TECHNISCHE UNIVERSITÄT WIEN

MASTERTHESIS

Development and Optimization of Reserving Models in Actuarial Science: A Python-Based Approach

written on January 25, 2025

Institute of Statistics and Mathematical Methods in Economics
Vienna University of Technology

supervised by

Univ.-Prof. Dipl.-Ing. Dr. techn. Stefan Gerhold

submitted by

Elena Mayr B.Sc.

[redacted]

Vienna, January 25, 2025

Statement of Originality

I hereby declare that I have authored the present master thesis independently and did not use any sources other than those specified. I have not yet submitted the work to any other examining authority in the same or comparable form. It has not been published yet.

Vienna, at January 25, 2025

[redacted]

Elena Mayr

Abstract

Estimating claim reserves is a relevant but challenging topic in insurance. Several laws and guidelines define the need for technical provisions where an accurate calculation of the claims reserves is needed. However, making accurate estimations is challenging, especially for claims that include personal damage and potential long-term effects. In this thesis, we address the optimisation of the Chain Ladder method, analysing the impact of various factors, including outlier exclusion, using simple max /min exclusion, Reverse Nearest Neighbour and Interquartile Distance as outlier detection methods. Other adjusted parameters are the periods considered, tail adjustment, inflation adjustment, and using a weighted average of both the paid and incurred triangle. The algorithm was implemented using Python. In this analysis, one key finding occurred, which is that optimal model parameters differ significantly between incurred and paid losses and between short-tail, long-tail, and volatile insurance branches. More straightforward methods, such as excluding the maximum value or limiting the number of periods, often perform better than more complex approaches. Although according to our analysis, the tail and inflation analysis have no positive impact on the accuracy of the estimations, they should be further examined in future studies. This thesis shows the potential for automated optimisation to improve claims reserving accuracy while reducing actuaries' manual workload. However, it highlights that expert judgment remains essential, primarily when sudden changes or external trends that historical data alone cannot capture, occur.

Acknowledgements

First I am deeply grateful to my Allianz supervisors for their invaluable guidance, expertise, and feedback. Mag. Radka Kvasnica, PhD. and Mag. Peter Perger provided professional insight and support for this thesis. I am also deeply grateful to Univ.-Prof. Dipl.-Ing. Dr. techn. Stefan Gerhold Stefan Gerhold from the Technical University of Vienna for his academic supervision and constructive feedback throughout the development of this thesis.

I would like to express my deepest gratitude to my incredible family for their unconditional support, encouragement, and belief in me throughout this journey. Their enduring presence has been invaluable during both challenging and rewarding moments.

A heartfelt thank you to Michael, my partner, whose patience, emotional support, and understanding have been a constant source of strength and balance during the writing of this thesis. You accompanied me through the highs and lows.

I'm genuinely grateful to my friends for always being there and providing such amazing emotional support, especially my friend Angela, who gave me invaluable feedback for the thesis.

I want to express my heartfelt thanks to my university colleagues, Katharina, Alexander, Robert, Elisabeth, Patricia and Markus, for all those hours we shared, navigating the stresses of university life together. We tackled challenges as a team and made wonderful memories at parties and karaoke nights!

Thank you all for your contributions, which have helped me complete this thesis!

Contents

1 Introduction

Estimating the total amount an insurance company must pay for a specific accident is a central yet challenging task in actuarial science. Claims reserving ensures that insurers can meet their future obligations by accurately predicting payments for incurred claims. On the one hand, there are Incurred But Not Reported Claims (IBNR), which refer to situations such as a vacation house experiencing water damage that the owner is unaware of. On the other hand, Incurred But Not Enough Reserved (IBNER) accidents where the extent of the final payment is unknown, such as personal damage and potential long-term effects [5]. Various laws and guidelines define that, insurers must have sufficient liquid funds to guarantee the fulfilment of obligations:

- **Insurance Supervisionlaw (VAG 2016):** § 150. (1) Technical provisions must be established to the extent necessary, based on reasonable entrepreneurial judgment, to ensure the ongoing fulfilment of obligations arising from insurance contracts. During the valuation, due consideration must be given to the principle of prudence[19].

- **Solvency II** - Solvency II demands technical provisions consisting of the best estimate and risk margin [4].

- **International Financial Reporting Standards (IFRS)** IFRS 17 demands technical provisions consisting of the best estimate and risk adjustment [10]

Even if all these different regulations have their specifications on how to determine the technical provisions, they all have in common that a good estimation of the loss reserve is required.
There are many different models for reserving claims, such as Chain-Ladder, Munich-Chain-Ladder, Bornhuetter-Ferguson, Machine Learning models for individual reserving, and many more. One of the most straightforward, and also one of the most used, methods is the Chain-Ladder method. It has the disadvantages of being very sensitive to outliers and assuming that the claims development pattern does not change over time, these are possible huge drawbacks in practice. The idea to counteract these disadvantages is to exclude the outliers and only consider a shorter period than the whole history. The challenge lies in identifying the correct outliers and determining the optimal period to consider. In practice, actuaries decide, based on their expert knowledge, which outliers to exclude, how many periods to consider, and whether other adaptations have to be made, like tail estimation, et cetera. However, since this produces a significant workload for actuaries, it is a logical progression to attempt to automate this process.

In this thesis, an optimization algorithm to improve the Chain-Ladder method for claims reserving is developed. The underlying idea is to evaluate model accuracy across different lines of business for various models and identify the best-performing model based on historical data. With this approach, we can automate key decisions such as outlier detection, period selection, and tail estimation.

The algorithm was implemented using Python, using the ChainLadder ([3]) package as its foundation. The various models are all Chain-Ladder-based but differ in their approaches to outlier detection, volume selection, tail factors, inflation adjustments, and incorporating weighted combinations of incurred and paid triangles. The Claims Development Result (CDR) is the error measure used, although the Mack Standard Error is also examined for selected models. Additionally, an Actual vs. Expected analysis is performed.

This work is structured as follows: In chapter 2 the following reserving models: Loss Development, Chain-Ladder, Bornhuetter-Ferguson, Cape Cod, and Munich-Chain-Ladder, their respective advantages and disadvantages are discussed. Chapter 3 highlights the importance of outlier exclusion, followed by an introduction to Reverse Nearest Neighbor (RNN) and Interquartile Range (IQR) as outlier detection methods. Two error measures used for model optimization are introduced in Chapter 4, the Mack Standard Error and the Claims Development Result. In Chapter 5, the optimisation algorithm is introduced. Chapter 6, Methodology and Data Foundation, begins with a description of the data, including its structure and characteristics. Subsequently, the implementation is described using specific code snippets. Results and discussion are presented in chapter 7, here is a description of the link ratios of the triangles. A comparison of different models is conducted: The impact of varying volumes on the CDR and eliminating maximum and minimum values. Similar analyses using IQR and RNN for outlier detection, as well as the removal of potentially problematic diagonals is done. The effect of exponential and inverse power tails, a weighted combination of paid and incurred losses, and inflation-adjusted data. After examining the various models, the minimum and maximum exclusion and different volumes concerning the MSE for all branches are analyzed. Finally, the optimal model is compared to two baseline models using an Actual vs. Expected analysis.

2 Claim reserving

This chapter focuses on claims reserving, which entails estimating future payments for claims already incurred and those yet to occur. It begins by explaining the theoretical framework common to all methods, detailing the available data and its structure. Subsequently, it discusses how this information is generalized and incorporated into a mathematical framework. Finally, it presents several methods for claims reserving, including the Chain Ladder Method, Loss Development, Bornhuetter-Ferguson, Cape Cod, and Munich Chain Ladder methods.

2.1 Mathematical Framework

The following part is based on [[5], 12.1]. When a claim occurs, it is often uncertain how much the insurance will ultimately have to pay. This uncertainty arises because sometimes the insured party may not immediately recognize the occurrence of a claim (e.g., water damage in a holiday house - IBNR, incurred but not reported), or it may be unclear how extensive the final damages will be (e.g., accidents resulting in personal injuries - IBNER, incurred but not enough reserved). Since insurers are obligated to fulfill their promise of coverage, they must build reserves to ensure they can meet future claim payments. The procedure from claim occurrence to final settlement is stated below since understanding this process is essential to determining the reserves.

- The claim occurs in an accident year.
- The claim is discovered.
- The claim is reported to the insurer.
- The insurer establishes a case reserve.
- The insurer makes initial payments and adjusts the case reserve for any additional required payments.
- The claim is finally settled and closed.

Additionally, a previously closed claim can be reopened. The gradual settlement of a claim is referred to as claims handling and can span over multiple settlement years.

The challenge lies in accurately estimating reserves; if reserves are too low, costs may not be covered, whereas excessive reserves may lead to tax implications due to reduced profits. Insurance portfolio data alone may not suffice for these estimations, particularly in cases where no claims have been reported and, thus, no case reserves exist. Therefore, statistical methods are necessary to estimate reserves.

2.1.1 Claims Development Triangles

Understanding the need for reserves and the importance of statistical estimation methods raises the question of which data should be utilized and how it should be represented. The relevant risk data includes:

- Number of reported claims
- Number of claims finally settled
- Claim payments
- Incurred claims (sum of all payments plus case reserves)

The most common method for visualizing this data is through the use of *claims development triangles*. These triangles typically contain either cumulative claims or incremental claims data, helping to illustrate the progression of claims over different periods. Table 2.1 presents

Accident year	Development year k					
	2019	2020	2021	2022	2023	2024
2019	1001	854	568	565	347	148
2020		1113	990	671	648	422
2021			1265	1168	800	744
2022				1490	1383	1007
2023					1725	1536
2024						1889

Table 2.1: Incremental claims development for different accident years across development periods [5].

an incremental claims development triangle example, assuming the values represent claim payments. The table summarizes payment information available by the end of 2024, starting from claims originating in 2019. For instance, if we examine the accident year 2023, we observe that the insurance company paid 1,725 units in the same year, followed by an additional 1,536 units in 2024.

Accident year	Development year k					
	0	1	2	3	4	5
0	1001	854	568	565	347	148
1	1113	990	671	648	422	
2	1265	1168	800	744		
3	1490	1383	1007			
4	1725	1536				
5	1889					

Table 2.2: Incremental claims development for different accident years across development periods [5].

A more common illustration of claims development triangles is that the development years and accident years are not portrayed as calendar years but as delays concerning the accident year, like in Table 2.2. In this format, the data from the accident year “2023” is now positioned in row “4” allowing the claims development to be analyzed across columns “0” and “1”. This representation facilitates the identification of development patterns, providing insights into the required reserves. Such a structure is commonly referred to as a *run-off triangle*.

2.1.2 Stochastic Modelling

The following chapter is adapted from [22]. To define the different reserving models, we need a general framework that generalizes the examples provided in the previous section. In Figure 2.1, i represents the accident year, which is the year in which the incident oc-

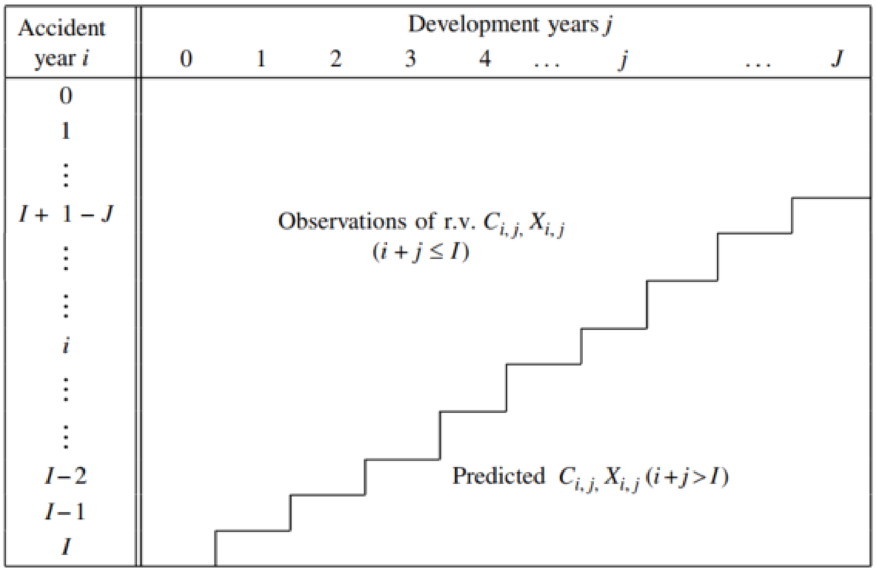


Figure 2.1: Claims development triangle [22]

curred (displayed on the vertical axis). j represents the development year, indicating the number of years or periods that have passed since the occurrence. Here, $i \in \{0, \ldots, I\}$ and $j \in \{0, \ldots, J\}$, where I denotes the latest accident year and J denotes the latest development year.

The variable $X_{i,j}$ in the figure denotes the payment in development year j corresponding to the claims that originated in year i. Thus, the payment arises in the accounting year $j + i$. The cumulative payments, $C_{i,j}$, are defined as:

$$C_{i,j} = \sum_{k=0}^{j} X_{i,k} \tag{2.1}$$

Therefore, knowing the cumulative payments, the incremental payments can be determined using:

$$X_{i,j} = \begin{cases} C_{i,0} & \text{if } j = 0 \\ C_{i,j} - C_{i,j-1} & \text{otherwise} \end{cases} \tag{2.2}$$

As illustrated in Figure 2.1, the accident years are typically displayed on the vertical axis, while the development years are positioned on the horizontal axis. A crucial point in time is I, up to which all payments are known. This means that the data from the upper triangle, i.e., $i+j \leq I$, consists of known observations, whereas the lower triangle, i.e., $i+j > I$, contains values that must be predicted. These unknown values are the focus of our estimations.

Accordingly, we can divide the payments into two sets: the known observations

$$\mathcal{D}_I = \{X_{i,j}; i + j \leq I, 0 \leq j \leq J\} \tag{2.3}$$

and their complement, the unknown values that require estimation:

$$\mathcal{D}_I^c = \{X_{i,j}; i + j > I, i \leq I, j \leq J\} \tag{2.4}$$

The payments occurring in accounting year k are represented by the diagonals $i + j = k$, $k \geq 0$ and can be expressed as the sum of individual elements along these diagonals:

$$X_k = \sum_{i+j=k} X_{i,j} \tag{2.5}$$

Incremental claims, denoted as $X_{i,j}$, can represent additional payments in cell (i, j). Alternatively, they can correspond to reported claims with a reporting delay j and an accident year i, or they might signify the change in reported claim amounts in this specific cell (i, j). Cumulative claims, denoted as $C_{i,j}$, may represent total payments, the overall number of reported claims, or the incurred claims (in the case of cumulative reported claims). $C_{i,J}$ is commonly referred to as the ultimate claim amount or the total number of claims for accident year i.
The outstanding loss liabilities are simply the payments that are expected but have not yet been made. Given that $X_{i,j}$ represents the incremental payments, the outstanding loss liabilities for accident year i at time j can be defined as:

$$R_{i,j} = \sum_{k=j+1}^{J} X_{i,k} = C_{i,J} - C_{i,j} \tag{2.6}$$

Two closely related concepts are the *outstanding loss liabilities* and the *claims reserves*. To predict the former, the latter must be estimated. Claims reserves represent the missing amount that, when added to the past claims $C_{i,j}$, provides a predictor for the total claims load, also known as the ultimate claim $C_{i,J}$ for accident year i.

General Assumption

We assume
$$I = J$$
and
$$X_{i,j} = 0 \text{ for all } j > J$$
The first assumption can easily be dropped, it is only a simplification. This suggests that we have to predict the outstanding loss liabilities for every but the 0th year, i.e. $i = 1, \ldots, I$

2.2 Loss Reserving Methods

The first big decision an actuary has to make is the model selection. This chapter gives an overview of the most common claims reserving methods. The first models, the chain-ladder (CL) and the Bornhuetter-Ferguson (BF) are the simplest but also the most common techniques. The Chain Ladder (CL) method and the Bornhuetter-Ferguson (BF) method can be perceived as algorithmic approaches for establishing claims reserves. These algorithms serve as systematic tools for forecasting future liabilities. Nevertheless, this conceptualization cannot generally quantify the uncertainties inherent in these predictions.

2.3 Loss Development method

This section is adapted from [5]. The simplest claims reserving method is arguably the Loss-Development method. This method is based on the premise that every claim follows the same loss development pattern. This implies that we assume the claims in the second development year to be a fixed percentage of the ultimate value, irrespective of whether the claim occurred in 2004 or 2023. The loss development pattern is denoted as
$$\gamma_0, \gamma_1, \ldots, \gamma_I$$
where the index represents the development year. An γ_i for $i \in \{1, \ldots, I\}$ represents the claim development from the accident year to the is development year.

This pattern is either given or must be estimated using either internal, external or a combination of internal and external information. Loss development factors calculated from internal information could be the age-to-age factors from the later described chain ladder method. Another use case is when an insurance company introduces a new class of insurance without having a similar existing one; external information could be information from a reinsurer or other expert opinion.

Given the estimators:

$$\hat{\gamma}_0, \hat{\gamma}_1, \ldots, \hat{\gamma}_I$$

the *loss-development-estimator* for the future claim status $C_{i,k}$ is calculated by:

$$C_{i,k}^{LD} := \hat{\gamma}_k \frac{C_{i,I-1}}{\hat{\gamma}_{I-1}},$$

where $\hat{\gamma}_I = 1$, indicating that no further payments are expected.

We can express the *loss-development estimator* as:

$$E[C_{i,k}] = \frac{E[C_{i,k}]}{E[C_{i,I}]} \cdot \frac{E[C_{i,I}]}{E[C_{i,I-1}]} \cdot E[C_{i,I-1}] = \gamma_k \cdot \frac{1}{\gamma_{I-1}} \cdot E[C_{i,I-1}] = \gamma_k \cdot \frac{E[C_{i,I-1}]}{\gamma_{I-1}}.$$

The reserves are then calculated according to equation (2.6).

2.4 Chain-Ladder Method (Distribution-Free)

As described by Wüthrich and Merz [22], one of the simplest yet most widely used loss reserving techniques is the Chain Ladder (CL) method. Although the CL method is distribution-free, various other stochastic models provide a theoretical foundation for it. The distribution-free derivation of the Chain Ladder method connects consecutive cumulative claims through appropriate link ratios, relying on the following characterization of the model.

2.4.1 Model Assumptions and Mathematical Foundations

Model Assumptions (distribution-free CL model)[22, page 16]

- Cumulative claims $C_{i,j}$ of different accident years i are independent
- There exist development factors $f_0, \ldots, f_{J-1}$ such that for all $0 \leq i \leq I$ and all $1 \leq j \leq J$ we have

$$E[C_{i,j}|C_{i,0}, \ldots, C_{i,j-1}] = E[C_{i,j}|C_{i,j-1}] = f_{j-1}C_{i,j-1} \tag{2.7}$$

The independence of cumulative claims across different accident years implies that the occurrence of numerous accidents in 2015 has no bearing on the number of accidents in 2022. It is important to note that this assumption does not always hold in real-world scenarios, nevertheless, we need it to eliminate the effects of accounting years.

Given the simplicity of the Chain Ladder method, it is sufficient to impose a restriction on the first moment (see Equation (2.7)). This is because we only need to calculate the conditionally expected future claims, i.e., the first moment.

We can further assume that the sequence $C_{i,0}, C_{i,1}, \ldots$ forms a Markov chain. This results in

$$(C_{i,j}\ \Pi_{k=0}^{j-1} f_k^{-1})_{j\geq 1}.$$

being a martingale. The development factors f_j are also referred to by several other names, such as *link ratios*, *development factors*, *CL factors*, or *age-to-age factors*. These variables are the core of the Chain Ladder method. They describe the development of consecutive cumulative claims, and the primary task is to find appropriate estimators [22, Remark 2.2]. The claims that occurred until the year I are represented through the upper triangle: $\mathcal{D}_I = \{X_{i,j}; i+j \leq I, 0 \leq j \leq J\}$

Lemma 1. *Under the Model Assumptions* (2.7) *we have*

$$E\Big[\ C_{i,J}\ \Big| \mathcal{D}_I\Big] = E[C_{i,J}|C_{i,I-i}] = C_{i,I-i} f_{I-i} \ldots f_{J-1} \tag{2.8}$$

for all $1 \leq i \leq I$

Proof. We can derive this lemma from

$$\begin{aligned} E[C_{i,J}|\mathcal{D}_I] &= E[C_{i,J}|C_{i,0}, \ldots, C_{i,I-i}] \\ &= E[E[C_{i,J}|C_{i,J-i}]|C_{i,0}, \ldots, C_{i,I-i}] \\ &= E[f_{J-1}\ C_{i,J-1}|C_{i,0}, \ldots, C_{i,I-i}] \\ &= f_{J-1} E[C_{i,J-1}|\mathcal{D}_I] \end{aligned}$$

- First equation: we use only the definition of the set $\mathcal{D}_I$, since $\mathcal{D}_I$ represents the information available up to development year $I - i$.
- Second equation: By the law of total expectation, we can re-express the expectation as an iterated expectation. This involves taking the conditional expectation of $C_{i,J}$ given $C_{i,J-i}$ and then taking the expectation of that result given the prior information.
- Last equations: this step relies on the key assumption of the Chain Ladder method (2.7), where the expected cumulative claims in the final development year J are proportional to the cumulative claims in the previous year $J-1$, with the proportionality factor being f_{J-1}.
- By substituting the Chain Ladder assumption into the iterated expectation, we obtain the final result. This shows that the expected cumulative claims in the final development year, given the available information, are equal to the development factor f_{J-1} times the expected cumulative claims in the previous development year, also given the available information.

□

The proof demonstrates how the Chain Ladder method projects future cumulative claims by relying on the proportional relationship between consecutive development years. The expectation for the final development year is derived from the expectation for the previous year, adjusted by the development factor. This iterative process utilizes the key assumption that cumulative claims follow a predictable pattern over time.

Hence based on the observations $\mathcal{D}_I$, the Lemma (1) can be used to create an algorithm for estimating the ultimate claim $C_{i,J}$. The calculation of the outstanding payment obligations for the loss year, based on $\mathcal{D}_I$, is carried out as follows, taking into account the known CL factors f_j

$$E[C_{i,J}|\mathcal{D}_I] - C_{i,I-i} = C_{i,I-i}(f_{I-i} \dots f_{J-1} - 1) \tag{2.9}$$

This is in line with determining the 'best estimate' reserves for accident year i at time $I-i$, relying on the information up to time $I-i$ and the known Chain Ladder factors f_j. It's noteworthy that, in this context, we employ the conditionally expected value mentioned in Lemma 1 to forecast the outcome of the random variable $C_{i,J} - C_{i,I-i}$, given information up to time I. Unfortunately, in many practical scenarios, the Chain Ladder factors are unknown and require estimation. The estimation of Chain Ladder factors f_j, where $j = 0, 1, \dots, J-1$, is carried out using the following formula:

$$\hat{f}_j = \frac{\sum_{i=0}^{I-j-1} C_{i,j+1}}{\sum_{i=0}^{I-j-1} C_{i,j}} = \sum_{i=0}^{I-j-1} \frac{C_{i,j}}{\sum_{i=0}^{I-j-1} C_{k,j}} \frac{C_{i,j+1}}{C_{i,j}} \tag{2.10}$$

In other words, the Chain Ladder factors f_j are approximated through a volume-weighted average of individual development factors $F_{i,j+1} = C_{i,j+1}/C_{i,j}$

Lemma 2. *The CL estimator for $E[C_{i,j}|\mathcal{D}_I]$ is given by*

$$\hat{C}_{i,j}^{CL} = E[\hat{C}_{i,j}|\mathcal{D}_I] = C_{i,I-i}\hat{f}_{I-i} \dots \hat{f}_{j-1} \tag{2.11}$$

for $i + j > I$

When viewed purely from an algorithmic perspective, equation (2) represents the procedure used to produce the Chain Ladder reserves. It is important to note that this method is often applied without reference to an appropriate underlying stochastic model. To further explore the characteristics of the Chain Ladder method, we introduce an additional variable that describes the available information.

$$\mathcal{B}_k = \{C_{i,j}; i + j \leq I, 0 \leq j \leq k\} \subseteq \mathcal{D}_I \tag{2.12}$$

This means: $\mathcal{B}_J = \mathcal{D}_I$, which describes the set of all observations at time I

Lemma 3. *For the estimators hold*

(a) *Given $\mathcal{B}_j$, $\hat{f}_j$ is an unbiased estimator for f_j, i.e. $E[\hat{f}_j|\mathcal{B}_j] = f_j$;*

(b) *$\hat{f}_j$ is (unconditionally) unbiased for f_j, i.e. $E[\hat{f}_j] = f_j$;*

(c) *$E[\hat{f}_0 \dots \hat{f}_j] = E[\hat{f}_0] \dots E[\hat{f}_j]$, i.e. $\hat{f}_0 \dots \hat{f}_{J-1}$ are uncorrelated;*

(d) *given $C_{i,I-i}$,$\hat{C}^{CL}_{i,J}$ is an unbiased estimator for $E[C_{i,J}|\mathcal{D}_I] = E[C_{i,J}|C_{i,I-i}]$ i.e. $E[\hat{C}_{i,J}|C_{i,I-i}] = E[C_{i,J}|\mathcal{D}_I]$*

(e) *$\hat{C}^{CL}_{i,J}$ is unconditionally unbiased for $E[C_{i,j}]$, i.e. $E[\hat{C}_{i,J}] = E[C_{i,J}]$*

At first glance, the lack of correlation between the chain ladder estimators $\hat{f}_j$ may seem surprising, given that adjacent estimates of age-to-age factors partially depend on the same data (appearing in both the numerator and the denominator).
Lemma 2 shows that the estimates $\hat{f}_j$ of the age-to-age factors are uncorrelated. It is important to note that this lack of correlation does not imply independence. It can be shown, that the squares of two successive estimators $\hat{f}_j$ and $\hat{f}_{j+1}$ are negatively correlated. Lemma 3(a.) illustrates that the chain ladder factors f_j are estimated by unbiased estimators $\hat{f}_j$(independent of any underlying distributional assumption). This aligns with the selection in equation (2.10). Although other unbiased estimators exist, we chose the estimator defined in Lemma 2 because it satisfies an optimality criterion under specific variance assumptions.
Note that Lemma (d) demonstrates the derivation of unbiased estimators $\hat{C}^{CL}_{i,J}$ for the best estimates $E[\hat{C}^{CL}_{i,J}]$. This justifies the use of the Chain Ladder algorithm in the context of the distribution-free Chain Ladder model.

Proof. We can easily get the proof by using some basic tools.

(a) First, we use the measurability of $C_{i,j}$ with respect to $\mathcal{B}_j$. The independence of different accident years allows us to move the expectation inside the sum. Using assumption 2.4.1, we can rewrite the expectation as follows:

$$\begin{aligned} E\left[\hat{f}_j \mid \mathcal{B}_j\right] &= E\left[\frac{\sum_{i=0}^{I-j-1} C_{i,j+1}}{\sum_{i=0}^{I-j-1} C_{i,j}} \mid \mathcal{B}_j\right] \\ &= \frac{E\left[\sum_{i=0}^{I-j-1} C_{i,j+1} \mid \mathcal{B}_j\right]}{\sum_{i=0}^{I-j-1} C_{i,j}} \\ &= \frac{\sum_{i=0}^{I-j-1} E\left[C_{i,j+1} \mid \mathcal{B}_j\right]}{\sum_{i=0}^{I-j-1} C_{i,j}} \\ &= \frac{\sum_{i=0}^{I-j-1} C_{i,j} f_j}{\sum_{i=0}^{I-j-1} C_{i,j}} \\ &= f_j. \end{aligned}$$

(b) To show that $\hat{f}_j$ is an unconditionally unbiased estimator for f_j, we need to demonstrate that the expectation of $\hat{f}_j$ is equal to f_j without conditioning on $\mathcal{B}_j$. This follows from the law of total expectation. Since $\hat{f}_j$ is an unbiased estimator given $\mathcal{B}_j$, we have:

$$E[\hat{f}_j|\mathcal{B}_j] = f_j$$

By taking the expectation over $\mathcal{B}_j$, we get:

$$E[\hat{f}_j] = E[E[\hat{f}_j|\mathcal{B}_j]] = E[f_j] = f_j$$

Therefore, $\hat{f}_j$ is unconditionally unbiased for f_j.

