Montecarlo Estimators of Value at Ruin and Tail Value at Ruin

Master thesis

Institute of Mathematical Statistics and Actuarial Sciences

Faculty of Science, University of Bern

handed in by

Boris Polanco

2018

Supervisor

Prof. Dr. Riccardo Gatto

Table of Contents

List of Tables

List of Figures

Abstract

In the classical Cramer Lundberg Risk model with claim amounts that follows a Weibull distribution, we consider the amount of capital required to have a fixed value of probability of ruin as risk measure (VaRu), as well as the coherent version of it, the Tail Value at Ruin (TVaRu). Both measures are based on the probability of ruin. Therefore, one of the goals of this work, is to estimate adequately this quantity, especially when its value is very low. There is a well known analytical formula that deals with the computation of this probability when the claim amounts are exponentially distributed. Then, the Weibull distributed claim amount assumption due to its parameter $\tau > 0$ allow us to consider three cases:

- The light tailed case when $\tau > 1$.
- The heavy tailed case if $\tau < 1$.
- The same as the exponential case when $\tau = 1$.

For the three cases we have performed crude Monte Carlo approaches. However, since this problem corresponds to rare event simulation, the Monte Carlo method can not be efficient. In order to avoid that, we have applied some variance reduction techniques. For the light tailed case, the importance sampling method by the exponential change of measure shows a better performance than the crude Monte Carlo. In the heavy tailed case, we have used control variates as well as conditional simulation algorithms, that works much better than the crude Monte Carlo method. At the end of this work we provide some R functions implemented that perform every step of the described methods and some numerical examples.

Introduction

The Cramer Lundberg Risk process is a stochastic model that describes the evolution of the reserve of an insurance company at time $t \geq 0$. For this model we require a claim arrival process $\{N_t\}_{t\geq 0}$ such that $N_0 = 0$ is an homogeneous Poisson process with rate $\lambda > 0$. We suppose that the claim amounts $\{X_i\}_{i\geq 0}$ are Weibull distributed and a constant premium rate $c > 0$ such that $c = (1+\beta)\lambda\mu$, where β is the security loading and μ is the mean of the claim amount distribution. The initial capital is r_0. By all this assumptions the risk process is described as follows.

$$Y_t = r_0 + ct - \sum_{i=0}^{N_t} X_i \tag{1}$$

From the previous model, the probability of ruin is the probability of the reserve falls below zero. i.e

$$\psi(x) = P(Y_t < 0) \tag{2}$$

and its complement is given by

$$F_S(x) = 1 - \psi(x)$$

that is the survival probability of the maximal aggregated loss.

We are interested in the estimation of two risk measures based on this quantity. The value at ruin that represents the amount of required capital in order to have a fixed level of probability of ruin and the tail value at ruin, that is the the conditional expectation of the maximal aggregated loss given that it is bigger than the corresponding VaRu at some level ϵ tipically very small. Both measures are presented in (Baumgartner & Gatto, 2014).

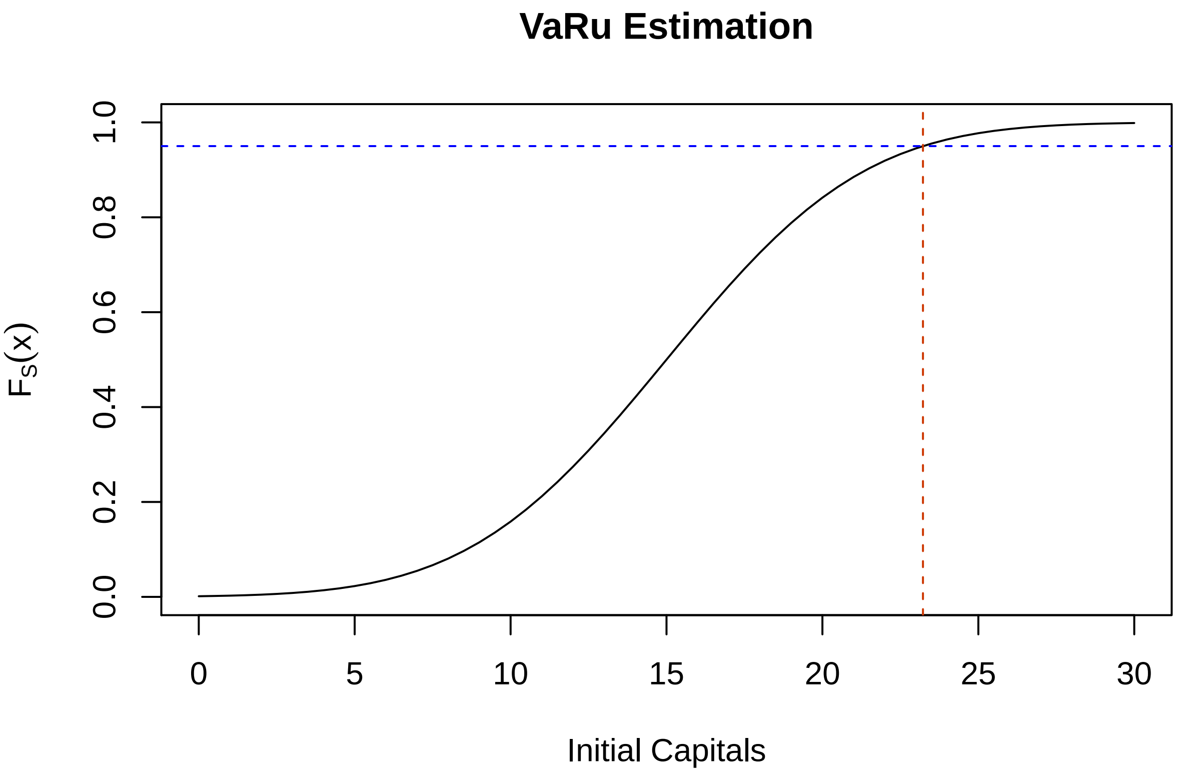
The estimation of the first quantity, $VaRu$ is crucial to obtain the second one. This measure is defined in such a way

$$\begin{aligned} VaRu(\epsilon) &= \{x \geq 0 | \psi(x) \leq \epsilon\} && (3)\\ &= \{x \geq 0 | F_S(x) \geq 1-\epsilon\} && (4)\\ &= q_{1-\epsilon} && (5) \end{aligned}$$

From the last expression, we conclude that in order to determine the value at ruin at some level $\epsilon \in (0,1)$, we only need to find the quantile at level $1-\epsilon$ of the distribution of the maximal aggregated loss S. By using the Monte Carlo method, we estimate the probability of ruin $\psi(x)$. Then, since it is the complement of F_S, we have found an estimator of it, let us call it $\hat{F}_S$.

Graphically, having the estimated distribution of the maximal aggregated loss S, we need to find the intersection of

$$\hat{F}_S(x) = 1 - \epsilon$$



Having estimated the value at ruin, we can fit a regression model in the tail and compute the corresponding estimator, the tail value at ruin can be found using an explicit formula that we will show later.

However, since we are dealing with rare event simulation, is important to consider variance reduction techniques. Supposing that the probability of ruin of interest is 0.00001 it means that we need at least to generate 100000 simulations in order to get one ruin. It explains why the Monte Carlo method is not efficient at all under this framework.

Considering light tailed claim amounts, the importance sampling approach by the exponential change of measure is used. A theoretical development of this method is found in (Søren Asmussen & Glynn, 2007). Also, we have used several results that can be seen in detail in (Gatto, 2014b). In the last reference, is explained the algorithm in the case of the Risk process for claim amounts with distribution F. A brief summary of this method also can be found in (Gatto, 2014a), where is presented the method of importance samping and some applications.

Next, for the heavy tailed case, we present three algorithms based on conditional simulation and control variaables, important references to this approaches are the works of (S Asmussen & Binswanger, 1997) who consider conditional simulation methods, (Juneja, 2007), who estimate efficiently tail probabilities of heavy tailed distributions and (Søren Asmussen & Kroese, 2006). We have used the Pollakzeck-Khinchine formula and some results related to subexponential distributions when the claim amunts are heavy tailed. Also, we have used the estimation of random variables following an integrated tail Weibull distribution using the method of acceptance rejection.

The idea of the conditional Monte Carlo methods presented in this work is only generate $k-1$ claim amounts and consider the probability that the next claim causes ruin. For the second conditional algorithm, we have included the use of order statistics considering that for subexponential distributions one large claim amount causes ruin. This idea has been discussed in (S Asmussen & Binswanger, 1997).

The structure of this master thesis is the following, in chapter 1 we start with a survey of risk theory, defining important key concepts to understand the Cramer Lundberg model and the measures of risk of interest. Next, in chapter 2, we present some results regarding to variance reduction methods. In chapter 3 we present the Monte Carlo estimators as well as the algorithms of variance reduction. A numerical study of the algorithms and a brief explanation of the programming schemes used is presented in chapter 4. Finally, we show the conclusions in chapter 5.

Acknowledgement

This master thesis would not have been possible without the invaluable guidance of my advisor the Prof. Dr Riccardo Gatto of the Institute of Mathematical Statistics and Actuarial Sciences at the University of Bern . The door to Prof. Gatto office was always open whenever I ran into a trouble spot or had a question about my thesis or the lectures attended with him.

I would also like to thank all the Professors and teacher assitants of the IMSV for share their knowledge with me during the master studies, studying in Switzerland has been a really awesome experience in my life. I want to thank also to the Prof. Illya Molchanov, he was always helping me since the beginning of the studies. Also I am very grateful with the Ecuadorian government for the economical support for my studies with the scolarship granted to me by the SENESCYT, now I should come back to my country to share my knowledge and my experience with other people involved in the fascinated world of the mathematics and statistics.

Thank you very much to all the new friends and colleagues that I met in the faculty and in Switzerland: Amanda, Haoyang, Tibor, Pete, Jin, Angeliki, Mirlind, Eddie, Amr, Fabrice, Luana, Alex, Hernando, Tito and Oso. My sincere thanks to my friends Diego, Sebastian, Gabriela, Esteban, Willy who have been talking with me in a daily basis about everything. I am very grateful with Janny, thank you for your support and share your time with me.

Last but not least, I would like to thank to my family here and my family in Ecuador specially to my parents who have been an important support in my life.

Boris.

Chapter 1

The Risk Process

1.1 Introduction

One of the key concepts in the actuarial field is the notion of risk, since the importance for insurance companies, banks or other financial institutions is relevant to have technical methods of managing risk analysis tools Therefore, our interest is to deal with a measurement of the insurer risk. For that reason, we present the classical model studied in actuarial risk theory, called the risk process introduced in 1903 by the Swedish actuary Filip Lundberg.

1.2 Risk Process

In order to define the classical model, we require the following quantities: Let $X_1, X_2, ...$ be independent individual random losses or claim amounts with distribution function F and expectation $\mu < \infty$.
k_t the number of claims occuring during $[0, t]$, $\forall t \geq 0$.
The total loss or claim amount is given by:

$$Z_t = \sum_{k=0}^{K_t} X_k$$

, where $X_0 = 0$, r_0 is the initial capital $c > 0$ is the premium rate (assumed constant).

$$Y_t = r_0 + ct - Z_t \tag{1.1}$$

We present now a simulated path of the classical risk process with initial capital $r_0 = 4$ and premium rate $c = 3$

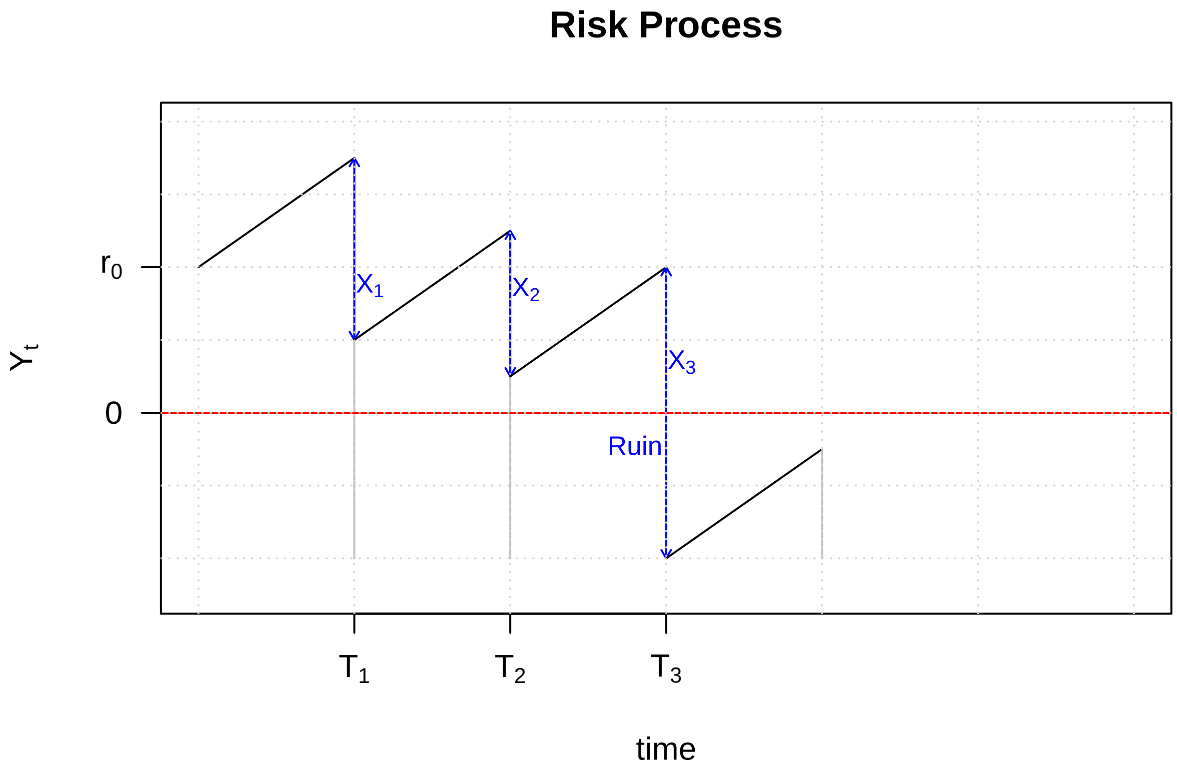


Figure 1.1: Sample path of the Cramer Lundberg Process

Notation

- T_k is the $k-th$ claim time, i.e

$$T_k = inf\{t \geq 0 : K_t \geq k\}, k = 0, 1, ..$$

- $D_k = T_k - T_{k-1}$ are the interclaim times.
- If $D_1, D_2, ...$ are i.i.d, then $\{T_k\}_{k\geq 0}$ or $\{K_t\}_{t\geq 0}$ are called renewal processes.
- We focus on renewal counting process. In this case

$$\rho = \frac{E[X_1]}{E[D_1]}$$

  is the *ruin parameter*.
- Furthermore, we define the *security loading* by

$$\beta = \frac{c-\rho}{\rho}$$

The main object of the model is to analyze the event when the risk process falls below zero, this event is called *Ruin* and its probability is given by:

Definition 1.1 (Probability of Ruin in the finite time horizon)**.**

$$\Psi(r_0, t^\dagger) = P\{\inf_{t\in[0,t^\dagger]} Y_t < 0\} \tag{1.2}$$

where $t^\dagger$ is any time horizon.

Aditionally, we define the probability of ruin in the infinite time horizon or simply called *probability of ruin*

Definition 1.2 (Probability of Ruin)**.**

$$\Psi(r_0) = \lim_{t^\dagger\to\infty} \Psi(r_0, t^\dagger) = P\{\inf_{t\geq 0} Y_t < 0\} \tag{1.3}$$

We also define the time of ruin given by:

Definition 1.3 (Time of Ruin)**.**

$$T = \begin{cases} inf\{t > 0 : Y_t \leq 0\} & \text{If the infimum exists,} \\ 0, & \text{otherwise} \end{cases} \tag{1.4}$$

Definition 1.4 (Loss Process)**.** Consider the risk process given above, the Loss Process is defined as

$$L_t = Z_t - ct, \quad \forall t \geq 0 \ \ (\text{w.l.o.g } c = 1). \tag{1.5}$$

That is the loss minus the new capital of the insurance.

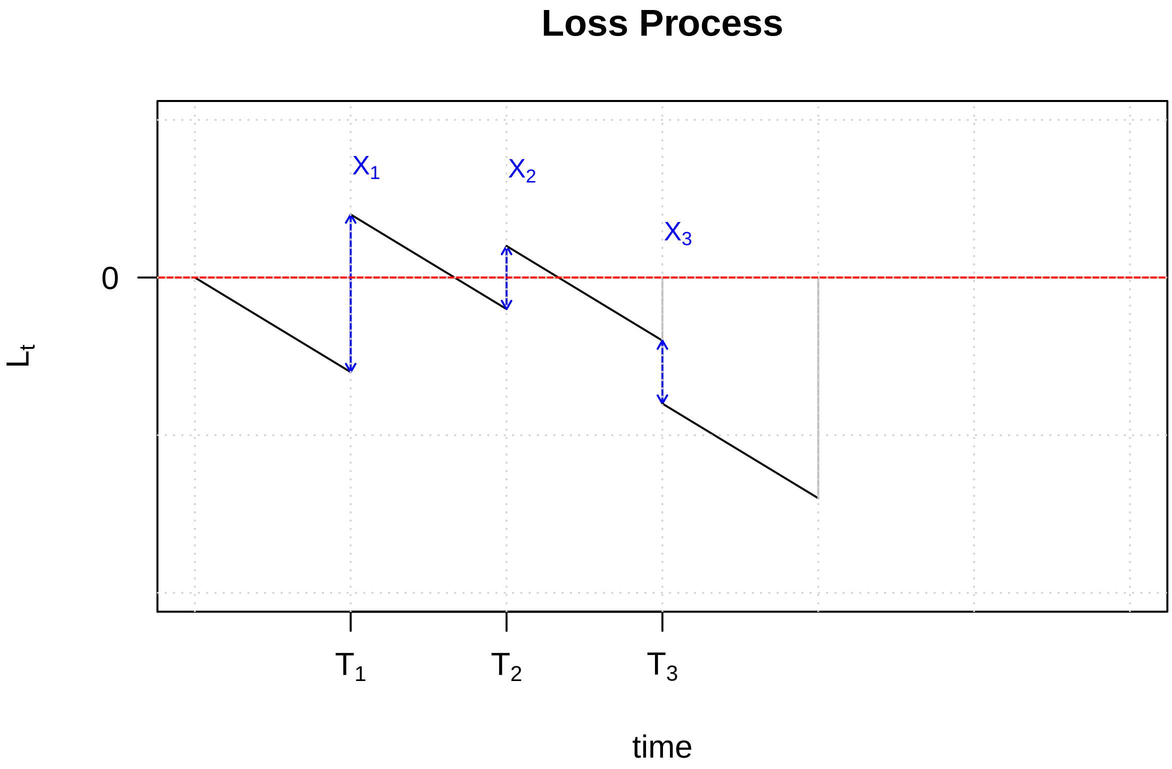


Figure 1.2: Sample path of the Loss Process

A random variable of interest is the maximal agreggate loss given by:

$$S = \sup_{t\geq 0} L_t$$

As a result of the previous definitions, we have the next relation between the risk process and the maximal agreggate loss.

$$\begin{aligned}\psi(r_0) &= P(\inf_{t<0} Y_t < 0)\\ &= P(\inf_{t\geq 0}\{r_0 + c.t - Z_t\} < 0)\\ &= P(\{r_0 + \inf_{t\geq 0}\{c.t - Z_t\}\} < 0)\\ &= P(\{r_0 + \inf_{t\geq 0}\{-L_t\}\} < 0)\\ &= P(r_0 - \sup_{t\geq 0} L_t < 0)\\ &= P(\sup_{t\geq 0} L_t > r_0)\\ &= 1 - P(S \leq r_0)\\ &= 1 - F_S(r_0)\end{aligned} \tag{1.6}$$

As we have seen, an important part of the classical risk process is the total claim amount, that is given by a Compound Poisson Process. In order to define it, we present some required definitions.

Definition 1.5 (Counting Process)**.** A counting process $\{N_t\}_{t\geq 0}$ is a collection of non-negative, integer-valued random variables, such that, if $0 \leq s \leq t$, then $N_s \leq N_t$

Now, we present three ways to characterize the Poisson Process, It can be done if we focus on *i*) the number of events that occur in fixed intervals, *ii*) when events occur, and the times between those events, or *iii*) the probabilistic behaviour of individual events on infinitesimal intervals.

Definition 1.6 (Poisson Process)**.** A Poisson process with parameter λ is a counting process $(N(t))_{t\geq 0}$ if the following conditions hold:

- The process starts at zero: $N(0) = 0$ a.s.
- For all $t > 0$, N_t has a Poisson distribution [1] with parameter λt.
- (Stationary increments) For all $s, t > 0, N_{t+s} - N_s$ has the same distribution as N_t. That is:

$$P(N_{t+s} - N_s = k) = P(N_t = k) = \frac{e^{\lambda t}(\lambda t)^k}{k!}, \text{for } k = 0, 1, ...$$

[1] An integer valued random variable M is said to have a Poisson distribution with parameter $\lambda > 0$ ($M \sim Pois(\lambda)$) if it has distribution

$$P(M = k) = e^{-\lambda}\frac{\lambda^k}{k!}, \quad k = 0, 1, ...$$

- (Independent increments)For $0 \leq q < r \leq s < t$, $N_t - N_s$ and $N_r - N_q$ are independent random variables.

