U.PORTO

FEP FACULDADE DE ECONOMIA
UNIVERSIDADE DO PORTO

ESTIMATING THE ADJUSTMENT COEFFICIENT IN RISK THEORY

Rui Pedro Antunes Martinho Pinto

Dissertation

Master in Modeling, Data Analysis and Decision Support Systems

Supervised by

Professor Ana Cristina Moreira Freitas

2018

Acknowledgments

First and foremost, I would like to thank my supervisor, Professor Ana Cristina Freitas, for all her availability, dedication and help. She provided true guidance in a subject that was new to me, and her commitment was fundamental in this work.

I am also grateful to my parents, António José and Maria Helena, and to my brother João Luis for all their attention and support.

Resumo

A presente Tese, no seu essencial, será elaborada no âmbito da Teoria do Risco, sendo analisada, mais especificamente, a Teoria da Ruína e as suas diversas componentes. Assim, a base teórica passará pela análise de diversos conceitos dos quais se destacam as indemnizações a pagar pela companhia seguradora, os prémios a receber e a reserva inicial, tal como o estudo da probabilidade de ruína no âmbito do modelo de Cramér-Lundberg. Neste contexto, através da desigualdade de Lundberg, é possível obter um majorante para a probabilidade de ruína a partir do coeficiente de ajustamento.

Tendo como objectivo estimar o coeficiente de ajustamento, fez-se uso de um estimador de tipo geométrico existente na literatura, introduzido no contexto da estimação do coeficiente de cauda exponencial. Pretende-se demonstrar que, mediante determinadas condições gerais, esse estimador pode ser utilizado na estimação do coeficiente de ajustamento e também apresenta propriedades relevantes como a consistência e a normalidade assimptótica.

Um dos objectivos deste trabalho será também a realização de estudos de simulação onde se procurará analisar o comportamento amostral do estimador geométrico, mediante a variação de certos parâmetros e pressupostos de base.

Palavras-Chave: coeficiente de ajustamento, coeficiente de cauda exponencial, desigualdade de Lundberg, estimador de Hill, estimador geométrico, estudos simulação, Modelo de Cramér-Lundberg, probabilidade de ruína, Teoria da Ruína, Teoria do Risco

Abstract

The present Thesis will be developed, essentially, in the scope of Risk Theory, more specifically analyzing Ruin Theory and its several components. Therefore, the theoretical framework will focus on several concepts from which we can stand out the claims to be payed by the insurance company, incoming premiums and the initial reserve. The ruin probability will be studied in the Cramér-Lundberg model. In this context, based on Lundberg inequality, it is possible to obtain an upper bound for the ruin probability which depends on the adjustment coefficient.

With the purpose of estimating the adjustment coefficient, it was considered a geometric-type estimator existing in the literature, introduced in the context of the estimation of the exponential tail coefficient. We intend to prove that, under general conditions, the estimator can be used to estimate the adjustment coefficient and that it presents relevant proprieties such as consistency and asymptotic normality.

One of the goals of this work is also the implementation of simulation studies where the sample behavior of the geometric-type estimator will be analyzed, with the variation of certain parameters and basic assumptions.

Keywords: adjustment coefficient, Cramér-Lundberg model, exponential tail coefficient, geometric-type estimator, Hill estimator, Lundberg inequality, Risk Theory, ruin probability, Ruin Theory, simulation studies

Index

1 Introduction 1

1.1 Insurance and ruin probability . . . 1

1.2 Structure of the Thesis and main goals . . . 2

2 Risk Theory and ruin probability 4

2.1 Process of aggregate claims . . . 4

2.2 Process of the number of claims . . . 7

2.3 Ruin Theory . . . 7

2.3.1 Reserve process, ruin time and ruin probability . . . 7

2.3.2 Ruin probability in the Cramér-Lundberg model . . . 9

2.3.3 Relation between ruin probability and maximum aggregate loss 12

2.3.4 Lundberg inequality . . . 20

3 Estimation of the exponential tail coefficient 21

3.1 Introduction . . . 21

3.2 A geometric estimator for the exponential tail coefficient, $\widehat{R}$ . . . 25

3.3 Properties of the geometric estimator $\widehat{R}$ . . . 26

4 Geometric estimator for the adjustment coefficient 29

4.1 Introduction . . . 29

4.2 Estimating the adjustment coefficient . . . 30

4.3 Properties of the geometric estimator for the adjustment coefficient . . . 31

5 Simulation studies 37

5.1 Models for the individual claims . . . 38

5.1.1 Exponential model . . . 38

5.1.2 Degenerate model . . . 39

5.1.3 Uniform model . . . 39

5.1.4 Normal model . . . 40

5.2 Simulation procedure . . . 41

5.2.1 Exponential model 41
5.2.2 Non-exponential models 42
5.2.3 Empirical method for the choice of k_n 42
5.3 Simulation results 43
5.3.1 Exponential model 45
5.3.2 Degenerate model 54
5.3.3 Uniform model 58
5.3.4 Normal model 62
5.4 Analysis of the results 66

6 Conclusions **67**

Bibliography **69**

Annex **71**

A *R* code of the simulation studies **71**
A.1 Exponential model 71
A.2 Degenerate model 82
A.3 Uniform model 94
A.4 Normal model 106

List of Tables

5.1 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - exponential model 45
5.2 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - exponential model 45
5.3 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - exponential model 46
5.4 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - exponential model 46
5.5 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 200$ - exponential model 47
5.6 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 200$ and $l = 5$ - exponential model 47
5.7 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - exponential model 48
5.8 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - exponential model 48
5.9 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - exponential model 49
5.10 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - exponential model 49
5.11 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 200$ - exponential model 50
5.12 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 200$ and $l = 5$ - exponential model 50
5.13 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - exponential model 51
5.14 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - exponential model 51

5.15 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - exponential model 52

5.16 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - exponential model 52

5.17 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 200$ - exponential model 53

5.18 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 200$ and $l = 5$ - exponential model 53

5.19 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - degenerate model 54

5.20 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - degenerate model 54

5.21 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - degenerate model 55

5.22 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - degenerate model 55

5.23 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - degenerate model 56

5.24 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - degenerate model 56

5.25 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - degenerate model 57

5.26 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - degenerate model 57

5.27 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - uniform model 58

5.28 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - uniform model 58

5.29 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - uniform model 59

5.30 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - uniform model 59

5.31 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - uniform model 60

5.32 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - uniform model 60

5.33 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - uniform model 61

5.34 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - uniform model 61

5.35 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - normal model 62

5.36 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - normal model 62

5.37 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - normal model 63

5.38 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - normal model 63

5.39 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - normal model 64

5.40 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - normal model 64

5.41 Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - normal model 65

5.42 Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - normal model 65

List of Figures

2.1 Evolution of $N(t)$ 5

2.2 Evolution of $S(t)$ 6

2.3 Evolution of $U(t)$ 8

2.4 Maximum aggregate loss L as the sum of records L_i 17

3.1 Distance between the point (z_i, y_i) and the point of a line with ordinate y_i 23

3.2 Distance between the point (z_i, y_i) and the point of a line with abscissa z_i 24

3.3 Areas of the rectangles to be minimized 25

5.1 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 5.8333 \times 10^{-5}$ (full line) - exponential model 45

5.2 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 5.8333 \times 10^{-5}$ (full line) - exponential model 46

5.3 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 120$, obtained from 1000 Z samples of size $n = 200$, where $R = 5.8333 \times 10^{-5}$ (full line) - exponential model 47

5.4 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 0.5$ (full line) - exponential model 48

5.5 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 0.5$ (full line) - exponential model 49

5.6 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 120$, obtained from 1000 Z samples of size $n = 200$, where $R = 0.5$ (full line) - exponential model 50

5.7 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 333.33$ (full line) - exponential model 51
5.8 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 333.33$ (full line) - exponential model 52
5.9 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 120$, obtained from 1000 Z samples of size $n = 200$, where $R = 333.33$ (full line) - exponential model 53
5.10 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 5.8 \times 10^{-5}$ (full line) - degenerate model 54
5.11 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 5.8 \times 10^{-5}$ (full line) - degenerate model 55
5.12 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 0.5$ (full line) - degenerate model 56
5.13 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 0.5$ (full line) - degenerate model 57
5.14 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 5.8 \times 10^{-5}$ (full line) - uniform model 58
5.15 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 5.8 \times 10^{-5}$ (full line) - uniform model 59
5.16 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 0.5$ (full line) - uniform model 60
5.17 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 0.5$ (full line) - uniform model 61
5.18 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 5.8 \times 10^{-5}$ (full line) - normal model 62

5.19 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 5.8 \times 10^{-5}$ (full line) - normal model 63

5.20 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 0.5$ (full line) - normal model 64

5.21 Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 0.5$ (full line) - normal model 65

Chapter 1

Introduction

1.1 Insurance and ruin probability

All individuals have expectations regarding life events and make plans, however, in lots of cases, the outcome is not what was expected and expectations do not materialize. In certain occasions plans are frustrated because they are based on unrealistic assumptions, in other situations fortuitous circumstances intervene. An insurance system is a mechanism to reduce the negative financial impact of random events that prevent the fulfillment of reasonable expectations. Naturally, a system of this kind has basic limitations, starting with the fact that it can only reduce the effects of random events whose consequences can be measured in monetary terms; other kind of losses can be relevant, however they can't be softened through insurance.

An insurance system, *per si*, has an economical justification and contributes to the general well-being, improving the perspectives of a certain plan not to be frustrated due to a certain random event. A system of this kind can also promote a production increase, since it encourages the development of riskier projects whose possibility of great losses would inhibit its implementation. The development of insurance related with maritime activity, reducing the financial impact of the dangers of the sea, is a perfect example of the situation previously mentioned.

Uncertainty about the future and even the adversities of life always preoccupied man, which led him to seek protection against the dangers to his family and property. There are many examples throughout history, such as the Mesopotamian and Phoenician traders who, in order to reduce the financial impact of the dangers of the sea, have created a system for the replacement of cargoes of ships in case of theft or shipwreck in their travels in the Mediterranean and Aegean seas. Also the Hebrews, concerned about their flocks, agreed that each member of the herding group who lost one animal was guaranteed to receive another animal, paid by the other shepherds. In the middle of the fourteenth century, with the growth of urban settlements, the first maritime insurance contracts with policy issuance, created by Italians and Spaniards, came about due to the great expansion of maritime trade.

Insurance theory is therefore based on the assumption that individuals, faced with a possible large loss, can reduce their financial effects by forming a group that shares it. This way, the vast majority of people prefer to make a commitment for the periodic payment of a fixed and reasonable amount of money (the premium) then having the risk of possibly facing a very large loss, even if it is unlikely. Insurance is a mechanism that transfers the risk of individuals, institutions, companies or organizations (insured) to insurance companies (insurers).

This raises the question of how is it possible for insurance companies to take on a large number of external risks without jeopardizing their solvency. The survival of insurers is due, on the one hand, to the fact that in reality few insureds report large claims, and secondly, premiums are calculated in order to safeguard the insurer's ability to cover the claims requested. Thus, it appears naturally in the theory about insurance activity, the need to deal with uncertainty about the occurrence of phenomena that could cause loss. Summing up, there are four main factors to be taken into account in this context: whether there are claims in the period in question, the moment of the occurrence of the claim, the claims paid and finally the amount of premiums to be charged to all policyholders in order to ensure the solvency and profitability of the insurer.

The insurer has to continuously assess the ability to remain in business by ensuring that the amount of premiums collected (to which the initial reserve is added) is sufficient to cover the claims to be paid. The ruin probability is the possibility of the collected reserves being exhausted at a certain moment of time leading, in short, the insurer to insolvency. Thus, the insurer must analyze its ruin probability and adjust the value of the initial reserve and also the incoming premiums, ensuring continuity.

1.2 Structure of the Thesis and main goals

This Thesis, in general, will be developed in the insurance context and in particular considering Risk Theory. With that purpose, a review of the relevant literature will be carried out in order to build the fundamental theoretical basis. Thus, Ruin Theory will be analyzed with its components, highlighting the claims to be paid by the insurance company, the incoming premiums and the initial reserve. Likewise, emphasis will be placed on the study of the probability of ruin under the Cramér-Lundberg model. In this context it is of particular relevance the adjustment coefficient that allows, through Lundberg inequality, to obtain an upper bound for the ruin probability of the insurance company.

In order to estimate the adjustment coefficient we will present a geometric-type estimator existing in the literature, which was introduced in the context of the estimation of the exponential tail coefficient.

One of the objectives of this work will be to show that this estimator, under general conditions, can be applied to the estimation of the adjustment coefficient and, as an estimator of this coefficient, it is consistent and asymptotically normal.

Another objective will be to carry out simulation studies, in order to analyze the finite sample behavior of the geometric estimator, by varying certain parameters and basic assumptions. The *R* software code (built entirely from scratch) used in the simulation studies is presented in the annex.

The contents of each chapter of this Thesis are summarized below. In Chapter 2, we expose some relevant theoretical concepts of Risk Theory and more specifically of Ruin Theory. We examine the process of aggregate claims in this chapter, including the process of the number of claims and the distribution of the amount of individual claims. In the scope of the Ruin Theory, we study the Cramér-Lundberg model, the adjustment coefficient and the ruin probability, as well as its relation with the maximum aggregate loss. At the end of the chapter, we present the well-known Lundberg inequality. In Chapter 3, we analyze the problem of the estimation of the exponential tail coefficient, and present, in particular, a geometric-type estimator and its properties. In Chapter 4 it will be noted that the adjustment coefficient previously defined can be estimated by the geometric estimator considered in the previous chapter. In particular, it will be demonstrated, in the context of the estimation of the adjustment coefficient, the consistency and the asymptotic normality of the geometric estimator. In Chapter 5 we perform simulation studies in order to analyze the behavior for finite samples of the geometric and Hill estimators, as estimators of the adjustment coefficient. In this chapter we will present all the assumptions considered, the adopted procedures, the results obtained and their comments. Finally, in Chapter 6, we will present the main conclusions of the work.

Chapter 2

Risk Theory and ruin probability

Risk Theory dates back to the end of the seventeenth century and its main objective is to study the mathematical and probabilistic models that best fit the insurance business (see, for example, Mikosch (2009)). In the next section we will describe the models of aggregate claims, which correspond to the total amount paid in claims by an insurance company over a certain period of time.

2.1 Process of aggregate claims

We begin by presenting the so-called collective risk model that adds the claims paid in the moments they occur (see, for example, Centeno (2003) and Fraga Alves (1997)).

Throughout the work we will represent the **process of aggregate claims** by $\{S(t)\}_{t\geqslant 0}$, where $S(t)$ stands for the total amount paid in claims in the $]0,t]$ time interval.

$$S(t)=\sum_{i=1}^{N(t)} X_i,$$

where X_i, with $i=1,2,\ldots,N(t)$, corresponds to the amount of the i-th claim that occurred in the $]0,t]$ interval. In case $N(t)=0$, we consider $S(t)=0$. Let's assume $\{S(t)\}_{t\geqslant 0}$ is a compound stochastic process where:

- $\{N(t)\}_{t\geqslant 0}$ is a process of integer values that counts the number of claims paid up to the moment t;

- $\{X_i\}_{i=1,2,\ldots,N(t)}$ is a sequence of independent random variables and identically distributed to X, with a common distribution function F_X, and independent of $N(t)$.

Let's assume that $t \geqslant 0$ and $h > 0$. For a given instant t, we consider that:

- $N(t+h)-N(t)$ is the number of claims paid between t and $t+h$;

- $S(t+h)-S(t)$ is the amount of aggregate claims paid between t and $t+h$.

Let us consider that:

- V_i - instant of the i-th claim $X_i, i=1,2,\ldots$. We assume that $V_1, V_2, \ldots$ are random variables such that $V_1 < V_2 < \ldots$, to exclude the possibility that two (or more) claims occur in the same instant;
- T_i - time elapsed between two successive claims X_{i-1} and X_i, $i=1,2,\ldots$. Let's consider that $T_1 = V_1$.

Therefore,

$$T_1 = V_1, \quad T_i = V_i - V_{i-1}, i = 2,3,\ldots,$$

and

$$V_i = \sum_{j=1}^{i} T_j,\ i = 1,2,\ldots$$

The usual representation of the process of the number of claims and the amount of aggregate claims can be seen in the figures below. It should be noted that both $N(t)$ and $S(t)$ are step functions, whose discontinuities occur in the moments $V_i, i=1,2,\ldots$, when the claims are paid. Each jump size is equal to 1 in the $N(t)$ case, and equal to the amount of the claim $X_i, i=1,2,\ldots$, in the $S(t)$ case.

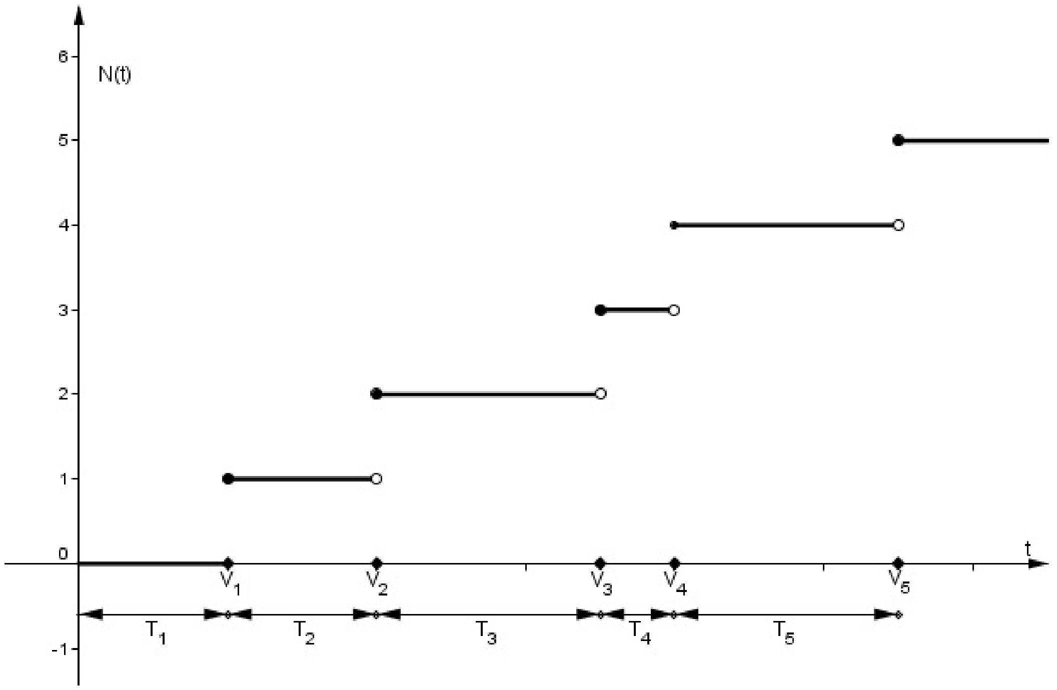

Figure 2.1: Evolution of $N(t)$

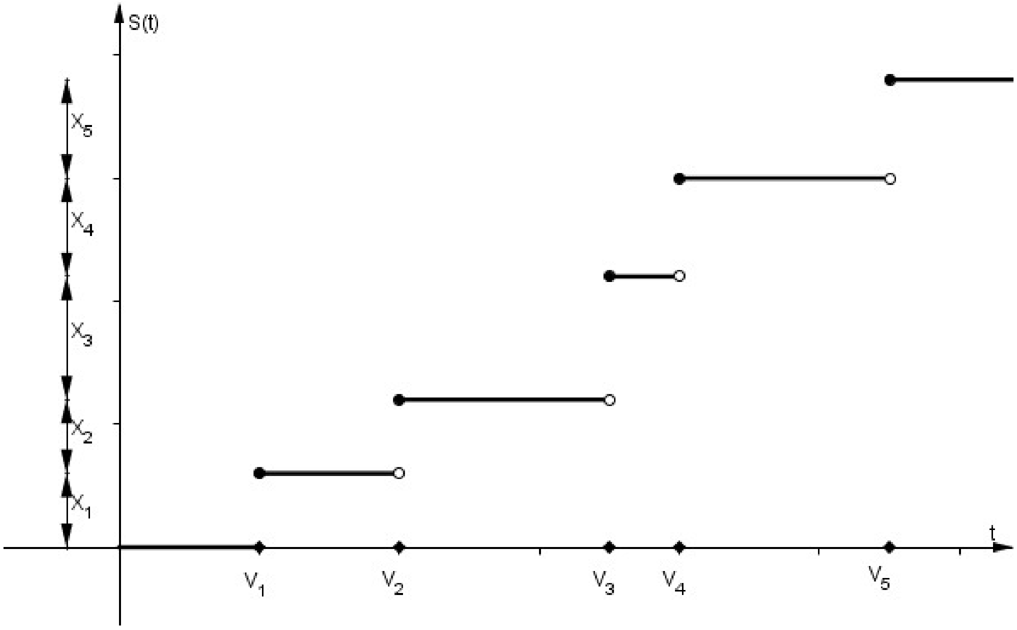

Figure 2.2: Evolution of $S(t)$

The expressions for the expected value, variance, moment generating function and distribution function of $S(t)$ as functions of the expected values, variances, moment generating functions and distribution functions of X and $N(t)$, are shown below.

$$E(S(t)) = E(X)E(N(t)) \tag{2.1.1}$$

$$V(S(t)) = E(N(t))V(X) + V(N(t))E(X)^2 \tag{2.1.2}$$

$$M_{S(t)}(r) = M_{N(t)}(\log M_X(r)) \tag{2.1.3}$$

$$F_{S(t)}(x) = \sum_{n=0}^{+\infty} F_X^{n\star}(x) P(N(t) = n), \tag{2.1.4}$$

with $F_X^{0\star}(x) = 1\ \forall x \geqslant 0$.

2.2 Process of the number of claims

In this Thesis, we will analyze the case where the process of the aggregate claims $\{S(t)\}_{t\geqslant 0}$, considered in Section 2.1, follows a compound Poisson process, that is, where the process of the number of claims, $\{N(t)\}_{t\geq 0}$, follows a Poisson process of intensity equal to λ. In this case, we can state that the times elapsed between successive damages, $T_1, T_2, \ldots$ (defined in Section 2.1, that is, where T_1 represents the instant of the first claim and T_i the time elapsed between claim $i-1$ and claim $i, i=2,3,\ldots$) are independent and identically distributed random variables following an exponential distribution with mean $\frac{1}{\lambda}$ (see Beard (1984)).

In this context, and taking into account generic expressions (2.1.1), (2.1.2), (2.1.3) and (2.1.4), we can write the following equalities regarding the compound Poisson process of aggregate claims, $\{S(t)\}_{t\geqslant 0}$:

$$\begin{aligned} E(S(t)) &= \lambda t E(X); \\ V(S(t)) &= \lambda t E(X^2); \\ M_{S(t)}(r) &= e^{\lambda t(M_X(r)-1)}; \\ F_{S(t)}(x) &= \sum_{n=0}^{+\infty} F_X^{n\star}(x)\frac{e^{-\lambda t}(\lambda t)^n}{n!}. \end{aligned}$$

2.3 Ruin Theory

2.3.1 Reserve process, ruin time and ruin probability

Let $\{S(t)\}_{t\geqslant 0}$ be the process of the aggregate claims of a certain insurance company studied in Section 2.1. The notation introduced there will be maintained throughout the text.

The **reserve process** of the insurer will be represented by $\{U(t)\}_{t\geqslant 0}$. Denoting the initial reserve $U(0)$ by u and assuming that the premiums are received continuously at a constant rate $c>0$, that is, the total amount of premiums received between 0 and t is equal to ct, we will consider that, in moment t, the risk reserve is

$$U(t) = u + ct - S(t), \ t \geqslant 0. \tag{2.3.1}$$

If the reserve becomes negative, at some instant, ruin occurs. The moment of the occurrence of ruin is denoted by

$$\mathscr{T} = \inf\{t : t \geqslant 0 \text{ and } U(t) < 0\},$$

with the assumption that $\mathscr{T} = \infty$ if $U(t) \geqslant 0$ for all t. This way, the **ruin probability** is given by

$$\psi(u) = P(\mathscr{T} < \infty).$$

The usual representation of the reserve process is illustrated in Figure 2.3: the premium rate is constant over time, so the graph has a constant slope and has linear growth, except in the V_i moments, when a claim occurs. At that moment, the value of the reserve decreases by the amount corresponding to the claim X_i. Ruin occurs when the reserve becomes negative. In that case, the addition of the initial reserve with the total premiums collected is not sufficient to cover the claims to be paid.