(c) Now we want to check for a lack of correlation. First, we use the law of total expectation.

$$\begin{aligned} E\left[\hat{f}_0 \dots \hat{f}_j\right] &= E\left[E\left[\hat{f}_0 \dots \hat{f}_j \mid \mathcal{B}_j\right]\right] \\ &= E\left[\hat{f}_0 \dots \hat{f}_{j-1} E\left[\hat{f}_j \mid \mathcal{B}_j\right]\right] \\ &= E\left[\hat{f}_0 \dots \hat{f}_{j-1}\right] f_j \\ &= E\left[\hat{f}_0 \dots \hat{f}_{j-1}\right] E\left[\hat{f}_j\right]. \end{aligned}$$

This can now be done for all CL factors, so by iteration, we arrive at the desired result.

(d) Now, the unbiasedness of the CL estimator:

$$\begin{aligned} E\left[\hat{C}_{i,J} \mid C_{i,I-i}\right] &= E\left[C_{i,I-i}\hat{f}_{I-i} \dots \hat{f}_j \hat{f}_{J-1} \mid C_{i,I-i}\right] \\ &= E\left[C_{i,I-i}\hat{f}_{I-i} \dots \hat{f}_j E\left[\hat{f}_{J-1} \mid C_{i,I-i}\right]\right] \\ &= f_{J-1} E\left[\hat{C}^{CL}_{i,J-1} \mid C_{i,I-i}\right]. \end{aligned}$$

Again, doing this iteratively for every chain Ladder factor we get: don't

$$E\left[\hat{C}^{CL}_{i,J} \mid C_{i,I-i}\right] = C_{i,I-i} f_{I-i} \dots f_{J-1} = E\left[C_{i,J} \mid \mathcal{D}_I\right].$$

(e) To show that $\hat{C}^{CL}_{i,J}$ is unconditionally unbiased for $E[C_{i,J}]$, we need to demonstrate that the expectation of $\hat{C}^{CL}_{i,J}$ equals $E[C_{i,J}]$ without conditioning on $C_{i,I-i}$. This again uses the law of total expectation. From (d), we have shown that:

$$E[\hat{C}^{CL}_{i,J}|C_{i,I-i}] = E[C_{i,J}|\mathcal{D}_I]$$

Taking the expectation over $C_{i,I-i}$, we get:

$$E[\hat{C}^{CL}_{i,J}] = E[E[\hat{C}^{CL}_{i,J}|C_{i,I-i}]] = E[E[C_{i,J}|\mathcal{D}_I]] = E[C_{i,J}]$$

Thus, $\hat{C}_{i,J}^{CL}$ is unconditionally unbiased for $E[C_{i,J}]$.

□

Like in equation (2.6) already described, the reserves can be calculated by

$$R_{i,j} = \sum_{k=j+1}^{J} X_{i,k} = \hat{C}_{i,J}^{CL} - C_{i,j}$$

2.4.2 Explanation of the Procedure

The following explains how the chain ladder method is used for given claim data of J years.

Create a Development Triangle

First, we construct a Triangular Claims Development Table; after that, we arrange the cumulative claims data $C_{i,j}$ into a triangle where i represents the accident year and j represents the development year. The rows represent accident years, and the columns represent development periods. The table usually looks like this:

	Development Year				
Accident Year	0	1	2	$\cdots$	J
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	$\cdots$	$C_{0,J}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	$\cdots$	
2	$C_{2,0}$	$C_{2,1}$	$\cdots$		
$\vdots$	$\vdots$				
I	$C_{I,0}$	$\cdots$			

Calculate Development Factors

Development factors represent the ratio of claims from one development year to the next. We calculate these factors for each development year j by averaging the ratios of cumulative claims:

$$f_j = \frac{\sum_{i=0}^{I-j-1} C_{i,j+1}}{\sum_{i=0}^{I-j-1} C_{i,j}}$$

Apply Development Factors to Project Future Claims

We use the development factors to project future cumulative claims for each accident year. Starting with the known cumulative claims, multiply by the development factors:

$$\hat{C}_{i,j+1} = \hat{C}_{i,j} \cdot f_j$$

We continue this process until the triangle is completed.

Estimate Ultimate Claims for Each Accident Year

The ultimate claims for an accident year are the total cumulative claims once all development periods have been accounted for. We sum the projected claims:

$$\hat{C}_{i,J} = C_{i,J-i} \cdot f_{J-i} \cdot f_{J-i+1} \cdots f_{J-1}$$

Calculate the Outstanding Claims Reserves

The reserve for each accident year is the difference between the estimated ultimate claims and the observed cumulative claims to date. For accident year i:

$$\hat{R}_i = \hat{C}_{i,J} - C_{i,J-i}$$

We sum these reserves to obtain the total reserve estimate:

$$\hat{R} = \sum_{i=0}^{I} \hat{R}_i$$

2.4.3 Advantages and Disadvantages

This subsection is based on [11] and [22]. In the following we compare the strengths and weaknesses of the different models.
Advantages:

- *Simplicity and Popularity*: The Chain Ladder (CL) method is one of the simplest and most widely used loss reserving techniques. It is straightforward to implement and does not require many assumptions, making it accessible and easy to understand.
- *Effective with Stable Patterns*: It is effective when there are stable patterns in historical data, which makes it reliable for predicting future developments if historical trends are consistent.
- *No Need for Distributional Assumptions*: The method is distribution-free, allowing it to be applied in various scenarios without needing an underlying probabilistic model.

Disadvantages:

- *Assumes Constant Development Patterns*: The CL method relies on the assumption that past development patterns will continue in the future. This can be problematic if sudden changes or unusual events occur.
- *Sensitive to Outliers*: The method can be distorted by outliers or irregular claim patterns, leading to inaccurate reserve estimates if anomalies are not addressed .
- *No Built-In Mechanism for Trend Changes*: It does not easily accommodate trend changes, such as sudden shifts in claim frequencies or severities, which might lead to over or under-reserving .

2.4.4 Conclusion

The Chain Ladder Method is a widely used systematic approach for estimating reserves based on historical claims data. By following these steps, actuaries can derive projections for future claims and determine the necessary reserves to cover outstanding liabilities. This method is valuable for its simplicity and effectiveness in providing reserve estimates. In practice, it is used with a few refinements, like the exclusion of outliers, using not the whole claims history but only a few years, if there is a change in claims development.

2.5 Bornhuetter-Ferguson Method

Another ubiquitous model is the Bornhuetter-Ferguson method; in comparison to the usual chain ladder method, it is very robust since it does not consider outliers in the observation but is more complicated, too. The origin of this method is from Bornhuetter and Ferguson [2] who published it in the article "The Actuary and IBNR". Like the chain ladder method, it is a purely mechanical algorithm for estimating reserves, but there are several ways to define a suitable underlying stochastic model for the BF method.

Model Assumptions 1

- Accumulation of claims $C_{i,j}$ from different accident years i are independent.
- There exist parameters $\mu_0, \dots, \mu_I > 0$ and a pattern $\beta_0, \dots, \beta_J > 0$ with $\beta_J = 1$ such that for all $0 \leq i \leq I,\ 0 \leq j \leq J-1$ and $1 \leq k \leq J-j$ we have

$$\begin{aligned} E\big[C_{i,0}\big] &= \beta_0 \mu_i \\ E\big[C_{i,j+k} \big| C_{i,0}, \dots, C_{i,j}\big] &= C_{i,j} + (\beta_{j+k} - \beta_j)\ \mu_i \end{aligned} \tag{2.13}$$

From the model assumptions follow

$$E\Big[C_{i,j}\Big] = \beta_j \mu_i \quad \text{and} \quad E\Big[C_{i,J}\Big] = \mu_i$$

As the equation (2.13) states, μ_i describes the ultimate claim for accident year i. Where β_j expresses the claims development pattern. This means that if we consider $C_{i,j}$ like in the last sections as cumulative payments, then β_j denotes the cumulative cashflow pattern (payout pattern). One use case for such patterns is the application of market-consistent/discounted reserves, which are time-dependent, e.g. inflation. As we can see in the Bornhuetter-Ferguson method we do not only consider the development pattern and start with the claims in the first year, but we estimate the future claims with the ultimate value we expect and the particular development pattern. Sometimes the Bornhuetter-Ferguson model is described by weaker assumptions:

Model Assumptions 2

- Accumulation of claims $C_{i,j}$ from different accident years i are independent.
- There exist parameters $\mu_0, \dots, \mu_I > 0$ and a pattern $\beta_0, \dots, \beta_J > 0$ with $\beta_J = 1$ such that for all $0 \leq i \leq I,\ 0 \leq j \leq J$ we have

$$E\left[C_{i,j}\right] = \beta_j \mu_i$$

Note that these assumptions follow from the assumptions mentioned first. A disadvantage of defining the model with assumption 2 is that it makes the algorithm harder to justify. This is clear by taking a look at the following equation:

$$= C_{i,I-i} + E\left[C_{i,J} - C_{i,I-i} \middle| C_{i,0}, \dots, C_{i,I-i}\right] \tag{2.14}$$

With assumption 2 we know, that $C_{i,J} - C_{i,I-i}$ is independent of $C_{i,0}, \dots, C_{i,I-i}$ which leads to

$$\begin{aligned} E\left[C_{i,J} \middle| \mathcal{D}_I\right] &= E\left[C_{i,J} \middle| C_{i,0}, \dots, C_{i,I-i}\right] \\ &= C_{i,I-i} + E\left[C_{i,J} - C_{i,I-i}\right] \\ &= C_{i,I-i} + \left(\beta_J - \beta_{I-i}\ \right)\mu_i - \underbrace{\left(\beta_{I-i} - \beta_{I-i}\ \right)}_{=0}\mu_i \\ &= C_{i,I-i} + \left(1 - \beta_{I-i}\ \right)\mu_i \end{aligned} \tag{2.15}$$

The problem with assumption 2 is, we can't make the first transformation because of the unknown dependence structure between incremental claims.
In both cases, we still need an estimate for the future claims, which are represented by the last part of the right-hand side in equations (2.14) and (2.15).

Lemma 4 (BF estimator)**.** *The BF estimator for* $E\left[C_{i,J} \middle| \mathcal{D}_I\right]$ *is given by*

$$\hat{C}_{i,J}^{BF} = \hat{E}\left[C_{i,J} \middle| \mathcal{D}_I\right] = C_{i,I-i} + \left(1 - \hat{\beta}_{I-i}\ \right)\hat{\mu_i} \tag{2.16}$$

for $1 \leq i \leq I$*, where* $\hat{\beta}_{I-i}$ *is an appropriate estimate for* β_{I-i} *an* μ_i *is a prior estimate for the expected ultimate claim* $E\left[C_{i,J}\right]$

This estimator gives us the tool to estimate BF reserves. This algorithm is often considered to be purely mechanical and used without consideration of an appropriate underlying stochastic model. However, there is a need for an estimate of the value of the $\hat{\beta}_j$ and $\hat{\mu}_i$

Parameter Estimation

$$E\Big[C_{i,j}\Big] = E\Big[E\Big[C_{i,j}\Big|C_{i,j-1}\Big]\Big] = f_{j-1}\ E\Big[C_{i,j-1}\Big] = E\Big[C_{i,0}\Big]\prod_{k=0}^{j-1} f_k$$

$$E\Big[C_{i,J}\Big] = E\Big[C_{i,0}\Big]\prod_{k=0}^{J-1} f_k$$

If we combine that we get

$$E\Big[C_{i,J}\Big] = \prod_{k=j}^{J-1} f_k^{-1} E\Big[C_{i,J}\Big]$$

By examining the assumption 2, we observe parallels to the representation of $E[C_{i,J}]$ as previously stated. Specifically, the parameter is dependent on i, the accident year. Here, $E[C_{i,J}]$ corresponds to the expected cumulative claims, and $\prod_{k=0}^{J-1} f-1_k$ represents $\beta_j$since it describes the factor already paid from the ultimate claim $\mu_i = E[C_{i,J}]$.
Using the chain-ladder factors, we can estimate β_j. The principle behind the calculation of μ and β is to differentiate external from internal information. μ_i represents the external information, which depicts the ultimate claim. This information can be obtained, for example, from expert opinions. Conversely, $(1-\beta_j)$ describes the proportion of claims that will materialize in the future, making it logical to incorporate available internal information, such as the chain ladder factors. Using the chain ladder factors we get an estimator for β_j

$$\hat{\beta}_j^{(CL)} = \hat{\beta}_j = \left(\frac{1}{\Pi_{k=j}^{J-1} f_k}\right) = \Pi_{k=j}^{J-1}\frac{1}{f_k} \tag{2.17}$$

2.5.1 Advantages and Disadvantages

This list is adapted from [11] and [22]
Advantages:

- *Robustness Against Outliers*: The Bornhuetter-Ferguson (BF) method is robust since it does not only rely on historical development patterns. Combining past data with external estimates (such as expected loss ratios) smooths out anomalies or outliers.
- *Suitable for High Variability*: The BF method is particularly effective for lines of insurance where claims variability is high, especially in early development periods where other methods might struggle.
- *Combination of Approaches*: It merges the expected loss ratio method with the development method, providing a balanced estimate that stabilizes the results.

Disadvantages:

- *Dependency on External Estimates*: The method's accuracy heavily depends on reliable external estimates (such as the expected loss ratio). If these inputs are incorrect, the reserve estimates can be misleading.

2.5.2 Conclusion

The Bornhuetter-Ferguson method is a systematic approach for estimating reserves by combining earlier estimates of ultimate losses with actual loss experience. By following these steps, actuaries get projections for future claims with which they calculate the necessary reserves to cover outstanding liabilities. This model provides more stable estimates by incorporating external information.

2.6 Cape-Cod

The Cape-Cod method is a specialized variant of the Bornhuetter-Ferguson method. It aims to enhance the robustness of the most recent diagonal of the claims development triangle. This adjustment counteracts the common challenge of the Chain Ladder method, which is very sensitive against outliers on the diagonal.

Model Assumptions

- Cumulative claims $C_{i,j}$ of different accident years i are independent.
- There exist parameters $\Pi_0, \ldots, \Pi_I \geq 0,\ \kappa \geq 0$ and a claims development pattern $\beta_0, \ldots, \beta_J > 0$ with $\beta_J = 1$ such that for all $0 \leq i \leq I,\ 0 \leq j \leq J$ we have

$$E\big[C_{i,j}\big] = \kappa\ \Pi_i\ \beta_j$$

Considering assumption 2 from the Bornhuetter-Ferguson method, we observe that using $\mu_i = \kappa\ \Pi_i$ aligns with the same underlying assumptions. In this context, κ can be interpreted as the average loss ratio, while Π_i represents the premium received for accident year i.

$$\hat{\kappa}_i = \frac{\hat{C}_{i,J}^{CL}}{\Pi_i} = \frac{C_{i,I-i}}{\prod_{j=I-i}^{J-1} f_j^{-1} \Pi_i} = \frac{C_{i,I-i}}{\beta_{I-i}\Pi_i}$$

The overall loss ratio $\hat{\kappa}$ is estimated by the weighted average

$$\hat{\kappa}^{CC} = \sum_{i=0}^{i} \frac{\beta_{I-i}\ \Pi_i}{\sum_{k=0}^{I}\ \beta_{I-k}\ \Pi_k}\ \hat{\kappa}_i = \frac{\sum_{i=0}^{I}\ C_{i,I-i}}{\sum_{i=0}^{I}\ \beta_{I-i}\ \Pi_i}$$

Lemma 5 (Cape-Cod estimator). *The Cape-Cod estimator for* $E\left[C_{i,J}\middle|\mathcal{D}_I\right]$ *is given by*

$$\hat{C}_{i,J}^{CC} = C_{i,I-i} - \hat{C}_{i,I-i}^{CC} + \prod_{j=I-i}^{J-1} f_j \hat{C}_{i,I-i}^{CC}, \qquad \textit{for } 1 \le i \le I \tag{2.18}$$

for $1 \le i \le I$*, where* $\hat{\beta}_{I-i}$ *is an appropriate estimate for* β_{I-i} *an* μ_i *is a prior estimate for the expected ultimate claim* $E\left[C_{i,J}\right]$

2.6.1 Advantages and Disadvantages

Advantages:

- *Smoothing Effect*: The Cape Cod method can smooth out outliers by robustly handling diagonal observations, reducing the impact of anomalies that might otherwise distort projections.
- *Adjustment of Estimates*: It uses premium and loss ratio data to create more robust estimates, which can help stabilize results across accident years.
- *Unbiased Estimation*: The method ensures unbiased estimates under its model assumptions, making it reliable when those assumptions are met.

Disadvantages:

- *Dependence on Premium Data*: The method relies heavily on accurate premium information. If the data is inaccurate, the resulting reserve estimates may be misleading.
- *Assumption of Constant Loss Ratios*: The method assumes a constant loss ratio across accident years, which may not be realistic in cases with fluctuating risk profiles.
- *Potential Over-Smoothing*: While smoothing can be advantageous, it may also lead to underestimation of variability, masking real patterns that should be addressed.

2.6.2 Conclusion

The Cape Cod method is a refined version of the Bornhuetter-Ferguson method, which is particularly robust. By gaining robustness, this method loses flexibility and dependency on external data.

2.7 Munich-Chain Ladder

The following section is adapted from [22]. The Munich Chain Ladder method integrates the information available from both paid and incurred data. While in theory, the ultimate paid loss values should align with the ultimate incurred values, significant discrepancies often arise in practice. In the Chain Ladder method, separate analyses are conducted for incurred data and paid data. To reconcile these two estimations, ratios of paid to incurred and incurred to paid are calculated.

$$Q_{i,j} = \frac{C_{i,j}^{Pa}}{C_{i,j}^{In}}, \quad Q_{i,j}^{-1} = \frac{C_{i,j}^{In}}{C_{i,j}^{Pa}}$$

The Munich Chain Ladder (MCL) method uses the ratios of paid to incurred claims (and vice versa) to adjust and smooth development factors. This smoothing means that when the ratio deviates from the average, the development factors for both paid and incurred claims are adjusted in opposite directions to account for this deviation. The introduction of these ratios adds additional model assumptions and structures beyond those present in the traditional Chain Ladder method as described in Mack's model.

Model Assumptions for the Munich Chain Ladder (MCL) Method

Before explaining the model assumptions, we will define the sigma-algebra of the paid, incurred and joint paid and incurred claim data, up to development year j:

$$\mathcal{G}_j^{Pa} = \{C_{ik}^{Pa};\ k \leq j,\ \leq i \leq I\}$$
$$\mathcal{G}_j^{In} = \{C_{ik}^{In};\ k \leq j,\ \leq i \leq I\}$$
$$\widetilde{\mathcal{G}}_j = \{C_{ik}^{Pa}, C_{ik}^{In};\ k \leq j,\ \leq i \leq I\}$$

2.7.1 Model Assumptions

- Cumulative payments $C_{i,j}^{Pa}$ of different accident years are independent. Claims incurred $C_{i,j}^{In}$ of different accident years are independent.
- There exist factors $f_0^{Pa}, \ldots, f_{J-1}^{Pa} > 0$, $f_0^{In}, \ldots, f_{J-1}^{In} > 0$ and variance parameters $\sigma_0^{Pa}, \ldots, \sigma_{J-1}^{Pa} > 0$, $\sigma_0^{In}, \ldots, \sigma_{J-1}^{In} > 0$ such that for all $0 \leq i \leq I$ and $1 \leq j \leq J$ we have

$$E\left[C_{i,j}^{Pa} \middle| \mathcal{G}_{j-1}^{Pa}\right] = f_{j-1}^{Pa}\ C_{i,j-1}^{Pa} \quad \text{and} \quad \mathrm{Var}\left(C_{i,j}^{Pa} \middle| \mathcal{G}_{j-1}^{Pa}\right) = (\sigma_{j-1}^{Pa})^2\ C_{i,j-1}^{Pa} \tag{2.19}$$

$$E\left[C_{i,j}^{In} \middle| \mathcal{G}_{j-1}^{In}\right] = f_{j-1}^{In}\ C_{i,j-1}^{In} \quad \text{and} \quad \mathrm{Var}\left(C_{i,j}^{In} \middle| \mathcal{G}_{j-1}^{In}\right) = (\sigma_{j-1}^{In})^2\ C_{i,j-1}^{In} \tag{2.20}$$

- There are constants $\lambda^{Pa}, \lambda^{In}$ such that for all $0 \le i \le I$ and $1 \le j \le J$ we have

$$\mathbb{E}\left[\frac{C_{i,j}^{Pa}}{C_{i,j-1}^{Pa}}\middle|\mathcal{G}_{j-1}\right] = f_{j-1}^{Pa} + \lambda^{Pa}\mathrm{Var}\left(\frac{C_{i,j}^{Pa}}{C_{i,j-1}^{Pa}}\middle|\mathcal{G}_{j-1}\right)^{1/2} \frac{Q_{i,j-1} - \mathbb{E}\left[Q_{i,j-1}\middle|\mathcal{G}_{j-1}\right]}{\mathrm{Var}\left(Q_{i,j-1}\middle|\mathcal{G}_{j-1}\right)^{1/2}}$$

(9.4) and

$$\mathbb{E}\left[\frac{C_{i,j}^{In}}{C_{i,j-1}^{In}}\middle|\mathcal{G}_{j-1}\right] = f_{j-1}^{In} + \lambda^{In}\mathrm{Var}\left(\frac{C_{i,j}^{In}}{C_{i,j-1}^{In}}\middle|\mathcal{G}_{j-1}\right)^{1/2} \frac{Q_{i,j-1} - \mathbb{E}\left[Q_{i,j-1}\middle|\mathcal{G}_{j-1}\right]}{\mathrm{Var}\left(Q_{i,j-1}\middle|\mathcal{G}_{j-1}\right)^{1/2}}$$

(9.5)

- Different accident years across both cumulative payments $C_{i,j}^{Pa}$ and claims incurred $C_{i,j}^{In}$ are independent, i.e. the sets

$$\left\{C_{0,j}^{Pa}, C_{0,j}^{In}, j = 0, \dots, J\right\}, \dots, \left\{C_{I,j}^{Pa}, C_{I,j}^{In}, j = 0, \dots, J\right\}$$

are independent.

We need to estimate the two correlation coefficients and the four conditional moments. The next estimator uses $\tilde{D}_j$, which describes, similar to equation (2.3), the information about the known observations up to j, including the paid and the incurred data.

MCL Estimators: The MCL estimators are given iteratively by

$$\hat{\mathbb{E}}\left[C_{i,j}^{Pa}\middle|\tilde{\mathcal{D}}_{j-1}\right] = \hat{\mathbb{E}}\left[C_{i,j-1}^{Pa}\middle|\tilde{\mathcal{D}}_{j-1}\right]\left(f_{j-1}^{Pa} + \lambda^{Pa}\frac{\sigma_{j-1}^{Pa}}{\hat{\mathbb{E}}\left[C_{i,j-1}^{Pa}\middle|\tilde{\mathcal{D}}_{j-1}\right]^{1/2}}\tilde{Q}_{i,j-1}^{-1}\right)$$

and

$$\hat{\mathbb{E}}\left[C_{i,j}^{In}\middle|\tilde{\mathcal{D}}_{j-1}\right] = \hat{\mathbb{E}}\left[C_{i,j-1}^{In}\middle|\tilde{\mathcal{D}}_{j-1}\right]\left(f_{j-1}^{In} + \lambda^{In}\frac{\sigma_{j-1}^{In}}{\hat{\mathbb{E}}\left[C_{i,j-1}^{In}\middle|\tilde{\mathcal{D}}_{j-1}\right]^{1/2}}\tilde{Q}_{i,j-1}^{-1}\right)$$

for $i + j > I$, where we set $\hat{\mathbb{E}}\left[C_{i,I-i}^{Pa}\middle|\tilde{\mathcal{D}}_{I-i}\right] = C_{i,I-i}^{Pa}$ and $\hat{\mathbb{E}}\left[C_{i,I-i}^{In}\middle|\tilde{\mathcal{D}}_{I-i}\right] = C_{i,I-i}^{In}$. Hence the MCL estimators for the conditionally expected ultimate claims are defined as

$$\hat{C}_{i,j}^{\mathrm{MCL},Pa} = \hat{\mathbb{E}}\left[C_{i,j}^{Pa}\middle|\tilde{\mathcal{D}}_J\right] \quad \text{and} \quad \hat{C}_{i,j}^{\mathrm{MCL},In} = \hat{\mathbb{E}}\left[C_{i,j}^{In}\middle|\tilde{\mathcal{D}}_J\right]$$

2.7.2 Advantages and Disadvantages

This list is based on [21].
Advantages:

- *Simultaneous Use of Paid and Incurred Data*: The Munich Chain Ladder (MCL) method uses both paid and incurred data, offering a more comprehensive approach compared to the standard CL method.
- *Better Prediction of Future Claims*: By including both datasets, MCL can provide a more balanced estimate, reducing discrepancies between reserves calculated using paid or incurred data separately.
- *Advanced Modelling*: MCL accounts for correlations between paid and incurred claims, leading to more refined estimations where standard assumptions may not hold.

Disadvantages:

- *Complexity*: The method is more complex than standard CL, requiring sophisticated parameter estimation and understanding of correlations, which can complicate implementation.

2.7.3 Conclusion

The Munich-Chain Ladder method is a systematic approach that uses both paid and incurred data. By including both datasets, we gain more advanced modelling and balanced estimation, but we also gain complexity. It should always be weighed whether the complexity outweighs the benefits. This may be the case for complex lines of business that take a long time to process, whereas such advanced modelling is often not required in fast-settling lines of business.

2.8 Tail Adjustment

This section is adapted from [9]. Tail adjustment methods are necessary to estimate claims development beyond the scope of the observed triangle, particularly for long-tail branches where some claims require reserves for 50 years or more. This section introduces several methods for tail estimation, though it is important to note that this is not an exhaustive list. For clarity, the following notations are used:

- $f(n)$: Link ratio for development year n
- $F(n)$: Tail factor for development year n

Bondy Methods

The Bondy method, introduced by Martin Bondy before the 1980s, is one of the earliest tail adjustment techniques. During its development, claims settlement processes were faster, and many branches were short-tailed, allowing simpler methods with less information. Several variations of the Bondy method exist:

- **Original Bondy Method:** This method uses the last link ratio as the tail factor, applying it repeatedly for subsequent periods:

$$F(n) = f(n-1)$$

- **Modified Bondy Method:** This method adjusts the tail factor to account for longer development tails, producing larger values:

$$F_{\text{multiplicative}}(n) = 1 + 2 \cdot \left[f(n-1) - 1\right]$$

$$F_{\text{additive}}(n) = f(n-1)^2$$

Algebraic Methods

Algebraic methods leverage the relationship between paid and incurred data to estimate tail factors. The following are two common approaches:

- **Equalizing Paid and Incurred Development Ultimate Losses:** This method calculates a paid loss tail factor by referencing the incurred loss ultimate as a benchmark. It assumes:
  - Paid and incurred loss development estimate the same ultimate value.
  - Incurred losses for the oldest period are accurate predictors of ultimate values.
  - Other periods will follow the same tail development pattern as the oldest period.