Example 1.1 (Poisson Process)**.** A trajectory of a Poisson Process with exponential interarrival times with parameter $\lambda = 0.6$ until time 15 is given by:

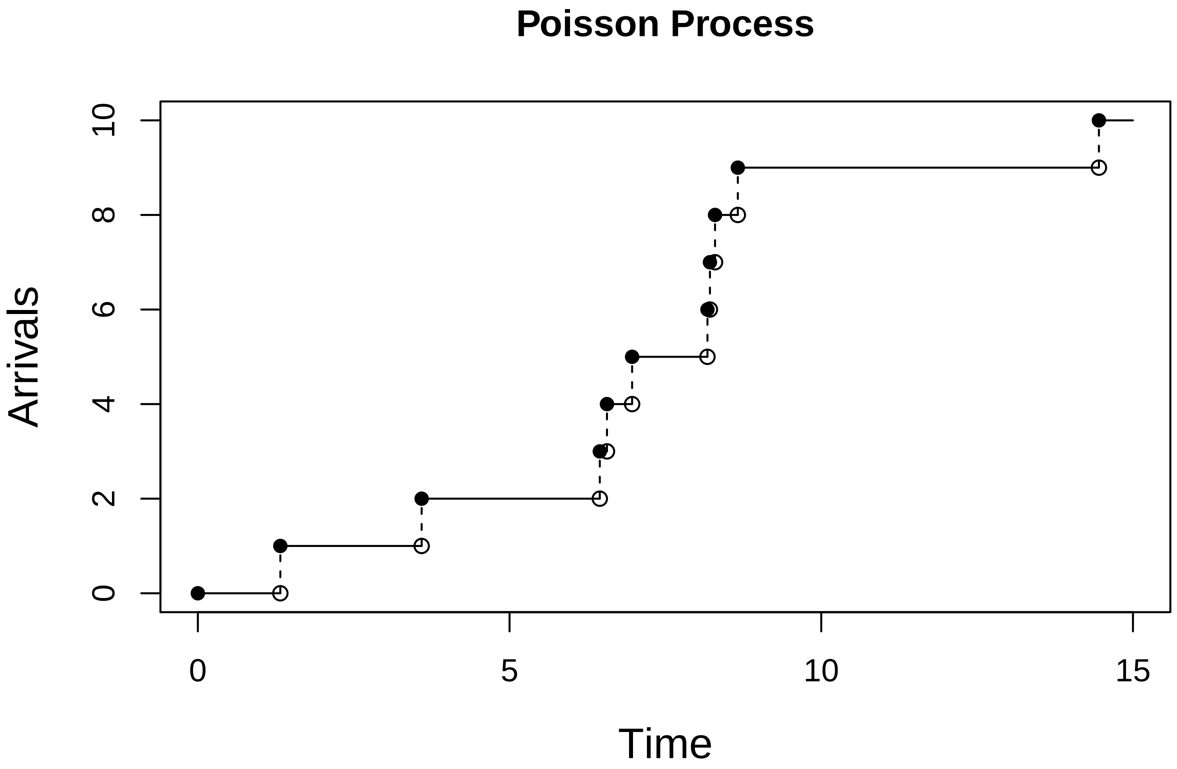


Figure 1.3: Trajectory of a Poisson Process

```
$arrival_times
 [1]  0.000000  1.323158  3.590259  6.447827  6.562793  6.966239  8.174458
 [8]  8.214520  8.296459  8.661917 14.454117

$cum_time
[1] 18.98112
```

Definition 1.7 (Compound Poisson Process)**.** Let $\{N_t =\}_{t\geq 0}$ be a Poisson Process with parameter λ and let $X_1, X_2, ...$ be a sequence of independent and identically distributed random variables, each with distribution function F, independent of N_t $\forall t > 0$. We define the Compound Poisson Process $\{S_t\}_{t\geq 0}$ with parameter λ by:

$$S_t = \sum_{i=1}^{N_t} X_i$$

with $S_t = 0$ when $N_t = 0$

Example 1.2 (Compound Poisson Process)**.** The next plot is a trajectory of a compound Poisson process, with interarrival time exponentially distributed with parameter $\lambda_1 = 2$ generated until time 15. The jumps are exponentially distributed with parameter $\lambda_2 = 0.5$.

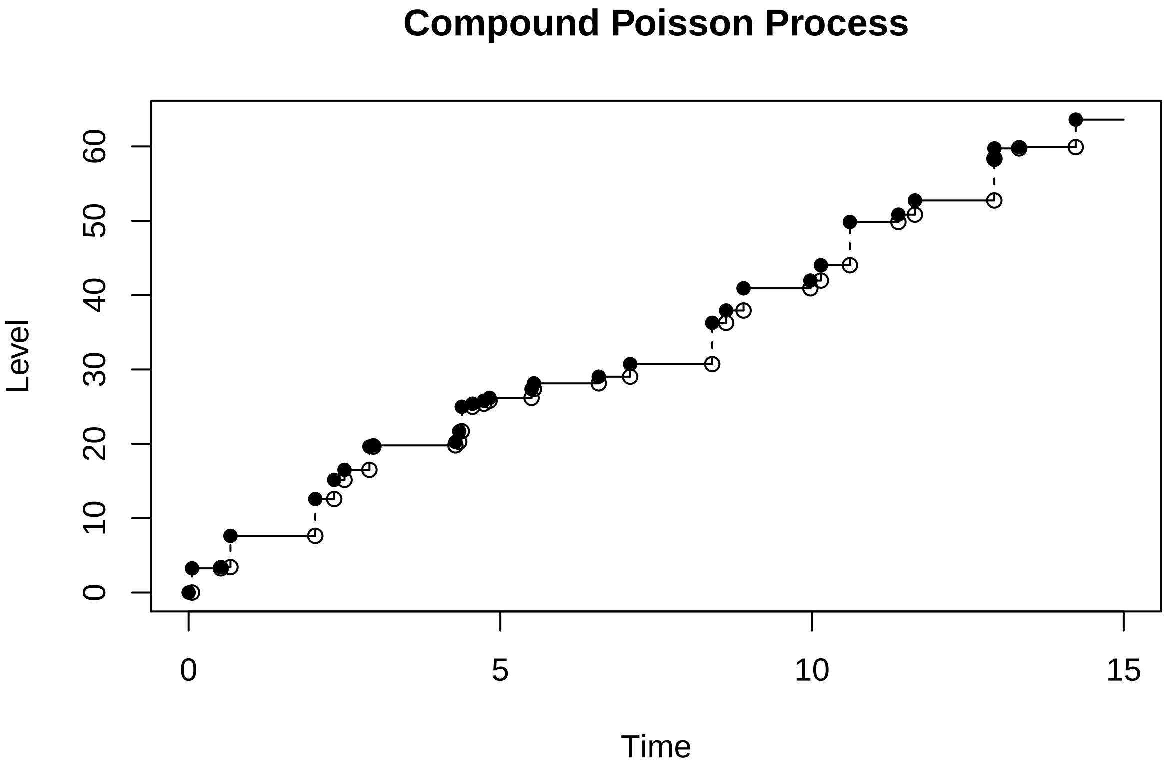


Figure 1.4: Trajectory of a Compound Poisson Process with exponential distributed claims

```
$arr_times
 [1]  0.00000000  0.05495328  0.51637879  0.67128581  2.03192660
 [6]  2.33587029  2.50071322  2.90023104  2.96352788  4.27985007
[11]  4.34058618  4.38185238  4.55373168  4.73687785  4.82749609
[16]  5.50122235  5.53815612  6.57913733  7.08290242  8.39951503
[21]  8.62248451  8.90194356  9.97407181 10.14225626 10.60762281
[26] 11.38558021 11.65090014 12.92376959 12.92600208 13.32256559
[31] 14.22993637

$level
 [1]  0.000000  3.254944  3.416789  7.621798 12.576941 15.153164 16.500528
 [8] 19.631373 19.789200 20.256232 21.689466 24.982743 25.391082 25.797858
[15] 26.179389 27.361660 28.132648 29.028627 30.717125 36.273224 37.930402
[22] 40.919202 41.958911 44.018289 49.840428 50.826397 52.733440 58.339177
[29] 59.734641 59.906415 63.603010

$cum_time
[1] 15.06618

$counting
[1] 31
```

1.3 Loss Distributions

We present a brief survey of some classes of distributions that are used to model claim amounts $X_1, X_2, X_3, \ldots$. These distribution can be distincted between the tails that tend to 0 faster or slower than the exponential distribution. We note the tail of a distribution F as $\bar{F}$

Definition 1.8 (Light Tailed Distribution)**.** Let X a random loss with distribution F, and $\lambda > 0$. Then X is called light tailed or lighter than exponential if

$$\lim_{x\to\infty} \frac{\bar{F}(x)}{e^{-\lambda x}} = 0 \tag{1.7}$$

Definition 1.9 (Heavy Tailed Distribution)**.** Let X a random loss with distribution F, and $\lambda > 0$. Then X is called heavy tailed or heavier than exponential if

$$\lim_{x\to\infty} \frac{\bar{F}(x)}{e^{-\lambda x}} = \infty \tag{1.8}$$

Traditional examples of loss distributions can be found in the actuarial literature. We present some loss distributions in the light and the heavy tailed case. More distributions can be found on: (Gatto, 2014b), (Søren Asmussen & Albrecher, 2010) and (H. Albrecher, Teugels, & Beirlant, 2017)

Example 1.3 (Exponential Distribution)**.** The random loss X follows an exponential distribution if X has a density $f_X(x) = \lambda e^{-\lambda x}$ and distribution function $F_X(x) = 1 - e^{-\lambda x}$ $\forall x > 0$ and $\forall \lambda > 0$. The parameter λ is called the rate of the exponential distribution

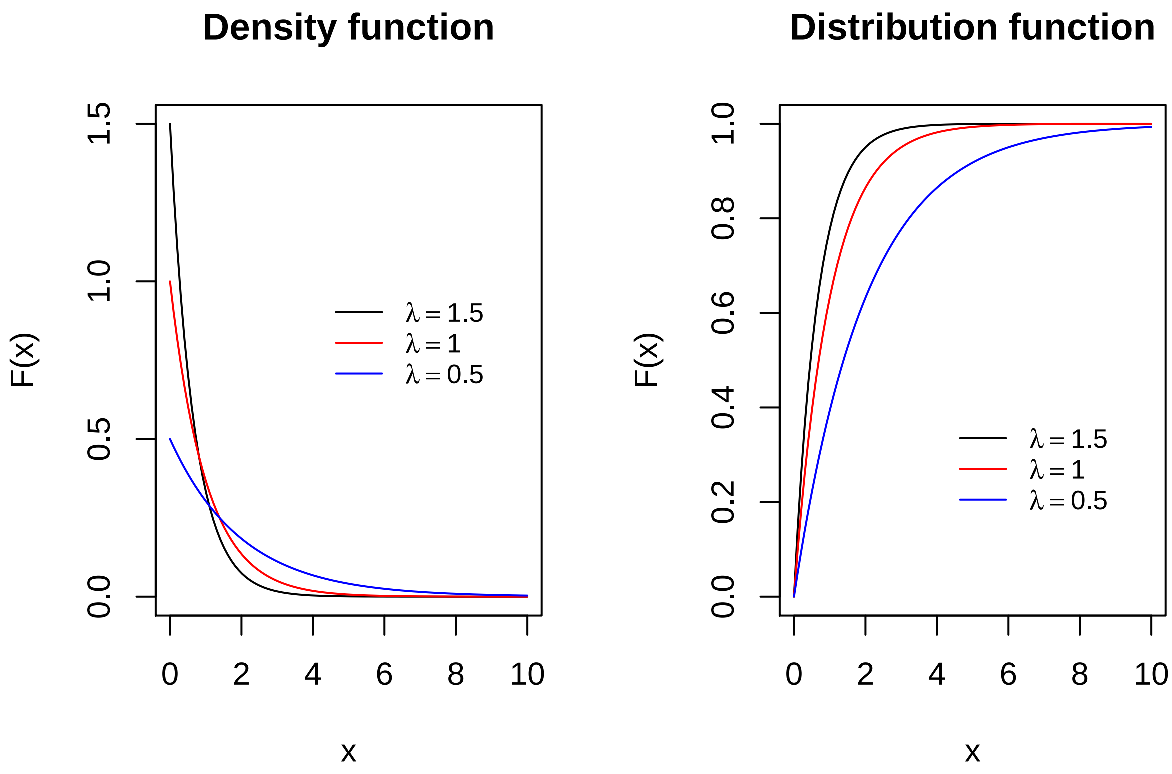


Figure 1.5: Exponential distribution

Example 1.4 (Weibull Distribution)**.** Let X a random loss that follows an exponential distribution with parameter λ, then $Y = X^{\frac{1}{\tau}} \forall \tau > 0$ follows a Weibull distribution. Its distribution is given by $F_Y(y) = F_X(y^\tau) = 1 - e^{-\lambda x^\tau}\ \forall y > 0$. The parameter τ is called the Weibull index i.e $Y \sim Weibull(\lambda, \tau)$.

The Weibull distribution can be classified according to the Weibull index in light tailed ($\tau > 1$), heavy tailed ($0 < \tau < 1$) or it simply is reduced to the exponential distribution if $\tau = 1$.

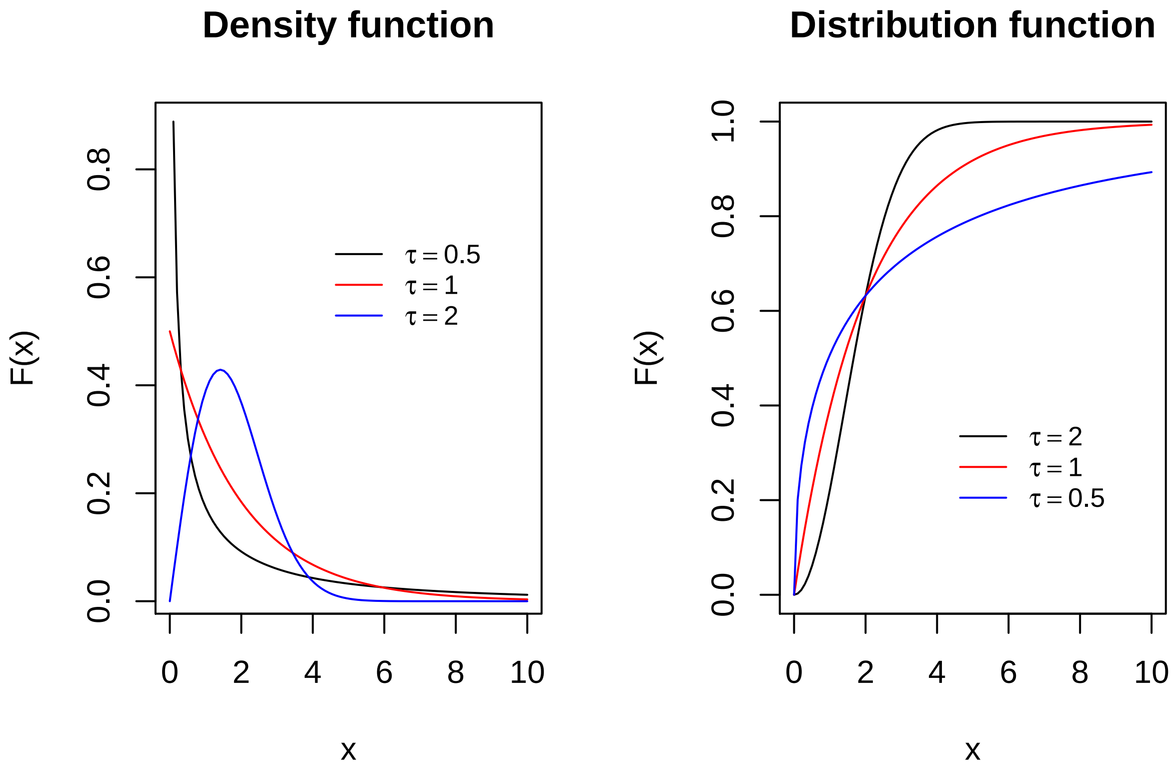


Figure 1.6: Weibull distribution

Moreover, for values of $\tau \in (0,1)$ the Weibull distribution belongs to the class of subexponential distributions. We introduce also an important definition that we are going to use in the development this thesis. We have followed the definition given in (Lin, 2006)

Definition 1.10 (Integrated tail distribution)**.** Let X a nonnegative random variable, with distribution function F, also denote the kth moment of X by $p_k = E[X^k]$, if it exists. The integrated tail distribution F_0 of X or $F_X(x)$ is a continuos distribution on $[0,\infty]$ such that its density is given by

$$f_1(x) = \frac{\bar{F}(x)}{p_1} \quad x > 0$$

where $\bar{F}(x) = 1 - F(x)$. This distribution is often called the equilibrium distribution.

Further details and properties of the integrated tail distribution can be found in the mentioned reference above.

For the Weibull distribution, we can express the integrated tail distribution in terms of the incomplete Gamma function performing the change of variable $y = t^\tau$ as follows.

$$\begin{aligned} F_0(x) &= \frac{1}{\mu_{F_X(x)}} \int_0^x (1 - F_X(t))dt \\ &= \frac{1}{\mu_{F_X(x)}} \int_0^x exp(-t^\tau)dt \\ &= \frac{1}{\mu_{F_X(x)}} \frac{1}{\tau} \int_0^{x^\tau} exp(-y)y^{\frac{1}{\tau}-1}dy \\ &= \frac{1}{\mu_{F_X(x)}} \frac{1}{\tau} \gamma(\frac{1}{\tau}, x^\tau) \end{aligned}$$

Similarly we obtain

$$\bar{F_0}(x) = \frac{1}{\tau \mu_{F_X}} \Gamma(\frac{1}{\tau}, x^\tau)$$

where $\Gamma(a, u) = \int_u^\infty t^{a-1} exp(-t)dt$ is the complement of the incomplete Gamma function. This representation is an important result for us since it allow us to evaluate the integrated tail distribution F_0 for a value of x since in most of the statistical packages the incomplete gamma function is implemented.

Something important that we are going to use in the next chapters is the Gamma distribution. We mention some facts important about this distribution:

Definition 1.11 (Gamma Distribution)**.** Let X a random loss that follows a Gamma distribution with shape parameter *alpha* and scale parameter θ, there is no a closed form for the Distribution function, the expression is given by:

$$F_X(x) = \int_0^x \frac{1}{\Gamma(\alpha)\theta^\alpha} t^{\alpha-1} e^{-t/\theta} dt \quad x > 0 \tag{1.9}$$

Using a change of variable $u = \frac{t}{\theta}$ we have $du = \frac{dt}{\theta}$ and $t = u\theta$ and the distribution

$$\begin{aligned} F_X(x) &= \int_0^{\frac{x}{\theta}} \frac{1}{\Gamma(\alpha)\theta^\alpha} (u\theta)^{\alpha-1} e^u \theta du \\ &= \frac{1}{\Gamma(\alpha)} \int_0^{\frac{x}{\theta}} u^{\alpha-1} e^{-u} du \\ &= \frac{1}{\Gamma(\alpha)} \gamma(\alpha, \frac{x}{\theta}) \end{aligned}$$

Where $\gamma(\beta, w) \int_0^w t^{\beta-1} e^{-t} dt$ and $\Gamma(\beta) = \int_0^\infty t^{\beta-1} e^{-t} dt$

When the Gamma distribution is raised to a power, the resulting CDF can be defined as a function of the distribution Gamma F_X.

1.4 Measures of Risk

A Risk Measure is a statistical tool used to determine “the chance of an adverse outcome” (Klugman, Panjer, & Willmot, 2012). In Risk Management is helpful in the determination of the amount of capital required to confront adverse outcomes. Also, it can be applicable by the governments through regulatory entities in order to avoid insolvency of insurance companies or banks. For example, Solvency II [2], which has been implemented in the European Union since January 2016, the required capital is the Value at Risk, in contrast the Swiss Solvency Test, that is a risk based capital standard for insurance companies in Switzerland uses the Tail Value at Risk.

1.5 Risk measures and coherence

A *Risk Measure* is a function $\rho : L_p(\Omega) \to \mathbb{R}_+$, representing the amount of capital required for protecting against a *random loss* Z.

Definition 1.12 (Coherent Measure of Risk)**.** Let $X, Y \in Ł_p(\Omega)$ be two random losses[3] $\rho : L_p(\Omega) \to \mathbb{R}_+$ is a coherent measure of risk if ρ fulfills.

- $\rho(X+Y) \leq \rho(X) + \rho(Y)$ (subadditivity).
- If $X \leq Y$ a.s $\Rightarrow \rho(X) \leq \rho(Y)$ (monotony).
- $\rho(cX) = c\rho(X)$, $\forall c > 0$ (scale invariant).
- $\rho(c+X) = c + \rho(X)$, $\forall c > 0$ (traslation invariant).

The idea of a risk measure is that given a loss X, $\rho(X)$ quantifies the riskiness of X, it means that large values of X tell us that X is dangerous.

1.5.1 Interpretation

1. The aggregation of risk is beneficial and in this case the insurance company will profit, i.e. the risk can be reduced by diversification.
2. The order between the corresponding two losses must be equal to the risks.
3. If a loss is multiplied by a constant, then the risk multiplies in accordance. Another interpretation is that a change in currency is ineffective.
4. With a resulting loss, results also a corresponding risk.

1.6 Value at Risk (VaR)

Value-at-Risk (VaR) has become the standard risk measure used to evaluate exposure to risk. In general terms, the VaR is the amount of capital required to ensure, with high degree of certainty, that the enterprise does not become technically insolvent.

[2] Solvency II is a set of rules in the European Union that regulates the insurance industries in EU. That rules are concerned about the amount of capital that EU insurance companies must hold to reduce the risk of insolvency.

[3] A randon loss is any non-negative random variable, its distribution function is called loss.