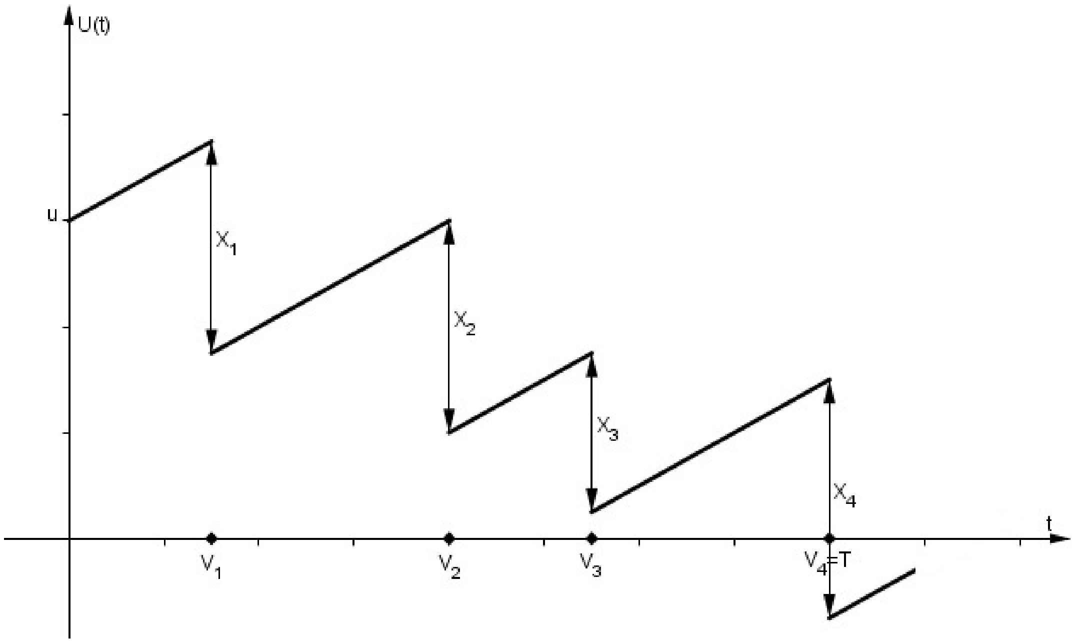

Figure 2.3: Evolution of $U(t)$

2.3.2 Ruin probability in the Cramér-Lundberg model

The reserve process $\{U(t)\}_{t\geqslant 0}$ can be studied through its relation with the process $\{S(t)\}_{t\geqslant 0}$, given by the equality (2.3.1).

In the case where $\{S(t)\}_{t\geqslant 0}$ follows a compound Poisson process, the model of the reserve process is called **Cramér-Lundberg model**. From now on we will focus on this specific case.

The notations used in Sections 2.1 and 2.2 will be maintained.

Let's admit that c exceeds the expected value of the claims paid in a period, that is $c > \lambda E(X)$. This means that, in average, per unit of time, the amount received in premiums is higher that the amount paid in claims.

The **safety coefficient** $\theta > 0$ can be defined through the following equation

$$c = (1+\theta)\lambda E(X).$$

We denote by η the supremum of the values of ν for which the moment generating function of X, $M_X(r)$, exists in the interval $]-\infty, \nu[$.

Let us suppose that $\eta > 0$ and $M_X(r)$ tends to ∞ as $r \to \eta$, $\lim_{r\to\eta} M_X(r) = \infty$.

We define now the **adjustment coefficient** as the only positive root $r = R$ of the equation

$$\lambda + rc = \lambda M_X(r), \ r < \eta, \tag{2.3.2}$$

which is equivalent to

$$1 + (1+\theta)E(X)r = M_X(r), \ r < \eta. \tag{2.3.3}$$

Proposition 2.3.1. *Under the above conditions, equation (2.3.3) has only one positive root.*

Example 2.3.1. *Computation of the adjustment coefficient in the case where the distribution of the individual claims is exponential with mean equal to β.*

$$\textit{If}\, X \sim Exp\left(\frac{1}{\beta}\right),\ E(X) = \beta \textit{ and } M_X(r) = \frac{\frac{1}{\beta}}{\frac{1}{\beta} - r},\ r < \frac{1}{\beta}.$$

Replacing in equation (2.3.3) the values of $E(X)$ and $M_X(r)$, we will have:

$$1+(1+\theta)\beta r = \frac{\frac{1}{\beta}}{\frac{1}{\beta}-r}$$

which is equivalent to

$$r\left((1+\theta)r-\frac{\theta}{\beta}\right)=0.$$

The roots of the previous equation are zero and the adjustment coefficient

$$R=\frac{\frac{\theta}{\beta}}{1+\theta}. \tag{2.3.4}$$

Considering that $c=(1+\theta)\lambda E(X)$ and since in this case $E(X)=\beta$, we can write R in the following way:

$$R=\frac{1}{\beta}-\frac{\lambda}{c}.$$

The **Fundamental Risk Theorem** will be presented and demonstrated next.

Theorem 2.3.1. *For $u \geqslant 0$*

$$\psi(u)=\frac{e^{-Ru}}{E(e^{-RU(\mathscr{T})}|\mathscr{T}<\infty)}, \tag{2.3.5}$$

where R denotes the adjustment coefficient.

Proof. For any $t>0$ and $r>0$,

$$E(e^{-rU(t)})=E(e^{-rU(t)}|\mathscr{T}<t)P(\mathscr{T}<t)+E(e^{-rU(t)}|\mathscr{T}\geqslant t)P(\mathscr{T}\geqslant t). \tag{2.3.6}$$

Since $U(t)=u+ct-S(t)$, and recalling the expression of the moment generating function of the aggregate claims in the case of a Poisson process of intensity λ, $E(e^{rS(t)})=e^{\lambda t(M_X(r)-1)}$ (see Section 2.2), we obtain that

$$E(e^{-rU(t)})=e^{-ru-rct}E(e^{rS(t)})=e^{-ru-rct+\lambda t(M_X(r)-1)}. \tag{2.3.7}$$

For $t > \mathscr{T}$, we can write

$$U(t) = U(\mathscr{T}) + c(t - \mathscr{T}) - (S(t) - S(\mathscr{T})).$$

Once $\{S(t)\}_{t \geqslant 0}$ has independent increments, we know that $U(\mathscr{T})$ is independent of $S(t) - S(\mathscr{T})$. Besides that, this difference follows a compound Poisson distribution with parameter $\lambda(t - \mathscr{T})$. Therefore,

$$E(e^{-rU(t)}|\mathscr{T} < t) = E(e^{-rU(\mathscr{T}) - rc(t-\mathscr{T}) + \lambda(t-\mathscr{T})(M_X(r)-1)}|\mathscr{T} < t). \tag{2.3.8}$$

Considering the case where r is the adjustment coefficient (R), the expressions (2.3.7) and (2.3.8) can be simplified obtaining the equality (2.3.6)

$$e^{-Ru} = E(e^{-RU(\mathscr{T})}|\mathscr{T} < t)P(\mathscr{T} < t) + E(e^{-RU(t)}|\mathscr{T} \geqslant t)P(\mathscr{T} \geqslant t). \tag{2.3.9}$$

Letting $t \to \infty$ in equation (2.3.9) the second addend of the second member tends to 0, because if ruin hasn't occurred then $\lim_{t\to\infty} U(t) = \infty$ (the proof of this can be found in Bowers et al. (1986), chapter 13). Consequently

$$e^{-Ru} = E(e^{RU(\mathscr{T})}|\mathscr{T} < \infty)\psi(u),$$

that is

$$\psi(u) = \frac{e^{-Ru}}{E(e^{-RU(\mathscr{T})}|\mathscr{T} < \infty)}.$$

□

Example 2.3.2. *In this example we will calculate the ruin probability considering that individual claims have exponential distribution of mean β.*

Let's admit that $\mathscr{T}$ is the instant of ruin. The reserve immediately before $\mathscr{T}$ is $\hat{u}$ and we will assume that $y > 0$. Taking into account that the event $-U(\mathscr{T}) > y$ given $\mathscr{T} < \infty$ is equivalent to $X > \hat{u} + y$ given $X > \hat{u}$ (where X is the claim that caused the ruin), then

$$
\begin{aligned}
P(-U(\mathcal{T})>y|\mathcal{T}<\infty) &= P(X>\hat{u}+y|X>\hat{u})\\
&= \frac{1-F_X(\hat{u}+y)}{1-F_X(\hat{u})}\\
&= \frac{e^{-\frac{(\hat{u}+y)}{\beta}}}{e^{-\frac{\hat{u}}{\beta}}}\\
&= e^{-\frac{y}{\beta}},\ y>0,
\end{aligned}
$$

and consequently

$$P(-U(\mathcal{T})<y|\mathcal{T}<\infty)=1-e^{-\frac{y}{\beta}},\ y>0.$$

Therefore

$$E(e^{-RU(\mathcal{T})}|\mathcal{T}<\infty)=\frac{\frac{1}{\beta}}{\frac{1}{\beta}-R}.$$

Taking into account the equality (2.3.4) of the Example 2.3.1, and also the Fundamental Risk Theorem, we obtain

$$\psi(u)=\frac{\frac{1}{\beta}-R}{\frac{1}{\beta}}e^{-\frac{\theta u}{\beta(1+\theta)}}=\frac{1}{1+\theta}e^{-\frac{\theta u}{\beta(1+\theta)}}.$$

Since $c=(1+\theta)\lambda E(X)$ and in this case $E(X)=\beta$, we can rewrite $\psi(u)$ in the following way:

$$\psi(u)=\frac{\beta\lambda}{c}e^{-\left(\frac{1}{\beta}-\frac{\lambda}{c}\right)u}.$$

Although we have just presented an example of application of the Theorem 2.3.1, we note that, in most cases, it is not possible to determine the denominator of its expression. Indeed, the great utility of the Fundamental Risk Theorem resides in its use for the achievement of inequalities of great relevance, such as those that will be presented in Section 2.3.4.

2.3.3 Relation between ruin probability and maximum aggregate loss

In this section it will be presented the relation between ruin probability and maximum aggregate loss with the purpose of finding, for certain particular distributions, explicit expressions for ruin probability.

It is important to remember that we are analyzing the specific case where the aggregate claims process $\{S(t)\}_{t\geqslant 0}$ is a compound Poisson process.

We will focus now on the moment when the reserve amount falls below its initial level u (notice that this situation may possibly never occur).

As previously mentioned $c = (1+\theta)\lambda E(X)$, where c refers to the premiums entrance rate, θ to the security coefficient and λ to the intensity of the counting process $\{N(t)\}_{t\geqslant 0}$.

Theorem 2.3.2. *The probability of the reserve ever falling below the initial value u, and take a value between $u-y$ and $u-y-dy$ in the first time this occurs, is given by*

$$\frac{\lambda}{c}[1-F_X(y)]\,dy = \frac{1-F_X(y)}{(1+\theta)E(X)}dy,\ y>0.$$

The demonstration of the previous result can be found in Bowers et al. (1986).

As an application of the previous theorem we have the following result.

Corollary 2.3.1. *The probability of reserves ever being lower than their initial level, u, is $\frac{1}{1+\theta}$.*

Proof. By Theorem 2.3.2 we have

$$P(U(t)<u \wedge u-y-dy<U(t)<u-y) = \frac{\lambda}{c}(1-F_X(y))\,dy$$

for $y>0$, so

$$P(U(t)<u) = \int_0^{+\infty}\frac{\lambda}{c}(1-F_X(y))\,dy = \frac{\lambda}{c}\int_0^{+\infty}(1-F_X(y))\,dy.$$

Since X is a non-negative random variable, we obtain $E(X) = \int_0^{+\infty}(1-F_X(y))\,dy$. Accordingly

$$P(U(t)<u) = \frac{\lambda E(X)}{c} = \frac{\lambda E(X)}{(1+\theta)\lambda E(X)} = \frac{1}{1+\theta}.$$

□

Corollary 2.3.1 allows us to conclude that the probability of the reserve ever falling below its initial level does not depend on this value, u, but only on the security coefficient θ.

In the particular case where $u=0$, the probability of the reserve falling below this initial level coincides with the ruin probability when the initial level is zero. Therefore,

$$\psi(0) = \frac{1}{1+\theta}.$$

It is important to notice that $\psi(0)$ depends only on the security coefficient and not on the distribution of the individual claims.

It will be denoted by L_1 the random variable that corresponds to the "amount by which reserves fall below their initial level, in the first time this occurs, in case that happens".

Corollary 2.3.2. *The probability density function f_{L_1} of the random variable L_1 is*

$$f_{L_1}(y) = \frac{1}{E(X)}[1 - F_X(y)],\ y > 0.$$

Proof. Let's start by noting that $L_1 \equiv u - U(t)|\{U(t) < u\}$. Thus, and as a consequence of the Theorem 2.3.2 and Corollary 2.3.1,

$$\begin{aligned} f_{L_1}(y)dy &= P(y < u - U(t) < y + dy|U(t) < u) \\ &= P(u - y - dy < U(t) < u - y|U(t) < u) \\ &= \frac{P(U(t) < u \wedge u - y - dy < U(t) < u - y)}{P(U(t) < u)} \\ &= \frac{\frac{\lambda}{c}(1 - F_X(y))dy}{\frac{1}{1+\theta}} = \frac{(1+\theta)\lambda}{(1+\theta)\lambda E(X)}(1 - F_X(y))dy \\ &= \frac{1}{E(X)}(1 - F_X(y))dy. \end{aligned}$$

So, the probability density function of L_1 is given by

$$f_{L_1}(y) = \frac{1}{E(X)}(1 - F_X(y)),\ y > 0.$$

□

Corollary 2.3.3. *The moment generating function of L_1, $M_{L_1}(r)$, and the moment generating function of X, $M_X(r)$, verify the following relation*

$$M_{L_1}(r) = \frac{1}{E(X)r}[M_X(r) - 1].$$

Proof. The moment generating function of the random variable L_1 is given by

$$\begin{aligned} M_{L_1}(r) = E(e^{rL_1}) &= \frac{1}{E(X)} \int_0^{+\infty} e^{ry}(1 - F_X(y))\,dy \\ &= \frac{1}{E(X)r} \left\{ e^{ry}(1 - F_X(y))|_0^{+\infty} + \int_0^{+\infty} e^{ry} f_X(y) dy \right\} \\ &= \frac{1}{E(X)r}(M_X(r) - 1). \end{aligned}$$

□

The **maximum aggregate loss** will be the random variable

$$L = \max_{t \geqslant 0} \{S(t) - ct\},$$

that corresponds to the maximum excess of aggregate claims over the premiums collected. Since $S(t) - ct = 0$ for $t = 0$, the random variable L is a non-negative variable, that is, $L \geqslant 0$.

It is possible to deduce the following relation between the distribution of the maximum aggregate loss and ruin probability.

Theorem 2.3.3. *Considering an initial reserve $u \geqslant 0$, the distribution function of the random variable L and ruin probability $\psi(u)$ verify the equality*

$$\psi(u) = 1 - F_L(u).$$

Proof.

$$\begin{aligned} F_L(u) &= P(L \leqslant u) = P(S(t) - ct \leqslant u, \forall t \geqslant 0) \\ &= P(u + ct - S(t) \geqslant 0, \forall t \geqslant 0) = P(U(t) \geqslant 0, \forall t \geqslant 0) \\ &= 1 - \psi(u), \end{aligned}$$

considering an initial reserve $u \geqslant 0$. □

The ruin probability, $\psi(u)$, is a decreasing function of the initial level u, starting from $\psi(u) = \frac{1}{1+\theta}$.

Corollary 2.3.4. *L is a mixed random variable, having probability mass point in $L = 0$, with*

$$P(L=0)=\frac{\theta}{1+\theta}.$$

Proof. By the previous theorem,

$$P(L=0)=P(L\leqslant 0)=F_L(0)=1-\psi(0)=1-\frac{1}{1+\theta}=\frac{\theta}{1+\theta}.$$

□

The following theorem is the most important of this section, presenting an explicit form for the moment generating function of L that, considering Theorem 2.3.3, can be used to obtain information about $\psi(u)$.

Theorem 2.3.4. *The moment generating function of the random variable L is*

$$M_L(r)=\frac{\theta E(X)r}{1+(1+\theta)E(X)r-M_X(r)}.$$

Proof. Consider the instants where the process of aggregate loss, $S(t)-ct$, hits new records.

After the occurrence of a record value, the probability of exceeding it is equal to $\psi(0)$ while the probability of not exceeding it is equal to $1-\psi(0)$. This finding is based on the fact that $\{S(t)\}_{t\geqslant 0}$ is a compound Poisson process and has stationary and independent increments.

Thus, the occurrence of a record value for the aggregate loss process is equally likely to the occurrence of a descent below the level from which the reserve process starts, that is, with a probability equal to $\psi(0)=\frac{1}{1+\theta}$.

In Figure 2.4 we can see that L can be represented through a sum of a random number of random variables referring to the amounts by which the records are exceeded:

$$L=\sum_{i=1}^{V}L_i, \tag{2.3.10}$$

where V is the "number of times aggregate loss breaks a record". So

$$P(V=n)=(1-\psi(0))(\psi(0))^n=\left(\frac{\theta}{1+\theta}\right)\left(\frac{1}{1+\theta}\right)^n,\ n=0,1,2,\ldots$$

V is then a random variable with geometric distribution,

$$V\sim Geo\left(\frac{\theta}{1+\theta}\right),$$

being its moment generating function given by

$$\begin{aligned}M_V(r) = E(e^{rV}) &= \sum_{n=0}^{+\infty} e^{rn}\left(\frac{\theta}{1+\theta}\right)\left(\frac{1}{1+\theta}\right)^n \\ &= \sum_{n=0}^{+\infty}\left(\frac{\theta}{1+\theta}\right)\left(\frac{1}{1+\theta}e^r\right)^n \\ &= \frac{\frac{\theta}{1+\theta}}{1-\frac{e^r}{1+\theta}} \\ &= \frac{\theta}{1+\theta-e^r}.\end{aligned}$$

Therefore, we have that

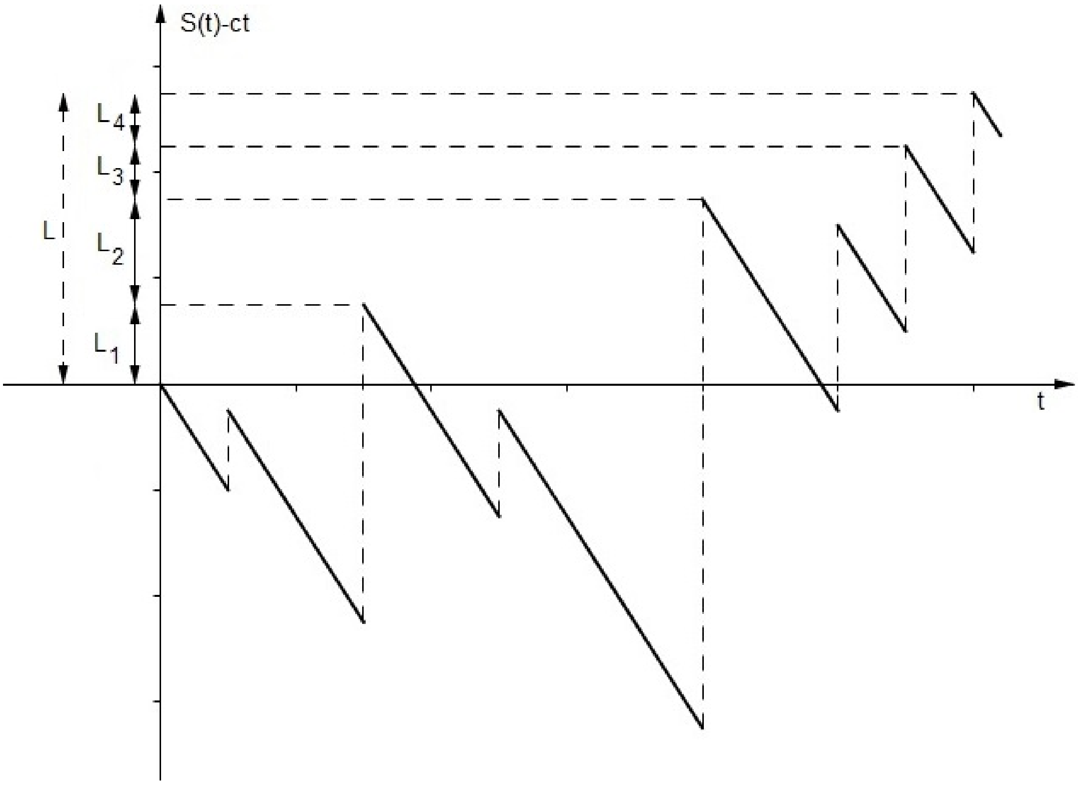

Figure 2.4: Maximum aggregate loss L as the sum of records L_i

- L_i are independent random variables and identically distributed to L_1, with probability density function;

$$f_{L_1}(y) = \frac{1}{E(X)}(1-F_X(y)),\ y \geqslant 0;$$

- L_i are independent of V.

Using a similar idea as the one applied in the proof of the expression 2.1.3 we obtain that the moment generating function of L is given by

$$M_L(r) = M_V(\log(M_{L_1}(r))) = \frac{\theta}{1+\theta - M_{L_1}(r)}.$$

Replacing $M_{L_1}(r)$ by the expression obtained in Corollary 2.3.3, we get

$$M_L(r) = \frac{\theta E(X) r}{1 + (1+\theta)E(X)r - M_X(r)}.$$

□

Theorem 2.3.5. *The following equality is verified*

$$\int_0^{+\infty} e^{ur}(-\psi'(u))du = \frac{1}{1+\theta} \times \frac{\theta(M_X(r)-1)}{1+(1+\theta)E(X)r - M_X(r)}. \qquad (2.3.11)$$

Proof. Let's start by noting that

$$\begin{aligned}
M_L(r) &= P(L=0) \times E(e^{rL}|L=0) + P(L>0) \times E(e^{rL}|L>0) \\
&= P(L=0) \times E(e^0) + P(L>0) \times \int_0^{+\infty} e^{ru} \frac{f_L(u)}{P(L>0)} du \\
&= \frac{\theta}{1+\theta} + \int_0^{+\infty} e^{ru} f_L(u) du.
\end{aligned}$$

Noting now that the probability function of L is given by

$$f_L(u) = \begin{cases} P(L=0) = \frac{\theta}{1+\theta}, & u=0 \\ -\psi'(u), & u>0 \end{cases},$$

then,

$$M_L(r) = \frac{\theta}{1+\theta} + \int_0^{+\infty} e^{ru}(-\psi'(u))du.$$

Based on the expression of the Theorem 2.3.4, we can write the moment generating function of L in the following way:

$$M_L(r) = \frac{\theta}{1+\theta} + \frac{1}{1+\theta} \times \frac{\theta(M_X(r)-1)}{1+(1+\theta)E(X)r - M_X(r)}. \qquad (2.3.12)$$

The desired result is then obtained by comparing the last two equalities:

$$\int_0^{+\infty} e^{ru}(-\psi'(u))du = \frac{1}{1+\theta} \times \frac{\theta(M_X(r)-1)}{1+(1+\theta)E(X)r - M_X(r)}.$$

□

Formula (2.3.11) can be used to determine explicit expressions for $\psi(u)$, for certain particular distribution families. In fact, its usefulness is proven in cases where individual claims have exponential distribution, gamma or mixture of exponentials.

Particular case. We will now consider the case where X is a mixture of exponentials, that is,

$$f_X(x) = \sum_{i=1}^{n} A_i \beta_i e^{-\beta_i x},\ x > 0, \tag{2.3.13}$$

with $\beta_i > 0, A_i > 0, \forall i = 1, 2, \ldots, n,$ and $\sum_{i=1}^{n} A_i = 1$.

In this case,

$$M_X(r) = \sum_{i=1}^{n} A_i \frac{\beta_i}{\beta_i - r},$$

for $r < \min\{\beta_1, \beta_2, \ldots, \beta_n\}$.

Replacing this expression of $M_X(r)$ in equation (2.3.11), we obtain

$$\int_0^{+\infty} e^{ru}(-\psi'(u))du = \sum_{i=1}^{n} \frac{C_i r_i}{r_i - r}, \tag{2.3.14}$$

for constants C_i, and in which the r_i, $i = 1, 2, \ldots, n$ are the roots of the denominator of the second member.