  The tail factor is determined by:

$$F_{\text{paid}} = \frac{\text{Incurred Loss Ultimate for Oldest Year}}{\text{Paid Loss to Date for Oldest Year}}$$

- **Sherman-Boor Method:** This method uses the ratio of case reserves to paid losses to estimate unpaid losses.

Benchmark Methods

Benchmark methods incorporate external data to complement triangle observations. These methods are useful when internal data is insufficient or lacks credibility. Common approaches include:

- **Directly Using Tail Factors from Benchmark Data:** Copy tail factors from industry benchmarks or interpolate/extrapolate from benchmark triangles.
- **Adjusting Benchmark Data to Match Pre-Tail Link Ratios:** Modify benchmark tail factors to reflect differences in pre-tail development patterns of the target dataset.
- **Benchmark Average Ultimate Severity:** Use the ratio of benchmark average severity to reported severity to estimate tail factors.
- **Benchmark Tail Factors Adjusted for Company-Specific Case Reserving:** Align benchmark tail factors with company-specific practices through claims audits.

Curve-Fitting Methods

Curve-fitting methods use mathematical curves to model development patterns and extrapolate tail factors. These approaches assume a relationship between link ratios and development age:

- **Exponential Decay Method:** Fits an exponential curve to development factors, assuming a constant decay rate.
- **Sherman's Method:** Fits inverse power curves to link ratios, sometimes with lag adjustments for improved accuracy.
- **McClenahan's Method:** Models incremental losses decaying at a constant rate after an initial lag, resulting in a closed-form tail factor formula.
- **Pipia's Weibull Curve Method:** Fits a Weibull curve to age-to-age development factors, minimizing squared differences to estimate tail factors.

2.8.1 Conclusion

How to choose the tail adjustment depends on the characteristics of the line of business, data availability. Simpler methods, like the Bondy approach, may suffice for short-tail branches, while curve-fitting or benchmark methods are more suitable for long-tail or volatile branches. Especially for long-tail branches Tail adjustment is crucial for accurate reserving.

2.9 Conclusion

In conclusion, after considering the various advantages and disadvantages of the different claims reserving methods discussed in this chapter, the Chain Ladder model will be the primary focus for the this thesis. The simplicity of the model outweighs the disadvantages, this is why it is widely utilized in actuarial practice. To counteract the limitations of the chain ladder method, actuaries can adjust several key factors, such as:

- Exclusion of outliers
- Tail factor estimation
- Adjustment of the volume (i.e., the number of periods considered)
- Weighting of link ratios

This thesis deals with the influence of changing the volume outlier elimination and tail adjustment on the quality of the model. To address its limitations, particularly its sensitivity to outliers and changes in patterns, we will implement separate outlier detection mechanisms and adapt the volume accordingly. This approach aims to enhance the robustness and reliability of the Chain Ladder method in our analysis.

3 Outlier Detection

As introduced in the previous chapter, the Chain Ladder method is one of the most common approaches for reserve estimation due to its simplicity. However, one disadvantage is sensitivity to extreme values, as it lacks inherent robustness. A natural choice for counteraction is the exclusion of outliers. This chapter presents two outlier detection methods: Reverse Nearest Neighbor and the Interquartile Range.

3.1 Introduction to Outlier Detection

3.1.1 Definition of Outliers

Outliers are typically defined as observations that differ significantly from other observations. According to Hawkins [7],

> An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.

Similarly, Henze [8] defines outliers as observations that diverge to such an extent that they no longer belong to a given data group. In these cases, removing the values may be advisable. To be able to identify outliers we have to define where most of the values lie. This can be done by using statistical measures of dispersion such as:

- **Mean Absolute Deviation**: $\frac{1}{n}\sum_{j=1}^{n}|x_j - \bar{x}|$
- **Sample Range**: $x_{(n)} - x_{(1)} = \max_{1\leq j\leq n} x_j - \min_{1\leq j\leq n} x_j$
- **Interquartile Range (IQR)**: The difference between the upper and lower quartiles.
- **Median Absolute Deviation**

3.1.2 Outlier Detection in Actuarial Models

To illustrate the significant effect that outliers can have on reserve estimations, this section provides an example using the claims development triangle presented earlier in Chapter 2. As discussed, the Chain Ladder method is sensitive to extreme values due to its cumulative structure. We introduce a clear and substantial outlier into the dataset by changing a single data entry in the triangle—changing a value of 800 to 1500 in the development period, which is marked as bold in table 3.1. With this modification, we can examine how one outlier can impact the calculated reserves, which shows the importance of outlier detection for chain ladder models.

Accident year	Development year k					
	0	1	2	3	4	5
2019	1001	854	568	565	347	148
2020	1113	990	671	648	422	
2021	1265	1168	**1500**	744		
2022	1490	1383	1007			
2023	1725	1536				
2024	1889					

Table 3.1: Claims development with an outlier

Accident year	Development year k						Ultimate
	0	1	2	3	4	5	
2019	1001	1855	2423	2988	3335	3483	3483
2020	1113	2103	2774	3422	3844		4015
2021	1265	2433	3933	4677			5471
2022	1490	2873	3880				5511
2023	1725	3261					6505
2024	1889						7157
f_k	-	1.899	1.404	1.214	1.120	1.044	sum = 32,141

Table 3.2: Cumulative claims development with an outlier

Since Chain Ladder reserves are derived from cumulative values, we transform this triangle into a cumulative form:
Table 3.2 shows the cumulative data and the ultimate values for each accident year. The last row shows the chain ladder factors already explained in 2.4. In table 3.3, we provide the same triangle as in table 3.2, but without the outlier to highlight its impact on ultimate reserves.
Comparing the ultimate reserve sums in table 3.2 and table 3.3, we see that a single outlier can significantly increase reserve estimates. This single outlier does not only have an impact on year 2021's reserves but on 2022, too, since the chain ladder factors change. This results in the drastic ultimate change which impacts the reserves. This emphasizes the importance of reliable outlier detection within Chain Ladder-based reserving methods since outlier elimination can enhance the robustness of the model drastically.

Accident year	Development year k						Ultimate
	0	1	2	3	4	5	
2019	1001	1855	2423	2988	3335	3483	3483
2020	1113	2103	2774	3422	3844		4015
2021	1265	2433	3233	3977			4652
2022	1490	2873	3880				5591
2023	1725	3261					6245
2024	1889						6871
f_k	-	1.899	1.329	1.232	1.120	1.044	sum = 27,375

Table 3.3: Cumulative claims development without outlier

3.2 Reverse Nearest Neighbour

This section is based on [17]. As proposed by Korn and Muthukrishnan [12], this principle doesn't use classic statistical measures of dispersion to detect outliers. Whether a data point is classified as an outlier depends on the number of other points for which it is the nearest neighbor. The procedure is illustrated in Figure 3.1.

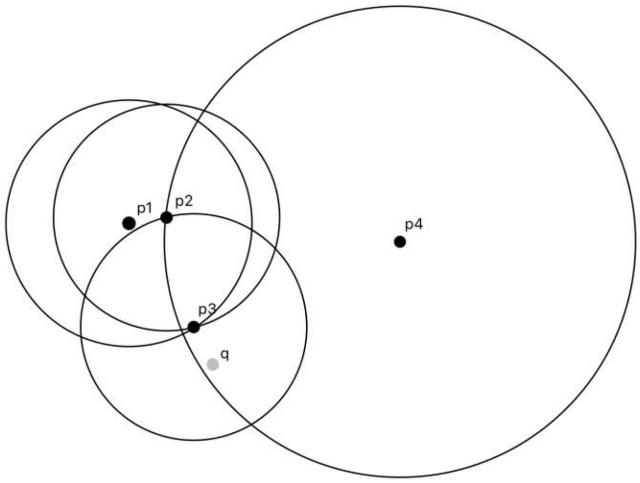


Figure 3.1: Reverse nearest neighbour; adapted from [17]

Visualisation of the 2NNs of the points $\{p_1, p_2, p_3, p_4\}$ in a set of five points in a two-dimensional space: $\{q, p_1, p_2, p_3, p_4\}$.

For a given multidimensional data set P and a point q, the RNN algorithm provides all points p in P that have q as NN. We can call the set RNN(q) the *influence set of q*

$$\text{RNN}(q) = \{p \in P \mid \nexists\, p' \in P : \text{dist}(p, p') < \text{dist}(p, q)\}$$

Note, that *dist* is in the following assumed as Euclidian distance, but it can be any distance metric.
A natural extension of this definition is the *reverse k nearest neighbor (RkNN)*. In this query the distinction between how far away are the different neighbours and ranks them according to their distance.

$$\text{RkNN}(q) = \{p \in P \mid \text{dist}(p, q) < \text{dist}(p, p_k) : p_k \text{ is the } k\text{-th NN of } p\}$$

Example

To illustrate how the query works we demonstrate the R2NN outlier detection with a set of five points in a two-dimensional space: $\{q, p_1, p_2, p_3, p_4\}$. To determine the R2NNs of q, we must first identify the 2NNs of the other points, illustrated with circles around the points in Figure 3.1. The reverse nearest neighbours of q are the points where q is included in the set of nearest neighbours of the other points. In this case, R2NN(q) = $\{p_3, p_4\}$ since q is nearer than the second nearest neighbour of p_3 and p_4.
It is important to note that p in NN does not necessarily result in q in RNN. This can be easily shown by examining the 2NN of q, which are p_3 and p_1; however, as mentioned earlier, p_1 is not an RNN of q. For further, more complex versions, the interested reader is referred to [17].

3.2.1 Applications of the Reverse Nearest Neighbour method

Some possible application scenarios are:

- **Profile-Based Marketing:** A real estate company profiles customers based on their preferences in a feature space (e.g., house area, neighborhood). When a new property enters the market, a Reverse Nearest Neighbor (RNN) query finds the clients for whom the new property is the closest match to their interests.
- **Decision Support Systems:**A franchise wants to open a new branch at location q to attract customers from competitors based on proximity. This scenario can be modeled as a bichromatic Reverse Nearest Neighbor (RNN) query, where one set P_1 represents competitors and the other set P_2 represents customers. The result identifies customers closer to q than to any competitor.
- **Peer-to-Peer Systems:** When a new user q enters a P2P system, a Reverse Nearest Neighbor (RNN) query identifies existing users for whom q will be their new nearest neighbor based on network latency. In a collaborative environment, q informs these users about its arrival to minimize future network costs. The RNN set also indicates q's potential workload, enabling resource management and control.

- **Outlier Detection:** RNN can be also used as an outlier detection, if we define an outlier as a point which has no RNN [18].

3.3 Interquartile Range (IQR)

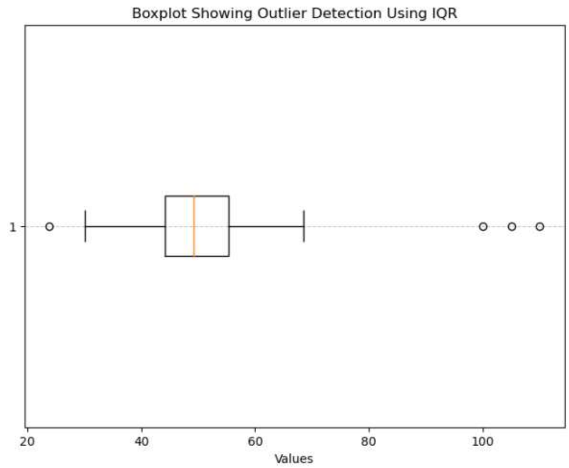


Figure 3.2: Boxplot with random data and outliers

Visual representation of the IQR an outliers regarding the IQR measure, represented by dots.

This section is based on [6]. The Interquartile Range (IQR) is a measure of statistical dispersion commonly used to identify outliers in continuous data. The IQR is calculated as the difference between the third quartile (Q3) and the first quartile (Q1), representing the middle 50% of the data:

$$\text{IQR} = Q3 - Q1.$$

The dataset is divided into quartiles: Q1 (first quartile), Q2 (second quartile or median), and Q3 (third quartile). Quartiles describe the distribution of data as follows:

- Q1: $x_{0.25}$, the value below which 25% of the data lie,
- Q2: $x_{0.50}$, the median, where 50% of the data lie below,
- Q3: $x_{0.75}$, the value below which 75% of the data lie.

Outliers are identified as data points that fall outside the lower and upper boundaries:

$$\text{Lower boundary} = Q1 - (1.5 \times \text{IQR}),$$
$$\text{Upper boundary} = Q3 + (1.5 \times \text{IQR}).$$

Figure 3.2 shows a boxplot, which visually represents the IQR and highlights outliers beyond these boundaries. The visual separation of the data median to the rest makes it easy to detect and interpret outliers. This method provides a straightforward way to detect and interpret outliers in datasets.

4 Error measure

This chapter presents two error measures that are suitable for application in this thesis. The first one is a very common measure, the Mack Standard Error introduced by Mack in [13], giving a long-term approach, while the Claims Development Result (CDR) introduced by Mertz and Wuethrich [14] gives a short-term approach by considering the fluctuations of the ultimate values that arise with new information.

4.1 Mack Standard Error

The following section is adapted from [13]. Mack introduced the Mack Standard Error, a long-term approach, in [13]. A long-term approach means we are interested in the overall error and not only in the deviation from one year to another. This measure is specifically for the Chain Ladder method. The MSE is the standard error of the chain ladder estimates. It is a measure of the uncertainty contained in the data to see whether the difference between the results of the chain ladder method and any other method is significant or not. The mean squared error *mse($\hat{C}_{iI}$)* of the estimator $\hat{C}_{iI}$ of C_{iI} is:

$$mse(\hat{C}_{iI}) = E((\hat{C}_{iI} - C_{iI})^2|\mathcal{D})$$

note that $\mathcal{D}_I = \{C_{ik}|i + k \leq I + 1\}$ is the information available.

$$mse(\hat{R}_i) = E((\hat{R}_i - R_i)^2|\mathcal{D}) = E((\hat{C}_{iI} - C_{iI})^2|\mathcal{D}) = mse(\hat{C}_{iI})$$

We want to calculate the expected quadratic error of the reserve. First, we are able to split the *mse* in a sum of the stochastic error and the estimation error, because a property of the expectation: $\mathbb{E}[(X - a)^2] = \mathrm{Var}(X) + (\mathbb{E}[X] - a)^2$

$$\mathrm{mse}(\hat{C}_{il}) = \mathrm{Var}(C_{il} \mid \mathcal{D}) + (\mathbb{E}[\hat{C}_{il} \mid \mathcal{D}] - \hat{C}_{il})^2 \tag{4.1}$$

When the chain ladder method was introduced, we only had model assumptions regarding the first moments, but since we need to calculate the variance, we need an additional assumption about the variance.

$$\mathrm{Var}(C_{i,k+1} \mid C_{1k}, \ldots, C_{ik}) = C_{ik}\sigma_k^2, \quad 1 \leq i \leq l, \quad 1 \leq k \leq l - 1 \tag{4.2}$$

where σ_k^2, $1 \leq k \leq l - 1$ are unknown parameters. This parameter can be estimated by:

$$\hat{\sigma}_k^2 = \frac{1}{I - k - 1} \sum_{i=1}^{I-k} C_{ik} \left(\frac{C_{i,k+1}}{C_{ik}} - \hat{f}_k \right)^2, \quad 1 \leq k \leq I - 2 \tag{4.3}$$

Note that $\hat{\sigma}_k^2$ are unbiased estimators of σ_k^2, $1 \leq k \leq I-1$. Nevertheless, there is still one estimator $\hat{\sigma}_{I-1}$ missing. We can differentiate between two different cases. In the first case, no further claims are expected since $\hat{f}_{I-1} = 0$. If this is the case, the estimate can no longer fluctuate, whereby $\hat{\sigma}_{I-1} = 0$. If this is not the case, then we extrapolate $\sigma_1, \ldots, \sigma_{I-2}$ by one additional member. The extrapolation can be made either with a log-linear regression or a much simpler approach by demanding:

$$\frac{\hat{\sigma}_{l-3}}{\hat{\sigma}_{l-2}} = \frac{\hat{\sigma}_{l-2}}{\hat{\sigma}_{l-1}}$$

if $\hat{\sigma}_{l-3} > \hat{\sigma}_{l-2}$. Then we can assume:

$$\hat{\sigma}_{I-1}^2 = min(\hat{\sigma}_{I-2}^4 / \hat{\sigma}_{I-3}^2, min(\hat{\sigma}_{I-3}^2, \hat{\sigma}_{I-2}^2))$$

Lemma 6. *Under the chain ladder assumptions 2.7 and the additional assumption 4.2 stated above an estimation of the mean squared error for the accident year i $mse(\hat{R}_i)$ can be made by:*

$$\widehat{mse}\,(\hat{R}_i) = \hat{C}_{il}^2 \sum_{k=I+1-i}^{I-1} \frac{\hat{\sigma}_k^2}{\hat{f}_k^2} \left(\frac{1}{\hat{C}_{ik}} + \frac{1}{\sum_{j=1}^{I-k} C_{jk}} \right) \tag{4.4}$$

with $\hat{C}_{ik} = C_{i,I+1-i} \cdot \hat{f}_{i+1-i} \cdot \cdots \cdot \hat{f}_{k-1}$, $k > I+1-i$, *are the estimates of the future* C_{ik} *and* $\hat{C}_{i,I+1-i} = \hat{C}_{i,I+1-i}$.

Proof. For a better readability we use the abbreviations:

$$E_i(X) = E(X \mid C_{i1}, \ldots, C_{i,I+1-i}),$$
$$\mathrm{Var}_i(X) = \mathrm{Var}(X \mid C_{i1}, \ldots, C_{i,I+1-i}).$$

We will use the property of the mse mentioned above, that we can write it as a sum.

$$\mathrm{mse}(\hat{R}_i) = \mathrm{Var}(C_{iI} \mid \mathcal{D}) + (E(C_{iJ} \mid \mathcal{D}) - \hat{C}_{iJ})^2.$$

First, we rewrite the variance into a term that we can use in our estimators, which easily follows from the chain ladder assumptions and the variance assumption, then we rewrite the second part:

$$\begin{aligned}
\mathrm{Var}(C_{iI} \mid \mathcal{D}) =& \mathrm{Var}_i(C_{iI}) \\
=& E_i(\mathrm{Var}(C_{iI} \mid C_{i1}, \dots, C_{i,I-1})) + \\
& + \mathrm{Var}_i(E(C_{iI} \mid C_{i1}, \dots, C_{i,I-1})) \\
=& E_i(C_{i,I-1})\sigma_{I-1}^2 + \mathrm{Var}_i(C_{i,I-1})f_{I-1}^2 \\
=& E_i(C_{i,I-2})f_{I-2}\sigma_{I-1}^2 + E_i(C_{i,I-2})\sigma_{I-2}^2 f_{I-1}^2 + \\
& + \mathrm{Var}_i(C_{i,I-2})f_{I-2}^2 f_{I-1}^2 \\
=& \text{ etc.} \\
=& C_{i,I+1-i} \sum_{k=I+1-i}^{l-1} f_{I+1-i} \cdots f_{k-1}\sigma_k^2 f_{k+1}^2 \cdots f_{I-1}^2
\end{aligned}$$

Because of $\mathrm{Var}_i(C_{i,I+1-i}) = 0$, the estimation error can be written like:

$$(E(C_{iI} \mid \mathcal{D}) - \hat{C}_{iI})^2 = C_{i,I+1-i}^2 \left((f_{I+1-i} \cdot \dots \cdot f_{l-1} - \hat{f}_{I+1-i} \cdot \dots \cdot \hat{f}_{I-1}) \right)^2.$$

Now, we have a representation of the *mse* where we could put in our estimators. This works with the variance, but the estimation error presents a problem since this would yield 0.

$$\begin{aligned}
\mathrm{Var}(C_{iI} \mid \mathcal{D}) =& C_{i,I+1-i} \left(\sum_{k=I+1-i}^{I-1} \hat{f}_{I+1-i} \cdot \dots \cdot \hat{f}_{k-1} \cdot \sigma_k^2 \hat{f}_{k+1}^2 \cdot \dots \cdot \hat{f}_{I-1}^2 \right) \\
=& \hat{C}_{i,I}^2 \sum_{k=I+1-i}^{I-1} \frac{\hat{\sigma}_k^2 / \hat{f}_k^2}{\hat{C}_{ik}}
\end{aligned}$$

The second term can be estimated with a little trick. We rewrite the part which would yield 0 into something postiv.

$$\begin{aligned}
F &= f_{I+1-i} \cdot \dots \cdot f_{I-1} - \hat{f}_{I+1-i} \cdot \dots \cdot \hat{f}_{I-1} \\
&= S_{I+1-i} + \dots \cdot + S_{I-1}
\end{aligned}$$

where

$$S_k = \hat{f}_{t+1-i} \cdots \hat{f}_{k-1}(f_k - \hat{f}_k) f_{k+1} \cdots f_{I-1}.$$

If we look at the square of F we can rewrite it due to the algebraic expansion of a squared sum.

$$\begin{aligned} F^2 &= (S_{I+1-i} + \cdots + S_{I-1})^2 \\ &= \sum_{k=I+1-i}^{I-1} S_k^2 + 2\sum_{j<k} S_j S_k. \end{aligned}$$

To further rewrite the equation we take a look at $E(S_k^2 \mid \mathcal{B}_k)$ instead of S_k^2 and S_jS_k, $j < k$, with $E(S_jS_k \mid \mathcal{B}_k)$. Note that B_k represents the claim available until year k: $\mathcal{B}_k = \{C_{ij} \mid j \le k, i + j \le I + 1\}$, for $1 \le k \le I$. Since the age-to-age factors are uncorrelated and hence $E((f_k - \hat{f}_k) \mid \mathcal{B}_k) = 0$, we know that $E(S_jS_k \mid \mathcal{B}_k) = 0$ holds for $j < k$.

To counteract the issue To address the issue with the estimation error yielding zero and rewrite the sum of the squares, we examine the following:

$$\begin{aligned} E((f_k - \hat{f}_k)^2 \mid \mathcal{B}_k) &= \mathrm{Var}(f_k \mid \mathcal{B}_k) \\ &= \sum_{j=1}^{I-k} \mathrm{Var}(C_{j,k+1} \mid \mathcal{B}_k) \Big/ \left(\sum_{j=1}^{I-k} C_{jk}\right)^2 \\ &= \sigma_k^2 \Big/ \sum_{j=1}^{I-k} C_{jk} \end{aligned}$$

Now we can simply replace the term $(f_k - \hat{f}_k)^2$.

$$\begin{aligned} E(S_k^2 \mid B_k) &= \hat{f}_{I+1-i}^2 \cdot \cdots \cdot f_{k-1}^2 E((f_k - \hat{f}_k)^2 \mid \mathcal{B}_k) f_{k+1}^2 \cdot \cdots \cdot f_{I-1}^2 \\ &= \hat{f}_{I+1-i}^2 \cdot \cdots \cdot \hat{f}_{k-1}^2 \sigma_k^2 f_{k+1}^2 \cdot \cdots \cdot f_{I-1}^2 \Big/ \sum_{i=1}^{I-k} C_{ik}. \end{aligned}$$

The sum consists now of only positive values, which means If we now look at $F^2 = \sum S_k^2$ with $\sum_k E(S_k^2 \mid \mathcal{B}_k)$ we can put in the unbiased estimators $\hat{f}_k, \hat{\sigma}_k^2$, instead of the unknown parameters f_k, σ_k^2. This gives us:

$$\begin{aligned} F^2 &= (f_{I+1-i} \cdots f_{I-1} - \hat{f}_{I+1-i} \cdots \hat{f}_{I-1})^2 \\ &\approx \sum_{k=I+1-i}^{l-1} \left(\hat{f}_{I+1-i}^2 \cdots \hat{f}_{k-1}^2 \hat{\sigma}_k^2 \hat{f}_{k+1}^2 \cdots \hat{f}_{I-1}^2 \Big/ \sum_{j=1}^{I-k} \frac{1}{C_{jk}} \right) \\ &= \hat{f}_{I+1-i}^2 \cdots \hat{f}_{I-1}^2 \sum_{k=t+1-i}^{I-1} \frac{\hat{\sigma}_k^2 / \hat{f}_k^2}{\sum_{j=1}^{I-k} C_{ik}}. \end{aligned}$$

Since the product of the squares of the age to age factors is the square of the estimator of C_{iI} simply adding up the results we get the estimator. $\square$

Most of the time, the focus is on the overall year, but because of the correlation of the yearly reserves via the mutual estimators $\hat{f}_k$ and $\hat{\sigma}_k$, we cannot simply use the use but need a different formula. In the following Lemma, we will use the term "s.e. $(\hat{R}_i)$", it is called the *standard error of* $\hat{R}_i$ and represents the square root of an estimator of the mean squared error.

Lemma 7. *Given the chain ladder assumptions and the additional assumption regarding the variance, The overall reserve estimate* $\hat{R} = \hat{R}_2 + \cdots + \hat{R}_I$ *can be estimated by*

$$\widehat{mse}\ (\hat{R}) = \sum_{i=2}^{I} \left\{ ((s.e.)(\hat{R}_i))^2 + \hat{C}_{iI} \left(\sum_{j=i+1}^{I} \hat{C}_{jI} \right) \sum_{k=I+1-i}^{I-1} \frac{2\hat{\sigma}_k^2 / \hat{f}_k^2}{\sum_{n=1}^{I-k} C_{nk}} \right\} \tag{4.5}$$

Proof. For the MSE over all years:

$$\begin{aligned} \text{mse}\left(\sum_{i=2}^{I} \hat{R}_i \right) &= E\left(\left(\sum_{i=2}^{I} \hat{R}_i - \sum_{i=2}^{I} R_i \right)^2 \middle| \mathcal{D} \right) \\ &= E\left(\left(\sum_{i=2}^{I} \hat{C}_{iI} - \sum_{i=2}^{I} C_{iI} \right)^2 \middle| \mathcal{D} \right) \\ &= \text{Var}\left(\sum_{i=2}^{I} C_{iI} \middle| \mathcal{D} \right) + \left(E\left(\sum_{i=2}^{I} C_{iI} \middle| \mathcal{D} \right) - \sum_{i=2}^{I} \hat{C}_{iI} \right)^2. \end{aligned}$$

Because of the assumption, that the accident years are independent we get

$$\begin{aligned} \text{Var}\left(\sum_{i=2}^{I} C_{iI} \middle| \mathcal{D} \right) &= \sum_{i=2}^{I} \text{Var}\ (C_{iI} | \mathcal{D}) \\ &= \sum_{i=2}^{I} C_{i,I+1-i} \sum_{k=I+1-i}^{l-1} f_{I+1-i} \cdots f_{k-1} \sigma_k^2 f_{k+1}^2 \cdots f_{I-1}^2 \end{aligned}$$

The second equation was proven above in the last proof.

$$\begin{aligned} \left(E\left(\sum_{i=2}^{I} C_{iI} \middle| \mathcal{D} \right) - \sum_{i=2}^{I} \hat{C}_{iI} \right)^2 &= \left(\sum_{i=2}^{I} \left(E\ (C_{iI} | \mathcal{D}) - \hat{C}_{iI} \right) \right)^2 \\ &= \sum_{i,j} \left(E\ (C_{iI} | \mathcal{D}) - \hat{C}_{iI} \right) \cdot \left(E\ (C_{iI} | \mathcal{D}) - \hat{C}_{iI} \right) \\ &= \sum_{i,j} C_{i,I+1-i} C_{j,I+1-j} F_i F_j \end{aligned}$$

Where F is defined like above

$$F_i = f_{I+1-i} \cdot \dots \cdot f_{I-1} - \hat{f}_{I+1-i} \cdot \dots \cdot \hat{f}_{I-1}$$

We get:

$$\mathrm{mse}(\hat{R}_i) = \mathrm{Var}(C_{iI} \mid \mathcal{D}) + (C_{i,I+1-i}F_i)^2$$

We can conclude from this

$$\mathrm{mse}\left(\sum_{i=2}^{I} \hat{R}_i\right) = \sum_{i=2}^{I} \mathrm{mse}(\hat{R}_I) + \sum_{1 \leq i < jleqI} 2 \cdot C_{i,I-1-i} C_{j,I+1-j} F_i F_j.$$

A similar procedure like in the last proof gives us the estimator

$$\sum_{k=I+1-i}^{I-1} \hat{f}_{I+1-j} \cdots \hat{f}_{I-i} \hat{f}^2_{I+1-i} \cdots \hat{f}^2_{k-1} \hat{\sigma}^2_k \hat{f}^2_{k+1} \cdots \hat{f}^2_{I-1} \Big/ \sum_{n=1}^{I-k} C_{nk}.$$

The assertion follows by combining the expressions. □

4.2 Claims Development Result

The following section is from [1]. The Claims Development Result (CDR) focuses on the fluctuation of the ultimate estimation of consecutive years. This measure is calculated retrospectively and does not rely on any assumptions about the distribution of claims.