Definition 1.13 (Value at Risk)**.** Let X denote a random loss, the Value-at-risk of X is the α_{th} quantile of the distribution of the loss X, $\forall \alpha \in (0,1)$. It is denoted by $VaR_\alpha(X)$,i.e.

$$q_\alpha = inf\{x \in \mathbb{R} : F_X(x) \geq \alpha\}$$

1.7 Tail Value at Risk (TVaR)

In practice, the Value at Risk (VaR), as a risk measure is used in risk management if the distributions of gains or losses is normal. However, the normal distribution is not used for describing insurance losses. Consequently, the use of VaR is problematic since it is not subadditive. A more useful measure of risk is Tail-Value-at-Risk ($TVaR$). It has been given some different names including, Conditional Value at Risk ($CVaR$), Conditional Tail Expectation (CTE) and Expected Shortfall (ES).

Definition 1.14 (Tail Value at Risk)**.** Let X denote a random loss variable, the Tail-Value-at-risk is defined by:

$$TVaR(X) = E(X|X > q_\alpha)$$

where q_α is the α_{th} quantile of $VaR(X)$, $\forall \alpha \in (0,1)$

Now we present two measures of risk based in the previous measures. Both measures of risk and its computation using saddlepoint approximations appear in (Baumgartner & Gatto, 2014). First we consider the capital required to obtain a fixed probability of ruin, tipically very small as a measure of insurer risk. This capital is called Value at Ruin VaRu. Similarly to TVaR we consider another measure that fulfills the coherency property, this measure is called Tail Value at Ruin TVaRu.

1.8 Value at Ruin $VaRu$ and the Tail Value at Ruin $TVaRu$

Definition 1.15 (Value at Ruin)**.** Consider the probability of ruin of a risk process $\psi(x) = P(T < \infty)$, for any initial capital $x \geq 0$. The associated value at ruin $VaRu$ at level $\epsilon \in (0,1)$ is given by:

$$VaRu(\epsilon) = \inf\{x \geq 0 : \psi(x) \leq \epsilon\} = q_{1-\epsilon} \tag{1.10}$$

Although, this measure of risk is not subadditive, then is not coherent. In analogy to the $TVaR$, we can obtain a coherent measure of risk from the $VaRu$ by considering the expectation of the maximal aggregate loss given that the maximal aggregate loss exceeds a fixed value at ruin $VaRu$.

Definition 1.16 (Tail Value at Ruin)**.** Consider the risk process given in 1.1 and the corresponding maximal agreggated loss S, then the associated Tail Value at Ruin $TVaRu$ at level $\epsilon \in (0,1)$

$$TVaRu(\epsilon) = E[S|S > VaRu(\epsilon)] \tag{1.11}$$

Chapter 2

Variance Reduction Methods

Variance reduction methods are useful to produce estimators with smaller variance than the crude Monte Carlo method. In the context of this thesis, we are interested in approximate a quantile. The Monte Carlo estimator for this quantity can be obtained by simulating several replications of the random variable of interest, (The maximal aggregated loss in our case S), with cdf F_S and computing the quantile

$$\hat{q}_{1-\epsilon} = inf\{x : \hat{F}_S(x) \geq 1 - \epsilon\}$$

If we suppose that the random variable S follows a normal distribution, the simulation of the respective quantile at level 0.995[1] will need more than 300 million of replications if we want a confidence interval of length 0.01. The main idea of the variance reduction techniques is to replace the original estimator by another with the same expectation, but smaller variance, getting gain in our computations and a better performance of the crude Monte Carlo method.

2.1 Importance Sampling

Importance sampling provide us an optimal simulation scheme for computing $z = E[Z]$ in the sense of minimizing the number of generations in order to reach a given level of precision. The main idea of this method is to find a distribution for the underlying random variable that asssigns a high probability to those values that are in the relevant region for the desired approximated quantity. For instance, we are interested in the probability of ruin in the tails that usually is very low.

2.2 Some definitions on measure theory.

Definition 2.1 (Absolute Continuity)**.** Let ν and λ be two measures on $\Omega, \mathcal{F}$. If $\nu(A) = 0$ implies $\lambda(A) = 0$ $\forall A \in \mathcal{F}$, then λ is absolutely ccontinuous with respect to ν, this is denoted by $\lambda << \nu$

[1] Solvency II and the Swiss Solvency Test suggest the quantile at level 0.995

Remark (Alternative characterization of absolute continuity)*.* If there exists $f : \Omega \rightarrow \mathbb{R}_+$ measurable such that $\lambda(A) = \int_A f(\omega)d\nu(\omega)\ \forall A \in \mathbb{F}$ then $\lambda << \nu$.

The converse of this fact is given by the theorem of Radon-Nikodyn.

Theorem 2.1 (Radon-Nikodyn)**.** *Consider $\lambda << \nu$, then there exists a measurable $f : \Omega \rightarrow \mathbb{R}_+$ s.t $\int_A f(\omega)d\nu(\omega), \forall A \in \mathcal{F}$. Further f is $v - a.e.$ unique (unique up to all sets A where $\nu(A) = 0$).*

We denote f in the above theorem as $\frac{d\lambda}{d\nu}$ and call it as the Radon-Nikodyn derivative or simply the density w.r.t ν.

Example 2.1 (Typical discrete case)**.** Let $\Omega = \{\omega_1, \omega_2, ...\}$ and let $\nu(A) = card(A)$ for $A \subset \Omega$. Let $f(w) = \sum\limits_{i=1}^{\infty} p_i I\{\omega = \omega_i\}$ where $p_1, p_2, ... > 0\backslash$ Now,$\lambda(A) = \int_A f(\omega)d\nu(\omega) = \sum_{i=1}^{\infty} p_i I\{\omega_i \in A\} = \sum\limits_{i \geq 1 s.t \omega_i \in A} p_i = \sum\limits_{i \geq 1 s.t \omega_i \in A} f(\omega_i)\backslash$ If $\sum\limits_{i=1}^{\infty} p_i = 1$, then $P(A) = \int_A f(\omega)d\nu(\omega)$, $A \subset \Omega$ defines a probability measure on Ω with density $f = \frac{dP}{d\nu}$, $(P << \nu)$.

Now we present the theorem of change of measure.

Theorem 2.2 (Change of Measure)**.** *Consider two measures λ and ν and assume that the density of λ w.r.t ν exists ($\frac{d\lambda}{d\nu}$). Then, for any measurable function f,*

$$\int_{\Omega} f(\omega)d\lambda(\omega) = \int_{\Omega} f(\omega)\frac{d\lambda}{d\nu}d\nu(\omega)$$

Some practical results are the following:

- If $\lambda << \tau$ and $\nu << \tau$, then for $\sigma = \lambda + \nu$, we have $\frac{d\sigma}{d\tau} = \frac{d\lambda}{d\tau} + \frac{d\nu}{d\tau}$
- If $\lambda << \nu << \tau$, then $\frac{d\lambda}{d\tau} = \frac{d\lambda}{d\nu}\frac{d\nu}{d\tau}$, τ. a.e.(chain rule).
- If $\lambda << \nu$ and $\nu << \lambda$, then $\frac{d\lambda}{d\nu} = (\frac{d\nu}{d\lambda})^{-1}$, In this case λ and ν are said to be equivalent measures.

Example 2.2. Let $\Omega = \{\omega_1, \omega_2, ...\}$, ν the counting measure and $p_1, p_2, ... > 0$ such that $\sum\limits_{i=1}^{\infty} p_i = 1$. Let $P[A] = \sum\limits_{i \geq 1, s.t. \omega_i \in A} p_i$, which is a probability measure equivalent to ν and let $g : \Omega \rightarrow \mathbb{R}_+\backslash$ We want to compute $z = \int_{\omega} g d\nu = \sum\limits_{i=1}^{\infty} g(\omega_i)$, thus we write $z = \int_{\omega} g\frac{d\nu}{dP}dP = \int_{\omega} g(\frac{dp}{d\nu})^{-1}dP = \sum_{i=1}^{\infty} \frac{g(\omega_i)}{f(\omega_i)}p_i = E_P[\frac{g(X)}{f(X)}]$, where $f = \frac{dP}{d\nu}$ $(f(\omega_i) = p_i)$ and $X(\omega) = \omega, \forall \omega \in \Omega\backslash$

With the results obtained in the previous example we can establish the next montecarlo simulation scheme. Assume $g = I_A$ and $card(A) < \infty$ but difficult to enumerate, thus we desire

$$z = \int_{\Omega} g d\nu = card(A)$$

- Generate X_i from f, viz. from P.

- Compute $\hat{z} = \frac{1}{n} \sum \_i = 1^n \frac{g(X_j)}{f(X_j)}$ with $z = \lim_{n \to \infty} \hat{z}$

We need this for the importance sampling theorem later. Let $(\Omega, \mathcal{F})$ be a measure space with probability measures P and $\tilde{P}$ and another measure ν. Assume $\tilde{P} << P$, $P << \nu$ and $\nu << P$. Denote $f = \frac{dP}{d\nu}$ and $\tilde{f} = \frac{d\tilde{P}}{d\nu}$, given that ν and P are equivalent, $\frac{d\nu}{dP} = \frac{1}{f}$(from property 3), $P - a.s$ or $\nu - a.e.$ (from property 2). Thus $\frac{d\tilde{P}}{dP} = \frac{\tilde{f}}{f}$.

(Setting of the next theorem)

Let Z be a random variable over $(\Omega, \mathcal{F})$ with induced distributions P_Z under P and $\tilde{P_Z}$ under $\tilde{P}$. Let f_Z be the density of $\tilde{P_Z}$ w.r.t some measure μ, such that $P_Z << \mu$, $\tilde{P_Z} << \mu$ and $\mu << \tilde{P_Z}$.

Under these circumstances we give the next theorem. It is used to deduce the importance sampling theorem.

Theorem 2.3 (Importance Sampling). $\frac{\tilde{f_Z(z)}}{f_Z(z)}$ *is equivalent to* $\frac{\tilde{f}}{f}$*, on the restriction of* $\mathcal{F}$ *to* $\sigma(Z)$*. This means that:*

$$\tilde{P}(A) = \int_A \frac{\tilde{f}}{f} dP = \int_A \frac{\tilde{f_Z(z)}}{f_Z(z)} dP$$

provided that $A \in \sigma(Z)$*, i.e that* $A = Z^{-1}(B)$*, for some* $B \in \mathcal{B}(\mathbb{R})$

$z = E[Z]$ is the desired quantity. We assume that $P << \tilde{P}$ on $\{Z \neq 0\} \cap \mathcal{F}$ and let $L := \frac{f_Z(z)}{\tilde{f_Z}(z)}$ be equivalent to $\frac{\tilde{f}}{f}$ on the restriction to $\{Z \neq 0\} \cap \sigma(Z)$. Then $z = E_{\tilde{P}}[Z.L]$ where $\tilde{P}$ is the importance sampling estimator of Z.

```
Data: $Z_1, ..Z_n$
Result: Importance sampling estimator of $Z$
initialization;
Let $Z_1L_1, ..., Z_nL_n$ be independent generations of $ZL$ under $\tilde{P}$. Take the
  empirical estimator of $z := \hat{z} = \frac{1}{n}\sum_{j=1}^{n} Z_j L_j$
```

Algorithm 1: Montecarlo Importance Sampling Estimator

Example 2.3. Let $X_1, X_2, ..., X_k$ be independent with common density f_X under P and let $Z = g(X_1, .., X_k)$, we are interested in $z = E_P[Z]$. Let $\tilde{f_X}$ be the density of X_1 under $\tilde{P}$,

$$L = \prod_{j=1}^{n} \frac{f_X(X_j)}{\tilde{f_X}(X_j)}$$

Then $z = E_{\tilde{P}}[ZL]$ and $\tilde{z}$ has the same form as before.

Lemma 2.1. *Let* $X : (\Omega, \mathcal{F}) \to (\Sigma, \mathcal{G})$ *and* ν *be a measure over* $(\Omega, \mathcal{F})$*. Let* $\mu = \nu \circ X^{-1}$ *be the induced measure over* $(\Sigma, \mathcal{G})$ *and* g *a* $\mu - integrable$ *function over* $(\Sigma, \mathcal{G})$ *(i.e* $\int_\Sigma |g| d\mu < \infty$*). Then, for all* $G \in \mathcal{G}$

$$\int_{X^{-1}(G)} g(X(\omega)) d\nu(\omega) = \int_G g(\sigma) d\mu(\sigma)$$

Example 2.4. If $\nu = P$ is a probability measure, $\mu = P_X$ and $G = \sigma = \mathbb{R}$ then:

$$\int_{\Omega} g(X(\omega))dP(\omega) = \int_{\mathbb{R}} g(t)dP_X(t)$$

This integral is denoted by $E[g(X)]$.

2.2.1 Some Remarks

- The previous theorem ($\frac{\tilde{f}}{f} = \frac{\tilde{f}_z(z)}{f_z(z)}$) can be directly shown with this lemma. In the example where $X_1, X_2, ..., X_k$ are independent with density f_X under P and independent with density $\tilde{f_X}$ under $\tilde{P}$ and where $Z = g(X_1, ...X_k)$. Then $z = E_p[Z] = E_{\tilde{P}}[ZL]$ where $L = \prod_{i=1}^{k} \frac{f_X(X_j)}{\tilde{f_X}(X_j)}$. This is due to the Lemma.

  $$z = \int_{\mathbb{R}^k} g(t_1, ..., t_k) \prod_{j=1}^{k} \tilde{f_X}(t_j)dt_1...dt_k \qquad = \int_{\mathbb{R}} \gamma dh(\gamma)$$

  Where h is the induced densty of $g(X_1, ..., X_k)$. Once we have these representations we ask ourselves w.r.t whom should we take the expectation. According to the choice what we make we will obtain a different likelihood ratio.

  $$\begin{aligned} z &= \int_{\mathbb{R}^k} g(t_1, ..., t_k) \prod_{j=1}^{k} \frac{f_X(t_j)}{\tilde{f_X}(t_j)} \prod_{j=1}^{k} \tilde{\tilde{f}}_X(t_j)dt_1...dt_k \\ &= E_{\tilde{P}}[g(X_1, ..., X_k)L] \end{aligned}$$

- Variance reduction can be obtained only by appropiate choices of the importance sampling distribution.

Example 2.5. Let $z = \int_0^\infty xe^{-x}dx = 1$. Under P_θ, $Z \sim Ex(\theta)$, $\theta > 0$. On the other hand

$$\begin{aligned} L_\theta &= \frac{e^{-z}}{\theta e^{-\theta z}} \\ &= \frac{1}{\theta} e^{-(1-\theta)} z \\ &= E_{P\theta}[ZL_\theta] = E_P[Z] \\ &= z \quad \text{where } P = P_1 \end{aligned}$$

Now the second moment is given by:

$$\begin{aligned} E_{P_\theta}[(ZL_\theta)^2] &= \frac{1}{\theta^2} \int_0^\infty x^2 e^{-2(1-\theta)x} \theta e^{-\theta x} dx \\ &= \frac{1}{\theta} \int_0^\infty x^2 e^{-(2-\theta)x} dx \\ &= \begin{cases} \frac{2}{\theta(2-\theta)^2} & \text{if } \theta \in (0, 2) \\ \infty & \text{if } \theta \geq 2 \end{cases} \end{aligned}$$

Consequently:

$$Var_{P_\theta}(ZL_\theta) = E_{P_\theta}[(ZL_\theta)^2] - E^2_{P_\theta}[ZL_\theta = E_{P_\theta}[(ZL_\theta)^2] - 1$$

We search for the θ maximizing $\theta(2-\theta)^3$, which is $\theta = \frac{1}{2}$. Now since $Var_P(Z) = 1$, then $Var_{P_{\frac{1}{2}}} = \frac{2}{\frac{1}{2}(2-\frac{1}{2})^2} - 1 = \frac{5}{27}$

If L is $P - a.s.$ non negative and $E_P[L] = 1$, then

$$\tilde{P}(A) = E_P[I_A L]$$

is a probability measure since given $A \subset \Omega$

$$E_P[I_A L] = \int_A L dP$$

and

$$0 \leq E_P P[I_A L] \leq E_P[I_\Omega L] = E_P[L] = 1$$

given $A \subset \Omega$

Example 2.6. Let $X \sim \mathcal{N}(0,1)$ under P and $L = e^{\mu(X-\mu)-\frac{\mu^2}{2}}$

$$\begin{aligned} E_P[L] &= e^{-\frac{\mu^2}{2}} E[e^{-\mu\mathcal{N}(0,1)}] \\ &= e^{-\frac{\mu^2}{2}} e^{\frac{\mu^2}{2}} \\ &= 1 \end{aligned}$$

Theorem 2.4 (Main Result)**.** *Let P and P^* be two probability measures s.t $\frac{dP^*}{dP} = \frac{|Z|}{E_P[|Z|]}$, $\quad P^*(A) = E_P[I_A \frac{dP^*}{dP}]$.*
$\forall A$ measurable and $L^* = \frac{dP}{dP^*} = \frac{E_P[|z|]}{|Z|}$ exists over $\{Z \neq 0\} \setminus$

Then, the importance sampling estimator ZL^* under P^* has smaller variance than the estimator ZL under $\tilde{P}$. i.e

$\forall \tilde{P}$ s.t $P << \tilde{P}$ over $\{Z \neq 0\}$, $L\frac{dP}{d\tilde{P}}$ exists over $\{Z \neq 0\}$ and

- *$Var_{P^*}(ZL^*) \leq Var_{\tilde{P}}(ZL)$*

- *If $Z \geq 0 P - a.s$, then $Var_{P^*}(ZL^*) = 0$*

Example 2.7 (Compound Poisson Process)**.** Let $Y_1, Y_2, ...$ be independent random variables with common density f and let $\{N_t\}_{t \geq 0}$ be an independent Poisson counting process with rate $\lambda > 0$, under P. Then $X_t = \sum_{i=0}^{N_t} Y_i$, $\forall t \geq 0$ with $Y_0 = 0$, is a compound Poisson process. Let $D_1 + D_2 + ... + D_n$ denote the time of the $n - th$ ocurrence of the Poisson process, for $n = 1, 2, ...$

Assume that $X_{t t \geq 0}$ remains a compound Poisson process under $\tilde{P}$, however with summands density $\tilde{f}$ and Poisson parameter $\tilde{\lambda}$. Then, using the likelihood ratio process expressing P in terms $\tilde{P}$ can be found as:

$$L_t = \begin{cases} \frac{e^{-\lambda R_t}}{e^{-\tilde{\lambda} R_t}} \prod_{n=1}^{N_t} \frac{f(Y_n)}{\tilde{f}(Y_n)} \frac{\lambda e^{-\lambda D_n}}{\tilde{\lambda} e^{-\tilde{\lambda} D_n}}, \quad if \quad N_t \geq 1, \\ \frac{e^{-\lambda R_t}}{e^{-\tilde{\lambda} R_t}} \quad if \quad N_t = 0, \end{cases}$$

2.2.2 Exponential Change of Measure

We have included this section related to risk measures based on a summary of importance sampling presented in (Gatto, 2014a)

Let $X \in R^d$ a random vector, with cumulant generating function $K_X(v) = logE_P[e^{\langle v,X\rangle}]$, for $v \in \mathbb{R}^k$. Let $\alpha dom(k) := \{v \in \mathbb{R}^d : K(v) < \infty\}$, then the probability measure P_α such that

$$\frac{dP}{dP_\alpha} = exp\{-\langle \alpha, X\rangle + K(\theta)\}$$

is equivalent to P on the $\sigma - algebra$ generated by X and it is called the exponential tilt of P.

2.2.3 Random walk

Considering the previous example and assume that $K'(0) < 0$, that is $E_P[X_1] < 0$. Thus $\alpha(0)$ is the solution in v of $K'(v) = 0$ and we define β as the positive solution of $K(v) = 0$, which is called adjustment coefficient or Lundberg exponent. For some $X > 0$, define the hitting time.

$$T_x = \begin{cases} \min\{n \geq 1 | S_n \leq x\}, & \text{if the infimium exists} \\ \infty & \text{otherwise} \end{cases}$$

The hitting probability is given by

$$\theta(x) = P(T_x < \infty) = E_{P_\alpha}[I\{T_x < \infty\} exp\{-\alpha S_{T_x} + S_{T_x} K(\alpha)\}]$$

From here, $S_n \to \infty$, $P_\alpha - a.s$, implies $T_x < \infty$, $P_{\alpha - a.s}$. We can even show: :

$$P_\alpha[T_x < \infty] = 1 \Leftrightarrow E_{P_\alpha}[X_1] \geq 0 \Leftrightarrow \alpha \geq \alpha(0)$$

Thus any choice of the tilting parameter α such that $\alpha \geq \alpha(0)$ is relevant and yields the importance sampling estimator $\exp\{-\alpha S_{T_x} + T_x K(\alpha)\}$, under P_α. However the optimal choise is $\alpha = \beta$. We define the overshoot $D_x = S_{T_x} - x$, over $\{T_x < \infty\}$. Then

$$\theta(x) = e^{-\beta x} E_{P_\beta} e^{-\beta D_x}$$

this estimator, was suggested by Siegmund (Siegmund, 1976)and has bounded relative error.

2.2.4 The Compound Poisson Risk Process.

We consider the Compound Poisson risk process given in 1.1, is the risk process of an insurance company. We consider the time of ruin in 1.4 The cumulant generating function of the loss process L_1 is given by

$$K(v) = \lambda\{M_X(v) - 1\} - v$$

for $v \in \mathbb{R}$. The exponentiallly tilted probability P_α P_α is defined with the likelihood ratio process

$$exp\{-\alpha S_t + tK(\alpha)\} \ \ \forall t \geq 0$$

and the probability of ruin is given by

$$\theta(x) = P[T_x < \infty] = E_{P_\alpha}[I\{T_x < \infty\}exp\{-\alpha S_{T_x} + T_x K(\alpha)\}]$$

As before we define β as the positive solution in v of $K(v) = 0$, then we have $E_{P_\beta}[S_1] = K'(\beta) > 0$ which means that ruin is certain under P_β. As we previously has defined the overshooy as $D_x = S_{T_x} - x$, over $\{T_x < \infty\}$. Then

$$\theta(x) = e^{-\beta x} E_{P_\beta}[e^{-\beta D_x}]$$

that also has bounded relative error, as $x \to \infty$.

