX(x, mu=list(Lambda1$models[[1]]$mu.coefficients[1]+
    Lambda1$models[[1]]$mu.coefficients[3],
    Lambda1$models[[2]]$mu.coefficients[1]+
    Lambda1$models[[2]]$mu.coefficients[3]),
    sigma = list(exp(Lambda1$models[[1]]$sigma.coefficients),
    exp(Lambda1$models[[2]]$sigma.coefficients)),
    pi = list(Lambda1$prob[1],Lambda1$prob[2]),
    family = list(LOGNO, LOGNO)) - a)
}

# Matrix containing realisation of a U(0,1)
```

```
PLR_Unif <- matrix(runif(length(CYData$ID)*NSimu),
                nrow = NSimu, ncol = length(CYData$ID))
# Simulation of cash-flows
for(k in 1:NSimu){
 for(i in 1:length(CYData$ID)){
    if(ASData[i,"CurrentAS",k] == "RBNS1␣2" &
      (ASData[i,"SimulatedAS",k] == "Closed+␣0" |
       ASData[i,"SimulatedAS",k] == "RBNS2␣0") &
       ASData[i,"FirstPayment",k] == "Less100k"){
          ASData[i,"P1_LRi",k] <- optimize(Pred.Lambda1.2.inf, c(0,100),
                                 tol = 0.0001, a = PLR_Unif[i])$minimum
        }
    }
}
```

For claims in the IBNR 0 artificial state :

```
# Creation of functions to optimise in order to retrieve
# the value of the payment depending on the quantile
Pred.P1.IBNR.0 <- function(x,a){
  abs(pMX(x, mu=list(P1$models[[1]]$mu.coefficients[1],
  P1$models[[2]]$mu.coefficients[1]),
  sigma = list(exp(P1$models[[1]]$sigma.coefficients),
  exp(P1$models[[2]]$sigma.coefficients)),
  pi = list(P1$prob[1],P1$prob[2]), family = list(LOGNO, LOGNO)) - a)
}
# Matrix containing realisation of a U(0,1)
PLR_Unif.IBNR <- matrix(runif(length(CurrentAS.IBNR)*NSimu),
                   nrow = NSimu, ncol = length(CurrentAS.IBNR))
# Simulation of cash-flows
for(k in 1:NSimu){
 for(i in 1:length(CurrentAS.IBNR)){
  if(ASData.IBNR[i,"CurrentAS",k] == "IBNR␣0" &
    (ASData.IBNR[i,"SimulatedAS",k] == "Closed+␣0" |
     ASData.IBNR[i,"SimulatedAS",k] == "RBNS1␣0")){
       ASData.IBNR[i,"P1_LRi",k] <- optimize(Pred.P1.IBNR.0,
                                       c(0,10^8), tol = 0.0001,
                                       a = PLR_Unif.IBNR[k,i])$minimum
    }
  }
}
```

Appendix B

Transition probability matrix and $\tilde{M}$ matrix

B.1 Transition probability matrix

	IBNR 0	RBNS1 0	Closed+ 0	RBNP 0	IBNR 1	Closed0 0	RBNS2 0	RBNS1 1	IBNR 2	RBNP 1	RBNS2 1	RBNS1 2	RBNS3 0	RBNS2 2	RBNS4 0	RBNS3 1	RBNS1 3	RBNS2 3