As described in chapter 2, the IBNR (Incurrred But Not Reported) reserves are calculated based on ultimate claims estimations. We gain more information about the claim's development each new year, and consequently, the ultimate estimation and reserves change. This raises the question of the relevance of examining the reserves depending on at which time k they were made. In an ideal scenario with perfect forecasts, this would be redundant since there would be already enough information, and it would stay the same with further information. As perfect forecasts are impossible, there will always be fluctuations over the years. Smaller fluctuations are preferable to bigger ones, as they indicate a more accurate and stable prediction. A number which describes this fluctuation is the *Claims Development Result* (CDR). The CDR represents the "difference between forecasts of the ultimate claims in consecutive periods", as introduced by Merz and Wuethrich [14]. The ultimate claims estimate changes with new information, resulting in potential profit or loss variations. The CDR serves as a measure of these changes.

In the following equations, we define the terms as follows:

- J: Last development period.
- j^*: Last development period for which information is available at time k.
- $C_{i,J}^k$: Ultimate value for accident year i, calculated using information available at time k.
- R_{i,j^*}^k: IBNR reserves for accident year i, calculated using information available at time k.
- $\hat{X}_{i,l}$: Expected incremental claims for accident year i in development period l.
- $X_{i,l}$: Actual incremental claims for accident year i in last available development period j^*.
- AvE_{i,j^*}: Actual versus Expected (AvE) result of the incremental claims on the new diagonal of the triangle.

The CDR for accident year i in calendar year k is defined as:

$$\begin{aligned}
CDR_i^k &= C_{i,J}^k - C_{i,J}^{k-1} \\
&= R_{i,j^*}^k + C_{i,j^*} - (R_{i,j^*}^{k-1} + C_{i,j^*}) \\
&= \sum_{l=j^*}^{J} \hat{X}_{i,l}^k + C_{i,j^*} - \Big(\sum_{l=j^*-1}^{J} \hat{X}_{i,l}^{k-1} + C_{i,j^*-1} \Big) \\
&= R_{i,j^*}^k - R_{i,j^*}^{k-1} + (R_{i,j^*} - \hat{X}_{i,j^*}^{k-1}) \\
&= R_{i,j^*}^k - R_{i,j^*}^{k-1} + AvE_{i,j^*}^k
\end{aligned}$$

The CDR describes the difference in the ultimate estimation between two consecutive years. As we can see in the equation, the CDR can be rewritten as a sum of:

- Difference of the IBNR Reserves of two consecutive years
- Difference between estimated and incremental claims for the new period.

The difference in IBNR reserves may result from claims payments made in the new year (which no longer require reserving) or changes in claim information, such as a court decision that releases the insurer from liability. The difference between estimated and incremental claims results in the new period is the object of the Actual-Versus-Expected (Ave) analysis actuaries use to make adjustments in the new IBNR reserves. The two parts of the CDR are crucial figures for actuaries. Although the CDR can only be calculated for past years, calculating it for many triangles can determine if a reserving model fits the given data.

Model Optimization Using the CDR

In this thesis, we determine the optimal model by examining the CDR. The goal is to identify a model that produces CDR values close to zero. To achieve this, the squared difference between the CDR and zero is minimized, i.e., $(CDR - 0)^2$. To avoid distortion of the CDR due to incremental claims, we look at the CDR score instead of the simple squared CDR. The score is calculated by weighing the absolute value of the incurred claims:

$$CDR_{score} = \sqrt{\frac{\sum_{i=1}^{I} |X_{i,j^*}|(CDR_i^k)^2}{\sum_{i=1}^{I} |X_{i,j^*}|}} \tag{4.6}$$

4.3 Conclusion

In this thesis we primarily study the optimal model using the CDR score. It also compares how the Mack Standard Error evolves in contrast to the CDR. Furthermore, we analyze the Actual-Versus-Expected (AvE) outcomes for the optimal model based on the CDR and the standard Chain Ladder model. The Mack Standard Error measures the uncertainty within the development triangle. If we want to know the uncertainty in the following years, we must look at the MSE. For short-tail branches, it likely will not make a significant difference whether we look at the MSE of the Continuous Daily Rate (CDR) or the overall MSE, since most of the uncertainty is concentrated in the next year. In contrast, a more significant difference is expected for Long-Tail branches, although it should be noted that approximately 75% of the uncertainty still lies in the following year.

5 The Algorithm

The previous chapters introduced various reserving models, outlier detection methods and error metrics. This chapter presents the algorithm identifying the best-performing reserving model for the given models. In chapter 2, we have already established the basic models and decided to proceed with the Chain Ladder model. The criteria for model performance are mainly the Claims Development Result; the Mack Standard Error will only be used as a comparative measure. The critical question is how the optimization process works in practice.

The core idea is as follows: the process begins with a small initial triangle representing a limited dataset for each model. This triangle is gradually expanded step by step by adding new diagonals of claims data, thus increasing the amount of available information. The CDR score is calculated at every step to evaluate the model's performance. Once the CDR for the whole triangle, including all available data, has been computed, the average CDR score across all steps is determined and recorded for that specific model. This process is repeated for all candidate models under consideration. By systematically comparing the average CDR scores of all models, the optimal model is identified as the one with the lowest average CDR score, since the overall average score provides a strong indicator of overall model performance.

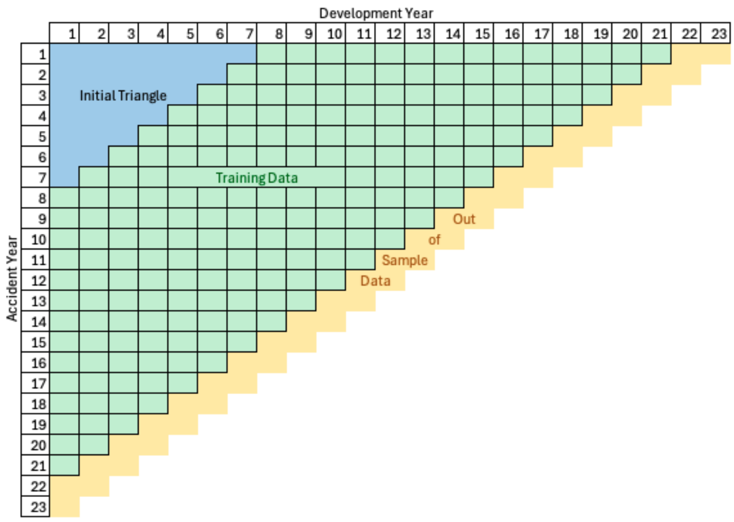


Figure 5.1: Full Triangle, adapted from [1]

Breakdown of the triangle data in the groups: 'initial triangle', 'training data' and 'out of sample data'.

5.1 The Algorithm

The following section describes the algorithm selecting the optimal reserving model based on the Claims Development Result (CDR) scores or Mack Standard Error. While the general process remains consistent with [1], one adjustment has been made: the reserves are calculated for the initial triangle before adding the first calendar period.$\mathcal{M}$ is a set of all models that will be compared. In the following workflow the algorithm is presented:

- First, we will decide on an "initial triangle" to supply data to fit all the models. The "initial triangle" is visualized in figure 5.1. In the following steps, it will be expanded to include new diagonals of experience.
- Select all but the last two years of the remaining data as the training set (shown as "Training Data" in green in figure 5.1)
- Select an arbitrary reserving model $M \in \mathcal{M}$ which has not been used yet. For each M, perform the following steps:
  1. Calculating the reserves for the *initial triangle*. This is done by creating a model with the chosen parameters for M.
  2. For each subsequent calendar year k in the training set:
     a) Calculate the score for each accident year as CDR_i^k based on the next diagonal of experience, or the MSE. Note that at this stage, this next diagonal has not been used to fit the reserving mode, i.e. it is out of sample data.- Calculate the reserves with the newly won data. The data available in the first step is illustrated in Figure 5.2)
     b) Calulate the weighted score across accident years, using the incremental claims as weights (as in equation 4.6)
     c) The new reserves $R_{i,j}^k = M(X = \Delta^k)$ are estimated by extending the model with the latest data. Figure 5.2 and 5.3 illustrate the first two iterations.
  3. Calculate and store the average score across all of the calendar years in the training set, S^M.
- The optimal model is chosen as the model with the lowest average score across all off the calendar years.

$$M_{\text{opt}} = argmin_{M \in \mathcal{M}}(S^M)$$

Setting in this thesis

In our case, the set $\mathcal{M}$ only consists of Chain Ladder models, differing in volume, outlier detection and tail estimation.

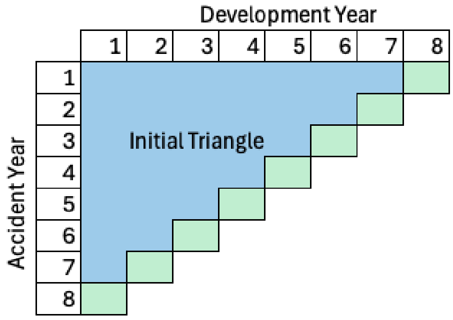


Figure 5.2: Data available in the first step, adapted from [1]

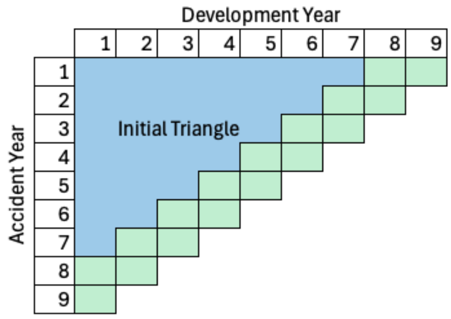


Figure 5.3: Data available in the second step, adapted from [1]

6 Methodology and Data Foundation

This thesis takes a practical approach by focusing on an optimisation algorithm based on the Chain Ladder method for claims reserving. The goal is to find the optimal model according to the CDR by adapting the number of periods to consider, handling outliers, experimenting with different tail factors and adjusting for inflation. By performing a case study, we can expose the challenges associated with the models and show possible solutions. For instance, looking closely at the reserving process of a specific insurance company allows for a deeper understanding than broader surveys or purely numerical analyses could provide. By working with actual data from a particular company, we can develop practical solutions that are both theoretically sound and implementable. This chapter gives insight into the data used and the concrete implementation of the code.

6.1 Data Foundation

The following section provides an overview of the dataset used for this study. It describes the data's origins, structure, and key characteristics, offering insight into how the information has been organized and prepared for analysis. Understanding the dataset is essential for appreciating the challenges and limitations associated with claims reserving.

6.1.1 Description of the Used Data

The data comes from the insurance company Allianz, consisting of claims data including payment information, reserves, and the number of incidents. Due to privacy reasons, some values have been adjusted. There are three different major types of insurance branches, when looking at the development pattern:

- **Short-Tail:** Short-tail branches refer to insurance lines characterized by quick and straightforward settlement processes. A typical example is motor vehicle collision insurance, where the extent of the damage is immediately visible, and the claim amount can be estimated promptly and accurately. Due to the clarity of the damage, it is uncommon for there to be many incurred but not reported (IBNR) claims extending beyond one year.
- **Long-Tail:** Long-tail businesses require a longer development period Classic examples of long-tail insurance policies include legal protection insurance and motor vehicle liability insurance. One explanation for this is that Austrian law mandates liability insurance to cover damages without any limitation periods. For instance, if an accident causes injury to a child, reserves must be maintained for this accident for the duration of the child's life, as there may be circumstances where compensation or even pension payments are required long after the incident.

- **Volatile:** Segments with a volatile settlement pattern are typically characterized by large claims. Large claims do not occur with the same intensity every year; their frequency and severity can vary greatly. Moreover, estimating the final claim amount is not as straightforward as in other segments, such as motor hull insurance, because these claims tend to be more complex and extensive. For instance, the flood damages associated with the construction of the subway during the 2024 autumn floods illustrate that such damages are challenging to predict and assess accurately.

The data used in this thesis consists of information about one short-tail, one long-tail, and one volatile branch. Additionally, the data includes several outlier years between 2000 and 2022, influenced by major events such as the 2007 financial crisis and the economic impacts of the COVID-19 pandemic.

6.1.2 Data Structure and Characteristics

The data is stored in a table in an xls file. The dataset consists of a table with five columns, organized to represent various attributes related to claims reserves. The columns are defined as follows:

- **Segment**: A categorical variable indicating the insurance branches: LongTail, ShortTail and Volatile.
- **Payment_inc**: A numerical value representing the incurred payments related to claims, expressed in euros.
- **Reserve_inc**: A numerical value indicating the incurred reserves for future claims payments, also expressed in euros.
- **Amountl_inc**: A numerical count of incidents or claims associated with the respective segment.
- **Accident Year**: The year when the claim was incurred.
- **Development Year**: The development year.

The data spans from 2000-2022 with entries in 1999 and 2099. The outlier detection has been done with Python, the implementation is shown in the next section in *Implementation of Models* 6.2.

6.2 Implementation of Models

In this section, the implementation of the algorithm is given, including a general explanation of the Python Environment, an overview of the most important classes, objects, and methods, and some concrete code snippets.

6.2.1 Explanation of the Python Environment

Generally, the code for this thesis is written in Python, primarily using the PyCharm interpreter. For certain visualizations, Jupyter Notebook was utilized. The most relevant packages employed include the Chainladder package, NumPy, and Pandas. **NumPy** is a Python library that provides tools for numerical computing. It is used, for instance, to replace NaN values with zeros or to create arrays [15]. **Pandas** is an open-source library that offers data structures and analysis tools for Python. It is used, for example, to read CSV data, which is then stored in a DataFrame object, [16]. The most critical library used in this code is the **Chainladder-Python** package, which provides the essential actuarial tools required for the analysis and methodologies presented in this thesis, [3]. The information about the following definitions is from the corresponding documentation homepages: [15], [16], [3]

6.2.2 Definitions of important classes

In this subsection, the most important classes, objects and methods are described in order not to disturb the flow of reading.

Dataframe One of the most frequently used objects in Pandas is the dataframe. A dataframe is a two-dimensional object that is comparable to a spreadsheet or an SQL table. It is made up of rows and columns. A dataframe takes various objects as input, such as arrays or dataframes. In this thesis, however, the elements of the dataframe will only be strings, floats and integers.

Triangle The `Triangle` class is the core data structure used within the Chainladder package. Compared with the development triangle explained in subsection 2.1.2, the Triangle object is the core structure of the development like the development triangle is the core of the theory of the chain ladder method. Chainladder is designed to work with `Triangle` objects. The `Triangle` is a four-dimensional (4D) data structure with labelled axes: index, columns, origin, and development.

- **Index (Axis 0)**: The index represents the lowest level at which the triangle is managed. Examples include states, business branches, companies, or any other relevant classification.
- **Columns (Axis 1)**: Columns are used to store the various numeric values in the dataset, such as paid claims, incurred claims, or claim counts. These different metrics can all be represented as columns within the triangle.
- **Origin (Axis 2)**: The origin denotes the period from which the data in the columns originates. Examples of origin periods include accident months, report years, policy quarters, or other time-based intervals.
- **Development (Axis 3)**: Development represents the age or progression of the data over time. Typical choices for development periods include valuation months, valuation years, or valuation quarters.

The interaction with `Triangle` class works similar to patterns in `pandas` with DataFrames, for example the index and columns can be managed as they would be in a `pandas` DataFrame. Conceptually, the 4D structure can be seen as a collection of `pandas` DataFrames, where each cell (row, column) corresponds to an individual triangle slice. Additionally, while the `values` property of a `Triangle` can be used to obtain its NumPy representation, the `Triangle` class provides various helper methods to ensure that the shape of the NumPy representation remains consistent with other properties of the triangle.

Development Creating link ratios works with the `Development` class. The `Development` class, part of the Chainladder package, is a transformer that facilitates the selection of basic loss development patterns. It offers various parameters to customize the handling and transformation of data, making it a flexible tool for actuarial analysis in this thesis. Some key parameters include:

- `n_periods`: Specifies the number of periods to consider for the development analysis. Setting this parameter allows for control over how many periods are included when calculating development factors.
- `average`: Determines the method used to average development factors across periods. Common choices include 'volume,' which averages based on the size or volume of claims.
- `drop`, `drop_high`, `drop_low`: These parameters allow for the exclusion of specific periods or values that might skew the development pattern. For example, `drop_high` can be used to exclude unusually high values that are considered outliers.
- `preserve`: Controls how many periods are preserved in the development pattern, ensuring that key information is not lost during transformation.
- `fillna`: Provides a method to handle missing values (e.g., replacing NaN values), which is crucial when working with incomplete datasets.

This class implemented the various model variables since one can easily adapt the outliers and volume, which are key factors.

TailCurve The `TailCurve` class from the Chainladder package is used to extrapolate loss development factors (LDFs) beyond the observed data, forming a tail factor. This process helps in estimating the expected development of claims that may extend beyond the available data, which is crucial for accurately reserving long-tail insurance lines. The key parameters of the `TailCurve` class include:

- `curve`: Specifies the type of curve used for extrapolation, with the default being 'exponential.' Other curve types may be selected depending on the characteristics of the data and the desired extrapolation behaviour.
- `fit_period`: Defines the period over which the curve is fitted. Setting this parameter allows for control over which development periods are considered when determining the tail factor.

- `extrap_periods`: Indicates the number of periods to which the extrapolation is extended. By default, it is set to 100, enabling a long projection of the tail, but this can be adjusted as necessary.
- `attachment_age`: Specifies the development age at which the tail extrapolation should begin, providing flexibility in defining where the tail is considered to start.
- `reg_threshold`: Sets a threshold for regularization, controlling the conditions under which the extrapolated curve adjusts to avoid unrealistic projections.
- `projection_period`: Defines the period over which the tail factor is applied, useful for projecting claims beyond the last observed data point.

The `TailCurve` class aids in the estimation of tail factors by providing methods to extend development factors smoothly beyond the observed range. This class was used for the Tail analysis in the next chapter.

Chainladder The `Chainladder` class, a key component of the Chainladder package, implements the basic deterministic chain ladder method, which is widely used in actuarial practice for estimating future claims development based on historical data, like already described in 2.4. With this class a Chainladder model gets fitted, for example with the unchanged Triangle data or a developed Triangle, which was transformed with `Development`. The `Chainladder` class offers several important attributes to assist with this process:

- `X_`: Returns the original input data (`X`) used to fit the triangle. This represents the base data on which the development factors are calculated.
- `ultimate_`: Provides the ultimate losses estimated by the method. Ultimate losses are the total projected losses, combining both reported and future claims.
- `ibnr_`: Represents the incurred but not reported (IBNR) claims, calculated as the difference between the ultimate losses and the reported losses. This gives an estimate of claims that are expected but have not yet been recorded.
- `full_expectation_`: Back-fills the estimated ultimate losses to each development period within the original data, replacing known values. This allows for a complete view of the expected claims across all periods.
- `full_triangle_`: Similar to `full_expectation_`, but retains the known data, filling in only the periods where estimates are needed. This provides a comprehensive projection while preserving the original claims data.

The `Chainladder` class simplifies the process of applying the chain ladder method by managing these calculations and providing access to key outputs such as ultimate loss estimates and IBNR values.

Pipeline The `Pipeline` method from the `Chainladder` package is almost a copy from `scikit-learn` library. It is used to sequentially implement several transformers, like `Development` and `TailCurve`. `Pipeline` combines several transformers and can be put into the `Chainladder` method.

6.2.3 Modelling approach

The goal of the model is to find the best reserving model. An optimal model is selected based on the given paid and incurred data. A model is considered optimal if it has the lowest Claims Development Result (CDR) on average over the last years, as described in the algorithm presented in chapter 5. Since the original dataset only contained information on payments and reserves, an additional column for incurred loss was calculated using: Payment + Reserve. From this point onward, all analyses were performed using Python. The data were imported from a CSV file into a Pandas DataFrame, and the only data cleaning required was the removal of values outside the observation period.
Various analyses were then conducted for each sector and data column. Specifically, the occurrence of outliers was examined using the Interquartile Range (IQR) and the Reverse Nearest Neighbour (RNN) methods.
The function to determine the optimal model was implemented based on the algorithm described in 5. The key model parameters to be adjusted include the number of periods, outlier handling techniques, and the choice of tail factors. Additionally, the impact of adjusting the data for inflation was analyzed using consumption index data and a simple Python script. The study also shows how the CDR changes when the ultimate value is calculated using the pure payment or paid loss triangle and a combination of both.

6.2.4 Data Preparation

First, the data is imported from a CSV file and saved as a DataFrame. The CSV file has the separator ';' and decimal places are marked with ','. As the data already has the required format to create a triangle object, only simple data cleansing is required for data preparation. This is done with the function '`drop_faulty_years(df,YEARS)`', which eliminates data outside the observation period 2000-2022.

```
#read the data
df = pd.read_csv('data/data.csv', sep=";", decimal=',')

# Clean data
df = drop_faulty_years(df, YEARS)

# Create the triangle
triangle_df = create_triangle(df)
```

Listing 6.1: Preprocessing

After the preprocessing, a Triangle is created with the function '`create_triangle(df)`', which creates a triangle object for the given data. Since the data given is incremental data, we set `cumulative=False`.

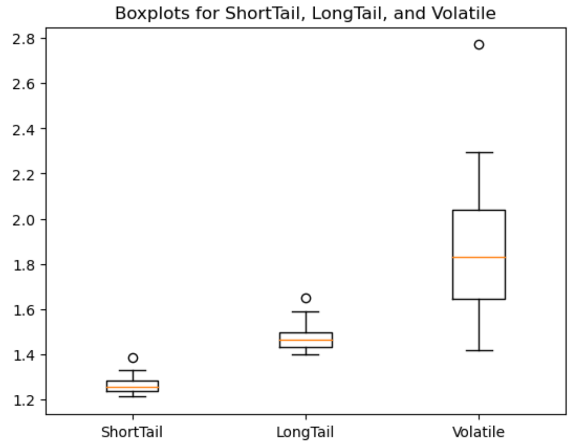


Figure 6.1: Boxplot Link Ratios

Visualisation of the dispersion of the link ratios in the first development period, for short-tail, long-tail and the volatile branch.

```
def create_triangle(df):
    """Create a ChainLadder triangle from the dataframe."""
    triangle_df = cl.Triangle(
        df,
        origin='Anfallsjahr',
        development='Abwicklungsjahr',
        columns=['zahlung_inc', 'reserve_inc', 'aufwand_inc','anzahl_inc'],
        index=['Segment'],
        cumulative=False,
    )
    return triangle_df
```

Listing 6.2: `create_triangle()`

6.2.5 Outlier Detection Using the Interquartile Range (IQR)

The outliers here refer to the outliers for the individual columns. The boxplots in Figure 6.1 visualise the distribution of the link ratios in the first year in the various divisions in the payment triangle, i.e. how the cumulative payments in the first year have developed in the second year.

Following the explanation in chapter 3.3, we perform an outlier detection, focusing on the link ratios of the different development periods. An example is given seen in figure 6.1, there are only upward outliers, characterised by dots, in this range in all sectors. The following algorithm identifies outliers in the claims data by applying the Interquartile Range (IQR) method to the link ratios of a development triangle. This approach is robust and does not assume a normal distribution of the data, making it suitable for claims reserving where the data might have skewed or heavy-tailed distributions.

Algorithm Implementation The outlier detection consists of two main functions: `calculate_bounds` and `find_iqr_outliers`. The process can be summarized as follows:

1. `calculate_bounds`: This function calculates the IQR-based lower and upper bounds for identifying outliers in the link ratios. The IQR is determined by subtracting the first quartile (25th percentile) from the third quartile (75th percentile). Using these quartiles, the lower and upper bounds are defined as:

$$\text{Lower Bound} = Q1 - 1.5 \times \text{IQR}, \quad \text{Upper Bound} = Q3 + 1.5 \times \text{IQR}$$

where $Q1$ and $Q3$ are the 25th and 75th percentiles, respectively.

```
def calculate_bounds(triangle):
    """Calculate the IQR and bounds for identifying outliers."""
    description_link_ratio = triangle.incr_to_cum().link_ratio.describe()
    IQR = description_link_ratio.loc['75%'] - description_link_ratio.loc['25%']
    lower_bound = description_link_ratio.loc['25%'] - 1.5 * IQR
    upper_bound = description_link_ratio.loc['75%'] + 1.5 * IQR
    return lower_bound, upper_bound
```

Listing 6.3: Function to calculate IQR bounds

2. `find_iqr_outliers`: This function identifies outliers based on the bounds calculated in `calculate_bounds`. It iterates over each column (each development period) of the link ratios and checks if the values fall outside the specified bounds. Values below the lower bound or above the upper bound are classified as outliers.

   The process begins by converting the triangle to cumulative values using `incr_to_cum()`, followed by extracting the link ratios. The function counts the non-null entries to ensure that only valid data points are checked for outliers. If a value is identified as an outlier, its index is recorded in a matrix (`bad_index`) for later use.

```
def find_iqr_outliers(triangle):
    """Identify IQR outliers for a given triangle."""
    lower_bound, upper_bound = calculate_bounds(triangle)
    bad_index = [[] for _ in range(22)]  # Create the matrix with the outlier entries

    for j in range(22):  # Number of columns
        values = triangle.incr_to_cum().link_ratio.values[0, 0, :, j]
        count_not_null = np.count_nonzero(~np.isnan(values))  # Count non-null
    entries

        for i in range(count_not_null):
            if values[i] < lower_bound[j] or values[i] > upper_bound[j]:
                bad_index[j].append(i)

    return transform_bad_index_to_iqr_outliers(bad_index)
```

Listing 6.4: Function to identify IQR outliers

Explanation:

- `calculate_bounds`: This function uses the `describe()` method to get statistical information about the link ratios. The IQR (Interquartile Range) is calculated as the difference between the 75th percentile (Q3) and the 25th percentile (Q1). The lower and upper bounds are then used to classify values that fall below or above as outliers.
- `find_iqr_outliers`: This function creates a matrix to store indices of outliers. It iterates over each development period (column), identifies the non-null values, and checks each value against the calculated IQR bounds. If a value is found to be outside the specified range, it is marked as an outlier.
- `transform_bad_index_to_iqr_outliers`: The indices of the detected outliers are converted into a format that can be used in the subsequent analysis or to exclude these values from further calculations.

6.2.6 Outlier Detection Using Reverse Nearest Neighbors (RNN)

In addition to the Interquartile Range (IQR) method, the practical analysis also incorporates the Reverse Nearest Neighbors (RNN) technique for detecting outliers, already introduced in section3.2 The following subsections describe the key functions used in the RNN-based outlier detection process.