Corollary 2.3.5. *Considering the case where X follows the mixture of exponentials defined in (2.3.13), with the previous notation, the function that satisfies simultaneously (2.3.14) and* $\psi(\infty) = 0$ *is*

$$\psi(u) = \sum_{i=1}^{n} C_i e^{-u r_i}. \tag{2.3.15}$$

Proof. Let's see that $\psi(u)$ defined as in (2.3.15) satisfies (2.3.14):

$$\begin{aligned}\int_0^{+\infty} e^{ur}(-\psi'(u))du &= \int_0^{+\infty} e^{ur}\left(\sum_{i=1}^{n} C_i r_i e^{-ur_i}\right)du \\ &= \sum_{i=1}^{n} C_i r_i \int_0^{+\infty} e^{(r-r_i)u}du = \sum_{i=1}^{n} \frac{C_i r_i}{r_i - r}.\end{aligned}$$

□

2.3.4 Lundberg inequality

As stated at the end of Section 2.3.2, the Fundamental Theorem of Risk (Theorem 2.3.1) allows to obtain very important inequalities, such as those presented below.

Corollary 2.3.6 (Lundberg inequality)**.**

$$\psi(u) < e^{-Ru}.$$

Proof. Just note that for $\mathscr{T} < \infty$, we have $U(\mathscr{T}) < 0$. Consequently, the denominator of (2.3.5) is greater than 1. □

Corollary 2.3.7. *If the individual claims are limited in such a way that $F_X(m) = 1$ for some finite m, then*

$$\psi(u) > e^{-R(u+m)}.$$

Proof. For $\mathscr{T} < \infty$, and considering that $F_X(m) = 1$, we have $U(\mathscr{T}) > -m$. So,

$$e^{-RU(\mathscr{T})} < e^{Rm},$$

and therefore

$$E(e^{-RU(\mathscr{T})}|\mathscr{T} < \infty) < e^{Rm}.$$

Combining this result with Theorem 2.3.1, we obtain the desired result.

□

Chapter 3

Estimation of the exponential tail coefficient

One of the most important problems in Risk Theory, and just discussed, is the study of the ruin probability of the insurance company, assuming that it has an initial reserve. Several studies have been devoted to that subject, aiming at both the exact calculation of the probabilities in particular models, and obtaining approximations and upper bounds to them.

From now on we will focus our attention on *Lundberg inequality* (Corollary 2.3.6), that reduces the problem of estimating an upper bound for the ruin probability to the estimation of the adjustment coefficient R. In order to estimate this coefficient we will present, in this chapter, the problem of the estimation of the exponential tail coefficient.

3.1 Introduction

In this section we begin by considering a sample $(Z_1,\ldots,Z_n)$ of i.i.d. r.v. (independent and identically distributed random variables), with d.f. (distribution function) F that satisfies:

$$1-F(z)=P[Z_1>z]=r(z)e^{-Rz}, \qquad z>0, \qquad (3.1.1)$$

where r is a function of regular variation in ∞, that is,

$$\forall z>0,\ \lim_{t\to\infty}\frac{r(tz)}{r(t)}=z^{\rho},\ \text{for some } \rho\in\mathbb{R}$$

and R is a positive constant, which is called the exponential tail coefficient.

Next we will also consider the following equivalent form to (3.1.1):

$$F^{-1}(1-s) = -\frac{1}{R}\log s + \log \tilde{L}(s), \qquad 0 < s < 1, \tag{3.1.2}$$

where $\tilde{L}$ is a slowly varying function at zero (see, for example, Schultze and Steinebach (1996), Lemma 2.2.a) and references therein), that is,

$$\forall y > 0, \ \lim_{t \to 0} \frac{L(ty)}{L(t)} = 1.$$

The problem that will be studied in this chapter is the estimation of the tail coefficient R. This problem has received special attention, since it has applications in a wide variety of fields, such as in hydrology, finance, insurance, telecommunications, geology and climatology. An overview of the existing literature is given in Csörgő and Viharos (1998). Following Csörgő and Steinebach (1991), we will consider, in a later chapter, an important application to risk theory, namely the estimation of the adjustment coefficient R.

We will begin by presenting three exponential tail coefficient estimators introduced by Schultze and Steinebach (1996).

These estimators were motivated by the fact that, being F a distribution function that verifies (3.1.1), $-\log(1-F(z))$, for a large z, is approximately linear with slope R, since $z^{-1}\log r(z) \to 0$ as $z \to \infty$. It is thus expected that $-\log(1-F_n(z))$ is also approximately linear for high values of n and z, where F_n stands for the e.d.f. (empirical distribution function) associated to the r.s. (random sample) $(Z_1, \ldots, Z_n)$.

To simplify the study, Schultze and Steinebach initially assumed $r(z) \equiv c$, $\forall z > 0$. Therefore,

$$y := -\log(1-F(z)) = Rz - \log c = Rz - d,$$

or equivalently,

$$z = R^{-1}(y+d) = ay + b,$$

where $a = R^{-1}, b = R^{-1}d \ and \ d = \log c.$

It is thus expected that those linear relations are, approximately, verified for the k_n largest observations of the sample $(Z_1, \ldots, Z_n)$, denoted by $z_i := z_{n-i+1,n}$, $i = 1, \ldots, k_n \leq n$. Schultze and Steinebach approximated $-\log(1-F(z_i))$ by $y_i := -\log(1-F_n(z_i^-)) = -\log(1-(n-i)/n) = \log(n/i)$. This way we can verify that y_i is near $Rz_i - d$, or equivalently, z_i is close to $ay_i + b$.

An estimator of a was obtained by minimizing the function $f_1(a,b) = \sum_{i=1}^{k_n}(z_i - ay_i - b)^2$. Considering Figure 3.1, the problem of estimating a corresponds to the computation of the inverse of the slope of the line that minimizes the sum of the squares of the distances between the points (z_i, y_i) and the points of a line with ordinate y_i, respectively, $i = 1, \ldots, k_n$ (that is, computing the inverse of the slope of the line that minimizes the sum of the squares of the distances, measured horizontally, between the points $\{(z_i, y_i), i = 1, \ldots, k_n\}$ and a line).

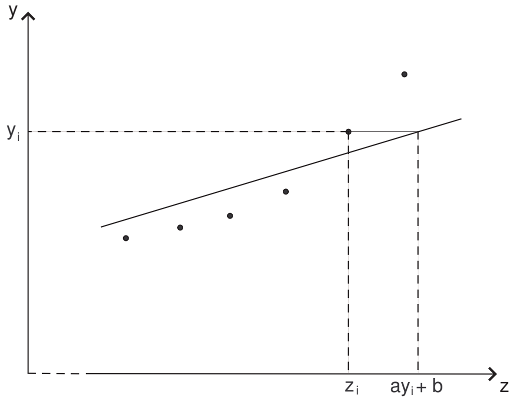

Figure 3.1: Distance between the point (z_i, y_i) and the point of a line with ordenate y_i

The R estimator obtained was $\widehat{R}_1(k_n) := \hat{a}_1^{-1}(k_n)$, that is,

$$\widehat{R}_1(k_n) = \frac{\sum_{i=1}^{k_n}\log^2(n/i) - \frac{1}{k_n}\left(\sum_{i=1}^{k_n}\log(n/i)\right)^2}{\sum_{i=1}^{k_n}\log(n/i)Z_{n-i+1,n} - \frac{1}{k_n}\left(\sum_{i=1}^{k_n}Z_{n-i+1,n}\right)\left(\sum_{i=1}^{k_n}\log(n/i)\right)} \tag{3.1.3}$$

where $Z_{1,n} \leq Z_{2,n} \leq \ldots Z_{n,n}$ denotes the o.s. (order statistics) of the sample $(Z_1, Z_2, \ldots, Z_n)$. In the particular case where $r(z) \equiv 1,\ z > 0\ (F(z) = 1 - e^{-Rz},\ z > 0)$, the previous problem corresponds to minimizing the function $f_2(a) = f_1(a,0) = \sum_{i=1}^{k_n}(z_i - ay_i)^2$. Schultze and Steinebach proposed the following estimator of R, in the sense of the least squares method:

$$\widehat{R}_2(k_n) := \hat{a}_2^{-1}(k_n) = \frac{\sum_{i=1}^{k_n}\log^2(n/i)}{\sum_{i=1}^{k_n}\log(n/i)Z_{n-i+1,n}}.$$

Another estimator of R was deduced directly from the equation $y = Rz - d$, minimizing the function $f_3(R,d) = \sum_{i=1}^{k_n}(y_i - Rz_i + d)^2$. Considering Figure 3.2, the problem corresponds to the computation of the slope of the line that minimizes the sum of the squares of the distances between the points (z_i, y_i) and the points of a line with abscissa z_i, respectively, $i = 1, \ldots, k_n$ (that is, calculating the slope of the line that minimizes the sum of the squares of the distances, measured vertically, between the points $\{(z_i, y_i), i = 1, \ldots, k_n\}$ and a line).

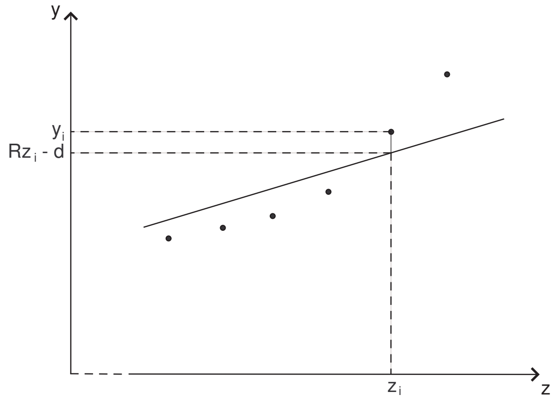

Figure 3.2: Distance between the point (z_i, y_i) and the point of a line with abscissa z_i

Schultze and Steinebach introduced then another estimator of R:

$$\widehat{R}_3(k_n) = \frac{\sum_{i=1}^{k_n} \log(n/i) Z_{n-i+1,n} - \frac{1}{k_n}\left(\sum_{i=1}^{k_n} Z_{n-i+1,n}\right)\left(\sum_{i=1}^{k_n} \log(n/i)\right)}{\sum_{i=1}^{k_n} Z_{n-i+1,n}^2 - \frac{1}{k_n}\left(\sum_{i=1}^{k_n} Z_{n-i+1,n}\right)^2}. \qquad (3.1.4)$$

The problem of estimating the exponential tail coefficient is equivalent to another one, which is the estimation of the tail index of the Pareto family. In fact, setting

$$X_i = e^{Z_i},$$

where Z_i, $i = 1, 2, \ldots$ are i.i.d. r.v. with d.f. that verifies (3.1.1), we have

$$1 - G(x) = P[X_1 > x] = x^{-1/\alpha} L(x), \qquad x > 1, \qquad (3.1.5)$$

where $\alpha = 1/R$,

$$L(x) = r(log x), \qquad (3.1.6)$$

and r is the regularly varying function in ∞ which appears in (3.1.1). Therefore, L is of slow variation in ∞, that is,

$$\forall y > 0, \lim_{t \to \infty} \frac{L(ty)}{L(t)} = 1.$$

So, $1-G$ is a regularly varying function in ∞ with index $-1/\alpha$. Note that (3.1.5) can also be written in an equivalent form

$$G^{-1}(1-s)=s^{-\alpha}\tilde{L}(s), \qquad 0<s<1, \tag{3.1.7}$$

for the slowly varying function at zero, $\tilde{L}$, which appears in (3.1.2).

In this context, many estimators have been proposed for α, among which stands out the classical Hill estimator, introduced by Hill (1975):

$$\widehat{H}(k_n)=\frac{1}{k_n}\sum_{i=1}^{k_n}\log X_{n-i+1,n}-\log X_{n-k_n,n}. \tag{3.1.8}$$

The corresponding estimator for the exponential tail coefficient is given by:

$$\widehat{H}^{-1}(k_n)=\left(\frac{1}{k_n}\sum_{i=1}^{k_n}Z_{n-i+1,n}-Z_{n-k_n,n}\right)^{-1}, \tag{3.1.9}$$

which we will call Hill estimator for R.

3.2 A geometric estimator for the exponential tail coefficient, $\widehat{R}$

Following the study of the estimators $\widehat{R}_1(k_n)$ and $\widehat{R}_3(k_n)$ corresponding to the Figures 3.1 and 3.2, the two points of view were considered simultaneously, minimizing the sum of the areas of the rectangles indicated in the following figure.

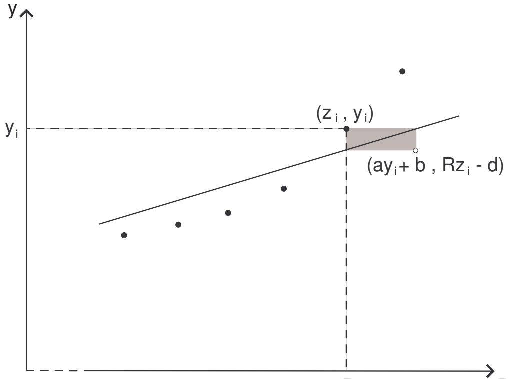

Figure 3.3: Areas of the rectangles to be minimized

Thereby, in Brito and Freitas (2003) a new estimator, of geometric type, was introduced for R, $\widehat{R}(k_n)$, which resulted from the minimization of the function

$$f(R,d)=\sum_{i=1}^{k_n}(y_i-Rz_i+d)(R^{-1}y_i+R^{-1}d-z_i).$$

The deduced estimator is as follows:

$$\widehat{R}(k_n)=\sqrt{\frac{\sum_{i=1}^{k_n}\log^2(n/i)-\frac{1}{k_n}\left(\sum_{i=1}^{k_n}\log(n/i)\right)^2}{\sum_{i=1}^{k_n}Z_{n-i+1,n}^2-\frac{1}{k_n}\left(\sum_{i=1}^{k_n}Z_{n-i+1,n}\right)^2}}. \tag{3.2.1}$$

Note that $\widehat{R}(k_n)$ is the geometric mean of the estimators $\widehat{R}_1(k_n)$ and $\widehat{R}_3(k_n)$, that is,

$$\widehat{R}(k_n)=\sqrt{\widehat{R}_1(k_n)\widehat{R}_3(k_n)}.$$

3.3 Properties of the geometric estimator $\widehat{R}$

To establish the asymptotic properties of the estimator it is necessary to impose some regularity conditions on the sequence k_n. The basic condition usually assumed in this context is that k_n is a sequence of positive integers satisfying:

$$1\leq k_n<n,\ \ k_n\to\infty\ \text{ and }\ k_n/n\to 0\ \text{ as }\ n\to\infty. \tag{3.3.1}$$

This condition will be assumed throughout the present work. We say, therefore, that k_n is an intermediate sequence of positive integers, essentially because k_n tends to infinity, but slower than n.

In Brito and Freitas (2003) and in Brito and Freitas (2010) the consistency of the geometric estimator was established, and the following result was obtained.

Theorem 3.3.1. *Let F be a d.f. that verifies (3.1.1) and k_n a sequence of integers satisfying (3.3.1). Then we have:*

$$\widehat{R}(k_n)\xrightarrow{P}R.$$

The following lemma describes the relation between the three estimators.

Lemma 3.3.1. *Let $\widehat{R}_1(k_n)$, $\widehat{R}_3(k_n)$ and $\widehat{R}(k_n)$ be the estimators defined in (3.1.3), (3.1.4) and (3.2.1), respectively. Then we have*

$$\widehat{R}_3(k_n)\leq\widehat{R}(k_n)\leq\widehat{R}_1(k_n).$$

Proof. To prove the result, it is enough to show that

$$\widehat{R}_3(k_n) \leq \widehat{R}_1(k_n),$$

because, since $\widehat{R}(k_n)$ is the geometric mean of the estimators $\widehat{R}_1(k_n)$ and $\widehat{R}_3(k_n)$, then $\widehat{R}(k_n)$ is necessarily between the other two.

Considering the expressions (3.1.3) and (3.1.4), that is equivalent to prove that

$$\begin{aligned}&\left(\sum_{i=1}^{k_n}\log(n/i)Z_{n-i+1,n}-\frac{1}{k_n}\sum_{i=1}^{k_n}Z_{n-i+1,n}\sum_{i=1}^{k_n}\log(n/i)\right)^2\leq\\ \leq&\left(\sum_{i=1}^{k_n}\log^2(n/i)-\frac{1}{k_n}\left(\sum_{i=1}^{k_n}\log(n/i)\right)^2\right)\left(\sum_{i=1}^{k_n}Z^2_{n-i+1,n}-\frac{1}{k_n}\left(\sum_{i=1}^{k_n}Z_{n-i+1,n}\right)^2\right)\end{aligned}\tag{3.3.2}$$

But,

$$\begin{aligned}&\sum_{i=1}^{k_n}\log(n/i)Z_{n-i+1,n}-\frac{1}{k_n}\sum_{i=1}^{k_n}Z_{n-i+1,n}\sum_{i=1}^{k_n}\log(n/i)=\\ &=\sum_{i=1}^{k_n}\left(\log(n/i)-\frac{1}{k_n}\sum_{i=1}^{k_n}\log(n/i)\right)\left(Z_{n-i+1,n}-\frac{1}{k_n}\sum_{i=1}^{k_n}Z_{n-i+1,n}\right),\\ &\sum_{i=1}^{k_n}\log^2(n/i)-\frac{1}{k_n}\left(\sum_{i=1}^{k_n}\log(n/i)\right)^2=\sum_{i=1}^{k_n}\left(\log(n/i)-\frac{1}{k_n}\sum_{i=1}^{k_n}\log(n/i)\right)^2\end{aligned}$$

and, similarly,

$$\sum_{i=1}^{k_n}Z^2_{n-i+1,n}-\frac{1}{k_n}\left(\sum_{i=1}^{k_n}Z_{n-i+1,n}\right)^2=\sum_{i=1}^{k_n}\left(Z_{n-i+1,n}-\frac{1}{k_n}\sum_{i=1}^{k_n}Z_{n-i+1,n}\right)^2.$$

The inequality (3.3.2) is thus equivalent to the following:

$$\begin{aligned}&\left(\sum_{i=1}^{k_n}\left(\log(n/i)-\frac{1}{k_n}\sum_{i=1}^{k_n}\log(n/i)\right)\left(Z_{n-i+1,n}-\frac{1}{k_n}\sum_{i=1}^{k_n}Z_{n-i+1,n}\right)\right)^2\leq\\ \leq&\sum_{i=1}^{k_n}\left(\log(n/i)-\frac{1}{k_n}\sum_{i=1}^{k_n}\log(n/i)\right)^2\sum_{i=1}^{k_n}\left(Z_{n-i+1,n}-\frac{1}{k_n}\sum_{i=1}^{k_n}Z_{n-i+1,n}\right)^2.\end{aligned}$$

The last inequality follows from the Cauchy-Schwarz inequality. □

The following result, obtained in Brito and Freitas (2003), establishes the asymptotic normality of $\widehat{R}(k_n)$.

Theorem 3.3.2. *Assume that F satisfies the condition (3.1.1), $\tilde{L}$ is the slowly varying function at zero that appears in (3.1.2) and k_n is a sequence of positive integers satisfying (3.3.1). If, as $n \to \infty$,*

$$k_n^{1/2} \sup_{\frac{1}{k_n} \leq y \leq 1} \left| \log \left(\frac{\tilde{L}(y t \frac{k_n}{n})}{\tilde{L}(\frac{k_n}{n})} \right) \right| \to 0$$

uniformly in t in compact sets of (0, ∞), then,

$$\frac{1}{\sqrt{2}R} k_n^{1/2} \left(\widehat{R}(k_n) - R \right) \xrightarrow{D} N(0,1).$$

Chapter 4

Geometric estimator for the adjustment coefficient

4.1 Introduction

In this chapter we will see that the adjustment coefficient defined in Section 2.3.2 can be estimated by any of the estimators of R considered in Chapter 3.

Let us start by noting that if the insurance company has an initial reserve u, then the ruin probability, already defined in Section 2.3.1, can be written as follows

$$\begin{aligned}\psi(u) &= P\left\{\inf_{t>0}\{u+ct-S(t)\}<0\right\}\\ &= P\left\{\sup_{t>0}\{S(t)-ct\}>u\right\}\\ &= P\left\{\max_{n\geqslant 1}\sum_{i=1}^{n}(X_i-cT_i)>u\right\}.\end{aligned}$$

Defining the i.i.d. r.v.

$$D_i := X_i - cT_i,\ i=1,2,\ldots \tag{4.1.1}$$

and the random walk induced by them:

$$U_0=0,\quad U_n=D_1+\ldots+D_n,\ n\geqslant 1,$$

the ruin probability can be written in the form

$$\psi(u)=P\left\{\max_{n\geqslant 1}U_n>u\right\}.$$

Note that if no conditions are imposed, ruin may occur almost surely. In fact, by the Strong Law of Large Numbers, $U_n/n \to E(D_1)$ almost surely. In the case where $E(D_1) > 0$, this implies that $\limsup_{n\to\infty} U_n = \infty$ and consequently $\psi(u) \equiv 1$. It also can be shown that $\psi(u) \equiv 1$ for $E(D_1) = 0$ (see, for example, Rolski et al. (1999), Theorem 6.3.1).

Note that assuming $E(D_1) < 0$ is equivalent to

$$c > \lambda E(X),$$

condition referred to in Section 2.3.2.

4.2 Estimating the adjustment coefficient

We now recall the *Lundberg inequality*:

$$\psi(u) < e^{-Ru},$$

where R is the adjustment coefficient defined in Section 2.3.2.

As already mentioned in the beginning of Chapter 3, this inequality allows to reduce the problem of the estimation of an upper-bound for the ruin probability to the estimation of the adjustment coefficient. Several estimators have been proposed for this coefficient using different methods (see, for example, Herkenrath (1986), Deheuvels and Steinebach (1990), Embrechts and Mikosch (1991), Pitts et al. (1996) and cited references).

Csörgő and Steinebach (1991) suggested estimating R, in the Cramér-Lundberg model, based on the auxiliary sequence (Z_k) of i.i.d. r.v. defined recursively by:

$$\begin{aligned} &W_0 = 0,\ W_n = max\left\{W_{n-1} + D_n, 0\right\},\ n = 1,2,\ldots, \\ &\nu_0 = 0,\ \nu_k = min\left\{n \geqslant \nu_{k-1} + 1 : W_n = 0\right\},\ k = 1,2,\ldots, \\ &Z_k = \max_{\nu_{k-1} \leqslant j < \nu_k} W_j,\ k = 1,2,\ldots, \end{aligned} \tag{4.2.1}$$

In this context W_n corresponds to the accumulated loss (of money) up to the moment of payment of the n-th claim, since the last money recovery. The r.v. $\nu_k - \nu_{k-1}$ can be interpreted as the number of claims payed during the k-th loss cycle and Z_k corresponds to the maximum accumulated loss of money in the k-th cycle.

Cohen (1969) determined, in the context of Queueing Theory, an exact distribution F of Z_k, in the case where the r.v. X_i or T_i are exponentially distributed. Based on this result, Csörgő and Steinebach (1991) observed that if $\{S(t); t \geqslant 0\}$ is a compound Poisson process, which is precisely the process under study in the present work (see Section 2.2), or claims X_i are exponentially distributed, then

$$1-F(z)=P[Z_k > z]=be^{-Rz}\left\{1+O(e^{-Az})\right\} \text{ as } z \to \infty, \tag{4.2.2}$$

with positive constants b and A. The generic notation big O (Landau notation) is presented next. Given $x_0 \in \{\mathbb{R} \cup \infty\}$, and considering two generic functions of real numbers f and g defined in a neighborhood of x_0, where g is non-null in this neighborhood, we write

$$f(x)=O(g(x)) \text{ as } x \to x_0$$

if and only if, for some $M \in \mathbb{R}^+$,

$$|f(x)| \leqslant M|g(x)| \text{ as } x \to x_0.$$

This way it can be easily verified that the d.f. F of the r.v. Z_k belongs to the family (3.1.1) of Section 3.1, where the exponential tail coefficient corresponds to the adjustment coefficient. Thus, the adjustment coefficient can be estimated by any of the estimators of R considered in Chapter 3.

4.3 Properties of the geometric estimator for the adjustment coefficient

In this section we will establish, in the context of the estimation of the adjustment coefficient, the consistency of the geometric estimator introduced in Section 3.2 and its asymptotic normality.

The following theorem establishes the consistency of the geometric estimator.

