JOHANNES KEPLER
UNIVERSITY LINZ

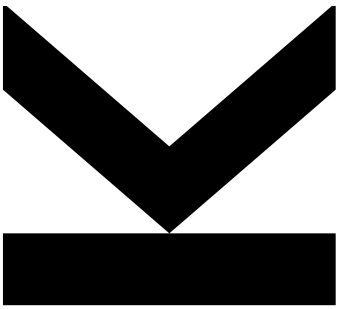
Submitted by
Hofmann Louisa
01555422

Submitted at
Institute of Financial Mathematics and Applied Number Theory

Supervisor
Assist.-Prof. Dr. habil. Sascha Desmettre

February 2022

Approximation Methods for the Total Claim Amount in Collective Risk Modeling



Master Thesis
to obtain the academic degree of
Diplom-Ingenieurin
in the Master's Program
Industrial Mathematics

JOHANNES KEPLER
UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Österreich
www.jku.at
DVR 0093696

Acknowledgments

First and foremost, I have to thank my supervisor, Assist.-Prof. Dr. habil. Sascha Desmettre of the Institute of Financial Mathematics and Applied Number Theory at Johannes-Kepler University Linz. He introduced me not only to the theory of insurance mathematics, which lies the foundation for this thesis but moreover made it possible for me to work on interesting projects at the LSQF in the field of quantitative finance. Throughout the whole time he always came up with helpful advice and solutions to all of my questions.
Additionally, I must express my very profound gratitude to my parents and to my sister for providing me with support and consistent encouragement throughout the last couple of years. The accomplishment of completing this thesis or rather my entire studies would not be possible without them.
Last but not least I would also like to thank Paul Kainberger who spent endless hours in different libraries throughout each annual exam period with me. The technical discussions and mutual motivation made a lot of stressful semesters much more pleasant.

Statutory declaration

I hereby declare under oath that the submitted Master's Thesis has been written solely by me without any third-party assistance, information other than provided sources or aids have not been used and those used have been fully documented. Sources for literal, paraphrased and cited quotes have been accurately credited.
The submitted document here present is identical to the electronically submitted text document.

Date

Louisa Hofmann

Abstract

The efficient evaluation of compound sums is an important task in actuarial science. The total claim amount an insurance company has to face within a given period of time is modeled as $S_N = X_1 + \cdots + X_N$, with N the number of occurring claims and $(X_j)_{j\in\{1,\dots N\}}$ the individual claim heights. Collective risk modeling addresses the calculation of expected value, variance and distribution of S_N in order lay the foundation for the determination of appropriate premiums, safety margins and ruin probabilities for the insurer.
An examination of the compound distribution based on the convolution is in general not of interest due to its complexity. Instead Harry H. Panjer suggested in 1981 in his paper *„Recursive Evaluation of a Family of Compound Distributions“* [1] a recursive approach for the determination of the relevant single probabilities of S_N. Compared to the convolution an improvement in terms of complexity was obtained, however a special type of distribution for the number of claims is required, which limits the range of application in practice.
A noteworthy alternative with no restrictions to any of the utilized distributions is given with an algorithm based on the Fast Fourier transform (FFT).
The paper *„Panjer Recursion versus FFT for Compound Distributions“* [2] by Paul Embrechts and Macro Frei lies the foundation for a theoretical examination, comparison and application of each of the suggested methods.

Contents

1. Introduction

1.1. Basics of non-life insurance

Insurances originated from a general demand of society asking for protection against unforeseeable events which may cause severe (financial) damage to individuals. Insurance companies provide a protection against such events and cover the arising financial expenses. The main idea behind it is the establishment of a community (*collective*), whereby each member contributes a certain amount of money (*premium*) and the emerging financial damages are covered by the funds of the community. The insurer himself is in charge of drafting the corresponding contracts, determining appropriate premiums and paying off potential claims.

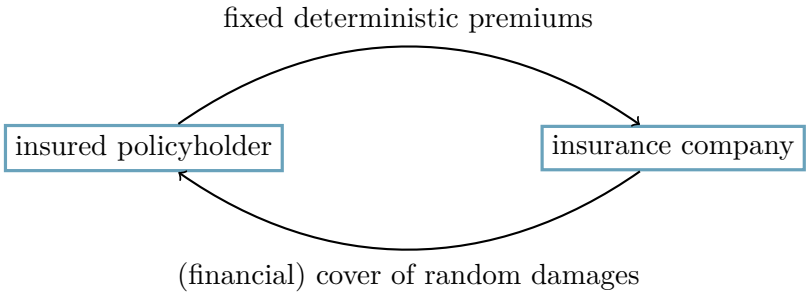


Figure 1.1.: Payment flows of an insurance contract

The choice of the community is crucial for a functioning insurance: all members have to face similar risks, what allows them to benefit from diversification in the form of the law of large numbers.
The history of insurances goes back to 1308, when the catholic church introduced the first retirement insurance. Followed by cargo- and reinsurances in Genua in the 14^{th} century. The main breakthrough for modern insurances began in 1666, when fire insurances were initiated after the disastrous Great Fire of London. Nowadays fire insurances are a part of the so-called *non-life insurances*. In Germany and Austria the following types of non-life insurances are common:

- Liability insurance
- Fire insurance
- Car insurance
- Travel insurance

- Accident insurance
- Property insurance
- Household insurance (typical for Austria)
- Legal expense insurance
- ...

The underlying period of time for such insurance contracts is typically one year. All insured events that occur within this time span and which are causing financial damage to the policyholder are indemnified. The corresponding payments are called *insurance claims*.
The insurance premium is usually payed at the beginning of the insurance period (*upfront*). In order to determine such premiums, the insurance company starts with pooling similar risks. These risks are described as a sequence of random variables $X_1, \dots X_n$, $n \in \mathbb{N}$. Since these individual claims are unknown at the beginning of the insurance period, they need to be treated in the framework of probability theory. Hence we choose an appropriated probability space $(\Omega, \mathcal{F}, P)$ and let X_i be independent and identically distributed random variables $\forall i \in \{1, \dots n\}$ with finite mean $\mathbb{E}[X_1] = \mu \leq \infty$. The Chebyshev inequality hence implies

$$\lim_{n\to\infty} P\left(\frac{|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu|}{\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right]} \geq \epsilon\right) = 0 \qquad \forall \epsilon > 0$$

and the *law of large numbers* indicates the following statement:

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow[n\to\infty]{P-\text{a.s.}} \mu.$$

This basically means that the total claim amount $\sum_{i=1}^{n} X_i$ becomes more and more „predictable“ with an increasing portfolio size n.
Since probability theory is inevitable throughout a detailed analysis of collective risk modeling, the corresponding theory is repeated in the following section **1.2**. Chapter **2** states subsequently the general model assumptions of a collective risk model and derives some properties of the total claim amount in addition to some general examples for possibly utilized distributions. A recursive approach for the determination of the total claim amount is introduced in chapter **3**, whilst chapter **4** yields the derivation of an alternative algorithm based on the Fourier transform. To conclude the thesis chapter **5** summarizes the drawn conclusions about complexity and error bounds of each of the formerly introduced concepts and points out the individual weaknesses and strengths. And since machine learning becomes more an more important nowadays chapter **6** provides a general outlook on the application possibilities of different machine learning tools in the context of non-life insurance. However one has to point out, that the corresponding theory is kept quite general, since a detailed analysis would exceed the framework of this thesis.

1.2. Probability Theory and Statistics

Subsequently a short repetition of the most important results and notations of probability theory is given, based on the book „*Versicherungsmathematik*“[3, Chapter B] by Klaus Schmidt.
We consider a non-empty set Ω and the corresponding σ-algebra $\mathcal{F}$, such that $(\Omega, \mathcal{F})$ is a measure space. If there exists a probability measure $P : \mathcal{F} \to [0,1]$ one can define the *probability space* $(\Omega, \mathcal{F}, P)$. A corresponding *random variable* with values in $(\mathbb{R}, \mathcal{B})$ is denoted by X, whereby $\mathcal{B}$ denotes the Borel-σ-algebra. The *distribution* of such a random variable X is given by

$$P^X(B) := P(X \in B) := P(X^{-1}(B)), \quad B \in \mathcal{B}.$$

The *cumulative distribution function (cdf)* of a real-valued random variable X, evaluated at some point x, is given by:

$$F^X(x) = P(X \leq x),$$

which is the probability of X being less or equal to $x \in \mathbb{R}$. $X \sim F$ denotes a random variable X, which follows the distribution $F = F^X$. A short examination of $F^X : \mathbb{R} \to [0,1]$ will yield that F is right-continuous and non-decreasing, satisfying

$$\lim_{x \to -\infty} F^X(x) = 0$$
$$\lim_{x \to +\infty} F^X(x) = 1$$

In general we have to distinguish between two types of random variables:

1. A random variable $X \sim F$ is called *discrete* if $x_0, \dots x_n$ or $x_1, x_2, \dots$ exist such that

   $$\sum_{k=0}^{n} p_k^X = 1 \quad \text{or} \quad \sum_{k \in \mathbb{N}} p_k^X = 1$$

   holds with $p_k^X = P(X = k) > 0$, the probability weight of X in $k \in \mathcal{A} \subset \mathbb{R}$. F is interpreted as a step function with countably many steps in discrete points $k \in \mathcal{A}$. In case of $\mathcal{A} \subset \mathbb{N}_0$ the random variable X, which only takes integer values, is called a *count random variable*.

2. A random variable $X \sim F$ is called *(absolutely) continuous* if

   $$F^X(x) = \int_{-\infty}^{x} f^X(y) dy$$

   holds $\forall x \in \mathbb{R}$ with a suitable measurable function $f \geq 0$ and $f = F'$. This specific function f is called *density* of X.

The following example will provide some of the most important distributions, which are going to play an important role throughout the thesis.

Example 1.2.1. Different distributions

- **Binomial distribution**: $X \sim \mathrm{Bin}(n,p)$ with $n \in \mathbb{N}$ and $p \in [0,1]$, if

$$\mathbb{P}(X=k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k \in \mathbb{N}_0$$

  is satisfied. It implies that k successes of an event occur with the probability of p^k, while $n-k$ failures occur with the probability $(1-p)^{n-k}$.
  One can directly derive the **Bernoulli distribution** $X \sim \mathrm{Ber}(p)$ for the special case $n=1$.

- **Negative binomial distribution**: $X \sim \mathrm{NB}(r,p)$ with $r \in \mathbb{N}$ and $p \in [0,1]$, if

$$\mathbb{P}(X=k) = \binom{r+k-1}{k} (1-p)^k p^r, \quad k \in \mathbb{N}_0.$$

- **Poisson distribution**: $X \sim \mathrm{Poi}(\lambda)$ with $\lambda > 0$, if

$$\mathbb{P}(X=k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad k \in \mathbb{N}_0.$$

Next up the most important moments of the random variable X have to be defined, preconditioned that they exist:

- The first moment, *mean* or *expected value*: $\mathbb{E}[X] = \int_{\mathbb{R}} x d\mathrm{F}(x)$
- *Variance* or the second moment: $\mathrm{Var}(X) := \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2$
- *Standard deviation*: $\sigma(X) = \sqrt{\mathrm{Var}(X)}$
- *Covariance* of $X \sim F$ and $Y \sim F$:

$$\mathrm{Cov}(X,Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$

- *Skewness*:

$$\gamma(X) = \frac{\mathbb{E}[(X - \mathbb{E}[X])^3]}{(\sigma(X))^3}$$

- *k-th moment*:

$$\mathbb{E}\left[X^k\right] = \int_{\mathbb{R}} x^k d\mathrm{F}(x)$$

For a more detailed description of the underlying probability theory and additional examples for various distributions see [4, Chapter 1.2] or [3, Chapter B] for example.

2. Collective Risk Modeling

2.1. Model Assumptions

The goal of the following chapter is the determination of the total claim amount S_N, which an insurance company has to face within a given period of time (usually one year). The presented theory is based on the lecture notes of Mario V. Wüthrich, see [4, Ch. 2.1]. In order to fully describe S_N, several random variables are necessary. They are defined on an appropriate probability space $(\Omega, \mathcal{F}, P)$:

- $N : \Omega \to \mathbb{N}_0$ a discrete random variable, which counts the number of occurring claims during the underlying period of time
- and a sequence of continuous and strictly positive random variables $(X_j)_{j\in\mathbb{N}}$, which are independent and identically distributed. $X_j : \Omega \to (0, \infty)$ specifies the height of each individual claim.

The total claim amount is calculated as the sum over all individual claim sizes:

$$S_N = X_1 + \cdots + X_N = \sum_{j=1}^{N} X_j.$$

Furthermore the independence of X_j and N is required for $\forall j \in \mathbb{N}$. A possible violation of this assumption is given, as soon as several risks of the same portfolio are opposed to the same danger. As an example one might think of a flooding: the number of claims and their height will rise with an increasing water level.
Under the consideration of these properties the following model assumptions can be formulated. They are sufficient to fully describe a collective risk model.

Model Assumptions 2.1.1.
Let Ω be a non-empty set with a corresponding σ-algebra $\mathcal{F}$ and the probability measure $P : \mathcal{F} \to [0, 1]$, such that $(\Omega, \mathcal{F}, P)$ defines a probability space.

$$S_N = X_1 + \cdots + X_N = \sum_{j=1}^{N} X_j \tag{2.1}$$

describes the total claim amount in collective risk modeling, assuming the following characteristics are satisfied:

1. *N is a discrete random variable with values in $\mathbb{N}_0$. $N = 0$ directly implies $S_N = 0$.*

2. $X_1, X_2, \ldots \overset{iid}{\sim}{}^{a} X$

3. N *and* $(X_j)_{j\in\mathbb{N}}$ *are independent* $\forall j$

All random variables are defined on the given probability space $(\Omega, \mathcal{F}, P)$.

[a] independent and identically distributed

Remark 2.1.2.

- The second assumption indicates that the claim sizes X_j do not affect each other in the sense that a large first claim X_1 does not reveal any information about the remaining claims $X_j,\ j \geq 2$ or vice versa.
- $X_j > 0$ holds for all $j \in \mathbb{N}_0$ P-a.s..

2.2. Theoretical Preparation

In order to study the distribution of S_N in detail, some basic stochastic knowledge is needed. The most important definitions will be repeated in the subsequent section.

2.2.1. Characteristic function

As a beginning the definition of the characteristic function is recalled, based on [5, Ch. B.7].

Definition 2.2.1.
The characteristic function of a real-valued random variable X *is given by*

$$\phi_X(t) = \mathbb{E}\left[e^{itX}\right]$$

with $t \in \mathbb{R}$ *and* $i = \sqrt{-1}$. *Additionally* $|\phi_X(t)| \leq 1$ *and* $\phi_X(0) = 1$ *holds.*

The characteristic function has the following properties:

Proposition 2.2.2.
Let X *and* Y *be random variables.*

1. *The distributions of* X *and* Y *are equivalent if and only if*

$$\phi_X = \phi_Y$$

holds. This indicates that the distribution of a RV is explicitly determined by it's characteristic function (and vice versa).

2. *If X and Y are independent*

$$\phi_{X+Y}(t) = \phi_X(t) + \phi_Y(t)$$

holds $\forall t \in \mathbb{R}$.

Proof:
Since this chapter yields only a quick repetition of the theory, we refer for a detailed proof the the cited literature, ie. [5, Ch. B.7].

□

In order to conclude this subsection we will state an example with some of the most common distributions and their characteristic functions.

Example 2.2.3.

1. **Poisson distribution**: $X \sim \text{Poi}(\lambda)$

$$\phi_X(t) = e^{\lambda(e^{it}-1)}$$

2. **Binomial distribution**: $X \sim \text{Bin}(n, p)$

$$\phi_X(t) = (1 - p + pe^{it})^n$$

3. **Negative binomial distribution**: $X \sim \text{NB}(r, p)$

$$\phi_X(t) = \left(\frac{1-p}{1-pe^{it}}\right)^r$$

2.2.2. Moment-generating function

Definition 2.2.4.
The moment-generating function (mgf) of a random variable X is given by

$$m_X(t) = \mathbb{E}[e^{tX}] = \int_{-\infty}^{\infty} e^{tX} \mathrm{d}F_X(x).$$

In the case of a non-negative random variable X, existence of the moment-generating function is ensured for $t \leq 0$. The following example will yield some explicit mgfs for various distributions:

Example 2.2.5.

1. **Poisson distribution**: $X \sim \text{Poi}(\lambda)$

$$m_X(t) = e^{\lambda(e^t-1)}$$

2. **Binomial distribution**: $X \sim \text{Bin}(n, p)$

$$m_X(t) = (1 - p + pe^t)^n$$

3. **Gamma distribution**: $X \sim \Gamma(\alpha, \beta)$

$$m_X(t) = (1 - \beta t)^{-\alpha}, \qquad \text{for } t < \frac{1}{\beta}$$

A more detailed examination of the moment-generating function and it's properties can be found in [4, Ch. 1.2] and [6, Ch. 2.2] respectively.

2.2.3. Probability-generating Function

Based on [5, Ch. B.7] the *(probability-) generating function* will be introduced in the subsequent section, since it plays an an important role during the examination of a random variable with values in $\mathbb{N}_0$. Throughout the following section we define

$$p_n := P(X = n)$$

for the random variable X and $n \in \mathbb{N}_0$. We obtain $\forall t \in [-1, 1]$:

$$\sum_{n=0}^{\infty} p_n |t^n| \leq \sum_{n=0}^{\infty} p_n = 1,$$

which implies the convergence of the power series $\sum_{n=0}^{\infty} p_n t^n$ on the interval $[-1, 1]$.

Definition 2.2.6.
The function $g_X : [0, 1] \to \mathbb{R}$ given by

$$g_X(t) := \sum_{n=0}^{\infty} p_n t^n \tag{2.2}$$

is called (probability-) generating function (pgf) of X. Note that $g_x(1) = 1$.

The following example will provide some explicit generating functions of the most important distributions.

Example 2.2.7.

1. **Poisson distribution**: $X \sim \text{Poi}(\lambda)$ leads to

$$g_X(t) = e^{\lambda(t-1)}$$

This result can be calculated with (2.2):

$$\begin{aligned} g_X(t) &= \sum_{n=0}^{\infty} e^{-\lambda} \frac{\lambda^n}{n!} t^n \\ &= e^{-\lambda} \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} \end{aligned}$$

$$= e^{-\lambda} e^{\lambda t}$$
$$= e^{\lambda(t-1)}$$

2. **Binomial distribution**: $X \sim \text{Bin}(n, p)$ results in

$$g_X(t) = (1 - p + pt)^n.$$

For the following calculations the index used in (2.2) is changed to j in order to avoid confusion with the parameter n from the binomial distribution.

$$\begin{aligned} g_X(t) &= \sum_{j=0}^{n} \binom{n}{j} p^j (1-p)^{n-j} t^j \\ &= \sum_{j=0}^{n} \binom{n}{j} (pt)^j (1-p)^{n-j} \\ &= (1 - p + pt)^n \end{aligned}$$

In order to obtain the last equation we apply the following binomial theorem:

$$(x + y)^n = \sum_{j=0}^{n} \binom{n}{j} x^j y^{n-j}$$

3. **Negative binomial distribution**: $X \sim \text{NB}(r, p)$

$$g_X(t) = \left(\frac{p}{1 - (1-p)t} \right)^r$$

what can be calculated as follows

$$\begin{aligned} g_X(t) &= \sum_{j=0}^{\infty} \binom{r+j-1}{j} p^r (1-p)^j t^j \\ &= p^r \sum_{j=0}^{\infty} \binom{r+j-1}{j} ((1-p)t)^j \\ &= p^r \sum_{j=0}^{\infty} \left[\binom{(-r-j+1)+j-1}{j} (-1)^j \right] ((1-p)t)^j \\ &= p^r \sum_{j=0}^{\infty} \binom{-r}{j} (-(1-p)t)^j \\ &= p^r (1 - (1-p)t)^{-r} \\ &= \left(\frac{p}{1 - (1-p)t} \right)^r \end{aligned}$$

Next up the most important properties of the pgf, which will be needed later on for some important proofs, are stated.

Theorem 2.2.8.
The probability-generating function g_X has the following features:

1. *g_X is monotonically increasing and continuous. Additionally*

$$0 \leq g_X(t) \leq g_X(1) = 1$$

   holds $\forall t \in [0,1]$.

2. *g_X is differentiable infinetly many times on the interval $[0,1)$.*

Proof:
For a complete proof we refer to the literature, for example the book of Klaus Schmidt [3, Ch. 4.5].

□

The previous theorem provides only the most important characteristics of the pgf, more precisely we only stated the properties that are necessary for the following theory. For a more comprehensive examination we refer once again to [3, Ch. 4.5], where a broad analysis of the subject can be found. The following remark postulates the name-giving property of the generating function.

Remark 2.2.9. Name-giving property of the pgf

$$p_n = P(X = n) = \frac{1}{n!} g_X^{(n)}(0)$$

is satisfied $\forall n \in \mathbb{N}_0$. This leads to the fact that the pgf fully determines the probability distribution of a random variable X: X_1 and X_2 have the same distribution if and only if $g_{X_1} = g_{X_2}$ holds. The relation can be understood as a bijection between the probability distribution and the generating function.

For a sequence of iid random variables $X_1, X_2, \dots X_N$ and $S_N = \sum_{j=1}^{N} X_j$ the following identity holds for $\forall t \in \mathbb{R}$:

$$g_{S_N}(t) = g_{X_1}(t) g_{X_2}(t) \dots g_{X_N}(t) \tag{2.3}$$

Example 2.2.10.
The following example applies (2.3) in order to calculate the distribution of a random variable $Z = X + Y$.