2.3 Control Variables

The following random variable C is a control variable if it is positively or negatively correlated with the random vriable Z, viz the initial Montecarlo estimator of $z = E[Z]$, and if its expectation $\mu_C = E[C]$ is known and easily computed.\

Let $D_\alpha(Z,C) = Z - \alpha(C - \mu_C)$ for any coeficient $\alpha \in \mathbb{R}$, then:

$$E[D_\alpha(Z,C)] = z$$

and the variance is given by:

$$Var(D_\alpha(Z,C)) = Var(Z) + \alpha^2 Var(C) - 2Cov(Z,C)$$

Therefore, if $\alpha^2 Var(C) - 2\alpha Cov(z,C) \leq 0$, then $Var(D_\alpha(Z,C)) \leq Var(Z)$

- Result:\

$$\min_{\alpha \in \mathbb{R}} = \{1 - Corr^2(Z,C)\}Var(Z)$$

The minimum is attained at $\alpha = \alpha_0 = \frac{Cov(C,Z)}{Var(Z)}$. Thus $\alpha_0 C$ is a good approximation to Z up to an additive constant. The control variate estimator is $Z - \alpha_0(C - \mu_C)$ and the control variable algorithm is given by:

$$\tilde{z}_{CV} = \frac{1}{n}\sum_{j=1}^{n}(Z_j - \alpha_0 Cj) + \alpha_0 \mu_C$$

Chapter 3

Monte Carlo Estimators for VaRu and TVaRu of the Maximal Aggregated Loss.

3.1 Introduction

In the previous chapters, we have reviewed important definitions and results regarding to risk theory and variance reduction techniques. The goal of this chapter is to present Monte Carlo Estimators for the Risk Measures presented in chapter one, the Value at Ruin and the Tail Value at Ruin.

First, we show the results considering exponential random losses for the Cramer Lundberg risk model. Afterthat, we present an extension of the results considering random losses following a Weibull distribution with Weibull index bigger and smaller than one, since the index equals to one corresponds to the exponential distribution. The estimations that we present in this chapter are based on an adequate estimation of the probability of ruin in the tails.

3.2 Monte Carlo Estimator for the Value at Ruin.

As we stated in 1.6

$$\psi(r_0) = 1 - F_S(r_0)$$

Where $S = \sup_{t\geq 0} L_t$ and L_t represents the loss process. Moreover, we note that the probability of ruin

$$\psi(r_o) = P(\inf_{t\geq 0} Y_t < 0)$$

and also we have that the time of ruin is given by:

$$T = \begin{cases} \inf\{t > 0 : Y_t < 0\} & \text{If the infimum exists,} \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

From this, we can conclude that

$$\psi(r_0) = P(S > r_0) = P(T < \infty) = 1 - F_S(r_0)$$

T is the first time when the reserve Y_t is greater than the initial capital r_0.

For large values of r_0, the computation of $P(S > r_0)$ is a rare event and it is not too obvious to estimate it, since we do not know how to simulate S in infinite time and we do not know when the random variable reaches its supremum. For that reason we work with the equivalent expression $\psi(r_0) = P(T < \infty)$.

With the estimated probability of ruin, we can construct the distribution function of the maximal aggregate loss S and numerically find the root of the equation

$$\hat{F}(VaRu(\epsilon)) = 1 - \epsilon$$

in order to get the estimator for the Value at Ruin.

3.3 Monte Carlo estimator for the probability of ruin

Since our previous results, we can consider this problem as the estimation of $P(T < \infty)$, the estimation of this quantity using the Monte Carlo method, is relatively straightforward. In each simulation step up to time τ and the result of the $i-th$ simulation Z_i is equal to 1 if ruin occurs between $[0, \tau]$ and 0 if not. After n runs the Monte Carlo estimator is given by

$$\hat{Z} = \frac{1}{n} \sum_{i=1}^{n} Z_i$$

This is summarized in the next algorithm:

```
  Data: $Z_1, ..Z_n$
  Result: Monte Carlo Estimator of the probability of ruin $\psi(r_0)$
1 Initialize the simulation setting $Y_0 = r_0$ and $t = 0$
2 Generate an exponential interarrival time $I$ with parameter $\lambda$ and a random loss
    with a distribution $F$.
3 If $t + I > \tau$ return $Z_i = 0$. If not, $Y_t = Y_t + cI - X$ and If $Y_t < 0$ return $Z_i = 1$.
    Set $t = t + I$. Otherwise, return to step 2.
```

Algorithm 2: Montecarlo Estimator of the Probability of Ruin

3.4 Importance Sampling Estimator of the probability of ruin.

We have implemented importance sampling via the exponential change of measure of the random variable T. We consider:

$$\begin{aligned} z(x) &= \psi(x) = P(T < \infty) \\ &= E_\theta[L_{T,\theta} \mathbb{1}_{\{T<\infty\}}] \\ &= E_\theta[e^{-\theta L_T + T\kappa(\theta)} \mathbb{1}_{T<\infty}] \end{aligned} \tag{3.2}$$

In order to choose a value of θ, we need to ensure that $P_\theta(T < \infty) = 1$ that follows from the fact that the changed drift in the exponential change of measure is given by $\mu_\theta = \kappa'(\theta)$. Therefore, we can perform simulation by Monte Carlo method considering $Z(x) = L_{T,\theta}$.

Another important fact is that

$$\begin{aligned} M_{X_\theta}(v) &= \int_{\mathbb{R}} e^{vx} dF_\theta \\ &= \int_{\mathbb{R}} e^{vx} e^{\theta x - \kappa(\theta)} dF \\ &= \int_{\mathbb{R}} e^{(v+\theta)x} e^{-\log M_X(\theta)} dF \\ &= \frac{1}{M_X(\theta)} \int_{\mathbb{R}} e^{(v+\theta)x} dF \\ &= \frac{M_X(v+\theta)}{M_X(\theta)} \end{aligned} \tag{3.3}$$

Aditionally, we conclude that:

$$\kappa_{X_\theta}(v) = \kappa(v+\theta) - \kappa(\theta)$$

In the insurance context the positive solution of $\kappa_X(v) = 0$ is called the adjustment coefficient and we note this by r. This value is optimal and ensures that $P_\theta(T < \infty) = 1$. Considering $\theta = r$ in the expression 3.2 the variable T is vanished and we get the next expression.

$$\begin{aligned} \psi(r_0) &= E_\theta[e^{-\theta L_T + T\kappa(\theta)} \mathbb{1}_{T<\infty}] E_r[e^{-rL_T}] \\ &= E_r[e^{-r(D(r_0)+r_0)}] \\ &= e^{rr_0} E_r[e^{rD(r_0)}] \end{aligned}$$

Where we used the fact that $D(r_0) = L_t - r_0 = -Y_t$, that is called the overshoot

3.5 Monte Carlo Estimator of the Tail Value at Ruin.

According to the definition of Tail Value at Ruin given in chapter 1, we have that

$$TVaRu(\epsilon) = E[S|S > VaRu(\epsilon)]$$

we can express this quantity as follows:

$$\begin{aligned}
TVaR(\epsilon) &= E[S|S > VaRu(\epsilon)] \\
&= \frac{E[S\mathbb{1}_{\{S>VaRu(\epsilon)\}}]}{P(S > VaRu(\epsilon))} \\
&= \frac{E[VaRu(\epsilon)\mathbb{1}_{\{S>VaRu(\epsilon)\}} + [S - VaRu(\epsilon)]^{+}]}{P(S > VaRu(\epsilon))} \\
&= VaRu(\epsilon) + \frac{1}{P(S > VaRu(\epsilon))} E[S - VaRu(\epsilon)]^{+} \\
&= VaRu(\epsilon) + \frac{1}{1-\epsilon} E[S - VaRu(\epsilon)]^{+}
\end{aligned} \tag{3.4}$$

Now we consider the next result:

$$\begin{aligned}
E[S - VaRu(\epsilon)]^{+} &= E[(X - a)\mathbb{1}_{\{S-VaRu(\epsilon>0)\}}] \\
&= E[(S - VaRu(\epsilon)\mathbb{1}_{\{S>VaRu(\epsilon)\}})] \\
&= E[S\mathbb{1}_{\{S>VaRu(\epsilon)\}}] - E[VaRu(\epsilon)\mathbb{1}_{\{S>VaRu(\epsilon)\}})] \\
&= E[S\mathbb{1}_{\{S>VaRu(\epsilon)\}}] - VaRu(\epsilon)P(S > VaRu(\epsilon)) \\
&= E[S\mathbb{1}_{\{S>VaRu(\epsilon)\}}] - VaRu(\epsilon)(1-\epsilon)
\end{aligned} \tag{3.5}$$

Consequently we get

$$TVaRu(\epsilon) = \frac{1}{1-\epsilon}\int_{VaRu(\epsilon)}^{\infty} sf(s)ds \tag{3.6}$$

However, since we do not know the distribution of S we can not generate S direclty. However, we know a relation between the previous expression and the probability of ruin, whose estimator was computed at the beginning of this chapter. We proceed in the next way.

$$\int_{VaRu(\epsilon)}^{\infty} sdF(s) = -\int_{VaRu(\epsilon)}^{\infty} sd\{1 - F(s)\}$$

Using partial integration we have

$$\int_{VaRu(\epsilon)}^{\infty} \{1 - F(s)\}ds - [s\{1 - F(s)\}]|_{VaRu(\epsilon)}^{\infty}$$

Since $\psi(x) = 1 - F(x)$ and $F(s) \to 1$ when $s \to \infty$

$$\int_{VaRu(\epsilon)}^{\infty} \{\psi(s)\}ds - VaRu(\epsilon) \tag{3.7}$$

Using the last expression in our previous results give us:

$$TVaRu(\epsilon) = \frac{1}{1-\epsilon}\int_{VaRu(\epsilon)}^{\infty} \psi(s)ds + VaRu(\epsilon) \tag{3.8}$$

To compute the integral related to the last expression, we fit the simulated points by importance sampling in the tail, starting in the $VaRu$ estimated until the intersection of the model fitted and the horizontal line $F_S(r_0) = 1$. This procedure is performed with the points obtained after importance sampling, due to the lower variance reduction. The idea behind this is that the simulated points has lower variability than the Crude Monte Carlo method.

3.6 Exact probabilities of ruin in infinte time using the Pollaczeck-Khinchine formula

For the Heavy tailed case, we have followed a different approach than generate the risk process. We consider the next representation, let R_1 be the height of the first record of $\{L_t\}_{t\geq 0}$. Then R_1 is the overshoot whose d.f is

$$F_R(y) = \frac{1}{\mu}\int_0^y \{1 - F(x)\}dx \quad \forall y \geq 0$$

Then R_2 is the height of the second record minus R_1, R_3 the height of the third record minus $(R_1 + R_2)$ and so on. Due to the independency and stationarity of the increments of $\{L_t\}_{t\geq 0}$ $R_1, R_2, ...$ are i.i.d.

Let N, the number of records (in infinite time horizon), the probability of no records is $p = 1 - \psi(0) = R(0)$. If observe a record is a failure and observing no records is a success, then N is the number of failures to obtain one sucess. It means that $N \sim Geom(\rho)$ and $P(N = n) = (1-\rho)^n\rho \quad n = 0, 1, 2, ...$ The individual experiments of geometric distributions are indeed independent and with the same success probability. Because $\{L_t\}_{t\geq 0}$ has independent and stationary increments. Thus

$$S = \sum_{j=0}^{N} R_j \quad , \; R_0 = 0$$

and

$$P(N = n) = \psi^n(0)\{1 - \psi(0)\} = \frac{\beta}{(1+\beta)^{n+1}} \quad n = 0, 1, ...$$

where β is the security loading.

3.7 Monte Carlo simulation using the Pollaczeck-Khinchine formula

We consider the compound Poisson risk model, let $X_1, X_2, ...$ be i.i.d with common density $F_0(x) = \frac{\bar{F}(x)}{\mu_F}$ that is the integrated tail distribution of F. Let $S_n = X_1 + X_2 + ... + X_n$ and let K be independent and geometric with parameter ρ then

$$P(K = k) = (1 - \rho)\rho^k$$

The Pollaczeck-Khinchine formula may be write as

$$\psi(u) = P(M > u)$$

where $M = S_k$. Thus $\psi(u) = z = z(u) = EZ$ with $Z = I_{\{M>u\}}$ may be generated with the next algorithm.

```
Data: $Z_1, ..Z_n$
Result: Monte Carlo Estimator of the probability of ruin $\psi(r_0)$ using the
        Pollaczeck-Khinchine formula
Generate $K$ as geometric

    $P(K = k) = (1 - \rho)\rho^k$

 Generate $X_1, X_2, ..., X_k$ from the density $F_0(x)$ and let $M \leftarrow S_k$. if $M > u$
 then
    $Z \leftarrow 1$
else
    $Z \leftarrow 0$
end
$Z_{MC} = \frac{1}{n} \sum_{i=1}^{N} Z_j$
```

Algorithm 3: Monte Carlo Estimator of the probability of ruin

As crude Monte Carlo estimator is not efficient for large values of u, for that reason we need to combine it with some variance reduction methods. In this case we are going to use the control variate estimator proposed in (S Asmussen & Binswanger, 1997). and conditional simulation algorithms. Both approaches were found in the work of (S Asmussen & Binswanger, 1997) as well as some references on (H. Albrecher et al., 2017)

3.7.1 Asmussen Kroese estimator for the Probability of Ruin

```
Result: Monte Carlo Estimator of the probability of ruin $\psi(r_0)$ using the
        Pollaczeck-Khinchine formula
Generate $K = K^{(i)}$ as geometric

    $P(K = k) = (1 - \rho)\rho^k$

 Generate $X_1, X_2, ..., X_{k-1}$ from the density $F_0(x)$ and let $M \leftarrow S_k$.
Calculate $M_{K-1}^{(i)} = max\{X_1^{(i)}, ..., X_{K-1}^{(i)}$ and $S_{K-1}^{(i)} = \sum_{j=1}^{K-1} X_j^{(i)}$
Set $Z^{(i)}(u) = K^{(i)} \bar{F}_0(max\{M_{K-1}^{(i)}, u - S_{K-1}^{(i)}\})$
$Z_{MC} = \frac{1}{n} \sum_{i=1}^{n} Z^{(i)}(u)$
```

Algorithm 4: Asmussen-Kroese estimator of the probability of ruin

The performance of the previous algorithm can be improved considering control variates. This aditional variation is proposed in (Søren Asmussen & Kroese, 2006). In this work, since in the subexponential distribution case, the tail probability of the sum is asymptotically equal to n times the tail probability of each single random variable, $N\bar{F_0}(u)$ is proposed as a control variate. Then the Control variate estimator has the form:

$$Z_{CV} = N\bar{F_0}(max\{M_{K-1}, u - S_{K-1}\}) - \bar{F_0}(u)) + E(N).\bar{F_0}(u) \tag{3.9}$$

3.7.2 Estimation of the probability of ruin by conditional simulation

Besides the control variate approach to estimate the probability of ruin, we can use a conditional simulation algorithm.

Conditional Simulation

The next algorithms has been proposed The problem is to estimate $z = E[Z]$, where Z is a random variable on $(\Omega, \mathcal{F})$, let W be a $\mathcal{F} - measurable$ r.v., then the conditional Monte Carlo estimator of z is:

$$\begin{aligned} Z_{cond} &= E[E[Z|W]] \\ &= E[Z] \\ &= z \end{aligned}$$

Furthermore,

$$\begin{aligned} Var(Z) &= Var(E[E[Z|W]]) + E[Var(Z|W)] \\ &= Var(Z_{cond}) + \overset{\geq 0}{\underbrace{E[Var(Z|W)]}} \\ &\geq 0 \end{aligned}$$

Therefore, this method always leads to a reducction of variance, but it will not always to find an approaipate Z_{cond} that works for this purpose.

Conditional Monte Carlo algorithm

$$\begin{aligned} \psi(u) &= P(X_1 + X_2 + ... + X_K > u) \\ &= E[P(X_1 + X_2 + ... + X_K > u)|K = k, X_1, X_2, ..., X_{k-1}] \\ &= E[\bar{B}_0(u - X_1 - ... - X_{k-1})] \end{aligned}$$

Thus we only generate $X_1, ..., X_{k-1}$, compute $Y = u - X_1 - ... - X_{k-1}$ and we set $Z = \bar{B}_0(Y)$, the probability that the next claim causes ruin. We obtain the next algorithm.

```
  Result: Probability of ruin $\psi(u)$
1 Generate $K_i \sim Geo(\rho)$
2 Generate $X_1^i, X_2^i, ..., X_{K_i-1}^i$ from the density $b_0$ and let
    $Y_i = u - X_1^i - X_2^i - ... - X_{K_i-1}^i$
3 Let $Z_i = \bar{B}_0(Y_i)$, $Z_i = 1 \quad if \quad Y_i < 0$
4 Repeat steps 1 to 3 n times.
5 Estimate $\psi(u)$ by $Z_{MC} = \frac{1}{n}\sum_{i=1}^{n} Z_i$
```

Algorithm 5: Conditional MC algorithm

We deal with the rare event by using the idea given before. We write the probability of ruin $\psi(u)$ as conditional expectation

Conditional Monte Carlo algorithm using order statistics.

The next algorithm, is not much complicated than the conditional one. It was proposed in (S Asmussen & Binswanger, 1997). The idea is that for subexponential distributions essentially one large claim causes ruin. This problem is solved by discarding the largest of the X_i and considering only the remaining ones.

```
  Result: Probability of ruin $\psi(u)$
1 Generate $K_i \sim Geo(\rho)$.
2 Generate $X_1^i, X_2^i, ..., X_{K_u-1}^i$ from the density $b_0$ and let
    $Y_i = u - X_{(1)}^i, X_{(2)}^i, ..., X_{(K_u-1)}^i$ and $m_i = X_{(k_i-1)}^i$
3 Let $Z_i = \frac{\bar{B}_0(Y_i \vee m_i)}{\bar{B}_0(m_i)}$.
4 Repeat steps 1 to 3 n times.
5 Estimate $\psi(u)$ by $Z_{MC} = \frac{1}{n}\sum_{i=1}^{n} Z_i$
```

Algorithm 6: Conditional MC algorithm

3.8 Some efficiency criteria

In order to know the performance of a Monte Carlo estimator for small probabilities, we introduce two performance criteria, base on the relative length of the corresponding confidence interval. We require that for a sequence of small probabilities, the variance of the estimator should converge faster to zero than the sequence itself. We consider a sequence of events $A(u)$ dependending on a parameter u with probabilities

$$P(A(u)) := z(u) \to 0 \quad for \quad u \to 0$$

Definition 3.1 (Bounded Relative Error)**.** An unbiased estimator $Z(u)$ of $z(u)$ is said to have a bounded relative error it if satisfies

$$\limsup_{u\to\infty} \frac{Var(Z(u))}{Z(u)^2} < \infty$$

Definition 3.2 (Logarithmic efficiency)**.** An unbiased estimator $Z(u)$ of $z(u)$ is said to be logarithmically efficient or asymptotically efficient if it satisfies

$$\limsup_{u\to\infty} \frac{Var(Z(u))}{Z(u)^{2-\epsilon}} = 0 \ \ \forall \epsilon > 0$$

In (S Asmussen & Binswanger, 1997), is shown that the conditional simulation algorithms are not efficient assymptotically when heavy tailed Weibull claim amounts are considered. However,is mentioned in that work that according to the numerical experience is convincing applying conditional simulation.

As we have seen in the algorithms presented in this chapter, we need to know how to simulate random variables R with the integrated tail distribution Weibull. For this purpose we are going to use an important method of simulation called the method of acceptance-rejection that is a general technique for generating random variables with agiven distribution.

3.9 The acceptance rejection method

Supposing that we are interested in simulate random variables with distribution F, whose density is given by f, for which $\forall X \in \mathbb{R}, f(x) = ch(x)g(x)$, where c is a constant, h has density function whose support is included in the support of f, and g is a function taking values on $[0,1]$, g is often called the shrinkage function and $c.h$ is the envelope.

```
Data: $f, c, h, g$
Result: Random variable with distribution $F$
Generate $U \sim U(0,1)$ and $Y$ with density $h$ independently.
Generate $U \leq g(Y)$ accept $Y$ as a realization of $f$. note that
  $U \leq g(Y) \Leftrightarrow U \geq \frac{f(Y)}{ch(Y)} \Leftrightarrow Uc.h(Y) \leq f(Y)$
If $U > g(Y) \rightarrow$ discard both $U$ and $Y$ and start again.
```

Algorithm 7: Acceptance Rejection algorithm

more details about this simulation method can be found on (Søren Asmussen & Glynn, 2007) and (Ross, 1990)

3.9.1 Acceptance Rejection method for generating random observations from the Weibull integrated tail distribution.

The integrated tail density is given by the next expression:

$$F_0(x) = \frac{1}{\mu_F} \int_0^x \bar{F}(t)dt$$

let us call its density as $f_0(x)$, note that in this case $F(x) = 1 - exp(\lambda x^\tau)$. Since we are considering the heavy tailed case $\tau \in (0,1)$ and $\mu_F = \lambda\Gamma(1 + \frac{1}{\tau})$. We consider $h(x)$ of the acceptance rejection algorithm, as the density of the exponential distribution with rate parameter λ. By the inverse transform method, If $U \sim U(0,1)$ we can generate an observation from the exponential distribution using

$$X = -\frac{1}{\lambda} log(1 - U)$$

.