Theorem 4.3.1. *Considering the Cramér-Lundberg model and the conditions of existence of the adjustment coefficient described in Section 2.3.2, let us assume that $(Z_1, \ldots, Z_n)$ is a sample of i.i.d. r.v. built recursively as in (4.2.1) and k_n a sequence of positive integers that satisfies (3.3.1). So, if $\widehat{R}(k_n)$ denotes the geometric estimator defined in (3.2.1), we have that*

$$\widehat{R}(k_n) \xrightarrow{P} R,$$

where R is the adjustment coefficient.

Proof. As noted by Csörgő and Steinebach (1991), in the Cramér-Lundberg model, the distribution of the r.v. Z_k built as in (4.2.1) satisfies

$$1-F(z)=P[Z_k>z]=be^{-Rz}\left\{1+O(e^{-Az})\right\} \text{ as } z\to\infty.$$

Taking $r(z)=b\left(1+O(e^{-Az})\right)$, we have that, for all $z>0$,

$$\lim_{t\to\infty}\frac{r(tz)}{r(t)}=\lim_{t\to\infty}\frac{b\left(1+O(e^{-Azt})\right)}{b\left(1+O(e^{-At})\right)}=\lim_{t\to\infty}\frac{1+O(e^{-Azt})}{1+O(e^{-At})}=\frac{1+0}{1+0}=1=z^0,$$

and therefore $r(z)$ is of regular variation in ∞. Consequently, the family (4.2.2) is a subfamily of the family (3.1.1). So, the d.f. F of the r.v. Z_k satisfies (3.1.1). Applying the Theorem 3.3.1 we obtain, in this context, that

$$\widehat{R}(k_n)\xrightarrow{P}R.$$

□

The following theorem establishes the asymptotic normality of the geometric estimator, in the context of the estimation of the adjustment coefficient.

Theorem 4.3.2. *Considering the Cramér-Lundberg model and the conditions of existence of the adjustment coefficient described in Section 2.3.2, let us assume that $(Z_1,\ldots,Z_n)$ is a sample of i.i.d. r.v. built recursively as in (4.2.1) and A the positive constant that appears in (4.2.2). If k_n is a sequence of positive integers that satisfy (3.3.1) and*

$$k_n^{1/2}\left(\frac{k_n}{n}\right)^{A/R}\to 0, \quad \textit{as } n\to\infty,$$

then

$$\frac{1}{\sqrt{2}R}k_n^{1/2}\left(\widehat{R}(k_n)-R\right)\xrightarrow{D}N(0,1)$$

where R is the adjustment coefficient and $\widehat{R}(k_n)$ the geometric estimator defined in (3.2.1).

Proof. As noted by Csörgő and Steinebach (1991), in the Cramér-Lundberg model, the distribution of the r.v. Z_k built as in (4.2.1) satisfies

$$F(z)=1-be^{-Rz}\left\{1+O(e^{-Az})\right\} \text{ as } z\to\infty,$$

where R is the adjustment coefficient and b and A are positive constants.

Let us start by calculating $z = F^{-1}(s)$ as $s \to 1$, which is equivalent to determine z so that $s = F(z)$ as $z \to \infty$.

$$\begin{aligned} & s = F(z) \\ \Leftrightarrow\ & s = 1 - be^{-Rz}\left\{1 + O(e^{-Az})\right\} \\ \Leftrightarrow\ & 1 - s = be^{-Rz}\left\{1 + O(e^{-Az})\right\} \\ \Leftrightarrow\ & e^{-Rz} = \frac{1}{b}\ \frac{1-s}{1+O(e^{-Az})} \end{aligned} \tag{4.3.1}$$

Using Taylor series expansion we have that, as $z \to \infty$,

$$\frac{1}{1+O(e^{-Az})} = 1 + O(e^{-Az})$$

and therefore $s = F(z)$ is equivalent to

$$\begin{aligned} & e^{-Rz} = \frac{1-s}{b}\ \left(1 + O(e^{-Az})\right) = O(1-s) \\ \Leftrightarrow & -Rz = \log(O(1-s)) \\ \Leftrightarrow\ & z = -\frac{1}{R}\log(O(1-s)) \\ \Leftrightarrow\ & e^{-Az} = (O(1-s))^{A/R} = O\left((1-s)^{A/R}\right). \end{aligned} \tag{4.3.2}$$

Replacing (4.3.2) in (4.3.1) we obtain

$$\begin{aligned} & 1 - s = be^{-Rz}\left\{1 + O\left((1-s)^{A/R}\right)\right\} \\ \Leftrightarrow\ & be^{-Rz} = \frac{1-s}{1+O((1-s)^{A/R})} = (1-s)\left\{1 + O\left((1-s)^{A/R}\right)\right\} \\ \Leftrightarrow & -Rz = \log\left[\frac{1}{b}\ (1-s)\left\{1 + O\left((1-s)^{A/R}\right)\right\}\right] \\ \Leftrightarrow\ & z = -\frac{1}{R}\left(\log(1-s) + \log\left(b^{-1}\left(1 + O\left((1-s)^{A/R}\right)\right)\right)\right) \\ \Leftrightarrow\ & z = -\frac{1}{R}\log(1-s) + \log\left[\left(b^{-1}\left(1 + O\left((1-s)^{A/R}\right)\right)\right)^{-1/R}\right] \\ \Leftrightarrow\ & z = -\frac{1}{R}\log(1-s) + \log\left(b^{1/R}\left(1 + O\left((1-s)^{A/R}\right)\right)^{-1/R}\right) \\ \therefore\quad & F^{-1}(s) = -\frac{1}{R}\log(1-s) + \log\left(b^{1/R}\left(1 + O\left((1-s)^{A/R}\right)\right)^{-1/R}\right) \text{ as } s \to 1. \end{aligned}$$

Therefore,

$$F^{-1}(1-s) = -\frac{1}{R}\log s + \log\left(b^{1/R}\left(1 + O\left(s^{A/R}\right)^{-1/R}\right)\right) \text{ as } s \to 1.$$

Considering the equality (3.1.2) we have, in this case,

$$\tilde{L}(s) = b^{1/R}\left(1+O\left(s^{A/R}\right)\right)^{-1/R} \text{ as } s \to 1.$$

Consider now $1/k_n \leqslant y \leqslant 1$ and $a_1 \leqslant t \leqslant a_2$, with $0 < a_1 < 1 < a_2 < \infty$.
Thus,

$$\begin{aligned}
\frac{\tilde{L}(yt\frac{k_n}{n})}{\tilde{L}(\frac{k_n}{n})} &= \frac{b^{1/R}\left(1+O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right)\right)^{-1/R}}{b^{1/R}\left(1+O\left(\left(\frac{k_n}{n}\right)^{A/R}\right)\right)^{-1/R}} \\
&= \left(\frac{1+O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right)}{1+O\left(\left(\frac{k_n}{n}\right)^{A/R}\right)}\right)^{-1/R} \\
&= \left(\left(1+O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right)\right)\left(1+O\left(\left(\frac{k_n}{n}\right)^{A/R}\right)\right)\right)^{-1/R}.
\end{aligned}$$

Then,

$$\begin{aligned}
&\sup_{\frac{1}{k_n}\leqslant y\leqslant 1}\sup_{a_1\leqslant t\leqslant a_2}\left|\log\left(\frac{\tilde{L}(yt\frac{k_n}{n})}{\tilde{L}(\frac{k_n}{n})}\right)\right| \\
&= \sup_{\frac{1}{k_n}\leqslant y\leqslant 1}\sup_{a_1\leqslant t\leqslant a_2}\log\left(\left(\left(1+O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right)\right)\left(1+O\left(\left(\frac{k_n}{n}\right)^{A/R}\right)\right)\right)^{-1/R}\right) \\
&= \sup_{\frac{1}{k_n}\leqslant y\leqslant 1}\sup_{a_1\leqslant t\leqslant a_2}\log\left(1+O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right)\right)^{-1/R} \\
&= \sup_{\frac{1}{k_n}\leqslant y\leqslant 1}\sup_{a_1\leqslant t\leqslant a_2}\left(-\frac{1}{R}\log\left(1+O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right)\right)\right)
\end{aligned}$$

Using Taylor series expansion we have that, as $n \to \infty$ and consequently $k_n/n \to 0$,

$$\log\left(1+O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right)\right) = O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right).$$

Thus,

$$\sup_{\frac{1}{k_n}\leqslant y\leqslant 1}\sup_{a_1\leqslant t\leqslant a_2}\left|\log\left(\frac{\tilde{L}(yt\frac{k_n}{n})}{\tilde{L}(\frac{k_n}{n})}\right)\right| = \sup_{\frac{1}{k_n}\leqslant y\leqslant 1}\sup_{a_1\leqslant t\leqslant a_2}\left(-\frac{1}{R}\,O\left(\left(yt\frac{k_n}{n}\right)^{A/R}\right)\right)$$
$$= O\left(\left(\frac{k_n}{n}\right)^{A/R}\right).$$

So if $k_n^{1/2}\times\left(\frac{k_n}{n}\right)^{A/R}\to 0$ as $n\to\infty$ then, by the Theorem 3.3.2, the result follows. □

Corollary 4.3.1. *In the context of the Cramér-Lundberg model and considering that the individual claims are exponentially distributed, let us assume that $(Z_1,\dots,Z_n)$ is a sample of i.i.d. r.v. built recursively as in (4.2.1). If k_n is a sequence of positive integers so that*

$$k_n\to\infty \ \ \textit{and}\ \ \frac{k_n^{3/2}}{n}\to 0 \ \ \textit{as}\ \ n\to\infty,$$

then

$$\frac{1}{\sqrt{2}R}k_n^{1/2}\left(\widehat{R}(k_n)-R\right)\xrightarrow{D} N(0,1).$$

Proof. Cohen (1969) demonstrated that, in the Cramér-Lundberg model where the individual claims are exponentially distributed, the distribution of the r.v. Z_k built as in (4.2.1) satisfies

$$1-F(z)=be^{-Rz}\left\{1+O(e^{-Rz})\right\}\ \ \text{as}\ \ z\to\infty,$$

where b is a positive constant.

So, this distribution belongs to the family (4.2.2), where the constant A coincides with the adjustment coefficient R.

Thus, replacing A by R in the condition of the Theorem 4.3.2 we obtain:

$$k_n^{1/2}\left(\frac{k_n}{n}\right)^{R/R}\to 0,\ \text{as}\ n\to\infty,$$

which is equivalent to

$$\frac{k_n^{3/2}}{n}\to 0\ \ \text{as}\ \ n\to\infty.$$

□

Under the conditions of Theorem 4.3.2 or Corollary 4.3.1, it is possible to construct asymptotic confidence intervals of level p for the adjustment coefficient. The natural confidence bounds in this context are the one-sided ones given by:

$$I_{\widehat{R}}(k_n, p) = \left\{ R : \frac{1}{\sqrt{2}R} k_n^{1/2} \left(\widehat{R}(k_n) - R \right) \leqslant \Phi^{-1}(p) \right\},$$

where Φ denotes the d.f. of a $N(0,1)$ random variable.

This means that, as $n \to \infty$

$$P\left(R \in I_{\widehat{R}}(k_n, p)\right) \to p.$$

We observe that the above confidence bounds can be written in the form

$$I_{\widehat{R}}(k_n, p) = \left\{ R : R \geqslant \frac{k_n^{1/2} \widehat{R}(k_n)}{k_n^{1/2} + \sqrt{2}\Phi^{-1}(p)} \right\}.$$

Chapter 5

Simulation studies

In this chapter we will include simulation studies where the sample behavior of the geometric-type estimator $\widehat{R}(k_n)$, as an estimator of the adjustment coefficient, will be analyzed. We will also study the behavior of the Hill estimator, $\widehat{H}^{-1}(k_n)$ (see the end of Section 3.1).

Throughout this chapter we will consider the model previously studied, in which $\{S(t)\}_{t\geqslant 0}$ follows a compound Poisson process, that is, the case where $\{N(t)\}_{t\geqslant 0}$ follows a Poisson process. We will denote, as in previous chapters, the intensity by λ, a parameter that corresponds to the average number of claims per unit of time. This is equivalent to assume that the time elapsed between two successive claims has an exponential distribution of parameter λ, or, equivalently, with mean $\frac{1}{\lambda}$ (see Section 2.2):

$$T \sim Exp(\lambda),\ E(T) = \frac{1}{\lambda}.$$

In Brito and Freitas (2003) it was presented a small simulation study for the exponential model for individual claims. That study was quite limited because it was only considered a particular parameter for the exponential distribution.

In this work we intend not only to extend the study to other parameters of the exponential distribution, but also to analyze other models for the individual claims. Since we are working in the context of the existence of the adjustment coefficient, we will obviously test models for which, in particular, the moment generating function exists. The models that will be studied in this work are, in addition to the Exponential model, the Degenerate model, the Uniform model and the Normal model.

5.1 Models for the individual claims

5.1.1 Exponential model

In this section we will consider the case where individual claims are exponentially distributed with mean β, or, equivalently, with parameter $\frac{1}{\beta}$:

$$X \sim Exp\left(\frac{1}{\beta}\right),\ E(X) = \beta.$$

In Example 2.3.1 it was determined the adjustment coefficient in this particular model, obtaining

$$R = \frac{1}{\beta} - \frac{\lambda}{c}.$$

where c represents the rate of premiums collected per unit of time, considered in (2.3.1).

Denoting now

$$\alpha := E(cT_1) = \frac{c}{\lambda},$$

the following expression for the adjustment coefficient is obtained:

$$R = \frac{1}{\beta} - \frac{1}{\alpha} = \frac{\alpha - \beta}{\alpha\beta}.$$

For the Exponential model, Cohen (1969) determined the explicit form of the d.f. F of the i.i.d. r.v. $Z_1, Z_2, \ldots$ defined in (4.2.1), obtaining:

$$F(z) = \frac{1 - ae^{-(1-a)z/\beta}}{1 - a^2 e^{-(1-a)z/\beta}}, \qquad z > 0, \qquad (5.1.1)$$

where $a := \frac{\beta}{\alpha}$.

The generalized inverse of the previous distribution function is given by:

$$F^{-1}(1-s) = \begin{cases} \frac{\beta}{1-a} log\left(\frac{a(1-a)}{s} + a^2\right), & 0 < s < \frac{a}{1+a} \\ 0, & \text{for other values of } s \end{cases}.$$

5.1.2 Degenerate model

In this section we will analyze the case where individual claims are constant and equal to β. So we will assume:

$$X \equiv \beta.$$

In this particular case, we have

$$E(X) = \beta \text{ and } M_X(r) = e^{r\beta}.$$

Replacing in equation (2.3.2) the expression of $M_X(r)$, we obtain

$$\lambda + rc = \lambda e^{r\beta}.$$

Dividing both members of the previous equation by λ and denoting

$$\alpha := E(cT_1) = \frac{c}{\lambda},$$

the adjustment coefficient is, therefore, the positive solution of the equation

$$e^{r\beta} - \alpha r - 1 = 0.$$

5.1.3 Uniform model

Now, we will consider the case where individual claims have Uniform distribution in the interval $[a, b]$:

$$X \sim U(a, b),\ E(X) = \frac{a+b}{2} =: \beta.$$

In this particular case, we have

$$M_X(r) = \frac{e^{rb} - e^{ra}}{r(b-a)}.$$

Replacing in equation (2.3.2) the expression of $M_X(r)$, we obtain

$$\lambda + rc = \lambda \left[\frac{e^{rb} - e^{ra}}{r(b-a)} \right].$$

Dividing both members of the previous equation by λ and denoting

$$\alpha := E(cT_1) = \frac{c}{\lambda},$$

the adjustment coefficient is, therefore, the positive solution of the equation

$$\frac{e^{rb} - e^{ra}}{r(b-a)} - \alpha r - 1 = 0.$$

5.1.4 Normal model

Next we will consider the case where individual claims have Normal distribution with parameters β and s:

$$X \sim N(\beta, s),\ E(X) = \beta \text{ e } V(X) = s^2.$$

In this particular case, we have

$$M_X(r) = e^{\frac{s^2 r^2}{2} + \beta r}.$$

Replacing in equation (2.3.2) the expression of $M_X(r)$, we obtain

$$\lambda + rc = \lambda e^{\frac{s^2 r^2}{2} + \beta r}.$$

Dividing both members of the previous equation by λ and denoting

$$\alpha := E(cT_1) = \frac{c}{\lambda},$$

the adjustment coefficient is, therefore, the positive solution of the equation

$$e^{\frac{s^2 r^2}{2} + \beta r} - \alpha r - 1 = 0.$$

5.2 Simulation procedure

The simulation studies were carried out using software R, through the construction of programs entirely from scratch.

5.2.1 Exponential model

To simulate the Exponential model described in Subsection 5.1.1, where the individual claims are exponentially distributed, we must generate first the sample $(Z_1, \ldots, Z_n)$ (of size n), where each random variable Z_i has d.f. F given by (5.1.1).

Then, this sample would have to be ordered, so that we could take the k_n largest observations of the sample in order to compute the values of the estimators. With the purpose of a greater efficiency and speed, rather than ordering the obtained vector, we will generate an ordered sample directly. To do this we relied on the fact that if we have a sample $(U_1, U_2, \ldots, U_{k_n})$ of i.i.d. r.v. uniformly distributed over the interval (0, 1) and if we do:

$$\begin{aligned} U_{n,n} &:= U_1^{1/n}, \\ U_{n-i+1,n} &:= U_{n-i+2,n} U_i^{1/(n-i+1)}, \qquad\qquad i = 2, 3, \ldots, k_n, \end{aligned}$$

we can obtain $Z_{n-i+1,n}$ by

$$Z_{n-i+1,n} := F^{-1}(U_{n-i+1,n}), \qquad\qquad i = 1, 2, \ldots, k_n.$$

After generating the sample $(Z_1, \ldots, Z_n)$ we can obtain the estimates for R corresponding to the geometric (3.2.1) and Hill (3.1.9) estimators. Thus, for each of the estimators, it will be simulated an estimate for R for each value of k_n.

This process will be repeated several times and it will be computed, for each k_n considered, the mean and the standard deviation of the values obtained for each estimator.

The procedure will be performed for different values of the sample size n.

5.2.2 Non-exponential models

Regarding the non-exponential models, referred to in Subsections 5.1.2 to 5.1.4, unlike the exponential model, the explicit expression of F^{-1} used directly to generate the samples $(Z_1, \ldots, Z_n)$ is not known. Thus, in this case, the observations of the random variables Z_i, are not generated directly, but built recursively as in (4.2.1) based on the sample of r.v. D_i (see Expression (4.1.1)).

It is important to note that if we consider a certain size for the D sample, this will result in a random size for the Z sample. Therefore, the idea was to generate a sufficiently large number of r.v. D_i in order to obtain a desired fixed size, n, for the sample of r.v. Z_i.

After generating the Z sample, it should be ordered, so that the largest observations can be used in the computation of the R estimates using the geometric and Hill estimators.

As in the exponential model, for both estimators, an estimate for R will be simulated for each value of k_n. The experiment will be repeated multiple times, obtaining the corresponding mean and standard deviation of the estimates for each considered value of k_n. The procedure will be performed for different values of the sample size n.

5.2.3 Empirical method for the choice of k_n

As we will see in the simulation results, the sample behavior of the estimators depends strongly on the number k_n of observations considered in the estimation. For this problem of choosing the optimal value of k_n, that we will denote by $\tilde{k}$, we present an empirical method proposed by Brito and Freitas (2003) and inspired in the work of Schultze and Steinebach (1996). The latter authors proposed the following:

- for the estimator $\widehat{R}_1$, choose $\tilde{k} \in \{l, l+1, \ldots, n\}$ $(l > 2)$ such that $\frac{1}{\tilde{k}} \sum_{i=1}^{\tilde{k}} \left(z_i - \widehat{a}(\tilde{k}) y_i - \widehat{b}(\tilde{k})\right)^2$ is minimal, where $\widehat{a}(\tilde{k}) = 1/\widehat{R}_1(\tilde{k})$ and $\widehat{b}(\tilde{k}) = \frac{1}{\tilde{k}} \sum_{i=1}^{\tilde{k}} \left(z_i - \frac{y_i}{\widehat{R}_1(\tilde{k})}\right)$ (that is, choose $\tilde{k} \in \{l, l+1, \ldots, n\}$ $(l > 2)$ such that the average of the squares of the distances, measured horizontally, between the points $\left\{(z_i, y_i), i = 1, \ldots, \tilde{k}\right\}$ and the corresponding linear regression line is minimal);

- for the estimator $\widehat{R}_3$, choose $\tilde{k} \in \{l, l+1, \ldots, n\}$ $(l > 2)$ such that $\frac{1}{\tilde{k}} \sum_{i=1}^{\tilde{k}} \left(y_i - \widehat{R}_3(\tilde{k}) z_i + \widehat{d}(\tilde{k})\right)^2$ is minimal, where $\widehat{d}(\tilde{k}) = \frac{1}{\tilde{k}} \sum_{i=1}^{\tilde{k}} \left(\widehat{R}_3(\tilde{k}) z_i - y_i\right)$ (that is, choose $\tilde{k} \in \{l, l+1, \ldots, n\}$ $(l > 2)$ such that the average of the squares of the distances, measured vertically, between the points $\left\{(z_i, y_i), i = 1, \ldots, \tilde{k}\right\}$ and the corresponding linear regression line is minimal), where $y_i := \log(n/i)$ and $z_i := z_{n-i+1,n}$ is the i-th largest observation of the sample $(Z_1, \dots, Z_n)$.

Brito and Freitas (2003), motivated by the simulation results obtained by Schultze and Steinebach, adapted this procedure to the estimator $\widehat{R}$ choosing $\tilde{k} \in \{l, l+1, \dots, n\}$ $(l > 2)$ such that $\frac{1}{\tilde{k}} \sum_{i=1}^{\tilde{k}} \left(y_i - \widehat{R}(\tilde{k}) z_i + \widehat{d}(\tilde{k})\right)\left(\widehat{a}(\tilde{k}) y_i + \widehat{b}(\tilde{k}) - z_i\right)$ is minimal, where $\widehat{a}(\tilde{k}) = 1/\widehat{R}(\tilde{k})$, $\widehat{b}(\tilde{k}) = \frac{1}{\tilde{k}} \sum_{i=1}^{\tilde{k}} \left(z_i - \frac{y_i}{\widehat{R}(\tilde{k})}\right)$ and $\widehat{d}(\tilde{k}) = \frac{1}{\tilde{k}} \sum_{i=1}^{\tilde{k}} \left(\widehat{R}(\tilde{k}) z_i - y_i\right)$ (that is, choose $\tilde{k} \in \{l, l+1, \dots, n\}$ $(l > 2)$ such that the mean of the areas of the rectangles indicated in Figure 3.3 is minimal).

In the simulations that we will present in the following section, we include the study of the optimal k_n using the empirical method that we have just exposed.

5.3 Simulation results

In the simulations for the exponential model, we considered three different sizes for the Z sample: $n = 50, 100, 200$. In each case, we computed the estimates for several values of k_n (for $n = 50$ we considered k_n up to 40, for $n = 100$ we considered k_n up to 70 and for $n = 200$ we considered k_n up to 120). After 1000 repetitions of the procedure we computed, for each k_n considered, the mean and the standard deviation of the obtained estimates.

For each sample size, three pairs for (α, β) were tested, corresponding to different orders of magnitude for R (computed analytically):

- for $(\alpha, \beta) = (24000, 10000)$ we obtain $R = 5.8333 \times 10^{-5}$;
- for $(\alpha, \beta) = (2, 1)$ we obtain $R = 0.5$;
- for $(\alpha, \beta) = (0.002, 0.0012)$ we obtain $R = 333.33$.

For each sample size and each pair of parameters, we will present a graph with the mean of the estimates. The horizontal axis will display the different values of k_n. The standard deviation of the estimates obtained for some values of k_n will be presented in tables.

We also present the mean and the standard deviation of $\tilde{k}$ obtained through the empirical method for the choice of k_n described in Subsection 5.2.3 (the procedure is repeated 1000 times).

Regarding the non-exponential models, we considered two sample sizes, $n = 50$ and $n = 100$, and two pairs of parameters for (α, β). For each sample size and pair of parameters we will present, in a similar way to what was done for the exponential model, a graph with the mean of the estimates, a table with the standard deviation for some values of k_n and the mean and standard deviation of $\tilde{k}$.