Algorithm Implementation

1. `extract_link_ratios`: This function extracts the link ratios from the claims triangle data for a given line of business and column. It iterates over the available years, collects the link ratios, and stores them in an array while filtering out any NaN values.

```python
def extract_link_ratios(triangle_df, line_of_business, column):
    """Extract link ratios from the triangle data."""
    link_ratios = []
    for i in range(YEARS_MAX_INT-2000):
        dummy = np.squeeze(triangle_df.loc[line_of_business,
    column].incr_to_cum().link_ratio.iloc[0, 0, :, i].values)
        link_ratios.append(dummy[~np.isnan(dummy)])
    return link_ratios
```

Listing 6.5: Function to Extract Link Ratios

2. `find_outliers_with_rnn`: This is the core function that applies the Reverse Nearest Neighbors (RNN) method. It identifies outliers by calculating the distance between points and checking how many other points are within a specified neighbourhood. The parameters include:
   - **threshold**: The maximum number of Reverse Nearest Neighbors a point can have to be considered an outlier.

- **n_neighbors**: The number of neighbours to be considered when calculating distances.

The function starts by extracting link ratios, reshapes the data, and uses the `NearestNeighbors` model to find the nearest points. For each point, the RNN method counts how many times it appears as a neighbour of other points. If a point has fewer neighbors than the threshold, it is classified as an outlier.

```
def find_outliers_with_rnn(Triangle, line_of_business = LOB, column = COL,
    threshold=1, n_neighbors=10):
    """
    Identify outliers using Reverse Nearest Neighbors (RNN).
    """
    link_ratios = extract_link_ratios(Triangle, line_of_business, column)
    all_outliers = []
    year_index = YEARS_MAX_INT -2000  # Describes how big the triangle is

    for i in range(len(link_ratios) - 3):
        # Reshape the data to a 2D array
        test = link_ratios[i][:year_index - i].reshape(-1, 1)

        # Fit Nearest Neighbors model
        nbrs = NearestNeighbors(n_neighbors=n_neighbors, algorithm='auto').fit(test)
        distances, indices = nbrs.kneighbors(test)

        # Create a dictionary to store RNNs for each point
        rnn_dict = {i: [] for i in range(len(test))}

        # Loop over each point to identify RNNs
        for k in range(len(test)):
            for j in indices[k]:
                if k != j:
                    rnn_dict[j].append(k)

        # Identify outliers
        outliers = [key for key, value in rnn_dict.items() if len(value) <=
    threshold]
        all_outliers.append(outliers)

    return all_outliers
```

Listing 6.6: Function to Identify Outliers with RNN

Explanation:

- The RNN method is particularly useful when dealing with datasets where traditional outlier detection methods (like IQR) may not suffice, especially in cases of complex patterns. By using the RNN approach, this algorithm can identify points that are isolated, indicating potential outliers.
- The integration of functions like `create_index_matrix` and `translate_indices_to_years` ensures that the detected outliers can be systematically excluded from the Chain Ladder model, improving the accuracy of the reserving calculations.

This implementation of RNN-based outlier detection helps refine the analysis by identifying data points that might distort the accuracy of claims reserves. It allows the algorithm to exclude these outliers and focus on more consistent and reliable data points.

Alternative Outlier Detection and Diagonal Exclusion

In addition to the core function described earlier, slightly different approaches were used for handling other outlier detection methods and the exclusion of diagonals. These approaches are detailed below. For alternative outlier detection methods, such as the Reverse Nearest Neighbour (RNN) and Interquartile Range (IQR) models, a modified function was employed:

```
filtered_data = filter_data(index_matrix, current_year_index)

if filtered_data == []:
    filtered_data = None

current_triangle_dev = cl.Pipeline([("dev", cl.Development(drop=filtered_data,
                                   n_periods=periods_index, preserve=MINIMUM_PERIODS)),
                                   ("tail", cl.TailConstant(tail =
    1))]).fit_transform(current_triangle)
current_triangle_model = cl.Chainladder().fit(current_triangle_dev)
```

Listing 6.7: CDR with RNN/IQR

In this approach, the parameters `drop_high` and `drop_low` were not used. Instead, the `drop` parameter relies on a matrix that indicates the positions of outliers. Importantly, this method considers outliers across the entire column of the triangle, not just within the defined `n_periods`.

For the exclusion of diagonals, an additional argument was included:

```
current_year_triangle_dev = cl.Pipeline(
    [("development", cl.Development(drop_high=j[periods_bool_index][0],
                                    drop_low=j[periods_bool_index][1],
                                    n_periods=j[periods_bool_index][2],
                                    preserve=MINIMUM_PERIODS, average="volume")),
     ("tail", cl.TailConstant(tail = 1.0)]).fit_transform(current_year_triangle)
```

Listing 6.8: CDR with diagonals

Note: The functions `Pipeline`, `Development`, `TailCurve`, and `Chainladder` are pre-existing functionalities within the Chain Ladder package. These were utilized to streamline the process of developing and testing different models, enabling flexible adjustments for outlier handling and tail estimation. `Development` filters which data to include, i.e. drops the irrelevant link ratios. `TailCurve` develops the tail factor. `Pipeline` connects the different functions like `Development` and `TailCurve`. The output of a `Pipeline` function can be the input of `Chainladder`, which develops the chain ladder model. Instead of `Chainladder`, `MackChainladder`, `BornhuetterFerguson` or other models can be used to develop a model.

6.2.7 Implementation of the Optimization Algorithm

The following section describes the implementation of the algorithm outlined in chapter 5. This algorithm is at the core of the model comparison process, identifying the optimal model based on Claims Development Results (CDR). The function automates the calculation of ultimate values using a Chain Ladder model and compares different parameter setups by computing the CDR scores. The algorithm takes the data as input, along with information specifying the line of business (LOB) and whether the data pertains to payment or incurred loss triangles. It also receives a list defining which periods to include and whether to exclude maximum and/or minimum values. Additional optional parameters determine the size of the initial triangle and the extent of the observation period.

A key challenge is selecting an appropriate initial triangle. It must not be too small, as this could distort results due to poor model performance in the early years. At the same time, it should not be too large, as this would limit the number of comparative years. This balance is important because reserving behaviors can vary significantly across different years. To make an informed decision, several initial triangles were tested, and the CDRs were analyzed to observe how they changed. Based on these observations, an optimal initial triangle was selected.

The model comparison is performed by iterating over various observation periods. Starting with a smaller triangle, the algorithm calculates ultimate values based on the available data. It then progresses year by year, recalculating new ultimate values and comparing them to the previous year's results to derive the CDR. This process continues until the most recent year, after which an average CDR is computed for each model. The function outputs a list containing the average CDRs and the corresponding model parameters.

Step-by-Step Overview

- **Defining the Initial Triangle:**
  The process begins by defining the initial triangle, which specifies the starting years of data, the line of business, and whether paid loss or incurred loss data is being used. This setup remains the same across all models.

- **Processing Each Model:**
  For each model setup (defined by the parameters vector), the following steps are executed:
  - A model is created for the initial triangle. This involves specifying which data to include, determining which outliers to exclude, and setting the volume (i.e., how many periods to consider). Based on these inputs, a Chain Ladder model is generated for the smallest triangle.

- **Calculating the Average CDR:**
  To compute the average CDR, the algorithm proceeds step-by-step, starting from one year after the initial triangle and continuing to the end year. For each step, a new triangle object is created, a model is fitted, and the CDR score is calculated.

Calculation of CDR Scores

The CDR is calculated as the difference between the ultimate values from the previous year and the current year. To avoid errors, "NaN" values are replaced with zeros. Since these differences are handled as triangle objects, they are converted into regular ndarrays to allow for further computations.
The CDR score is then determined by applying a straightforward implementation of the formula previously described in 4.2.
It is important to note that the Chain Ladder package was used for model development, and the treatment of outliers is based on the observation period. For instance, if `n_periods=10`, the algorithm considers the highest and lowest values from the last 10 years, meaning only the middle 8 values are used in the calculation.

```python
def calculate_cdr_with_j(triangle_df, line_of_business, column, j, start_year =
    START_YEAR_STR, end_year = INITIAL_TRIANGLE_YEAR_STR):
    cdr_score_mean = []

    initial_triangle = triangle_df.loc[line_of_business, column][
        (triangle_df.valuation >= start_year) & (triangle_df.valuation < end_year)
    ]

    for periods_bool_index in range(len(j)):
        initial_triangle_dev = (cl.Pipeline([
                        ("dev", cl.Development(
                                drop_high = j[periods_bool_index][0],
                                drop_low = j[periods_bool_index][1],
                                n_periods = j[periods_bool_index][2],
                                preserve = MINIMUM_PERIODS,
                                average = "volume")),
                        ("tail", cl.TailConstant(tail = 1.0))
                        ]).fit_transform(initial_triangle))


        initial_triangle_model = cl.Chainladder().fit(initial_triangle_dev) #fit the model

        # Create a cdr_score for each triangle with more and more information
        cdr_score = []
        for current_year in range(int(end_year) + 1, YEARS_MAX_INT + 1):
            if current_year == int(end_year) + 1:
                last_year_triangle_model = initial_triangle_model
            else:
                last_year_triangle_model = current_year_triangle_model


            current_year_triangle = triangle_df.loc[line_of_business, column]
                                [(triangle_df.valuation >= start_year) &
                                (triangle_df.valuation < str(current_year))]

            current_year_triangle_dev = cl.Pipeline([
                                ("dev", cl.Development(
                                    drop_high = j[periods_bool_index][0],
                                    drop_low = j[periods_bool_index][1],
                                    n_periods = j[periods_bool_index][2],
                                    preserve = MINIMUM_PERIODS,)),
                                ("tail", cl.TailConstant(tail = 1.0))
                                ]).fit_transform(current_year_triangle)

            current_year_triangle_model = cl.Chainladder()
                                        .fit(current_year_triangle_dev)

        if cdr_score:
            cdr_score_mean.append([statistics.mean(cdr_score), j[periods_bool_index]])

    return cdr_score_mean
```

Listing 6.9: CDR-Score calculation

Verification of the Algorithm

To ensure the correctness of the implementation, the results were validated through spot checks using Excel, essentially calculated by hand. This approach helped to confirm the reliability and accuracy of the code.

6.2.8 Weighted Average paid loss loss and incurred loss

For the weighted ultimate, we use the same algorithm as for calculating the usual paid or incurred CDR scores. However, instead of calculating the CDR score with the simple paid or incurred ultimate, we take the weighted ultimate of the paid and incurred ultimate and compare this with the paid ultimate. We compared weight values w from 0 to 1, while 1 results in an incurred loss ultimate and 0 results in a paid loss ultimate.

$$\text{ultimate}_{\text{mixed}} = \text{w} \cdot \text{ultimate}_{\text{incurred}} + (1 - \text{w}) \cdot \text{ultimate}_{\text{paid}} \tag{6.1}$$

```
def weighted_zahlung_aufwand_triangle(triangle, line_of_business: str,  weight_zahlung =
    0.5):
    """This function returns a weighted Triangle"""
    triangle.loc[line_of_business, "anzahl_inc"] = (weight_zahlung *
                                    * triangle.loc[line_of_business, "zahlung_inc"]
                                    + (1 - weight_zahlung)
                                    * triangle.loc[line_of_business, "aufwand_inc"])

    return triangle.loc[line_of_business, "anzahl_inc"]
```

Listing 6.10: Weighted Triangle

6.2.9 Inflation Adjustment

The inflation is calculated according the calendar year with the function ‘`inflation_adjustment`´. The inflation factor is the quotient of the Consumer Price Index (CPI) of the reference year and the current calendar year. The CPI data originates from [20]. The inflation adjustment is done by multiplying the inflation factor with the data.

```python
def inflation_adjustment(df, reference_year=2022):
    df_adjusted = df.copy()

    if reference_year in CPI_DICT_2000:
        cpi_reference = CPI_DICT_2000[reference_year]

    development_year = df_adjusted['Abwicklungsjahr']
    if development_year in CPI_DICT_2000:
        cpi_development = CPI_DICT_2000[development_year]
        inflation_factor = cpi_reference / cpi_development

        # Bereinige die Betraege
        df_adjusted['zahlung_inc'] *= inflation_factor
        df_adjusted['reserve_inc'] *= inflation_factor
        df_adjusted['aufwand_inc'] *= inflation_factor
    return df_adjusted
```

Listing 6.11: Inflation Adjustment

6.2.10 MSE Implementation

The difference between the code for the MSE implementation and the CDR implementation is, that we did not calculate a score, but the total mack standard error and used `MackChainladder` instead of `Chainladder` since the function `total_mack_std_err_` is already implemented in the `ChainLadder`-package.

6.2.11 Tail-Ajustment

The tail adjustment was done by the `TailCurve` method. The CDR calculation works the same, but instead of `("tail", cl.TailConstant(tail = 1))` we use `("tail", cl.TailCurve(tail = "exponential"))` or `("tail", cl.TailCurve(tail = "inverse_power'"))` The TailCurve uses a curve fitting.

```python
current_year_triangle_dev = cl.Pipeline([
                    ("dev", cl.Development(
                        drop_high = j[periods_bool_index][0],
                        drop_low = j[periods_bool_index][1],
                        n_periods = j[periods_bool_index][2],
                        preserve = MINIMUM_PERIODS,)),
                    ("tail", cl.TailCurve(tail = "exponential"))
                    ]).fit_transform(current_year_triangle)
```

7 Results and Discussion

In this chapter, the results of the implemented optimization algorithm are analyzed. The aim is to evaluate the effects of different model configurations on the Claims Development Result (CDR) and to determine the optimal model settings for accurate claims reserving. The analysis focuses on various factors, such as the number of periods, outlier handling, tail factors, and adjustments for inflation.

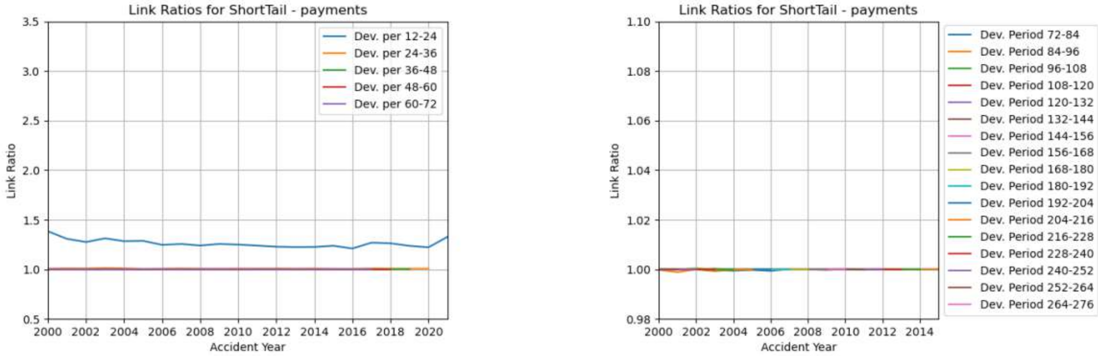
	12-24	24-36	36-48	48-60	60-72	72-84	84-96	96-108	108-120	120-132	132-144	144-156	156-168	168-180	180-192	192-204	204-216	216-228	228-240	240-252	252-264	264-276
2000	1.3839	1.0079	1.0021	1.0008	1.0005	1.0001	0.9998	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2001	1.3079	1.0103	0.9999	0.9999	1.0000	1.0000	0.9989	0.9999	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
2002	1.2759	1.0091	1.0021	1.0001	0.9994	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0003	1.0005	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000		
2003	1.3130	1.0140	1.0019	0.9995	0.9990	0.9997	0.9993	0.9997	1.0003	1.0000	1.0000	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000			
2004	1.2842	1.0110	1.0001	1.0006	0.9998	0.9994	0.9998	0.9998	1.0000	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000				
2005	1.2875	1.0047	1.0004	1.0007	1.0007	0.9998	0.9998	1.0000	0.9999	1.0001	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000					
2006	1.2471	1.0070	1.0024	1.0000	0.9994	0.9994	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000						
2007	1.2563	1.0095	1.0023	1.0006	0.9997	1.0001	1.0000	1.0000	1.0000	1.0001	1.0000	1.0000	1.0000	0.9999	1.0000							
2008	1.2403	1.0072	1.0033	1.0002	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000								
2009	1.2563	1.0067	1.0011	1.0008	1.0002	1.0000	1.0001	1.0000	0.9998	1.0001	1.0000	1.0000	1.0000									
2010	1.2498	1.0081	1.0004	0.9999	1.0006	1.0001	0.9999	1.0000	1.0000	0.9999	1.0000	1.0000										
2011	1.2395	1.0075	1.0045	1.0012	0.9998	1.0000	0.9998	1.0000	0.9999	1.0000	1.0000											
2012	1.2287	1.0088	1.0030	1.0009	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000												
2013	1.2254	1.0072	1.0017	1.0008	1.0000	1.0000	1.0000	1.0000	1.0000													
2014	1.2263	1.0086	1.0011	1.0016	1.0004	1.0001	1.0000	1.0000														
2015	1.2381	1.0063	1.0020	1.0003	1.0000	1.0001	1.0000															
2016	1.2108	1.0053	1.0024	1.0001	0.9978	1.0000																
2017	1.2696	1.0093	1.0018	1.0000	1.0003																	
2018	1.2631	1.0077	1.0017	1.0004																		
2019	1.2371	1.0075	1.0025																			
2020	1.2228	1.0079																				
2021	1.3290																					

Figure 7.1: Link Ratios and Heatmap for Short-Tail Paid Loss

This figure shows the triangle data for paid losses in the form of a heatmap.

7.1 Link Ratio Analysis

This section visualises the link ratios using line plots and heat maps in figures 7.1 to 7.11. The link ratio represents the change in cumulative paid or incurred losses from one development year to the next for each accident year, calculated based on the development triangle.



(a) Link Ratios paid loss - first 5 Link Ratios

(b) Link Ratios paid loss - other Link Ratios

Figure 7.2: Link Ratios for Short-Tail Paid Loss

(a) shows the link ratios of the paid loss triangle for the first five development periods, plotted as a function of the accident year. (b) shows the link ratios of the paid loss triangle for the remaining development periods, plotted as a function of the accident year.

The line plots display link ratios for each development period. In these plots, the x-axis shows the accident year (denoted as "accident year minus 2000"), and the y-axis represents the link ratios. A link ratio of 1 indicates no change in cumulative losses from one period to the next, while ratios above 1 indicate an increase in cumulative values. Ratios below 1 may signify a decrease, which can result from adjustments such as overestimated reserves or correction of payments. In addition to these plots, we include heatmaps to further illustrate the variability in link ratios across accident and development years. The heatmaps visually represent the development triangle, with link ratios as values, and use a color gradient ranging from dark red to dark blue. Darker shades represent values farther from the mean: red indicates higher values, while blue denotes lower values. This coloring highlights deviations and patterns across the triangle, providing insight into the volatility of link ratios within each development period.

ShortTail

The link ratios exhibit a high level of consistency, with minimal fluctuation observed throughout the entire period. They range from approximately 1.2-1.5. Volatility is predominantly confined to the beginning and end of the observation period. The elevated values at the start of the period could be attributed to differences in claims settlement practices at that time, while the spike towards the end might be explained by the impact of high inflation. The relatively stable lines further suggest that losses in this branch occur consistently over the years, without being heavily influenced by extreme events. Notably, values significantly above 1 are only observed in the development period from 12 to 24 months, indicating that the majority of payments were completed within the first two years. A reason could be the slower settlement back then. The first development factor is the factor from the amount of the first year to the second year. If more money gets paid in the first year because settlement is so quick, the link ratios of the first development year will be smaller.

	12-24	24-36	36-48	48-60	60-72	72-84	84-96	96-108	108-120	120-132	132-144
2000	1.0650	0.9869	0.9975	1.0004	1.0015	1.0003	0.9999	1.0000	1.0000	1.0000	1.0001
2001	1.0716	0.9958	0.9988	0.9988	1.0001	1.0002	0.9991	0.9999	1.0001	1.0004	1.0000
2002	1.0213	0.9968	1.0013	1.0003	0.9973	0.9958	1.0000	1.0000	1.0010	1.0000	1.0002
2003	1.0386	0.9923	1.0028	1.0009	1.0000	1.0002	1.0000	1.0003	1.0002	1.0000	1.0000
2004	1.0348	1.0040	0.9995	1.0002	1.0001	1.0000	1.0017	1.0000	1.0007	1.0001	0.9999
2005	1.0497	0.9956	1.0042	1.0001	0.9998	1.0013	0.9997	1.0010	1.0000	1.0001	0.9999
2006	1.0005	1.0033	0.9995	1.0003	1.0014	0.9997	1.0008	1.0000	1.0009	0.9991	1.0000
2007	1.0264	1.0000	1.0012	1.0020	1.0004	1.0008	1.0001	1.0010	0.9992	1.0000	1.0000
2008	1.0345	1.0023	1.0019	1.0017	1.0007	1.0004	1.0000	1.0000	1.0000	1.0000	1.0000
2009	1.0455	1.0035	1.0011	1.0020	1.0002	1.0000	1.0001	0.9999	0.9997	1.0000	1.0000
2010	1.0575	1.0028	1.0016	1.0004	1.0007	1.0000	0.9998	1.0000	1.0001	0.9999	1.0000
2011	1.0569	1.0026	1.0036	1.0003	1.0001	0.9996	0.9998	0.9999	1.0000	1.0000	1.0000
2012	1.0380	1.0052	1.0021	1.0005	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
2013	1.0399	1.0027	1.0013	1.0001	1.0000	1.0000	0.9999	1.0001	0.9999		
2014	1.0542	1.0009	0.9998	1.0012	1.0003	1.0002	0.9999	1.0000			
2015	1.0404	1.0007	1.0017	1.0000	1.0001	0.9999	1.0001				
2016	1.0224	1.0006	1.0010	1.0002	1.0006	0.9977					
2017	1.0208	1.0007	1.0015	0.9999	1.0002						
2018	1.0436	1.0029	1.0011	0.9997							
2019	1.0330	1.0015	1.0011								
2020	1.0508	1.0027									
2021	1.0680										

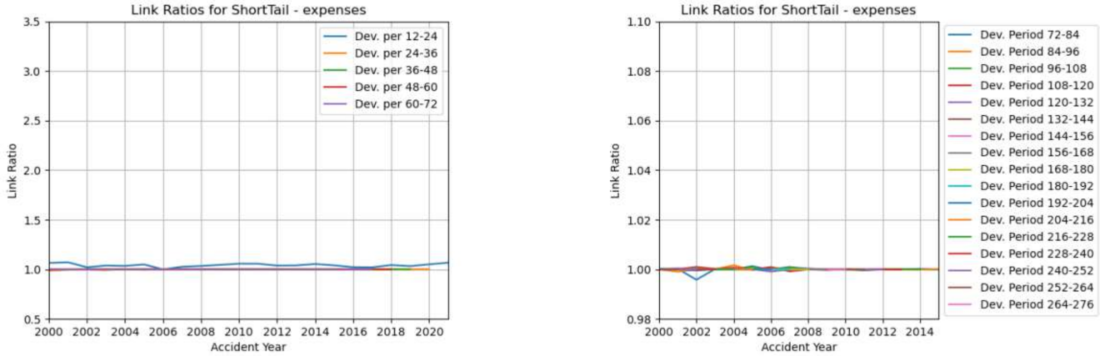
	144-156	156-168	168-180	180-192	192-204	204-216	216-228	228-240	240-252	252-264	264-276
2000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2001	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
2002	1.0002	1.0007	1.0000	1.0000	0.9996	1.0000	1.0000	1.0000	1.0000		
2003	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000			
2004	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000				
2005	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000					
2006	1.0000	1.0000	1.0000	1.0000	1.0000						
2007	1.0000	1.0000	0.9999	1.0000							
2008	1.0000	1.0000	1.0000								
2009	1.0000	1.0000									
2010	1.0000										

Figure 7.3: Heatmap for ShortTail incurred loss

This figure is structured the same way as in figure 7.1.

Examining the incurred loss, it becomes evident that they are even smoother than the paid loss. The values are between 1.02-1.07. Since incurred loss comprise both paid loss and reserves, their smoothness and proximity to 1 imply that claims were settled efficiently, with minimal need for large reserves.

Overall, the findings from the link ratios support the classification of this branch as fast-carrying and show expected patterns seen in short-tail insurance such as motor own damage and property damage coverage. This analysis clarifies the claims behaviour for the examined branch and provides a framework for understanding similar branches in the industry.



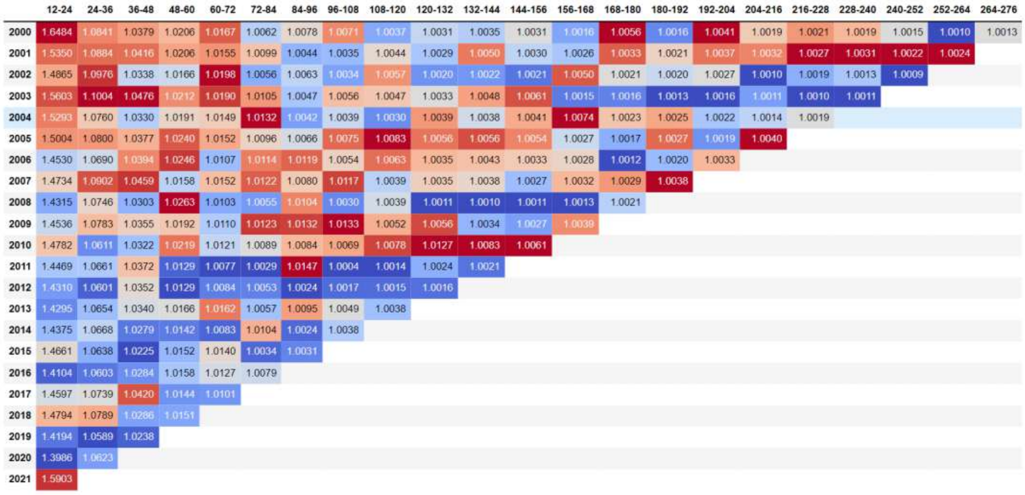
(a) Link Ratios incurred loss - first 4 Link Ratios (b) Link Ratios incurred loss - other Link Ratios

Figure 7.4: Link Ratios for ShortTail incurred loss

(a), (b) are structured the same way as in figure 7.2.

Long Tail

The link ratios for long-tail policies exhibit a degree of consistency, they only vary from 1.41-1.65; however, unlike the fast-settling branches, the ratios do not converge to 1 after the initial development periods. Instead, they remain slightly above 1 even towards the end of the development period, indicating that the settlement of claims continues over a prolonged timeframe. Similar to the fast-settling branch, it can be observed that the link ratios are higher during the initial and final years of development. One possible explanation for the elevated values in the early years post-2000 is the difference in settlement processes, which were less digitalized compared to today's standards. The high values towards the end of the period may be attributed to significant inflation rates. In the years between, the ratios demonstrate relative stability, although closer inspection reveals variations within a narrow range of approximately 0.02.
When examining the link ratios for incurred loss, at first glance a calmer pattern emerges. Although the values are relatively stable in the first development periods, they exhibit more volatility than in fast-settling segments. Notably, there are no significant outliers on initial inspection. However, it is evident that the incurred loss link ratios predominantly fall below 1 in the later development periods, indicating a decline. This trend can likely be attributed to the conservative nature of reserve estimates in long-tail segments, where overestimation leads to gradual adjustments as claims mature.