We need to find c such that $f_0(y) \geq ch(y) \quad \forall y > 0$. i.e $c = \max \frac{f_0(y)}{h(y)}$.
let

$$g(y) = \frac{f_0(y)}{h(y)} = \lambda\mu_F \exp(-\lambda y + (\lambda y)^\tau)$$

$$\Rightarrow g'(y) = \mu_F \lambda^2 \exp(-\lambda y + (\lambda y)^\tau)\{\tau(\lambda y)^{\tau-1} - 1\}$$

.

Therefore $g'(y) = 0 \Rightarrow \tau(\lambda y)^{\tau-1} - 1 = 0$ since $\lambda^2 \mu_F \exp(-\lambda y + (\lambda y)^\tau) \neq 0$

$$\Rightarrow y = \frac{1}{\lambda} \tau^{\{-(\frac{1}{\tau-1})\}}$$

We can show that for this value that $g''(y) < 0$. In conclusion

$$c = \max \frac{f_0(y)}{h(y)} = \lambda \mu_F \exp(-\tau^{-(\frac{1}{\tau-1})} + \tau^{-(\frac{\tau}{\tau-1})})$$

and $\frac{f_0(y)}{h(y)} = \exp(-\lambda y + (\lambda y)^{\tau} - K)$ with $K = -\tau^{-(\frac{1}{\tau-1})} + \tau^{-(\frac{\tau}{\tau-1})}$.

Consequently, the algorithm for generating random variables from integrated tail Weibull distribution is the next one.

```
   Data: $\lambda, \tau$
   Result: Random variable with Integrated tail distribution Weibull
1  Generate $U_1 \sim U(0,1)$ and set $Y = -\frac{1}{\lambda} log(1 - U_1)$.
2  Generate $U_2 \sim U(0,1)$ and if $U_2 < \exp(-\lambda y + (\lambda y)^{\tau} - K)$ set $X = Y$, otherwise
     return to step 1.
```

Algorithm 8: Acceptance Rejection algorithm for generating Integrated tail Weibull distributed random variables.

For the estimation of the probability of ruin using the Pollaczek-Khinchine formula, the last algorithm is very important. Due to in the second step we alays need to generate random variables with the integrated tail distribution.

Chapter 4

Numerical Examples

In this chapter we present the numerical results obtained after implementing the algorithms proposed in chapter 3.

4.1 The method of Monte Carlo

4.1.1 Cramer Lundberg Risk model with exponential claim amounts.

As we have seen before, is not so easy to simulate the maximal aggregated loss, since the process is random we can not know when the process reaches its supremum or maximum.

For that reason, we proceed by simulating the risk process until ruin is reached. Then we store all the variables involved.

The next code performs the algorithm 2 given in section 3.3. The output of the function used is a list containing two objects. The first one called *data* that is a list containing n risk processes until time τ. The columns of the first object called *data* are the time, that is the random time of the claim following an exponential distribution, level: that is the Y_t, i.e the reserve at time t, the interarrival time that follows an exponential distribution and the amount of the claim, that for this example is exponentially distributed. The second object is a logical vector that means

$$TRUE = 1$$

if the process reached ruin i.e when the reserve is $Y_t \leq 0$ and

$$FALSE = 0$$

otherwise. We have specified a max time τ since we can not perform until infinity.

An example of the output of the code with rate of exponential interrival time $\lambda = 2$, security loading $\theta = 0.3$, claim rate of the exponential claim amount $\lambda_{claim} = 2$, initial capital $u0 = 5$, time horizon $T = 10$, Number of simulations $runs = 2$.

```
prob.ruin<-function(lambda,theta,claimrate,T,u0,runs){
  c <- (1 + theta ) * lambda /claimrate ;
  result <- c (); # Result
  data<-list()
  for (r in 1: runs )
  {
    time <- 0 # Keep track of the time
    level <- u0 # Keep track of the level
    interarrivaltime <- rexp (1, lambda )
    table<-NULL
    i<-1
    while ( time + interarrivaltime < T)
    {

      # A new claim arrives
      time <- time + interarrivaltime
      claim <- rexp (1, claimrate )

      # Update level
      level <- level + c* interarrivaltime - claim
      table<-rbind(table,c(time,level,interarrivaltime,claim))
      if( level < 0)
      {
        break ;
      }

      interarrivaltime <- rexp (1, lambda );

      i<-i+1
    }
    data[[r]]<-table
    result <- c( result , level < 0);
  }
  return(list(data,result))
}
example<-prob.ruin(lambda=2,theta=0.3,claimrate=2,u0=5,T=10,2)
colnames(example[[1]][[1]])<-c("time","level","inter.arrival.time",
                               "claim amount")
example
```

```
[[1]]
[[1]][[1]]
         time     level inter.arrival.time claim amount
 [1,] 1.306139  5.983107        1.306138805   0.71487335
 [2,] 1.513081  5.786008        0.206942036   0.46612390
 [3,] 1.862431  6.163184        0.349349963   0.07697867
 [4,] 2.130366  6.313628        0.267935068   0.19787165
 [5,] 2.729329  6.880402        0.598963542   0.21187873
 [6,] 2.823897  6.944416        0.094568047   0.05892472
 [7,] 4.817025  9.382364        1.993127729   0.15311752
 [8,] 6.321820 10.843042        1.504794558   0.49555513
 [9,] 6.701808 11.049632        0.379988733   0.28739488
[10,] 6.706679 10.683044        0.004870738   0.37291995
[11,] 7.218886 11.034379        0.512206290   0.31453374
[12,] 7.994975 11.684889        0.776089649   0.35840659
[13,] 9.855096 13.936539        1.860120448   0.16650624

[[1]][[2]]
         [,1]     [,2]      [,3]       [,4]
 [1,] 2.255470 7.915038 2.2554698 0.01707321
 [2,] 2.387754 7.621327 0.1322841 0.46568027
 [3,] 2.518967 7.728471 0.1312133 0.06343338
 [4,] 2.647943 7.497266 0.1289755 0.39887296
 [5,] 3.083395 7.955656 0.4354526 0.10769776
 [6,] 3.290652 7.675654 0.2072563 0.54943590
 [7,] 3.461647 7.863720 0.1709950 0.03422701
 [8,] 3.753011 7.351681 0.2913644 0.89081288
 [9,] 4.749244 7.413731 0.9962328 1.23305278
[10,] 5.051921 7.556109 0.3026771 0.25110203
[11,] 5.207139 7.257842 0.1552183 0.50005083
[12,] 5.346481 6.755603 0.1393416 0.68338262
[13,] 5.545584 6.843042 0.1991028 0.17139463
[14,] 5.749384 6.462124 0.2038006 0.64585905
[15,] 8.151954 8.856750 2.4025699 0.72871542
[16,] 8.619643 8.603803 0.4676892 0.86094282
[17,] 8.749269 8.562237 0.1296256 0.21007946
[18,] 9.171773 7.211791 0.4225037 1.89970047
[19,] 9.619876 7.403925 0.4481036 0.39040052


[[2]]
[1] FALSE FALSE
```

To compute the probability by the Monte Carlo estimator we consider the average of the elements of the vector *result* after a considerable number of runs. For the computation of the probability of ruin in infinte time horizon we consider a large value of T, we have chosen $T = 1000$.

We have compared the results of this Monte Carlo estimation with the analytical formula of the probability of ruin of the Cramer Lundberg Risk model with exponential claim amounts, in this case it should be approximately the same. For example with the same parameters given before, we perform 1000 runs with time horizon $T = 100$. We also include the equivalent solution using the actuar library of the R software[1]. To summarize the function of this we present the next table, where every row show us the probability of ruin given an initial capital $u_0 = 5, 10, 15, 20$ for a different number of simulations $100, 1000, 10000$.

	Probability of Ruin
Monte Carlo Estimator	0.0720
Analytical Formula	0.0765
actuar package	0.0765

	Number of runs			
u_0	100	1000	10000	Erlang
5	0.1600	0.1380	0.1602	0.1574
10	0.0200	0.0280	0.0291	0.0297
15	0.0000	0.0080	0.0047	0.0056
20	0.0000	0.0020	0.0008	0.0011

We can conclude from the last table that if we increase the initial capital, we will need more simulations. For example if we suppose that the probability of ruin is 0.001, it means that we would need at least 1000 simulations in order to obtain one ruin.

Something useful also is to construct the distribution of the maximal aggreagate loss, given by the complement of the probability of ruin. For that we consider a vector of initial capitals and run the function given.

For this example we consider a vector of 40 initial capitals from $u_0 = 1$ to $u_0 = 40$

[1] Package with functions and data sets for actuarial science, more information can be found on https://cran.r-project.org/web/packages/actuar/index.html

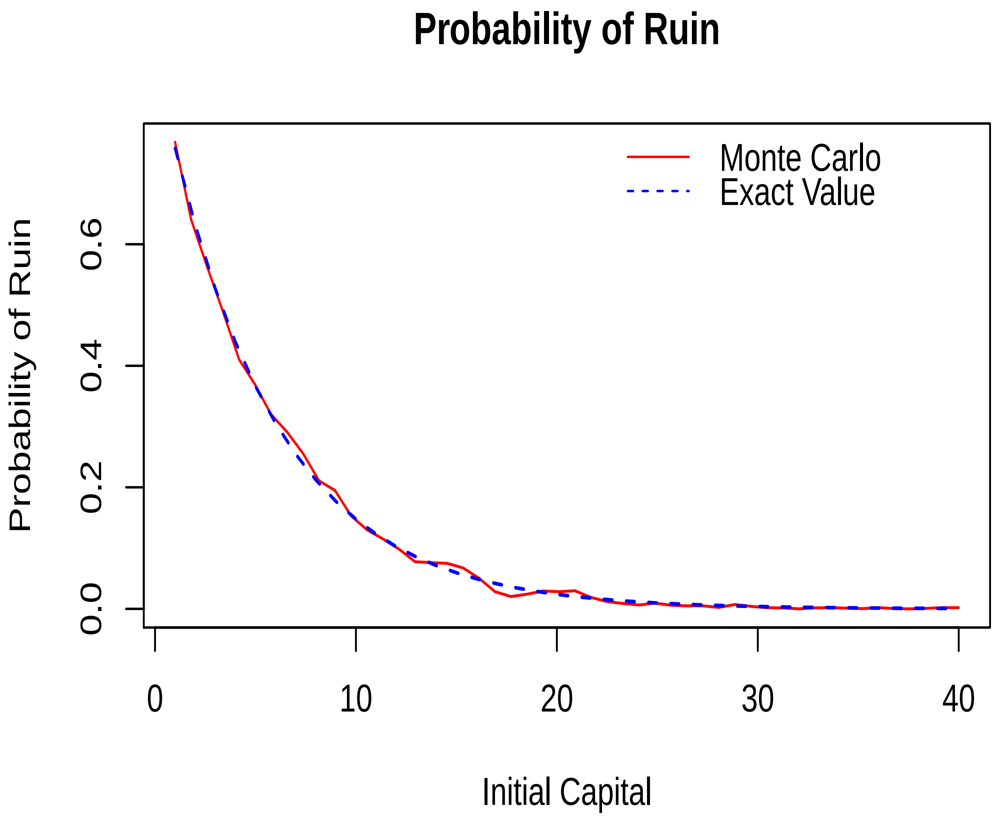


Figure 4.1: Distribution of the maximal agreggated loss

4.2 Estimation of the probability of ruin via Importance Sampling.

For the estimation using the importance sampling approach, we need to compute the Lundberg coefficient.

4.2.1 The adjustment coefficient

In the insurance context, the Lundberg coefficient that is the positive solution of $M_S(v) = 1$ coincides with the adjustment coefficient of the Cramer Lundberg Risk Process.

Definition 4.1 (adjustment coefficient)**.** The adjustment coefficient r is the positive solution in v of

$$E[e^{L_1 v}] = 1$$

We determine the adjustment coefficient considering exponential claim amounts.

Where $L_1 = Z_1 - c$ is the loss process at time 1. Note that

$$\begin{aligned} M_{L_1}(v) &= M_{Z_1}(v)e^{-vc} \\ \Leftrightarrow exp\{\lambda(M_X(v)-1)\} &= e^{-vc} \\ \Leftrightarrow \lambda[M_X(v)-1] - vc &= 0 \\ \Leftrightarrow M_X(v) &= 1 + (1+\beta)\mu v \end{aligned}$$

Where M_X is the moment generating function of X_1 and $\mu = E[X_1]$. For the computation of the adjustment coefficient, we have generated the next code.

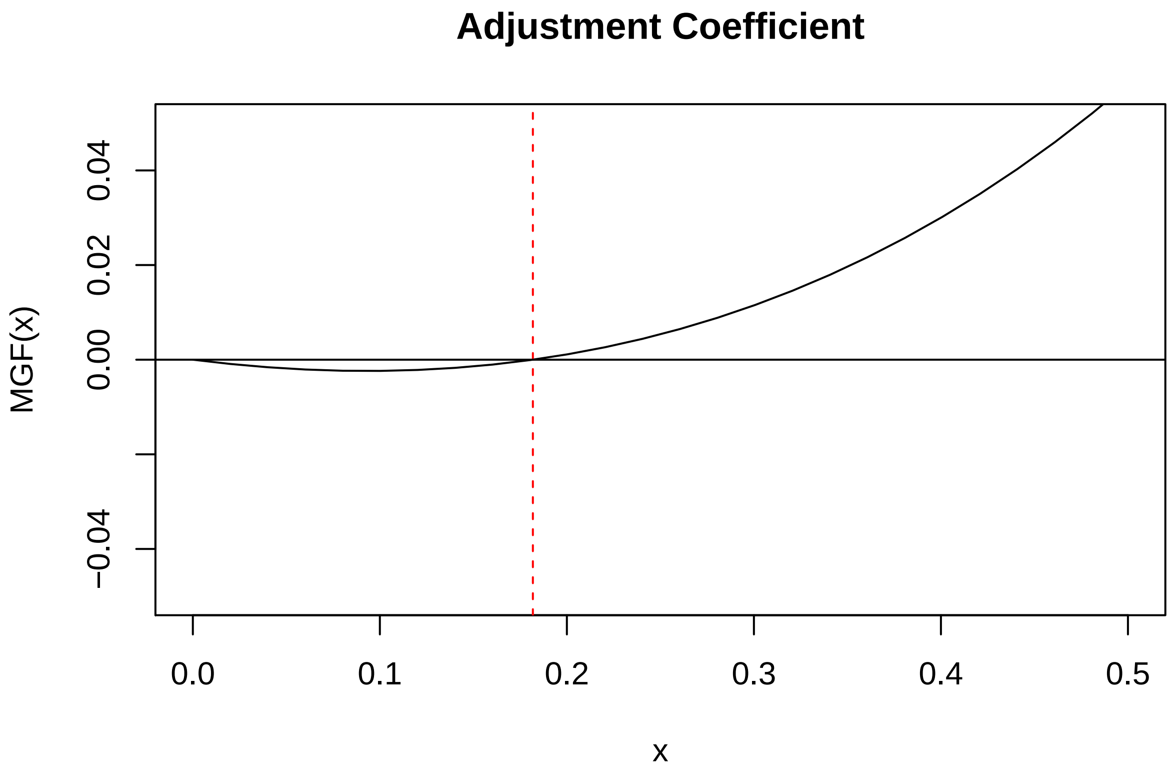
The parameters considered are: $\lambda = 2, \beta = 0.1$, claim rate= 2, $u_0 = 5, T = 1000$

```
##Define the moment generating function exponential claim amounts

   lambda=2
    theta=0.1
claimrate=2
u0=5
 T=1000

mgfexp1<-function(x)(1/(1-(1/claimrate)*x))
fun1<-function(x)(1/(1-(1/claimrate)*x))-((1+theta)*(1/claimrate)*x)-1
      x<-seq(0,10,length.out = 500)
      y<-fun1(x)
####premium rate
     c <- (1 + theta ) * lambda / claimrate

####Computation of the adjustment coefficient

r<-uniroot(fun1,interval=c(0.1,0.5))$root
r<-(claimrate)-(lambda)/c

#We check if our adjustment coefficient is correct with the
#function provided by the actuar package

adjCoef(mgf.claim = mgfexp1,premium.rate = c,
        upper.bound = 1,mgf.wait = 1/(1-0.5*x))
```

```
[1] 0.1818182
```

```
#Our solution
r
```

```
[1] 0.1818182
```



We see that the analytical solution coincides with the solution given by our code and the solution of an existent package.

With the value of the adjustment coefficient we generate the new rate λ_r that is obtained after the exponential tilt. Considering this value and since the compound poisson process after exponential tilt is again a compound Poisson process we can use the first code that computes the risk process until ruin is reached. Note that in this case we perform a small number of runs.

Let us see an example about the computation of the probability of ruin based on the Cramer Lundberg Process with exponential claim amounts. Since we know the analytical formula for this probability of ruin we can compare our results with the Monte Carlo approach and the Importance Sampling approach.

The parameters used for the next example are the next ones: initial capital $u_0 = 30$, security loading $beta = 0.1$, claim rate 2.

```
prob.ruin.erlang(30,beta=theta,claimrate=2)
```

```
[1] 0.003888018
```

We present the next plot that as a results of the simulations using the importance sampling method vs the Crude Monte Carlo method, we have included the time horizon $T = 1000$ and a maximum number of simulations 5000. The rate of the interarrival times $\lambda = 2$.

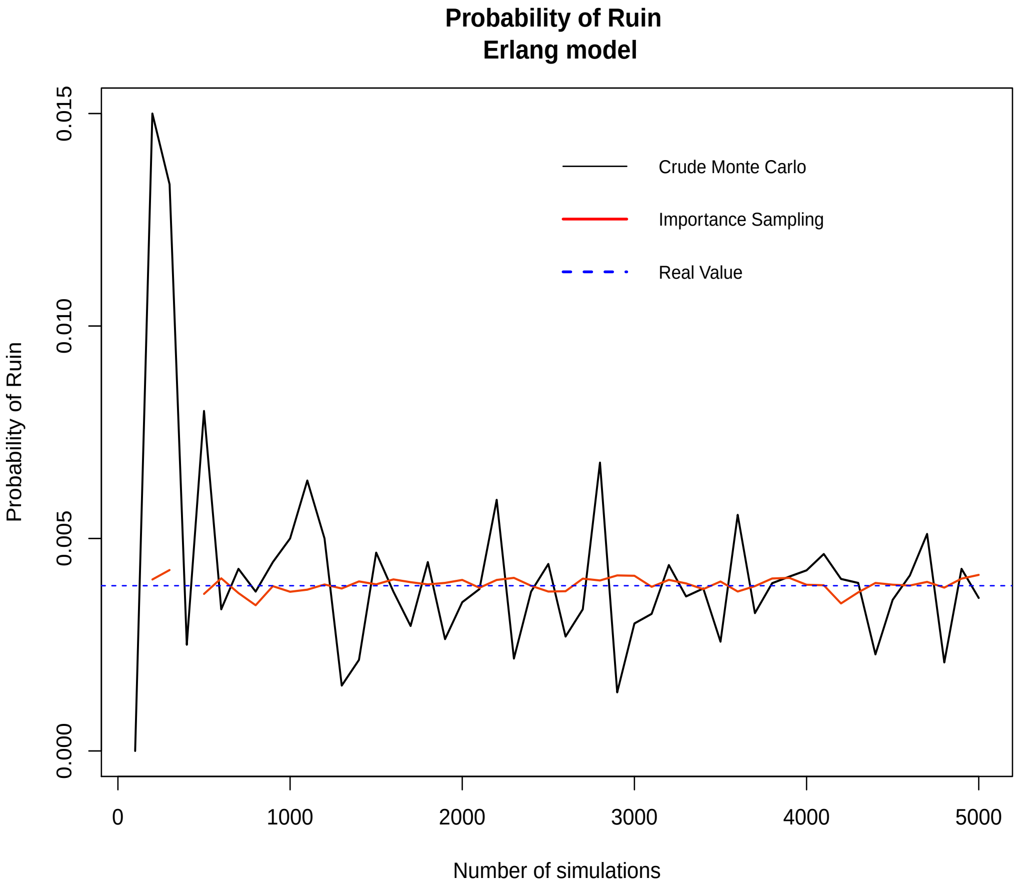


Figure 4.2: Importance Sampling Method vs Crude Monte Carlo for the computation of the Probability of Ruin for different number of simulations

We can see that for the Crude Monte Carlo estimator for different number of simulations the variance present a notorious oscillation. In contrast the red line that depicts the estimation using the importance sampling method is closer than the real value given by the blue segmented line. The importance sampling estimation presents smaller variance than the Crude Monte Carlo estimation.

4.3 Estimation of the Value at Ruin.

For the Value at Ruin $VaRu$ we fix the level $\alpha = 0.01$, i.e we are looking for the quantile at level $\epsilon = 0.99$. In order to do that, we generate the probability of ruin for several initial capitals. Since we know that $F_S(r_0) = 1 - \psi(r_0)$ we construct the distribution of the maximal aggregated loss S, the desired quantile will be the intersection between the distribution and the horizontal line $\epsilon = 0.99$.

First for comparison, we compute the $VaRu(0.99)$ using the explicit formula, that can be deducted from the fact that

$$\psi(x) = \frac{1}{1+\beta} e^{-rx}$$

Since we know that $F_S(x) = 1 - \psi(x)$ then

$$F_S(x) = 1 - \frac{1}{1+\beta} e^{-rx}$$

and the $\epsilon - quantile$ is given by

$$log[(1-\epsilon) \times (1+\beta) \times (-\frac{1}{r})]$$

```
is<-log((1-0.99)*(1+theta))*(-1/r)
is
```

```
[1] 24.80423
```

```
prob.ruin.erlang(is,beta=theta,claimrate=2)
```

```
[1] 0.01
```

The exact $VaRu$ at level 0.99 is equals to 24.804023.