The parameters (α, β) where chosen in order to obtain, approximately, the same values for R as in the exponential model (5.8×10^{-5} and 0.5), and also considering the same values for α used in that model.

Thus, in the case of the degenerate model, we started by considering $\alpha = 24000$ and in the equation of the adjustment coefficient for this model (indicated at the end of Subsection 5.1.2) we replaced R by 5.8×10^{-5}. In this way, the equation only depends on β. By solving the same one in order to this parameter, we obtain a value close to 15000. So, the first pair considered will be $(\alpha, \beta) = (24000, 15000)$ (which corresponds to $R = 5.8 \times 10^{-5}$). Still for the degenerate model, we considered $R = 0.5$ and $\alpha = 2$ (as in the exponential model), obtaining $\beta = 1.385$ approximately. The same procedure was applied to the uniform and normal models.

In the uniform model, we considered $a = 0$ (see Subsection 5.1.3), and so, fixing the value of R (and of α), we obtained the value of b by solving the equation of the adjustment coefficient (indicated at the end of Subsection 5.1.3). Thus, for this model, considering $R = 5.8 \times 10^{-5}$ and $\alpha = 24000$, we obtained $b = 26695$, and considering $R = 0.5$ and $\alpha = 2$, we obtained $b = 2.513$.

For the normal model, it was considered the case where the standard deviation is equal to 0.1 times the mean (10% of β). Taking this into account in the equation indicated at the end of Subsection 5.1.4, it will depend only on β (if we fix α). So, considering $R = 5.8 \times 10^{-5}$ and $\alpha = 24000$, we obtained β close to 15000, and considering $R = 0.5$ and $\alpha = 2$, we obtained β close to 1.385.

5.3.1 Exponential model

Simulation with $n = 50$ and $(\alpha, \beta) = (24000, 10000)$

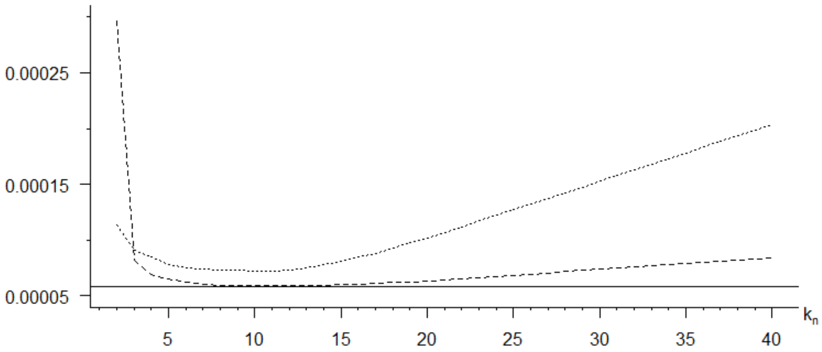

Figure 5.1: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 5.8333 \times 10^{-5}$ (full line) - exponential model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
5	6.17×10^{-5}	5.04×10^{-5}
10	3.08×10^{-5}	3.22×10^{-5}
15	2.63×10^{-5}	3.54×10^{-5}
20	2.57×10^{-5}	4.69×10^{-5}
25	2.69×10^{-5}	5.88×10^{-5}
30	2.88×10^{-5}	7.06×10^{-5}
35	3.08×10^{-5}	8.23×10^{-5}
40	3.28×10^{-5}	9.41×10^{-5}

Table 5.1: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - exponential model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
11.01	6.37

Table 5.2: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - exponential model

Simulation with $n = 100$ and $(\alpha, \beta) = (24000, 10000)$

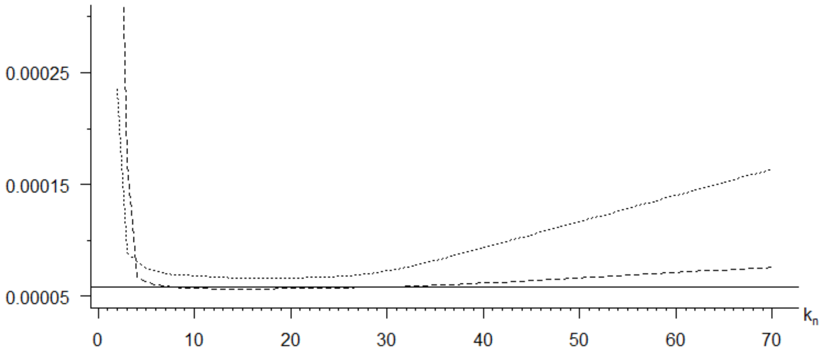

Figure 5.2: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 5.8333 \times 10^{-5}$ (full line) - exponential model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
10	2.66×10^{-5}	2.51×10^{-5}
20	1.88×10^{-5}	1.62×10^{-5}
30	1.60×10^{-5}	1.78×10^{-5}
40	1.58×10^{-5}	2.54×10^{-5}
50	1.67×10^{-5}	3.18×10^{-5}
60	1.78×10^{-5}	3.82×10^{-5}
70	1.89×10^{-5}	4.45×10^{-5}

Table 5.3: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - exponential model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
19.02	13.65

Table 5.4: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - exponential model

Simulation with $n = 200$ and $(\alpha, \beta) = (24000, 10000)$

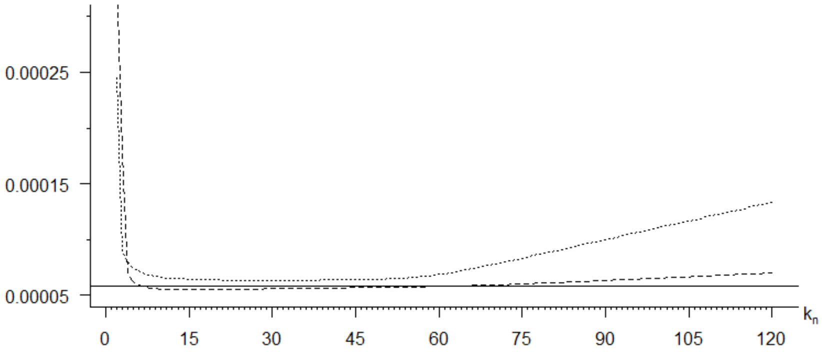

Figure 5.3: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 120$, obtained from 1000 Z samples of size $n = 200$, where $R = 5.8333 \times 10^{-5}$ (full line) - exponential model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
15	2.04×10^{-5}	1.81×10^{-5}
30	1.46×10^{-5}	1.19×10^{-5}
45	1.22×10^{-5}	1.02×10^{-5}
60	1.10×10^{-5}	1.11×10^{-5}
75	1.07×10^{-5}	1.47×10^{-5}
90	1.09×10^{-5}	1.77×10^{-5}
105	1.13×10^{-5}	2.07×10^{-5}
120	1.18×10^{-5}	2.36×10^{-5}

Table 5.5: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 200$ - exponential model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
39.07	28.46

Table 5.6: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 200$ and $l = 5$ - exponential model

Simulation with $n = 50$ and $(\alpha, \beta) = (2, 1)$

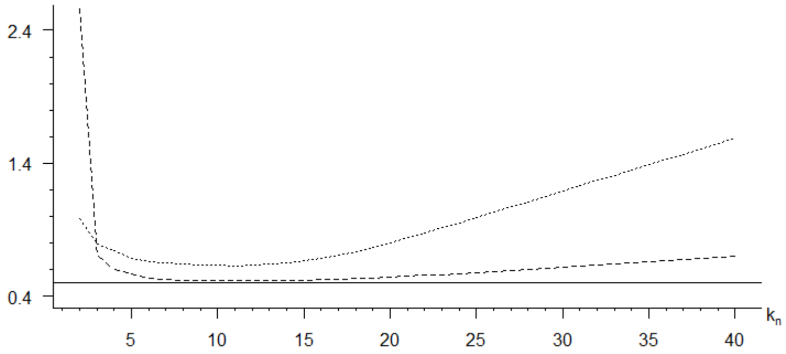

Figure 5.4: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 0.5$ (full line) - exponential model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
5	5.43×10^{-1}	4.45×10^{-1}
10	2.74×10^{-1}	2.79×10^{-1}
15	2.30×10^{-1}	2.57×10^{-1}
20	2.16×10^{-1}	3.28×10^{-1}
25	2.19×10^{-1}	4.12×10^{-1}
30	2.29×10^{-1}	4.95×10^{-1}
35	2.43×10^{-1}	5.77×10^{-1}
40	2.57×10^{-1}	6.59×10^{-1}

Table 5.7: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - exponential model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
11.81	7.12

Table 5.8: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - exponential model

Simulation with $n = 100$ and $(\alpha, \beta) = (2, 1)$

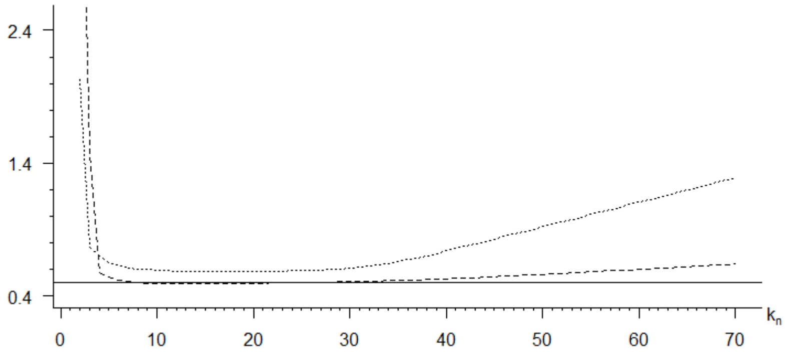

Figure 5.5: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 0.5$ (full line) - exponential model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
10	2.33×10^{-1}	2.21×10^{-1}
20	1.67×10^{-1}	1.44×10^{-1}
30	1.41×10^{-1}	1.30×10^{-1}
40	1.34×10^{-1}	1.85×10^{-1}
50	1.38×10^{-1}	2.33×10^{-1}
60	1.45×10^{-1}	2.80×10^{-1}
70	1.53×10^{-1}	3.26×10^{-1}

Table 5.9: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - exponential model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
20.45	15.21

Table 5.10: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - exponential model

Simulation with $n = 200$ and $(\alpha, \beta) = (2, 1)$

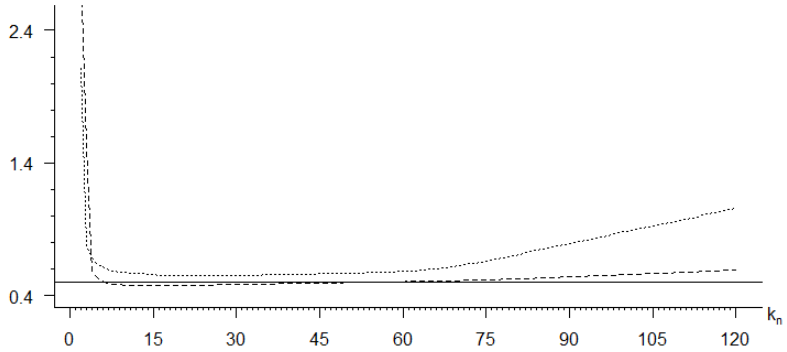

Figure 5.6: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 120$, obtained from 1000 Z samples of size $n = 200$, where $R = 0.5$ (full line) - exponential model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
15	1.77×10^{-1}	1.58×10^{-1}
30	1.28×10^{-1}	1.05×10^{-1}
45	1.08×10^{-1}	9.11×10^{-2}
60	9.73×10^{-2}	8.40×10^{-2}
75	9.22×10^{-2}	1.08×10^{-1}
90	9.21×10^{-2}	1.32×10^{-1}
105	9.44×10^{-2}	1.54×10^{-1}
120	9.77×10^{-2}	1.76×10^{-1}

Table 5.11: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 200$ - exponential model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
43.46	32.16

Table 5.12: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 200$ and $l = 5$ - exponential model

Simulation with $n = 50$ and $(\alpha, \beta) = (0.002, 0.0012)$

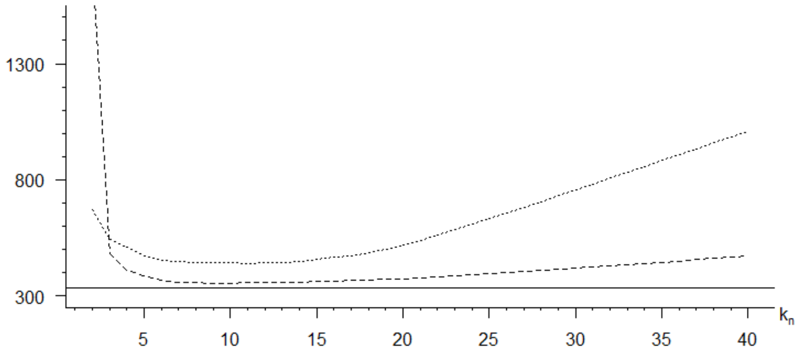

Figure 5.7: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 333.33$ (full line) - exponential model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
5	3.78×10^2	3.12×10^2
10	1.95×10^2	2.00×10^2
15	1.64×10^2	1.67×10^2
20	1.51×10^2	1.91×10^2
25	1.49×10^2	2.42×10^2
30	1.53×10^2	2.90×10^2
35	1.60×10^2	3.38×10^2
40	1.68×10^2	3.86×10^2

Table 5.13: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - exponential model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
12.40	7.83

Table 5.14: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - exponential model

Simulation with $n = 100$ and $(\alpha, \beta) = (0.002, 0.0012)$

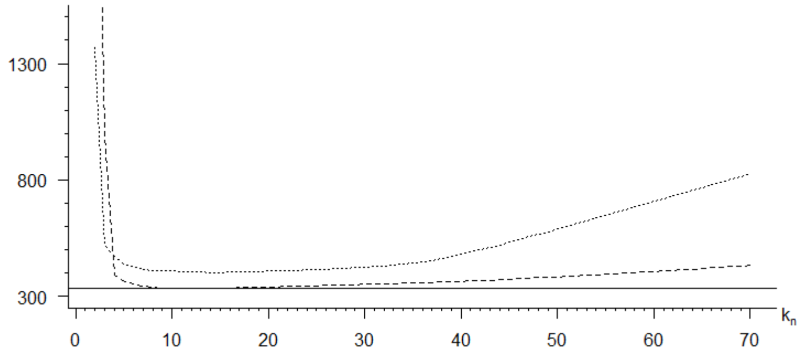

Figure 5.8: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 333.33$ (full line) - exponential model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
10	1.61×10^2	1.54×10^2
20	1.18×10^2	1.03×10^2
30	1.01×10^2	8.86×10^1
40	9.35×10^1	1.09×10^2
50	9.42×10^1	1.41×10^2
60	9.81×10^1	1.70×10^2
70	1.03×10^2	1.98×10^2

Table 5.15: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - exponential model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
21.45	16.64

Table 5.16: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - exponential model

Simulation with $n = 200$ and $(\alpha, \beta) = (0.002, 0.0012)$

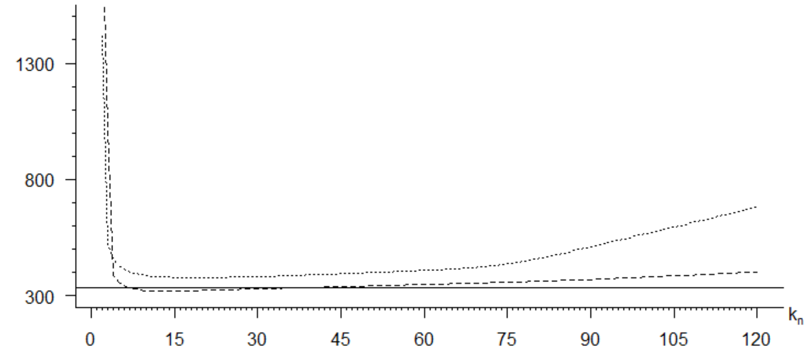

Figure 5.9: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 120$, obtained from 1000 Z samples of size $n = 200$, where $R = 333.33$ (full line) - exponential model

k_n	Standard deviation ($\widehat{R}(k_n)$)	Standard deviation ($\widehat{H}^{-1}(k_n)$)
15	1.21×10^2	1.09×10^2
30	8.87×10^1	7.40×10^1
45	7.57×10^1	6.52×10^1
60	6.92×10^1	5.92×10^1
75	6.53×10^1	6.33×10^1
90	6.39×10^1	8.10×10^1
105	6.47×10^1	9.47×10^1
120	6.64×10^1	1.08×10^2

Table 5.17: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 200$ - exponential model

Mean ($\tilde{k}$)	Standard deviation ($\tilde{k}$)
45.26	34.72

Table 5.18: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 200$ and $l = 5$ - exponential model

5.3.2 Degenerate model

Simulation with $n = 50$ and $(\alpha, \beta) = (24000, 15000)$

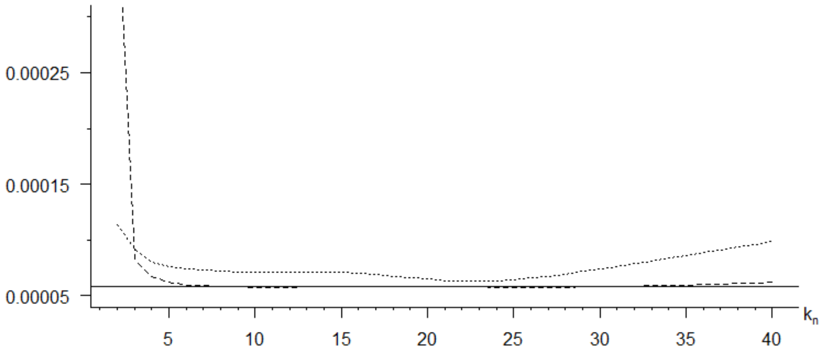

Figure 5.10: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 5.8 \times 10^{-5}$ (full line) - degenerate model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
5	4.68×10^{-5}	4.29×10^{-5}
10	2.84×10^{-5}	2.81×10^{-5}
15	2.34×10^{-5}	2.20×10^{-5}
20	1.97×10^{-5}	1.65×10^{-5}
25	1.66×10^{-5}	1.45×10^{-5}
30	1.51×10^{-5}	1.82×10^{-5}
35	1.49×10^{-5}	2.14×10^{-5}
40	1.51×10^{-5}	2.45×10^{-5}

Table 5.19: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - degenerate model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
17.30	12.26

Table 5.20: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - degenerate model

Simulation with $n = 100$ and $(\alpha, \beta) = (24000, 15000)$

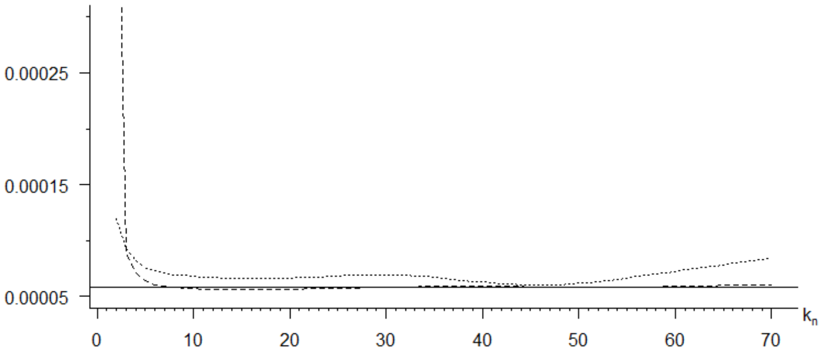

Figure 5.11: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 5.8 \times 10^{-5}$ (full line) - degenerate model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
10	2.74×10^{-5}	2.42×10^{-5}
20	1.85×10^{-5}	1.67×10^{-5}
30	1.56×10^{-5}	1.43×10^{-5}
40	1.36×10^{-5}	1.09×10^{-5}
50	1.16×10^{-5}	1.00×10^{-5}
60	1.07×10^{-5}	1.26×10^{-5}
70	1.06×10^{-5}	1.47×10^{-5}

Table 5.21: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - degenerate model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
33.33	26.05

Table 5.22: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - degenerate model

Simulation with $n = 50$ and $(\alpha, \beta) = (2, 1.385)$

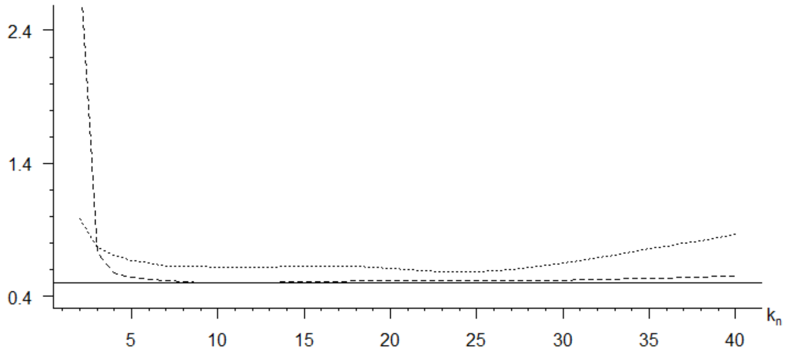

Figure 5.12: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 0.5$ (full line) - degenerate model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
5	5.04×10^{-1}	4.39×10^{-1}
10	2.53×10^{-1}	2.42×10^{-1}
15	2.09×10^{-1}	1.92×10^{-1}
20	1.80×10^{-1}	1.59×10^{-1}
25	1.55×10^{-1}	1.31×10^{-1}
30	1.41×10^{-1}	1.58×10^{-1}
35	1.38×10^{-1}	1.86×10^{-1}
40	1.39×10^{-1}	2.13×10^{-1}

Table 5.23: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - degenerate model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
17.46	12.82

Table 5.24: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - degenerate model

Simulation with $n = 100$ and $(\alpha, \beta) = (2, 1.385)$

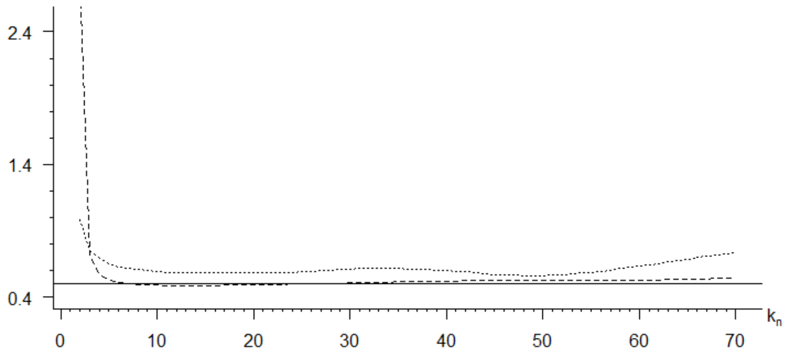

Figure 5.13: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 0.5$ (full line) - degenerate model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H^{-1}}(k_n))$
10	2.41×10^{-1}	2.24×10^{-1}
20	1.62×10^{-1}	1.43×10^{-1}
30	1.37×10^{-1}	1.29×10^{-1}
40	1.24×10^{-1}	1.02×10^{-1}
50	1.10×10^{-1}	8.61×10^{-2}
60	1.01×10^{-1}	1.08×10^{-1}
70	9.85×10^{-2}	1.26×10^{-1}

Table 5.25: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - degenerate model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
36.23	27.11

Table 5.26: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - degenerate model

5.3.3 Uniform model

Simulation with $n = 50$, $\alpha = 24000$, $a = 0$ and $b = 26695$

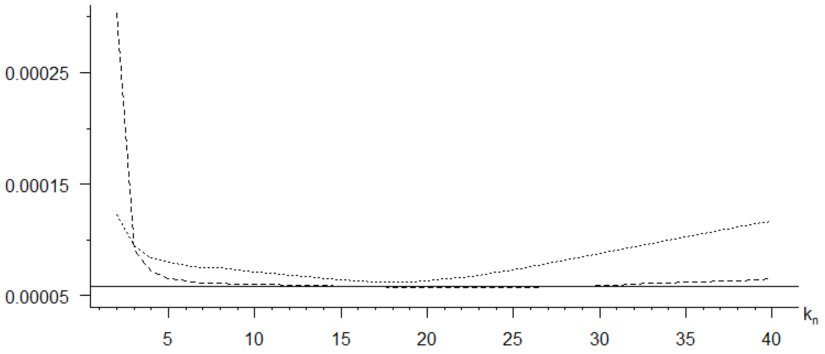

Figure 5.14: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 5.8 \times 10^{-5}$ (full line) - uniform model

k_n	Standard deviation ($\widehat{R}(k_n)$)	Standard deviation ($\widehat{H}^{-1}(k_n)$)
5	5.44×10^{-5}	4.56×10^{-5}
10	2.82×10^{-5}	2.55×10^{-5}
15	2.17×10^{-5}	1.79×10^{-5}
20	1.74×10^{-5}	1.58×10^{-5}
25	1.55×10^{-5}	2.08×10^{-5}
30	1.53×10^{-5}	2.53×10^{-5}
35	1.58×10^{-5}	2.95×10^{-5}
40	1.64×10^{-5}	3.38×10^{-5}