1. **Poisson distribution**: $X \sim \text{Poi}(\lambda)$ and $Y \sim \text{Poi}(\alpha)$ leads to

$$Z \sim \text{Poi}(\lambda + \alpha)$$

since

$$\begin{aligned} g_Z(t) &= g_X(t)g_Y(t) \\ &= e^{\lambda(t-1)}e^{\alpha(t-1)} \\ &= e^{(\lambda+\alpha)(t-1)} \end{aligned}$$

2. **Binomial distribution**: $X \sim \text{Bin}(n,p)$ and $Y \sim \text{Bin}(m,p)$

$$Z \sim \text{Bin}(n+m,p)$$

whereby

$$\begin{aligned} g_Z(t) &= g_X(t)g_Y(t) \\ &= (1-p+pt)^n(1-p+pt)^m \\ &= (1-p+pt)^{n+m} \end{aligned}$$

3. **Negative binomial distribution**: $X \sim \text{NB}(r,p)$ and $Y \sim \text{NB}(s,p)$

$$Z \sim \text{NB}(r+s,p)$$

which is calculated as follows

$$\begin{aligned} g_Z(t) &= g_X(t)g_Y(t) \\ &= \left(\frac{1-pt}{1-p}\right)^r \left(\frac{1-pt}{1-p}\right)^s \\ &= \left(\frac{1-pt}{1-p}\right)^{r+s} \end{aligned}$$

2.2.4. Laplace-Transform

Considering the lecture notes [6, Ch. 2.2], another helpful tool for the examination of random variables is introduced:

Definition 2.2.11.
The Laplace-Transform of a random variable X is defined by

$$L_X(t) := \mathbb{E}[e^{-Xt}].$$

In the case of independent and identically distributed random variables $X_1, X_2, \dots X_N$ and $S_N = \sum_{j=1}^{N} X_j$ the following identity holds:

$$L_{S_N}(t) = L_{X_1}(t) \dots L_{X_N}(t).$$

If X has values in $\mathbb{R}_+$ the existence of the Laplace-Transform is given $\forall t \geq 0$.

2.2.5. Convolution

The probability distribution of the sum of two or more independent random variables is the convolution of their individual distributions. In the case of two independent random variables X and Y, both with values in $\mathbb{N}_0$, the convolution is given by:

$$F^{X+Y}(m) = P(X+Y \leq m) = \sum_{l=0}^{m} P(X+Y \leq l) = \sum_{l=0}^{m}\sum_{j=0}^{l} P(X=j)P(Y=l-j)$$
$$=: F^X * F^Y(m)$$

For independent and continuous RVs X, Y in $\mathbb{R}_+$ this is equivalent to

$$F^{X+Y}(m) = P(X+Y \leq m) = \int_0^m \int_0^y f^X(x) f^Y(y-x)dx\, dy =: F^X * F^Y(m)$$

The corresponding density is given by:

$$f^X * f^Y(y) := \int_0^y f^X(x) f^Y(y-x)dx, \qquad y \geq 0.$$

The previous definitions can be abbreviated such that they hold for k different random variables as well. The *k-th convolution* is defined by $F_X^{*k} := F_X^{*(k-1)} * F^X$. For a more detailed definition we refer to [3, Ch. 3.3] and [5, Ch. B.3].

2.3. Properties of S_N

In the following section the most important moments, such as variance and expected value of the total claim amount are calculated. Additionally a statement which enables a more efficient calculation of $L_{S_N}(t)$ is shown.

Proposition 2.3.1.
Under the assumption of S_N having a compound distribution, we obtain

$$\mathbb{E}[S_N] = \mathbb{E}[N]\mathbb{E}[X] \tag{2.4}$$
$$\mathrm{Var}(S_N) = \mathbb{E}[N]\mathrm{Var}(X) + \mathrm{Var}(N)(\mathbb{E}[X])^2 \tag{2.5}$$
$$L_{S_N}(t) = g_N(L_X(t)), \quad t \geq 0 \tag{2.6}$$
$$g_{S_N}(t) = g_N(g_X(t)), \quad t \in [0,1] \tag{2.7}$$

whenever the corresponding moments exist.

Proof:
The underlying idea of the subsequent proof is taken from [5, Ch. 3.2] and [3, Ch. 7.1] respectively.
(2.4): The following calculation is based on the idea of splitting the underlying space in

accordance to possible values of N with $\sum_{n=0}^{\infty} \mathcal{X}_{\{N=n\}} = 1$ by introducing $\mathcal{X}$ as indicator function.

$$\begin{aligned}
\mathbb{E}[S_N] &= \mathbb{E}\left[\sum_{j=1}^{N} X_j\right] \\
&= \mathbb{E}\left[\sum_{n=0}^{\infty} \mathcal{X}_{\{N=n\}} \sum_{j=1}^{n} X_j\right] \\
&= \sum_{n=0}^{\infty} \mathbb{E}\left[\mathcal{X}_{\{N=n\}} \sum_{j=1}^{n} X_j\right] \\
&\downarrow \text{Independence of } X_j \text{ and } N;\ \mathbb{E}\left[\mathcal{X}_{\{N=n\}}\right] = P(N=n) \\
&= \sum_{n=0}^{\infty} P(N=n)\mathbb{E}\left[\sum_{j=1}^{n} X_j\right] \\
&\downarrow X_j \overset{iid}{\sim} X \\
&= \sum_{n=0}^{\infty} P(N=n) n \mathbb{E}[X] \\
&= \mathbb{E}[N]\mathbb{E}[X]
\end{aligned}$$

✓

(2.5): In order to calculate the variance of S_N, we have to start with $\mathbb{E}[S_N^2]$:

$$\begin{aligned}
\mathbb{E}\left[S_N^2\right] &= \mathbb{E}\left[\left(\sum_{j=1}^{N} X_j\right)^2\right] \\
&= \mathbb{E}\left[\sum_{n=0}^{\infty} \mathcal{X}_{\{N=n\}} \left(\sum_{j=1}^{n} X_j\right)^2\right] \\
&= \sum_{n=0}^{\infty} \mathbb{E}\left[\mathcal{X}_{\{N=n\}} \left(\sum_{j=1}^{n} X_j\right)^2\right] \\
&\downarrow \text{Independence of } X_j \text{ and } N;\ \mathbb{E}\left[\mathcal{X}_{\{N=n\}}\right] = P(N=n) \\
&= \sum_{n=0}^{\infty} P(N=n)\mathbb{E}\left[\left(\sum_{j=1}^{n} X_j\right)^2\right] \\
&\downarrow \mathbb{E}\left[(X)^2\right] = \mathrm{Var}(X) + (\mathbb{E}[X])^2
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{n=0}^{\infty} P(N=n) \left(\mathrm{Var}\left(\sum_{j=1}^{n} X_j \right) + \left(\mathbb{E}\left[\sum_{j=1}^{n} X_j \right] \right)^2 \right) \\
&\downarrow X_j \overset{iid}{\sim} X \\
&= \sum_{n=0}^{\infty} P(N=n)(n\mathrm{Var}(X) + (n\mathbb{E}[X])^2) \\
&= \mathbb{E}[N]\mathrm{Var}(X) + \mathbb{E}[N^2](\mathbb{E}[X])^2
\end{aligned}
$$

The desired result is obtained by:

$$
\begin{aligned}
\mathrm{Var}(S_N) &= \mathbb{E}[S_N^2] - \mathbb{E}[S_N]^2 \\
&\downarrow \text{applying the identities of } \mathbb{E}[S_N^2] \text{ and (2.4)} \\
&= \mathbb{E}[N]\mathrm{Var}(X) + (\mathbb{E}[N^2] - (\mathbb{E}[N])^2)(\mathbb{E}[X])^2 \\
&= \mathbb{E}[N]\mathrm{Var}(X) + \mathrm{Var}(N)(\mathbb{E}[X])^2
\end{aligned}
$$

✓

(2.6): The first equality in the subsequent proof follows directly from the definition of the Laplace-Transform $L_X(t) = \mathbb{E}[e^{-tX}]$.

$$
\begin{aligned}
L_{S_N}(t) &= \mathbb{E}\left[e^{-t\sum_{j=1}^{N} X_j} \right] \\
&= \mathbb{E}\left[\prod_{j=1}^{N} e^{-tX_j} \right] \\
&\downarrow \text{Tower rule; Law of total expectation} \\
&= \mathbb{E}\left[\mathbb{E}\left[\prod_{j=1}^{N} e^{-tX_j} | N \right] \right] \\
&\downarrow \text{Independence of } X_j \ \forall j \\
&= \mathbb{E}\left[\prod_{j=1}^{N} \mathbb{E}\left[e^{-tX_j} | N \right] \right] \\
&\downarrow X_j \overset{iid}{\sim} X \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[e^{-tX} | N \right] \right)^N \right] \\
&\downarrow \text{Independence of } X \text{ and } N \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[e^{-tX} \right] \right)^N \right] \\
&\downarrow \text{Definition of the Laplace-Transform} \\
&= \mathbb{E}\left[(L_X(t))^N \right]
\end{aligned}
$$

$$\begin{aligned} &\downarrow \text{Definition of the pgf} \\ &= g_N(L_X(t)), \qquad t \geq 0 \end{aligned}$$

✓

(2.7): The last equation is shown similar to the previous one.

$$\begin{aligned} g_{S_N}(t) &= \mathbb{E}[t^{S_N}] \\ &= \mathbb{E}\left[\sum_{n=0}^{\infty} \mathcal{X}_{\{N=n\}} t^{\sum_{j=1}^{n} X_j}\right] \\ &\downarrow \text{Independence of } X_j \text{ and } N;\ \mathbb{E}\left[\mathcal{X}_{\{N=n\}}\right] = P(N=n) \\ &= \sum_{n=0}^{\infty} P(N=n)\mathbb{E}\left[\prod_{j=1}^{n} t^{X_j}\right] \\ &\downarrow X_j \overset{iid}{\sim} X \\ &= \sum_{n=0}^{\infty} P(N=n)(\mathbb{E}[t^X])^n \\ &\downarrow \text{Definition pgf} \\ &= \sum_{n=0}^{\infty} P(N=n)(g_X(t))^n \\ &= \mathbb{E}[(g_X(t))^N] \\ &\downarrow \text{Definition pgf} \\ &= g_N(g_X(t)) \end{aligned}$$

□

Remark 2.3.2.

- Equation (2.4) is known as *1st equation of Wald*. The proof can be abbreviated in the case of X_i only having values in $\mathbb{N}_0$: if so, one can apply the chain rule to the probability generating function, presumed (2.6) holds.
- (2.5) is referred to as *2nd equality of Wald* or also as *Blackwell-Girshik equality*.
- (2.6) leads to an important simplification in terms of calculating $L_{S_N}(t)$: it states that the Laplace-Transform of the total claim size can also be computed as the moment generating function of the Laplace-Transform of the individual claim sizes. The same abbreviation is possible for the generating function, as shown in (2.7).

M. Wüthrich laid the foundations for the previous section, see [4, Ch. 2]. Additionally the books [5, Ch. 3] and [3, Ch. 7.1] completed the theory.

2.4. Compound Distribution

In order to get a better understanding of the total claim size, a broad analysis of the random variable and it's behavior is necessary. In the following section two crucial examinations will be presented: the first one will yield an upper bound for the probability of exceeding the total claim amount with the help of the Cantelli Inequality. The second approach computes the compound distribution based on the convolution.

2.4.1. Cantelli Inequality

Based on [5, Ch. 1.2] the following section develops a first idea of the behavior of S_N. To do so the so-called Cantelli Inequality has to be recapitulated.

Lemma 2.4.1. ***General Cantelli Inequality***
Let $(\Omega, \mathcal{F}, P)$ be a suitable probability space and $X : \Omega \to \mathbb{R}$ a corresponding real-valued random variable. If the second moment of X is finite, i.e. $\mathbb{E}[X^2] < \infty$,

$$P(X \geq \mathbb{E}[X] + c) \leq \frac{\mathrm{Var}(X)}{c^2 + \mathrm{Var}(X)} \tag{2.8}$$

holds for some $c \in \mathbb{R}_+$.

(2.8) can be adapted such that it finds an application in collective risk modeling:

Lemma 2.4.2. ***Cantelli Inequality in collective risk modeling***

$$\begin{aligned} P(S_N \geq \mathbb{E}[S_N] + c) &\leq \frac{\mathrm{Var}(S_N)}{c^2 + \mathrm{Var}(S_N)} \\ &= \frac{\mathbb{E}[N]\mathrm{Var}(X) + \mathrm{Var}(N)(\mathbb{E}[X])^2}{c^2 + \mathbb{E}[N]\mathrm{Var}(X) + \mathrm{Var}(N)(\mathbb{E}[X])^2} \end{aligned} \tag{2.9}$$

holds $\forall c \in (0, \infty)$.

Proof:
Follows directly from the Cantelli inequality (2.8) and the second equality of Wald (2.5). □

With a view to the practical application of (2.9) we have to introduce the term *premium*: Non-life insurance companies provide contracts to prevent loss. The costs a client has to face in order to purchase such an insurance are called premiums. The final fee is put together from several individual components. The most important ones are introduced below:

- c is the *safety margin*,
- $\mathbb{E}[S_N]$ is called *net premium* and
- $\mathbb{E}[S_N] + c$ is the *risk premium*.

The safety margin has the purpose to cover extremely high claim sizes in the case of potential disasters for example. The risk premium is an intermediate step in the calculation of the final premium a customer will have to face. Additional steps during the calculation also take taxes or profit markups into account. The complete composition of such premiums can be found in [7] for example.
If the premiums are not large enough to cover all occurring insurance claims within the underlying period of time, we use the term *ruin*. The probability of *ruin with respect to the safety margin* c is given by

$$P(S_N > \mathbb{E}[S_N] + c)$$

In practical applications the Cantelli Inequality (2.9) is used in order to find a safety margin c, such that the probability of ruin does not exceed a given bound.

Proposition 2.4.3. ***Safety margin***
Let

$$c \geq \sqrt{\frac{1-\epsilon}{\epsilon}}\sqrt{\mathrm{Var}(S_N)}$$

such that $\epsilon \in (0,1)$ *is a bound for the probability of ruin. Then*

$$P(S_N \geq \mathbb{E}[S_N] + c) \leq \epsilon$$

holds.

Proof:
We substitute $c \in (0,\infty)$ with $d := \frac{c}{\sqrt{\mathrm{Var}(S_N)}}$. With (2.9) we obtain:

$$P\left(S_N \geq \mathbb{E}[S_N] + d\sqrt{\mathrm{Var}(S_N)}\right) \leq \frac{\mathrm{Var}(S_N)}{d^2\mathrm{Var}(S_N) + \mathrm{Var}(S_N)} = \frac{1}{d^2+1} \leq \epsilon$$

The inequality

$$\frac{1}{d^2+1} \leq \epsilon$$

is equal to

$$\sqrt{\frac{1-\epsilon}{\epsilon}} \leq d = \frac{c}{\sqrt{\mathrm{Var}(S_N)}}$$

which proves the statement.

□

2.4.2. Convolution-based calculation of the distribution of S_N

Since the Cantelli Inequality only yields a general idea of the probability of ruin, a deeper understanding of the distribution of S_N is necessary. Following the idea of [4, Ch. 2.1] we will have a closer look on the compound distribution $F^{S_N}(x)$:

$$
\begin{aligned}
F^{S_N}(x) &= P(S_N \leq x) \\
&\downarrow \text{Law of total probability} \\
&= \sum_{k \in \mathbb{N}_0} P(S_N \leq x, N = k) \\
&\downarrow \text{Definition of the conditional expectation} \\
&= \sum_{k \in \mathbb{N}_0} P(N = k) P(S_N \leq x | N = k) \\
&\downarrow k \to N \\
&= \sum_{k \in \mathbb{N}_0} P(N = k) P(S_k \leq x) \\
&\downarrow \text{Definition of the convolution; } X_j \sim X; \\
&= \sum_{k \in \mathbb{N}_0} P(N = k) F_X^{*k}(x)
\end{aligned}
$$

Since F_X^{*0} is not well-defined, we have to refine the result slightly.

$$
\begin{aligned}
F^{S_N}(x) &= \sum_{k \in \mathbb{N}_0} P(N = k) F_X^{*k}(x) \\
&= P(N = 0) P(S_0 \leq x) + \sum_{k \in \mathbb{N}} P(N = k) F_X^{*k}(x) \\
&\downarrow P(N = 0) P(S_0 \leq x) = P(N = 0) P(0 \leq x) = P(N = 0) \\
&= P(N = 0) + \sum_{k \in \mathbb{N}} P(N = k) F_X^{*k}(x) \qquad (2.10)
\end{aligned}
$$

Overall we obtained a closed solution for the distribution function of S_N with (2.10). Calculating the n-th convolution requires $\mathcal{O}(n^3)$ operations, which results in reduced practical interest of doing so. The following two chapters **3** and **4** will provide some alternative approaches to calculate the distribution of the total claim amount S_N. One method is based on a recursion while another one applies a Fast Fourier Transform, both will yield a (much) better result in terms of complexity.

2.5. Possible distributions for the individual claim sizes

The chapter about collective risk modeling will be concluded with a few examples for possible distributions for the claim sizes, as well as the claim heights.
In order to model the individual claim sizes appropriately we need some basic assumptions for the utilized distributions:

- values of X should be in $\mathbb{R}_+$,
- the majority of the mass should be near 0 and

- depending on the exact application, large statistical outliers might be necessary.

To ensure an appropriate adjustment to real data a distribution with two parameters is required. Compare for example with [6, Ch. 2.3].
In general it depends strongly on the scope of application of the considered insurance, which characteristics are more important than others.

Example 2.5.1.
We introduce several distributions, which fulfill the given properties and hence find a wide application in various insurances.

1. **Gamma distribution**: $X \sim \Gamma(\alpha, \beta)$ with the density

$$f^X(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \qquad x > 0.$$

   If we choose $\alpha < 1$ we fulfill the property that as much mass of the distribution as possible lies near 0. The Gamma-distribution is often used for insurance contracts without extreme claim heights, such as car insurances.

2. **Log-normal distribution**: $X \sim \mathrm{LN}(\mu, \sigma^2)$ with the parameters $\mu \in \mathbb{R}$ and $\sigma > 0$. The distribution function is given by

$$F^X(x) = \Phi\left(\frac{(\ln x) - \mu}{\sigma}\right)$$

   With the cumulative distribution function Φ of the standard normal distribution . This distribution is used if extreme claim heights might occur, for example in the case of fire.

3. **Pareto distribution**: $X \sim \mathrm{Par}(\alpha, c)$ with $a, c > 0$ and

$$f^X(x) = \begin{cases} \frac{\alpha}{c}\left(\frac{c}{x}\right)^{\alpha+1} & \text{if } x > c \\ 0 & \text{if } x \leq c \end{cases}$$

   The Pareto distribution is used for large claim amounts which might spread across a very wide range.

The choice of the most suitable distribution apparently depends on the exact type of insurance and it's insured goods. A more extensive analysis of this relation can be found in [8, Ch. 1.2] , as well as in [6, Ch. 2.3].

2.6. Models for the distribution of N

The following section will show the most common distributions for the number of claims. Additionally the expected value and the variance of S_N are going to be calculated with the

previously shown equations of Wald: (2.4) and (2.5). Due to simplicity we introduce the following notations for the expected value and higher moments of X respectively.

$$\begin{aligned} \mathbb{E}[X] &= \mu \\ \mathbb{E}[X^k] &= \mu_k \qquad \forall k \geq 2,\ k \in \mathbb{N} \end{aligned}$$

2.6.1. Compound binomial distribution

For an adequate model based on the binomial distribution one needs some additional assumptions: the underlying time interval $[0, T]$ can be split up into n smaller and independent intervals I_k with $k = 1, \dots n$. Furthermore there is only one claim per interval I_k which happens with the probability $p \in (0, 1)$. This implies $N \sim \text{Bin}(n, p)$ and allows us to compute the expected value and the variance accordingly.

$$\begin{aligned} \mathbb{E}[S_N] &= np\mu \\ \text{Var}(S_N) &= np(1-p)\mu^2 + np(\mu_2 - \mu^2) = np(\mu_2 - p\mu^2) \end{aligned}$$

The example is based on [8, Ch. 1.2], where the properties of the compound binomial distribution were examined in detail.

2.6.2. Compound Poisson distribution

In order to present an example based on the compound Poisson distribution, some technical preparation is necessary. Therefore we will start with a basic definition of the distribution:

Definition 2.6.1.
A random variable X follows a compound Poisson distribution if $\lambda > 0$ and a distribution function F exist, such that

$$F^X(x) = \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} e^{-\lambda} F^{*k}(x), \quad x \in \mathbb{R}$$

holds. One writes $X \sim \text{CPoi}(\lambda, F)$.

Such a random variable can easily be constructed for the total claim amount S_N: presumed $\lambda > 0$ and a distribution F are given, if one chooses $N \sim \text{Poi}(\lambda)$ independent from $X_1, X_2, \dots \overset{iid}{\sim} X$ with $F^X = F$, we obtain

$$S_N = \sum_{j=1}^{N} X_j \quad \sim \text{CPoi}(\lambda, F).$$

The compound Poisson distribution has two important properties, which are presented in the following remark.

Remark 2.6.2.

- **Aggregation**: describes the property, that the sum over independent and CPoi - distributed random variables is again CPoi-distributed. This property can be quite helpful if one has a couple of independent sub-portfolios, which follow a compound Poisson distribution and one wants to derive a statement about the whole portfolio, which is put together from those individual sub-portfolios. The aggregation property implies that the whole portfolio follows again a CPoi distribution. This approach is also referred to as *bottom-up approach*.
- **Decomposition**: also known as *top-down approach*, it's the exact opposite of the aggregation property. One can hence start with a CPoi-distributed portfolio and derive independent sub-portfolios which each follow a Poisson distribution.