IBNR 0	0	0.3203	0.5147	0.0938	0.0683	0.003	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 0	0	0	0.1679	0	0	0.4507	0.226	0.1554	0	0	0	0	0	0	0	0	0	0
Closed+ 0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNP 0	0	0.4008	0.5582	0	0	0.027	0	0	0	0.0139	0	0	0	0	0	0	0	0
IBNR 1	0	0.1517	0.7641	0.0399	0	0.0015	0	0	0.0428	0	0	0	0	0	0	0	0	0
Closed0 0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 0	0	0	0.0554	0	0	0.0554	0	0	0	0	0.6748	0	0.2145	0	0	0	0	0
RBNS1 1	0	0	0.0196	0	0	0.0632	0.0887	0	0	0	0	0.8285	0	0	0	0	0	0
IBNR 2	0	0.2781	0.1291	0.0762	0	0.0033	0	0	0.5132	0	0	0	0	0	0	0	0	0
RBNP 1	0	0.2121	0.0043	0	0	0.0693	0	0	0	0.7143	0	0	0	0	0	0	0	0
RBNS2 1	0	0	0.0133	0	0	0.0749	0	0	0	0	0	0	0.0273	0.8845	0	0	0	0
RBNS1 2	0	0	0.0104	0	0	0.0966	0.0299	0	0	0	0	0	0	0	0	0	0.8631	0
RBNS3 0	0	0	0.0885	0	0	0.027	0	0	0	0	0	0	0	0	0.4549	0.4297	0	0
RBNS2 2	0	0	0.0072	0	0	0.0991	0	0	0	0	0	0	0.019	0	0	0	0	0.8747
RBNS4 0	0	0	0.0764	0	0	0.0163	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 1	0	0	0.0275	0	0	0.0825	0	0	0	0	0	0	0	0	0.0766	0	0	0
RBNS1 3	0	0	0.0101	0	0	0.0961	0.0226	0	0	0	0	0	0	0	0	0	0	0
RBNS2 3	0	0	0.0037	0	0	0.1023	0	0	0	0	0	0	0.0103	0	0	0	0	0
RBNS5+ 0	0	0	0.073	0	0	0.0081	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 2	0	0	0.0121	0	0	0.0993	0	0	0	0	0	0	0	0	0.0508	0	0	0
RBNS1 4	0	0	0.0044	0	0	0.1109	0.0154	0	0	0	0	0	0	0	0	0	0	0
RBNS4 1	0	0	0.028	0	0	0.0981	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 4	0	0	0.0042	0	0	0.1317	0	0	0	0	0	0	0.0037	0	0	0	0	0
RBNS5+ 1	0	0	0.0385	0	0	0.0385	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 3	0	0	0.0117	0	0	0.1173	0	0	0	0	0	0	0	0	0.0293	0	0	0
RBNS1 5	0	0	0.0058	0	0	0.2033	0.0082	0	0	0	0	0	0	0	0	0	0	0
RBNS4 2	0	0	0.0231	0	0	0.1618	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 5	0	0	0.0029	0	0	0.205	0	0	0	0	0	0	0.0026	0	0	0	0	0
RBNS5+ 2	0	0	0.0422	0	0	0.0552	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 4	0	0	0.0035	0	0	0.1228	0	0	0	0	0	0	0	0	0.0246	0	0	0
RBNS4 3	0	0	0.0233	0	0	0.1628	0	0	0	0	0	0	0	0	0	0	0	0
RBNS5+ 3	0	0	0.0326	0	0	0.0688	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 5	0	0	0.0016	0	0	0.2461	0	0	0	0	0	0	0	0	0.0031	0	0	0
RBNS4 4	0	0	0.04000	0	0	0.180000	0	0	0	0	0	0	0	0	0	0	0	0
RBNS5+ 4	0	0	0.0166	0	0	0.0788	0	0	0	0	0	0	0	0	0	0	0	0
RBNS4 5	0	0	0.0191	0	0	0.2102	0	0	0	0	0	0	0	0	0	0	0	0