Table 5.27: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - uniform model

Mean ($\tilde{k}$)	Standard deviation ($\tilde{k}$)
15.63	9.86

Table 5.28: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - uniform model

Simulation with $n = 100$, $\alpha = 24000$, $a = 0$ and $b = 26695$

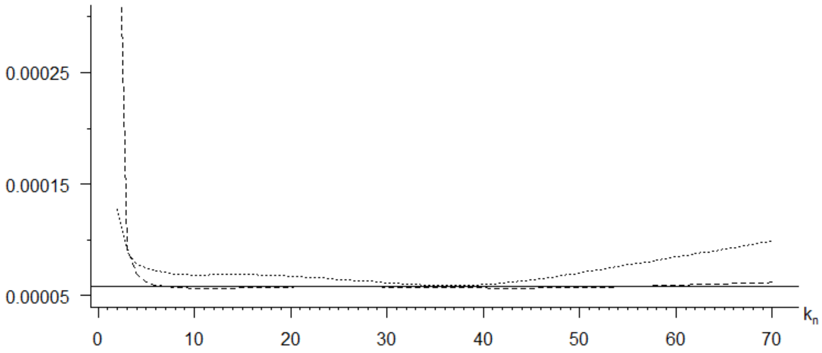

Figure 5.15: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 5.8 \times 10^{-5}$ (full line) - uniform model

k_n	Standard deviation ($\widehat{R}(k_n)$)	Standard deviation ($\widehat{H^{-1}}(k_n)$)
10	2.59×10^{-5}	2.44×10^{-5}
20	1.86×10^{-5}	1.61×10^{-5}
30	1.50×10^{-5}	1.20×10^{-5}
40	1.24×10^{-5}	1.04×10^{-5}
50	1.13×10^{-5}	1.41×10^{-5}
60	1.12×10^{-5}	1.70×10^{-5}
70	1.15×10^{-5}	1.98×10^{-5}

Table 5.29: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - uniform model

Mean ($\tilde{k}$)	Standard deviation ($\tilde{k}$)
30.22	20.98

Table 5.30: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - uniform model

Simulation with $n = 50$, $\alpha = 2$, $a = 0$ and $b = 2.513$

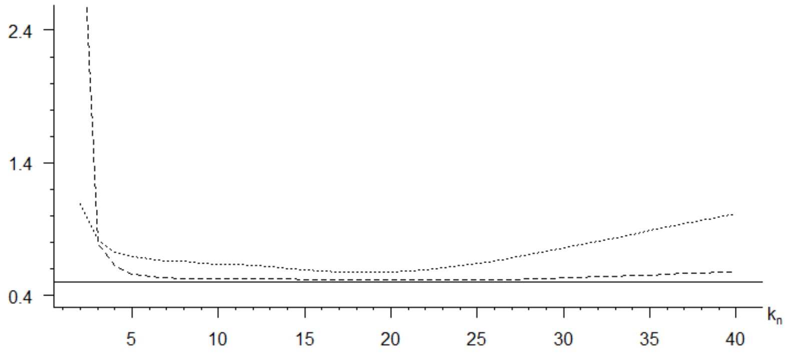

Figure 5.16: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \dots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 0.5$ (full line) - uniform model

k_n	Standard deviation ($\widehat{R}(k_n)$)	Standard deviation ($\widehat{H^{-1}}(k_n)$)
5	4.89×10^{-1}	4.25×10^{-1}
10	2.61×10^{-1}	2.38×10^{-1}
15	2.06×10^{-1}	1.72×10^{-1}
20	1.70×10^{-1}	1.41×10^{-1}
25	1.51×10^{-1}	1.76×10^{-1}
30	1.48×10^{-1}	2.18×10^{-1}
35	1.51×10^{-1}	2.55×10^{-1}
40	1.57×10^{-1}	2.91×10^{-1}

Table 5.31: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - uniform model

Mean ($\tilde{k}$)	Standard deviation ($\tilde{k}$)
15.98	10.65

Table 5.32: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - uniform model

Simulation with $n = 100$, $\alpha = 2$, $a = 0$ and $b = 2.513$

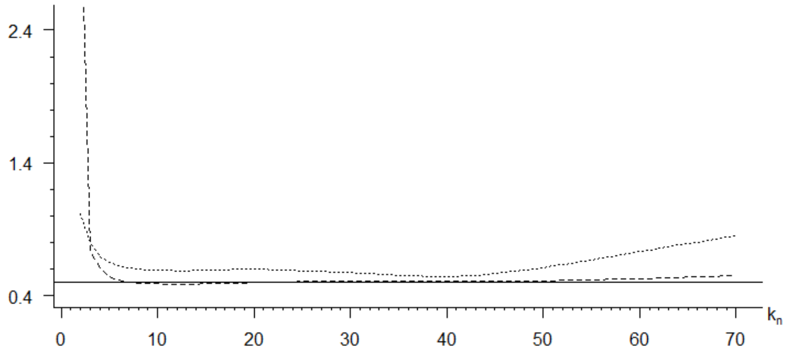

Figure 5.17: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \dots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 0.5$ (full line) - uniform model

k_n	Standard deviation ($\widehat{R}(k_n)$)	Standard deviation ($\widehat{H}^{-1}(k_n)$)
10	2.31×10^{-1}	2.11×10^{-1}
20	1.66×10^{-1}	1.52×10^{-1}
30	1.40×10^{-1}	1.13×10^{-1}
40	1.20×10^{-1}	9.10×10^{-2}
50	1.07×10^{-1}	1.16×10^{-1}
60	1.05×10^{-1}	1.41×10^{-1}
70	1.07×10^{-1}	1.65×10^{-1}

Table 5.33: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - uniform model

Mean ($\tilde{k}$)	Standard deviation ($\tilde{k}$)
29.98	22.31

Table 5.34: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - uniform model

5.3.4 Normal model

Simulation with $n = 50$ and $(\alpha, \beta) = (24000, 15000)$

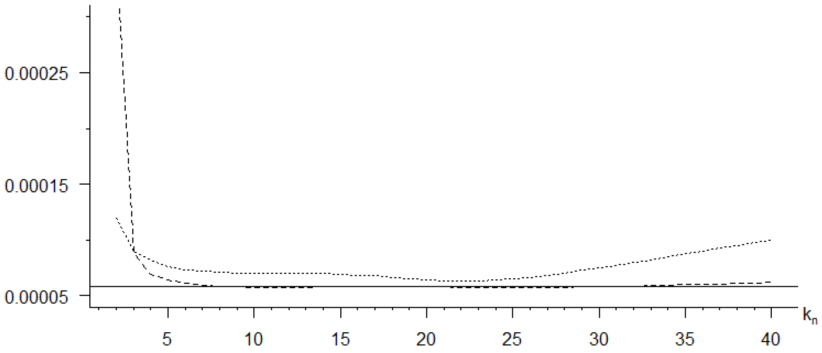

Figure 5.18: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 5.8 \times 10^{-5}$ (full line) - normal model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
5	5.21×10^{-5}	4.29×10^{-5}
10	2.74×10^{-5}	2.69×10^{-5}
15	2.24×10^{-5}	2.14×10^{-5}
20	1.92×10^{-5}	1.61×10^{-5}
25	1.66×10^{-5}	1.62×10^{-5}
30	1.53×10^{-5}	2.00×10^{-5}
35	1.52×10^{-5}	2.34×10^{-5}
40	1.55×10^{-5}	2.67×10^{-5}

Table 5.35: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - normal model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
16.62	11.80

Table 5.36: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - normal model

Simulation with $n = 100$ and $(\alpha, \beta) = (24000, 15000)$

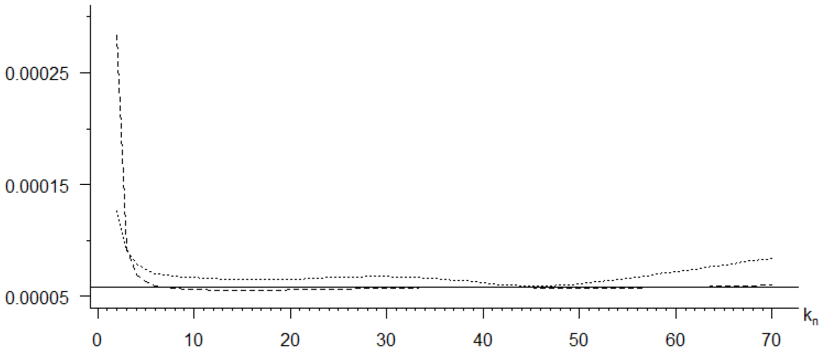

Figure 5.19: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 5.8 \times 10^{-5}$ (full line) - normal model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
10	2.64×10^{-5}	2.35×10^{-5}
20	1.74×10^{-5}	1.56×10^{-5}
30	1.48×10^{-5}	1.41×10^{-5}
40	1.31×10^{-5}	1.10×10^{-5}
50	1.13×10^{-5}	9.92×10^{-6}
60	1.06×10^{-5}	1.24×10^{-5}
70	1.05×10^{-5}	1.45×10^{-5}

Table 5.37: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - normal model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
33.79	25.21

Table 5.38: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - normal model

Simulation with $n = 50$ and $(\alpha, \beta) = (2, 1.385)$

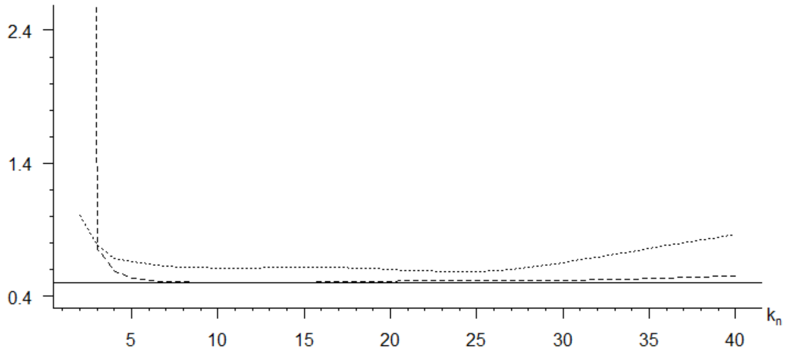

Figure 5.20: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 40$, obtained from 1000 Z samples of size $n = 50$, where $R = 0.5$ (full line) - normal model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
5	3.85×10^{-1}	3.85×10^{-1}
10	2.47×10^{-1}	2.35×10^{-1}
15	1.99×10^{-1}	1.89×10^{-1}
20	1.76×10^{-1}	1.58×10^{-1}
25	1.57×10^{-1}	1.39×10^{-1}
30	1.45×10^{-1}	1.64×10^{-1}
35	1.42×10^{-1}	1.93×10^{-1}
40	1.43×10^{-1}	2.20×10^{-1}

Table 5.39: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 50$ - normal model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
18.49	12.92

Table 5.40: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 50$ and $l = 5$ - normal model

Simulation with $n = 100$ and $(\alpha, \beta) = (2, 1.385)$

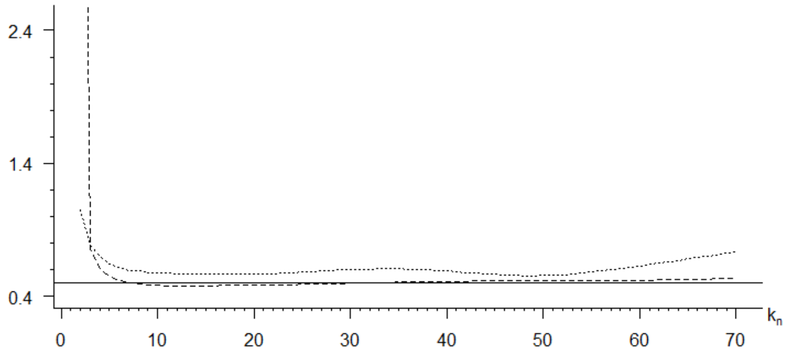

Figure 5.21: Mean of the $\widehat{R}$ estimates (dashed line) and Hill estimates (dotted line) as a function of $k_n = 2, \ldots, 70$, obtained from 1000 Z samples of size $n = 100$, where $R = 0.5$ (full line) - normal model

k_n	Standard deviation $(\widehat{R}(k_n))$	Standard deviation $(\widehat{H}^{-1}(k_n))$
10	2.18×10^{-1}	2.01×10^{-1}
20	1.51×10^{-1}	1.34×10^{-1}
30	1.29×10^{-1}	1.23×10^{-1}
40	1.18×10^{-1}	1.04×10^{-1}
50	1.05×10^{-1}	8.31×10^{-2}
60	9.70×10^{-2}	1.02×10^{-1}
70	9.52×10^{-2}	1.20×10^{-1}

Table 5.41: Standard deviations of $\widehat{R}(k_n)$ and $\widehat{H}^{-1}(k_n)$ for some values of k_n, over the 1000 repetitions, with $n = 100$ - normal model

Mean $(\tilde{k})$	Standard deviation $(\tilde{k})$
37.53	26.99

Table 5.42: Mean and standard deviation of $\tilde{k}$, over the 1000 repetitions, with $n = 100$ and $l = 5$ - normal model

5.4 Analysis of the results

Analyzing the estimates obtained for the different models presented previously, for the several sample sizes (n) and different parameters, we see that, as expected, the sample behavior of the estimators depends heavily on the number k_n of largest observations considered in the estimation. For all the situations analyzed, except for very small values of k_n, the estimates obtained for the Hill estimator are higher than those obtained for $\widehat{R}$. Moreover, we may observe that the curves obtained for the Hill estimator are less "smooth" then those obtained for the geometric estimator. The Hill estimator presents a more irregular behavior specially for small samples.

Regarding the exponential model, in the different situations studied, it is important to note that with the increase of the sample size, the estimates obtained for $\widehat{R}(k_n)$, excluding the ones referring to very small values of k_n, are very close to the real value of R (better performance than the estimates obtained by the Hill estimator).

In the case of the non-exponential models, the geometric estimator presents also a better performance than the Hill estimator. Even for smaller sample sizes, the estimates given by $\widehat{R}(k_n)$ are quite reliable for a wide range of values of k_n.

Concerning the standard deviations of the estimates we note that, in general, in the exponential model, they are smaller for $\widehat{R}$ than for $\widehat{H}^{-1}$ for intermediate and higher values of k_n. As the sample size increases, the standard deviations for $\widehat{R}$ are smaller than the ones for $\widehat{H}^{-1}$ essentially for higher values of k_n.

The standard deviations of the estimates in the non-exponential models are smaller for $\widehat{R}$ than for $\widehat{H}^{-1}$ mostly for higher values of k_n, in particular in the degenerate and normal models.

In general, as expected, the standard deviations of the estimates corresponding to the two considered estimators decrease with the increase of the sample size.

Regarding the estimates for the different parameters, we observe that the above conclusions remain valid for the different models.

The empirical method for the choice of k_n, as corroborated by the graphical analysis, allows us to obtain very good estimates for $\widehat{R}(k_n)$, even for small sample sizes.

Chapter 6

Conclusions

The theoretical basis of this Thesis is framed in the Risk Theory, in the context of the insurance activity. Within that framework, we analyzed the Ruin Theory and, in particular, we studied the ruin probability in the Cramér-Lundberg model. One of the important results in this context and pointed out in this work is the Lundberg inequality, which allows us to reduce the problem of estimating an upper bound for the ruin probability to the estimation of the adjustment coefficient.

The main purpose of this Thesis was precisely the study of the problem of estimating the adjustment coefficient. With this objective, it was presented an estimator existing in the literature for the exponential tail coefficient. The problem of estimating this tail coefficient has been studied in a variety of fields, such as in hydrology, finance, telecommunications, geology, climatology and insurance. Several estimators for the exponential tail coefficient have been proposed using different methods. In this Thesis, we will give special attention to a geometric-type estimator.

One of the objectives of this work was precisely to show that the geometric estimator, under general conditions, can be applied to the estimation of the adjustment coefficient and that, as an estimator of this coefficient, is consistent and asymptotically normal.

After presenting the theoretical properties, we developed simulations studies in order to analyze the finite sample behavior of the geometric-type estimator and compared it with the classical Hill estimator. The *R* software code (built entirely from scratch) used in the simulation studies is presented in the annex.

The simulations were done for several parameters and some different models for the individual claims, such as the exponential, the degenerate, the uniform and the normal one. We note that, in the literature, before this study and regarding the geometric estimator, there was only a small simulation study for the exponential model and even in the scope of this model, that study was quite limited because it considered just one particular parameter.

From the results of the simulations we can highlight:

- regarding the obtained estimates for the different models, using different sample sizes and different parameters, it is noted that the sample behavior of the estimators depends heavily on the number k_n of observations considered in the estimation;

- since we are doing simulations, we know the exact value of the coefficient R to be estimated. Except for very small values of k_n, the estimates of $\widehat{R}(k_n)$ are very close to the real R;

- the previous conclusion is valid for the different models analyzed, although the exponential model is slightly more sensitive to the considered sample size; in the non-exponential models, even for smaller sample sizes, the estimates of $\widehat{R}(k_n)$ are very good;

- the estimates obtained by the geometric estimator present a better performance than the ones computed by the Hill estimator;

- in the exponential model, the standard deviations of the obtained estimates are, generally, smaller for $\widehat{R}$ than for $\widehat{H}^{-1}$ for intermediate and higher values of k_n. Usually, as the sample size increases, the standard deviations for $\widehat{R}$ are smaller than the ones for $\widehat{H}^{-1}$, essentially for higher values of k_n;

- in the non-exponential models, the standard deviations for $\widehat{R}$ are smaller than the ones for $\widehat{H}^{-1}$, mostly for high values of k_n;

- the standard deviation of the estimates of the two considered estimators decreases with the increase of the sample size;

- the referred conclusions for the estimates remain valid even with the change of the parameters in the several models;

- as previously mentioned, the sample behavior of the estimators depends heavily on the number k_n of observations considered in the estimation, so the choice of k_n is particularly relevant. The implemented empirical method allowed us to obtain very good estimates for $\widehat{R}(k_n)$, even for small sample sizes.

Bibliography

Beard, R. E., Pentikäinen, T., and Pesonen, E. (1984). *Risk Theory (3rd edition)*. Chapman and Hall Ltd., London.

Bowers, N., Gerber, H., Hickman, J., Jones, D., and Nesbitt, C. (1997). *Actuarial Mathematics*. The Society of Actuaries.

Brito, M. and Freitas, A. C. M. (2003). Limiting behaviour of a geometric-type estimator for tail indices. *Insurance: Mathematics and Economics*, (33):211–226.

Brito, M. and Freitas, A. C. M. (2010). Consistent estimation of the tail index for dependent data. *Statistics and Probabability Letters*, (80):1835–1843.

Centeno, M. L. (2003). *Teoria do risco na actividade seguradora*. Celta Editora.

Cohen, J. W. (1969). *The single server queue*. North-Holland, Amsterdam.

Csörgő, M. and Steinebach, J. (1991). On the estimation of the adjustment coefficient in risk theory via intermediate order statistics. *Insurance Math. Econom.*, 10(1):37–50.

Csörgő, S. and Viharos, L. (1998). Estimating the tail index. *Asymptotic Methods in Probability and Statistics, (B. Szyszkowicz, ed), North Holland, Amesterdam*, pages 833–881.

Embrechts, P. and Mikosch, T. (1991). A bootstrap procedure for estimating the adjustment coeficient. *Insurance: Mathematics and Economics*, (10):181–190.

Fraga Alves, M. I. (1997). *Introdução à Teoria do Risco. Working paper n.º 62*. CEMAPRE.

Herkenrath, U. (1986). On the estimation of the adjustment coeficient in risk theory by means of stochastic approximation procedures. *Insurance: Mathematics and Economics*, (5):305–313.

Hill, B. M. (1975). A simple general approach to inference about the tail of a distribution. *Ann. Statist.*, 3(5):1163–1174.

Mikosch, T. (2009). *Non-life insurance Mathematics - An Introduction with the Poisson process*. Springer-Verlag Berlin Heidelberg.

Paul Deheuvels, J. S. (1990). On some alternative estimates of the adjustment coeficient in risk theory. *Scandinavian Actuarial Journal*, pages 135–159.

Pitts, S. M., Grübel, R., and Embrechts, P. (1996). Confidence bounds for the adjustment coefficient. *Advances in Applied Probability*, 28(3):802–827.

Rolski, T., Schmidli, H., Schmidt, V., and Teugels, J. (1999). *Stochastic Processes for Insurance and Finance*. John Wiley and Sons Ltd.

Schultze, J. and Steinebach, J. (1996). On least squares estimates of an exponential tail coefficient. *Statist. Decisions*, 14(4):353–372.