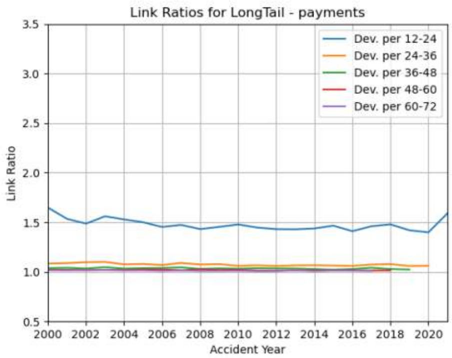


	12-24	24-36	36-48	48-60	60-72	72-84	84-96	96-108	108-120	120-132	132-144
2000	1.6484	1.0841	1.0379	1.0206	1.0167	1.0062	1.0078	1.0071	1.0037	1.0031	1.0035
2001	1.5350	1.0884	1.0416	1.0206	1.0155	1.0099	1.0044	1.0035	1.0044	1.0029	1.0050
2002	1.4865	1.0976	1.0338	1.0166	1.0198	1.0056	1.0063	1.0034	1.0057	1.0020	1.0022
2003	1.5603	1.1004	1.0476	1.0212	1.0190	1.0105	1.0047	1.0056	1.0047	1.0033	1.0048
2004	1.5293	1.0760	1.0330	1.0191	1.0149	1.0132	1.0042	1.0039	1.0030	1.0039	1.0038
2005	1.5004	1.0800	1.0377	1.0240	1.0152	1.0096	1.0066	1.0075	1.0083	1.0056	1.0056
2006	1.4530	1.0690	1.0394	1.0246	1.0107	1.0114	1.0119	1.0054	1.0063	1.0035	1.0043
2007	1.4734	1.0902	1.0459	1.0158	1.0152	1.0122	1.0080	1.0117	1.0039	1.0035	1.0038
2008	1.4315	1.0746	1.0303	1.0263	1.0103	1.0055	1.0104	1.0030	1.0039	1.0011	1.0010
2009	1.4536	1.0783	1.0355	1.0192	1.0110	1.0123	1.0132	1.0133	1.0052	1.0056	1.0034
2010	1.4782	1.0611	1.0322	1.0219	1.0121	1.0089	1.0084	1.0069	1.0078	1.0127	1.0083
2011	1.4469	1.0661	1.0372	1.0129	1.0077	1.0029	1.0147	1.0004	1.0014	1.0024	1.0021
2012	1.4310	1.0601	1.0352	1.0129	1.0084	1.0053	1.0024	1.0017	1.0015	1.0016	
2013	1.4295	1.0654	1.0340	1.0166	1.0162	1.0057	1.0095	1.0049	1.0038		
2014	1.4375	1.0668	1.0279	1.0142	1.0083	1.0104	1.0024	1.0038			
2015	1.4661	1.0638	1.0225	1.0152	1.0140	1.0034	1.0031				
2016	1.4104	1.0603	1.0284	1.0158	1.0127	1.0079					
2017	1.4597	1.0739	1.0420	1.0144	1.0101						
2018	1.4794	1.0789	1.0286	1.0151							
2019	1.4194	1.0589	1.0238								
2020	1.3986	1.0623									
2021	1.5903										

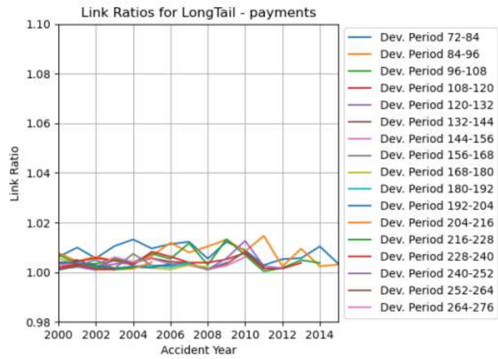
	144-156	156-168	168-180	180-192	192-204	204-216	216-228	228-240	240-252	252-264	264-276
2000	1.0031	1.0016	1.0056	1.0016	1.0041	1.0019	1.0021	1.0019	1.0015	1.0010	1.0013
2001	1.0030	1.0026	1.0033	1.0021	1.0037	1.0032	1.0027	1.0031	1.0022	1.0024	
2002	1.0021	1.0050	1.0021	1.0020	1.0027	1.0010	1.0019	1.0013	1.0009		
2003	1.0061	1.0015	1.0016	1.0013	1.0016	1.0011	1.0010	1.0011			
2004	1.0041	1.0074	1.0023	1.0025	1.0022	1.0014	1.0019				
2005	1.0054	1.0027	1.0017	1.0027	1.0019	1.0040					
2006	1.0033	1.0028	1.0012	1.0020	1.0033						
2007	1.0027	1.0032	1.0029	1.0038							
2008	1.0011	1.0013	1.0021								
2009	1.0027	1.0039									
2010	1.0061										

Figure 7.5: Heatmap for LongTail paid loss

(c) is structured the same way as in figure 7.1.



(a) Link Ratios paid loss - first 4 Link Ratios



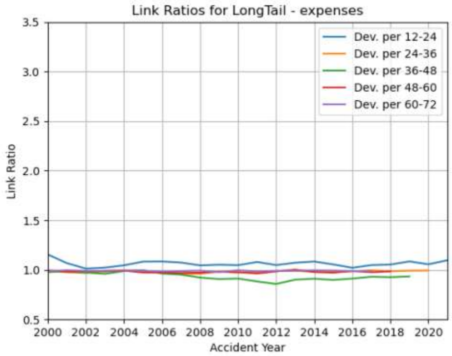
(b) Link Ratios paid loss - other Link Ratios

Figure 7.6: Link Ratios for LongTail paid loss

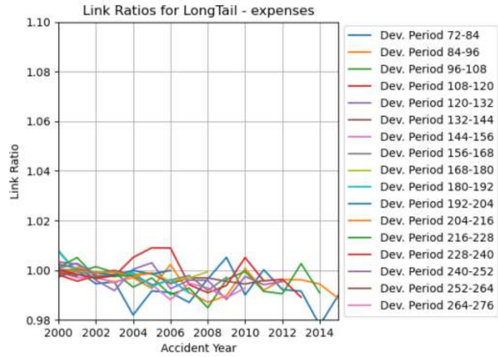
(a) and (b) are structured the same way as in figure 7.2.

	12-24	24-36	36-48	48-60	60-72	72-84	84-96	96-108	108-120	120-132	132-144	144-156	156-168	168-180	180-192	192-204	204-216	216-228	228-240	240-252	252-264	264-276
2000	1.1545	0.9900	0.9772	0.9954	0.9869	0.9973	0.9981	1.0079	0.9982	1.0034	1.0025	0.9992	1.0003	0.9992	1.0075	0.9998	0.9992	1.0011	1.0001	1.0008	0.9991	0.9994
2001	1.0677	0.9758	0.9866	0.9819	0.9966	0.9998	1.0001	0.9991	0.9956	1.0025	1.0008	0.9970	1.0003	1.0012	0.9992	0.9997	0.9997	1.0051	0.9984	1.0028	0.9977	
2002	1.0127	0.9688	0.9727	0.9876	0.9902	0.9946	0.9970	1.0015	0.9977	0.9960	0.9987	0.9973	0.9966	0.9993	0.9990	0.9991	0.9991	0.9971	0.9967	0.9981		
2003	1.0231	0.9925	0.9604	0.9840	0.9899	0.9953	0.9954	0.9991	0.9977	0.9918	1.0000	0.9948	0.9984	0.9991	0.9974	0.9980	0.9993	0.9976	0.9979			
2004	1.0461	0.9959	0.9882	0.9920	0.9932	0.9820	0.9970	0.9932	1.0052	1.0004	0.9979	0.9983	0.9976	0.9985	0.9991	0.9999	0.9975	0.9980				
2005	1.0835	0.9934	0.9959	0.9733	0.9913	0.9917	0.9927	0.9969	1.0091	1.0030	0.9987	0.9943	0.9941	0.9989	0.9935	0.9985	0.9990					
2006	1.0849	0.9723	0.9629	0.9736	0.9845	0.9910	1.0023	0.9903	1.0090	0.9925	0.9947	0.9881	0.9961	0.9957	0.9962	0.9999						
2007	1.0740	0.9610	0.9521	0.9695	0.9880	0.9871	0.9910	0.9929	0.9942	0.9960	0.9969	0.9948	0.9979	0.9969	0.9908							
2008	1.0453	0.9608	0.9223	0.9724	0.9928	0.9965	0.9872	0.9849	0.9910	0.9960	0.9969	0.9928	0.9919	0.9994								
2009	1.0521	0.9868	0.9073	0.9804	0.9777	1.0052	0.9897	0.9963	0.9939	0.9882	0.9855	0.9889	0.9972									
2010	1.0476	0.9719	0.9126	0.9771	0.9963	0.9901	1.0007	0.9996	1.0051	0.9975	0.9945	0.9929										
2011	1.0801	0.9809	0.8830	0.9637	0.9849	1.0002	0.9920	0.9915	0.9957	0.9942	0.9959											
2012	1.0484	0.9882	0.8582	0.9848	0.9905	0.9922	0.9962	0.9906	0.9964	0.9954												
2013	1.0724	0.9902	0.9016	1.0029	0.9922	0.9916	0.9962	1.0026	0.9892													
2014	1.0844	0.9810	0.9109	0.9780	0.9956	0.9779	0.9944	0.9909														
2015	1.0549	0.9804	0.8986	0.9717	0.9933	0.9900	0.9887															
2016	1.0214	0.9848	0.9123	0.9891	0.9878	0.9952																
2017	1.0482	0.9955	0.9308	0.9764	0.9899																	
2018	1.0541	0.9860	0.9264	0.9849																		
2019	1.0855	0.9920	0.9336																			
2020	1.0566	0.9938																				
2021	1.0970																					

Figure 7.7: Heatmap for LongTail incurred loss
(c) is structured the same way as in figure 7.1.



(a) Link Ratios incurred loss - first 4 Link Ratios



(b) Link Ratios incurred loss - other Link Ratios

Figure 7.8: Link Ratios and Heatmap for LongTail incurred loss
(a) and (b) are structured the same way as in figure 7.2.

Volatile

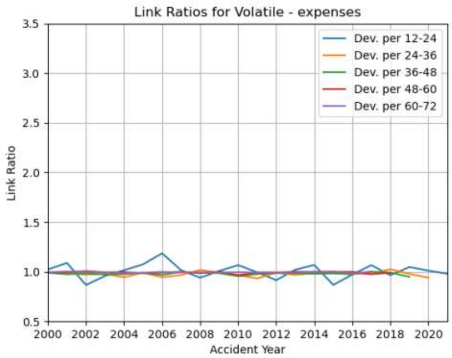
As observed in other segments, the highest volatility is found at the beginning of the settlement process, in terms of paid loss and incurred loss, The link ratios are between 1.4-2.8, which means they are almost double in some periods. paid loss exhibit significant instability, with pronounced outliers, such as in 2006 in the development period 12-24. This reflects the initial uncertainty when large claims are first reported, as their full impact might not yet be known, leading to significant fluctuations. Unlike other segments, relatively high spikes are still visible in the later development periods. This indicates that the settlement process remains relatively uncertain even in the later months, potentially due to claims that evolve over time, such as long legal disputes or new information that emerges about existing claims, prompting adjustments in reserves. The volatile segment does not settle quickly as the fast-carrying branch.

	12-24	24-36	36-48	48-60	60-72	72-84	84-96	96-108	108-120	120-132	132-144	144-156	156-168	168-180	180-192	192-204	204-216	216-228	228-240	240-252	252-264	264-276
2000	1.9849	1.0412	1.0243	1.0020	1.0020	1.0092	1.0001	1.0019	0.9997	1.0000	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2001	1.8107	1.1500	1.0253	1.0065	1.0036	1.0049	1.0010	1.0016	1.0598	1.0006	1.0063	1.0103	1.0006	1.0003	1.0003	1.0003	1.0003	1.0000	1.0000	1.0000	1.0000	
2002	1.4196	1.0067	1.0081	1.0103	1.0191	1.0043	0.9987	1.0006	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000		
2003	1.6700	1.0330	1.0073	1.0098	1.0015	1.0005	1.0005	1.0001	1.0003	0.9936	1.0002	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000			
2004	1.7512	1.0988	1.0153	1.0055	0.9898	1.0015	1.0000	1.0002	0.9954	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000				
2005	2.2947	1.0793	1.0232	1.0043	0.9952	1.0000	1.0004	1.0000	1.0000	1.0001	1.0323	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000					
2006	2.7710	1.1296	1.0011	1.0016	1.0197	1.0012	1.0037	1.0004	1.0000	1.0001	1.0003	1.0004	1.0004	1.0031	1.0004	1.0011						
2007	1.6180	1.0678	1.0176	1.0684	1.0039	0.9950	0.9999	1.0044	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000							
2008	1.7106	1.0633	1.0253	1.0033	0.9985	1.0000	1.0000	1.0000	1.0000	1.0006	1.0000	1.0000	1.0000	1.0000								
2009	2.0601	1.0302	1.0450	1.0044	1.0021	1.0024	1.0005	1.0000	1.0001	1.0000	1.0000	1.0000	1.0000									
2010	1.6583	1.1067	1.0226	1.0033	1.0009	1.0010	1.0006	0.9995	0.9995	1.0000	1.0000	1.0000										
2011	1.8471	1.0446	1.0153	1.0049	0.9990	1.0034	1.0000	1.0001	1.0000	1.0000	1.0001											
2012	1.5300	1.0350	1.0057	1.0054	1.0033	1.0004	1.0001	1.0000	1.0000	1.0000												
2013	1.6322	1.0445	1.0131	1.0030	1.0013	1.0000	1.0010	1.0000	1.0003													
2014	2.0865	1.0887	1.0062	1.0017	1.0002	1.0001	1.0001	1.0001														
2015	2.0955	1.0800	1.0239	1.0073	1.0004	0.9998	1.0004															
2016	1.6392	1.0392	1.0087	1.0065	1.0155	1.0000																
2017	1.8510	1.0672	1.0214	1.0036	1.0021																	
2018	2.2168	1.0868	1.0707	1.0094																		
2019	1.8699	1.0984	1.0539																			
2020	1.5616	1.0914																				
2021	1.8650																					

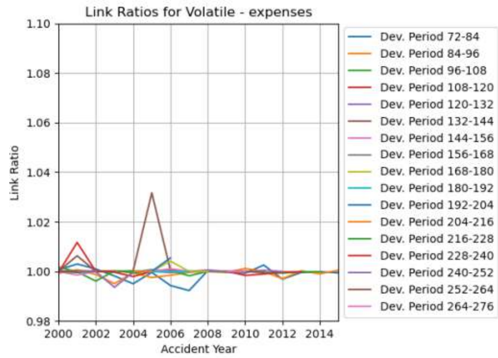
Figure 7.9: Heatmap for Volatile paid loss

(c) is structured the same way as in figure 7.1.

However, when looking at the developments over time, it tends to approach a value of 1 more closely than the long-tail segment. By development period 108-120, almost all link ratios are approximately 1, while the triangle of the long-tailed branch consistently shows link ratios bigger than 1 for all development periods, including the last one available. The stabilisation in the first development periods suggests that, despite initial instability, claims in the volatile segment eventually reach a more predictable settlement pattern, though this process takes longer and is less linear compared to shorter-tail segments.



(a) Link Ratios paid loss - first 4 Link Ratios



(b) Link Ratios paid loss - other Link Ratios

Figure 7.10: Link Ratios for Volatile paid loss

(a) and (b) are structured the same way as in figure 7.2.

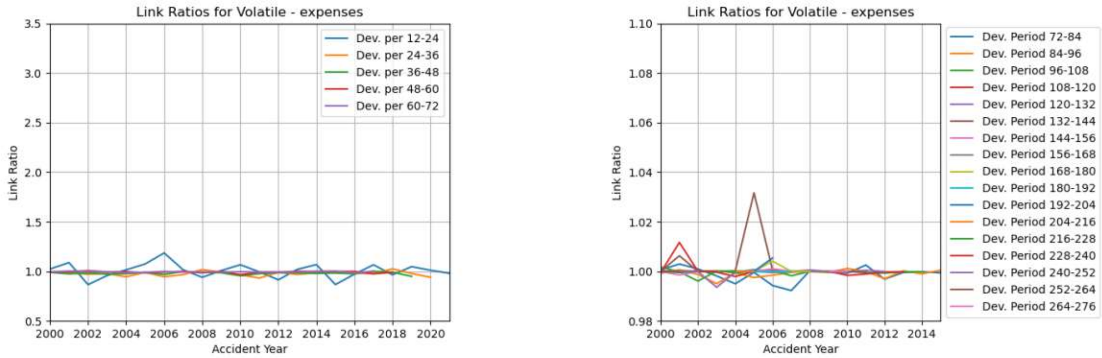
Another important aspect to consider is the unpredictability of event-driven claims in this segment. Natural disasters, major accidents, or other significant events can lead to sudden spikes, which may not be immediately apparent in other, more stable segments. For instance, the development spikes seen in 2008 could have been driven by such an event, emphasizing the need for flexibility and preparedness in reserving for volatile lines of business.

	12-24	24-36	36-48	48-60	60-72	72-84	84-96	96-108	108-120	120-132	132-144
2000	1.6484	1.0841	1.0379	1.0206	1.0167	1.0062	1.0078	1.0071	1.0037	1.0031	1.0035
2001	1.5350	1.0884	1.0416	1.0206	1.0155	1.0099	1.0044	1.0035	1.0044	1.0029	1.0050
2002	1.4865	1.0976	1.0338	1.0166	1.0198	1.0056	1.0063	1.0034	1.0057	1.0020	1.0022
2003	1.5603	1.1004	1.0476	1.0212	1.0190	1.0105	1.0047	1.0056	1.0047	1.0033	1.0048
2004	1.5293	1.0760	1.0330	1.0191	1.0149	1.0132	1.0042	1.0039	1.0030	1.0039	1.0038
2005	1.5004	1.0800	1.0377	1.0240	1.0152	1.0096	1.0066	1.0075	1.0083	1.0056	1.0056
2006	1.4530	1.0690	1.0394	1.0246	1.0107	1.0114	1.0119	1.0054	1.0063	1.0035	1.0043
2007	1.4734	1.0902	1.0459	1.0158	1.0152	1.0122	1.0080	1.0117	1.0039	1.0035	1.0038
2008	1.4315	1.0746	1.0303	1.0263	1.0103	1.0055	1.0104	1.0030	1.0039	1.0011	1.0010
2009	1.4536	1.0783	1.0355	1.0192	1.0110	1.0123	1.0132	1.0133	1.0052	1.0056	1.0034
2010	1.4782	1.0611	1.0322	1.0219	1.0121	1.0089	1.0084	1.0069	1.0078	1.0127	1.0083
2011	1.4469	1.0661	1.0372	1.0129	1.0077	1.0029	1.0147	1.0004	1.0014	1.0024	1.0021
2012	1.4310	1.0601	1.0352	1.0129	1.0084	1.0053	1.0024	1.0017	1.0015	1.0016	
2013	1.4295	1.0654	1.0340	1.0166	1.0162	1.0057	1.0095	1.0049	1.0038		
2014	1.4375	1.0668	1.0279	1.0142	1.0083	1.0104	1.0024	1.0038			
2015	1.4661	1.0638	1.0225	1.0152	1.0140	1.0034	1.0031				
2016	1.4104	1.0603	1.0284	1.0158	1.0127	1.0079					
2017	1.4597	1.0739	1.0420	1.0144	1.0101						
2018	1.4794	1.0789	1.0286	1.0151							
2019	1.4194	1.0589	1.0238								
2020	1.3986	1.0623									
2021	1.5903										

	144-156	156-168	168-180	180-192	192-204	204-216	216-228	228-240	240-252	252-264	264-276
2000	1.0031	1.0016	1.0056	1.0016	1.0041	1.0019	1.0021	1.0019	1.0015	1.0010	1.0013
2001	1.0030	1.0026	1.0033	1.0021	1.0037	1.0032	1.0027	1.0031	1.0022	1.0024	
2002	1.0021	1.0050	1.0021	1.0020	1.0027	1.0010	1.0019	1.0013	1.0009		
2003	1.0061	1.0015	1.0016	1.0013	1.0016	1.0011	1.0010	1.0011			
2004	1.0041	1.0074	1.0023	1.0025	1.0022	1.0014	1.0019				
2005	1.0054	1.0027	1.0017	1.0027	1.0019	1.0040					
2006	1.0033	1.0028	1.0012	1.0020	1.0033						
2007	1.0027	1.0032	1.0029	1.0038							
2008	1.0011	1.0013	1.0021								
2009	1.0027	1.0039									
2010	1.0061										
2011											
2012											
2013											
2014											
2015											
2016											
2017											
2018											
2019											
2020											
2021											

Figure 7.11: Heatmap for Volatile incurred loss

(c) is structured the same way as in figure 7.1.



(a) Link Ratios incurred loss - first 4 Link Ratios (b) Link Ratios incurred loss - other Link Ratios

Figure 7.12: Link Ratios for Volatile incurred loss

(a) and (b) are structured the same way as in figure 7.2.

Another noteworthy observation is the behavior of incurred loss in the volatile segment. Unlike other segments, incurred loss here show a distinctly jagged pattern, which is not as evident in the short-tail or long-tail segments. In short-tail lines, incurred loss tend to be smoother and lower, reflecting more predictable and consistent handling costs for smaller, frequent claims. On the other hand, long-tail segments show a smoother but generally higher level of incurred loss, as these claims often involve prolonged settlement processes that accumulate handling costs over time.

In conclusion, managing reserves for volatile segments requires more than just standard models; it involves constantly adapting to new information and considering potential changes long after the initial claim event. This makes predictive modeling more complex and highlights the importance of using robust models that can handle such uncertainties effectively.

7.2 Comparison of Approaches

In this section, we evaluate the impact of various parameters and adjustments on improving the CDR-score. The following factors are analyzed in detail:

- **Number of periods considered**
- **Outlier detection methods:**
  - Exclusion of maximum and minimum values
  - Reverse Nearest Neighbor (RNN) outlier detection
  - Interquartile Range (IQR) outlier detection
  - Removal of suspicious diagonals
- **Tail adjustment**
- **Weighted average of paid loss and incurred loss**
- **Inflation-adjusted data**

This comparison aims to determine which combinations of parameters and methods yield the most accurate reserving results.

7.2.1 Impact of Removing Outliers

This section examines three branches, each analyzed separately for incurred and paid losses. The models in this section differ in their outlier detection method, each represented by a different colour:

- **RNN** (violet line): The RNN-outlier are excluded.
- **IQR** (cyan line): The IQR-outlier are excluded.
- [**True, True**] (blue line): Both the highest and lowest values are excluded.
- [**True, False**] (orange line): Only the highest value is excluded.
- [**False, True**] (green line): Only the lowest value is excluded.
- [**False, False**] (red line): No values are excluded.

Note that the highest and lowest values, RNN-outliers and IQR-outliers, are determined within the context of the specified volume. For instance, if the volume is set to 10 with parameters [True, True], only the middle 8 values are included in the calculation.

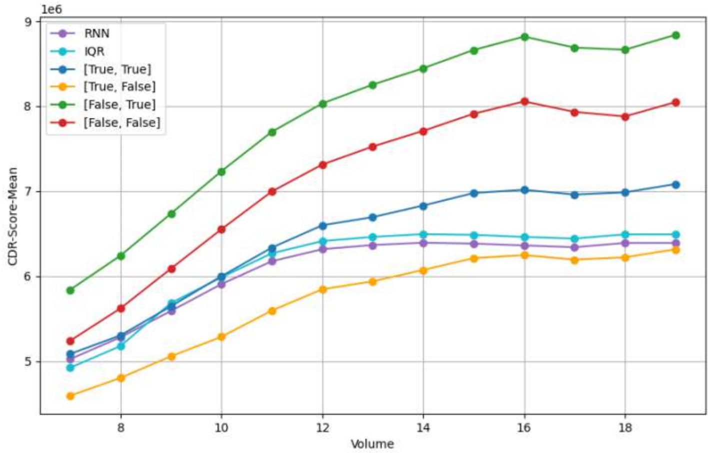


Figure 7.13: CDR - Short Tail Payments

The plot displays the mean CDR score for six distinct models, each represented by a different colour, indicating which potential outliers were excluded. The y-axis shows the average CDR score, while the x-axis indicates the volume, representing the number of periods considered in calculating the chain ladder factors. The values on the y-axis range from approximately 4,5 Million to almost 9 Million.

ShortTail

Figure 7.13 shows that the maximum CDR-Score-Mean is in the [False, True] model with 16 periods for the ShortTail - paid loss, with values reaching almost 9 million, while the minimum is in the [True, False] model, considering 7 periods, at around 4.5 million.
It is important to note that the [True, False] model consistently yields lower CDR-Score-Mean values than the other models. The highest values are attributed to the [False, True] model, followed by the [False, False] model, with the [True, True] model ranking as the second-best performer. This performance disparity significantly correlates with the choice of outlier detection method. The lowest point for the least performing model, in terms of the CDR-Score, is approximately 5.9 million, while the best model exhibits a worst-case value of 6.2 million. Given that the range of values lies between 4.5 and 9 million, this indicates that the overall best model outperforms others considerably.

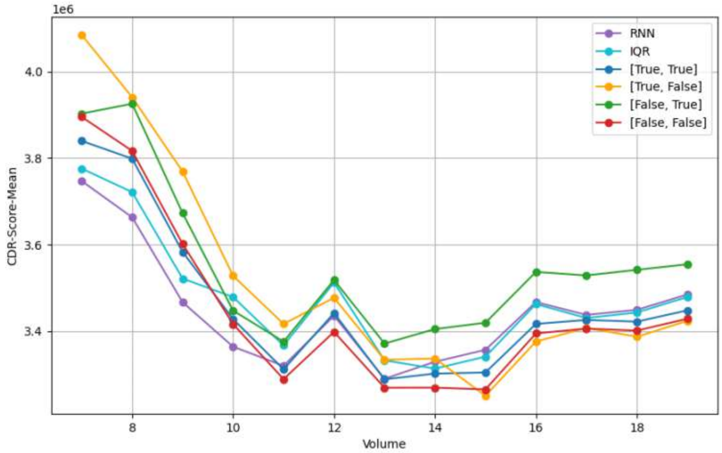


Figure 7.14: CDR - Short Tail Incurred

This plot has the same structure as figure 7.13. The values on the y-axis, range from approximately 3.25 Million to 4.1 Million.

Generally, there is a clear pattern in all models. As the volume increases, the CDR-Score-Mean generally rises across all models, which may suggest that including more periods results in a higher mean CDR score for paid losses. It is worth noting, that the RNN- and IQR-models are relatively flat between 12 periods and more. One possible explanation for the general trend is, that we need fewer periods for an adequate comparison due to the branch's fast-settling nature and the stability of link ratios across accident years. Including too many periods may introduce distortions in the chain ladder factor due to system changes over time. For example, claim settlement practices from 20 years ago differ significantly from today's. Historically, link ratios tended to stabilize later, unlike now, where most claims resolve after the first development period. This shift leads to smaller chain ladder factors in the early development periods but more greater factors in the later periods. Given that settlement patterns have evolved, incorporating link ratios from 20 years ago could considerably distort the chain ladder factors of today. The values slightly diverge as the volume increases beyond 19 periods. While they only differ by about 1.6 Million for volume 7, the difference increases to approximately 2.5 Million. Until approximately 12 periods, the increase of the CDR-Score-Mean is relatively steep in contrast to the remaining periods.

Shifting to the plot of incurred losses, one can see that the maximum CDR-Score-Mean is observed in the [True, False] model with 7 periods for the ShortTail - incurred loss, with values reaching around 4.1 million, while the minimum is in the [True, False] model, considering 15 periods, at around 3.25 million. Nevertheless, it is essential to note that the values only vary around 0.35 Million after 10 periods. The rapid decline at the start could be explained by the fact that the incurred claims contain information not only about the payment data but also about the reserves.