RBNS5+ 5	0	0	0.1459	0	0	0.1141	0	0	0	0	0	0	0	0	0	0	0	0

	RBNS5+ 0	RBNS3 2	RBNS1 4	RBNS4 1	RBNS2 4	RBNS5+ 1	RBNS3 3	RBNS1 5	RBNS4 2	RBNS2 5	RBNS5+ 2	RBNS3 4	RBNS4 3	RBNS5+ 3	RBNS3 5	RBNS4 4	RBNS5+ 4	RBNS4 5	RBNS5+ 5
IBNR 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Closed+ 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNP 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IBNR 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Closed0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IBNR 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNP 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS4 0	0.5561	0	0	0.3512	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 1	0	0.8134	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 3	0	0	0.8712	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 3	0	0	0	0	0.8837	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS5+ 0	0	0	0	0	0	0.9189	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 2	0	0	0	0	0	0	0.8378	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 4	0	0	0	0	0	0	0	0.8693	0	0	0	0	0	0	0	0	0	0	0
RBNS4 1	0.0654	0	0	0	0	0	0	0	0.8084	0	0	0	0	0	0	0	0	0	0
RBNS2 4	0	0	0	0	0	0	0	0	0	0.8604	0	0	0	0	0	0	0	0	0
RBNS5+ 1	0	0	0	0	0	0	0	0	0	0	0.9231	0	0	0	0	0	0	0	0
RBNS3 3	0	0	0	0	0	0	0	0	0	0	0	0.8416	0	0	0	0	0	0	0
RBNS1 5	0	0	0	0	0	0	0	0.7826	0	0	0	0	0	0	0	0	0	0	0
RBNS4 2	0.0578	0	0	0	0	0	0	0	0	0	0	0	0.7572	0	0	0	0	0	0
RBNS2 5	0	0	0	0	0	0	0	0	0	0.7895	0	0	0	0	0	0	0	0	0
RBNS5+ 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9026	0	0	0	0	0
RBNS3 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8491	0	0	0	0
RBNS4 3	0.031	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7829	0	0	0
RBNS5+ 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8986	0	0
RBNS3 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7492	0	0	0	0
RBNS4 4	0.010000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.770000	0
RBNS5+ 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9046
RBNS4 5	0.0127	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.758	0
RBNS5+ 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7401

B.2 $\tilde{M}$ matrix depending on the value of the first payment

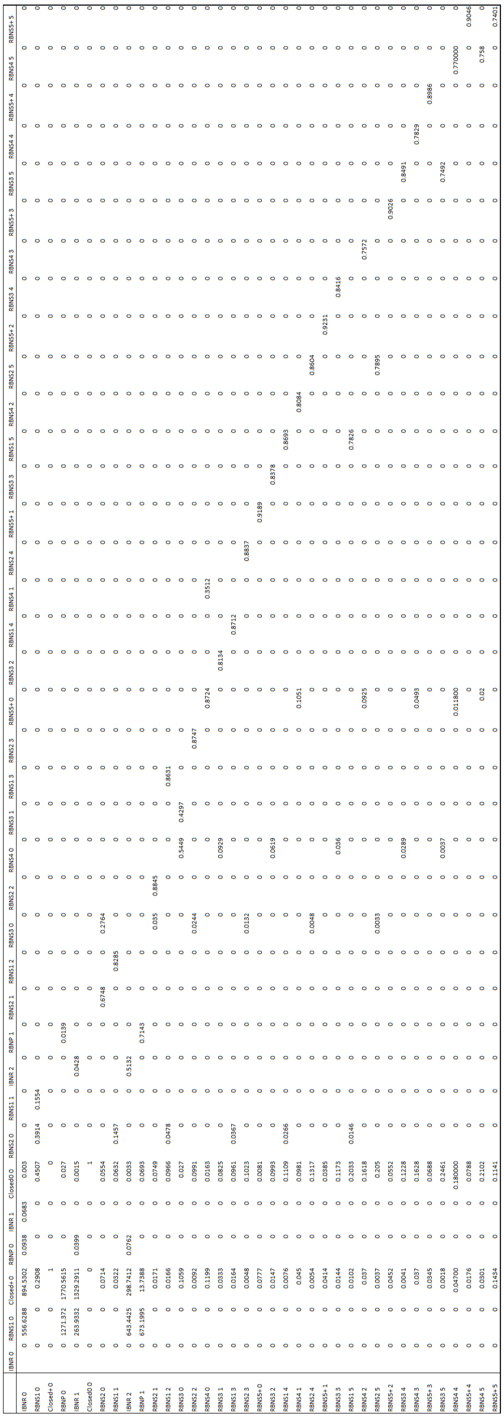

Figure B.2.1: $\tilde{M}$ matrix with $P_1 < 100000$.