Since we only need the top-down approach for further calculations, we refer to [6, Ch. 3.2] for a detailed description of the aggregation property and a complete technical analysis of the compound Poisson distribution. The following theorem yields the theoretical background for the decomposition of a CPoi-distributed random variable:

Theorem 2.6.3.
Let $S \sim \text{CPoi}(\lambda, F)$ *be given with* $F = F^X$ *and* $P(X = y_j) = p_j$ *for* $j = 1, \dots m$, $\sum_{j=1}^{m} p_j = 1$. *This implies the existence of independent Poisson distributed random variables* $N_1, \dots N_m$ *with* $N \sim \text{Poi}(p_k \lambda)$, *such that*

$$S \sim \sum_{k=1}^{m} y_k N_k.$$

Proof: For a complete proof we refer to the literature, ie. [6, Ch. 3.2]. □

In order to examine a compound Poisson model with parameter λ, we take [8, Ch. 1.2] as an example and start with the calculation of the most important moments such as expected value and variance of the total claim amount.

$$\begin{aligned} \mathbb{E}[S_N] &= \lambda\mu \\ \text{Var}(S_N) &= \lambda\mu^2 + \lambda(\mu_2 - \mu^2) = \lambda\mu_2 \end{aligned}$$

At last we will calculate the skewness of S_N in order to point out a possible weakness of the model in practical applications.

$$\gamma = \frac{\mathbb{E}[(S_N - \mathbb{E}(S_N))^3]}{(\text{Var}(S_N))^{3/2}} = \frac{\mu_3}{\sqrt{\lambda\mu_2^3}} > 0$$

The fact that the skewness is always positive might complicate the fitting to real data, since it does not allow too much fluctuation. Overall it would be beneficial if γ could take positive as well as negative values.

2.6.3. Doubly stochastic Poisson Process

The doubly stochastic Poisson process, also known as the *Cox process* is introduced in order to allow more fluctuation in comparison to the previous model. It's based on the stochastic choice of the parameter λ. Let H be the distribution function of λ:

$$P(N = k) = \mathbb{E}[P(N = k|\lambda)] = \mathbb{E}\left[\frac{\lambda^k}{k!}e^{-\lambda}\right] = \int_0^\infty \frac{l^k}{k!}e^{-l}\mathrm{d}H(l)$$

This change, compared to the compound Poisson distribution, allows the skewness of S_N to get negative as well. An appropriate Cox process therefore approximates real data in general better than a compound Poisson process. This model is based on similar calculations, which can be found in [8, Ch. 1.2].

2.7. Single probabilities of the total claim amount

In order to apply the previous theory, we will present an example based on the idea of [6, Ch. 3.3]. We consider the following collective risk model: let $N \sim \mathrm{Poi}(6)$ and $X_1, X_2, \ldots$ be independent and identically distributed random variables.

$$P(X_1 = x) = \begin{cases} \frac{1}{3} = p_1, & x = 1 \\ \frac{1}{2} = p_2, & x = 2 \\ \frac{1}{6} = p_3, & x = 3 \end{cases}$$

These assumptions yield, based on Theorem 2.6.3, the following fundamentals for the subsequent calculations: $S_N = \sum_{j=1}^3 y_j N_j = N_1 + 2N_2 + 3N_3$ with

$$N_1 \sim \mathrm{Poi}(2), \quad N_2 \sim \mathrm{Poi}(3) \quad \text{and} \quad N_3 \sim \mathrm{Poi}(1).$$

The explicit distribution of each N_j follows from $N_j \sim \mathrm{Poi}(\lambda y_j)$ and $P(X = y_j) = p_j$ for $j = 1, 2, 3$. As final result we obtain

$$\begin{aligned} P(S_N = 0) &= e^{-6} \\ P(S_N = 1) &= 2e^{-6} \\ P(S_N = 2) &= 5e^{-6} \\ P(S_N = 3) &= \frac{25}{3}e^{-6} \\ &\ldots \end{aligned}$$

after a very expensive calculation of

$$P(N_j = k) = \frac{\lambda_j^k}{k!}e^{-\lambda_j}$$

under the consideration of $P(S_N = k) = P(N_1 + 2N_2 + 3N_3 = k)$ and the following table 2.1.

The computation is apparently quite costly and not of practical interest, therefore we will focus on possible approximation methods for a collective risk model from now on. The same example will be recalculated with the help of the so-called Panjer recursion in the subsequent section, see 3.4.2. In doing so, we can show a much quicker approach for calculating the single probabilities of S_N.

$\mathbf{k}$	$\mathbf{P(N_1 = k)}$, $\lambda_1 = 2$	$\mathbf{P(2N_2 = k)}$, $\lambda_2 = 3$	$\mathbf{P(3N_3 = k)}$, $\lambda_3 = 1$	$\mathbf{P(S_N = k)}$
0	$\frac{2^0}{0!}e^{-2} = e^{-2}$	$\frac{3^0}{0!}e^{-3} = e^{-3}$	$\frac{1^0}{0!}e^{-1} = e^{-1}$	$\begin{aligned} P(S_N = 0) &= P(N_1 = 0)P(2N_2 = 0)P(3N_3 = 0) \\ &= e^{-2}e^{-3}e^{-1} \\ &= e^{-6} \end{aligned}$
1	$\frac{2^1}{1!}e^{-2} = 2e^{-2}$	0	0	$\begin{aligned} P(S_N = 1) &= P(N_1 = 1)P(2N_2 = 0)P(3N_3 = 0) \\ &= 2e^{-2}e^{-3}e^{-1} \\ &= 2e^{-6} \end{aligned}$
2	$\frac{2^2}{2!}e^{-2} = 2e^{-2}$	$\frac{3^{\frac{2}{2}}}{\frac{2}{2}!}e^{-3} = 3e^{-3}$	0	$\begin{aligned} P(S_N = 2) &= P(N_1 = 2)P(2N_2 = 0)P(3N_3 = 0) \\ &\quad +P(N_1 = 0)P(2N_2 = 2)P(3N_3 = 0) \\ &= 2e^{-2}e^{-3}e^{-1} + e^{-2}3e^{-3}e^{-1} \\ &= 5e^{-6} \end{aligned}$
3	$\frac{2^3}{3!}e^{-2} = \frac{4}{3}e^{-2}$	0	$\frac{1^{\frac{3}{3}}}{\frac{3}{3}!}e^{-1} = e^{-1}$	$\begin{aligned} P(S_N = 3) &= P(N_1 = 3)P(2N_2 = 0)P(3N_3 = 0) \\ &\quad +P(N_1 = 0)P(2N_2 = 0)P(3N_3 = 3) \\ &\quad +P(N_1 = 1)P(2N_2 = 2)P(3N_3 = 0) \\ &= \frac{4}{3}e^{-2}e^{-3}e^{-1} + e^{-2}e^{-3}e^{-1} \\ &\quad +2e^{-2}3e^{-3}e^{-1} \\ &= \frac{25}{3}e^{-6} \end{aligned}$

Figure 2.1.: Calculations for the distribution of S_N

3. Recursive calculation of the total claim amount

The aim of the following chapter is the derivation of a formula for the determination of the single probabilities of S_N, whereby a specific property of the claims count distribution is assumed and a recursion for the compound distribution of S_N is derived subsequently. The Panjer recursion was originally introduced by Harry H. Panjer in 1981 in his publication „*Recursive Evaluation of a Family of Compound Distributions*“ [1].

3.1. The family of Panjer distributions

The theoretical examination of the so-called Panjer distributions in the subsequent section is in general based on [5, Ch. 3.4]. However [3, Ch. 7.2] provides in some cases a more detailed analysis of the same theory. We will presume

$$P(X \in \mathbb{N}_0) = 1 \quad \Rightarrow \quad P(S_N \in \mathbb{N}_0) = 1.$$

Until further notice we define the following probability weights

$$p_k = P(N = k), \quad k \in \mathbb{N}_0.$$

Since $X_j = 1 \ \forall j$ implies $S_N = N$ it is sufficient for the existence of a recursion formula for the total claim amount that there exists a recursion for N. In order to define the family of Panjer distributions, we start with the *Panjer recursion* itself:

$$p_k = \left(a + \frac{b}{k}\right) p_{k-1} \tag{3.1}$$

with $a, b \in \mathbb{R}$. One can impose a further condition on the choice of a and b after a short examination of the given formula:

→ The case $k = 1$ leads to $p_1 = (a+b)p_0$ which implies $a+b \geq 0$, since probability weights always have to be positive.

→ The condition $a + b \geq 0$ can be reinforced such that it becomes strict. Considering the case $a + b = 0$ one obtains

$$p_1 = (a + b)\, p_0 = 0$$
$$p_2 = \left(a + \frac{b}{2}\right) p_1 = 0$$

$$\dots$$
$$p_n = \left(a + \frac{b}{n}\right) p_{n-1} = 0$$

which implies $p_0 = 1$, because of $\sum_{j=0}^{N} p_j = 1$. Hence $P(N = 0) = 1$ holds, which is not of interest in an examination of S_N and therefore the case of $a + b = 0$ will be excluded from now on.

The following lemma gives a first idea of which distributions might satisfy (3.1):

Lemma 3.1.1. ***Recursive representation of different distributions***

1. ***Poisson distribution***: $N \sim \text{Poi}(\lambda)$

$$p_k = \frac{\lambda}{k} p_{k-1} \tag{3.2}$$

2. ***Binomial distribution***: $N \sim \text{Bin}(n, p)$

$$p_k = \left(\frac{n+1}{k} - 1\right) \frac{p}{1-p} p_{k-1} \tag{3.3}$$

3. ***Negative binomial distribution***: $N \sim \text{NB}(r, p)$

$$p_k = \left(\frac{r-1}{k} + 1\right) (1-p) p_{k-1} \tag{3.4}$$

Proof:
(3.2): The Poisson distribution yields the following probability weights

$$p_k = \frac{\lambda^k}{k!} e^{-\lambda}$$
$$p_{k-1} = \frac{\lambda^{k-1}}{(k-1)!} e^{-\lambda}$$

which lead to

$$p_k \overset{(3.2)}{=} \frac{\lambda}{k} \left(\frac{\lambda^{k-1}}{(k-1)!} e^{-\lambda}\right) = \frac{\lambda^k}{k!} e^{-\lambda}$$

and therefore proofs the assumption. ✓
(3.3): The next proof follows the same idea as before: one starts with

$$p_k = \binom{n}{k} p^k (1-p)^{n-k}$$
$$p_{k-1} = \binom{n}{k-1} p^{k-1} (1-p)^{n-(k-1)}$$

which implies

$$p_k \overset{(3.3)}{=} \left(\frac{n+1}{k} - 1\right) \frac{p}{1-p} \left(\binom{n}{k-1} p^{k-1}(1-p)^{n-(k-1)}\right)$$
$$p_k = \left(\frac{n+1}{k} - 1\right) \binom{n}{k-1} p p^{k-1}(1-p)^{-1}(1-p)^{n-k+1}$$
$$p_k \overset{(*)}{=} \binom{n}{k} p^k (1-p)^{n-k}$$

In order to show $(*)$ the following identity of the binomial coefficient is necessary:

$$\binom{n}{k} = \frac{n-k+1}{k} \binom{n}{k-1}$$

✓

(3.4): The negative binomial distribution has the following probability weights:

$$p_k = \binom{r+k-1}{k} p^r (1-p)^k$$
$$p_{k-1} = \binom{r+k-2}{k-1} p^r (1-p)^{k-1}$$

We obtain:

$$p_k \overset{(3.4)}{=} \left(\frac{r-1}{k} + 1\right)(1-p) \left(\binom{r+k-2}{k-1} p^r (1-p)^{k-1}\right)$$
$$= \left(\frac{r-1}{k} + 1\right) \binom{r+k-2}{k-1} p^r (1-p)(1-p)^{k-1}$$
$$\overset{(**)}{=} \binom{r+k-1}{k} p^r (1-p)^k$$

Equation $(**)$ follows from the subsequent property of the binomial coefficient:

$$\binom{n}{k} = \frac{n}{k} \binom{n-1}{k-1},$$

□

This Lemma allows the conclusion, that if N follows either a Poisson, binomial or negative binomial distribution, one can find $a, b \in \mathbb{R}$ with $a + b > 0$ such that equation (3.1) is fulfilled for all $k \in \mathbb{N}$. Note that the Bernoulli distribution is a special case of the binomial distribution with $n = 1$, therefore $\text{Ber}(p)$ also satisfies the Panjer recursion. The family of *all* distributions which fulfill the recursion is classified in the following theorem. It takes into account that the

recursion can be written as a differential equation with respect to the probability generating function of N.

Theorem 3.1.2. ***Panjer class***
The following statements are equivalent for $a, b \in \mathbb{R}$ with $a + b > 0$:

1. $\forall k \in \mathbb{N}$:

$$p_k = \left(a + \frac{b}{k}\right) p_{k-1} \tag{3.5}$$

2. $\forall t \in [0, 1)$:

$$(1 - at)g'_N(t) = (a + b)g_N(t) \tag{3.6}$$

3. $\forall k \in \mathbb{N}$, $\forall t \in [0, 1)$:

$$(1 - at)g_N^{(k)}(t) = (ka + b)g_N^{(k-1)}(t) \tag{3.7}$$

in this case $a < 1$ holds.

Proof:
The following proof is based on the idea of [3, Ch. 7.2]. At first one will show that the first assumption indicates $a < 1$, this proof will be done by contradiction. Afterwards the equivalence of the three statements is shown.
We start with assuming that (3.5) holds, which indicates $p_0 > 0$ and $p_1 > 0$ because of $p_1 = (a + b)p_0$ and $a + b > 0$. In the next step we can derive an estimation of the recursion formula and show the boundedness from below for $\forall k \in \mathbb{N}$:

$$\begin{aligned} p_k &= \left(a + \frac{b}{k}\right) p_{k-1} \\ &= \frac{(k-1)a + (a+b)}{k} p_{k-1} \\ &\geq \frac{k-1}{k} a p_{k-1} \end{aligned}$$

The second equation was obtained by remodeling the fraction appropriately. The inequality is based on the assumption of $a + b > 0$, which leads to:

$$p_k \geq \frac{1}{k} p_1, \qquad \forall k \in \mathbb{N}.$$

Assuming $a \geq 1$ indicates

$$p_0 + p_1 \sum_{k=1}^{\infty} \frac{1}{k} \leq \sum_{k=0}^{\infty} p_k = 1$$

and the divergence of the harmonic series for $p_1 > 0$ we obtain a contradiction to our as-

sumption.
$\Rightarrow a < 1$ ✓
It remains to show the equivalence of the three statements.

- **(3.5) ⇒ (3.6):** Starting with the definition of the generating function

$$g_N(t) = \sum_{k=0}^{\infty} p_k t^k$$

one can calculate the first derivative and remodel the equation in order to achieve the desired result:

$$\begin{aligned}
g'_N(t) &= \sum_{k=1}^{\infty} k p_k t^{k-1} \\
&\overset{(3.5)}{=} \sum_{k=1}^{\infty} \left(a + \frac{b}{k}\right) k p_{k-1} t^{k-1} \\
&= \sum_{k=1}^{\infty} \left(\frac{(k-1)a + (a+b)}{k}\right) k p_{k-1} t^{k-1} \\
&= \sum_{k=1}^{\infty} ((k-1)a + (a+b)) p_{k-1} t^{k-1} \\
&= \sum_{k=1}^{\infty} (k-1) a p_{k-1} t^{k-1} + (a+b) \sum_{k=1}^{\infty} p_{k-1} t^{k-1} \\
&= at \sum_{k=2}^{\infty} p_{k-1} (k-1) t^{k-2} + (a+b) \sum_{k=1}^{\infty} p_{k-1} t^{k-1} \\
&\overset{(*)}{=} at \sum_{j=1}^{\infty} j p_j t^{j-1} + (a+b) \sum_{j=0}^{\infty} p_j t^j \\
&= a t g'_N(t) + (a+b) g_N(t)
\end{aligned}$$

Equality $(*)$ is based on an index shift, which allows us to rewrite the two sums as the generating function or rather the corresponding first derivative. Rewriting the final equality yields:

$$(1 - at) g'_N(t) = (a+b) g_N(t)$$

✓

- (3.6) ⇒ (3.7) : This part of the proof follows the basic induction principle. ✓

- (**3.7**) $\Rightarrow$ (**3.5**) : In order to prove this implication, remark (2.2.9) is needed, which describes the name giving property of the generating function.

$$\begin{aligned} p_k &= \frac{1}{k!} g_N^{(k)}(0) \\ &\overset{(3.7)}{=} \frac{1}{k!}(ka+b) g_N^{(k-1)}(0) \\ &= \left(a + \frac{b}{k}\right) \frac{1}{(k-1)!} g_N^{(k-1)}(0) \\ &\overset{(2.2.9)}{=} \left(a + \frac{b}{k}\right) p_{k-1} \end{aligned}$$

□

Based on Theorem 3.1.2 one can derive a characterization of all distributions which fulfill the recursion

$$p_k = \left(a + \frac{b}{k}\right) p_{k-1}$$

by solving the differential equation

$$(1 - at)h'(t) = (a + b)h(t)$$

with $a, b \in \mathbb{R}$ and $a+b > 0$. Additionally the boundary condition $h(1) = 1$ becomes necessary, which follows directly from the fact that the sum over all probability weights has to be equal to one. This leads to the following conclusion:

Conclusion 3.1.3. The following statements are equivalent:

1. There exist $a, b \in \mathbb{R}$ with $a + b > 0$ and

   $$p_k = \left(a + \frac{b}{k}\right) p_{k-1}$$

   for $\forall k \in \mathbb{N}$.

2. The claims count variable N follows either a Poisson, binomial or negative binomial distribution.

Since these distributions are explicitly defined with respect to a recursion, they can be summarized in a specific class of distributions. This class is known as *Panjer class* and the related distributions are referred to as *Panjer distributions*.
Based on Lemma 3.1.1 we can calculate the explicit values of a and b for each of the Panjer distributions. The obtained result will lay the foundation for a graphical representation of the Panjer class.

Distribution	**P(N=k)**	**a**	**b**
$\mathrm{Poi}(\lambda)$	$\frac{\lambda^k}{k!}e^{-\lambda}$	0	λ
$\mathrm{Bin}(n,p)$	$\binom{n}{k}p^k(1-p)^{n-k}$	$\frac{p}{p-1}$	$\frac{p(n+1)}{1-p}$
$\mathrm{NB}(r,p)$	$\binom{r+k-1}{k}p^r(1-p)^k$	$(1-p)$	$(1-p)(r-1)$

Figure 3.1.: Values of $a, b \in \mathbb{R}$ for the Panjer distributions

Those values or more specifically the related distributions provide necessary conditions in order to display the possible choices of (a, b) appropriately. The variables a and b have to fulfill $b \in (0, \infty)$ if the Poisson distribution is considered. In the case of a binomial distribution $-\frac{b}{a} \in \mathbb{N}$ has to hold and for the negative binomial distribution we obtain $a \in (0, 1)$ as well as $b \in (-a, \infty)$.

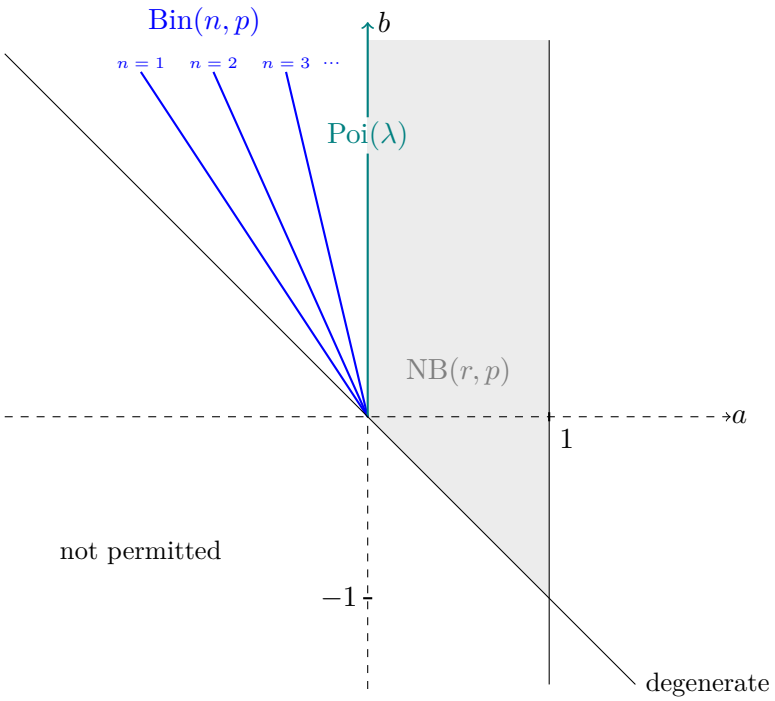


Figure 3.2.: Illustration of the Panjer class, [9]

Example 3.1.4. Binomial distribution
We will derive the quoted choices of a and b using the example of the binomial distribution. From Lemma 3.1.1 we know that

$$p_k = \left(\frac{n+1}{k} - 1\right) \frac{p}{1-p} p_{k-1}$$

holds. This has to be remodeled in such a way that one obtains an equation of the form $p_k = \left(a + \frac{b}{k}\right) p_{k-1}$.

$$\begin{aligned} p_k &= \left(\frac{n+1}{k} - 1\right) \frac{p}{1-p} p_{k-1} \\ &= \left(\frac{p}{p-1} + \frac{\frac{p(n+1)}{1-p}}{k}\right) p_{k-1} \end{aligned}$$

Which confirms the values for a and b given in table 3.1. At last the additional requirement $-\frac{b}{a} \in \mathbb{N}$ has to be examined.

$$\begin{aligned} -\frac{b}{a} &= -\frac{\frac{p(n+1)}{1-p}}{\frac{p}{p-1}} \\ &= \frac{p(n+1)}{1-p} \frac{1-p}{p} \\ &= n+1 \end{aligned}$$

Since $n \in \mathbb{N}_+$ has to be fulfilled for the binomial distribution, $(n+1) \in \mathbb{N}_+$ has to hold as well.