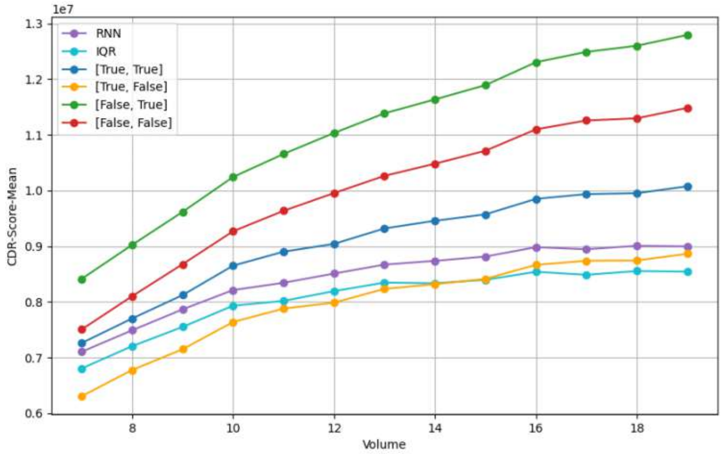


Figure 7.15: CDR - Long Tail Payments

This plot has the same structure as figure 7.13. The values on the y-axis range from approximately 6.3 Million to almost 13 Million.

Comparing figure 7.13 and figure 7.14 we can see a notable difference in the range of the CDR-Score-Means, scores for the paid losses are more significant than for the incurred data. Where for incurred data, the values only vary about 1 million; for the paid data, the values almost double. A further difference that is worthy of note is the pattern different from that of the paid data. By comparing figure 7.13 and figure 7.14, we can see a difference in their volatility. Even if they are smooth compared to the volatile branch, they still have some fluctuations. If we take them into perspective, they are not as small as they seem. For example, in the Short Tail branch in 2006, the Link Ratio was only 1.0005, while in 2001, it was 1.0716. Although the differences in link ratios appear minor at first glance—for example, 0.07 between years—their multiplicative effect results in progressively more significant discrepancies as they are applied across successive periods. If we have 1 million € and multiply them by 1,0716, we have 1.071.600€, whereas, with 1,0005, we only have 1.000.500 €. We can see that we have even greater values by looking at reserves.
Interestingly, the RNN and IQR outlier detection perform well compared to the other models, but they are not the best-performing outlier detection for either of the two triangles; however, they seem to be a reasonable choice for the paid and incurred data.

Comparing figure 7.13 and figure 7.14, fitting the incurred data leads to more accurate results, as even the models for the incurred data yield lower CDR-Score-mean values across all models.

Long Tail

By looking at figure 7.15 It is essential to mention that the [True, False] model consistently yields lower CDR-Score-Mean values than the other models. The highest CDR-Score-Mean is observed in the [False, True] model with 19 periods, with values reaching around 13 million, while the minimum score is in the [True, False] model, considering 7 periods, at around 6.3 million. The highest values are attributed to the [False, True] model, followed by the [False, False] model, with the [True, True] model ranking as the second-best performer. This performance discrepancy significantly correlates with excluding the highest or the lowest link ratio or none. The lowest point for the least-performing model, in terms of the CDR-Score, is approximately 8.4 million, while the best model shows a worst-case value of 8.9 million. Given that the range of values lies between 6.3 and 12.9 million, the overall best model outperforms others considerably.

Generally, there is a clear pattern in all models. As the volume increases, the CDR-Score-Mean generally rises across all models, which may suggest that including more periods results in a higher mean CDR score for paid losses. The possible explanation is the same as for the short-tail paid claims data.

The values slightly diverge as the volume increases, while they only differ by about 2.2 million for volume 7, the difference increases to approximately 4 Million until volume 19. The steepness of the graph also declines with the inclusion of more and more periods. In comparison to the Short-tail paid plot, the pattern indeed seems similar, but the scope is different. The CDR score mean of the Long Tail models is approximately 7 million bigger than the Short Tail models. It is worth noting that the RNN and IQR outlier detection are two of the three best models, but for volumes smaller than 15, the [True, False] model has smaller CDR-Score-mean values, similar to the Short Tail data.

We proceed by analyzing figure 7.16. The maximum CDR-Score-Mean is observed in the [False, True] model with 11 periods for the long-tail incurred loss, reaching around 10.41 million, while the minimum is found in the [False, False] model, with 7 periods, at approximately 8.9 million. The values oscillate almost like a wave, starting at a low point, peaking, then declining, followed by a small rise before dipping again. The volatility decreases as the volume increases.

Note that the [False, True] model consistently produces the highest scores, while the other models perform similarly. Except for volume 7, the [True, False] model outperforms the other models across all volumes. The RNN model performs for some volumes worse than others, the results are comparable to the [False, True] model. While the IQR model has a higher CDR-Score-Mean as the [False, True] model, it is comparable to the other models in the midfield for volumes greater than 9.

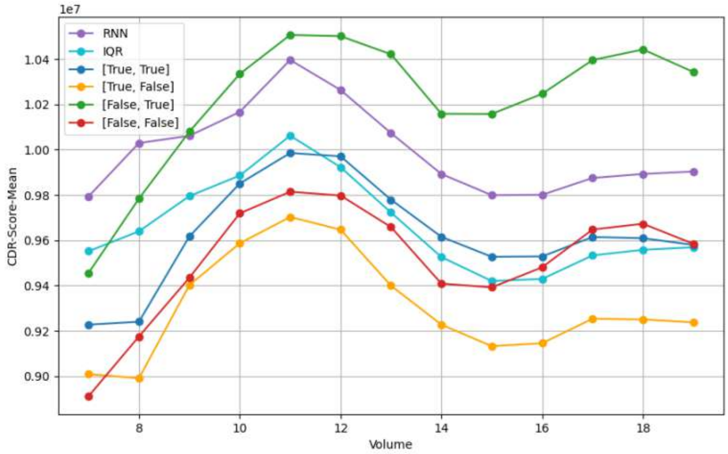


Figure 7.16: CDR - Long Tail Incurred

This plot has the same structure as figure 7.13. The values on the y-axis range from approximately 8.9 million to 10.41 million.

Like in Short Tail, the plot of the incurred loss looks utterly different from the plot of the paid loss. Also, the range of the CDR-Score-Mean values is notably narrower than that of the paid data. When using the right outlier detection, the paid data is a better choice for accurate reserving than incurred data since the CDR values are lower for the best performing paid models than the values for all incurred models.

Volatile

Our analysis now shifts to the plot of paid losses for the volatile-tail branch. The maximum CDR-Score-Mean is observed in the [False, True] model with 7 periods for the Volatile- paid loss, with values reaching around 15.5 million, while the minimum is in the RNN model, considering 12 periods, at around 12.5 million.

Interestingly, the peak is already appearing at volume 7 and rapidly declines to a low point. After the global low, an inflexion point appears, and the values stabilize. This pattern could result from the unstable nature of the link ratios from volatile branches. Since high-severity claims primarily drive losses on the volatile branches, more periods are probably needed for comparison.
It should be noted that the [False, True] model consistently has the highest scores, while the other three models perform similarly, and the RNN model has the lowest scores for volumes larger than 12, reaching the lowest CDR Score Mean at volume 12. It is worth noting that the volatile paid data is the first triangle, where the more complex outlier detection methods have better results than the other; the RNN outlier detection outperforms the IQR method. The reason for that could be the volatile nature of the branch, which makes it harder to identify the outliers.

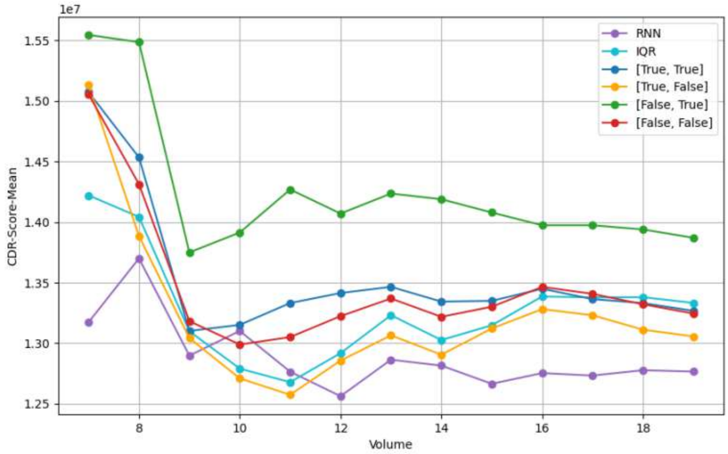


Figure 7.17: CDR - Volatile Payments

This plot has the same structure as figure 7.13. The values on the y-axis range from roughly 12.5 Million to 15.5 Million.

As one can see in figure 7.17 the pattern of the volatile branch has a completely different shape compared to the other branches, since it almost seems inverted in the progression over the volume, in the beginning, there is a rapid decrease, and after that, it stabilizes. Comparing the link ratios shows the pattern looks so different from the other two branches. The other branches have very steady Link ratios, whereas the volatile branch has volatile values. This branch is driven by major loss events, which only happen in some years, and the claims settlement can be complicated—naturally, the more history, the better for an excellent ultimate estimation.
Turning to the plot of incurred losses for the volatile-tail branch shows, that the maximum CDR-Score-Mean is observed in the RNN model with 7 periods for the volatile incurred loss, reaching around 7.3 million, while the minimum is seen in the [True, True] model, with 13 periods, at roughly 6 million.

For the incurred claims, the behaviour of the individual models is analogous, though they differ significantly in magnitude. Notably, the [True, False], RNN and IQR models exhibit the poorest performances, while the remaining models perform comparably. Interestingly, the maximum value appears for volume 7, and within volumes 7 and 10, it sharply declines toward a dip. Following this decrease, a peak materializes between 10 and 13 periods, succeeded by the global minimum. The values from 13 to 19 periods remain relatively stable. Notably, unlike other branches, the shape of the graph for incurred data in this branch does not deviate remarkably from that of the paid data. Disregarding the prominent spike from volume 10 to 13, the patterns are largely congruent.
Comparing the two plots of the volatile data, the incurred triangle model has significantly lower CDR-Score-Means, suggesting the use of the incurred data for reserving.

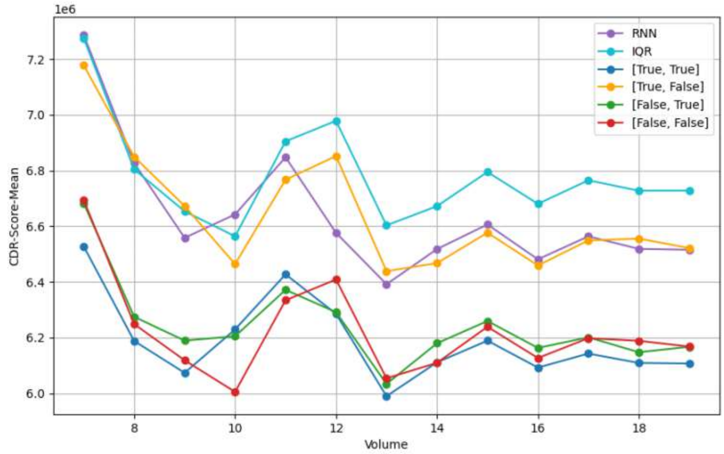
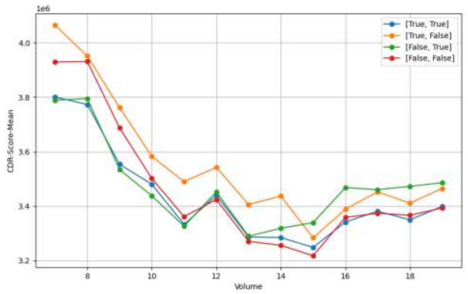


Figure 7.18: CDR - Volatile Incurred

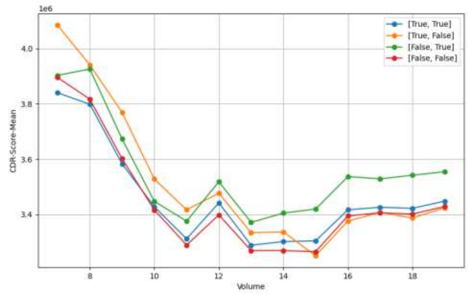
This plot has the same structure as figure 7.13. The values on the y-axis range from 6 million to 7.3 million.

Conclusion

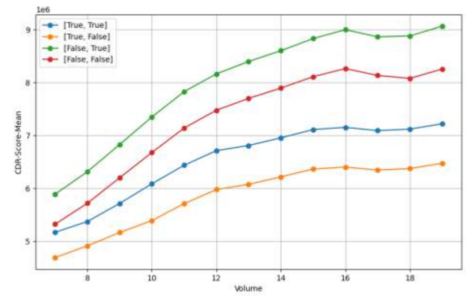
The volume has a significant impact on the model's performance. The impact of the volume on the CDR score mean depends on the branch and whether to look at the paid or incurred losses to decide which model parameters to use. Long and Short Tail are similar because of the stable link ratios over the different accident years; they both provide better results for a smaller volume, whereas the model for the Volatile branch performs better with a greater volume. Notably, all models for the paid and incurred long data perform best when excluding the highest value and including the lowest value. This indicates that the model is more sensitive to high outliers than to low ones. For every triangle, except the volatile payment triangle, the more straightforward outlier detection methods performed better than the IQR and RNN outlier detection. This shows that simpler methods sometimes outperform more sophisticated approaches.



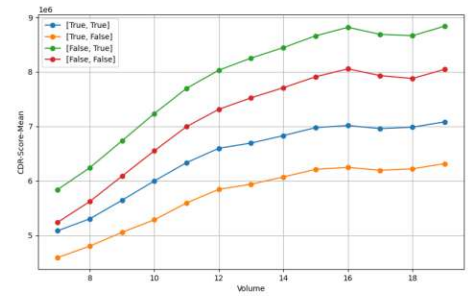
(a) incurred loss - 2010



(b) incurred loss



(c) paid loss - 2011



(d) paid loss

Figure 7.19: CDR-Score-Mean ShortTail

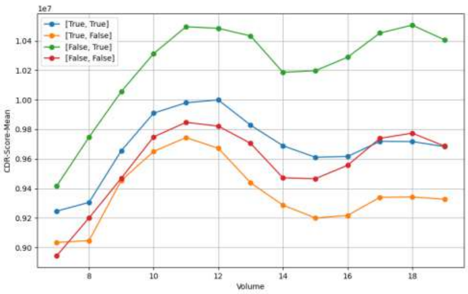
(a) and (c) show the CDR-Score-Mean values for the models by additionally excluding one diagonal. (b) and (d) show the CDR-Score-Mean values for the models without the exclusion of specific diagonals, compare to figure 7.13 and 7.14.

7.2.2 Analysis of removing whole diagonals from the triangle data

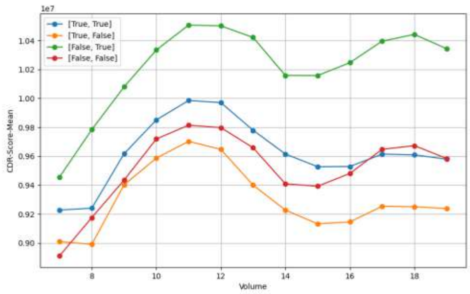
In the following section we will examine the impact of removing a conspicuous diagonal to the CDR-Score-Mean values. This diagonal is selected according to the number of RNN outliers on the full triangle, occuring on the diagonal. The diagonal with the most outliers will be selected, if two diagonals have the same absolute amount of outliers the diagonal with the most relative amount of outliers of them will be selected.

ShortTail

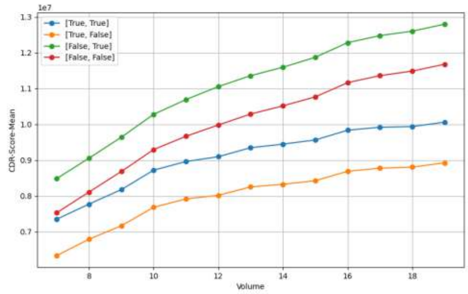
As we can see in figure 7.19 the graphs slightly change for the incurred loss data. The diagonal 2010 for the incurred data has 8 outliers, which results in 72,7% outlier in this diagonal. There are no severe changes, but the CDR-Mean-Score overall is slightly lower for the models excluding the diagonal of 2010.
Diagonal 2011 for paid loss only has 3 outliers, resulting that 25% of this diagonal are outlier. Excluding this diagonal does not improve the results, for example the [False, True]



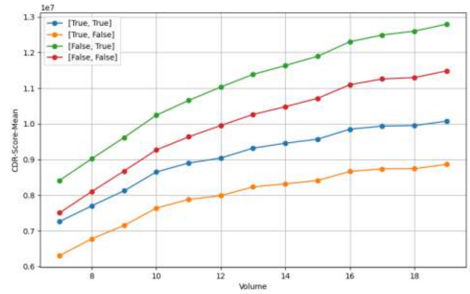
(a) incurred loss - 2019



(b) incurred loss



(c) paid loss - 2016



(d) paid loss

Figure 7.20: CDR-Score-Mean LongTail

(a) and (c) show the CDR-Score-Mean values for the models by additionally excluding one diagonal. (b) and (d) show the CDR-Score-Mean values for the models without the exclusion of specific diagonals, compare to figure 7.15 and 7.16.

model for volume 12 an CDR-Score-mean of 8 million, with the additional exclusion of the diagonal 2010 is is slightly more than 8 million.

LongTail

The diagonal with the most outliers for the incurred data is 2019 with 3 outliers, resulting in 15 % outlier in this diagonal. Exluding this diagonal even deteriorates the results as seen in figure 7.20 for [False, True] model for volume 14.
For the paid data the diagonal with the most outliers is 2016, with 3 outliers, resulting in 17,6% outliers in this diagonal. The plot does not significantly change.

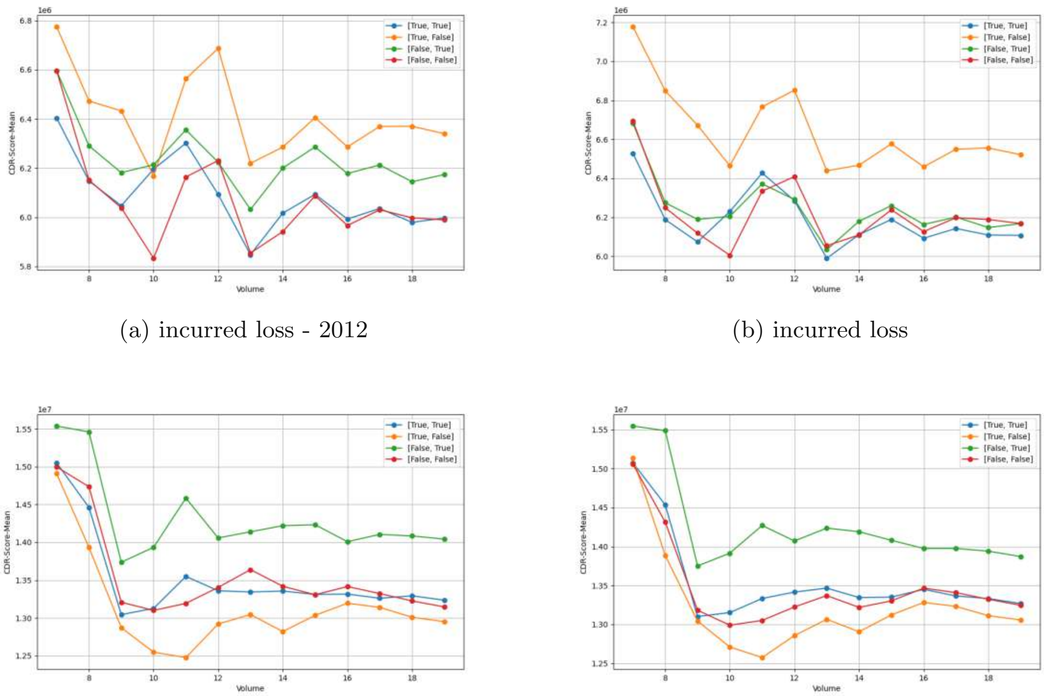
Volatile

In diagonal 2012, for the incurred data, the most outliers occur. There are 5 outliers in this diagonal, resulting in a 38.5% outlier quota. Excluding this diagonal clearly improves the results, as we can see in figure 7.21.
For the paid data, with 5 outliers, the diagonal with the most outliers is 2012, subsequently

38.5% of the values in this diagonal are outliers, nonetheless the exclusion of this diagonal does not significantly improve the results.

Conclusion

Concluding the findings in this section, the greatest improvement brings excluding the diagonal of 2012 in the incurred loss triangle from the volatile branch. An improvement for the approach used in this thesis for finding a diagonal to exclude could be, to look at the relative frequency as the main measure.



(a) incurred loss - 2012

(b) incurred loss

(c) paid loss - 2012

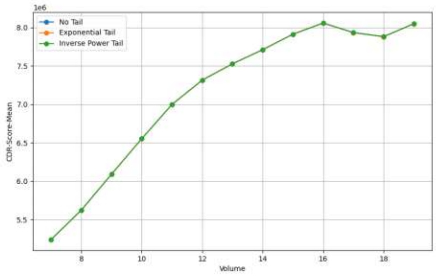
(d) paid loss

Figure 7.21: CDR-Score-Mean Volatile

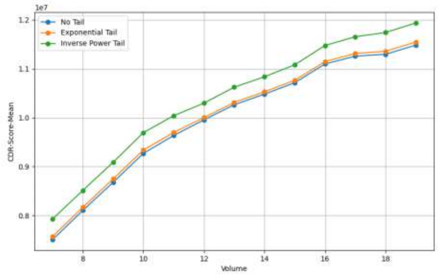
(a) and (c) show the CDR-Score-Mean values for the models by additionally excluding one diagonal. (b) and (d) show the CDR-Score-Mean values for the models without the exclusion of specific diagonals, compare to figure 7.17 and 7.18.

7.2.3 Tail Adjustment

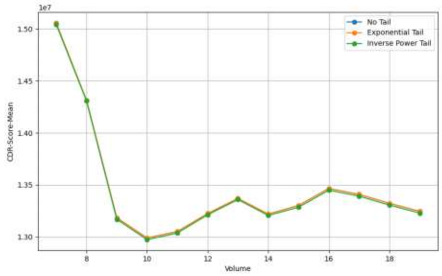
This Section shows the different CDR-Score Means for models, including the tail factor. As we can see in figure 7.22, the CDR-Scores look almost identical to the scores of the models without the tails. The similarity in the course of the CDR scores is an expected result since CDR is a very short-term approach, and the tail factor shows its impact in the long term.



(a) ShortTail



(b) LongTail



(c) Volatile

Figure 7.22: CDR-Score with Tails

The plots show the CDR-Score-Mean values for the model including all periods and excluding neither the maximum value nor the minimum value. The blue line shows the values for considering no tail, the orange line considering an exponential tail and the green line considering an inverse power tail.

ShortTail

Since the short-tailed branch does not have a tail, the plots are identical. Whether we apply an exponential curve fit or an inverse power fit, the estimated tail will be 1.0.

LongTail

Referring to figure 7.22b, they all seem the same at first glance, but on closer inspection, one can see a shift to the detriment of the inverse power tail. The values of the exponential tail model are slightly higher than those for no tail.

Volatile

In the volatile sector, the CDR results for both models, the exponential and no-tail models, are almost identical, while the inverse power-tail model shows some minor variations, but generally, the differences are negligible.

Conclusion

In the case of this study, the CDR is unsuitable for determining the optimal tail. The tail represents the extrapolation of claim payments into the future. A tail adjustment must consider the development of the entire claim if the claims are not fully settled by the end of the claims triangle. Every new year, we only get one cell more information about the tail since the top right corner of the triangle describes it. The little information won has the consequence of only having a minor influence on the CDR. When calculating the CDR score, the values are normalized based on incremental claims already known. Since the tail is an additional estimation and, therefore, primarily increases the ultimate estimates, it is reasonable to observe an increase in the CDR values. Even if the relative difference between the ultimate values decreases slightly with the tail adjustment, the absolute difference may become significantly larger because of the ultimate increase with the tail adjustment. That could result in the ultimate increase outweighing the relative convergence. To address the tail fit assessment more effectively, it may be beneficial to analyze the tail factors separately. For instance, focusing only on the first row of the triangle and comparing the initial estimates made for the years in question with the actual realized values could be a better approach to checking the quality of the tail measure.

7.2.4 Weighted Average of paid loss and incurred loss

This section shows the impact of taking a weighted average of the paid and incurred ultimate on the CDR-Score, illustrated in figure 7.23. In this section three different models are considered:

- [**False, False, -1**] (red line): model without interference, no exclusion and inclusion of all periods.
- [**True, False, 7**] (green line): The highest value is dropped and considering only 7 periods.
- [**False, True, -1**] (blue line): The lowest value is dropped and considering all periods.

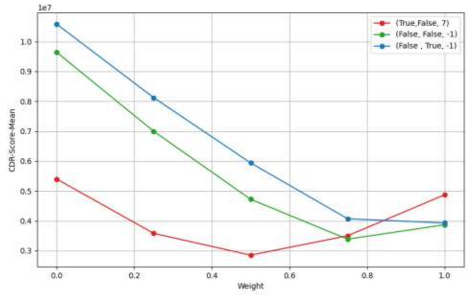
These three models were chosen because they had very different performances for the various models. We measure the CDR between the weighted triangle and the paid ultimate. We use the paid ultimate because the incurred ultimate and the paid should technically be the same since the reserves converge to zero and the incurred ultimate only illustrates the payments.

ShortTail

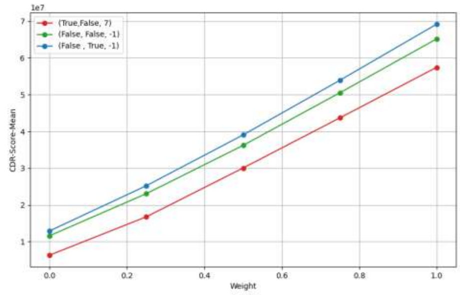
As seen in figure 7.23a, the CDR-Score values range from 3 million to almost 11 million. The low point reaches the model with no interference and a weight of 0.5, while the peak is for the model excluding the lowest value and using the paid triangle. The weight that is preferable depends on the volume chosen. For models that consider the whole history, the incurred triangle performs better, while the best accuracy is achieved by the model that considers only 7 periods and takes 50% paid ultimate and 50% incurred ultimate.

LongTail

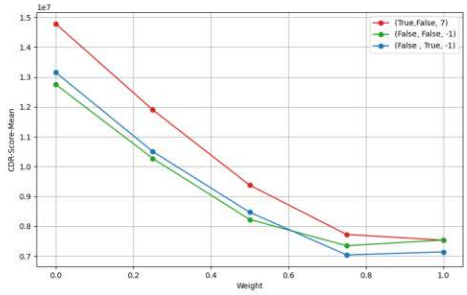
The CDR-Score values range from approximately 9.5 million to almost 70 million, as we can see in figure 7.23b. The low point, with a weight of 0.5, reaches the model with no interference. The peak is for the model excluding the lowest value and using the paid triangle. An explanation for the remarkably high CDR-Score values could be that the paid ultimate does not have a tail considered. However, the ultimate values will differ since the incurred claims consist of reserves plus payment, and tails are included in the reserves. Nevertheless, the analysis suggests using only the paid data or only a small weight, which is consistent with the finding from subsection 7.2.1, which showed that the paid triangle leads to better results. It is important to note, that reserving the long tail data without a tail and only payment data will very likely result in underestimating the ultimate values, since as we can see in the link ratio triangle in figure 7.2.1, the claims have not been settled.



(a) Weights ShortTail



(b) Weights LongTail



(c) Weights Volatile

Figure 7.23: CDR-Score-Mean Weighted Ultimate

This figures have a structure similar to figure 7.13. The y-axis shows the CDR-score mean, while the x-axis represents the weight. As one can see in equation 6.1, weight = 1.0 means the ultimate only results from the incurred triangle, and weight = 0.0 indicates that only the paid ultimate was considered.