	IBNR 0	RBNS1 0	Closed+ 0	RBNP 0	IBNR 1	Closed0 0	RBNS2 0	RBNS1 1	IBNR 2	RBNP 1	RBNS2 1	RBNS1 2	RBNS3 0	RBNS2 2	RBNS4 0	RBNS3 1	RBNS1 3	RBNS2 3	RBNS5+ 0	RBNS3 2	RBNS1 4	RBNS4 1	RBNS2 4	RBNS5+ 1	RBNS3 3	RBNS1 5	RBNS4 2	RBNS2 5	RBNS5+ 2	RBNS3 4	RBNS4 3	RBNS5+ 3	RBNS3 5	RBNS4 4	RBNS5+ 4	RBNS4 5	RBNS5+ 5
IBNR 0	0	71728.09	115270.62	0.0938	0.0683	0.003	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 0	0	0	0.3094	0	0	0.4507	0.4165	0.1554	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Closed+ 0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNP 0	0	108426.5	150998.84	0	0	0.027	0	0	0	0.0139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IBNR 1	0	26048.45	131192.19	0.0399	0	0.0015	0	0	0.0428	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Closed0 0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 0	0	0	0.0678	0	0	0.0554	0	0	0	0	0.6748	0	0.2624	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 1	0	0	0.0334	0	0	0.0632	0.1512	0	0	0	0	0.8285	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IBNR 2	0	47754.77	22171.859	0.0762	0	0.0033	0	0	0.5132	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNP 1	0	42953.29	876.5977	0	0	0.0693	0	0	0	0.7143	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 1	0	0	0.0162	0	0	0.0749	0	0	0	0	0	0	0.0331	0.8845	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 2	0	0	0.0166	0	0	0.0966	0.0479	0	0	0	0	0	0	0	0	0	0.8631	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 0	0	0	0.0987	0	0	0.027	0	0	0	0	0	0	0	0	0.5076	0.4297	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 2	0	0	0.0087	0	0	0.0991	0	0	0	0	0	0	0.023	0	0	0	0	0.8747	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS4 0	0	0	0.113	0	0	0.0163	0	0	0	0	0	0	0	0	0	0	0	0	0.8225	0	0	0.3512	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 1	0	0	0.031	0	0	0.0825	0	0	0	0	0	0	0	0	0.0864	0	0	0	0	0.8134	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 3	0	0	0.0171	0	0	0.0961	0.0382	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8712	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS2 3	0	0	0.0046	0	0	0.1023	0	0	0	0	0	0	0.0127	0	0	0	0	0	0	0	0	0	0.8837	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS5+ 0	0	0	0.0969	0	0	0.0081	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9189	0	0	0	0	0	0	0	0	0	0	0	0	0
RBNS3 2	0	0	0.0138	0	0	0.0993	0	0	0	0	0	0	0	0	0.0578	0	0	0	0	0	0	0	0	0	0.8378	0	0	0	0	0	0	0	0	0	0	0	0
RBNS1 4	0	0	0.0082	0	0	0.1109	0.0287	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8693	0	0	0	0	0	0	0	0	0	0	0
RBNS4 1	0	0	0.0425	0	0	0.0981	0	0	0	0	0	0	0	0	0	0	0	0	0.0991	0	0	0	0	0	0	0	0.8084	0	0	0	0	0	0	0	0	0	0
RBNS2 4	0	0	0.0051	0	0	0.1317	0	0	0	0	0	0	0.0045	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8604	0	0	0	0	0	0	0	0	0
RBNS5+ 1	0	0	0.0557	0	0	0.0385	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9231	0	0	0	0	0	0	0	0
RBNS3 3	0	0	0.0135	0	0	0.1173	0	0	0	0	0	0	0	0	0.0336	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8416	0	0	0	0	0	0	0
RBNS1 5	0	0	0.0111	0	0	0.2033	0.0159	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7826	0	0	0	0	0	0	0	0	0	0	0
RBNS4 2	0	0	0.0349	0	0	0.1618	0	0	0	0	0	0	0	0	0	0	0	0	0.0873	0	0	0	0	0	0	0	0	0	0	0	0.7572	0	0	0	0	0	0