The next plots show the result of the simulation schemes used. We compare the Monte Carlo simulation, represented by the blue points, with the importance sampling estimation represented by the red points. The segmented black line is the theoretical value of the probability of ruin computed with the analytical formula. Is evident that the simulation provided by the importance sampling approach is very similar to the exact value of the probability of ruin. In the first plot we see the distribution of the maximal aggregated loss S in a range of initial capitals $[0, 30]$. The second plot shows the same but in a smaller range in order to see better wich approximation is more accurate.

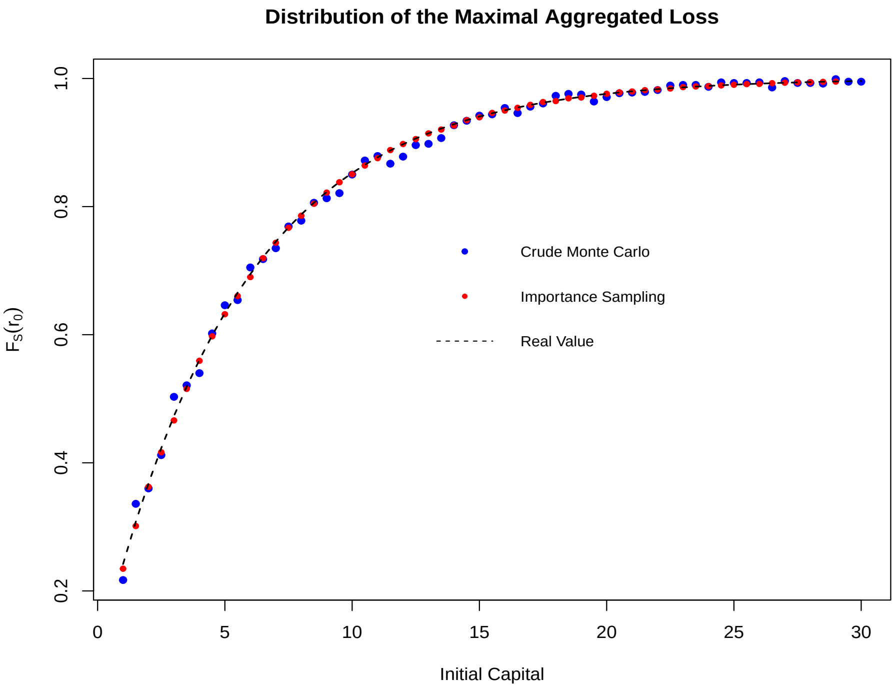


Figure 4.3: VaRu Estimation

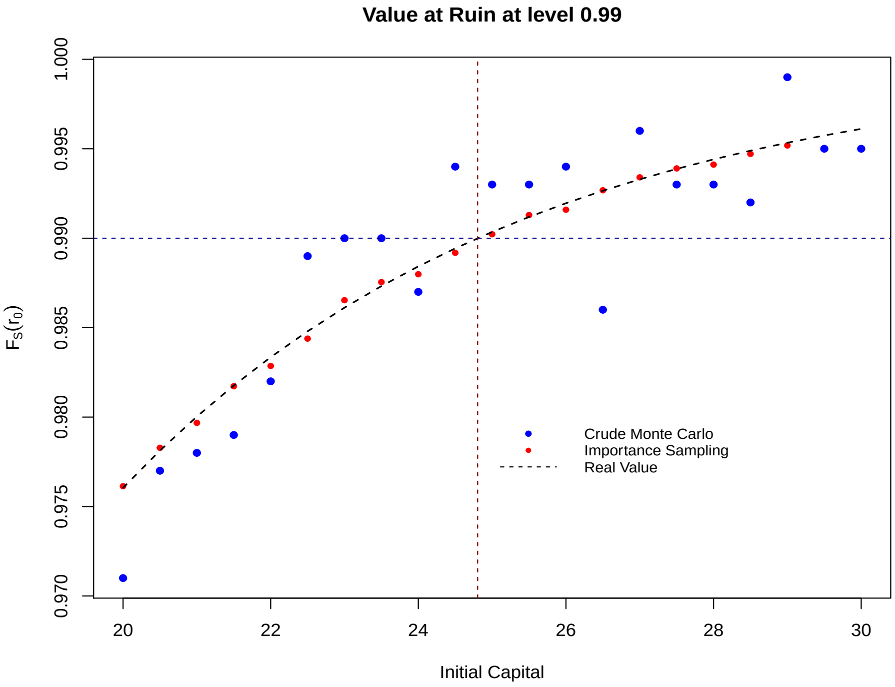


Figure 4.4: VaRu Estimation

Starting from the simulation of the probability of ruin for several initial capitals, we look the closer point to the intersection intersection to the horizontal line, we can generate different grids. in the previous plots the grid was performed considering jumps of 0.5. After the first step there is evidence that the intersection is between $[23, 26]$. Then we proceed computing new values of probability of ruin in such interval. We repeat this process until a certain precision is reached. The codes used for this estimation are the next:

```
####Monte Carlo Estimation
probruins<-NULL
# We define a sequence of initial capitals
init.cap<-seq(1,30,length.out = 59)
#We perform the Monte Carlo Estimation
for(i in c(1:length(init.cap))){
a<-prob.ruin(lambda=2,theta=0.1,claimrate=2,
             u0=init.cap[i],T=1000,runs = 1000)[[2]]
probruins[i]<-mean(a)
}


#Plot of the graphic with black points
plot(approx(init.cap,1-probruins)$x
     ,approx(init.cap,1-probruins)$y,pch=20,cex=0.9)


##Importance Sampling Estimation

pruinis<-NULL
init.cap<-seq(1,30,length.out = 59)
for(i in c(1:length(init.cap))){
a<-prob.ruin(lambda=lambda1,theta=0.1,claimrate=2,u0=init.cap[i],
             T=1000,runs = 1000)
index<-which(a[[2]]=="TRUE")
length(index)
lista<-a[[1]][index]
deficits<-unlist(lapply(lista, FUN=function(x)tail(x[,2],n=1)))
pruinis[i]<-(exp(-r*init.cap[i])*sum(exp(r*deficits))*(1/length(index)))
}
```

4.4 Estimation of Value at Ruin considering Weibull claim amounts with Weibull index equals to 1

After we checked that the algorithm works pretty well in the Erlang case, we perform the same steps considering Weibull claim amounts with $\tau = 1$, that corresponds to the Erlang model. We perform these computations in order to check if our programming scheme is well defined. If it is done, since the R functions used are parametrized, we can generalize for the case $\tau > 1$.

```
prob.ruin<-function(lambda,scale,theta,shape,T,u0,runs){
  mu<-scale*gamma(1+1/shape)
  c <- (1 + theta ) * lambda*mu ;
  result <- c (); # Result
  data<-list()
  for (r in 1: runs )
  {
    time <- 0 # Keep track of the time
    level <- u0 # Keep track of the level
    interarrivaltime <- rexp (1, lambda )
    table<-NULL
    i<-1
    while ( time + interarrivaltime < T)
    {

      # A new claim arrives
      time <- time + interarrivaltime
      claim <- rweibull (1, shape = 1/shape,scale = scale)

      # Update level
      level <- level + c* interarrivaltime - claim
      table<-rbind(table,c(time,level,interarrivaltime,claim))
      if( level < 0)
      {
        break ;
      }

      interarrivaltime <- rexp (1, lambda );

      i<-i+1
    }
    # names(table)<-c("time","level","intarriv","claim")
    data[[r]]<-table
    result <- c( result , level < 0);
  }
  return(list(data,result))
}
#The shape parameter correspond to the Weibull index
mean(prob.ruin(lambda=2,theta=0.1,shape=1,
               scale=0.5,u0=3,T=1000,runs=200)[[2]])
```

```
[1] 0.515
```

As well as in the first example with exponential claim amounts, we can consider a vector of initial capitals and construct the probability of ruin for such capitals.

```
prob.ruin<-function(lambda,scale,theta,shape,T,u0,runs){
  mu<-scale*gamma(1+1/shape)
  c <- (1 + theta ) * lambda*mu ;
  result <- c (); # Result
  data<-list()
  for (r in 1: runs )
  {
    time <- 0 # Keep track of the time
    level <- u0 # Keep track of the level
    interarrivaltime <- rexp (1, lambda )
    table<-NULL
    i<-1
    while ( time + interarrivaltime < T)
    {

      # A new claim arrives
      time <- time + interarrivaltime
      claim <- rweibull (1, shape = 1/shape,scale = scale)

      # Update level
      level <- level + c* interarrivaltime - claim
      table<-rbind(table,c(time,level,interarrivaltime,claim))
      if( level < 0)
      {
        break ;
      }

      interarrivaltime <- rexp (1, lambda );

      i<-i+1
    }
    # names(table)<-c("time","level","intarriv","claim")
    data[[r]]<-table
    result <- c( result , level < 0);
  }
  return(list(data,result))
}
#Example
mean(prob.ruin(lambda=2,theta=0.1,shape=1,scale=2,
               u0=1,T=50,runs=10)[[2]])
```

```
[1] 0.8
```

The moment generating function for the Weibull distribution that is given by the next expression

$$\sum_{n=0}^{\infty}\frac{t^n\lambda^n}{n!}\gamma(1+\frac{n}{k}) \tag{4.1}$$

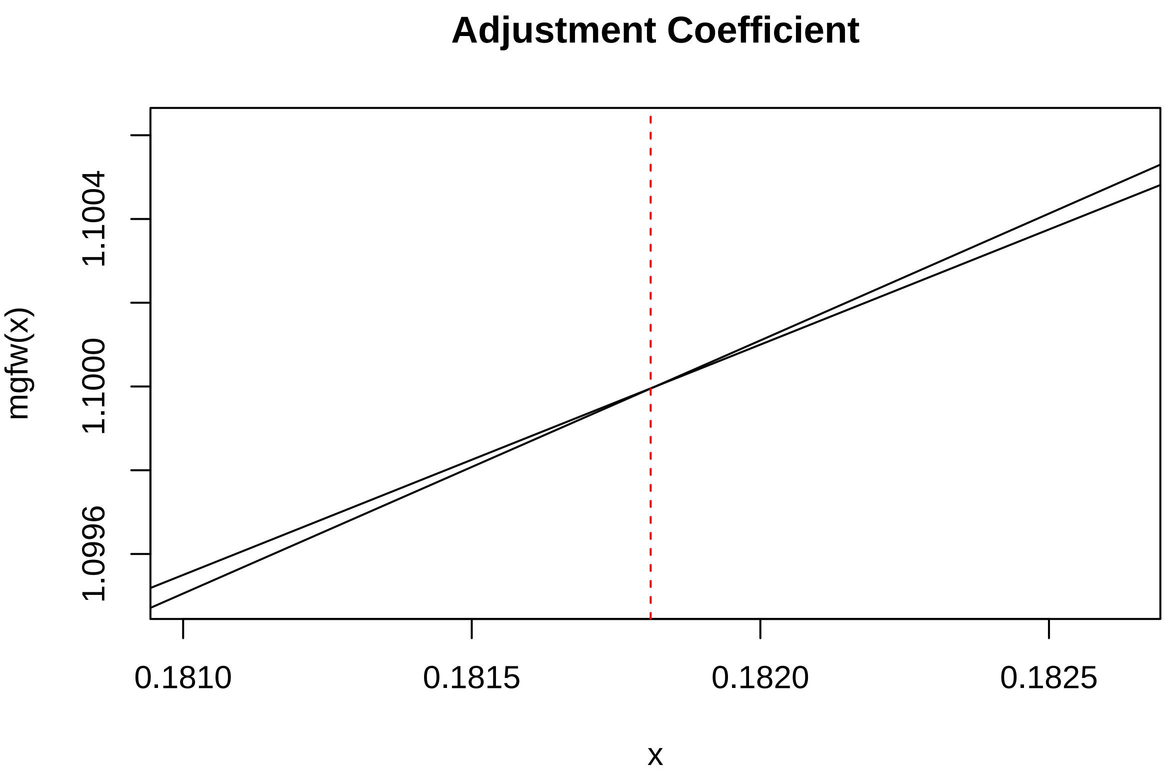
We constructed a code that evaluates the moment generating function of the Weibull distribution, since it has an infinite sum, we consider a upper limit big enough, quantities much away from this limit are almost zero.

```
##Define the moment generating function Weibull claim amounts
scale<-1/2
shape<-1
lambda<-2
betha<-0.1

mgfw <- function(x){
  nmax <- 150
  scale <- scale
  shape <- shape
  suma <- 0
  for(n in 0:nmax){
    suma <- suma + ((x^n)*((scale)^n))*gamma(1+(n/shape))/factorial(n)
  }
  return(suma)
}


mu<-(scale)*gamma(1+(1/shape))
c <- (1 + betha ) * lambda*mu

fun2<-function(x)1+x*(1+betha)*mu
x<-seq(0,10,length.out = 100)
y<-fun2(x)
###Plot that shows the intersection
```

```
curve(mgfw, 0.1, 0.3, ylim = c(1.1 - 0.00051, 1.1 + 0.00062),
      xlim = c(r - 0.00081, r + 0.00081),main="Adjustment Coefficient")
curve(fun2,from = 0,10,add = TRUE)

####Computation of the adjutment coefficient using other packages

library(rootSolve)
r<-uniroot.all(function(x) mgfw(x)-fun2(x),c(0.10,0.47))
r
```

```
[1] 0.18181
```

```
abline(v=r,col="red",lty=2)
```



```
adjCoef(mgf.claim = mgfw,premium.rate = c,
        upper.bound = 1,mgf.wait =1/(1-0.5*x))
```

```
[1] 0.1818182
```

```
# After comparing we got the same results.
```

With the adjustment coefficient we can compute the value of λ_θ that is the tilted parameter. We present the codes of the Monte Carlo estimation with the Importance sampling estimation as well as the estimation using the analytical formula for the erlang model (exponential claim amounts).

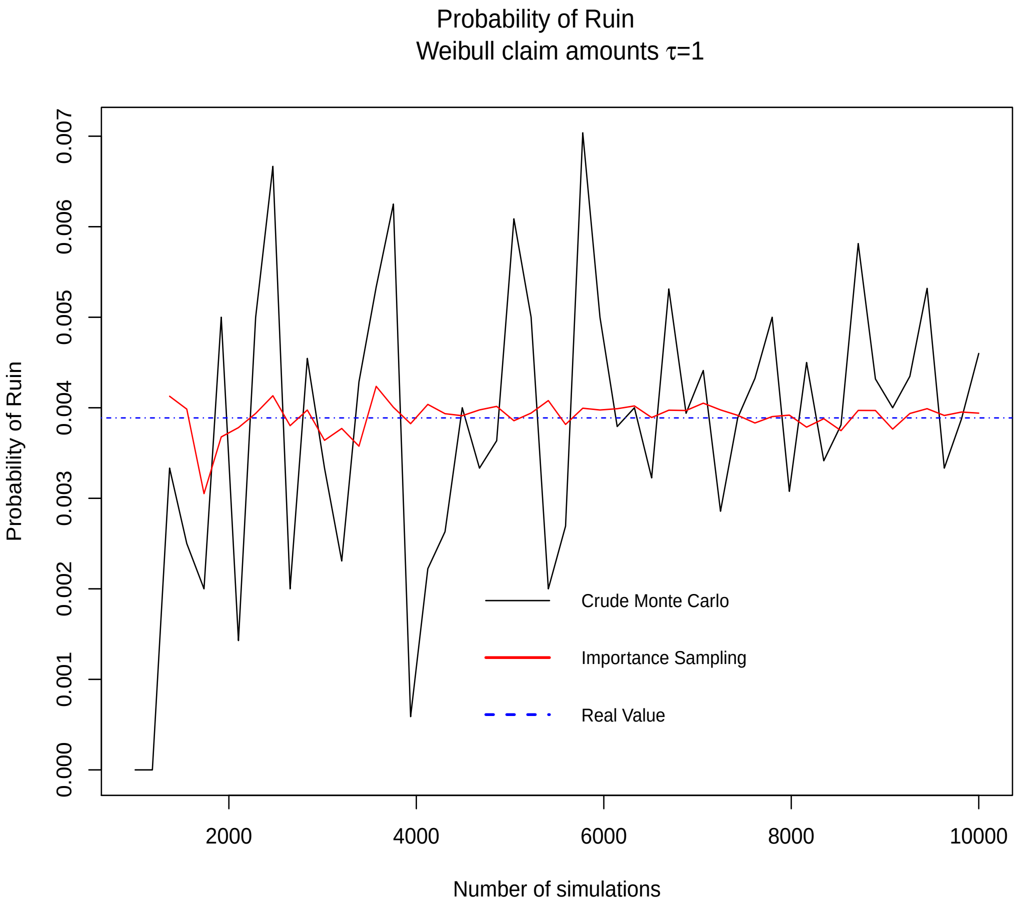


Figure 4.5: Adjustment Coefficient

The resulting estimator of the value at ruin is given by closer point to the horizontal line in 0.99. In the next plot we can see that the importance sampling estimator is more accurate than the crude Monte Carlo estimator of the probability of ruin. In this case also to fund the exact quantile is easier in the importance sampling case.

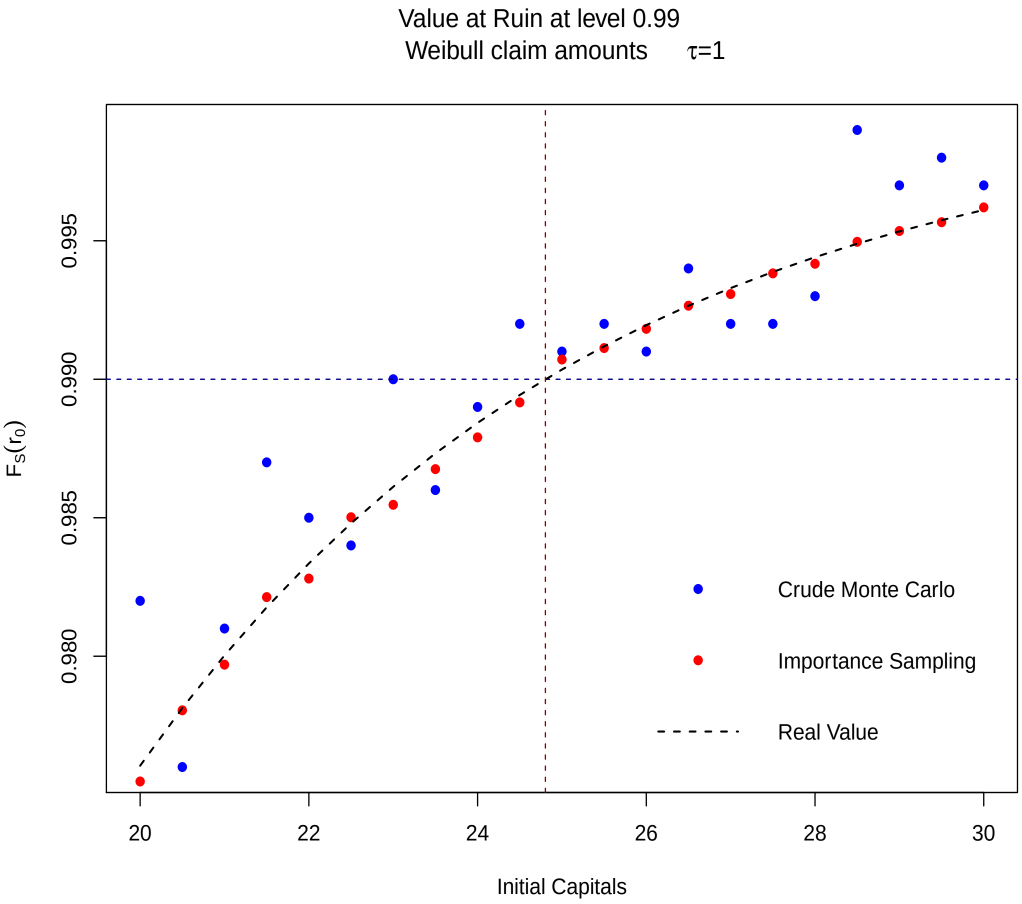


Figure 4.6: Importance Sampling Method vs Crude Monte Carlo for the computation of the Probability of Ruin for different number of simulations (Weibull with index equals to one)

4.5 Value at Ruin considering the risk process with light tailed claim amounts.

By definition of light tailed distribution, we know that the moment generating function is finite. Therefore, we can apply the importance sampling method.

We have performed the simulation for the parameters: $\beta = 0.1$ scale $\alpha = 0.5$, shape $\tau = 1.5$, initial capital $u_0 = 20$,. Time horizon $T = 1000$, using a number of simulations from 1000 to 10000 with jumps of 250. The result also shows an important variance reduction. We have included the blue line as the sample mean of the Importance sampling estimation.