Annex A

R code of the simulation studies

A.1 Exponential model

```
#Cálculo do R real - Modelo exponencial

#Pressupostos
#Inserir valores

alfa_cal_r <-
beta_cal_r <-
r_real_cal <-

#Fórmula do R real

r_real_cal <- (alfa_cal_r - beta_cal_r)/(alfa_cal_r*beta_cal_r)

#Função para determinar parametros

funcao_cal_parametros <- function(alfa_cal_r, beta_cal_r, r_real
   _cal)
  {(alfa_cal_r - beta_cal_r)/(alfa_cal_r*beta_cal_r) - r_real_
     cal}

#Obter beta com base no valor de alfa e R
```

```
beta_cal_r <- uniroot(funcao_cal_parametros, alfa_cal_r = alfa_cal_r, r_real_cal = r_real_cal, lower = 0, upper = 1000000)$root

########################################

#Estimativas de R com base no estimador geométrico

#Pressupostos
#Inserir valores

num_rep_exp <-
n <-
kn <- n
vals_kn <- c(2:(kn-1))
alfa <-
beta <-
r_real <-
l <-

#Gerar amostra individual
#Gerar amostra ordenada u de tamanho n a partir da distribuição Uniforme

set.seed(num_rep_exp)

gera_amostra <- function(n,alfa,beta){

  u <- rep(0,n)
  for (i in n:1) {
    if(i==n) u[i] <- runif(1)^(1/n)
    else u[i] <- u[i+1]*runif(1)^(1/i)
  }

  #Obter s a partir de u
  s <- 1- u

  #Definição de a
  a <- beta/alfa
  #Obtenção da amostra z a partir da fórmula da inversa
  z <- ifelse(s < a/(1+a),beta/(1-a)*log(a*(1-a)/s + a^2),0)
  z
}

#Geração da amostra completa

gera_amostra_comp <- replicate(num_rep_exp, gera_amostra(n,alfa,
  beta))

#Gerar a matriz com o estimador do r geométrico para cada kn e
  cada experiência

tab_global_result_r_geo <- matrix(0, length(vals_kn), num_rep_
  exp)
rownames(tab_global_result_r_geo) <- vals_kn
for (q in 1:num_rep_exp) {

#Função que gera o r geométrico para cada kn e cada experiência

gera_r_geo_kn <- function(kn,n){

#Função que gera o estimador do r geométrico para cada kn

gera_r_geo_por_kn <- function(kn,n){

#Obter as kn maiores observações de z

z_kn <- gera_amostra_comp[(n-kn+1):n,q]

#Aplicação da fórmula do estimador geométrico

vec_aux <- c(n/(1:kn))

r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_aux))
  ^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

r_geo_est

}
#Resultados para diferentes valores do kn

resul_r_geo_kn <- sapply(vals_kn,gera_r_geo_por_kn,n)

resul_r_geo_kn

}

tab_global_result_r_geo[,q] <- gera_r_geo_kn(kn,n)

}

#Resultados r geométrico

vec_resul_r_geo_media <- rowMeans(tab_global_result_r_geo)

vec_resul_r_geo_desvpadrao <- apply(tab_global_result_r_geo,1,sd)

r_geo_est_nomes<-list(k=c(vals_kn), Estimativas_R_geométrico=c("Média", "Desvio␣padrão"))
r_geo_result_dif_kn <- matrix(cbind(vec_resul_r_geo_media,vec_resul_r_geo_desvpadrao),length(vals_kn),2, dimnames = r_geo_est_nomes)

##################################################

#Estimativas de R com base no estimador de Hill
#Utilização dos mesmos pressupostos e da mesma amostra Z

#Gerar a matriz com o estimador do r com base no estimador de Hill para cada kn e cada experiência

tab_global_result_r_hill <- matrix(0, length(vals_kn), num_rep_exp)
rownames(tab_global_result_r_hill) <- vals_kn
for (g in 1:num_rep_exp) {

  #Função que gera o r Hill para cada kn e cada experiência
  gera_r_hill_kn <- function(kn,n){
  #Função que gera o estimador do r Hill para cada kn
    gera_r_hill_por_kn <- function(kn,n){
  #Definição de z
  z <- gera_amostra_comp[,g]

  #Obter as kn maiores observações de z
  z_kn <- gera_amostra_comp[(n-kn+1):n,g]
  #Aplicação da fórmula do estimador Hill
  r_hill_est <- ((1/kn)*sum(z_kn)-z[n-kn])^(-1)
  r_hill_est
  }
  #Resultados para diferentes valores do kn
  resul_r_hill_kn <- sapply(vals_kn,gera_r_hill_por_kn,n)
  resul_r_hill_kn
  }
  tab_global_result_r_hill[,g] <- gera_r_hill_kn(kn,n)
}
#Resultados r Hill
vec_resul_r_hill_media <- rowMeans(tab_global_result_r_hill)
```

```
vec_resul_r_hill_desvpadrao <- apply(tab_global_result_r_hill ,1 ,
    sd)

r_hill_est_nomes<-list(k=c(vals_kn), Estimativas_R_Hill=c("Média
    ", "Desvio␣padrão"))
r_hill_result_dif_kn <- matrix(cbind(vec_resul_r_hill_media,vec_
    resul_r_hill_desvpadrao),length(vals_kn),2, dimnames = r_hill
    _est_nomes)

####################################################

#Tabela global resultados

tab_global_result <- cbind(r_geo_result_dif_kn,r_hill_result_dif
    _kn)
colnames(tab_global_result) <- c("Média␣est␣geo","DP␣est␣geo","
    Média␣est␣Hill␣","DP␣est␣Hill")


####################################################

#Método empírico para a escolha de kn
#Utilização dos mesmos pressupostos e da mesma amostra Z

#Construção do vector dos resultados do kn optimo para as
    diferentes repetições da experiência

vec_resul_kn_optimo <- rep(0,num_rep_exp)
for (j in 1:num_rep_exp) {

#Função que gera o kn optimo

gera_kn_optimo <- function(kn,n){

#Função que gera a média das áreas dos rectângulos para cada kn

  ger_media_area_rect <- function(kn,n){


#Obter as kn maiores observações de z
```

```
    z_kn <- gera_amostra_comp[(n-kn+1):n, j]

#Ordenação decrescente do vector z_kn

z_kn_ord <- sort(z_kn, decreasing = TRUE)

#Aplicação da fórmula do estimador geométrico

    vec_aux <- c(n/(1:kn))

    r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
       aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

#Definições de a_est_kn, b_est_kn e d_est_kn

    a_est_kn <- 1/r_geo_est

    b_est_kn <- 1/kn*(sum(z_kn_ord)-(sum(log(vec_aux))/r_geo_est
       ))

    d_est_kn <- 1/kn*(r_geo_est*sum(z_kn_ord)-(sum(log(vec_aux))
       ))

    #Fórmula média das áreas dos rectângulos kn

    media_area_rect <- 1/kn*(sum((log(vec_aux) - r_geo_est*z_kn_
       ord + d_est_kn)*(a_est_kn*log(vec_aux) +b_est_kn - z_kn_
       ord)))


    media_area_rect

  }


   #Resultados para diferentes valores de kn

  media_area_rect_dif_kn <- sapply(vals_kn,ger_media_area_rect,n
     )
  #Determinação do kn óptimo

  kn_optimo <- 1 -1 + match(min(tail(media_area_rect_dif_kn,
      length(vals_kn) +2 - 1)),tail(media_area_rect_dif_kn,
      length(vals_kn) +2 - 1))

  kn_optimo
}
vec_resul_kn_optimo[j] <- gera_kn_optimo(kn,n)
}


#Resultados do cálculo do kn óptimo

kn_optimo_media <- mean(vec_resul_kn_optimo)

kn_optimo_desvp <- sd(vec_resul_kn_optimo)

kn_optimo_result <- c(kn_optimo_media,kn_optimo_desvp)


#Gerar o vector do estimador do r geométrico para cada kn óptimo

vec_resul_r_geo_kn_optimo <- rep(0,num_rep_exp)
for (w in 1:num_rep_exp) {

#Função que gera o r geométrico para cada kn optimo

  gera_r_geo_kn_optimo <- function(kn,n){

#Função que gera o estimador do r geométrico para cada kn

gera_r_geo_por_kn <- function(kn,n){

#Obter as kn maiores observações de z

z_kn <- gera_amostra_comp[(n-kn+1):n,w]

#Aplicação da fórmula do estimador geométrico
vec_aux <- c(n/(1:kn))

r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_aux))
    ^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

r_geo_est

}

#Resultados para diferentes valores do kn óptimo

resul_r_geo_kn_optimo <- sapply(vec_resul_kn_optimo,gera_r_geo_
    por_kn,n)

resul_r_geo_kn_optimo[w]

}

vec_resul_r_geo_kn_optimo[w] <- gera_r_geo_kn_optimo(kn,n)

}

#Resultados do R geométrico decorrentes do kn óptimo

r_geo_kn_optimo_media <- mean(vec_resul_r_geo_kn_optimo)

r_geo_kn_optimo_desvp <- sd(vec_resul_r_geo_kn_optimo)

r_geo_kn_optimo_result <- c(r_geo_kn_optimo_media,r_geo_kn_
    optimo_desvp)

#Resultados global

tab_global_result_kn_optimo <- matrix(c(r_geo_kn_optimo_result,
    kn_optimo_result),1,4)
colnames(tab_global_result_kn_optimo) <- c("Média␣R␣est␣geo␣kn␣
    óptimo","DP␣R␣est␣geo␣kn␣óptimo","Média␣kn␣óptimo","DP␣kn␣
    óptimo")

#####################################################
```

```
#Resultados Tese

#Pressupostos
#Inserir valores

num_rep_exp <-
n <-
kn <- n
vals_kn_output <-
sequencia_vals_kn <-
vals_r_est_output <-
alfa <-
beta <-
r_real <-
l <-

#Gráfico
#Linha a cheio R real
#Tracejado R estimado geométrico
#Ponteado R estimado Hill
#Ajustar eixo dos y - axis 2 (by); rug (by)

plot(vals_kn_output[2:length(vals_kn_output)],r_geo_result_dif_
   kn[vals_kn_output[1:length(vals_kn_output) - 1],"Média"],
   ylim = vals_r_est_output, lty = 2, xlab = "", ylab = "" ,
   type = "l", xaxt="n", yaxt = "n", bty ="l")
lines(vals_kn_output[2:length(vals_kn_output)],r_hill_result_dif
   _kn[vals_kn_output[1:length(vals_kn_output) - 1],"Média"],
   ylim = vals_r_est_output, lty = 3)
abline(h = r_real ,lty = 1)
axis(1, at = seq(0, length(vals_kn_output), by = sequencia_vals_
   kn))
mtext(expression(k[n]), side = 1, adj = 1.03)
rug(x = seq(0, length(vals_kn_output)), ticksize = -0.01, side =
     1, quiet = T)
axis(2, las =2, at = seq(vals_r_est_output[1], vals_r_est_output
   [2], by = 0.0001))
rug(x = seq(0, max(vals_r_est_output), by = 0.00005), ticksize =
     -0.01, side = 2, quiet = T)
```

```
#Tabelas resultados

#Utilização do package xtable e definição das respectivas opções
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")

#Tabela resultados desvio padrão

result_dp_output <- tab_global_result[seq(sequencia_vals_kn - 1,
     max(vals_kn_output)-1, by = sequencia_vals_kn), c(2,4)]
colnames(result_dp_output) <- c("Desvio␣padrão␣$(\\widehat{R}(k)
    )$", "Desvio␣padrão␣$(\\widehat{H}^{-1}(k))$")
tabela_dp_output <- xtable(result_dp_output,  align = c("c","c",
    "c") ,display = c("d","e","e"))
print(tabela_dp_output,sanitize.text.function=function(x){x},
    math.style.exponents = TRUE)

#Tabela resultados kn óptimo

result_kn_opt_output <- matrix(tab_global_result_kn_optimo[, c
    (3,4)], 1, 2)
colnames(result_kn_opt_output) <- c("Média␣($\\tilde{k}$)", "
    Desvio␣padrão␣($\\tilde{k}$)")
tabela_kn_opt_output <- xtable(result_kn_opt_output,  align = c(
    "c","c","c") ,display = c("f","f","f"))
print(tabela_kn_opt_output,sanitize.text.function=function(x){x
    })
```

A.2 Degenerate model

```
#Cálculo do R real - Modelo indemnizacoes constantes contagem
    Poisson

#Pressupostos
#Inserir valores

alfa_cal_r <-
r_real_cal <-

#Fórmula do R real

funcao_cal_parametros <- function(alfa_cal_r, beta_cal_r, r_real
    _cal)
{exp(r_real_cal*beta_cal_r) - alfa_cal_r*r_real_cal - 1}

#Obter beta com base no valor de alfa e R

beta_cal_r <- uniroot(funcao_cal_parametros, alfa_cal_r = alfa_
    cal_r, r_real_cal = r_real_cal, lower = 0, upper = 1000000)$
    root

################################################

#Estimativas de R com base no estimador geométrico

#Pressupostos
#Inserir valores

num_rep_exp <-
nA <-
n <-
kn <- n
vals_kn <- c(2:(kn-1))
alfa <-
beta <-
r_real <-
l <-
```

```
#Gerar amostra individual

set.seed(num_rep_exp)

gera_amostra_base <- function(nA, alfa, beta){

#Gerar amostra D de tamanho nA

d <- beta - rexp(nA, 1/alfa)

#Gerar w

w <- rep(0,nA)
  for (i in 1:nA) {
    if(i==1) w[i] <- ifelse(d[i]<0, 0, d[i])
    else w[i] <- ifelse(w[i-1] + d[i] > 0, w[i-1] + d[i],0)
   }

#Gerar nu

nu <- which(w %in% c(0))

#Gerar vector z base

z_base <- rep(0,length(nu))
for (b in 1:length(nu)) {
  if(b==1) z_base[b] <- ifelse(nu[1]==1,0,max(w[1:nu[b]]))
  else z_base[b] <- max(w[nu[b-1]:(nu[b]-1)])
  }

z_base

}

#Geração da amostra base completa

gera_amostra_base_comp <- replicate(num_rep_exp, gera_amostra_
    base(nA, alfa, beta))
```

```
#Obtenção do tamanho das amostras base geradas

tam_amostra_base_comp <- lapply(gera_amostra_base_comp,length)

tam_amostra_base_comp <- unlist(tam_amostra_base_comp, use.names
    =FALSE)

#Tamanho das amostras base geradas, média e desvio padrão

tam_amostra_base_comp_media <- mean(tam_amostra_base_comp)

tam_amostra_base_comp_desvp <- sd(tam_amostra_base_comp)

tam_amostra_base_comp_result <- c(tam_amostra_base_comp_media,
    tam_amostra_base_comp_desvp)

names(tam_amostra_base_comp_result) <- c("Média␣tamanho␣amostra␣
    base","DP␣tamanho␣amostra␣base")

#Geração da amostra a utilizar (Z) em função do n assumido como
    pressuposto

gera_amostra_comp <- matrix(0, n, num_rep_exp)

for (h in 1:num_rep_exp) {

  #Considerar as n primeiras entradas

  z_n_ent <- gera_amostra_base_comp[[h]][1:n]

  #Ordenar o vector anterior por ordem crescente

  z <- sort(z_n_ent)

  gera_amostra_comp[,h] <- z

}

#Gerar a matriz com o estimador do r geométrico para cada kn e
    cada experiência
```

```
tab_global_result_r_geo <- matrix(0, length(vals_kn), num_rep_
    exp)
rownames(tab_global_result_r_geo) <- vals_kn
for (q in 1:num_rep_exp) {

  #Função que gera o r geométrico para cada kn e cada
      experiência

  gera_r_geo_kn <- function(kn,n){

    #Função que gera o estimador do r geométrico para cada kn

    gera_r_geo_por_kn <- function(kn,n){

      #Obter as kn maiores observações de z

      z_kn <- gera_amostra_comp[(n-kn+1):n,q]

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))

      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
          aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      r_geo_est

    }

    #Resultados para diferentes valores do kn

    resul_r_geo_kn <- sapply(vals_kn,gera_r_geo_por_kn,n)

    resul_r_geo_kn

  }

  tab_global_result_r_geo[,q] <- gera_r_geo_kn(kn,n)
```

```
}

#Resultados r geométrico

vec_resul_r_geo_media <- rowMeans(tab_global_result_r_geo)

vec_resul_r_geo_desvpadrao <- apply(tab_global_result_r_geo,1,sd
    )

r_geo_est_nomes<-list(k=c(vals_kn), Estimativas_R_geométrico=c("
    Média", "Desvio␣padrão"))
r_geo_result_dif_kn <- matrix(cbind(vec_resul_r_geo_media,vec_
    resul_r_geo_desvpadrao),length(vals_kn),2, dimnames = r_geo_
    est_nomes)

###########################################

#Estimativas de R com base no estimador de Hill
#Utilização dos mesmos pressupostos e da mesma amostra Z

#Gerar a matriz com o estimador do r com base no estimador de
    Hill para cada kn e cada experiência

tab_global_result_r_hill <- matrix(0, length(vals_kn), num_rep_
    exp)
rownames(tab_global_result_r_hill) <- vals_kn
for (g in 1:num_rep_exp) {

  #Função que gera o r Hill para cada kn e cada experiência

  gera_r_hill_kn <- function(kn,n){

    #Função que gera o estimador do r Hill para cada kn

    gera_r_hill_por_kn <- function(kn,n){

      #Definição de z

      z <- gera_amostra_comp[,g]
      #Obter as kn maiores observações de z
      z_kn <- gera_amostra_comp[(n-kn+1):n,g]
      #Aplicação da fórmula do estimador Hill
      r_hill_est <- ((1/kn)*sum(z_kn)-z[n-kn])^(-1)
      r_hill_est
    }
    #Resultados para diferentes valores do kn
    resul_r_hill_kn <- sapply(vals_kn,gera_r_hill_por_kn,n)
    resul_r_hill_kn
  }
  tab_global_result_r_hill[,g] <- gera_r_hill_kn(kn,n)
}
#Resultados r Hill
vec_resul_r_hill_media <- rowMeans(tab_global_result_r_hill)
vec_resul_r_hill_desvpadrao <- apply(tab_global_result_r_hill ,1, sd)
r_hill_est_nomes<-list(k=c(vals_kn), Estimativas_R_Hill=c("Média", "Desvio␣padrão"))
r_hill_result_dif_kn <- matrix(cbind(vec_resul_r_hill_media,vec_resul_r_hill_desvpadrao),length(vals_kn),2, dimnames = r_hill_est_nomes)
###########################################
```

```
#Tabela global resultados

tab_global_result <- cbind(r_geo_result_dif_kn,r_hill_result_dif
    _kn)
colnames(tab_global_result) <- c("Média␣est␣geo","DP␣est␣geo","
    Média␣est␣Hill␣","DP␣est␣Hill")

##########################################

#Método empírico para a escolha de kn
#Utilização dos mesmos pressupostos e da mesma amostra Z

#Construção do vector dos resultados do kn optimo para as
    diferentes repetições da experiência

vec_resul_kn_optimo <- rep(0,num_rep_exp)
for (j in 1:num_rep_exp) {

  #Função que gera o kn optimo

  gera_kn_optimo <- function(kn,n){

    #Função que gera a média das áreas dos rectângulos para cada
        kn

    ger_media_area_rect <- function(kn,n){


      #Obter as kn maiores observações de z
      z_kn <- gera_amostra_comp[(n-kn+1):n,j]

      #Ordenação decrescente do vector z_kn

      z_kn_ord <- sort(z_kn, decreasing = TRUE)

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))

      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
         aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      #Definições de a_est_kn, b_est_kn e d_est_kn

      a_est_kn <- 1/r_geo_est

      b_est_kn <- 1/kn*(sum(z_kn_ord)-(sum(log(vec_aux))/r_geo_
         est))

      d_est_kn <- 1/kn*(r_geo_est*sum(z_kn_ord)-(sum(log(vec_aux
         ))))

      #Fórmula média das áreas dos rectângulos kn

      media_area_rect <- 1/kn*(sum((log(vec_aux) - r_geo_est*z_
         kn_ord + d_est_kn)*(a_est_kn*log(vec_aux) +b_est_kn - z
         _kn_ord)))


      media_area_rect

    }


    #Resultados para diferentes valores de kn

    media_area_rect_dif_kn <- sapply(vals_kn,ger_media_area_rect
       ,n)

    #Determinação do kn óptimo

    kn_optimo <- 1 -1 + match(min(tail(media_area_rect_dif_kn,
       length(vals_kn) +2 - 1)),tail(media_area_rect_dif_kn,
       length(vals_kn) +2 - 1))

    kn_optimo
  }
  vec_resul_kn_optimo[j] <- gera_kn_optimo(kn,n)
}
```

```
#Resultados do cálculo do kn óptimo

kn_optimo_media <- mean(vec_resul_kn_optimo)

kn_optimo_desvp <- sd(vec_resul_kn_optimo)

kn_optimo_result <- c(kn_optimo_media,kn_optimo_desvp)


#Gerar o vector do estimador do r geométrico para cada kn óptimo

vec_resul_r_geo_kn_optimo <- rep(0,num_rep_exp)
for (w in 1:num_rep_exp) {

  #Função que gera o r geométrico para cada kn optimo

  gera_r_geo_kn_optimo <- function(kn,n){

    #Função que gera o estimador do r geométrico para cada kn

    gera_r_geo_por_kn <- function(kn,n){

      #Obter as kn maiores observações de z

      z_kn <- gera_amostra_comp[(n-kn+1):n,w]

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))

      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
         aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      r_geo_est

    }

    #Resultados para diferentes valores do kn óptimo
    resul_r_geo_kn_optimo <- sapply(vec_resul_kn_optimo,gera_r_
        geo_por_kn,n)

    resul_r_geo_kn_optimo[w]

  }

  vec_resul_r_geo_kn_optimo[w] <- gera_r_geo_kn_optimo(kn,n)

}

#Resultados do R geométrico decorrentes do kn óptimo

r_geo_kn_optimo_media <- mean(vec_resul_r_geo_kn_optimo)

r_geo_kn_optimo_desvp <- sd(vec_resul_r_geo_kn_optimo)

r_geo_kn_optimo_result <- c(r_geo_kn_optimo_media,r_geo_kn_
    optimo_desvp)

#Resultados global

tab_global_result_kn_optimo <- matrix(c(r_geo_kn_optimo_result,
    kn_optimo_result),1,4)
colnames(tab_global_result_kn_optimo) <- c("Média␣R␣est␣geo␣kn␣
    óptimo","DP␣R␣est␣geo␣kn␣óptimo","Média␣kn␣óptimo","DP␣kn␣
    óptimo")

##########################################

#Resultados Tese

#Pressupostos
#Inserir valores

num_rep_exp <-
n <-
kn <- n
vals_kn_output <-
sequencia_vals_kn <-
```

```
vals_r_est_output <-
alfa <-
beta <-
r_real <-
l <-

#Gráfico
#Linha a cheio R real
#Tracejado R estimado geométrico
#Ponteado R estimado Hill
#Ajustar eixo dos y - axis 2 (by); rug (by)

plot(vals_kn_output[2:length(vals_kn_output)],r_geo_result_dif_
   kn[vals_kn_output[1:length(vals_kn_output) - 1],"Média"],
   ylim = vals_r_est_output, lty = 2, xlab = "", ylab = "" ,
   type = "l", xaxt="n", yaxt = "n", bty ="l")
lines(vals_kn_output[2:length(vals_kn_output)],r_hill_result_dif
   _kn[vals_kn_output[1:length(vals_kn_output) - 1],"Média"],
   ylim = vals_r_est_output, lty = 3)
abline(h = r_real,lty = 1)
axis(1, at = seq(0, length(vals_kn_output), by = sequencia_vals_
   kn))
mtext(expression(k[n]), side = 1, adj = 1.03)
rug(x = seq(0, length(vals_kn_output)), ticksize = -0.01, side =
    1, quiet = T)
axis(2, las =2, at = seq(vals_r_est_output[1], vals_r_est_output
   [2], by = 0.0001))
rug(x = seq(0, max(vals_r_est_output), by = 0.00005), ticksize =
    -0.01, side = 2, quiet = T)

#Tabelas resultados

#Utilização do package xtable e definição das respectivas opções
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")

#Tabela resultados desvio padrão

result_dp_output <- tab_global_result[seq(sequencia_vals_kn - 1,
    max(vals_kn_output)-1, by = sequencia_vals_kn), c(2,4)]
colnames(result_dp_output) <- c("Desvio␣padrão␣$(\\widehat{R}(k)
    )$", "Desvio␣padrão␣$(\\widehat{H}^{-1}(k))$")
tabela_dp_output <- xtable(result_dp_output,  align = c("c","c",
    "c") ,display = c("d","e","e"))
print(tabela_dp_output,sanitize.text.function=function(x){x},
    math.style.exponents = TRUE)

#Tabela resultados kn óptimo

result_kn_opt_output <- matrix(tab_global_result_kn_optimo[, c
    (3,4)], 1, 2)
colnames(result_kn_opt_output) <- c("Média␣($\\tilde{k}$)", "
    Desvio␣padrão␣($\\tilde{k}$)")
tabela_kn_opt_output <- xtable(result_kn_opt_output,  align = c(
    "c","c","c") ,display = c("f","f","f"))
print(tabela_kn_opt_output,sanitize.text.function=function(x){x
    })
```