These observations lead to the following conclusion:

Conclusion 3.1.5. The constraints for a, b, namely

- $b \in (0, \infty)$ for the Poisson distribution,
- $-\frac{b}{a} \in \mathbb{N}$ for the binomial distribution and
- $a \in (0,1)$, $b \in (-a, \infty)$ for the negative binomial distribution

follow directly from the constraints regarding the parameters of each distribution.

A more detailed examination of the Panjer distributions, including some additional statements about the binomial coefficients of S_N, can be found in [3, Ch. 7.2].

3.2. Technical Preparation

In order to derive a recursion for the total claim amount based on (3.1), some additional theoretical preparation is necessary. The subsequent section is based on [5, Ch. 3.5]. Based

on $P(X = 1) = 1$ we obtain $g_X(t) = t$ and $S_N = N$ and hence are able to extend 3.1.2 to:

Theorem 3.2.1.
The following statements are equivalent for $a, b \in \mathbb{R}$ with $a + b > 0$:

1. *$\forall k \in \mathbb{N}$:*

$$p_k = \left(a + \frac{b}{k}\right) p_{k-1} \tag{3.8}$$

2. *For every choice of P^X and $\forall t \in [0, 1)$:*

$$(1 - ag_X(t))g'_{S_N}(t) = (a + b)g_{S_N}(t)g'_X(t) \tag{3.9}$$

3. *For every choice of P^X and $\forall k \in \mathbb{N}$, $\forall t \in [0, 1)$:*

$$(1 - ag_X(t))g^{(k)}_{S_N}(t) = \sum_{j=1}^{k} \binom{k}{j} \left(a + \frac{jb}{k}\right) g^{(k-j)}_{S_N}(t)g^{(j)}_X(t) \tag{3.10}$$

Proof:
At first we consider the case $P(X = 0) = 1$, which leads to $g_X(t) = 1$ and $g_{S_N}(t) = g_N(1) = 1$ because of equation (2.7), ie.

$$g_{S_N}(t) = g_N(g_X(t)) \tag{3.11}$$

This indicates the equivalence of the equations (3.9) and (3.10) for $\forall t \in [0, 1)$.
The following proof is all in all quite similar to the proof of theorem 3.1.2.

- **(3.8) ⇒ (3.9):** Equation (2.7) implies

$$\begin{aligned} g_{S_N}(t) &= g_N(g_X(t)) \\ g'_{S_N}(t) &= g'_N(g_X(t))g'_X(t) \end{aligned}$$

  this consideration, as well as equation (3.6) of theorem 3.1.2 lead to:

$$\begin{aligned} (1 - ag_X(t))g'_{S_N}(t) &= (1 - ag_X(t))g'_N(g_X(t))g'_X(t) \\ &= (a + b)g_N(g_X(t))g'_X(t) \\ &= (a + b)g_{S_N}(t)g'_X(t) \end{aligned}$$

  which proves the statement. ✓

- **(3.9) ⇒ (3.10):** Shown by induction. ✓

- **(3.10) ⇒ (3.8):** $P(X = 1) = 1$ implies $g_X(t) = t$ and $S_N = N$, which proves

$$(1 - at)g^{(k)}_N(t) = (ka + b)g^{(k-1)}_N(t)$$

Which concludes the proof.

□

The previous theorem almost completes the technical preparation. Last but not least we have to ensure that the values of X lie on an appropriate mesh in order to generalize the calculations. To do so some discretization methods will be introduced subsequently.

3.3. Discretization methods

Based on the paper „*Panjer Recursions versus FFT for Compound Distributions*“, see [2], we give a short overview of available discretization methods. We have to ensure that the values of X_j lie on the lattice $\{kh : k \in \mathbb{N}_0\}$ with a given mesh size $h > 0$. For simplicity we introduce the following notation:

$$f_{k,h} = P(X_1 = kh), \qquad \sum_{k=0}^{\infty} f_{k,h} = 1.$$

Additionally we set $X_j \overset{iid}{\sim} F$ for the duration of this chapter.
In practical applications an initial arithmetization is almost always necessary since F is based on a continuous distribution. Hence we choose an appropriate discretization method in order to approximate F such that the individual claim heights are on the chosen lattice. We will introduce three different approaches:

- **Rounding method**:
  The method is based on rounding the individual claim sizes to the closest integer multiple of h, ie.
  $$f_{k,h} := F\left(kh + \frac{h}{2}\right) - F\left(kh - \frac{h}{2}\right)$$
  F is replaced by $F_h = \{f_{k,h}\}_{k\in\mathbb{N}_0}$. We obtain a weak convergence of $F_h \to F$ as well as $F_h^{S_N} \to F^{S_N}$ as $h \to 0$. Even though this method is quite intuitive and easy to implement, it has a major problem: we do not know how $F_h^{S_N}$ and F^{S_N} are related to each other. This lack of knowledge results in not being able to define appropriate error bounds, which would be necessary for a better examination.

- **Forward and backward difference**:
  Since we were not able to derive error bounds for the previous method, we will now have a closer look on a method that enables to do so. The approach is based on the definition of:
  $$f_{k,h}^{+} := F((k+1)h) - F(kh)$$
  $$f_{k,h}^{-} := F(kh) - F((k+1)h)$$
  $F_{h^+}^{S_N}$ yields an upper bound for F^{S_N} while $F_{h^-}^{S_N}$ provides a corresponding lower bound. Basically we are able to sandwich the compound distribution with any given precision.

The obtained limits imply weak convergence of $F_h^{S_N} \to F^{S_N}$ as $h \to 0$.

- **Moment matching method**:
  Note that the two previous approaches did not preserve any of the corresponding moments, hence one might utilize the moment matching method instead. The main problem about this approach is the fact that it's well-defined only for $k = 1$, since it might lead to negative probability mass on some lattice points for $k \geq 2$. Therefore we will not present the theory in detail, instead we refer to [2] for example.

The following example will give a first idea of the practical application and differences of the introduced methods. The open source package *actuar* provides an implementation of all mentioned discretization methods in *R*. The corresponding functions can be called by adding „*rounding*“, „*upper*“, „*lower*“ or „*unbiased*“ as method to the *discretize* function.

Example 3.3.1. Discretization of a Gamma distribution
Consider Gamma distributed severities $X \sim \Gamma(2, 1)$. In order to compare the individual methods, we apply each of them to X, which leads to the following result:

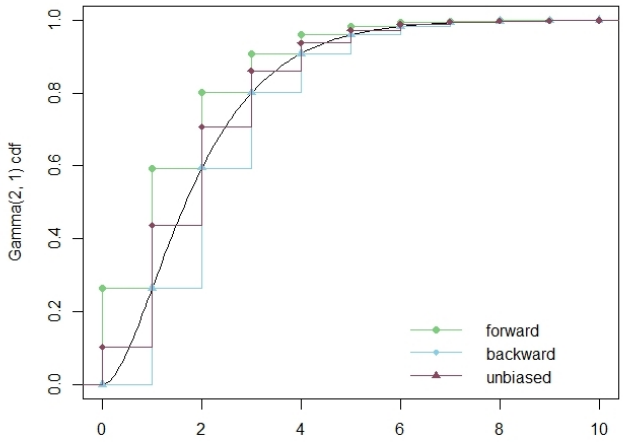


Figure 3.3.: Different discretization methods applied to $\Gamma(2, 1)$

Since the results of the rounding as well as the backwards difference were almost identical in this specific example only the backward difference was displayed in the plot. Note that we chose a rather large mesh size $h = 1$ for this example, whereby in practical applications one should in general choose a smaller h for an even better approximation of the given distribution. Example 5.1.3 will examine the appropriate choice of the bandwidth in detail.

3.4. Panjer Recursion

The technical preparation allows us to state a recursion formula for the single probabilities of the total claim amount based on [5, Ch. 3.5]:

Theorem 3.4.1. ***Panjer recursion for the total claim amount***
Let $a, b \in \mathbb{R}$ with $a + b > 0$ be, such that

$$p_k = \left(a + \frac{b}{k}\right) p_{k-1} \tag{3.12}$$

is satisfied for all $k \in \mathbb{N}$. Additionally we have $X_1, X_2, \ldots$ iid. with values on the lattice $\{kh : k \in \mathbb{N}_0\}$ and the corresponding mesh size $h > 0$ such that $f_k = P(X_1 = kh)$. These assumptions lead to:

$$P(S_N = 0) = \begin{cases} p_0, & \text{if } f_0 = 0 \\ g_N(f_0), & \text{if } f_0 > 0 \end{cases} \tag{3.13}$$

$$P(S_N = kh) = \frac{1}{1 - af_0} \sum_{j=1}^{k} \left(a + \frac{jb}{k}\right) f_j P(S_N = (k-j)h) \tag{3.14}$$

Note that the discretized probability weights f_k were denoted with $f_{k,h}$ in the previous section. The additional index h is dropped from now on in order to simplify the notation.
Proof:
(3.13): Follows directly from Lemma 3.1.1, conclusion 3.1.3 and

$$P(S_N = 0) = g_{S_N}(0) \overset{(3.11)}{=} g_N(g_X(0)) = g_N(f_0).$$

✓

(3.14): Based on Theorem 3.2.1 we obtain

$$\begin{aligned}
(1 - af_0)P(S_N = k) &= (1 - ag_X(0))\frac{1}{k!} g_{S_N}^{(k)}(0) \\
&\overset{(3.10)}{=} \frac{1}{k!} \sum_{j=1}^{k} \binom{k}{j} \left(a + \frac{jb}{k}\right) g_{S_N}^{(k-j)}(0) g_X^{(j)}(0) \\
&\downarrow \quad \frac{1}{k!}\binom{k}{j} = \frac{1}{\cancel{k!}} \frac{\cancel{k!}}{j!(k-j)!} \\
&= \sum_{j=1}^{k} \left(a + \frac{jb}{k}\right) \frac{1}{j!} g_X^{(j)}(0) \frac{1}{(k-j)!} g_{S_N}^{(k-j)}(0) \\
&= \sum_{j=1}^{k} \left(a + \frac{jb}{k}\right) f_j P(S_N = k - j)
\end{aligned}$$

✓

At last we still have to consider the mesh size $h > 0$, which we obtained from the discretization of the values of S_N:

$$(1 - af_0)P(S_N = kh) = \sum_{j=1}^{k} \left(a + \frac{jb}{k}\right) f_j P(S_N = (k-j)h)$$

$$P(S_N = kh) = \frac{1}{(1 - af_0)} \sum_{j=1}^{k} \left(a + \frac{jb}{k}\right) f_j P(S_N = (k-j)h)$$

Hence the overall result is shown.

□

The following example applies the previous conclusion in order to determine $P(S_N = k)$.

Example 3.4.2. Extension of 2.7

All values are chosen equivalent to the previous example 2.7, hence we examine a Poisson distribution with parameter $\lambda = 6$. In order to apply conclusion 3.4.1 for the calculation of the corresponding probabilities, we have to check whether the distribution of N fulfills the Panjer recursion. Since we chose $N \sim \text{Poi}(6)$ Lemma 3.1.1 implies that the recursion is satisfied with $a = 0$ and $b = \lambda$ (see table 3.1 for instance).
We obtain for $\lambda = 6$, $f_1 = \frac{1}{3}$, $f_2 = \frac{1}{2}$ and $f_3 = \frac{1}{6}$:

- $\mathbf{k = 0}$:
  $P(S_N = 0) = p_0 = e^{-6}$
- $\mathbf{k \geq 1}$:

$$\begin{aligned} P(S_N = k) &= \sum_{j=1}^{3} \frac{\lambda * j}{k} f_j P(S_N = k - j) \\ &= \frac{6}{k} * \frac{1}{3} P(S_N = k-1) + \frac{12}{k} * \frac{1}{2} P(S_N = k-2) + \frac{18}{k} * \frac{1}{6} P(S_N = k-3) \\ &= \frac{1}{k}[2P(S_N = k-1) + 6P(S_N = k-2) + 3P(S_N = k-3)] \end{aligned}$$

Based on these equations we are able to calculate the following probabilities for $k = 1, 2, 3$:

- $P(S_N = 1) = \frac{1}{1}[2P(S_N = 0)] = 2e^{-6}$
- $\begin{aligned} P(S_N = 2) &= \frac{1}{2}[2P(S_N = 1) + 6P(S_N = 0)] \\ &= \frac{1}{2}[4e^{-6} + 6e^{-6}] = 5e^{-6} \end{aligned}$
- $\begin{aligned} P(S_N = 3) &= \frac{1}{3}[2P(S_N = 2) + 6P(S_N = 1) + 3P(S_N = 0)] \\ &= \frac{1}{3}[10e^{-6} + 12e^{-6} + 3e^{-6}] = \frac{25}{3}e^{-6} \end{aligned}$

The results are equivalent to those in the original example.
The *R*-code which was used for the calculations can be found in appendix A. For an efficient implementation of the algorithm an appropriate stopping criterion is necessary, since $\sum_k f_k = 1$ is unlikely to be achieved. The user-defined function *Panjer.Poisson* is carried out until $P(S_N = 0) + P(S_N = 1) + \cdots > 0.99999$ is satisfied. For this specific example the criterion was reached after $k = 38$ iterations.
The following plot yields a graphical representation of the obtained single probabilities:

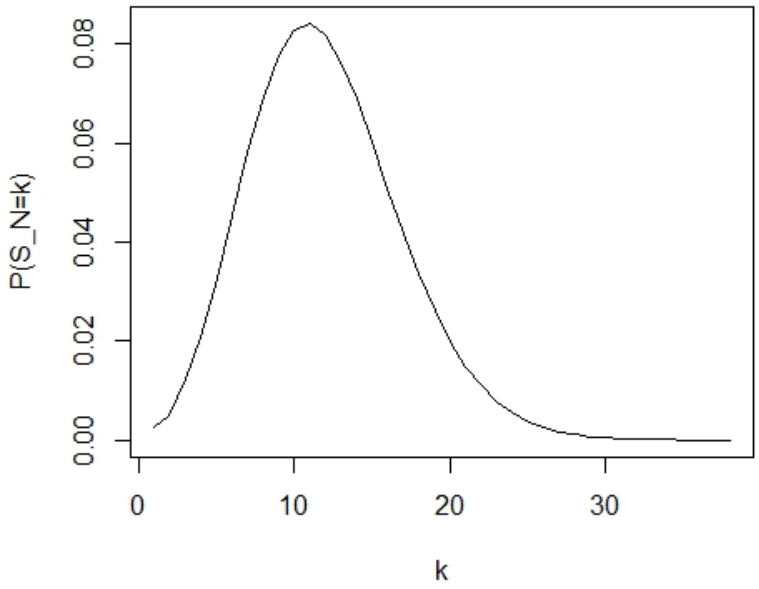


Figure 3.4.: Example 3.4.2

And the exact results are given by:

```
Panjer.Poisson(6, c(0, 1/3, 1/2, 1/6))
 [1] 2.478752e-03 4.957504e-03 1.239376e-02 2.065627e-02 3.263690e-02
 [6] 4.527854e-02 5.805788e-02 6.938539e-02 7.786921e-02 8.291382e-02
[11] 8.411991e-02 8.175731e-02 7.641463e-02 6.890253e-02 6.011177e-02
[16] 5.085884e-02 4.181849e-02 3.347796e-02 2.613574e-02 1.992604e-02
[21] 1.485502e-02 1.084160e-02 7.754156e-03 5.440128e-03 3.747082e-03
[26] 2.535896e-03 1.687487e-03 1.104874e-03 7.122271e-04 4.522814e-04
[31] 2.830849e-04 1.747271e-04 1.064002e-04 6.395205e-05 3.795549e-05
[36] 2.225212e-05 1.289148e-05 7.382761e-06
```

Overall we obtained $\sum_{k=1}^{38} P(S_N = k) = 0.9999906$, with the chosen stopping criterion.

To conclude the section, we will recall the introduced discretization methods in order to give an idea how they behave, if applied to a compound distribution.

Example 3.4.3. Discretization of a compound distribution
Consider the random variable $N \sim$Poi(10). The severity follows a Gamma distribution $X \sim \Gamma(2,1)$. Discretizing S_N under the given assumptions with a step size of $h = 1$ leads to:

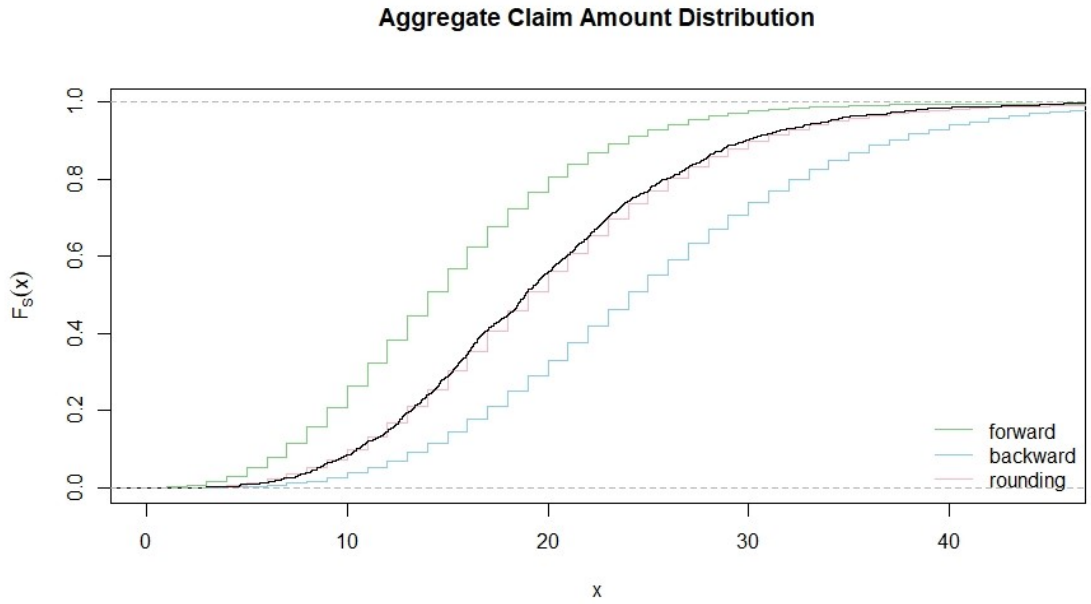


Figure 3.5.: Discretization methods for a compound distribution

As one can see, the rounding method approximates the compound distribution better than the forward or backward difference. The fact that the upper discretization is significantly bigger than the rounded one is in accordance with the theory: the forward difference assigns all the probability in $(0, 0.5)$ to 0, which means a part of the risk is being ignored.
In practical applications this might lead to the feeling, that the portfolio is safer than it actually is. The lower discretization faces the inverse problem.

Example 3.4.4.
The last example of the current section will apply the code of appendix A in order to calculate the single probabilities for $N \sim \text{NB}(4, 0.2)$ with Fréchet distributed individual claim heights $X \sim \text{Frechet}(1.7, 1)$. The lattice width was set to $h = 0.4$.

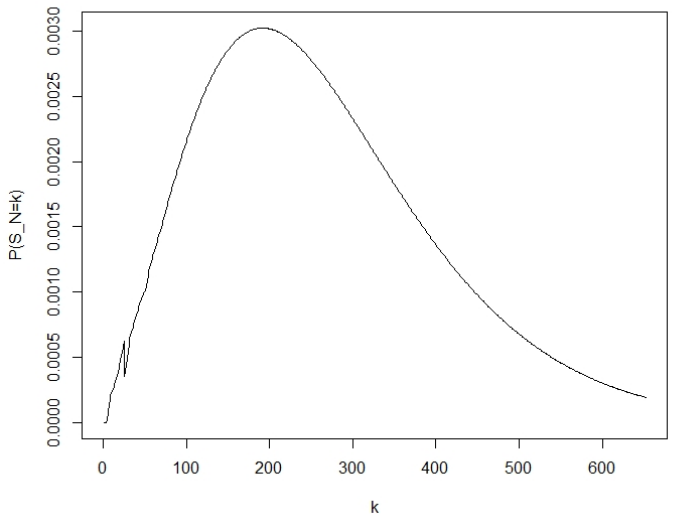


Figure 3.6.: Example 3.4.4

This example indicates an obstacle one has to keep in mind while choosing the stopping criterion: the code stops after $k = 653$ iterations with

$$\sum_{k=1}^{653} P(S_N = kh) = 1.000149$$

satisfying $\sum_k P(S_N = kh) > 0.99999$. But obviously a result larger than 1 is not beneficial under these circumstances. Comparing the obtained value to

$$\sum_{k=1}^{652} P(S_N = kh) = 0.9999555$$

would suggest to choose a slightly smaller stopping criterion in order to obtain a value like $0.9999 < \sum_k P(S_N = kh) \leq 1$.