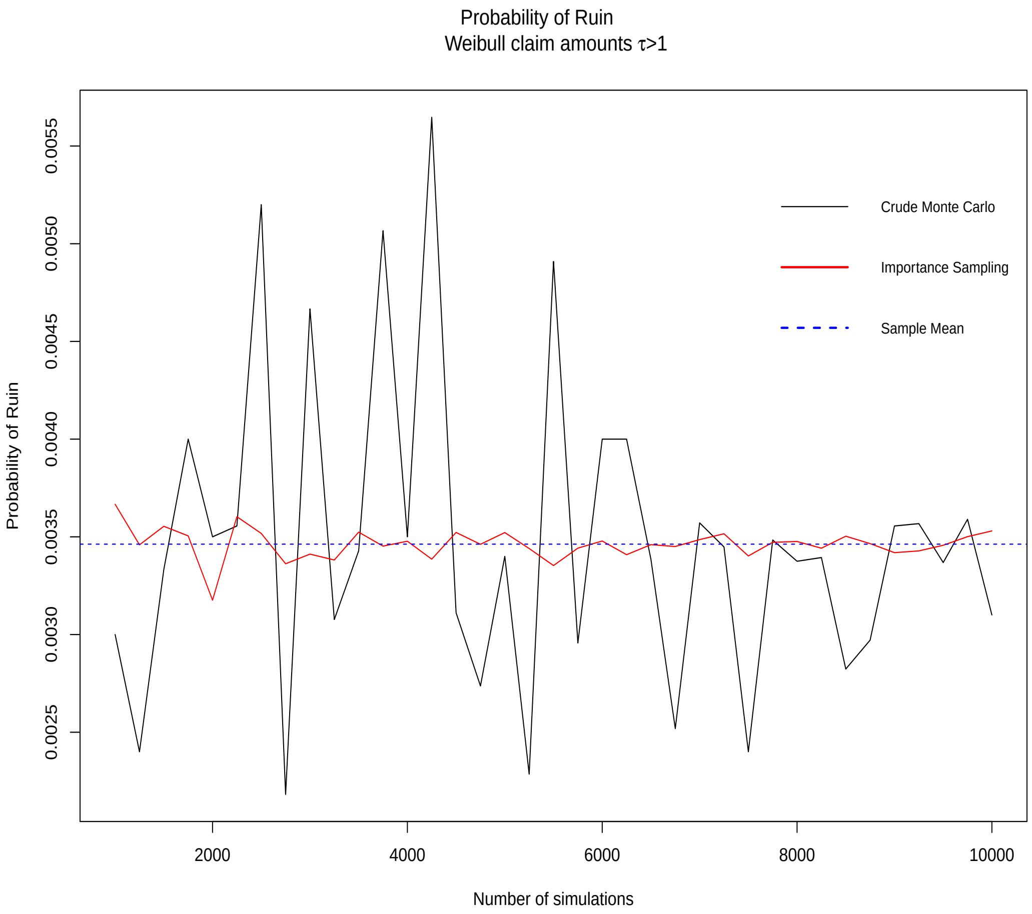


Figure 4.7: Importance Sampling Method vs Crude Monte Carlo for the computation of the Probability of Ruin for different number of simulations (light tailed claim amounts)

Now, we present an example where we compute the VaRu at level 0.99 for a Cramer Lundberg Risk Process with Weibull Claim amounts, we have started our simulation scheme with several initial capitals between 1 and 30

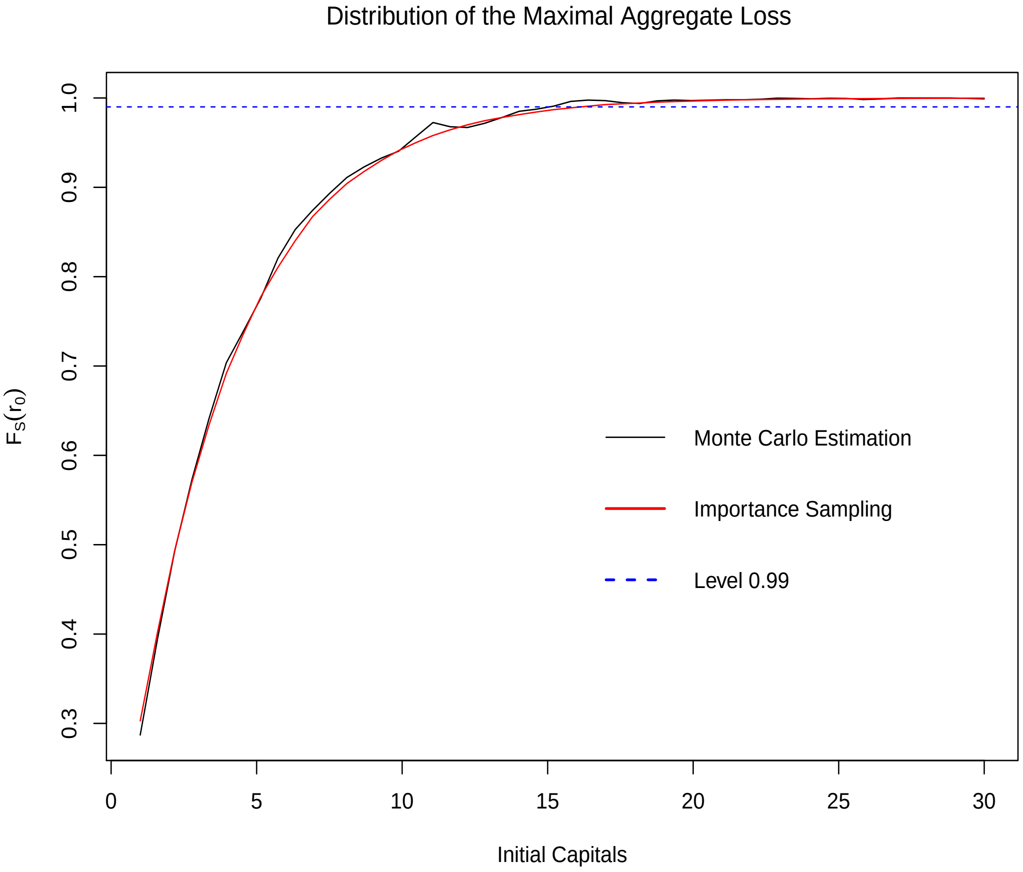


Figure 4.8: Distribution of the maximal agreggated loss using Monte Carlo method and Importance Sampling

The previous plot show us that the intersection between the blue horizontal segmented line and the red line obtained by using the Importance Sampling method is intersected around 15 and 20. Now we expand the plot generating a simulation in that interval to see the behavior of both approaches. Moreover, due to we can not compare as before with the Erlang model, we have performed a polynomial regression of order two that gave us a good fitting of the distribution of the maximal aggregated loss S. After that, we have found the intersection of the importance sampling estimator of the maximal aggregated loss and the horizontal line corresponding to the level 0.99.

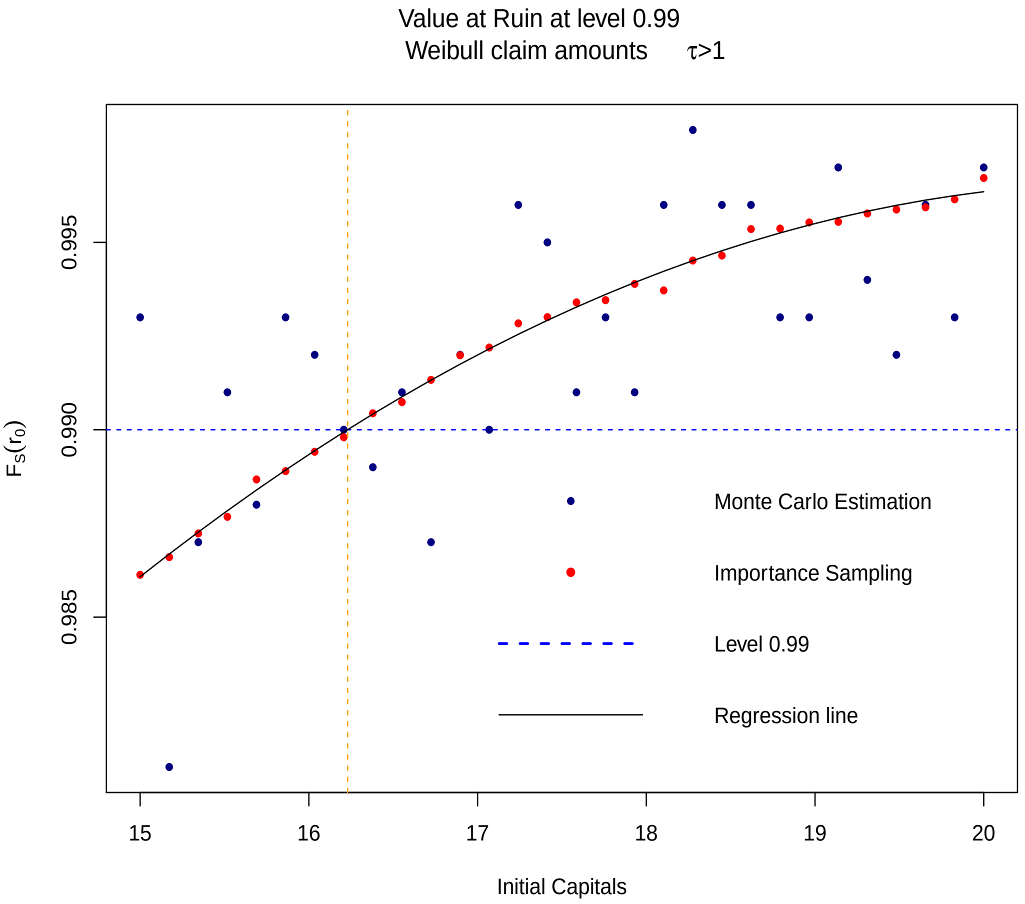


Figure 4.9: Value at Ruin Estimated

In this case the VaRu obtained is equals to 16.23.

4.5.1 Estimation of TVaRu

For the Tail Value at Ruin we can use the equation 3.8. We have simulated using the importance sampling method from the VaRu found 16.23 until 40. Then we have performed a polynomial regression for the tail that we are interested. The advantage of the importance sampling method is that is easier to draw a line through the points obtained in the simulation. On the other hand with the points obtained in the Monte Carlo method, the linea obtained is not smooth. Also , we have computed the value where the estimated function is equals to one, due to we approximate a distribution function we truncate our computations until that value.

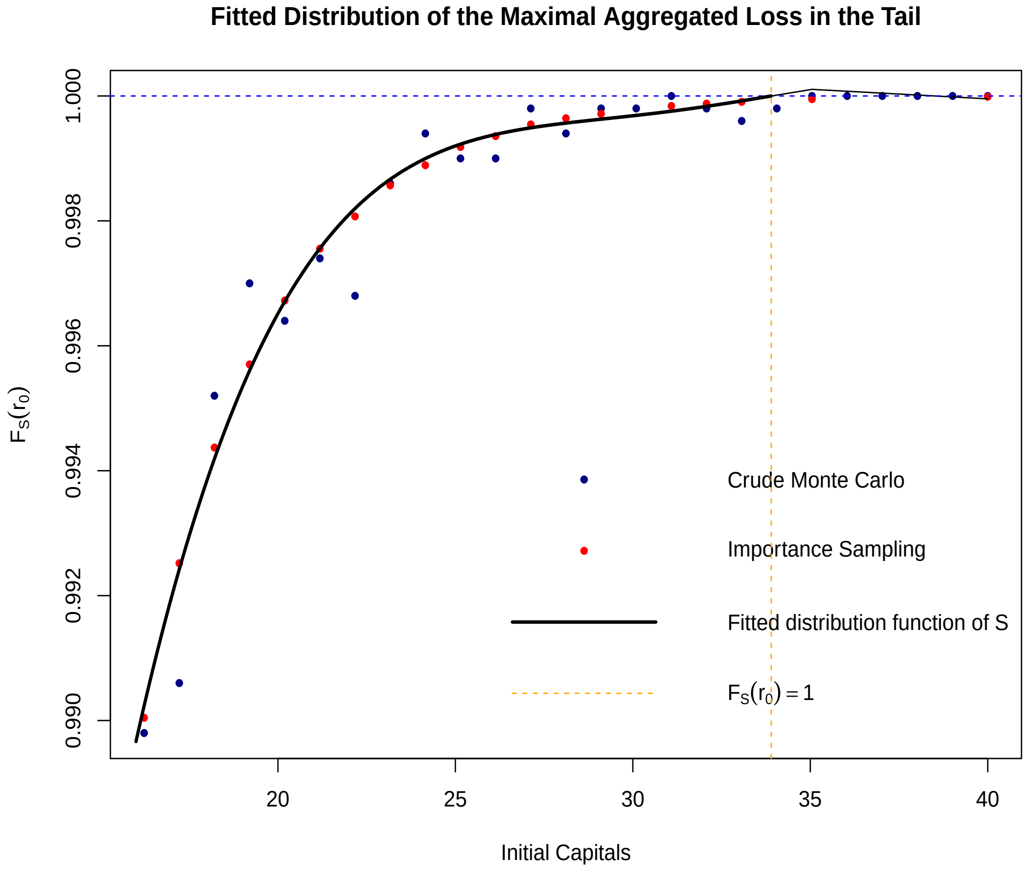


Figure 4.10: Fitted distribution of the maximal agreggated loss in the tail

The results of the fitted distributions of the Maximal Aggregated Loss in the Tail are presented in the next tables. The godness of fit given by the regression is almost

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	8.2872e-01	8.1536e-03	1.02e+02	1.7431e-21
poly(x, 4, raw = TRUE)1	2.1812e-02	1.2684e-03	1.72e+01	8.2356e-11
poly(x, 4, raw = TRUE)2	-1.0447e-03	7.1873e-05	-1.45e+01	7.7168e-10
poly(x, 4, raw = TRUE)3	2.2194e-05	1.7598e-06	1.26e+01	4.9306e-09
poly(x, 4, raw = TRUE)4	-1.7587e-07	1.5727e-08	-1.12e+01	2.3002e-08

Table 4.1: Polynomial Regression in the tail of importance sampling estimation

perfect, it can be concluded by the high value of $R^2 = 0.999$. Therefore, the estimated equation for the distribution of the maximal agreggated loss is:

$$\hat{F_S}(x) = 8.287174e{-}01{+}2.181237e{-}02x{-}1.044650e{-}03x^2{+}2.219378e{-}05x^3{-}1.758671e{-}07x^4$$

Using the relation $\hat{\psi}(r_0) = 1 - \hat{F}_S(r_0)$ we get an expression for the probability of ruin in the tail. From here we can use the equation 3.8. In order to apply the formula we compute first the integral $\int_{16.23}^{33.9} \hat{F}_S(x)dx = 6.01763$ and finally the $TVaRu(0.99) = \frac{1}{1-0.01}6.01763 + 16.23 = 22.3084$

4.6 Probability of Ruin of the Cramer Lundberg Risk Process with heavy tailed claim amounts.

For the heavy tailed claims i.e considering a Weibull distribution with $0 < \tau < 1$.we have used the Crude Monte Carlo algorithms considering conditional simulation, the Asmussen-Kroese estimator and the Asmussen-Kroese Estimator considering control variates.

These algorithms use simulation bby the Pollackzek-Khinchine formula. We know that for this purpose we need to simulate the ladder heights that follows the integrated tail distribution of Weibull.

The next function generates a vector of size n from the integrated tail distribution by the acceptance rejection method presented in chapter 3.

```
#Algorithm 7
# shape: is the Weibull index,
# scale: is the scale parameter

geo <- function(shape, scale, n){
  i <- 0
  #scale is the scale parameter of the Weibull distribution
  x <- numeric(n)
  k <- shape^(-1/(shape-1))-shape^(-shape/(shape-1))
  while(i < n){
    i <- i+1
    y <- -(scale)*log(1-runif(1))
    if(runif(1) < exp((y/scale)-(y/scale)^shape - k)) {x[i] <- y}
  }
  if(n==0){x <- 0}
  return(x)
}

###Use:
##The next line of code generates 10 random numbers from the
##Equilibrium distribution of Weibul
geo(0.5,2,10)
```

```
[1] 0.4046394 6.3806692 0.3632065 2.3112093 1.0261343 4.8584540 1.2651290
[8] 2.7188567 5.9253652 0.6684070
```

4.6.1 Probability of Ruin via the Pollackzek-Khinchine formula using the crude Monte Carlo algorithm

The next code computes the probability of ruin by the Pollaczek Khinichine formula using Monte Carlo simulation. We have included in the output the confidence intervals, the relative error, the empirical variance and the variable S_{k_i} that is the sum of the ladder heights.

```
## Monte Carlo using Pollakzeck Khinchine formula
## The output is the estimated valueof the probability of ruin
## Method I--- Monte Carlo

sim1 <- function(theta, u0, nsim,shape,scale){
  Sk<-NULL
  Sk1<-NULL
  rho <- 1/(1+theta)
  z <- numeric(nsim)
  for(i in 1:nsim){
    k <- rgeom(1, prob=rho)
    Sk<-sum(geo(scale=scale,shape=shape,n=k))
    Sk1<-c(Sk1,Sk)
    if(Sk> u0){
      z[i] <- 1
    } else {
      z[i] <- 0
    }
  }
  res <- list(Sk1=Sk1,media=mean(z),
            liminf=mean(z) - 1.96*sqrt(var(z))/length(z),
            limsup=mean(z) + 1.96*sqrt(var(z))/length(z),var=var(z),
            rel.error=(var(z))/(mean(z)^2),
            log.eff=abs(log(var(z)))/abs(log(mean(z)^2)))
  return(res)
}
#### Example of usage
sim1(theta=0.3, u0=10, nsim=10,shape=0.4,scale=15)
```

```
$Sk1
 [1] 0.000000 0.000000 0.000000 0.000000 7.214143 7.673459 0.000000
 [8] 0.000000 0.000000 0.000000

$media
[1] 0

$liminf
```

```
[1] 0

$limsup
[1] 0

$var
[1] 0

$rel.error
[1] NaN

$log.eff
[1] NaN
```

The algorithms 4 and 5 that are based on conditional simulation require two aditional functions respectively. One of them extracts the highest claim amount after sorting and the second one extracts the last claim amount.

4.6.2 Conditional Simulation of the probability of Ruin

The conditional simulation algorithms have been implementated by the next functions:

```
# Conditional simulation algorithm
sim2 <- function(theta, scale, shape, u0, nsim){
  rho <- 1/(1+theta)
  muB <- scale* gamma(1+1/shape)
  z <- numeric(nsim)
  for(i in 1:nsim){
    k <- rgeom(1, prob=rho)
    if(k<=1){
      y <- u0
      Bc <- (1/(muB*shape))*(incgam(y^shape, 1/shape))
      z[i] <- 0
    } else {
      gen <- geo(scale=scale,shape=shape,n=k)
      y <- u0 - sum(mmu2(gen))
      if(y<0){
        z[i] <-1
      } else {
        Bc <- (1/(muB*shape))*(incgam(y^shape, 1/shape))
        z[i] <- Bc
      }}
  }

  return(data.frame(media=mean(z),
                    liminf=mean(z) - 1.96*sqrt(var(z))/length(z),
                    limsup=mean(z) + 1.96*sqrt(var(z))/length(z),var=var(z),
                    rel.error=(var(z))/(mean(z)^2),
                    log.eff=abs(log(var(z)))/abs(log(mean(z)^2))))
}
#### Example of usage
sim2(theta=0.3, u0=10, nsim=10,shape=0.4,scale=15)
```

```
  media liminf limsup var rel.error log.eff
1     0      0      0   0       NaN     NaN
```

4.6.3 Asmussen-Kroese Estimator

```
## Asmussen-Kroese
sim4 <- function(theta, scale, shape, u0, nsim){
  rho <- theta/(1+theta)
  muB <- (scale) * gamma(1+1/shape)
  z <- numeric(nsim)
  for(i in 1:nsim){
    k <- rgeom(1, prob=rho)

    if(k==0){
      z[i] <- 0
    } else if(k==1){
      y <- u0
      m <- 0
      Bc <- (1/(muB*shape))*incgam(x=y^shape, a=1/shape)
      z[i] <-0*((1/(muB*shape))*incgam(x=max(y,m)^shape, a=1/shape))

    } else {
      gen <- geo(scale=scale,shape=shape,n=k-1)
      y <- u0-sum(gen)
      m <- max(gen)
      Bc <- k*((1/(muB*shape))*incgam(x=max(y,m)^shape, a=1/shape))


      z[i] <- Bc

    }
  }
  res <- data.frame(media=mean(z),
                    liminf=mean(z) - 1.96*sqrt(var(z))/length(z),
                    limsup=mean(z) + 1.96*sqrt(var(z))/length(z),
                    variance=var(z),
                    rel.error=(var(z))/((mean(z)^(2))),
                    log.eff=abs(log(var(z)))/abs(log(mean(z)^2)))
  return(res)
}

## Asmussen Kroesse and Control Variates

sim5<- function(theta, scale, shape, u0, nsim){
  rho <- theta/(1+theta)
  muB <- (scale) * gamma(1+1/shape)
  z <- numeric(nsim)
  for(i in 1:nsim){
    k <- rgeom(1, prob=rho)

    if(k==0){
      z[i] <- 0
    } else if(k==1){
      y <- u0
      m <- 0
    Bc <- (1/(muB*shape))*incgam(x=y^shape, a=1/shape)
    z[i]<-((1/(muB*shape))*(incgam(x=max(y,m)^shape, a=1/shape)-
                incgam(x=u0^shape, a=1/shape)))+(((1-rho)/rho)*(1/(muB*shape))*(incgam(x

    } else {
      gen <- geo(scale=scale,shape=shape,n=k-1)
      y <- u0-sum(gen)
      m <- max(gen)
      Bc <- k*((1/(muB*shape))*(incgam(x=max(y,m)^shape, a=1/shape)-incgam(x=u0^shape, a
      #sum(mmu(z4)[[1]])

      z[i] <- Bc

    }
  }
  res <- data.frame(media=mean(z),
                    liminf=mean(z) - 1.96*sqrt(var(z))/length(z),
                    limsup=mean(z) + 1.96*sqrt(var(z))/length(z),
                    variance=var(z),
                    rel.error=(var(z))/(mean(z)^2),
                    log.eff=abs(log(var(z)))/abs(log(mean(z)^2)))
  return(res)
}
```

We see the performance of the functions implemented with the next example:

4.6.4 Numerical Example considering heavy tailed claim amounts

To estimate the probability of ruin, we consider Weibull distributed claim amounts with shape parameter $\tau = 0.35$, a scale parameter equals to 1, a security loading $\beta = 0.1$ and a vector of initial capitals from $u_0 = 100, ..., 5000$ with jumps of 100. In this case we have gotten that the Crude Monte Carlo method fails, when the initial capital takes large values. We have performed 5000 simumlations.