A.3 Uniform model

```
#Cálculo do R real - Modelo Uniforme

#Pressupostos
#Inserir valores

alfa_cal_r <-
a_cal <-
r_real_cal <-

#Fórmula do R real

funcao_cal_parametros <- function(alfa_cal_r, a_cal, b_cal, r_
    real_cal)
{(exp(r_real_cal*b_cal) - exp(r_real_cal*a_cal))/(r_real_cal*(b_
    cal - a_cal)) - alfa_cal_r*r_real_cal - 1}

#Obter b com base no valor de alfa, a e R

b_cal <- uniroot(funcao_cal_parametros, alfa_cal_r = alfa_cal_r,
     a_cal = a_cal, r_real_cal = r_real_cal, lower = 0.1, upper =
     1000000)$root

##################################################

#Estimativas de R com base no estimador geométrico

#Pressupostos
#Inserir valores

num_rep_exp <-
nA <-
n <-
kn <- n
vals_kn <- c(2:(kn-1))
alfa <-
a_press <-
b_press <-
```

```
r_real <-
l <-

#Gerar amostra individual

set.seed(num_rep_exp)

gera_amostra_base <- function(nA, alfa ,a_press ,b_press){

#Gerar amostra D de tamanho nA

d <- runif(nA, min = a_press , max = b_press) - rexp(nA, 1/alfa)

#Gerar w

w <- rep(0,nA)
  for (i in 1:nA) {
    if(i==1) w[i] <- ifelse(d[i]<0, 0, d[i])
    else w[i] <- ifelse(w[i-1] + d[i] > 0, w[i-1] + d[i],0)
   }

#Gerar nu

nu <- which(w %in% c(0))

#Gerar vector z base

z_base <- rep(0,length(nu))
for (b in 1:length(nu)) {
  if(b==1) z_base[b] <- ifelse(nu[1]==1,0,max(w[1:nu[b]]))
  else z_base[b] <- max(w[nu[b-1]:(nu[b]-1)])
  }

z_base

}

#Geração da amostra base completa

gera_amostra_base_comp <- replicate(num_rep_exp, gera_amostra_
    base(nA, alfa ,a_press ,b_press))

#Obtenção do tamanho das amostras base geradas

tam_amostra_base_comp <— lapply(gera_amostra_base_comp,length)

tam_amostra_base_comp <— unlist(tam_amostra_base_comp, use.names
    =FALSE)

#Tamanho das amostras base geradas, média e desvio padrão

tam_amostra_base_comp_media <— mean(tam_amostra_base_comp)

tam_amostra_base_comp_desvp <— sd(tam_amostra_base_comp)

tam_amostra_base_comp_result <— c(tam_amostra_base_comp_media,
    tam_amostra_base_comp_desvp)

names(tam_amostra_base_comp_result) <— c("Média␣tamanho␣amostra␣
    base","DP␣tamanho␣amostra␣base")

#Geração da amostra a utilizar (Z) em função do n assumido como
    pressuposto

gera_amostra_comp <— matrix(0, n, num_rep_exp)

for (h in 1:num_rep_exp) {

  #Considerar as n primeiras entradas

  z_n_ent <— gera_amostra_base_comp[[h]][1:n]

  #Ordenar o vector anterior por ordem crescente

  z <— sort(z_n_ent)

  gera_amostra_comp[,h] <— z

}
```

```
#Gerar a matriz com o estimador do r geométrico para cada kn e
   cada experiência

tab_global_result_r_geo <- matrix(0, length(vals_kn), num_rep_
   exp)
rownames(tab_global_result_r_geo) <- vals_kn
for (q in 1:num_rep_exp) {

  #Função que gera o r geométrico para cada kn e cada
     experiência

  gera_r_geo_kn <- function(kn,n){

    #Função que gera o estimador do r geométrico para cada kn

    gera_r_geo_por_kn <- function(kn,n){

      #Obter as kn maiores observações de z

      z_kn <- gera_amostra_comp[(n-kn+1):n,q]

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))

      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
         aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      r_geo_est

    }

    #Resultados para diferentes valores do kn

    resul_r_geo_kn <- sapply(vals_kn,gera_r_geo_por_kn,n)

    resul_r_geo_kn

  }
  tab_global_result_r_geo[,q] <- gera_r_geo_kn(kn,n)

}

#Resultados r geométrico

vec_resul_r_geo_media <- rowMeans(tab_global_result_r_geo)

vec_resul_r_geo_desvpadrao <- apply(tab_global_result_r_geo,1,sd
   )

r_geo_est_nomes<-list(k=c(vals_kn), Estimativas_R_geométrico=c("
   Média", "Desvio␣padrão"))
r_geo_result_dif_kn <- matrix(cbind(vec_resul_r_geo_media,vec_
   resul_r_geo_desvpadrao),length(vals_kn),2, dimnames = r_geo_
   est_nomes)

################################################

#Estimativas de R com base no estimador de Hill
#Utilização dos mesmos pressupostos e da mesma amostra Z

#Gerar a matriz com o estimador do r com base no estimador de
   Hill para cada kn e cada experiência

tab_global_result_r_hill <- matrix(0, length(vals_kn), num_rep_
   exp)
rownames(tab_global_result_r_hill) <- vals_kn
for (g in 1:num_rep_exp) {

  #Função que gera o r Hill para cada kn e cada experiência

  gera_r_hill_kn <- function(kn,n){

    #Função que gera o estimador do r Hill para cada kn

    gera_r_hill_por_kn <- function(kn,n){

      #Definição de z
      z <- gera_amostra_comp[,g]

      #Obter as kn maiores observações de z
      z_kn <- gera_amostra_comp[(n-kn+1):n,g]
      #Aplicação da fórmula do estimador Hill
      r_hill_est <- ((1/kn)*sum(z_kn)-z[n-kn])^(-1)
      r_hill_est
    }
    #Resultados para diferentes valores do kn
    resul_r_hill_kn <- sapply(vals_kn,gera_r_hill_por_kn,n)
    resul_r_hill_kn
  }
  tab_global_result_r_hill[,g] <- gera_r_hill_kn(kn,n)
}
#Resultados r Hill
vec_resul_r_hill_media <- rowMeans(tab_global_result_r_hill)
vec_resul_r_hill_desvpadrao <- apply(tab_global_result_r_hill ,1 ,
   sd)
r_hill_est_nomes<-list(k=c(vals_kn), Estimativas_R_Hill=c("Média
   ", "Desvio␣padrão"))
r_hill_result_dif_kn <- matrix(cbind(vec_resul_r_hill_media,vec_
   resul_r_hill_desvpadrao),length(vals_kn),2, dimnames = r_hill
   _est_nomes)
```

```
##################################################

#Tabela global resultados

tab_global_result <- cbind(r_geo_result_dif_kn,r_hill_result_dif_kn)
colnames(tab_global_result) <- c("Média␣est␣geo","DP␣est␣geo","Média␣est␣Hill␣","DP␣est␣Hill")

##########################################

#Método empírico para a escolha de kn
#Utilização dos mesmos pressupostos e da mesma amostra Z

#Construção do vector dos resultados do kn optimo para as diferentes repetições da experiência

vec_resul_kn_optimo <- rep(0,num_rep_exp)
for (j in 1:num_rep_exp) {

  #Função que gera o kn optimo

  gera_kn_optimo <- function(kn,n){

    #Função que gera a média das áreas dos rectângulos para cada kn

    ger_media_area_rect <- function(kn,n){


      #Obter as kn maiores observações de z
      z_kn <- gera_amostra_comp[(n-kn+1):n,j]

      #Ordenação decrescente do vector z_kn

      z_kn_ord <- sort(z_kn, decreasing = TRUE)

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))
      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      #Definições de a_est_kn, b_est_kn e d_est_kn

      a_est_kn <- 1/r_geo_est

      b_est_kn <- 1/kn*(sum(z_kn_ord)-(sum(log(vec_aux))/r_geo_est))

      d_est_kn <- 1/kn*(r_geo_est*sum(z_kn_ord)-(sum(log(vec_aux))))

      #Fórmula média das áreas dos rectângulos kn

       media_area_rect <- 1/kn*(sum((log(vec_aux) - r_geo_est*z_kn_ord + d_est_kn)*(a_est_kn*log(vec_aux) +b_est_kn - z_kn_ord)))


      media_area_rect

    }


    #Resultados para diferentes valores de kn

    media_area_rect_dif_kn <- sapply(vals_kn,ger_media_area_rect,n)

    #Determinação do kn óptimo

    kn_optimo <- 1 -1 + match(min(tail(media_area_rect_dif_kn, length(vals_kn) +2 - 1)),tail(media_area_rect_dif_kn, length(vals_kn) +2 - 1))

    kn_optimo
  }
  vec_resul_kn_optimo[j] <- gera_kn_optimo(kn,n)
```

```
}

#Resultados do cálculo do kn óptimo

kn_optimo_media <- mean(vec_resul_kn_optimo)

kn_optimo_desvp <- sd(vec_resul_kn_optimo)

kn_optimo_result <- c(kn_optimo_media,kn_optimo_desvp)


#Gerar o vector do estimador do r geométrico para cada kn óptimo

vec_resul_r_geo_kn_optimo <- rep(0,num_rep_exp)
for (w in 1:num_rep_exp) {

  #Função que gera o r geométrico para cada kn optimo

  gera_r_geo_kn_optimo <- function(kn,n){

    #Função que gera o estimador do r geométrico para cada kn

    gera_r_geo_por_kn <- function(kn,n){

      #Obter as kn maiores observações de z

      z_kn <- gera_amostra_comp[(n-kn+1):n,w]

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))

      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
          aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      r_geo_est

    }
    #Resultados para diferentes valores do kn óptimo

    resul_r_geo_kn_optimo <- sapply(vec_resul_kn_optimo,gera_r_
      geo_por_kn,n)

    resul_r_geo_kn_optimo[w]

  }

  vec_resul_r_geo_kn_optimo[w] <- gera_r_geo_kn_optimo(kn,n)

}

#Resultados do R geométrico decorrentes do kn óptimo

r_geo_kn_optimo_media <- mean(vec_resul_r_geo_kn_optimo)

r_geo_kn_optimo_desvp <- sd(vec_resul_r_geo_kn_optimo)

r_geo_kn_optimo_result <- c(r_geo_kn_optimo_media,r_geo_kn_
   optimo_desvp)

#Resultados global

tab_global_result_kn_optimo <- matrix(c(r_geo_kn_optimo_result,
   kn_optimo_result),1,4)
colnames(tab_global_result_kn_optimo) <- c("Média␣R␣est␣geo␣kn␣
   óptimo","DP␣R␣est␣geo␣kn␣óptimo","Média␣kn␣óptimo","DP␣kn␣
   óptimo")

#########################################

#Resultados Tese

#Pressupostos
#Inserir valores

num_rep_exp <-
n <-
kn <- n
```

```
vals_kn_output <-
sequencia_vals_kn <-
vals_r_est_output <-
alfa <-
b_press <-
r_real <-
l <-

#Gráfico
#Linha a cheio R real
#Tracejado R estimado geométrico
#Ponteado R estimado Hill
#Ajustar eixo dos y - axis 2 (by); rug (by)

plot(vals_kn_output[2:length(vals_kn_output)],r_geo_result_dif_
   kn[vals_kn_output[1:length(vals_kn_output) - 1],"Média"],
   ylim = vals_r_est_output, lty = 2, xlab = "", ylab = "" ,
   type = "l", xaxt="n", yaxt = "n", bty ="l")
lines(vals_kn_output[2:length(vals_kn_output)],r_hill_result_dif
   _kn[vals_kn_output[1:length(vals_kn_output) - 1],"Média"],
   ylim = vals_r_est_output, lty = 3)
abline(h = r_real,lty = 1)
axis(1, at = seq(0, length(vals_kn_output), by = sequencia_vals_
   kn))
mtext(expression(k[n]), side = 1, adj = 1.03)
rug(x = seq(0, length(vals_kn_output)), ticksize = -0.01, side =
    1, quiet = T)
axis(2, las =2, at = seq(vals_r_est_output[1], vals_r_est_output
   [2], by = 0.0001))
rug(x = seq(0, max(vals_r_est_output), by = 0.00005), ticksize =
    -0.01, side = 2, quiet = T)

#Tabelas resultados

#Utilização do package xtable e definição das respectivas opções
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")

#Tabela resultados desvio padrão
```

```
result_dp_output <- tab_global_result[seq(sequencia_vals_kn - 1,
    max(vals_kn_output)-1, by = sequencia_vals_kn), c(2,4)]
colnames(result_dp_output) <- c("Desvio␣padrão␣$(\\widehat{R}(k)
   )$", "Desvio␣padrão␣$(\\widehat{H}^{-1}(k))$")
tabela_dp_output <- xtable(result_dp_output,  align = c("c","c",
   "c") ,display = c("d","e","e"))
print(tabela_dp_output,sanitize.text.function=function(x){x},
   math.style.exponents = TRUE)

#Tabela resultados kn óptimo

result_kn_opt_output <- matrix(tab_global_result_kn_optimo[, c
   (3,4)], 1, 2)
colnames(result_kn_opt_output) <- c("Média␣($\\tilde{k}$)", "
   Desvio␣padrão␣($\\tilde{k}$)")
tabela_kn_opt_output <- xtable(result_kn_opt_output,  align = c(
   "c","c","c") ,display = c("f","f","f"))
print(tabela_kn_opt_output,sanitize.text.function=function(x){x
   })
```

A.4 Normal model

```
#Cálculo do R real — Modelo Normal

#Pressupostos
#Inserir valores

alfa_cal_r <-
r_real_cal <-

#Fórmula do R real

funcao_cal_parametros <- function(alfa_cal_r, beta_cal_r, r_real
    _cal)
{exp(((0.1*beta_cal_r)^2*r_real_cal^2)/2 + beta_cal_r*r_real_cal
    ) - alfa_cal_r*r_real_cal - 1}

#Obter beta com base no valor de alfa, s e R

beta_cal_r <- uniroot(funcao_cal_parametros, alfa_cal_r = alfa_
    cal_r, r_real_cal = r_real_cal, lower = 0, upper = 1000000)$
    root

##########################################

#Estimativas de R com base no estimador geométrico

#Pressupostos
#Inserir valores

num_rep_exp <-
nA <-
n <-
kn <- n
vals_kn <- c(2:(kn-1))
alfa <-
beta <-
r_real <-
l <-
```

```
#Gerar amostra individual

set.seed(num_rep_exp)

gera_amostra_base <- function(nA, alfa ,beta){

#Gerar amostra D de tamanho nA

s <- 0.1*beta

d <- rnorm(nA, mean = beta, sd = s) - rexp(nA, 1/alfa)

#Gerar w

w <- rep(0,nA)
  for (i in 1:nA) {
    if(i==1) w[i] <- ifelse(d[i]<0, 0, d[i])
    else w[i] <- ifelse(w[i-1] + d[i] > 0, w[i-1] + d[i],0)
   }

#Gerar nu

nu <- which(w %in% c(0))

#Gerar vector z base

z_base <- rep(0,length(nu))
for (b in 1:length(nu)) {
  if(b==1) z_base[b] <- ifelse(nu[1]==1,0,max(w[1:nu[b]]))
  else z_base[b] <- max(w[nu[b-1]:(nu[b]-1)])
  }

z_base

}

#Geração da amostra base completa

gera_amostra_base_comp <- replicate(num_rep_exp, gera_amostra_
    base(nA,alfa,beta))

#Obtenção do tamanho das amostras base geradas

tam_amostra_base_comp <- lapply(gera_amostra_base_comp,length)

tam_amostra_base_comp <- unlist(tam_amostra_base_comp, use.names
    =FALSE)

#Tamanho das amostras base geradas, média e desvio padrão

tam_amostra_base_comp_media <- mean(tam_amostra_base_comp)

tam_amostra_base_comp_desvp <- sd(tam_amostra_base_comp)

tam_amostra_base_comp_result <- c(tam_amostra_base_comp_media,
    tam_amostra_base_comp_desvp)

names(tam_amostra_base_comp_result) <- c("Média␣tamanho␣amostra␣
    base","DP␣tamanho␣amostra␣base")

#Geração da amostra a utilizar (Z) em função do n assumido como
    pressuposto

gera_amostra_comp <- matrix(0, n, num_rep_exp)

for (h in 1:num_rep_exp) {

  #Considerar as n primeiras entradas

  z_n_ent <- gera_amostra_base_comp[[h]][1:n]

  #Ordenar o vector anterior por ordem crescente

  z <- sort(z_n_ent)

  gera_amostra_comp[,h] <- z

}
```

```
#Gerar a matriz com o estimador do r geométrico para cada kn e
   cada experiência

tab_global_result_r_geo <- matrix(0, length(vals_kn), num_rep_
   exp)
rownames(tab_global_result_r_geo) <- vals_kn
for (q in 1:num_rep_exp) {

  #Função que gera o r geométrico para cada kn e cada
     experiência

  gera_r_geo_kn <- function(kn,n){

    #Função que gera o estimador do r geométrico para cada kn

    gera_r_geo_por_kn <- function(kn,n){

      #Obter as kn maiores observações de z

      z_kn <- gera_amostra_comp[(n-kn+1):n,q]

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))

      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
         aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      r_geo_est

    }

    #Resultados para diferentes valores do kn

    resul_r_geo_kn <- sapply(vals_kn,gera_r_geo_por_kn,n)

    resul_r_geo_kn

  }
  tab_global_result_r_geo[,q] <- gera_r_geo_kn(kn,n)

}

#Resultados r geométrico

vec_resul_r_geo_media <- rowMeans(tab_global_result_r_geo)

vec_resul_r_geo_desvpadrao <- apply(tab_global_result_r_geo,1,sd
    )

r_geo_est_nomes<-list(k=c(vals_kn), Estimativas_R_geométrico=c("
    Média", "Desvio␣padrão"))
r_geo_result_dif_kn <- matrix(cbind(vec_resul_r_geo_media,vec_
    resul_r_geo_desvpadrao),length(vals_kn),2, dimnames = r_geo_
    est_nomes)

#####################################################

#Estimativas de R com base no estimador de Hill
#Utilização dos mesmos pressupostos e da mesma amostra Z

#Gerar a matriz com o estimador do r com base no estimador de
    Hill para cada kn e cada experiência

tab_global_result_r_hill <- matrix(0, length(vals_kn), num_rep_
    exp)
rownames(tab_global_result_r_hill) <- vals_kn
for (g in 1:num_rep_exp) {

  #Função que gera o r Hill para cada kn e cada experiência

  gera_r_hill_kn <- function(kn,n){

    #Função que gera o estimador do r Hill para cada kn

    gera_r_hill_por_kn <- function(kn,n){

      #Definição de z
      z <- gera_amostra_comp[,g]

      #Obter as kn maiores observações de z
      z_kn <- gera_amostra_comp[(n-kn+1):n,g]
      #Aplicação da fórmula do estimador Hill
      r_hill_est <- ((1/kn)*sum(z_kn)-z[n-kn])^(-1)
      r_hill_est
    }
    #Resultados para diferentes valores do kn
    resul_r_hill_kn <- sapply(vals_kn,gera_r_hill_por_kn,n)
    resul_r_hill_kn
  }
  tab_global_result_r_hill[,g] <- gera_r_hill_kn(kn,n)
}
#Resultados r Hill
vec_resul_r_hill_media <- rowMeans(tab_global_result_r_hill)
vec_resul_r_hill_desvpadrao <- apply(tab_global_result_r_hill,1,
   sd)
r_hill_est_nomes<-list(k=c(vals_kn), Estimativas_R_Hill=c("Média
   ", "Desvio␣padrão"))
r_hill_result_dif_kn <- matrix(cbind(vec_resul_r_hill_media,vec_
   resul_r_hill_desvpadrao),length(vals_kn),2, dimnames = r_hill
   _est_nomes)
```

```
##################################################

#Tabela global resultados

tab_global_result <- cbind(r_geo_result_dif_kn,r_hill_result_dif
   _kn)
colnames(tab_global_result) <- c("Média␣est␣geo","DP␣est␣geo","
   Média␣est␣Hill␣","DP␣est␣Hill")

#################################################

#Método empírico para a escolha de kn
#Utilização dos mesmos pressupostos e da mesma amostra Z

#Construção do vector dos resultados do kn optimo para as
   diferentes repetições da experiência

vec_resul_kn_optimo <- rep(0,num_rep_exp)
for (j in 1:num_rep_exp) {

  #Função que gera o kn optimo

  gera_kn_optimo <- function(kn,n){

    #Função que gera a média das áreas dos rectângulos para cada
        kn

    ger_media_area_rect <- function(kn,n){


      #Obter as kn maiores observações de z
      z_kn <- gera_amostra_comp[(n-kn+1):n,j]

      #Ordenação decrescente do vector z_kn

      z_kn_ord <- sort(z_kn, decreasing = TRUE)

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))
      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
         aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      #Definições de a_est_kn, b_est_kn e d_est_kn

      a_est_kn <- 1/r_geo_est

      b_est_kn <- 1/kn*(sum(z_kn_ord)-(sum(log(vec_aux))/r_geo_
         est))

      d_est_kn <- 1/kn*(r_geo_est*sum(z_kn_ord)-(sum(log(vec_aux
         ))))

      #Fórmula média das áreas dos rectângulos kn

      media_area_rect <- 1/kn*(sum((log(vec_aux) - r_geo_est*z_
         kn_ord + d_est_kn)*(a_est_kn*log(vec_aux) +b_est_kn - z
         _kn_ord)))


      media_area_rect

    }


    #Resultados para diferentes valores de kn

    media_area_rect_dif_kn <- sapply(vals_kn,ger_media_area_rect
       ,n)

    #Determinação do kn óptimo

    kn_optimo <- 1 -1 + match(min(tail(media_area_rect_dif_kn,
       length(vals_kn) +2 - 1)),tail(media_area_rect_dif_kn,
       length(vals_kn) +2 - 1))

    kn_optimo
  }
  vec_resul_kn_optimo[j] <- gera_kn_optimo(kn,n)
```

```
}

#Resultados do cálculo do kn óptimo

kn_optimo_media <- mean(vec_resul_kn_optimo)

kn_optimo_desvp <- sd(vec_resul_kn_optimo)

kn_optimo_result <- c(kn_optimo_media,kn_optimo_desvp)


#Gerar o vector do estimador do r geométrico para cada kn óptimo

vec_resul_r_geo_kn_optimo <- rep(0,num_rep_exp)
for (w in 1:num_rep_exp) {

  #Função que gera o r geométrico para cada kn optimo

  gera_r_geo_kn_optimo <- function(kn,n){

    #Função que gera o estimador do r geométrico para cada kn

    gera_r_geo_por_kn <- function(kn,n){

      #Obter as kn maiores observações de z

      z_kn <- gera_amostra_comp[(n-kn+1):n,w]

      #Aplicação da fórmula do estimador geométrico

      vec_aux <- c(n/(1:kn))

      r_geo_est <- sqrt((sum(log(vec_aux)^2)-(1/kn)*sum(log(vec_
          aux))^2)/(sum(z_kn^2)-(1/kn)*sum(z_kn)^2))

      r_geo_est

    }
    #Resultados para diferentes valores do kn óptimo

    resul_r_geo_kn_optimo <- sapply(vec_resul_kn_optimo,gera_r_
       geo_por_kn,n)

    resul_r_geo_kn_optimo[w]

  }

  vec_resul_r_geo_kn_optimo[w] <- gera_r_geo_kn_optimo(kn,n)

}

#Resultados do R geométrico decorrentes do kn óptimo

r_geo_kn_optimo_media <- mean(vec_resul_r_geo_kn_optimo)

r_geo_kn_optimo_desvp <- sd(vec_resul_r_geo_kn_optimo)

r_geo_kn_optimo_result <- c(r_geo_kn_optimo_media,r_geo_kn_
   optimo_desvp)

#Resultados global

tab_global_result_kn_optimo <- matrix(c(r_geo_kn_optimo_result,
   kn_optimo_result),1,4)
colnames(tab_global_result_kn_optimo) <- c("Média␣R␣est␣geo␣kn␣
   óptimo","DP␣R␣est␣geo␣kn␣óptimo","Média␣kn␣óptimo","DP␣kn␣
   óptimo")

##########################################

#Resultados Tese

#Pressupostos
#Inserir valores

num_rep_exp <-
n <-
kn <- n
```

```
vals_kn_output <-
sequencia_vals_kn <-
vals_r_est_output <-
alfa <-
beta <-
r_real <-
l <-

#Gráfico
#Linha a cheio R real
#Tracejado R estimado geométrico
#Ponteado R estimado Hill
#Ajustar eixo dos y - axis 2 (by); rug (by)

plot(vals_kn_output[2:length(vals_kn_output)],r_geo_result_dif_
   kn[vals_kn_output[1:length(vals_kn_output) - 1],"Média"],
   ylim = vals_r_est_output, lty = 2, xlab = "", ylab = "" ,
   type = "l", xaxt="n", yaxt = "n", bty ="l")
lines(vals_kn_output[2:length(vals_kn_output)],r_hill_result_dif
   _kn[vals_kn_output[1:length(vals_kn_output) - 1],"Média"],
   ylim = vals_r_est_output, lty = 3)
abline(h = r_real,lty = 1)
axis(1, at = seq(0, length(vals_kn_output), by = sequencia_vals_
   kn))
mtext(expression(k[n]), side = 1, adj = 1.03)
rug(x = seq(0, length(vals_kn_output)), ticksize = -0.01, side =
    1, quiet = T)
axis(2, las =2, at = seq(vals_r_est_output[1], vals_r_est_output
   [2], by = 0.0001))
rug(x = seq(0, max(vals_r_est_output), by = 0.00005), ticksize =
    -0.01, side = 2, quiet = T)

#Tabelas resultados

#Utilização do package xtable e definição das respectivas opções
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")

#Tabela resultados desvio padrão
```

```
result_dp_output <- tab_global_result[seq(sequencia_vals_kn - 1,
    max(vals_kn_output)-1, by = sequencia_vals_kn), c(2,4)]
colnames(result_dp_output) <- c("Desvio␣padrão␣$(\\widehat{R}(k)
   )$", "Desvio␣padrão␣$(\\widehat{H}^{-1}(k))$")
tabela_dp_output <- xtable(result_dp_output,  align = c("c","c",
   "c") ,display = c("d","e","e"))
print(tabela_dp_output,sanitize.text.function=function(x){x},
   math.style.exponents = TRUE)

#Tabela resultados kn óptimo

result_kn_opt_output <- matrix(tab_global_result_kn_optimo[, c
   (3,4)], 1, 2)
colnames(result_kn_opt_output) <- c("Média␣($\\tilde{k}$)", "
   Desvio␣padrão␣($\\tilde{k}$)")
tabela_kn_opt_output <- xtable(result_kn_opt_output,  align = c(
   "c","c","c") ,display = c("f","f","f"))
print(tabela_kn_opt_output,sanitize.text.function=function(x){x
   })
```