3.5. Quality of the Recursion

In the previous section 2.4.2 the total claim amount was calculated based on the convolution. As already mentioned this brute-force calculation is not of practical interest, since it requires $\mathcal{O}(n^3)$ operations in order to calculate n single probabilities. Comparing this result to the Panjer recursion, which has the complexity of $\mathcal{O}(n^2)$, we obtained a slight advantage in terms of the calculation costs. An even better result was shown by Hipp in 2003 by assuming a

phase-type distribution[1] for the claim amount, obtaining $\mathcal{O}(n)$ necessary operations. The detailed theoretical background for this theory can be found in [10]. Besides the complexity one also has to keep the stability of the calculation as well as the possibility of an under- or overflow in mind, both cases will be examined in the subsequent sections.

3.5.1. Robustness

H. Panjer studied together with S. Wang the robustness of the recursion for several different claims count distributions in their paper „*On the Stability of Recursive Formulas*“, [11]. They were able to show a strong stability for the compound negative binomial and the compound Poisson distribution. Concerned with the occurrence of possible rounding errors Panjer and Wang were able to prove that the accumulated error bound of the recursion (3.4.1) grows linearly with a slope bounded by one. This is of great interest, since the application of computers and their limited possibilities in displaying the results with only a finite number of digits, make rounding errors inevitable. The exact derivation of this result can be found in the previously cited paper of Panjer and Wang. Moreover they were able to derive a bound for the number of valid digits $\#(r, n)$, if r digits are used to calculate $P(S_N = n)$:

$$\#(r, n) \geq r + \left\lfloor \log_{10} \frac{4}{3} - \frac{1}{2} \log_{10}(n+1) \right\rfloor \tag{3.15}$$

which holds at a 99% significance level. The practical impact of this result can be shown with a short example: considering a distribution with mean 1000 for the claim heights and number of claims, an error smaller than 10^{-7} would be desirable. This can be obtained with 14 digits. Applying (3.15) we can show with 99% confidence that a similar result can be obtained with only 11 digits. The result is also of great interest for the application of discretization methods: increasing the number of lattice points by a factor 100, only 2 more digits are necessary to maintain the accuracy level.
However these statements do not hold for the binomial case, since the binomial distribution only has finite support if the claim sizes have a finite support. Applying the previously drawn conclusions hence might lead to serious error blow-ups, due to the occurrence of mixed signs which cancel out during the calculation. The practical impacts are debatable: the binomial distribution does not find a wide application in practice anyway. Panjer and Wang still suggested a possibility to cope with the arising instabilities in [11, Ch. 10].

3.5.2. Under- and overflow

Another computational problem, which has to be kept in mind throughout the calculations is the occurrence of a potential over- or underflow[2]. For a better understanding of the problem, the origin and potential countermeasures, a fitting example is yet to follow:

[1] Probability distribution constructed by a convolution or mixture of exponential distributions.
[2] Occurrence of a number outside of the range of representable numbers of a computer, which leads to the output 0 or ∞.

Example 3.5.1. Underflow in the PR
In order to make the problem more understandable the corresponding R code and the output is presented. We consider a Poisson distributed random variable $N \sim \text{Poi}(\lambda)$ for the number of claims, as well as the following simplified severity distribution.

```
fk <- c(0, 0.05, 0.14, 0.2, 0.15, 0.12, 0.075, 0.05, 0.05, 0.1,
        0.065)
```

Note that the indices in R start with 1 instead of 0, such that $fk[1] = P(S_N = 0) = 0$. With this knowledge and the *actuar* package, we investigate a couple of possible choices for the parameter λ until we face the problem of underflow. In order to calculate the Panjer recursion, the function *aggregateDist* is used, which computes the cumulative distribution function (cdf) of the aggregate claim amount of a portfolio over a certain period of time using one of five available methods. If one wants to choose the Panjer recursion, the method has to be set to „*recursive*“ and the frequency of the model has to be either „*poisson*“, „*binomial*“, „*geometric*“ or „*negative binomial*“. We start by recursively calculating the distribution of the total claim amount S_N with $N \sim$Poi(2):

```
FSN1 <- aggregateDist("recursive", model.freq = "poisson",
                      model.sev = fk, lambda = 2)
```

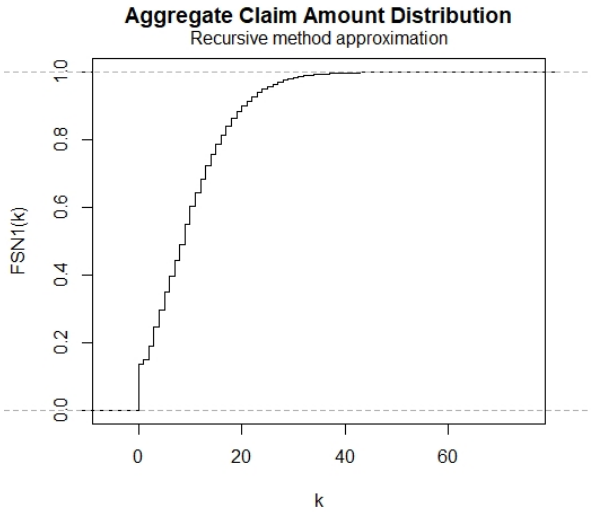


Figure 3.7.: Example 3.5.1 a)

```
summary(FSN1)
Aggregate Claim Amount Empirical CDF:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
 0.000000  4.000000  9.000000  9.759943 14.000000 70.000000
```

In order to provoke the desired underflow, we have to increase λ, since the probability of a large loss increases, the larger the parameter λ is chosen.

```
FSN2 <- aggregateDist("recursive", model.freq = "poisson",
                      model.sev = fk, lambda = 20)
```

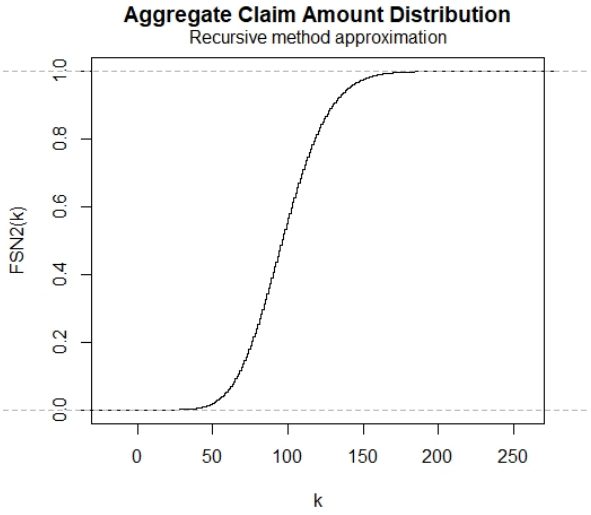


Figure 3.8.: Example 3.5.1 b)

```
summary(FSN2)
Aggregate Claim Amount Empirical CDF:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
  0.00000  80.00000  96.00000  97.59976 114.00000 240.00000
```

With regard to the presented plots, one can already see a difference between the two choices for λ: the recursion needs more iterations in order to obtain a cdf close to 1. This property directly follows from the fact that a large mean of the frequency distribution directly implies a smaller probability of S_N being small.
In the next step we will set $\lambda = 2000$, which leads to an underflow:

```
FSN3 <- aggregateDist("recursive", model.freq = "poisson",
                      model.sev = fk, lambda = 2000)

Error in panjer(fk = model.sev, dist = dist, p0 = p0,
                x.scale = x.scale):

Pr[S = 0] is numerically equal to 0; impossible to start the
 recursion
```

The choice of $\lambda = 2000$ is enough for the recursion to fail, since the probability of zero occurring claims is given by

$$P(S_N = 0) = e^{\lambda*(f_0-1)} = e^{-2000}$$

which is represented by 0 by the computer.
As a conclusion we can keep in mind that the possibility of an underflow increases with an increasing mean of the chosen severity distribution: the larger the utilized mean, the smaller the possibility of S_N being small.

A rather naive ansatz to avoid underflow would be scaling of λ with an appropriate constant for example. This would ensure $P(S_N = 0) > 0$. The second example of this section tests the suggested approach.

Example 3.5.2. Linear scaling
We suggest $\lambda_{new} = \frac{\lambda}{2^k}$, such that $P(S_N = 0) > 0$ is satisfied and subsequently try to calculate the first step of the recursion with $\lambda_{new} = 500$:

```
FSN4 <- aggregateDist("recursive", model.freq = "poisson",
                      model.sev = fk, lambda = 500, convolve = 1,
                      maxit = 500)

Warning message:
In panjer(fk = model.sev, dist = dist, p0 = p0, x.scale = x.scale:

  maximum number of recursions reached before the probability
  distribution was complete
```

For a better understanding of the error message we cite the help site of the *actuar* package with *?aggregateDist*: „Failure to obtain a cumulative distribution function less than tol away from 1 within maxit iterations is often due to a too coarse discretization of the severity distribution“. As suggested by the warning message the very small support of our severity distribution causes the recursion to fail. To avoid this problem, we have several possibilities:

1. Choose a customized „tol“ value, which is the tolerance level of the stopping criterion of the Panjer recursion: the recursion stops if the cdf is only „tol“ away from 1.

2. Increase the number of performed iterations.

3. Choose a larger k.

The first suggestion will be excluded from further examinations since we already know from example 3.4.2 that the choice of such a value is not that obvious. Instead we start by increasing the number of performed iterations from $maxit = 500$ to $maxit = 4000$:

```
FSN5 <- aggregateDist("recursive", model.freq = "poisson",
                      model.sev = fk, lambda = 500, convolve = 1,
                      maxit = 4000)
```

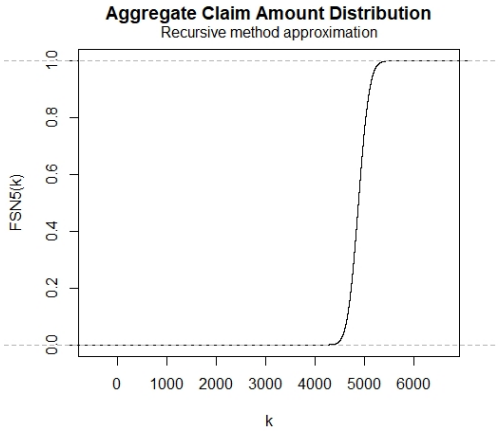


Figure 3.9.: Example 3.5.2 a)

```
summary(FSN5)
Aggregate Claim Amount Empirical CDF:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
    0.000  4761.000  4879.000  4879.994  4998.000  6148.000
```

Additionally we suggested that a larger k could solve the problem as well, hence we set *convolve* $= 2$ and obtain the following result:

```
FSN6 <- aggregateDist("recursive", model.freq = "poisson",
                      model.sev = fk, lambda = 250, convolve = 2,
                      maxit = 2000)
```

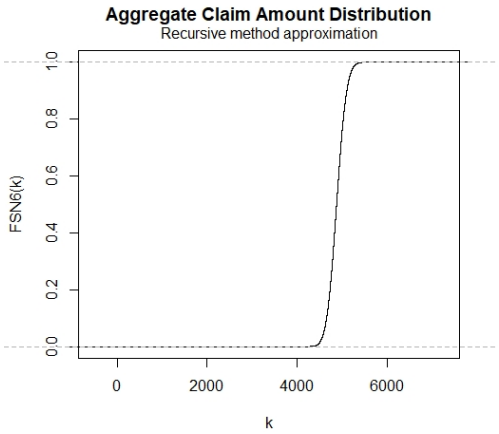


Figure 3.10.: Example 3.5.2 b)

```
summary(FSN6)
Aggregate Claim Amount Empirical CDF:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
   0.000 4761.000 4879.000 4879.995 4998.000 6760.000
```

The two implemented methods both solved the underflow. Additionally one might notice how similar the plotted cdfs are: only in the very end of the right tail slight differences arise.

Overall one has to be really careful with the presented suggestions, since the scaling might also cause an overflow, if the parameters are not chosen appropriately. Panjer and Willmot addressed a couple of solutions to prevent under- as well as overflow as a consequence of scaling in their paper „*Computational Aspects of Recursive Evaluation of Compound Distributions*“, [12]. They developed reliant solutions for the Poisson as well as the negative binomial distribution. Both approaches are based on a decomposition of the compound distribution, an individual evaluation of all independent sub-portfolios, before uniting the obtained subresults to a final conclusion. The necessary decomposition of the Poisson distribution was already a subject of the section 2.6.2.

3.6. An extension and a counterpart of the Panjer Recursion

In order to conclude the section about the Panjer recursion a short outlook on a possible extension and a related formula is given.

3.6.1. Panjer Recursion of higher orders

Hess et al. investigated in their paper „*An Extension of Panjer's Recursion*“ [13] possible extensions of the Panjer recursion. They were able to extend

$$p_k = \left(a + \frac{b}{k}\right) p_{k-1}$$

for $k \geq 1$ to $k \geq l$ with arbitrarily $l \in \mathbb{N}$. This leads to the following extension of the Panjer recursion:

Conclusion 3.6.1. Consider

$$p_k = \left(a + \frac{b}{k}\right) p_{k-1}, \quad k \geq l,$$

with the initial values $p_0 = p_1 = \cdots = p_{l-1} = 0$ and $f_0 = 0$. This leads to:

$$P(S_N = kh) = p_l f_n^{*l} + \sum_{j=1}^{k} \left(a + \frac{jb}{k}\right) f_j P(S_N = (k-j)h), \quad \forall k \geq l$$

The corresponding proof follows the same reasoning as the proof of the general version of the recursion. Throughout the previous analysis of the Panjer class the main focus lied

on distributions of the form $(a; b; 0)$ which was equivalent to the collection of all binomial, Poisson or negative binomial distributions. Considering now the Panjer class of order l we focus on distributions of the form $(a; b; l)$. This extension enables us to consider several more claims count distributions than before: the Panjer recursion of order l is also satisfied for the logarithmic distribution, the extended logarithmic, the extended negative binomial and the Engen distribution. This is indicated by the fact that p_0 has become a so-called free parameter, which can be chosen freely such that $\sum_{k=0}^{\infty} p_k = 1$ is satisfied. The collection of these distributions is referred to as *basic claim number distributions*. All previously drawn conclusions on the general Panjer class can be extended to these special cases. Especially for the catastrophe excess of loss reinsurance the Panjer recursion of order l plays an important role.

3.6.2. DePril Recursion

The following section gives a short idea of the so-called dePril recursion, which finds an application for example in the modeling of a portfolio in life insurance policies. It can be seen as a counterpart to the Panjer recursion and was introduced in 1986 by Nelson De Pril in his paper „*On the Exact Computation of the Aggregate Claims Distribution in the Individual Life Model*“, [14]. In this context we consider a portfolio of n independent life insurance policies, in opposition to before n is known and not modeled as a random variable. Each of these policies y has it's own probability of an occurring claim p_y and the benefit B_y. De Pril suggested a recursive formula in order to determine the aggregate payout distribution exactly, if the following assumptions are satisfied:

- The benefits B_y are fixed and integer multiples of some appropriately chosen base and a maximum value $M * base$, i.e. $B_i = \{1, \dots M\} * base$.
- The portfolio can be classified in to groups of J values with regard to mortality rates: $p_j = \{p_1, \dots p_J\}$.

The benefit B_y is payable if death occurs within a certain time period. The factor p_j can be understood as survival probability for the policies j, whilst $1 - p_j$ is the death probability respectively.
Let $P(S_n = k)$ describe the probability of the total claim amount being equal to k, as we already know from the previous chapters. Additionally we introduce n_{ij} as the number of policies with benefit i and the claim probability p_j. De Pril was able to prove the following recursive formula for the determination of the total claim amount:

Conclusion 3.6.2.
Let

$$A(i,j) = i(-1)^{j+1} \sum_{l=1}^{J} n_{il} \left(\frac{p_l}{1-p_l} \right)^l$$

and

$$P(S_n = 0) = \prod_{j=1}^{J} (1-p_j)^{n_{ij}}$$

then the following recursion holds

$$P(S_n = k) = \frac{1}{k} \sum_{i=1}^{\min\{a,k\}} \sum_{j=1}^{[k/i]} A(i,j) * P(S_n = k - ji)$$

where $[\dots]$ denotes the greatest integer less than or equal to x.

This formula has the benefit of being exact, but it is very computational intensive. However the calculation time can be reduced quite a bit, if one accepts small aggregate costs to the insurer. If we accept for example some figure K which the insurer has to pay by himself, the recursion can be slightly adapted to:

$$P(S_n = k)_K = \frac{1}{k} \sum_{i=1}^{\min\{a,k\}} \sum_{j=1}^{\min\{[K,k/i]\}} A(i,j) * P(S_n = k - ji)$$

Note that the upper limit in the second sum was modified. However the choice of such a parameter K is a research area on it's own.
Panjer's recursion served as motivation for De Pril to develop his recursion, which is nowadays a basic component of insurance risk analysis modeling.

4. An algorithm based on the discrete Fourier transform

After the examination of a recursive approach to determine the total claim amount, another possibility to do so will be presented in the following chapter. The basic idea is an application of a Fast Fourier Transform (FFT) in order to significantly improve the complexity of the calculation. Another main advantage of this technique is the opportunity of utilizing distributions which are not a part of the Panjer class.

4.1. Discrete Fourier Transform

In order to get a better understanding of the subsequent algorithm, we will start with the corresponding theory and recall the most important properties of a (discrete) Fourier transform.

4.1.1. General idea

The main idea behind the new approach is an efficient calculation of the characteristic function of the total claim amount S_N. Considering Paul Embrechts paper *„Panjer Recursion versus FFT for Compound Distributions“* [2] the subsequent conclusions can be drawn.

Proposition 4.1.1.
Let S_N satisfy the Model assumptions 2.1.1. Then

$$\phi_{S_N}(s) = g_N(\phi_X(s)) \tag{4.1}$$

holds with $s = e^{-it}$ and $i = \sqrt{-1}$.

Proof:
Recalling some basic properties of the total claim amount, the definition of the characteristic function ϕ and the generating function g, one obtains:

$$\begin{aligned}\phi_{S_N}(s) &= \mathbb{E}\left[e^{is\sum_{j=1}^{N} X_j}\right] \\ &= \mathbb{E}\left[\prod_{j=1}^{N} e^{isX_j}\right] \\ &\downarrow \text{Tower rule; Law of total expectation}\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\mathbb{E}\left[\prod_{j=1}^{N} e^{isX_j}|N\right]\right] \\
&\downarrow \text{Independence of } X_j\ \forall j \\
&= \mathbb{E}\left[\prod_{j=1}^{N} \mathbb{E}\left[e^{isX_j}|N\right]\right] \\
&\downarrow X_j \overset{iid}{\sim} X \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[e^{isX}|N\right]\right)^N\right] \\
&\downarrow \text{Independence of } X \text{ and } N \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[e^{isX}\right]\right)^N\right] \\
&\downarrow \text{Definition of the characteristic function} \\
&= \mathbb{E}\left[\left(\phi_X(s)\right)^N\right] \\
&\downarrow \text{Definition of the pgf} \\
&= g_N\left(\phi_X(s)\right)
\end{aligned}
$$

Which proves the desired result.

□

Since the distribution of a random variable is explicitly determined by it's characteristic function, it's sufficient to calculate the right hand side of (4.1) in order to specify the distribution of the total claim amount S_N.
However an efficient computation is in this case only possible if the severity distribution is piecewise linear or piecewise constant: Heckman and Meyers introduced in their paper *„The Calculation of Aggregate Loss Distributions from Claim Severity and Claim Count Distributions“* [15] several explicit formulas for some relevant special cases. In our case we will resort directly to the discrete version of the problem, which can be solved by applying the discrete Fourier transform.

4.1.2. Theoretical preparation

In order to explain the detailed scope of application of a Fourier transform in collective risk modeling, the general definition is yet to be repeated.

Definition 4.1.2.
Let M be some truncation point and set $f = (f_0, f_1, \dots f_{M-1}) \in \mathbb{R}^M$ such that $f_k = 0$ for $k \geq M$. The discrete Fourier transform (DFT) is defined as the following function

$$\hat{f}_t = \sum_{k=0}^{M-1} f_k e^{-i2\pi tk/M} \tag{4.2}$$

$$= \sum_{k=0}^{M-1} f_k \left[\cos\left(\frac{2\pi}{M} tk\right) - i \sin\left(\frac{2\pi}{M} tk\right) \right]$$

for $t = 0, 1, \ldots, M-1$. *The result is denoted by* $\hat{f} = \left(\hat{f}_0, \hat{f}_1, \ldots \hat{f}_{M-1}\right)$.

The last equation in the previous definition follows directly from Euler's formula

$$e^{ix} = \cos(x) + i \sin(x).$$

Considering the DFT in combination with some previously derived characteristics of the total claim amount, we are able to derive the following result:

Remark 4.1.3.
Let f be a sequence of single probabilities of the individual claim heights X and q of the total claim height S_N respectively.

$$f := \{P(X = k)\}_{k=0}^{\infty} \quad \text{and} \quad q := \{P(S_N = k)\}_{k=0}^{\infty}$$

Starting with the Fourier transform of q, we obtain the following identity:

$$\begin{aligned}
\hat{q}_t :&= \sum_{k=0}^{\infty} e^{-i2\pi tk} P(S_N = k) \\
&\downarrow \text{Definition of } \mathbb{E} \\
&= \mathbb{E}\left[e^{-i2\pi t \sum_{j=1}^{N} X_j} \right] \\
&\downarrow \text{Tower rule; Law of total expectation} \\
&= \mathbb{E}\left[\mathbb{E}\left[\prod_{j=1}^{N} e^{-i2\pi t X_j} | N \right] \right] \\
&\downarrow \text{Independence of } X_j \ \forall j \\
&= \mathbb{E}\left[\prod_{j=1}^{N} \mathbb{E}\left[e^{-i2\pi t X_j} | N \right] \right] \\
&\downarrow X_j \overset{iid}{\sim} X \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[e^{-i2\pi t X} \right] \right)^N \right] \\
&\downarrow \text{Definition of the Fourier Transform} \\
&= \mathbb{E}\left[\left(\hat{f}_t \right)^N \right] \\
&\downarrow \text{Defintion of the pgf} \\
&= g_N\left(\hat{f}_t \right)
\end{aligned}$$

Note that an infinite sequence without truncation point was considered here, hence the sum runs up to ∞ and the division by M in the exponent was dropped, compared to (4.2).
The derived equation gives a first hint on how to efficiently determine the distribution of the total claim amount S_N: overall we do know the single probabilities of the individual claim heights X, hence we are able to calculate the right hand side of

$$\hat{q} = g_N(\hat{f})$$

without a problem. This leaves us with the task of deriving the probabilities $q_k := P(S_N = k)$ from their discrete Fourier transform $\hat{q}$.