For the methods after applying some of the variance reduction techniques, we present the plots of the one of efficiency measures presented before. In this case we analyze the logarithmically efficiency of the methods, where we can verify that the methods in general have a good performance.

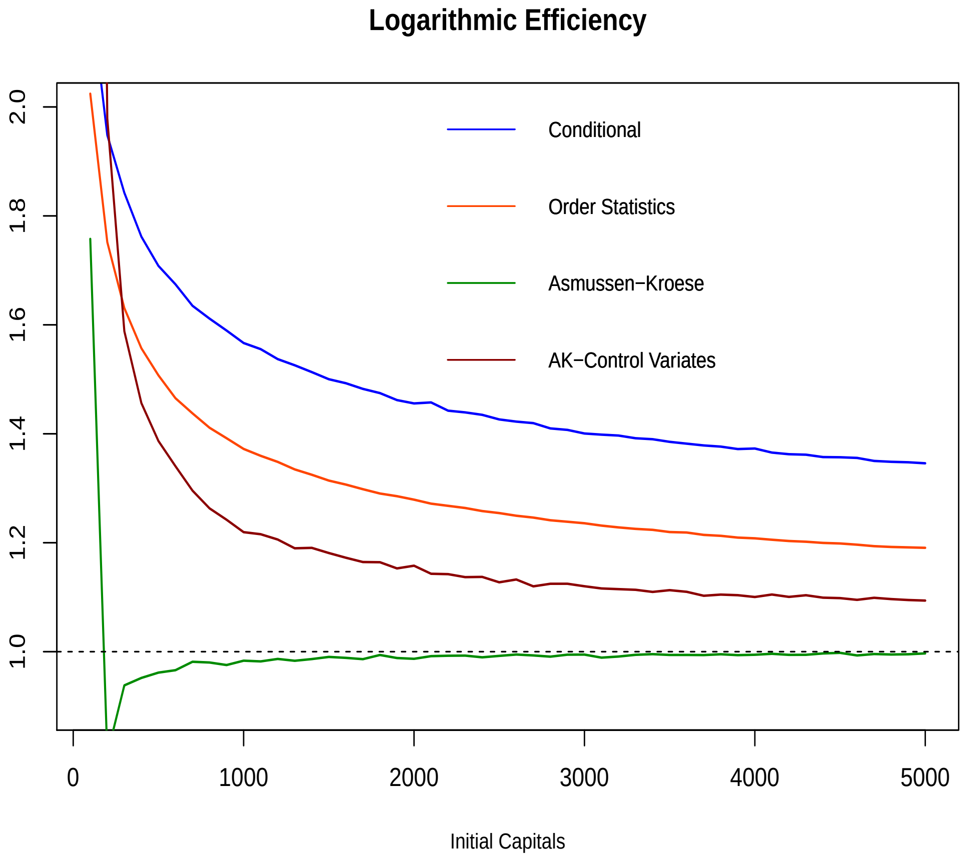


Figure 4.11: Logarithmic Efficiency of the simulation methods considering heavy tailed claim amounts

From the previous plot we can conclude that the better performance is reached by the conditional algorithms as well as the Asmusse-Kroese estimator with control variables. These algorithms as the definition given before states, when the initial capital goes to infinity, the performance criteria is bigger than one.

Also, we include a plot with the relative errors to check that when the initial capital is large, the relative error is bounded.

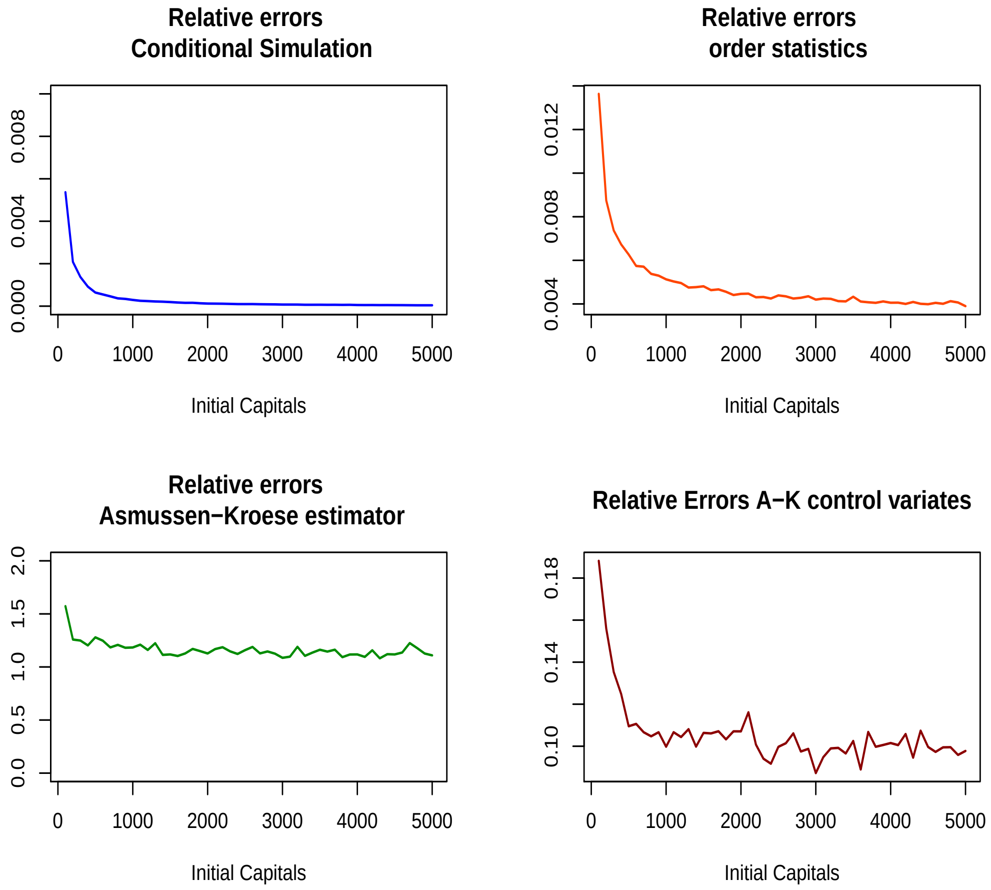


Figure 4.12: Relative errors of the simulation methods considering heavy tailed claim amounts

From the previous plot, we can see that the methods work very well in the sense of relative errors, we can see that when the initial capital goes to infinity, the relative errors are bounded.

Now we have estimated the Value at Ruin using the methods presented. We have fitted a regression line on the interval where the value at ruin is located approximately, then with the regression line we have found the intersection between the estimated regression line and the horizontal corresponding to the level of the quantile desired.

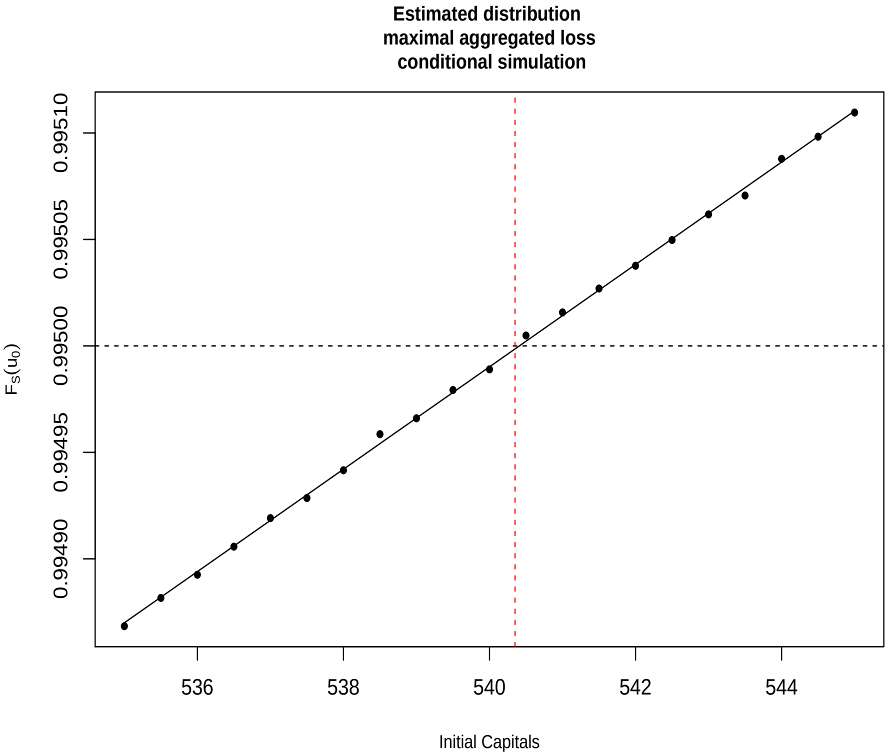


Figure 4.13: Value at Ruin estimation considering heay tailed Weibull claim amounts, using conditional simulation

$$VaRu(0.995)_1 = 540.35$$

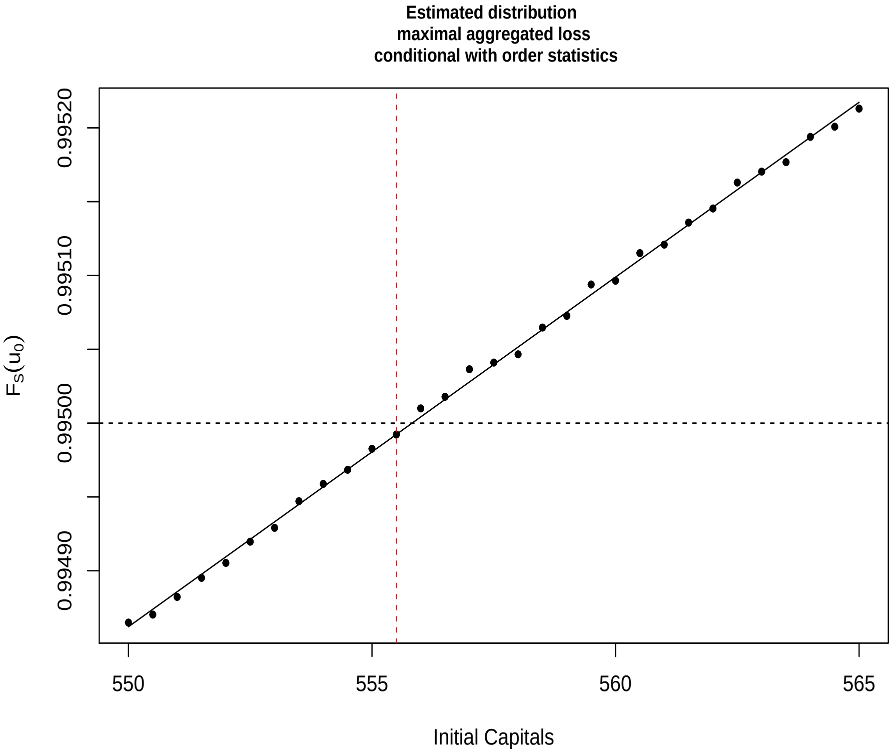


Figure 4.14: Value at Ruin estimation considering heay tailed Weibull claim amounts, using conditional simulation and order statistics

$VaRu(0.995)_2 = 555.5$

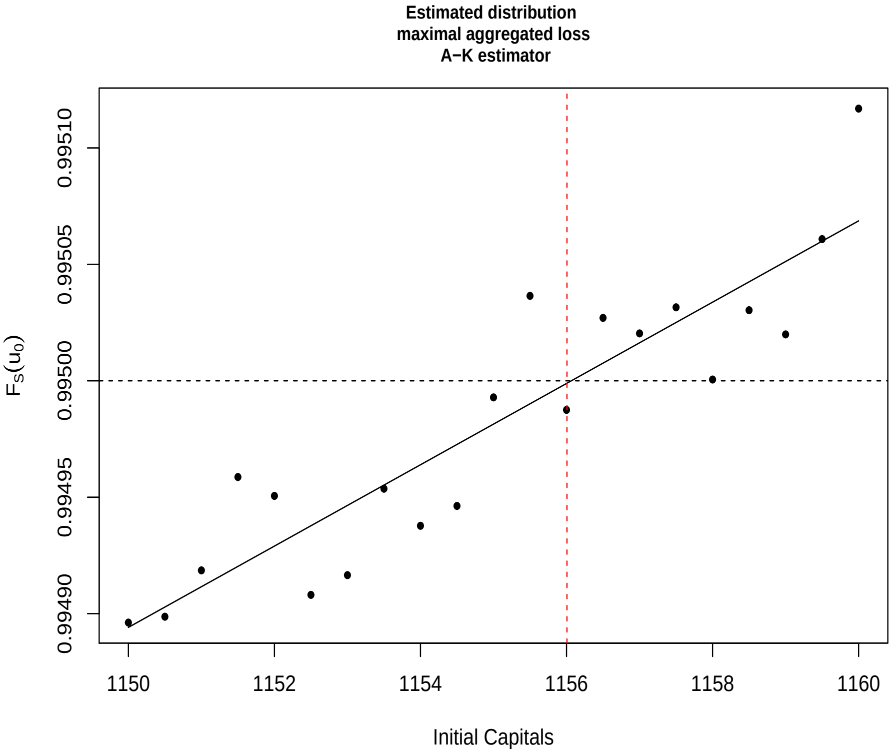


Figure 4.15: Value at Ruin estimation considering heay tailed Weibull claim amounts, using the Asmussen-Kroese estimator

$$VaRu(0.995)_1 = 1156.06$$

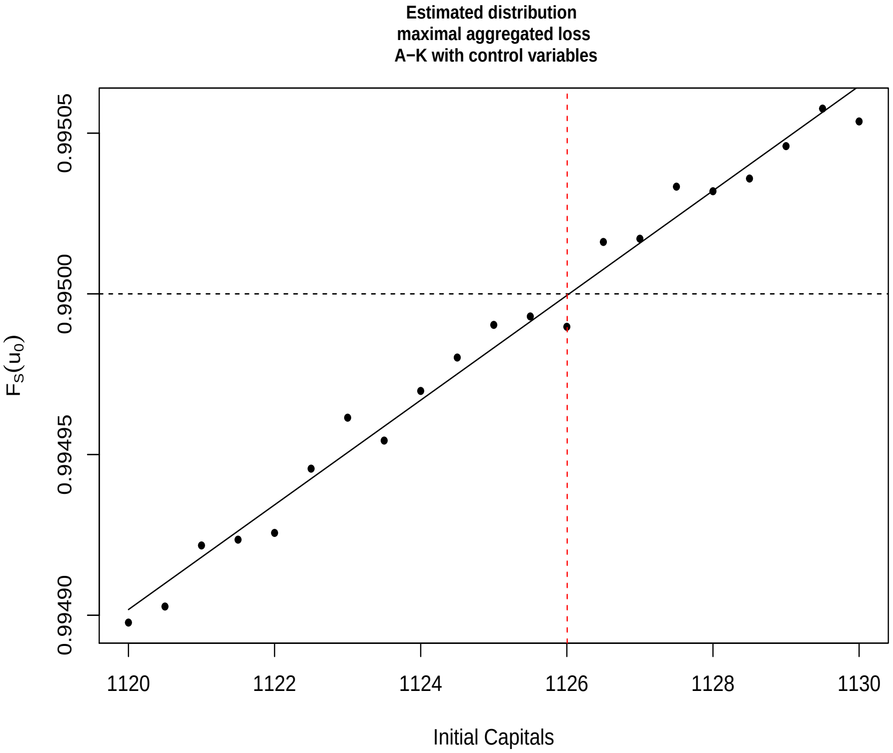


Figure 4.16: Value at Ruin estimation considering heay tailed Weibull claim amounts, using Asmussen Kroese estimator and control variates

$VaRu(0.995)_1 = 1126.005$

From the these results, we can conclude that the conditional simulation methods, could be underestimating the probability of ruin. Moreover, during the development of numerical procedures, we have seen that these methods give similar values of probability of ruin when the initial capital is significantly large.

4.7 Another approach for the Value at Ruin estimation

Another way to approximate the quantile after simulating n replicates of S, is by

$$\hat{q}_{1-\epsilon} = inf\{x : \hat{F}_S(x) \geq 1-\epsilon\}$$

From the simulated values for the probability of ruin, we can construct the empirical c.d.f $\hat{F}_S(x)$. Assuming that the density function f_S exists and applying the central limit theorem for quantiles,we can get a 95% confidence interval for the true value of $q_{1-\epsilon}$ by:

$$\left[Z_{(1-\epsilon)(R+1)} - 1.96\frac{\sqrt{\epsilon(1-\epsilon)}}{f_S(\hat{q}_{1-\epsilon})\sqrt{n}}, Z_{(1-\epsilon)(R+1)} + 1.96\frac{\sqrt{\epsilon(1-\epsilon)}}{f_S(\hat{q}_{1-\epsilon})\sqrt{n}} \right]$$

where $Z_{(1-\epsilon)(n+1)}$ is the empirical quantile (the $(1-\epsilon)(n+1)$ order statistic). Here, since f_S is not available, we can estimate it by a kernel density estimator for example.

Conclusions

The achievement of this thesis work had required the understanding of risk theory concepts and stochastic simulation. We have studied one of the most important concepts in risk theory that is the probability of ruin considering the Cramer Lundberg Risk process with Weibull claim amounts. The probability of ruin is crucial in the estimation of two risk measures, the Value at Ruin and the Tail Value at Ruin. Therefore, the problem of its estimation has been more about the estimation of the probability of ruin. We have presented variance reduction techniques as improvements to the Crude Monte Carlo method.

The conclusions are the next:

- For the light tailed case, the performance of the importance sampling method is evidently much better than the crude Monte Carlo method. The use of this method allow us to get estimators of the probability of ruin with smaller variance. This is helpful in the computation of the VaRu as well as the TVaRu, since with the importance sampling simulation is easier to fit a polynomial regression. In the example presented, the fit is almost perfect, with a statistic $R^2 \approx 1$.

- The simulation of the path of the risk process takes more time in contrast to the heavy tailed case where we used the Pollackzek Khinchine formula. However, it is a good and useful way to understand how the risk process works and also due to the importance sampling algorithm we require some quantities that can be computed only with the simulation of the risk process.

- For the heavy tailed case, we have presented three approaches more than the crude Monte Carlo method. One related to the control variates estimation, where we used results of subexponential distribution and the simulation of variables from the integrated tail distribution of Weibull by the acceptance rejection method. This simulation method behaves more accurately than the crude Monte Carlo, since as in the light tailed case the obtained points can be joined by a regression line that also was fitted almost perfectly with a value of $R^2 \approx 1$.

- In the heavy tailed case, we have also proposed two conditional simulation algorithms. The behaviour of both is very similar and present the same advantage of the previous simulation methodologies. Since the variability is smaller than the crude Monte Carlo, the simulated points for every initial capital can be joined by a regression line which is used to compute the TVaRu with the formula 3.8.

- The probability of ruin and the risk measures associated to this quantity, are sensible to high values of initial capital, specially when the probabilities are very low. For that reason the usage of variance reduction techniques is a good choice to solve the problem.

- As we can see in the heavy tailed case, the Monte Carlo method without variance reduction techniques, does not works properly. In our computations most of the times with an adequate number of simulations the crude Monte Carlo method simply gave us a probability of ruin equals to zero.

Extensions

After the understanding of these concepts an application including a diffusion or Wiener process could be interesting.

The estimation of the value at ruin using a kernel density estimator, as we mentioned at the end of chapter 4, is a different approach about how to solve this problem.

References

Albrecher, H., Teugels, J. L., & Beirlant, J. (2017). *Reinsurance: Actuarial and statistical aspects*. John Wiley & Sons.

Asmussen, S., & Albrecher, H. (2010). *Ruin probabilities*. World Scientific Publishing Co Pte Ltd.

Asmussen, S., & Binswanger, K. (1997). Ruin probability simulation for subexponential claims. *ASTIN Bull*, *27*, 297–318.

Asmussen, S., & Glynn, P. W. (2007). *Stochastic simulation: Algorithms and analysis* (Vol. 57). Springer Science & Business Media.

Asmussen, S., & Kroese, D. P. (2006). Improved algorithms for rare event simulation with heavy tails. *Advances in Applied Probability*, *38*(2), 545–558.

Baumgartner, B., & Gatto, R. (2014). Value at ruin and tail value at ruin of the compound poisson process with diffusion and efficient computational methods. *Methodology and Computing in Applied Probability*, *16*(3), 561–582. `http://doi.org/10.1007/s11009-012-9316-5`

Gatto, R. (2014a). Stochastic simulation of rare events. *Wiley StatsRef: Statistics Reference Online*, 1–11.

Gatto, R. (2014b). *Stochastische Modelle der Aktuariellen Risikotheorie: Eine mathematische Einführung*. Springer-Verlag.

Juneja, S. (2007). Estimating tail probabilities of heavy tailed distributions with asymptotically zero relative error. *Queueing Systems*, *57*(2-3), 115–127.

Klugman, S. A., Panjer, H. H., & Willmot, G. E. (2012). *Loss models: From data to decisions* (Vol. 715). John Wiley & Sons.

Lin, X. S. (2006). Integrated tail distribution. in encyclopedia of actuarial science.

Ross, S. M. (1990). *A course in simulation*. Prentice Hall PTR.

Siegmund, D. (1976). Importance sampling in the monte carlo study of sequential tests. *The Annals of Statistics*, 673–684.