RBNS2 5	0	0	0.0035	0	0	0.205	0	0	0	0	0	0	0.0031	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7895	0	0	0	0	0	0	0	0	0
RBNS5+ 2	0	0	0.0607	0	0	0.0552	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9026	0	0	0	0	0
RBNS3 4	0	0	0.0039	0	0	0.1228	0	0	0	0	0	0	0	0	0.027	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8491	0	0	0	0
RBNS4 3	0	0	0.0348	0	0	0.1628	0	0	0	0	0	0	0	0	0	0	0	0	0.0464	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7829	0	0	0
RBNS5+ 3	0	0	0.0456	0	0	0.0688	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8986	0	0
RBNS3 5	0	0	0.0017	0	0	0.2461	0	0	0	0	0	0	0	0	0.0034	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7492	0	0	0	0
RBNS4 4	0	0	0.04430	0	0	0.180000	0	0	0	0	0	0	0	0	0	0	0	0	0.011100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.770000	0
RBNS5+ 4	0	0	0.0236	0	0	0.0788	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9046
RBNS4 5	0	0	0.0283	0	0	0.2102	0	0	0	0	0	0	0	0	0	0	0	0	0.0189	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.758	0
RBNS5+ 5	0	0	0.1482	0	0	0.1141	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7401

Figure B.2.2: $\tilde{M}$ matrix with $P_1 \geq 100000$.

Bibliography

[1] Elja Arjas. The Claims Reserving Problem in Non-Life Insurance: Some Structural Ideas. *ASTIN Bulletin*, 19:139–152, 1989.

[2] Ragnar Norberg. Prediction of Outstanding Liabilities in Non-Life Insurance. *ASTIN Bulletin*, 23:95–115, 1993.

[3] Ragnar Norberg. Prediction of Outstanding Liabilities II. Model Variations and Extensions. *ASTIN Bulletin*, 29:5–25, 1999.

[4] Alexandre Boumezoued and Laurent Devineau. Individual claims reserving: a survey. 2017.

[5] Arthur Charpentier and Mathieu Pigeon. Macro vs. Micro Methods in Non-Life Claims Reserving (an Econometric Perspective). *Risks*, 4:12, May 2016.

[6] Damien Drieskens, Marc Henry, Jean-François Walhin, and Jürgen Wielandts. Stochastic projection for large individual losses. *Scandinavian Actuarial Journal*, 2012:1–39, March 2012.

[7] Mario V. Wüthrich and Michael Merz, editors. *Stochastic Claims Reserving Methods in Insurance: Wüthrich/Stochastic*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2008.

[8] Mario V. Wuthrich and Michael Merz. Stochastic Claims Reserving Manual: Advances in Dynamic Modeling. *SSRN Electronic Journal*, 2015.

[9] Muhammed Taher Al-Mudafer, Benjamin Avanzi, Greg Taylor, and Bernard Wong. Stochastic loss reserving with mixture density neural networks. *Insurance: Mathematics and Economics*, 105:144–174, July 2022.

[10] Carole Bettonville, Louise d'Oultremont, Michel Denuit, Julien Trufin, and Robin Van Oirbeek. Matrix calculation for ultimate and 1-year risk in the Semi-Markov individual loss reserving model. *Scandinavian Actuarial Journal*, 2021:380–407, May 2021.

[11] Vlad Stefan Barbu and N. Limnios. *Semi-Markov chains and hidden semi-Markov models toward applications: their use in reliability and DNA analysis*. Number v. 191 in Lecture notes in statistics. Springer, New York, 2008. OCLC: ocn173718690.

[12] Andrea Gabrielli and Mario V. Wüthrich. An Individual Claims History Simulation Machine. *Risks*, 6:29, March 2018.