To do so, the following theorem is crucial:

Theorem 4.1.4. ***Inversion theorem***
Let f be a finite sequence, such that there exists a $M \in \mathbb{N}$ with the property

$$f_k = 0 \quad \text{for } k \geq M.$$

This implies

$$f_k = \frac{1}{M} \sum_{t=0}^{M-1} \hat{f}_t e^{i2\pi tk/M}$$

for all $k = 0, 1, 2, \ldots$.

Proof:
The following proof follows the idea of [16, Ch. 3.2]. We start by recalling the definition of the DFT (4.2) for $t = 0, 1, \ldots, M-1$:

$$\hat{f}_t = \sum_{l=0}^{M-1} f_l e^{-i2\pi tl/M} \tag{4.3}$$

The overall goal is now the evaluation of f_l in terms of $\hat{f}_t$. Since we work with different frequency domains we refined the utilized indices slightly and replaced the index k in the original definition by l in order to avoid confusion. The next step calculates the sum over the interval $k \in [0, M-1]$ and multiplies both sides of (4.3) with $e^{i2\pi tk/M}$:

$$\begin{aligned}
\sum_{t=0}^{M-1} \hat{f}_t e^{i2\pi tk/M} &= \sum_{t=0}^{M-1} e^{i2\pi tk/M} \sum_{l=0}^{M-1} f_l e^{-i2\pi tl/M} \\
&= \sum_{l=0}^{M-1} f_l \sum_{t=0}^{M-1} e^{i2\pi t(l-k)/M} \\
&\downarrow (*) \\
&= M * f_l
\end{aligned}$$

Whereby $(*)$ follows from

$$\sum_{t=0}^{M-1} e^{i2\pi t(l-k)/M} = \begin{cases} M & \text{for } l = k \\ 0 & \text{for } l \neq k \end{cases}$$

which leads to the final result

$$f_l = \frac{1}{M} \sum_{t=0}^{M-1} \hat{f}_t e^{i2\pi tl/M}.$$

□

Theorem 4.1.4 allows us to conclude the considerations of the previous remark 4.1.3 it enables us to execute the last step of deriving the single probabilities $q_k := P(S_N = k)$ from their Fourier transform $\hat{q}$, which was obtained by calculating $g_N(\hat{f}_k)$ with $f_k = P(X = k)$.
We will stick to the introduced notation for the rest of this chapter and set

$$q_k := P(S_N = k)$$

for a better comprehensibility.

4.2. Derivation of the algorithm

The previously drawn conclusions provide the underlying structure of the algorithm, however there might occur some computational problems throughout it's execution. Nevertheless we will start by pursuing the suggested approach until possible weaknesses arise, which makes other concepts inevitable. The following section is based on the paper of Paul Embrechts and Marco Frei [2], as well as the paper of Grübel and Heresmeier [17].

4.2.1. Basic structure of the algorithm

In the previous remark **4.1.3** the following approach for determining the distribution of the total claim amount was proposed:

Algorithm 4.2.1.

1. Choose a truncation point $M \in \mathbb{N}$.
2. Set $f = (f_0, f_1, \ldots, f_{M-1}) \in \mathbb{R}^M$.
3. Calculate the DFT $\hat{f} \in \mathbb{R}^M$.
4. Apply the generating function of N to the result and set $g_N(\hat{f}) =: \hat{q}$.
5. Take the inverse DFT in order to obtain q as an approximation for the distribution of the total claim amount S_N.

However one has to keep in mind, that a truncated sequence f is used during the calculation. This might lead to so-called truncation errors, which occur as soon as only partial data is

considered and there is a difference between the truncated and the actual value. This type of error can only be prevented if

$$\sum_{k=0}^{M-1} f_k = 1$$

holds, which ensures an exact calculation of the compound distribution. If this property is not fulfilled, the occurring error might become quite an issue. The subsequent section provides a detailed theoretical investigation of the error.

4.2.2. Aliasing error and exponential tilting

The introduction of a truncation point M replaces the usual summation of integers by a summation modulo M, which leads to a wrap around effect of the truncated data. The compound mass beyond M is not considered but „wrapped around“, which implies it's incorrect appearance in the observed range $[0, \dots (M-1)]$. This incident is referred to as aliasing error. Utilizing transform methods with heavy tailed distributions in the given framework, might lead to difficulties, since the whole distribution has to be considered. Rudolf Grübel and Renate Heresmeier therefore investigated in their paper *„Computation of Compound Distributions I: Aliasing Error and Exponential Tilting“* [17] the effects of such errors and concurrently suggested possible approaches on how to prevent them. Concerned with the question what impact the choice of M has on the size of the error, Grübel and Heresmeier came up with the following connection: if we want to calculate the compound probabilities on $\{0, 1, \dots n\}$ namely $P(S_N = 0), \dots P(S_N = n)$, let q^M be the truncated sequence of q and suppose the frequency distribution has finite variance, then

$$\sum_{n=0}^{M-1} |q_n^M - q_n| = \sum_{n=M}^{\infty} q_n - g'_N(1) \sum_{n=M}^{\infty} f_n + \mathcal{O}\left(\left(\sum_{n=M}^{\infty} f_n\right)^2\right). \tag{4.4}$$

The left side of the equation describes the aliasing error, while the right hand side is put together from the tail of the compound distribution, namely $\sum_{n=M}^{\infty} q_n$ and the tail from the individual claim heights. A detailed derivation of the individual terms can be found in [17]. The overall conclusion of (4.4) is, that M should be chosen much larger than n in order to obtain appropriate results with a small aliasing error. Since this statement only holds if the tail of the distribution decays fast, it's a mandatory feature in order to ensure an apt result. Grübel and Heresmeier were able to come up with a tilting procedure in order to make sure that the tail of each utilized distribution satisfies this requirement.

4.2.3. Algorithm with tilting parameter

Under the consideration of the previously drawn conclusions, the original algorithm is refined by the introduction of a tilting parameter $\theta > 0$ and

$$E_\theta f = \left(e^{-\theta j} f_j\right)_{j=0,1,\dots M-1}.$$

This adaption guarantees that tail of $E_\theta f$ decays at an exponential rate and thus much faster than the originally considered f. This leads to the following algorithm:

Algorithm 4.2.2.

1. Choose a truncation point $M \in \mathbb{N}$ and a tilting parameter $\theta > 0$.
2. Set $f = (f_0, f_1, \ldots, f_{M-1}) \in \mathbb{R}^M$.
3. Tilt the sequence: $f \mapsto E_\theta f = (e^{-\theta j} f_j)_{j=0,1,\ldots,M-1}$.
4. Calculate the DFT $\widehat{E_\theta f}$.
5. Apply the generating function of N to the result and set $g_N(\widehat{E_\theta f}) =: \hat{q}$.
6. Take the inverse DFT in order to obtain q as an approximation for the distribution of the total claim amount.
7. Untilt by applying $E_{-\theta}$.

The applied tilting procedure is of much practical value and can reduce the aliasing error tremendously. However one has to pick the utilized tilting parameter carefully since an under- or overflow might occur throughout the calculation. This might happen for an arbitrarily large chosen θ for example. A rough guideline on how to choose the parameter was suggested in [17]: the minimal and maximal values arise in the steps 3. and 5. and are of the size $\exp(\pm\theta M)$. Hence a value of $\theta M \approx 20$ will in general not lead to numerical difficulties with a common (64bit) double precision.

Example 4.2.3. Visualization of the aliasing error

According to [2], a general example is presented in order to visualize the wrap-around effect, which might occur if the FFT-based algorithm is considered without a tilting parameter θ. To do so, the general assumptions are defined: we choose $N \sim \text{Poi}(20)$ and $X \sim \text{Pareto}(4, 3)$. The Pareto distribution is given by

$$f(x) = \frac{\alpha}{\beta}\left(1 + \frac{x}{\beta}\right)^{-(\alpha+1)}$$

with $x \geq 0$, $\alpha = 4 > 0$ and $\beta = 3 > 0$ in this specific example. The rounding method is chosen in order to discretize the individual claim heights X with mesh size $h = 0.1$, considering the range $[0, h, 2h, \ldots (M-1)h]$. We set $M = 256$ as truncation point and choose the tilting parameter $\theta = 0.01$ relatively small, such that the wrap-around effect becomes clearly visible. These assumptions allow us to execute both FFT algorithms, as well as the Panjer recursion. The utilized, user-defined R code for the FFT-based approaches can be found in appendix B. Before plotting the final results one has to deal with two minor things: the obtained results are likely to be complex with imaginary parts due to occurring rounding errors, hence one only consider the real parts. Additionally the results are divided by h in order to derive an approximation for the compound density on $(0, (M-1)h]$. These considerations imply the following result:

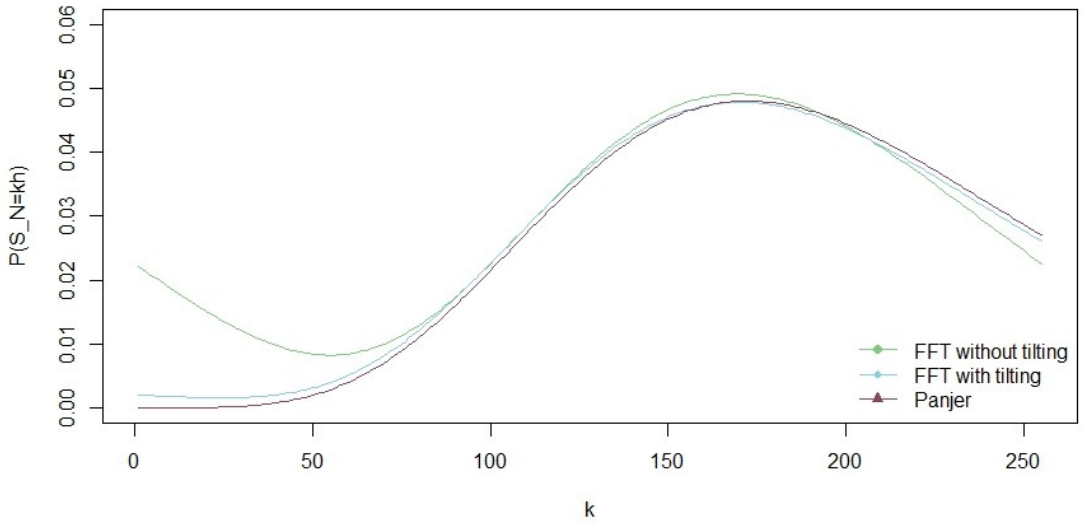


Figure 4.1.: Visualization of aliasing error

The wrap-around effect becomes clearly visible in the interval $[0, 75]$, where a large numerical difference between the different approaches occurs. If θ would have been picked according to the rule of thumb by Grübel and Heresmeier in [17], namely $\theta \approx 20/M \approx 0.1$ the difference between the tilted result and the Panjer recursion will no longer be visible. The absolute difference would be less than ≈ 0.0005 in the observed range. The following table yields the exact numerical values for both $\theta = 0.01$ and $\theta = 0.1$ in order to underline the drawn conclusions:

k	**Panjer recursion**	**Algorithm** 4.2.1	**Algorithm** 4.2.2 $\theta = 0.01$	**Algorithm** 4.2.2 $\theta = 0.1$
1	$6.183461e^{-8}$	0.02218828	0.002012904	$6.222214e^{-8}$
2	$1.401403e^{-7}$	0.02179226	0.001987839	$1.481059e^{-7}$
2	$2.783654e^{-7}$	0.02139838	0.001962983	$2.782164e^{-7}$
...				
49	0.001838579	0.008491519	0.002991653	0.001905342
50	0.001990036	0.008417497	0.003136664	0.002003394
51	0.002150051	0.008356833	0.003290767	0.002568951
...				
99	0.02093230	0.02199997	0.02206724	0.020445384
100	0.02150081	0.02255468	0.02263223	0.0211238
101	0.02207166	0.02311414	0.02319908	0.022396167

k	Panjer recursion	Algorithm 4.2.1	Algorithm 4.2.2 $\theta = 0.01$	Algorithm 4.2.2 $\theta = 0.1$
...				
174	0.04800009	0.04900417	0.04775446	0.048001346
175	0.04797742	0.04894368	0.04770945	0.04797564
176	0.04794442	0.04887147	0.04765436	0.04794159
...				
249	0.02902649	0.02501246	0.02807862	0.02901572
250	0.02869084	0.02460408	0.02774572	0.02869105
251	0.02835630	0.02419719	0.02741413	0.02835692

Figure 4.2.: Numerical results for different θ with $N \sim \text{Poi}(20)$ and $X \sim \text{Pareto}(4, 3)$

As overall conclusion we can keep in mind that it would be beneficial to stick to the thumb rule of Grübel and Heresmeier, if it comes to choosing an appropriate θ for the algorithm.

4.3. Quality of the algorithm

To conclude the theoretical section about the FFT-based approach to determine the distribution of the total claim amount, the quality of the presented ansatz has to be investigated even further. For either of the presented FFT algorithms it's possible to choose any claims count and severity distribution. The selection depends only on the scope of application since there are no restrictions which have to be kept in mind in contrast to the Panjer recursion and its limited possibilities of choices for the claims count distribution. The complexity of the calculations was increased even further: in order to obtain the probabilities $q_0 = P(S_N = 0), \dots\ q_n = P(S_N = n)$ algorithm 4.2.2 needs $\mathcal{O}(n \log(n))$ operations, while the recursion takes $\mathcal{O}(n^2)$.
Hans Bühlmann investigated in his paper *„Numerical Evaluation of the Compound Poisson Distribution: Recursion or Fast Fourier Transform?“* [18] the differences between both approaches in detail, considering a compound Poisson distribution. He stated that for $n \geq 256$ FFT basically always beets the Panjer recursion. This statement must be taken with care, since it was drawn in 1984 and today's processing powers enable the corresponding calculations for both approaches in split seconds, if implemented in the best possible way. However in a realistic setup, with potentially heavy-tailed distributions and several (ten) thousand calculated knots, the Panjer recursion will be outperformed by far. A numerical example for the FFT algorithm and the Panjer recursion is presented in the following chapter.

5. Comparison

In order to conclude the examination of the Panjer recursion and the FFT algorithms, the subsequent section presents a few more practical applications to point out the obtained theoretical differences.

5.1. Numerical examples

For a better understanding of the presented theory, some specific examples are shown in order to draw attention to special characteristics of the different approaches. The first example can be seen as an extension of 4.2.3, where the effect of the aliasing error was shown and a rule of thumb to choose θ was suggested. It was indicated that a fitting choice of the tilting parameter implies similar results from both Panjer recursion and the tilted FFT algorithm. To verify this statement, the following example is presented:

Example 5.1.1. Accordance between the three approaches with correctly chosen tilting parameter θ
The subsequent parameters are used in order to calculate the single probabilities of the total claim amount S_N: let $N \sim \text{Poi}(15)$ and $X \sim \Gamma(20, 12)$. Additionally we set $h = 0.5$, $M = 100$ and $\theta = 20/M = 0.2$. Applying the code given in appendix A and B, yields the following result:

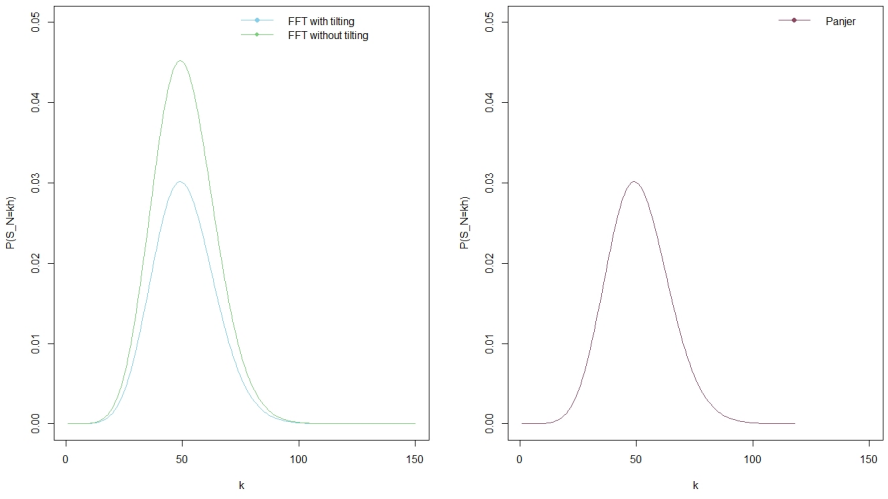


Figure 5.1.: Comparison of FFT with/without tilting and the PR

The plots indicate that the outcome for both the Panjer recursion and the tilted FFT algorithm are quite similar, while the untilted approach yields a very different result. The exact numerical values for a couple of selected iterations are given in the following table:

k	**Panjer recursion**	**Algorithm** 4.2.1	**Algorithm** 4.2.2
1	$3.059023e^{-7}$	$4.592373e^{-7}$	$3.059023e^{-7}$
2	$4.845280e^{-9}$	$7.550209e^{-9}$	$4.845280e^{-9}$
3	$5.677560e^{-7}$	$8.518413e^{-7}$	$5.677560e^{-7}$
...			
14	0.0002149132	0.0003223698	0.0002149132
15	0.0002945491	0.0004418237	0.0002945491
16	0.0004127620	0.0006191430	0.0004127620
...			
49	0.03016026	0.04524039	0.03016026
50	0.03009777	0.04514666	0.03009777
51	0.02986828	0.04480242	0.02986828
...			
99	$1.376731e^{-4}$	0.0002065096	$1.376731e^{-4}$
100	$1.133669e^{-4}$	0.0001700504	$1.133669e^{-4}$
101	$9.309926e^{-5}$	0.0001396489	$9.309926e^{-5}$
...			

Figure 5.2.: Exact results for $N \sim$Poi(15) and $X \sim \Gamma(20, 12)$

Under the exemplary assumptions and the given precision, the Panjer recursion and the tilted algorithm yield the exact same results. The untilted algorithm however differs quite a lot from the other two approaches. For $k = 50$ the difference becomes the largest with the tilted value being ~ 0.015 smaller than the untilted one. These differences can be traced back to the previously introduced aliasing error, since truncated data gets wrapped around and incorrectly appears in the observed range.

Under the consideration of this example one can keep in mind that an appropriate choice of θ directly implies a high correspondence between the tilted FFT algorithm and the Panjer recursion. Hence both algorithms are equally applicable for a small number of calculated single probabilities $q_N = P(S_N = n)$. However if n gets tremendously increased or if several 1000 lattice points are introduced throughout the discretization, the complexity of the approaches has to be taken into accordance. This leads to the recursion being outperformed by the tilted FFT algorithm in terms of calculation speed by far. So far we never gave a proof or theoretical justification for such statements about the complexities obtained by the different approaches. The following example will use an open-source *R*-package in order to determine the complexity of the code given in A and B.