Volatile

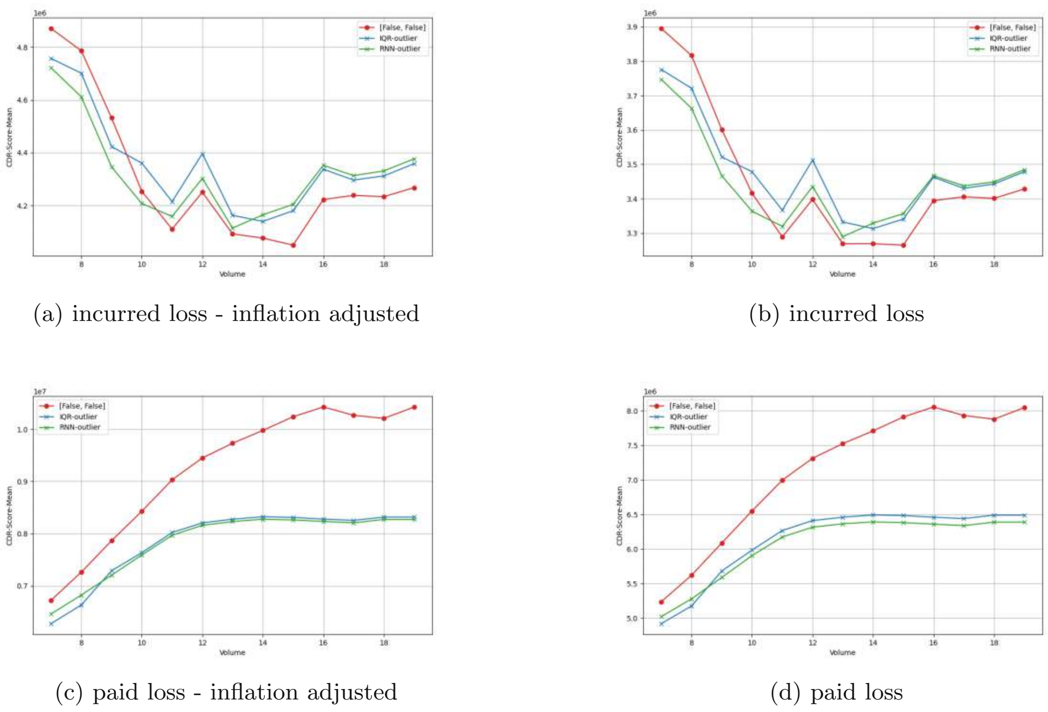
The values from figure 7.23c range from 7 million to almost 15 million. We can see a drastic decrease by increasing the weight of the incurred ultimate. In contrast to the LongTail branch, using the most incurred ultimate is preferable; the CDR values are significantly lower.

Conclusion

Generally, the choice of weight strongly depends on the branch. According to this analysis, the estimation for the short and volatile branches benefits from including the incurred ultimate. However, the long-tailed branch model has the most accurate results using only the paid ultimate. This insights are consistent with the findings in 7.2.1. However, we need to mention that the outcome from the long-tailed branch could result from the lack of tail inclusion.

7.2.5 Inflation adjusted Data

In recent years, inflation has started to become a significant topic again. The question is, how do inflation-adjusted claim values impact the performance of the models regarding the CDR? It can be answered by examining figures 7.24 to 7.26. One can see that the patterns do not change, but the level does. Interestingly, the values increase instead of decrease. The reason may be that inflation is already accounted for to some extent in the data. Specifically, it is integrated into the reserves but needs to be adjusted annually. For example, consider a claim that occurs in the year 2000. A reserve is set up at that time, which already factors in the estimated inflation. In the initial years, adjustments may be made as payments are issued. However, payments must be made before 2005 to ensure further adjustments to the reserve occur. If, by 2015, it becomes evident that the allocated reserve is insufficient, an adjustment is made. Importantly, this adjustment is not only for the upcoming year but for the remaining period of the claim, as would logically be necessary. Therefore, the applied inflation adjustments might not align with the actual practice. The approach assumes annual adjustments, which do not reflect differences from how reserves are typically managed in the real world.



(a) incurred loss - inflation adjusted

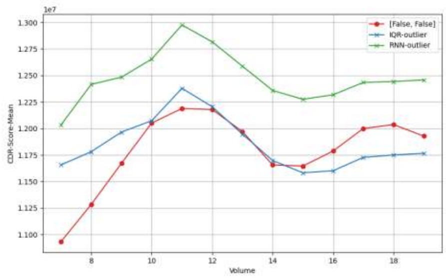
(b) incurred loss

(c) paid loss - inflation adjusted

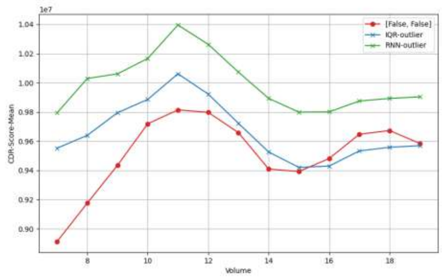
(d) paid loss

Figure 7.24: CDR-Score-Mean ShortTail

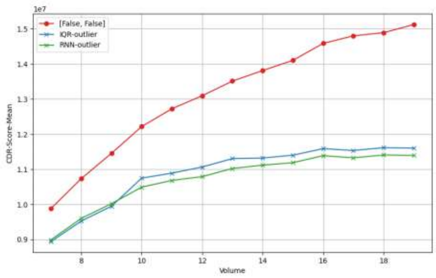
This figure shows the CDR-Scores for the various branches using inflation-adjusted and raw data.



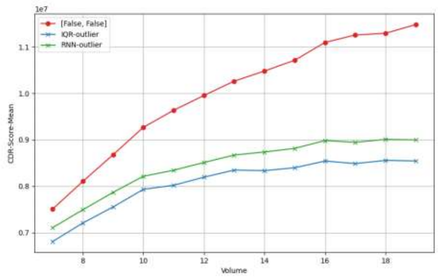
(a) incurred loss - inflation adjusted



(b) incurred loss



(c) paid loss - inflation adjusted



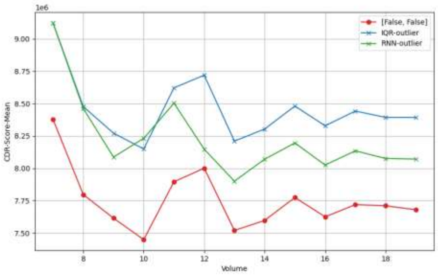
(d) paid loss

Figure 7.25: CDR-Score-Mean LongTail

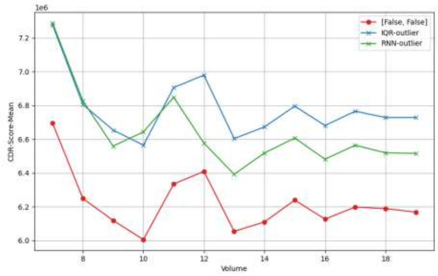
This figure shows the CDR-Scores for the various branches using inflation-adjusted and raw data.

7.2.6 Conclusion

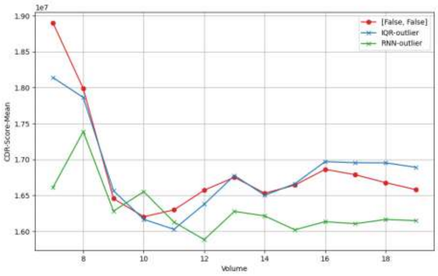
The analysis highlights that the number of periods considered and the selection of excluded values significantly impact model performance. However, the Reverse Nearest Neighbors (RNN) and Interquartile Range (IQR) methods do not outperform more straightforward approaches, such as excluding the maximum and minimum values. Removing the diagonals by excluding the diagonals with the most RNN outliers did not improve the models. A different approach would likely be needed to produce meaningful improvements. Additionally, as previously mentioned, the CDR score is unsuitable for evaluating tail factors. For certain branches, incorporating both the paid and incurred ultimate estimation significantly improves the model accuracy, particularly for volatile lines of business, where this approach leads to significant improvements. However, the inflation adjustment method applied in this thesis does not seem appropriate and fails to improve the CDR results.



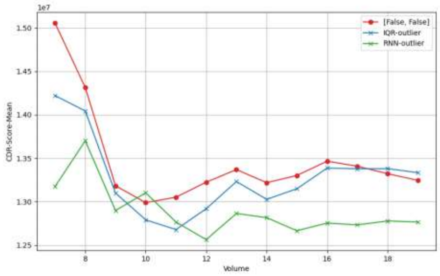
(a) incurred loss - inflation adjusted



(b) incurred loss



(c) paid loss - inflation adjusted



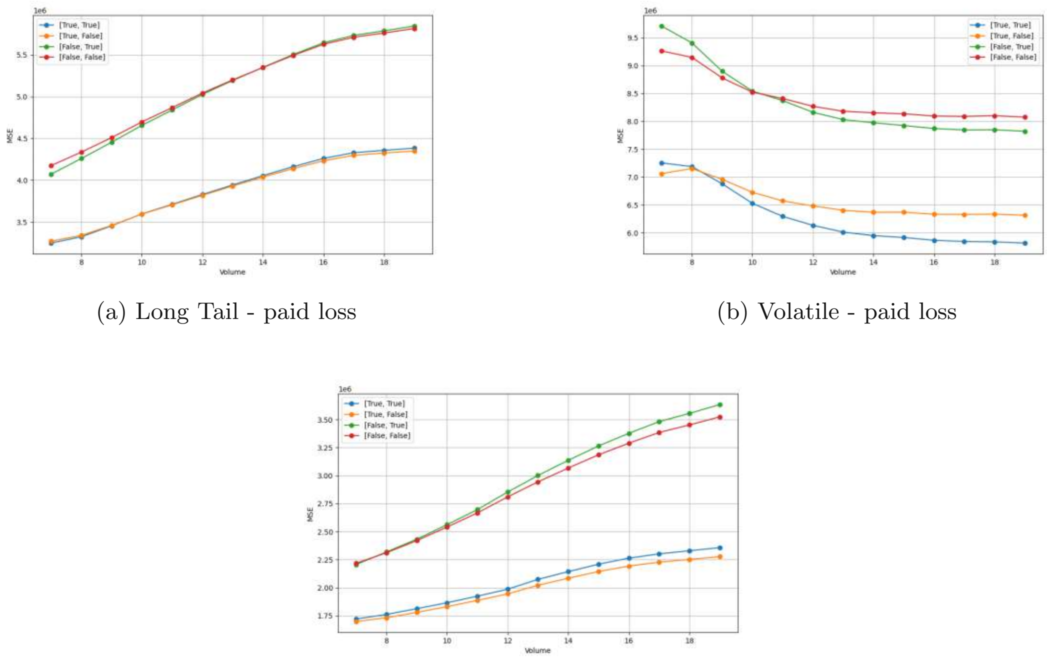
(d) paid loss

Figure 7.26: CDR-Score-Mean Volatile

This figure shows the CDR-Scores for the various branches using inflation-adjusted and raw data.

7.3 Comparison with MSE

This section shows the behaviour of the MSE regarding the simple outlier detection already mentioned in subsection 7.2.1. The comparison with the MSE helps us to understand the results of the CDR score. The plots for visualising the data are structured in the same way as those in subsection 7.2.1. Note that we only consider paid data here, that is because the Mack Standard Error needs the chain ladder factors to be greater than one, as we can see in the 7.7, the incurred columns are smaller than one for most of the development years, hence we cannot use the MSE.



(a) Long Tail - paid loss

(b) Volatile - paid loss

(c) Short Tail - paid loss

Figure 7.27: Mack Standard Error

Visualisation of the MSE for the given models, structured like the figures before, like figure 7.19d.

Short-Tail

Figure 7.27c shows that the MSE values for the Short Tail range from approximately 1,75 million to 3,6 million. Where the peak is reached by the [False, True] model using all periods and the low point by the [True, False] model using only 7 periods. All lines show the same pattern: the MSE increases as the volume increases. The graphs can be divided into two groups: The two models exclude the highest value, and the two models include the highest value. The models excluding the outliers show significantly better results than those including the outliers. The shape of the graphs looks similar to the shape of the CDR graphs but smoother.

Long-Tail

As seen in figure 7.27a for the Long-Tail, the values range from approximately 3.2 million to 5.8 million. The [False, True] model reaches the peak considering the most periods, while the [True, True] model reaches the lowest point using 7 periods. The plot looks similar to the Short Tail plot. The same groups form, and the graphs have a very similar shape, increasing CDR for increasing volume. Generally, the uncertainty is driven by the maximal values in the Long Tail branch.

Volatile

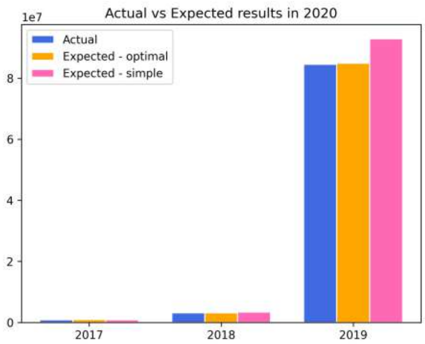
The MSE values range from 5.75 million to 9.75 million, while the maximum is reached with the [False, True] model for 7 periods and the minimum is from the [True, True] model considering the most periods. The models can be again divided into the same two groups as in the other two branches. The shape is almost inverse to the shape of the long tail graph; the values decrease with increasing volume, similar to the CDR plot. In contrast to the CDR plot, the curve is way smoother; there is no sudden severe decrease in the first volumes.

Conclusion

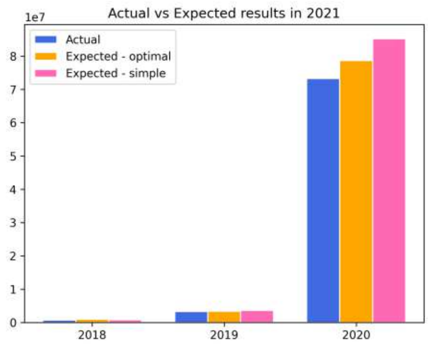
The Mean Squared Error (MSE) values show significantly smoother plots than the CDR values. The fact that both measures portray similar trends implies a correlation between the overall uncertainty in the data. The pattern should be similar, especially in the Short Tail branch, since the horizon where the uncertainty lies is almost only the next year.

7.4 Actual vs. Expected

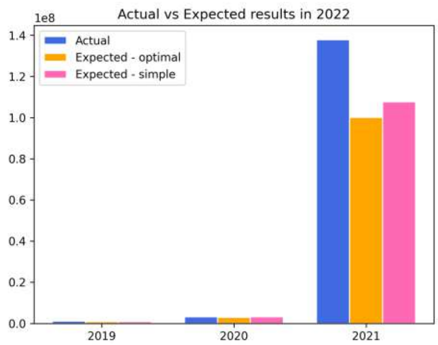
The figures 7.28 to 7.30 show an actual versus expected analysis. For each branch, three plots are presented, corresponding to the last three years. The analysis compares the actual payments (Actual) with the expected payments (Expected) as projected by the models. Among the expected values, a distinction is made between the benchmark model (depicted in pink) and the model identified as optimal by our algorithm (depicted in orange). Since most payments occur in the most recent years, model accuracy is particularly critical during this period. As such, the analysis focuses exclusively on the data from the final three years. This ensures that the evaluation emphasizes the models' performance in accurately predicting payments during the most relevant time frame.



(a) 2020



(b) 2021

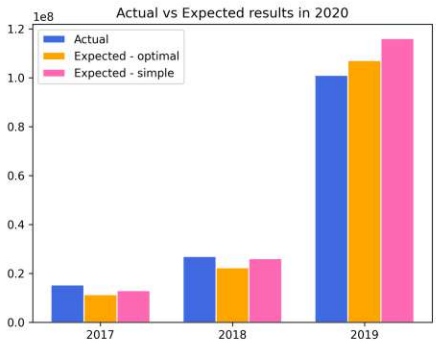


(c) 2022

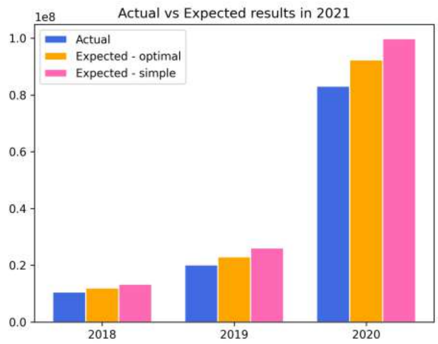
Figure 7.28: Actual Vs. Expected Short

ShortTail

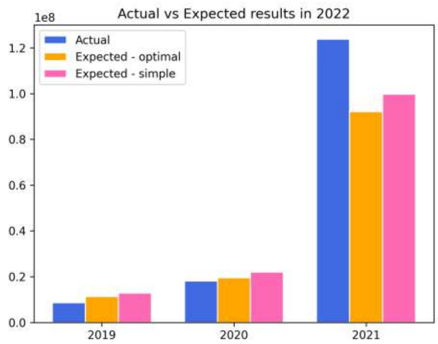
As seen in figure 7.28, the optimal model has more minor deviations from the actual result for two of the three years than the classic chain ladder model. The one year in which the gap is more significant is 2022. 2022 was an exceptional year because inflation rose very sharply. Only the most recent year is relevant because the short-tailed branch payments are negligible after the second year, as one can see. In recent years, the optimal model has produced smaller estimations than the other model. This could result from eliminating the highest value and including only the most recent 7 years. As we can see in the heatmap in figure 7.1, the highest link ratios are in the earlier 2000s; when including only the most recent seven years, the link ratios are lower, which results in lower ultimate estimates.



(a) 2020



(b) 2021

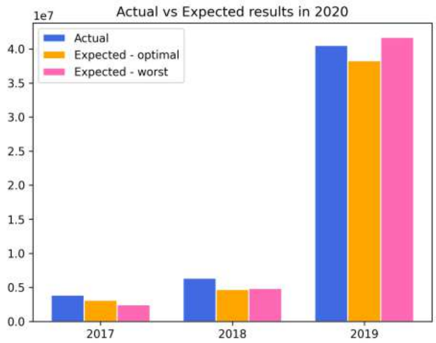


(c) 2022

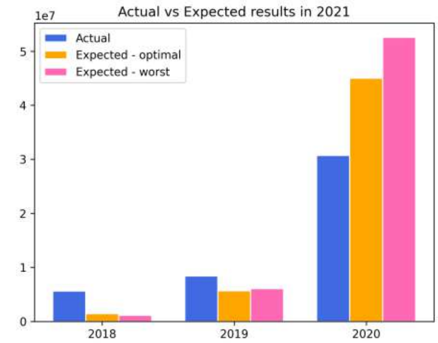
Figure 7.29: Actual Vs. Expected Long

LongTail

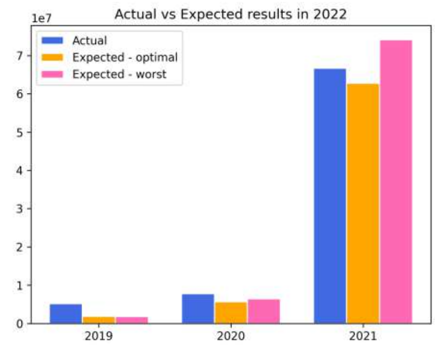
As seen in figure 7.29, similar assertions have been held for the long-tailed branch in the most recent years. Only in 2022 did the simple chain ladder model have better results. Contrary to the short-tailed branch, the older years are not negligible. The optimal model performs better in the older years, except for 2020. The development of the link ratios looks similar to the development of the link ratios of the short-tailed branch; therefore, the same argument holds for the classic model aims for higher ultimate values.



(a) 2020



(b) 2021



(c) 2022

Figure 7.30: Actual Vs. Expected Volatile compared with [False,True] periods=7.

Volatile

Looking at the most recent year, the optimal model performs better than the classic or almost the same. Interestingly, this model can better deal with inflation; 2022 payments are not as massively underestimated as in the other sectors.

Conclusion

In all three branches, we can see that the optimal model performs in two of the three years with more accurate estimations than the other model. The optimal model has lower estimations than the other model, while the simple model overestimates the claims more. The goal to get a more accurate model is achieved, but since, by Austrian law, the principle of prudence holds, often more conservative measures are preferable. An idea would be to adapt the error measure and include a penalty for underestimating the claims.

8 Conclusion

This thesis explores the optimisation of claims reserving methods, focusing on the Chain Ladder model. The analysis examines the impact of various parameters, including outlier exclusion, period selection, inflation adjustment, weighing of the paid and incurred ultimate, and tail adjustments across different insurance branches. The goal is to identify optimal configurations that improve accuracy and provide robust reserving estimates.

One central insight is the importance of applying different approaches for incurred and paid losses and specific insurance branches. The results demonstrate that the optimal parameters differ significantly between incurred and paid losses. For example in the Short-Tail branch, the volume selection is the opposite for the two datasets. Considering the different branches, two groups are identifiable: volatile and non-volatile branches. Methods for Short-Tail and Long-Tail branches yield the best results when fewer periods are considered, due to their lower volatility and sensitivity to settlement patterns. Models for volatile branches require more periods for stabilisation. This finding indicates that simpler Chain Ladder models without adjustments perform better for volatile branches. Regarding outlier detection, methods such as Reverse Nearest Neighbor (RNN) and Interquartile Range (IQR) produce comparable results to models that exclude only the maximum value. However, simpler exclusion methods are preferable for their efficiency. Removing diagonals has limited impact overall, except for volatile branches, where excluding specific diagonals improves results.

The analysis also shows that the CDR Score is not a suitable measure for optimising the tail factor, and the inflation adjustment has negligible effects. The Mack Standard Error (MSE) aligns closely with the Claims Development Result (CDR). In the final Actual vs. Expected Analysis, the optimised models mainly provide more accurate results but often lead to lower ultimate values. This outcome is not always favourable, as Austrian prudence principles favour conservative reserving, prioritising overestimations over underestimations to ensure financial stability.

This thesis introduces an anatomised approach for optimising Chain Ladder models, which provides tools for actuaries to improve reserving accuracy while minimising manual effort. The findings highlight that simplicity often outperforms complexity. While volatile branches benefit from stabilisation through extended periods, simpler exclusion methods prove effective across most scenarios.

The algorithm relies solely on historical data, which makes it less suitable for sudden events such as catastrophes or prolonged high inflation. Actuarial judgment remains essential, as experts are required to classify external factors, for example if a high link ratio in the last year represents an outlier or the start of a prolonged high inflation period.

This work opens several ideas for future research. First, advanced outlier detection methods accounting for link ratio patterns could be developed; increasing link ratios might indicate a real change, such as a prolonged high inflation period, while the implemented outlier detection methods could classify them as outliers. Another crucial topic for future research is the incorporation of an advanced method for tail adjustment. The implemented CDR Score is found to be unsuitable, but its refinement is particularly urgent for long-tail branches. Improvements could be made by incorporating a more refined inflation adjustment. A penalty for underestimating reserves could also be included in the error measure to counteract the tendency to underestimate claims payments.

This thesis demonstrates the potential of automated optimisation in actuarial reserving. It offers a structured approach to enhancing claims reserving accuracy. While the tools developed support automated decision-making, they are not intended to replace actuarial judgment.

Bibliography

[1] Caesar Balona and Ronald Richman. The actuary and IBNR techniques: A machine learning approach. *SSRN Electronic Journal*, 2020. doi: 10.2139/ssrn.3697256. URL `https://ssrn.com/abstract=3697256`.

[2] Ronald L. Bornhuetter and Ronald E. Ferguson. The actuary and IBNR. In *Proceedings of the casualty actuarial society*, volume 59, pages 181–195, 1972.

[3] Chainladder Development Team. Chainladder python package. `https://chainladder-python.readthedocs.io/en/stable/intro.html`, 2024. Accessed: 2024-06-12.

[4] FMA - Finanzmarktaufsicht Österreich. Solvency II - Finanzmarktaufsicht. `http://www.fma.gv.at/de/sonderthemen/solvency-ii.html`, 2024. Accessed: 2024-06-12.

[5] Heinz-Willi Goelden, Klaus Th. Hess, Martin Morlock, Klaus D. Schmidt, and Klaus J. Schröter. *Schadenversicherungsmathematik*. Lehrbuch. Springer Spektrum, Berlin Heidelberg, 2. edition, 2023. ISBN 3662686236. URL `10.1007/978-3-662-68623-2`.

[6] Andreas Handl and Torben Kuhlenkasper. *Einführung in die Statistik: Theorie und Praxis mit R*. Springer Berlin Heidelberg Imprint: Springer Spektrum, Berlin, Heidelberg, 2018. ISBN 3662564408.

[7] D.M. Hawkins. *Identification of Outliers*. Monographs on applied probability and statistics. Chapman and Hall, 1980. ISBN 9780412219009. URL `https://books.google.at/books?id=fb0OAAAAQAAJ`.

[8] Norbert Henze. *Stochastik für Einsteiger : eine Einführung in die faszinierende Welt des Zufalls*. Lehrbuch. Springer Spektrum, Berlin, 13. edition, 2021. ISBN 3662638398.

[9] Steven C. Herman, Aaron Halpert, Michael R. Murray, Mark R. Shapland, S. Haria, Bernard A. Pelletier, Mohammed Q. Ashab, Bertram A. Horowitz, Susan R. Pino, Joseph A. Boor, Gloria A. Huberman, Anthony J. Pipia, Anthony R. Bustillo, Richard Kollmar, Fionnuala Ryan, David Anthony Tyeeme Clark, Joshua Merck, Scott Sobel, Robert J. Foskey, and Rasa Varanka McKean. The estimation of loss development tail factors: A summary report. In *Casualty Actuarial Society Forum*, 2013. URL `https://api.semanticscholar.org/CorpusID:110253797`.

[10] IFRS Foundation. IFRS - International Financial Reporting Standards. `https://www.ifrs.org/`, 2024. Accessed: 2024-06-12.

[11] Jelena Kocovic, Mirela Mitrasevic, and Dejan Trifunovic. Advantages and disadvantages of loss reserving methods in non-life insurance. *Yugoslav Journal of Operations Research*, 29(4):553–561, 2019. ISSN 0354-0243.

[12] Flip Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. *SIGMOD Rec.*, 29(2):201–212, may 2000. ISSN 0163-5808. doi: 10.1145/335191.335415. URL `https://doi.org/10.1145/335191.335415`.

[13] Thomas Mack. Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin: The Journal of the IAA*, 23(2):213–225, 1993. doi: 10.2143/AST.23.2.2005092.

[14] Michael Merz and Mario V. Wüthrich. Modelling the claims development result for solvency purposes. In *CAS E-Forum, Fall*, pages 542–568, 2008.

[15] NumPy Community. Numpy documentation. `https://numpy.org/about/`, 2024. Accessed: 2024-06-12.

[16] Pandas Development Team. Pandas documentation. `https://pandas.pydata.org/docs/index.html`, 2024. Accessed: 2024-06-12.

[17] Dimitris Papadias and Yufei Tao. *Reverse Nearest Neighbor Query*, pages 2434–2438. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_318. URL `https://doi.org/10.1007/978-0-387-39940-9_318`.

[18] Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Reverse nearest neighbors in unsupervised distance-based outlier detection. *IEEE transactions on knowledge and data engineering*, 27(5):1369–1382, 2014.

[19] Republic of Austria. Austrian Insurance Supervision Act 2016 (VAG 2016). `https://ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20009095`, 2017. Translated by author.

[20] Statistik Austria. Verbraucherpreisindex VPI/HVPI. `https://www.statistik.at/statistiken/volkswirtschaft-und-oeffentliche-finanzen/preise-und-preisindizes/verbraucherpreisindex-vpi/hvpi`, 2024. Accessed: 2024-12-03.

[21] Victor Sundberg. Application and bootstrapping of the munich chain ladder method. Master's thesis, KTH, Mathematical Statistics, 2016.

[22] Mario V. Wüthrich and Michael Merz. *Stochastic claims reserving methods in insurance*. Wiley finance series. John Wiley Sons, Chichester, England ; Hoboken, NJ, 2008. ISBN 111920626X.

List of Figures

List of Tables