Example 5.1.2. Determination of the complexity
In order to analyse the convolution, the Panjer recursion and the tilted FFT algorithm, we use the *GuessCompx*-package, which is an open-source *R*-package used for the empiric estimation of the complexity of some (user-defined) algorithm in terms of running time and memory. A detailed documentation of the package can be found online, compare [19]. Prior to our calculations we give a short introduction to the utilized *CompEst* function and point out some helpful features. It's general form is given by:

```
CompEst(f, g, random.sampling = FALSE, max.time = 30,
                    replicates = 4)
```

For a better understanding we start with some of the most important input variables:

- f - data on which the algorithm is tested, can be a vector or matrix
- g - the (user-defined) function which takes f as only input variable
- *random.sampling* - boolean value, if it's TRUE a random sample is used at each step of the calculation, if it's FALSE it uses the first N observations at each step instead
- *max.time* - maximum time allowed for each step of the calculation, default is 30 seconds
- *replicates* - number of replicated runs of the algorithm for each sample size, default is 2

Note that there are a couple more input variables available, which might become important for other applications of the function, however in our case it's enough to consider the previously presented ones. If we execute *CompEst* with some appropriate input the given function g is run on a set of increasing datasets and subsequently determines the complexity trend, which is most likely to fit the algorithm. Possible solutions are:

- $\mathcal{O}(1)$ - constant
- $\mathcal{O}(n)$ - linear
- $\mathcal{O}(n^2)$ - quadratic
- $\mathcal{O}(n^3)$ - cubic
- $\mathcal{O}(\sqrt{n})$ - squareroot
- $\mathcal{O}(\log(n))$ - log
- $\mathcal{O}(n\log(n))$ - NlogN

Additionally one obtains the estimated running time and memory usage for the full sample of f. These performance characteristics yield a rather good impression of the given code and enable a first idea of the detailed utilization possibilities.
For our examination we will consider the following model assumptions:

$$N \sim \text{Poi}(20), \qquad X \sim \text{Pareto}(4, 3)$$

and apply the rounding method with step size $h = 0.5$ for the discretization. Since the observed function g allows only one input variable, namely the vector f, we have to refine our given functions *Panjer.Poisson* and *FFT.tilting* slightly, such that they only take a vector

of single probabilities as input. The other necessary parameters such as λ, M and θ will be defined in each of the functions seperatly for the duration of this example. With these preparations done, we start by examining the convolution based approach, which was introduced in Chapter 2.4.2:

$$F^{S_N}(x) = P(N = 0) + \sum_{k \in \mathbb{N}} P(N = k) F_X^{*k}(x), \tag{5.1}$$

Equation (5.1) was implemented as an additional *Convolution(f)* function in *R* such that we can determine its complexity by applying *CompEst*. As sample we chose different-sized vectors of Pareto-distributed and discretized claim heights. These assumptions yield the following graphical output:

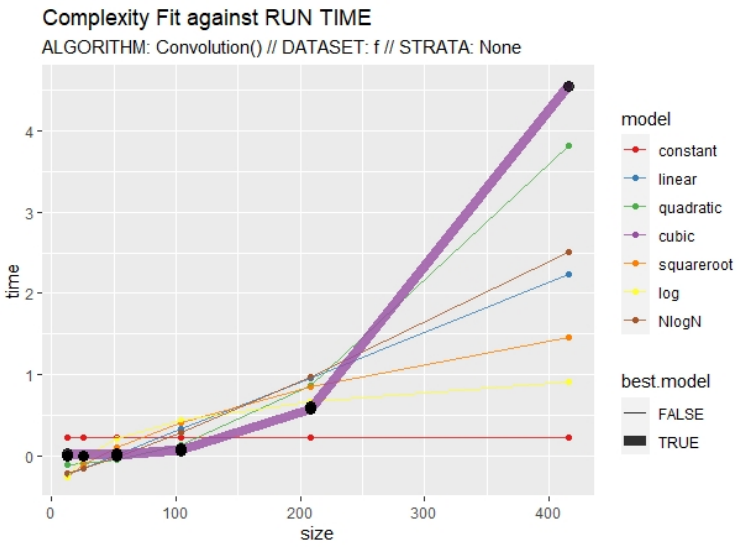


Figure 5.3.: Complexity of the convolution

The bold line describes the complexity of the given *Convolution* function. As suggested throughout previous examinations, we obtain $\mathcal{O}(n^3)$ as result. For a better classification other complexity types, such as linear or logarithmic, are plotted for the observed range as well. This enables a better understanding of the impacts of the obtained result, which is rather inopportune for an excessive application of the function.
Next up we will investigate the Panjer recursion under the consideration of the given model assumptions, which leads to:

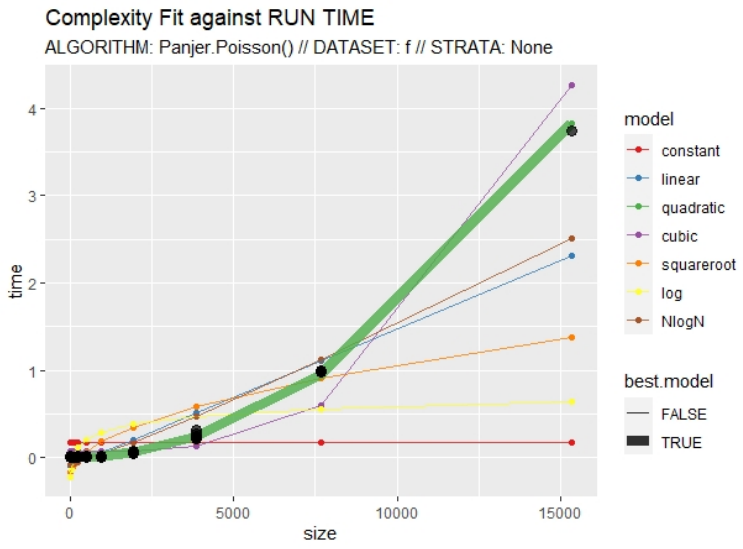


Figure 5.4.: Complexity of the Panjer recursion

We obtain $\mathcal{O}(n^2)$ as result and an overall runtime for the whole dataset of 47.12 seconds. Note that the utilized sample was tremendously increased in order to verify the statement from Section 4.3 that the recursion gets outperformed by far from the FFT-based approach if several ten thousand lattice points are introduced within the calculation.
In closing we run the *CompEst*-function with the tilted FFT algorithm as input:

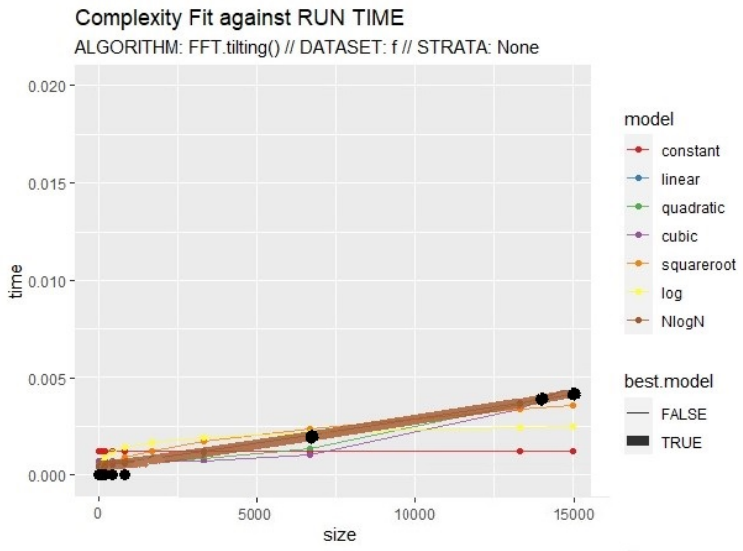


Figure 5.5.: Complexity of the convolution

The complexity is given by $\mathcal{O}(n \log n)$ and the complete dataset of 15.000 observations can

be evaluated in 0.2 seconds.
This result proves our previously drawn conclusions that the FFT-based algorithm outperforms the Panjer recursion in terms of running time and complexity by far!

After we showed that the expected complexities of the different approaches are in accordance with the user-defined R-functions, we focus on another important part of the previous calculations: the discretization. We already saw that the rounding method is in general the best one to choose, however we never made a statement about the choice of the bandwidth h. In Chapter 3.3 we already indicated that there is no formula to categorize the arising discretization error, hence the size of h was more of an educated guess so far. The next example is concerned with the question of choosing h appropriately and compares the obtained results for each of the available methods for different bandwidths. The examination of h is based on [2, Ch. 3.2].

Example 5.1.3. Bandwidth Choice
Let $N \sim \text{Poi}(25)$ and $X \sim \text{Exp}(2)$, by applying the rounding method, forward- and backward-difference to the assumptions, we obtain:

Discretization method	**h = 1**	**h = 0.5**	**h = 0.2**	**h = 0.1**	**h = 0.05**	**Exact**
Forward	10	15	18.6	20.2	20.68	21.8
Rounding	20	21.5	21.8	21.8	21.8	
Backward	44	31.5	25.4	23.5	22.72	

Figure 5.6.: 99%−quantile of different bandwidths and discretization methods for $N \sim \text{Poi}(25)$ and $X \sim \text{Exp}(2)$

Overall we can see that the forward- and backward-difference have a rather pessimistic way to discretize the given compound distribution, something Example 3.4.3 already showed. The rounding method yields already for the disadvantageous choice of $h = 1$ a result quite similar to exact value, while the two other methods show a clear difference. A simple rule to choose the bandwidth, derived from the results of this example, would be using the bounds from the forward- and backward-difference to decrease h until their difference is less than an appropriately chosen value. However this makes it necessary to choose such an acceptable margin of error, which is once more not obvious. As a result it would be beneficial to choose h suitably small, apply the rounding method and compute the distribution until the difference between the obtained values is smaller than some threshold.

These examples complete the examination of the Panjer recursion and a FFT-based approach and their application in collective risk modeling. The following conclusion recapitulates the most important assertions of the previous chapters, which have to be kept in mind in order to successfully derive the distribution of the total claim amount under the consideration of the given model assumptions.

Conclusion 5.1.4. Summary of necessary (parameter) choices for the different methods in practical applications
If the model assumptions 2.1.1 are satisfied and the single probabilities of the total claim amount S_N are sought-after, the subsequent parameter choices and characteristics have to be kept in mind, depending on the utilized approach:

- **Choice of an appropriate discretization method and bandwidth h**:
  The rounding method yields the most convenient approximations. For the bandwidth there is no fixed size, which automatically implies the „best“ result. Instead h should be decreased step by step until the relative improvement of the obtained results is smaller than some given threshold.
- **Panjer recursion**:
  A recursive calculation is feasible if the mean of the frequency distribution is not too large and the discretization method has a small number of lattice points. Otherwise computational issues might arise. The limited options for the claims count distributions can be expanded, if a higher order Panjer recursion is considered.
- **Tilted FFT algorithm**:
  Alternative approach whenever the recursion reaches its limits. The truncation point M and the tilting parameter have to satisfy $\theta \approx 20/M$ in order to obtain decent results. The utilized distributions can be chosen completely arbitrarily, depending on the scope of application.

With the previous conclusion we stated the most important aspects for an effective practical application of the different methods. Additionally the following table will complete the underlying theoretical aspects and give a short overview of the derived characteristics:

	Convolution 2.4.2	**Panjer recursion** 3	**FFT algorithm** 4
Complexity	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n \log n)$
Error bound	–	*yes* 3.5	*yes* 4.3
Extendability	–	*yes* 3.6	–
Scope of application	universal	restricted 3.1	universal

Figure 5.7.: Comparison of the convolution, Panjer recursion and the FFT algorithm

The associated theory and it's chapters are given below the corresponding statements.

5.2. Conclusion

The previous chapters gave a theoretical overview of the main approaches for determining

the distribution of the total claim amount S_N in a collective risk model. For an „exact“ evaluation of the corresponding compound distributions the Panjer recursion is the most common method in actuarial science, however FFT is a feasible alternative. The main drawbacks of the recursion are the limited choices in terms of claims count distribution as well as computational difficulties, which might arise as soon as a large number of lattice points is considered. The FFT-based algorithms enable an more efficient way to determine the desired single probabilities, whilst additionally allowing arbitrary frequency distributions. Especially the use of exponential tilting, which eliminates the „wrap-around“ effect, simplifies the application tremendously and allows for example the examination of the lower tail of a frequency, something quite costly for the Panjer recursion.

6. Outlook: possible applications of machine learning in non-life insurance

For the insurance industry it is mandatory to constantly ask itself the question whether its' utilized mathematical approaches are sufficient to stay in the picture on such a highly competitive market. Overall one can think of the insurance business as a business of data processing, which is consistently affected by the digitalization and the changing nature of data. Since it becomes more and more convenient to utilize different types of data in order to effectively determine appropriate insurance contracts, the insurers themselves have to be able to effectively analyze and handle extremely large amounts of data. Under these circumstances machine learning (ML) algorithms are taken into consideration and find a broad scope of application as shown by Grize et al. [20] for example.
Before a general idea of such methods in collective risk modeling is given, a definition of the terminology „machine learning“ is necessary. However this is not as obvious as one might think: ML summarizes a broad range of different algorithms such as classification and regression trees, as well as different types of learning. A schematic classification of the different methods can be displayed as follows:

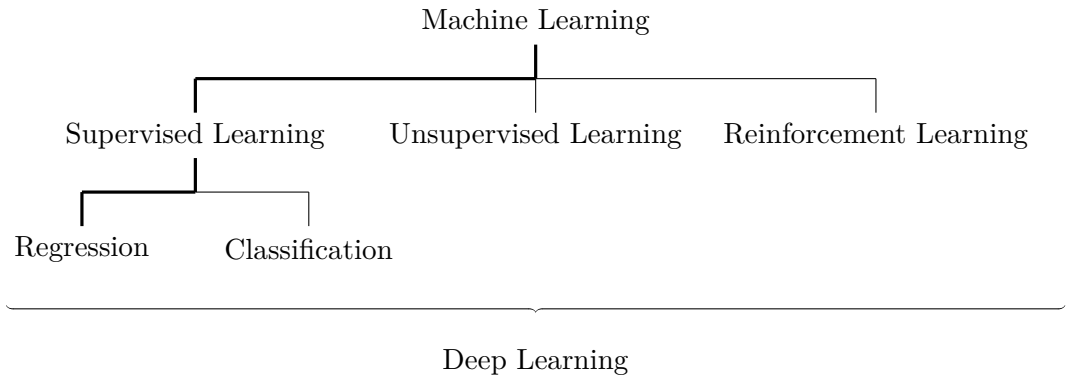


Figure 6.1.: Simplified classification of different machine learning methods, [21]

One has to adhere that an overall definition of ML is almost not possible due to the wide range of constantly changing and expanding branches. However Grize suggested in [22] that the following three characteristics distinguish ML from other quantification methods:

- The calculation is done in an automatized way.

- Large datasets, that might consist of different data types and/or many dimensions are considered.
- All within a short computing time.

A more detailed description of machine learning is usually based on the particular scope of application.
In actuarial science for example we are especially interested in regression models, since many common insurance problems can be expressed in that way. That's why the corresponding branch is printed bold in the above depiction 6.1. The following exemplary actuarial problems yield a first idea of possible applications of ML in non-life insurance:

- **Premiums and pricing**
  The determination of individualized premiums is of great interest for the insurance companies since they have to stay competitive in a contested market. To do so one has to take several hundreds or thousands aspects into account in order to assess the actual risk for an individual person and derive an insurance contract.
- **Claims management**
  The processing of occurring claims plays another important role, since the number of claims is usually way too large in order to evaluate each of them by hand. Therefore it's important to automatize the classification and a possible fraud detection.
- **Marketing**
  Since customer retention is of great interest within an insurance company, it's necessary to evaluate the customers' individual behavior in order to offer personalized insurance contracts.

The following chapter will focus on the application of machine learning tools in the framework of modeling claim amounts and claim heights. However this is done in a merely superficial way in order to avoid overloading the thesis. A detailed analysis can be found in *„Data Analytics for Non-Life Insurance Pricing“* by Wuthrich and Buser [22] instead.

6.1. Supervised learning and generalized linear models

It was already indicated that supervised learning and regression problems are the most familiar part of machine learning from an actuaries' point of view. It is all about making predictions about a target variable Y from p describing variables $X_1, \dots X_p$. Based on the book *„Actuarial Data Science“* by Seehafer et al. [23, Ch. 6.3] the basics of regression problems are assessed subsequently. In order to describe the hitherto unknown connection between Y and X an appropriate link function f is introduced:

$$Y = f(X) + \epsilon. \tag{6.1}$$

The term $\epsilon \in \mathbb{R}^n$ represents a disturbance variable, which is usually additive. Non-additive disturbance variables can be changed into additive ones by transforming y suitably. The

overall goal is the derivation of an ideal estimator $\hat{f}$ for our function f, in such a way that the separation between f and ϵ is as strict as possible. The observations x_{ij} of X are described in a so-called *feature matrix* $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$:

$$\begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \dots & \vdots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{pmatrix}$$

At this point we introduce so-called *generalized linear models* (GLMs), which find a broad application in insurance applications and can also be applied in the environment of supervised learning. By considering a linear regression model, we can rewrite (6.1) in such a way, that

$$Y = \mathbf{X}\beta + \epsilon$$

holds, if the following aspects are satisfied:

1. The disturbances are 0 on average, $\mathbb{E}[\epsilon] = 0$.
2. The variance of ϵ is constant on average, $\mathrm{Var}(\epsilon_i) = \sigma^2$.
3. The disturbances have to be uncorrelated, such that $\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0$ holds for $i \neq j$.

Under these assumptions we can derive the following properties of Y_i:

$$\begin{aligned} \mathbb{E}[Y_i] &= \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} \\ \mathrm{Var}(Y_i) &= \sigma^2 \\ \mathrm{Cov}(Y_i, Y_j) &= 0 \qquad\qquad \text{for } i \neq j \end{aligned}$$

[24] yields appropriate theory in order to obtain estimators for the model coefficient

$$\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$$

and the variance parameter

$$\hat{\sigma}^2 = \frac{1}{n-(p+1)}\sum_{i=1}^{n}\left(Y_i - \left(\mathbf{X}\hat{\beta}\right)_i\right)^2.$$

These derivations are a classic example for a *linear regression model*, which is only one of many different regression models available. Depending on the exact scope of application we could also use a Poisson or Gamma regression, a logistic or a Bayesian linear regression, to name only a few alternatives. For more details we refer to the paper *„AI in Actuarial Science“* [21] by Richman, which introduces several possible applications of supervised learning in the framework of collective risk modeling.
The following table gives a first idea of the structure of Y and $\mathbf{X}$ depending on present applications.

	Description	Feature matrix X	Output vector Y
1	Short-term pricing	Policyholder details (age, gender, credit score, postal code, details of the insured item)	Frequency and/or severity of the claims or the pure premium
2	IBNR model	Accident and reporting year	Claims amount at time t
3	Mortality estimation	Age and gender	Number of deaths in different age groups

Figure 6.2.: Feature matrices $\mathbf{X}$ and outputs Y for several problems in actuarial science

Since not all examples were a subject of the previously treated non-life insurance cases, we give a brief explanation of each of the issues for a better understanding:

1. **Short-term pricing**
   One talks about short-term insurances, if they cover ones possessions. It refers for example to car insurances, travel or home insurances. Those policies are priced by independently considering the frequency and the severity with GLMs.

2. **IBNR models**
   A main aspect for insurance companies is the evaluation and estimation of so-called IBNR claims, which are incurred but not reported within the year of them actually occurring. Such claims bring an extra uncertainty to the calculation of appropriate premiums and safety margins for the insurers. These IBNR models have to be much more advanced than the models suggested for short-term pricing, wherefore one can apply generalized linear mixed models (GLMMs) or Bayesian hierarchical models instead. They enable in general the derivation of a statement about the cumulative claim amount at a certain time t.

3. **Mortality estimation**
   The last example is taken from the subject of life-insurances: a research area, which benefits tremendously from the newest possibilities enabled by machine learning. The main goal is the development of a mortality table with survival probabilities for each age and gender. Based on such tables the insurance companies are able to calculate appropriate premiums for their insurance contracts. Since a lot of external aspects like climate, the quality of life or the availability and accessibility of doctors have to be considered for each potential customer on its own, machine learning enables an efficient way to analyze and interpret these large amounts of data.

6.2. Estimating severity and claim frequency with regression models

Throughout the previous example we already indicated that GLMs enable the individual evaluation of severity and claim frequency in a collective risk model. Wuthrich and Buser suggested in [22, Ch. 2] a Poisson rate regression for the number of claims and a Gamma regression for the height of the individual claims. For what reason we are going to present both models in the following subsections with a corresponding example.

6.2.1. Poisson regression

The so-called Poisson regression, also known as log-linear model, is a certain form of a GLM, which is especially well-suited for modeling the number of claims.
These models are based on the count data Y_i and the describing variables $\underline{x}_i$. The goal of the Poisson regression is an appropriate modeling of

$$\mathbb{E}[Y_i|\underline{x}_i].$$

In order to ensure a better understanding of the theory behind the model, an example is introduced to accompany the drawn results.

Example 6.2.1. Shipwrecks
We utilize the *R*-package *MASS*, which was created in order to support Venables and Ripleys book *„Modern Applied Statistics with S“* [25]. It provides datasets and functions in order to illustrate their examples.
For the subsequent example we are interested in the *ship* dataset, which yields some information about a fleet of ships, their damage incidents and aggregate months of service for different ship types. The following data is given:

- Y describes the number of defects
- X includes
  - the months of service M,
  - ship type $T \in \{A, B, C, D, E\}$,
  - year of construction $C \in \{60, 65, 70, 75\}$ and
  - the period of operation $P \in \{0, 1\}$, whereby we differ between $1960-1974$, denoted by 0 and $1975-1979$ denoted by 1.

The following table provides a small part of the data:

	T	**C**	**P**	**M**	**Y**
1	A	60	0	127	0
2	A	60	1	63	0
3	A	65	0	1095	3
⋮					
33	E	70	1	2161	12
34	E	75	1	542	1

Figure 6.3.: Ships damage data

Based on this data we now try to find a connection between the number of occurring claims for the different ship types. We would like to make statements such as „the probability that a ship of type E has a damage is ??% higher than for a ship of type B“for example. To do so, we have to continue with the theoretical description of the Poisson regression model, before we come back to the shipwreck dataset.

In order to derive a fully defined model from the presumptions we assume that Y_i follows a Poisson distribution, namely $Y_i \sim \text{Poi}(\lambda_i)$ with

$$\mathbb{E}[Y_i] = \lambda_i \quad \text{and} \quad \text{Var}(i) = \lambda_i,$$

whereby $\lambda_i > 0$ holds. We therefore want to model λ_i in dependence on $\underline{x}_i$ under the consideration of the inequality restriction $\lambda_i > 0$.
At this point we have to introduce a link function, which provides a connection between the linear predictor and the mean of the distribution function. Such functions have to be chosen depending on the respective example. They assure in general that (inequality) restrictions concerning the expected value, are satisfied throughout the calculations. Some common ones are for example:

- *Identity link function*: $g(\mu) = \mu$ if $\mathbb{E}[Y]$ is arbitrary or
- *Logit link function*: $g(\mu) = \log(\frac{\mu}{1-\mu})$ if $0 < \mathbb{E}[Y] < 1$.

In our case we can choose the *logarithmic link function* in order to incorporate our inequality restriction. This leads to

$$\begin{aligned} \log(\lambda_i) &= \underline{x}_i^T \underline{\beta} \quad \in \mathbb{R} \\ \Leftrightarrow \qquad \lambda_i &= e^{\underline{x}_i^T \underline{\beta}} \quad > 0. \end{aligned} \tag{6.2}$$

The parameter β can be estimated with the maximization of the log-likelihood function for example, which leads to the following Poisson regression model::

$$\mathbb{E}[Y_i|\underline{x}_i] = e^{\beta_0} * e^{\beta_1 \underline{x}_i^{(1)}} * \cdots * e^{\beta_m \underline{x}_i^{(m)}}.$$

The result enables an important conclusion: we can easily see that the addition of another parameter $x^{(j)}$ leads to the result being multiplied with e^{β_j}. Which in turn results in an increase of the expected value if β_j is positive and a decrease if β_j is negative. These relations enable a comparison of the occurring damages in our example based on different $\underline{x}_i$. We are therefore able to determine which ship type has a higher probability of an occurring defect compared to the other ones.

Example 6.2.2. Shipwrecks 2
In practice one might reason that the number of damages is proportional to the number of months the ship is traveling. This would suggest to use Y_i/M_i as regression rate. However it's not possible in our case, since Y_i/M_i does not follow a Poisson distribution.
Equation (6.2) has therfore to be adapted slightly to:

$$\begin{aligned} & \lambda_i = M_i * e^{\underline{x}_i^T \underline{\beta}} \\ \Leftrightarrow \quad & \log(\lambda_i) = \log(M_i) + \underline{x}_i^T \underline{\beta}. \end{aligned} \tag{6.3}$$

The remodeling is based on the fact, that we do not have to estimate an additional parameter for M_i in this specific example! The *R*-code is given below:

```
# Load data
data(ships, package="MASS")
x.ships        <- ships

# Remove structural zeroes
x.ships <- subset(x.ships, subset=(service > 0))
```

With the data loaded and stored appropriately, the Poisson regression model can be calculated with the *glm*-function by setting „*family=poisson*“. In addition to that specification we have to pay attention to our refinement (6.3), whereby we requested that M_i should be excluded from the parameter estimation itself. Therefore we set „*offset = log(service)*“, which leads to the subsequent code fragment:

```
# Poisson regression model
fit <- glm(incidents ~ type + period + year,
           offset = log(service),
           family = poisson(link= "log"),
           data = x.ships)
```

And the following result:

```
Call:
glm(formula = incidents ~ type + period + year, family = poisson,
    data = x.ships, offset = log(service))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6768  -0.8293  -0.4370   0.5058   2.7912
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.40590    0.21744 -29.460  < 2e-16
typeB       -0.54334    0.17759  -3.060  0.00222
typeC       -0.68740    0.32904  -2.089  0.03670
typeD       -0.07596    0.29058  -0.261  0.79377
typeE        0.32558    0.23588   1.380  0.16750
period75     0.38447    0.11827   3.251  0.00115
year65       0.69714    0.14964   4.659 3.18e-06
year70       0.81843    0.16977   4.821 1.43e-06
year75       0.45343    0.23317   1.945  0.05182
```

The obtained results use ship type A as a reference. If we want to compare type B with type E for example, we obtain

$$\exp(-0.54334 - 0.32558) = 0.4194.$$

Which means that ship type E is likely to have only 41% of the damages type B might suffer from. If we compare type B with C instead, the calculation yields

$$\exp(-0.54334 - (-0.68740)) = 1.1549$$

which is equivalent to B having 15% more damages than type C. Another comparison of A and D leads to:

$$\exp(-0.07596) = 0.9268$$

and hence a quite similar probability of an occurring damage for those two types.

This example gave a general idea of a possible application of a Poisson regression model in order to obtain an estimation for the number of claims, derived from some given description variables X. Since the section was built upon this particular application, additional theoretical background about Poisson regression models was skipped. Hence we refer to [23] by Seehafer et al. for a complete analysis of such Poisson regression models. For an even more detailed introduction to generalized linear models, link functions and important special cases, we refer to *„Generalized Linear Models“* [26] by McCullagh and Nelder.

6.2.2. Gamma regression

Gamma regression models are often utilized to model costs, since they are similar to Poisson models, but do not require integers and are able to handle more dispersion. We assume

$$Y \sim \Gamma(\alpha, \beta)$$

with

$$\mathbb{E}[Y] = \mu = \frac{\alpha}{\beta} \quad \text{and} \quad \mathrm{Var}(Y) = \frac{\alpha}{\beta^2}.$$

As before we want to model the expected value of Y. In order to avoid overloading the analysis of generalized linear models, we will apply a logarithmic link function again. However one has to keep in mind that the choice of the link function depends on the exact scope of application and possible constraints. The log link function might therefore not be the ideal choice for other examples.
For the subsequent example, we use another *R*-package *HoRM* which was created to underline the book *„Handbook of Regression Methods“* [27].

Example 6.2.3. Canadian Automobile Insurance Claims
The dataset we are interested in, is from $1956-1957$ and was collected by the Statistical Unit of the Canadian Underwriters' Association. It includes data for private passenger automobile liability for non-farmers in Canada, whereby the province of Saskatchewan was excluded.
The data is divided up into 20 categories and 6 main variables:

- Merit - measures the number of years since the last claim was made in the corresponding policy. More specifically:
  - 3: licensed and accident free for ≥ 3 years
  - 2: licensed and accident free for 2 years
  - 1: licensed and accident free for 1 year
  - 0: everything else
- Class - collects among other things sex and marital status of the policy holder. For example:
  - 1: pleasure, no male operator
  - 2: pleasure, non-principal male operator
  - 3: business use
  - 4: unmarried owner or principal operator
  - 5: married operator or principal operator
- Insured - determines for how many years the policy is running.
- Premiums - describes the earned premium at present rates and is given in 1000's of Canadian dollars.
- Claims - number of incurred claims.
- Costs - amount of incurred losses (also measured in 1000's of Canadian dollars).

A part of the data is presented in the following table for a better understanding:

	Merit	Class	Insured	Premium	Claims	Cost
1	3	1	2757520	159108	217151	63191
2	3	2	130535	7175	14506	4598
3	3	3	247424	15663	31964	9589
⋮						
19	0	4	56730	2756	11345	3971
20	0	5	8601	461	1291	382

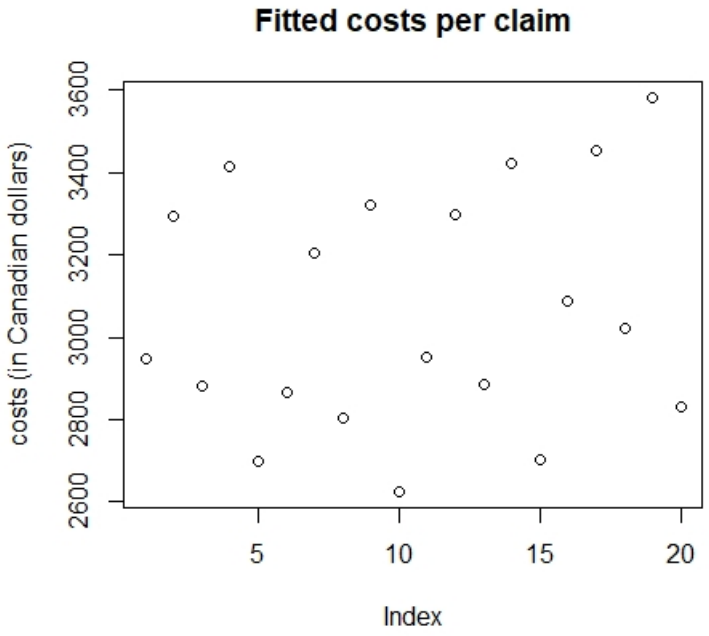
Figure 6.4.: Canadian auto insurance dataset

We start again by loaging the data and applying the *glm*-function as follows:

```
data(Auto)
x.auto <- Auto

fit <- glm(Cost/Claims ~ Merit + Class,
           family= Gamma(link = "log"),
           data = x.auto)
```

Since we are especially interested in the fitted costs per claim in each of the 20 categories, the following plot displays the average costs per claim for the different categories:



The exact results can be easily obtained by „*fit$fitted.values*“and are given below:

```
1         2         3         4         5         6         7
2945.698  3292.642  2881.179  3415.566  2699.566  2865.181  3202.643

8         9         10        11        12        13        14
2802.426  3322.207  2625.777  2950.659  3298.188  2886.032  3421.319

15        16        17        18        19        20
2704.113  3088.623  3452.402  3020.974  3581.290  2830.549
```

These results allow us to draw the following conclusions: the highest costs per claim are given in group 19, which contains drivers, who are unmarried and their last accident was within the last year. The opposite result, namely the lowest costs per claim are in group 10, where drivers with more than 2 years without accident, as well as married owners can be found.

Note that in real-life applications one has to take many more characteristics, such as age, type and value of the car into account to derive similar statements. Each characteristic has to be weighted in order to describe the influence on the final result. For our example this simplified version was sufficient in order to point out the possibilities of applying Gamma regression models for the costs in individual risk classes. *„Generalized Linear Models for Insurance Data“* [28, Ch. 8] yields a more detailed analysis of the presented theory and can be seen as an extension of the previous example.

6.3. Conclusion

We were able to give a short idea of a few applications of machine learning in the context of non-life insurance. However it became rather obvious that the broad possibilities of such methods cannot be summarized in such a short section, without the theory becoming too detailed for the actual framework. Hence we refer to the already cited books

- *„Generalized Linear Models for Insurance Data“* by de Jong and Heller [28]
- *„Handbook of Regression Methods“* by Young [27]
- *„Actuarial Data Science“* by Seehafer et al. [23]

if the reader is interested in a more detailed and specific analysis of the topic.

A. Panjer recursion

```
#####################################################################
#  Panjer.Poisson(lambda, f)
#
#  INPUT: lambda (parameter for the Poisson distribution)
#         and f (severity distribution)
#  OUTPUT: vector with the corresponding single probabilities
#          q[k] = P(S_N=k)
#
#  ATTENTION: R indexes from 1, so f[i]=P(X=i-1) -> f[1]=P(X=0)
#####################################################################

Panjer.Poisson <- function(lambda, f){
  # check, if necessary requirements are satisfied
  if (sum(f) != 1 | any(f<0)){
    stop("f is not a density.")
  }
  if(lambda * sum(f) > 727){
    stop("underflow.")
  }
  if(length(f)==3 & f[2]==0 & f[3]==1){
    stop("P(SN=0)=1")
  }

  s <- rep(0,100000)
  k <- 1

  # first term of the recursion
  cumul <- q <- f0 <- exp(-lambda*(1-f[1]))
  l <- length(f)

  # run the calculation until q[0]+q[1]+...+q[k]>0.99999 holds
  while(cumul < 0.99999){
    for(j in 1:min(l-1, k)){
      s[j] <- lambda/k*j*f[j+1]*q[k+1-j]
    }
    q <- c(q, sum(s))
    cumul <- cumul+sum(s)
    k <- k+1
  }
  return(q)
}
```

```
#########################################################################
#  Panjer.Bin(n, p, f)
#
#  INPUT: n, p (parameter for the Binomial distribution)
#         and f (severity distribution)
#  OUTPUT: vector with the corresponding single probabilities
#          q[k] = P(S_N=k)
#
#  ATTENTION: R indexes from 1, so f[i]=P(X=i-1) -> f[0]=P(X=0)
#########################################################################

Panjer.Bin <- function(n, p, f){
  # check, if necessary requirements are satisfied
  if (sum(f) != 1 | any(f<0)){
    stop("f is not a density.")
  }
  if(length(f)==3 & f[2]==0 & f[3]==1){
    stop("P(SN=0)=1")
  }
  if(p >= 1 | p <= 0){
    stop("p is not a probability.")
  }

  s <- rep(0,100000)
  k <- 1
  a <- p/(p-1)
  b <- p*(n+1)/(1-p)

  # first term of the recursion
  cumul <- q <- f0 <- ((a-1)/(a*f[1]-1))^((a+b)/a)
  l <- length(f)

  # run the calculation until q[0]+q[1]+...+q[k]>0.99999 holds
  while(cumul < 0.9){
    for(j in 1:min(l-1, k)){
      s[j] <- (1/(1-a*f0))*(a+j*b/k)*f[j+1]*q[k+1-j]
    }
    q <- c(q, sum(s))
    cumul <- cumul+sum(s)
    k <- k+1
  }
  return(q)
}
```

```
#####################################################################
#  Panjer.NB(r, p, f)
#
#  INPUT: r, p (parameter for the negative binomial distribution)
#         and f (severity distribution)
#  OUTPUT: vector with the corresponding single probabilities
#          q[k] = P(S_N=k)
#
#  ATTENTION: R indexes from 1, so f[i]=P(X=i-1) -> f[0]=P(X=0)
#####################################################################

Panjer.NB <- function(r, p, f){
  # check, if necessary requirements are satisfied
  if (sum(f) != 1 | any(f<0)){
    stop("f is not a density.")
  }
  if(length(f)==3 & f[2]==0 & f[3]==1){
    stop("P(SN=0)=1")
  }
  if(p >= 1 | p <= 0){
    stop("p is not a probability.")
  }

  s <- rep(0,100000)
  k <- 1
  a <- 1-p
  b <- (1-p)*(r-1)

  # first term of the recursion
  cumul <- q <- f0 <- (p/(1-f[1]+p*f[1]))^r
  l <- length(f)

  # run the calculation until q[0]+q[1]+...+q[k]>0.99999 holds
  while(cumul < 0.99999){
    for(j in 1:min(l-1, k)){
      s[j] <- (1/(1-a*f0))*(a+j*b/k)*f[j+1]*q[k+1-j]
    }
    q <- c(q, sum(s))
    cumul <- cumul+sum(s)
    k <- k+1
  }
  return(q)
}
```

B. FFT with and without tilting

```
#####################################################################
#  FFT(f, M, param, distN)
#
#  INPUT: f (sequence), M (truncation point), distN (the
#         distribution of N) and param (list with the parameters
#         of the severity distribution)
#  OUTPUT: vector with the the corresponding single probabilities
#          q[k] = P(S_N=k)
#####################################################################

FFT<- function(f, M, param, distN){
  fhat <- fft(f, inverse = FALSE)

  # apply the generating function of N to fhat
  if(distN == "Poi"){
    # param <- c(lambda)
    qhat <- exp(param[1]*(fhat-1))
  }
  if(distN == "Bin"){
    # param <- c(n,p)
    qhat <- (1-param[2]+param[2]*fhat)^param[1]
  }
  if(distN == "NB"){
    # param <- c(r, p)
    qhat <- (param[2]/(1-(1-param[2])*fhat))^param[1]
  }

  q <- 1/M*fft(qhat, inverse = TRUE)
  return(q)
}
```

```
#####################################################################
#  FFT.tilting(f, M, param, theta, distN)
#
#  INPUT: f (sequence), M (truncation point), distN (the
#         distribution of N) and param (list with the parameters
#         of the severity distribution)
#  OUTPUT: vector with the the corresponding single probabilities
#          q[k] = P(S_N=k)
#####################################################################

FFT.tilting <- function(f, M, param, theta, distN){
  fnew <- exp(-theta*(0:(M-1)))*f
  fhat <- fft(fnew, inverse = FALSE)

  # apply the generating function of N to fhat
  if(distN == "Poi"){
    # param <- c(lambda)
    qhat <- exp(param[1]*(fhat-1))
  }
  if(distN == "Bin"){
    # param <- c(n,p)
    qhat <- (1-param[2]+param[2]*fhat)^param[1]
  }
  if(distN == "NB"){
    # param <- c(r, p)
    qhat <- (param[2]/(1-(1-param[2])*fhat))^param[1]
  }

  q <- exp(theta*(0:(M-1)))*(1/M*fft(qhat, inverse = TRUE))
  return(q)
}
```

List of Figures

Bibliography

[1] Harry H. Panjer. "Recursive Evaluation of a Family of Compound Distributions". In: *ASTIN Bulletin* 12.1 (1981), 22–26. DOI: 10.1017/S0515036100006796.

[2] Paul Embrechts and Marco Frei. "Panjer recursion versus FFT for compound distributions". In: *Mathematical Methods of Operations Research* 69 (July 2009), pp. 497–508. DOI: 10.1007/s00186-008-0249-2.

[3] Klaus D. Schmidt. *Versicherungsmathematik*. 2009. URL: https://www.springer.com/de/book/9783642011757.

[4] Mario V. Wuthrich. *Non-Life Insurance: Mathematics & Statistics*. ETH Zurich, Dec. 17, 2020. URL: https://ssrn.com/abstract=2319328.

[5] Heinz-Willi Goelden et al. *Schadensversicherungsmathematik*. 2016. URL: https://www.springer.com/de/book/9783662488591.

[6] Sascha Desmettre. *Schadensversicherungsmathematik*. Lecture Notes. Johannes Kepler University Linz, Jan. 20, 2021.

[7] Fred Wagner. *Gabler Verischerungslexikon*. 2017. URL: https://www.springer.com/de/book/9783834946249.

[8] Hansjoerg Albrecher. *Finanz- und Versicherungsmathematik*. Lecture Notes. TU Graz, Feb. 2006. URL: https://www.math.tugraz.at/~aistleitner/Lehre/WS2011/Finanz_Vers/FinanzVersicherung20112012/skriptum_finanz_und_versicherungsmathematik.pdf.

[9] Bjørn Sundt and William S. Jewell. "Further Results on Recursive Evaluation of Compound Distributions". In: *ASTIN Bulletin* 12.1 (1981), 27–39. DOI: 10.1017/S0515036100006802.

[10] Christian Hipp. "Speedy Panjer for phase-type claims". In: (2003).

[11] Harry H. Panjer and Shaun Wang. "On the Stability of Recursive Formulas". In: *ASTIN Bulletin* 23.2 (1993), 227–258. DOI: 10.2143/AST.23.2.2005093.

[12] Harry H. Panjer and Willmot. "Computational aspects of recursive evaluation of compound distributions". In: *Insur. Math. Econ.* (1986).

[13] Klaus Th. Hess, Anett Liewald, and Klaus D. Schmidt. "An Extension of Panjer's Recursion". In: *ASTIN Bulletin* 32.2 (2002), 283–297. DOI: 10.2143/AST.32.2.1030.

[14] Nelson De Pril. "On the Exact Computation of the Aggregate Claims Distribution in the Individual Life Model". In: *ASTIN Bulletin* 16.2 (1986), 109–112. DOI: 10.2143/AST.16.2.2015003.

[15] Philip E. Heckman and Glenn G. Meyers. "The calculation of aggregate loss distributions from claim severity and claim count distributions". In: *Proceedings* PCAS LXXI (1983). URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.139.1118&rep=rep1&type=pdf.

[16] D. Sundararajan. *The Discrete Fourier Transform: Theory, Algorithms and Applications*. 2001. DOI: 10.1142/4610.

[17] Rudolf Grübel and Renate Hermesmeier. "Computation of Compound Distributions I: Aliasing Errors and Exponential Tilting". In: *ASTIN Bulletin* 29.2 (1999), 197–214. DOI: 10.2143/AST.29.2.504611.

[18] Hans Bühlmann. "Numerical evaluation of the compound Poisson distribution: Recursion or Fast Fourier Transform?" In: *Scandinavian Actuarial Journal* 1984.2 (1984), pp. 116–126. DOI: 10.1080/03461238.1984.10413759.

[19] Marc Agenis and Neeraj Bokde. *GuessCompx: Empirically Estimates Algorithm Complexity*. 2019. URL: https://CRAN.R-project.org/package=GuessCompx.

[20] Mario V. Wuthrich and Christoph Buser. *Data Analytics for Non-Life Insurance Pricing*. Oct. 27, 2021. URL: https://ssrn.com/abstract=2870308.

[21] Ronald Richman. "AI in actuarial science – a review of recent advances – part 1". In: *Annals of Actuarial Science* 15.2 (2021), 207–229. DOI: 10.1017/S1748499520000238.

[22] Yves-Laurent Grize, Wolfram Fischer, and Christian Lützelschwab. "Machine learning applications in nonlife insurance". In: *Applied Stochastic Models in Business and Industry* 36.4 (2020), pp. 523–537. DOI: https://doi.org/10.1002/asmb.2543. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asmb.2543. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/asmb.2543.

[23] Martin Seehafer et al. *Actuarial Data Science: Maschinelles Lernen in der Versicherung*. De Gruyter, 2021. ISBN: 9783110659344. DOI: doi:10.1515/9783110659344. URL: https://doi.org/10.1515/9783110659344.

[24] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, New York, NY, 2009. ISBN: 978-0-387-84857-0. DOI: https://doi.org/10.1007/978-0-387-84858-7.

[25] W.N. Venables and B.D. Ripley. *Modern Applied Statistics with S*. Springer-Verlag New York, 2002. ISBN: 978-1-4419-3008-8. DOI: https://doi.org/10.1007/978-0-387-21706-2.

[26] P. McCullagh and J.A. Nelder FRS. *Generalized Linear Models*. Chapman and Hall/CRC, 1989. ISBN: 978-0-412-31760-6. URL: http://www.utstat.toronto.edu/~brunner/oldclass/2201s11/readings/glmbook.pdf.

[27] D.S. Young. *Handbook of Regression Methods*. Chapman and Hall/CRC, 2017. DOI: https://doi.org/10.1201/9781315154701.

[28] Piet de Jong and Gillian Z. Heller. *Generalized Linear Models for Insurance Data*. Cambridge Books 9780521879149. Cambridge University Press, 2008. URL: https://ideas.repec.org/b/cup/cbooks/9780521879149.html.